



**UNIVERSIDAD DE CUENCA  
FACULTAD DE INGENIERIA  
ESCUELA DE INFORMÁTICA**

**HERRAMIENTA WEB PARA  
PLANIFICAR, GESTIONAR Y  
GOBERNAR TERRITORIOS**

**AUTORES:**

**JORGE IVÁN ESPINOSA VÁSQUEZ  
JORGE LUIS LEÓN MOROCHO**

Trabajo de graduación previo a la obtención de título de  
Ingeniero de Sistemas

**DIRECTOR:**

**ING. ÁNGEL ESPINOZA**

Cuenca – Ecuador

2011

A Dios por saberme guiar por el camino de la sabiduría. A mis padre Jorge León y Mariana Morocho por ser las personas que me han sabido enseñar lo importante de la vida y por el sacrificio que han realizado con el afán de verme a mí como una persona profesional.

A mis hermanos Marco León, Daniel León y Gabriela León por ser las personas que me han sabido entender y apoyar siempre cuando lo necesitaba.

A mis abuelitos Miguel Morocho, Rosa Mancilla, Celia León y Tíos quienes me han inculcado valores muy importantes sobre el amor y la humildad.

A mis amigos que he ido conociendo con el transcurso del tiempo y que han llegado a ser una parte importante en mi vida.

Y a mi sobrino Justin León que se ha sumado como un miembro más de mi familia y que llegó a mi vida para hacerme muy feliz y saber lo significativo que es la familia.

*Jorge León*

A Dios por sus bendiciones, por las personas que ha puesto en mi vida y que han guiado mi camino con tanto amor y sabiduría como son mis padres Jorge Iván y Aurita y mi hermana Xime, dedicado principalmente a ellos por su preocupación y dedicación sin medida, son lo mejor en mi vida.

A mi abuelito Rosalino por sus oraciones y porque con la gran fe y alegría con que encara la vida me motiva a seguir adelante en momentos difíciles.

A mi grupo de amigos del colegio, por su interés en seguir cultivando la gran amistad que nos une y que supo vencer fronteras.

A mis amigos de la universidad con quienes compartí muchas experiencias, juntos logramos superar cada reto que nos presentó la vida universitaria.

*Jorge Espinosa*

## **AGRADECIMIENTOS**

A nuestros padres y familiares por el apoyo incansable, desinteresado e incondicional, que nos han brindado durante estos años.

A nuestro director Ing. Ángel Espinoza por la guía que nos brindó durante el desarrollo de nuestra tesis, por ser una persona admirable y que nos ha inspirado a seguir preparándonos académicamente.

Al Arq. Pedro Zeas Sacoto quien nos brindó ayuda y estuvo siempre predispuesto a transmitir sus conocimientos y experiencia para la elaboración de este proyecto.



## INDICE GENERAL

	Pág.
DEDICATORIA .....	ii
AGRADECIMIENTOS .....	iv
<b>INDICE GENERAL</b> .....	<b>5</b>
RESUMEN.....	x
CAPITULO 1 .....	11
1. INTRODUCCIÓN.....	12
1.1 Antecedentes .....	13
1.2 Problema.....	14
1.3 Justificación.....	14
1.4 Objetivos .....	15
1.4.1 Objetivo General.....	15
1.4.2 Objetivos Específicos .....	15
CAPITULO 2 .....	16
2 MARCO TEÓRICO .....	17
2.1 Aplicaciones Web .....	17
2.1.1 ¿A qué se debe la popularidad de las aplicaciones web?.....	17
2.1.2 Antecedentes.....	18
2.1.3 Interfaz.....	20
2.1.4 Ventajas de las Aplicaciones Web.....	20
2.1.5 Inconvenientes en las Aplicaciones Web.....	22
2.1.6 Diferencia entre aplicación web y aplicación de internet enriquecida ...	22
2.2 Aplicaciones de Internet Enriquecidas RIA .....	23
2.3 JavaScript Asíncrono y XML (AJAX).....	24
2.3.1 ¿Qué no es AJAX?.....	28
1.5 Persistencia de Datos.....	29
2.4.1 Persistencia Básica en Java .....	30
2.4.2 ¿Qué usar?.....	34

2.5	Llamada a procedimientos remotos (RPC) .....	35
2.5.1	Pase de parámetros.....	38
2.5.2	Enlace.....	38
2.5.3	Protocolo de transporte .....	38
2.5.4	Semántica de los RPC.....	38
2.5.5	Representación de los Datos .....	39
2.5.6	Desempeño y seguridad .....	39
2.6	Estudio de las Herramientas .....	40
2.6.1	Java 40	
2.6.2	Jasper Reports .....	43
2.6.3	Google Web Toolkit (GWT).....	44
2.6.4	Ext GWT .....	66
2.6.5	Smart GWT.....	72
2.6.6	Servidor de Aplicaciones .....	81
2.6.7	MySql.....	90
CAPITULO 3 .....		95
2	Desarrollo .....	96
3.1	Modelamiento del negocio .....	96
3.1.1	Actores y trabajadores .....	98
3.1.2	Caso de uso del negocio.....	102
3.1.3	Visión, ambiente y terminología de Ge@.....	108
3.2	Requerimientos.....	139
3.2.1	Especificación de requisitos de software .....	139
3.2.2	Casos de uso del sistema.....	163
3.2.3	Resumen de los casos de uso .....	200
3.2.3	Diagramas de actividades .....	202
3.3	Análisis y diseño .....	225
3.3.1	Modelo de diseño.....	225
3.3.2	Modelo de datos: Diagrama de Entidad-Relación y Diccionario .....	226
3.3.3	Arquitectura .....	240
3.4	Implementación.....	244
3.4.1	Prototipo de interfaces .....	246
3.4.2	Diagrama de Paquetes.....	250
3.4.3	Diagrama de Componentes .....	251

3.5 Pruebas .....	254
3.5.1 CU9.- Desactivar Perfil de Usuario.....	254
3.5.2 CU16.- Ingresar Nuevo Proyecto .....	256
3.5.3 CU18.- Compartir Proyectos.....	258
3.5.4 CU31.- Modificar Institución.....	259
3.5.5 CU36.- Modificar Mensaje Publicitario.....	261
3.5.6 CU48.- Manipular Objetos .....	262
3.5.7 CU49.- Indicar Causas y Razones.....	264
3.5.8 CU50.- Mantenimiento de Realimentación.....	266
3.5.9 CU52.- Generar Sumas .....	268
3.5.10 CU51.- Generar Reporte .....	270
3.6 Despliegue.....	273
3.6.1 Manual de Usuario.....	273
 CAPITULO 4 .....	 274
4.1 Conclusiones.....	275
4.2 Recomendaciones .....	276
 BIBLIOGRAFIA.....	 277
 A N E X O S.....	 279
Anexo A: MANUAL DE USUARIO.....	280
 LOGGING.....	 281
Botón “¿Olvido su contraseña?” .....	281
Botón “ <i>Registro de nuevo usuario</i> ” .....	282
 ADMINISTRADOR .....	 284
Panel Navegador.....	285
Área de Trabajo .....	286
Barra de Título y Menú de Perfil de Usuario.....	286
Edición del Perfil de Usuario .....	287
Mensaje de Publicidad.....	288
Módulos del Sistema Ge@ .....	288
Administración de Proyectos .....	288

Administración de Instituciones.....	296
Administración de Usuarios .....	298
Administración del Sistema .....	299
INVITADO .....	300
Componentes de la Pantalla Inicial del Usuario Invitado.....	300
Botones de Configuración.....	301
Listado de Proyectos .....	302
Barra de Paginación .....	302
Operaciones del Usuario Invitado.....	303
Creación de nuevo proyecto.....	303
Eliminación de un Proyecto .....	305
Configurar Proyecto .....	305
Sistema Ge@ .....	307
Escritorio Ge@ .....	307
Menú Principal.....	307
Estructura del Software.....	309
Barra de Herramientas.....	309
Propiedades .....	312
Estructura del Sistema Ge@ .....	318
Manejo de ítems y archivos digitales .....	319
Componentes del Sistema Ge@ .....	320
1. Principios Esenciales .....	321
Opciones de cada Principio.....	321
2. Principios Sistémicos.....	326
Opciones de cada Principio.....	326
3. Interacciones .....	331
Opciones de cada Ejercicio de Interacción .....	332
4. Modelización Digital .....	333
5. Base de Conocimiento Capitalizada .....	334
6. Momentos de Intervención .....	335
7. Prototipos .....	335
7.1 Historia Sistémica .....	337
7.2 Variables e Indicadores .....	337
7.3 Diagrama Sistémico de las Variables.....	338

7.4 Relación con los elementos del Modelo.....	339
7.5 Procesos de Solucion.....	340
7.6 Sumas 340	
7.7 Interacciones Externas.....	340
7.8 Eliminar.....	341
8. Caja de Herramientas.....	344
9. Variables e Indicadores.....	346

## RESUMEN

El presente proyecto previo la obtención del título de ingenieros en sistemas consistió en desarrollar un sistema web que sirva de apoyo a la Fundación HABITierra, dedicada al trabajo con comunidades y organizaciones con el fin de dar soluciones a temas de desarrollo territorial.

*La “Herramienta web para planificar, gestionar y gobernar territorios”* es una etapa importante en el proyecto Ge@ de la fundación, que lleva cerca de once años desarrollando conceptos y metodologías en cuanto a ordenamientos y gobernabilidad territorial.

En cuanto al Sistema como tal, fue desarrollado en una arquitectura cliente-servidor de tres capas, haciendo uso del lenguaje multiplataforma Java en el cual se generó una aplicación desplegada en el servidor de aplicaciones Glassfish que fue instalado en un servidor Linux, lo que permite que la herramienta Web pueda ser accedida desde cualquier lugar con conexión a internet y a través de un navegador.

Palabras Claves: gestión, planificación, gobernabilidad, sistemas, territorio.

# **CAPITULO 1**

## 1. INTRODUCCIÓN

Históricamente las sociedades han manejado sus territorios, en función de las leyes y procesos de la naturaleza; y, de las necesidades, intereses y aspiraciones humanas. En esta interacción, la ciencia, tecnología, arte y religión han sido herramientas fundamentales para producir, usar, transformar y administrar los espacios de desarrollo.

El destino del Universo, de la Tierra y de la vida parece ser la complejidad. Por ejemplo si se realiza una comparación entre las sociedades cazadoras-recolectoras con las urbanas de ahora se ve que el nivel de complejidad es inmenso. O si se realiza el mismo análisis con la Tierra hace 3.000 millones de años con la de ahora, los niveles biológicos de complejidad se han multiplicado millones de veces. Por deducción, nuestros sistemas territoriales de la actualidad son inmensamente complejos y requieren de teorías, métodos y herramientas que faciliten la planificación, gestión y gobernabilidad.

Actualmente, cada ser humano ha llegado a familiarizarse con el término Internet o la Web, prácticamente la mayor parte de las actividades que cada uno realiza a diario se las puede realizar por medio de la Internet por ejemplo, una transferencia bancaria, compras, entretenimiento multimedia, consultas de estados de cuenta, etc., ahora es posible realizar cualquier tarea y con mucha facilidad. Basándonos en esta experiencia, porque no permitir al usuario también llevar la planificación, gestión y gobernabilidad de sus territorios desde la comodidad del hogar o trabajo, con ayuda de las Tecnologías de la Información con las que se cuenta actualmente.

El sistema o herramienta para el manejo territorial al que se le denominará Ge@<sup>1</sup>, se ocupa precisamente de lo último; es decir de ofertar teorías, métodos y herramientas (nuevas y alternativas) para garantizar un manejo territorial científico, democrático y sostenible, que inserte a los sistemas territoriales en una transición hacia las sostenibilidad integral a través de la web.

Tres aspectos básicos han revolucionado el trabajo con los territorios:

---

<sup>1</sup> Herramienta web utilizada para la planificación, gestión y gobernabilidad territorial.



- 1) El desarrollo de la nueva ciencia, especialmente con las teorías: cuántica, relatividad, caos, emergencia, complejidad, fractales y sistémica, ha posibilitado entender mejor y comprender más sobre la estructura y funcionamiento de los territorios.
- 2) La creación del sistema de información geográfico permite comprender y manejar con facilidad la complejidad geográfica de los sistemas territoriales y al mismo tiempo facilita la generación de modelos territoriales.
- 3) En la actualidad se desarrollan muchas experiencias relativas al territorio, especialmente ordenación territorial, manejo de riesgos, planificación urbana, transporte y otros.

Ge@, se inserta en esta realidad con alternativas reales de planificación gestión y gobernabilidad territorial. Para lo cual toma lo mejor de los tres aspectos descritos anteriormente y añade dos cosas fundamentales:

- 1) La participación activa y creativa de la comunidad y trabajo en equipo a través de la web.
- 2) La capitalización del conocimiento humano.

### **1.1 Antecedentes**

Ge@ es un producto de más de 12 años de I+D+i<sup>2</sup>, sobre temas de desarrollo territorial sostenible y gobernabilidad local. En esta experiencia están trabajos con comunidades, organizaciones sociales, profesionales e instituciones. Esta experiencia fue capitalizada con la creación de Ge@ -Sistema de Manejo Territorial- (febrero de 2002) y en la actualidad es gestionado por HABITierra.org, una organización privada con fines sociales y científicos.

Desde el año 2002 Ge@ ha experimentado una constante construcción-aplicación-corrección-evolución dentro del ambiente I+D+i, esto garantiza la solvencia y sustento teórico-práctico. Se ha aplicado Ge@ en algunos trabajos puntuales como diagnósticos de sistemas territoriales y organizacionales, procesos de capacitación y talleres de diálogo. También se ha aplicado en proyectosde *governabilidad territorial, planificación, diseño y*

---

<sup>2</sup>Investigación, Desarrollo Tecnológico e Innovación.

*construcción de asentamientos humanos sostenibles, ordenación territorial, capitalización del conocimiento y en parte en riesgos territoriales.*

Entre 1992 y 2002, se desarrollaron una serie de actividades y procesos creando y experimentando teorías, métodos y herramientas en acciones concretas y proyectos de desarrollo. A partir de 2002, año en el que se configuró la primera estructura de Ge@, se ejecutaron actividades y procesos más organizados en función del desarrollo de Ge@. También se aplicó Ge@ en el diseño y ejecución en algunos proyectos de desarrollo local.

### **1.2 Problema**

Debido a la aplicabilidad y aceptación que ha tenido sistema Ge@ su evolución y crecimiento ha sido inevitable; y, con ello han surgido inconvenientes. Experiencias previas de los usuarios que han utilizado Ge@ en sus proyectos indican que cuando hay un número significativo de personas involucradas, la digitalización de la información se centra en una sola persona pudiendo esta perder información o tergiversarla, a la vez que los otros usuarios no tienen una participación activa en los procesos, y en caso de querer involucrarlos representaría un retraso en el progreso debido al que el sistema no es portable ni multiusuario. Además de ello no cuenta con una facilidad de mantenimiento. El sistema Ge@ actual no consta con una administración de usuarios, proyectos e instituciones.

### **1.3 Justificación**

Actualmente Ge@ es un sistema de escritorio monousuario cuya necesidad a corto plazo es mejorar su modelación digital, para que sea más amigable y fácil de operar y esté orientado a un ambiente web. Es por ello que la “Herramienta web para planificar, gestionar y gobernar territorios” al trabajar sobre este entorno incorporará la participación activa y creativa de la comunidad y trabajo en equipo, para poder así alcanzar la capitalización del conocimiento humano que es un pilar fundamental para que el sistema sea exitoso. Además de ser multiusuario permite a los administradores recolectar información, interpretarla y brindar asesoría que representaría un beneficio mutuo.

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

Desarrollar una herramienta orientada a la web para planificar, gestionar y gobernar territorios que permita promover, apoyar y participar en el cambio de los sistemas territoriales hacia una transición de sostenibilidad integral.

### **1.4.2 Objetivos Específicos**

- a) Lograr el entendimiento y adopción de la modelización teórica de Ge@.
- b) Construir un modelo digital de Ge@ orientado a la web que sea amigable y que sea de fácil uso.
- c) Implementar un módulo de administración para proyectos, usuarios e instituciones.
- d) Permitir la gestión y control del sistema Ge@.
- e) La herramienta web debe conseguir la capitalización del conocimiento humano.
- f) Apoyar a la planificación, manejo y ordenamiento territorial.
- g) Presentación de reportes que permitan a los administradores la asesoría para las instituciones o clientes.

## **CAPITULO 2**

## **2 MARCO TEÓRICO**

### **2.1 Aplicaciones Web**

En la ingeniería de software se le denomina **aplicaciones web** a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de internet o una intranet mediante un navegador. En otras palabras, significa que son aplicaciones que se codifican en un lenguaje soportado por los navegadores web en los cuales se confía la ejecución al navegador.

#### **2.1.1 ¿A qué se debe la popularidad de las aplicaciones web?**

La popularidad de las aplicaciones web se debe a lo práctico de los navegadores web existentes hoy en día, ya que se basa en un cliente ligero, es decir, es independiente del sistema operativo (multiplataforma), así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales, además que la mayoría de los sistemas operativos incluyen un explorador web y a la facilidad para poder descargar de la web otros navegadores. Existen infinidad de aplicaciones web como los web mails, wikis, weblogs, tiendas en línea, que son claros ejemplos de aplicaciones web.

Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

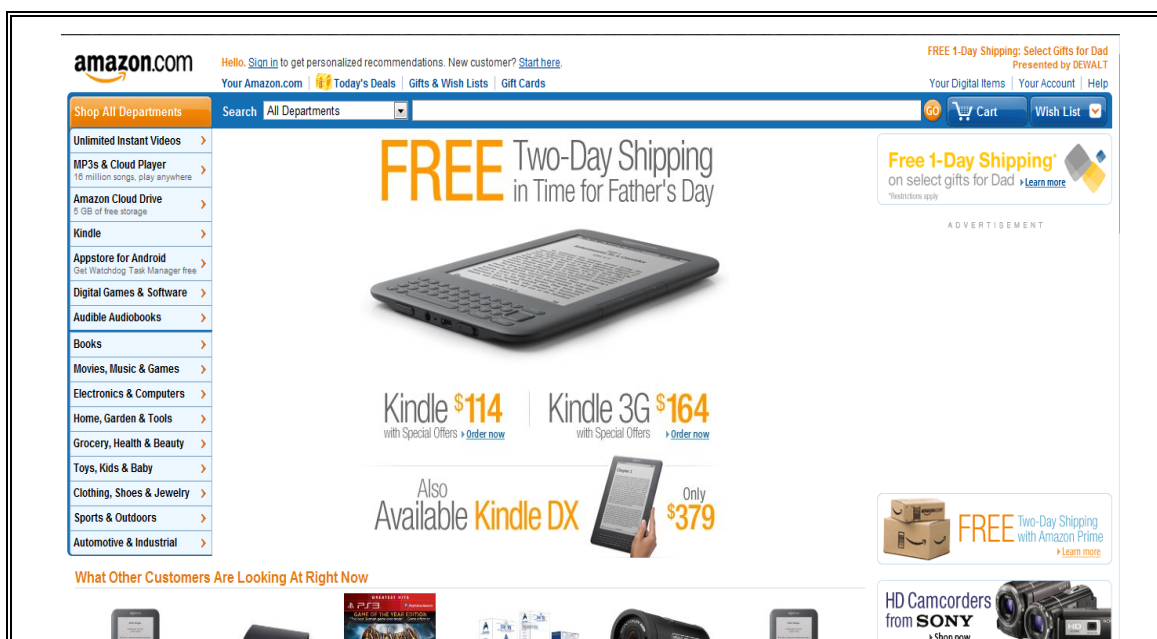


Figura 2-1: Amazon.com, ejemplo típico de aplicación web (tienda en línea)

### 2.1.2 Antecedentes

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente que servía de interfaz y que tenía que ser instalado en cada máquina del usuario y de esta manera poder realizar peticiones al servidor que le daba la respuesta, esto implicaba un problema, ya que una mejora en el servidor involucraba un cambio en las aplicaciones instaladas en cada ordenador personal, añadiendo un coste de soporte técnico y disminuyendo la productividad.

Luego con el surgimiento del internet, en inicio la web era sencillamente una colección de páginas estáticas, documentos, etc., para su consulta o descarga. El paso inmediatamente posterior en su evolución fue la inclusión de un método para elaborar páginas dinámicas que permitieran que lo mostrado tuviese un carácter dinámico (es decir, presentar información a partir de las peticiones del usuario). Este método fue conocido como CGI (“Common Gateway Interface”) y definía un mecanismo en el cual se podía pasar información entre el servidor y ciertos programas externos.

El funcionamiento de los CGIs tenía un punto débil: cada vez que se recibía una petición, el servidor debía lanzar un proceso para ejecutar el programa CGI. Como la mayoría de CGIs estaban escritos en lenguajes interpretados, como Perl o Python, o en lenguajes que requerían "Runtime Environment", como Java o Visual Basic, el servidor se veía sometido a una gran carga. La concurrencia de múltiples accesos al CGI podía comportar problemas graves.

Por eso se empiezan a desarrollar alternativas a los CGIs que solucionaran el problema del rendimiento. Las soluciones llegan básicamente por 2 vías: 1) se diseñan sistemas de ejecución de módulos mejor integrados con el servidor, que evitan la instanciación y ejecución de varios programas, y 2) se dota a los servidores un intérprete de algún lenguaje de programación que permita incluir el código en las páginas de forma que lo ejecute el servidor, reduciendo el intervalo de respuesta.

Entonces se experimenta un aumento del número de arquitecturas y lenguajes que permiten desarrollar aplicaciones web. Todas siguen alguna de estas vías. Las más útiles y las más utilizadas son las que permiten mezclar los 2 sistemas: un lenguaje integrado que permita al servidor interpretar comandos "incrustados" en las páginas HTML y, además, un sistema de ejecución de programas mejor enlazado con el servidor, que no implique los problemas de rendimiento propios de los CGIs.

Las aplicaciones web actuales generan dinámicamente una serie de páginas en un formato estándar, como HTML o XHTML, soportados por los navegadores web comunes. Se utilizan lenguajes interpretados en el lado del cliente, directamente o a través de plugins tales como JavaScript, Java, Flash, etc., para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web en particular se envía al cliente como un documento estático, pero la secuencia de páginas ofrece al usuario una experiencia interactiva. Durante la sesión, el navegador web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación web.

Existen tecnologías muy potentes y utilizadas por la mayoría de los programadores para el desarrollo de aplicaciones web, una es la tecnología desarrollada por Sun

Microsystems, integrado por 2 componentes; un lenguaje que permite la incrustación de código en las páginas HTML que el servidor convierte en programas ejecutables, JSP ("Java Server Pages" o "Páginas de Servidor de Java"), y un método de programación muy ligado al servidor, con un rendimiento superior a los CGIs, denominado "Java Servlet".

Otra tecnología de éxito y una de las más utilizadas es el lenguaje PHP. Se trata de un lenguaje interpretado que permite la incrustación de HTML en los programas, con una sintaxis derivada de C y Perl. El hecho de ser sencillo y potente ha contribuido a hacer de PHP una herramienta muy apropiada para determinados desarrollos.

Y por último, y no menos importante es la desarrollada por la empresa Microsoft que es la tecnología ASP el cual intenta ser una solución para un modelo de programación rápida como si se estuviera programando en Visual Basic y C#, por supuesto con muchas limitaciones y algunas ventajas específicas para entornos web.

### **2.1.3 Interfaz**

La interfaz gráfica de una aplicación web puede ser sumamente completa y funcional, gracias a las variadas tecnologías web que existen: Java, JavaScript, DHTML, Flash, Silverlight, Ajax, entre otras.

Prácticamente no hay limitaciones, las aplicaciones web pueden hacer casi todo lo que está disponible para aplicaciones tradicionales: acceder al mouse, al teclado, ejecutar audio o video, mostrar animaciones, soporte para arrastrar y soltar, y otros tipos de tecnologías de interacción usuario-aplicación.

Ajax es un ejemplo de una tecnología de desarrollo web que le da gran poder de interactividad a las aplicaciones web y del cual se hablara a fondo en la sección 2.3.

### **2.1.4 Ventajas de las Aplicaciones Web**

Entre las ventajas que brindan las aplicaciones web tenemos las siguientes:

- Ahorra tiempo: Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.



- No hay problemas de compatibilidad: Basta tener un navegador actualizado para poder utilizarlas.
- No ocupan espacio en nuestro disco duro.
- Actualizaciones inmediatas: Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- Consumo de recursos bajo: Dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador.
- Multiplataforma: Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Portables: Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil...) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet). La reciente tendencia al acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.
- La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- Los virus no dañan los datos porque éstos están guardados en el servidor de la aplicación.
- Colaboración: Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.
- Los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones web ricas (RIAs).

### **2.1.5 Inconvenientes en las Aplicaciones Web**

Habitualmente ofrecen menos funcionalidades que las aplicaciones de escritorio. Se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo. Pero cada vez los navegadores están más preparados para mejorar en este aspecto. La aparición de HTML 5 representa un hito en este sentido. Es posible añadir funcionalidades a estas aplicaciones gracias al uso de Aplicaciones de Internet Ricas.

La disponibilidad depende de un tercero, el proveedor de la conexión a internet o el que provee el enlace entre el servidor de la aplicación y el cliente. Así que la disponibilidad del servicio está sujeta al proveedor.

### **2.1.6 Diferencia entre aplicación web y aplicación de internet enriquecida**

Las aplicaciones web se ejecutan nativamente desde el navegador. Pero existen algunas aplicaciones que funcionan desde el navegador pero además requieren la instalación de un software en el ordenador para poder utilizarse. Estas aplicaciones se denominan Aplicaciones de Internet Ricas. El motivo de usar este software adicional es que hay muchas funcionalidades que los navegadores no pueden ofrecer, y él enriquece a las aplicaciones web ofreciendo dichas funcionalidades. .

Ejemplos de funcionalidades que pueden ofrecer los programas online gracias al uso de software instalado:

- Procesamiento de imágenes
- Captura de imágenes
- Uso de webcam / Captura de video, etc.

## 2.2 Aplicaciones de Internet Enriquecidas RIA

RIA, acrónimo de Rich Internet Application (Aplicaciones de Internet Enriquecidas). Son aplicaciones web que tienen la mayoría de las características de las aplicaciones tradicionales, estas aplicaciones utilizan un “navegador web” estandarizado para ejecutarse y por medio de un “plugin” o independientemente una “virtual machine”, se agregan las características adicionales.

Esta surge como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales. Buscan mejorar la experiencia del usuario. Normalmente en las aplicaciones Web, hay una recarga continua de páginas cada vez que el usuario pulsa sobre un enlace. De esta forma se produce un tráfico muy alto entre el cliente y el servidor, llegando muchas veces, a recargar la misma página con un mínimo cambio.

En los entornos RIA, en cambio, no se producen recargas de página, ya que desde el principio se carga toda la aplicación, y sólo se produce comunicación con el servidor cuando se necesitan datos externos como acceso a una Base de Datos u otros ficheros externos.

Actualmente, existen muchas tecnologías, propietarias y no propietarias, que son capaces de introducir en la web el paradigma de la aplicación de escritorio. Esto es, de crear aplicaciones web con dos características fundamentales:

1. Una alta interactividad con el usuario, a través multitud de elementos de interacción que antes sólo eran viables en entornos de escritorio (como menús, árboles, deslizadores, etc.), programables bajo cualquier evento de usuario (como clic de ratón, pulsación de tecla, *drag and drop*, etc.) y con un aspecto visual totalmente personalizable.
2. Una alta velocidad de respuesta a la interacción del usuario. La unidad de información mínima es la que desee el desarrollador: una etiqueta, una tabla o quizás toda la página. De este modo, es posible conseguir una respuesta inmediata a las acciones del usuario, descargando sólo los datos absolutamente necesarios.

No cabe duda que este tipo de aplicaciones, que normalmente reciben el nombre de Aplicaciones Ricas de Internet o *Rich Internet Applications* (RIA), pueden mejorar sustancialmente la experiencia de usuario en la web. Sin embargo, existen como mínimo dos aspectos que conviene tener en cuenta:

1. La usabilidad. El modelo de interacción en las páginas convencionales es extremadamente simple pero también extremadamente claro. En las RIA, los desarrolladores pueden crear, utilizar y personalizar nuevos y sofisticados elementos de interacción que permiten crear interfaces absolutamente complejas que, en algunos casos, más que ayudar al usuario, pueden provocar confusión y dudas sobre su uso.
2. Accesibilidad Web. En muchas ocasiones, los avances de la tecnología representan una amenaza para la accesibilidad web. Actualmente, en lo que se refiere a las páginas convencionales, la accesibilidad web ha tenido un recorrido de más de una década y se puede decir que no existe ninguna razón técnica que justifique la no accesibilidad. Un recorrido que, desgraciadamente, no es suficiente para dotar de accesibilidad a todos los cambios que han venido de la mano de las RIA.

A pesar de que ambos aspectos son muy interesantes, este artículo se va a centrar en el segundo apartado: la accesibilidad web de las RIA.

### **2.3 JavaScript Asíncrono y XML (AJAX)**

El término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo.

En realidad, el término AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML".

El artículo define AJAX de la siguiente forma:

“Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.”

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

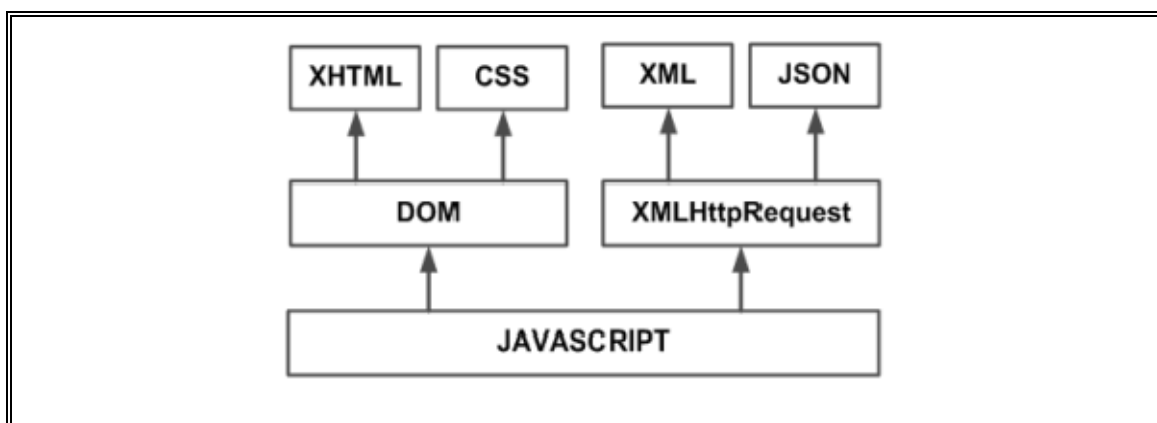


Figura 2-2: Tecnologías agrupadas bajo el término de AJAX

Desarrollar aplicaciones AJAX requiere un conocimiento avanzado de todas y cada una de las tecnologías anteriores.

En las aplicaciones web tradicionales, las acciones del usuario en la página (pinchar en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario.

En el siguiente esquema, la imagen de la izquierda muestra el modelo tradicional de las aplicaciones web. La imagen de la derecha muestra el nuevo modelo propuesto por AJAX:

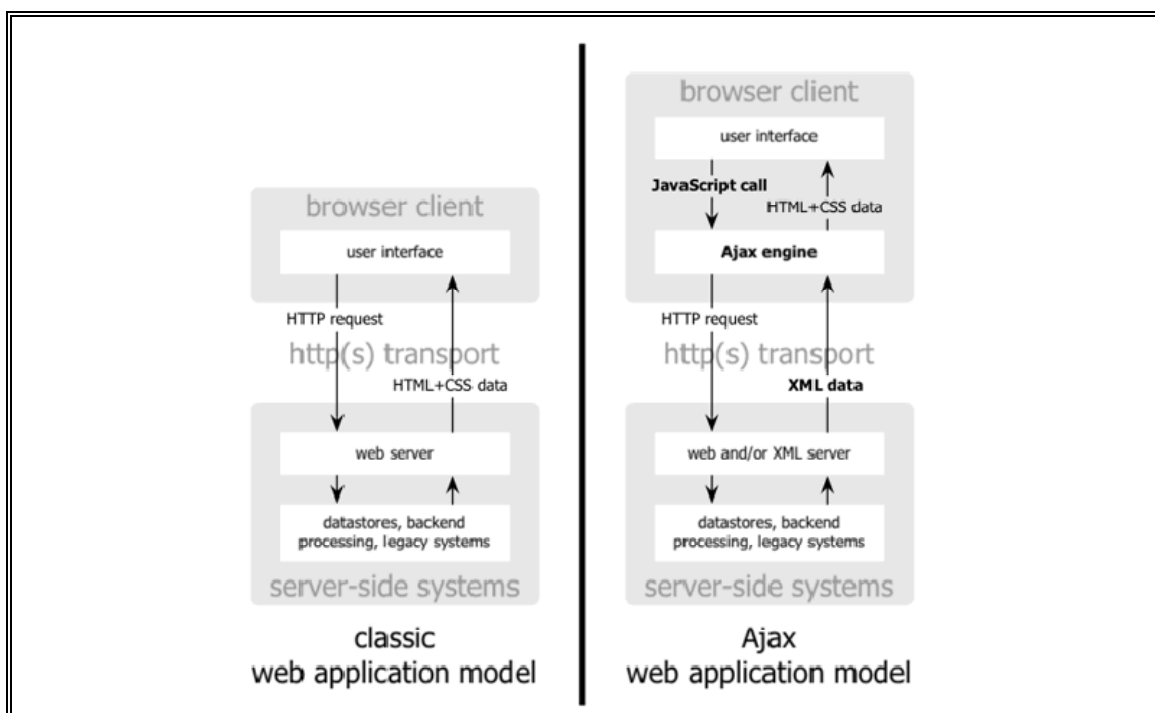


Figura 2-3. Comparación gráfica del modelo tradicional de aplicación web y del nuevo modelo propuesto por AJAX. (Imagen original creada por Adaptive Path)

Esta técnica tradicional para crear aplicaciones web funciona correctamente, pero no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, su uso se convierte en algo molesto.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

El siguiente esquema muestra la diferencia más importante entre una aplicación web tradicional y una aplicación web creada con AJAX. La imagen superior muestra la interacción síncrona propia de las aplicaciones web tradicionales. La imagen inferior muestra la comunicación asíncrona de las aplicaciones creadas con AJAX.

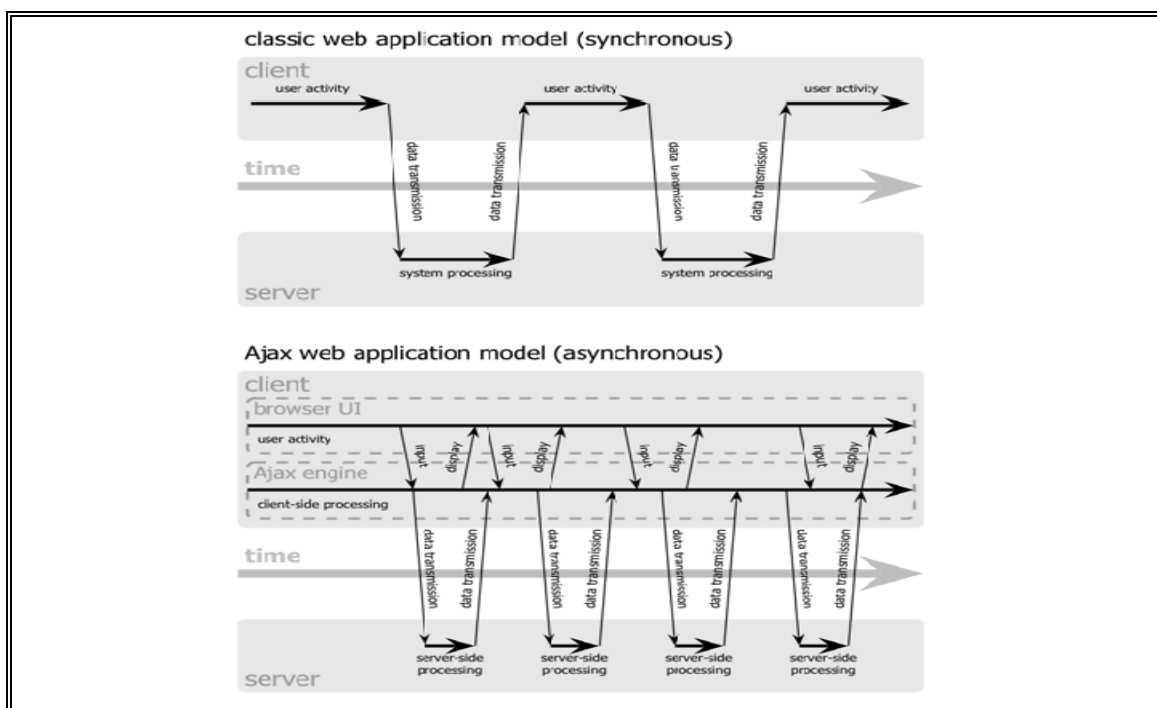


Figura 2-4. Comparación entre las comunicaciones síncronas de las aplicaciones web tradicionales y las comunicaciones asíncronas de las aplicaciones AJAX (Imagen original creada por Adaptive Path)

Las peticiones HTTP al servidor se sustituyen por peticiones JavaScript que se realizan al elemento encargado de AJAX. Las peticiones más simples no requieren intervención del servidor, por lo que la respuesta es inmediata. Si la interacción requiere una respuesta del servidor, la petición se realiza de forma asíncrona mediante AJAX. En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor.

Desde su aparición, se han creado cientos de aplicaciones web basadas en AJAX. En la mayoría de casos, AJAX puede sustituir completamente a otras técnicas como Flash.

Además, en el caso de las aplicaciones web más avanzadas, pueden llegar a sustituir a las aplicaciones de escritorio.

A continuación se muestra una lista de algunas de las aplicaciones más conocidas basadas en AJAX:

- Gestores de correo electrónico: Gmail (<http://www.gmail.com>) , Yahoo Mail (<http://mail.yahoo.com>) , Windows Live Mail (<http://www.hotmail.com>) .
- Cartografía: Google Maps (<http://maps.google.com>) , Yahoo Maps (<http://maps.yahoo.com>), Windows Live Local (<http://maps.live.com>) .
- Aplicaciones web y productividad: Google Docs (<http://docs.google.com>) , Zimbra (<http://www.zimbra.com/>), Zoho (<http://www.zoho.com/>) .
- Otras: Netvibes (<http://www.netvibes.com>) [metapágina], Digg (<http://www.digg.com>) [noticias], Meebo (<http://www.meebo.com>) [mensajería], 30 Boxes (<http://www.30boxes.com>) [calendario], Flickr (<http://www.flickr.com>) [fotografía].

La “*Herramienta web para la planificación, gestión y gobernabilidad territorial*” fue desarrollada en su totalidad bajo esta modalidad, los usuarios al momento de utilizarlo van a notar como la herramienta web se carga una sola vez, y los datos son presentados conforme se trabaja con la aplicación, es por ello de la buena velocidad de ejecución y la carencia de refrescos de página, una buena manera de trabajo para hacer del sistema Ge@ un agradable uso.

### **2.3.1 ¿Qué no es AJAX?**

Ajax no es un script, no es un servidor, no es un archivo, no es un programa, no es un módulo de apache, no es una extensión de PHP, NO ES UNA TECNOLOGIA.

Por ejemplo en un proceso normal una aplicación web recarga o va de una página a otra porque debe enviar una petición a algún servidor web esta debe procesarla y mandar el resultado final y el resultado final es la página a la cual nos dirige la aplicación o la recarga de la página misma. En todo esto AJAX hace lo mismo con la diferencia que lo hace en background o sea que realiza todo el proceso de comunicación con algún servidor web sin



que el usuario pueda verlo y una vez terminada con éxito solo añadir el resultado del proceso a la página en curso de esta manera evitar la recarga o una página receptora de los resultados del proceso.

Entiéndase que AJAX no se refiere a usar el objeto XMLHttpRequest de manera indispensable porque en realidad AJAX no es una tecnología, sino la unión de varias tecnologías que juntas puede lograr cosas realmente impresionantes.

### 1.5 Persistencia de Datos

Podemos encontrar diferentes definiciones del término persistencia, según distintos puntos de vista y autores. Veamos dos que con más claridad y sencillez, concretan el concepto de persistencia de objetos.

La primera, más antigua, dice así: *«Es la capacidad del programador para conseguir que sus datos sobrevivan a la ejecución del proceso que los creó, de forma que puedan ser reutilizados en otro proceso. Cada objeto, independiente de su tipo, debería poder llegar a ser persistente sin traducción explícita. También, debería ser implícito que el usuario no tuviera que mover o copiar los datos expresamente para ser persistentes»*.<sup>3</sup>

Esta definición nos recuerda que es tarea del programador, determinar cuándo y cómo una instancia pasa a ser persistente o deja de serlo, o cuando, debe ser nuevamente reconstruida; asimismo, que la transformación de un objeto en su imagen persistente y viceversa, debe ser transparente para el programador, sin su intervención; y que todos los tipos, clases, deberían tener la posibilidad de que sus instancias perduren.

La otra definición dice así: *Persistencia es «la capacidad de un lenguaje de programación o entorno de desarrollo de programación para almacenar y recuperar el estado de los objetos de forma que sobrevivan a los procesos que los manipulan»*<sup>4</sup>

---

<sup>3</sup> M. P. ATKINSON, F. BANCILIONI, D. J. DEWITT, K. R. DITRICH, D. MAIER, AND S. B. ZDONIK. The object-oriented database system manifesto. SIGMOD Conference, May 1990

<sup>4</sup> KENT BECK, Extreme Programming Explained, Embrace Change, Addison-Wesley Octubre, 1999

En definitiva, el programador debería disponer de algún medio para poder convertir el estado de un objeto, a una representación adecuada sobre un soporte de información, que permitirá con posterioridad revivir o reconstruir el objeto, logrando que como programadores, no debamos preocuparnos de cómo ésta operación es llevada a cabo

### **2.4.1 Persistencia Básica en Java**

Al abordar el desarrollo de una aplicación en Java, uno de los primeros requerimientos a resolver es la integración con una base de datos para guardar, actualizar y recuperar la información que utiliza nuestra aplicación.

Se llama “persistencia” de los objetos a su capacidad para guardarse y recuperarse desde un medio de almacenamiento. La persistencia en Base de Datos relacionales se suele implementar mediante el desarrollo de funcionalidad específica utilizando la tecnología JDBC o mediante frameworks que automatizan el proceso a partir de mapeos (conocidos como Object Relational Mapping, ORM) como es el caso de Hibernate.

#### **2.4.1.1 JDBC**

Java DataBase Connectivity es el API de Java que define como una aplicación cliente accederá a una base de datos, independientemente del motor de base de datos al que accedamos.

#### **Características**

- Interfaz de programación para acceso a BD
- Rutinas para manipular los datos y metadatos
- Acceso directo a la base de datos
- Control de las conexiones y transacciones (autoCommit)
- Lenguaje SQL

#### **Problemas**

- Código específico de Base de Datos
  - MySQL: outer join, autoincrement

- Oracle (+), Sequences
- Acceso directo a conexiones y transacciones
- Transacciones a un único DataSource
- El desarrollador Java debe saber también otro lenguaje (SQL)
- El código se hace repetitivo
  - Inserts, Selects, Updates, Deletes para cada objeto del modelo

A continuación mostramos un ejemplo del código necesario para insertar un elemento y hacer una consulta:

```
// Cargar el driver
Class.forName(jdbcDriver);

// Crear una conexión
Connection con = DriverManager.getConnection(jdbcUrl, userName,
password);

// Crear un Statement para obtener el máximo id
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT MAX(id) FROM " + schema
+ "DEMO_QUERIES");
int id = 0;
if (rs.next()) {
id = rs.getInt(1) + 1;
}
rs.close();
stmt.close();

// Crear un PreparedStatement para insert
PreparedStatement prepareInsert = con.prepareStatement(
"INSERT INTO " + schema + "DEMO_QUERIES (id, nombre, fecha)"
+ "VALUES (?, ?, ?)");

// Completar sus atributos
int i = 1;
prepareInsert.setInt(i++, id);
prepareInsert.setString(i++, "Ejemplo JDBC");
prepareInsert.setDate(i++, new Date((new Date()).getTime()));

// Ejecutar el insert
prepareInsert.executeUpdate();
prepareInsert.close();

// Crear un PreparedStatement para select
PreparedStatement prepareSelect = con
.prepareStatement("SELECT id, nombre, fecha FROM " + schema
+ "DEMO_QUERIES WHERE fecha < ?");

// Completar sus atributos
```

```

prepareSelect.setDate(1, new Date((new java.util.Date()).getTime()
+ DAY_IN_MILLIS));

// Ejecutar la select
rs = prepareSelect.executeQuery();
while (rs.next()) {
System.out.println("Id = " + rs.getInt(1) + " - nombre = "
+ rs.getString(2) + " - fecha = " + rs.getDate(3) + "");
}
rs.close();
prepareSelect.close();

```

El código es esencialmente simple aunque si se tiene un número amplio de clases y problemáticas de maestro/detalle o soporte multi-base de datos la implementación puede volverse muy difícil de mantener.

#### 2.4.1.2 Hibernate

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO.

Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate ofrece un lenguaje de consulta de datos llamado **HQL** (*Hibernate Query Language*)

#### Ejemplo de Hibernate:

```

Entidad entidad = new Entidad();
entidad.setId(id);
entidad.setNombre("Ejemplo Hibernate\n");

Session session = sessionFactory.getCurrentSession();

```

```

session.beginTransaction();
session.save(entidad);
session.getTransaction().commit();

```

O para obtener el listado:

```

Session session = sessionFactory.getCurrentSession();
session.beginTransaction();
Query q = session
.createQuery("from e in class com.persistencia.hibernate.Entidad where
e.fecha < :fecha");
// se inyecta el valor del parámetro utilizando el nombre
q.setParameter("fecha", new java.sql.Date(fecha.getTime()));
List entidades = q.list();

```

En este ejemplo *Entidad* es un POJO (Plain Old Java Object), es decir una simple clase Java que tiene métodos get y set para cada uno de los atributos; es en el fichero de mapeo donde se encuentra la esencia de Hibernate, dicho fichero es similar al siguiente:

```

<hibernate-mapping>
<class name="com.hemasoft.demos.queries.hibernate.Entidad"
table="DEMO_QUERIES">
<id name="id" type="java.lang.Integer" />
<property name="nombre" />
<property name="fecha" type="java.sql.Date" />
</class>
</hibernate-mapping>

```

Este fichero de mapeo lo declaramos en hibernate.cfg.xml junto a la información de acceso a la base de datos de la siguiente forma:

```

<hibernate-configuration>

<session-factory>

<!-- Database connection settings -->
<property name="connection.driver_class">oracle.jdbc.driver.OracleDriver<
/property>
<property name="connection.url">jdbc:oracle:thin:@192.168.1.20:1521:qemu<
/property>
<property name="connection.username">test</property>
<property name="connection.password">test</property>

<!-- JDBC connection pool (use the built-in) -->
<property name="connection.pool_size">1</property>

<!-- SQL dialect -->
<property name="dialect">org.hibernate.dialect.Oracle9Dialect</property>

```

```
<mapping resource="com/hemasoft/demos/queries/hibernate/Entidad.hbm.xml" /
>

</session-factory>

</hibernate-configuration>
```

Los costes de aprendizaje de este framework en términos de tiempo dedicado se van recompensando desde el primer día con el ahorro en tiempo de desarrollo y depuración de código JDBC. Incluso, a partir de la versión 3 de Hibernate su configuración se simplifica notablemente mediante Anotaciones.

Además, Hibernate nos proporciona grandes beneficios como:

- Soporte a múltiples motores de base de datos
- Bajo acoplamiento entre negocio y persistencia, ya que su diseño está orientado a objetos así como el soporte a consultas y operaciones (HQL).
- Desarrollo robusto, ya que el framework ha madurado tras años de uso en decenas de miles de proyectos
- Optimizado, ya que el SQL generado contiene optimizaciones específicas para cada motor de base de datos mediante componentes especializados llamados dialectos.
- Rápido y completo, ya que con la funcionalidad estándar de Hibernate podremos cubrir el 80 - 90% de la persistencia de nuestra aplicación

Todo esto nos permite centrar nuestros esfuerzos en desarrollar la funcionalidad de la aplicación.

#### **2.4.2 ¿Qué usar?**

Lo aconsejable es usar tecnologías de persistencia más avanzadas cuando se trata de modelos de datos con muchas clases (muchas tablas) y complejos (herencias), salvo que el sistema de base de datos sea muy peculiar o tenga características avanzadas que sólo se aprovecharían con un uso directo y a medida de JDBC.

Actualmente disponemos de toda una artillería de tecnologías de persistencia más avanzadas, que de hecho casi siempre están construidas sobre JDBC aunque lo oculten, tal y como JDO, J2EE CMP, JCA y SQLJ por citar las estándar, e Hibernate, Castor, TopLink y CocoBase por citar productos no estándar libres y comerciales.

No siempre las tecnologías más completas son las que se deben usar, debemos tomar en consideración temas como:

- Necesidades del proyecto
- Tamaño del proyecto
- Necesidades de la organización
- Experiencia de los desarrolladores
- Madurez de la tecnología
- Documentación y Soporte

## **2.5 Llamada a procedimientos remotos (RPC)**

Una llamada a un procedimiento remoto (función o subrutina) es un método bien conocido para transferir el control de una parte del programa a otra, con un retorno de control a la primera. Asociado con la llamada a un procedimiento están el pase de parámetros y el retorno de uno o varios resultados. Cuando el código que invoca un procedimiento y dicho procedimiento está en un mismo proceso en un computador dado, se dice que ha ocurrido una llamada a un procedimiento local.

Por el contrario, en una llamada a un procedimiento remoto (RPC, Remote Procedure Call) el sistema local invoca, a través de la red, a una función alojada en otro sistema. Lo que se pretende es hacerle parecer al programador que está ocurriendo una simple llamada local.

Se utiliza el término solicitud para referirse a la llamada que realiza el cliente al procedimiento remoto (servidor) y, el término respuesta, para describir el resultado devuelto por este último.

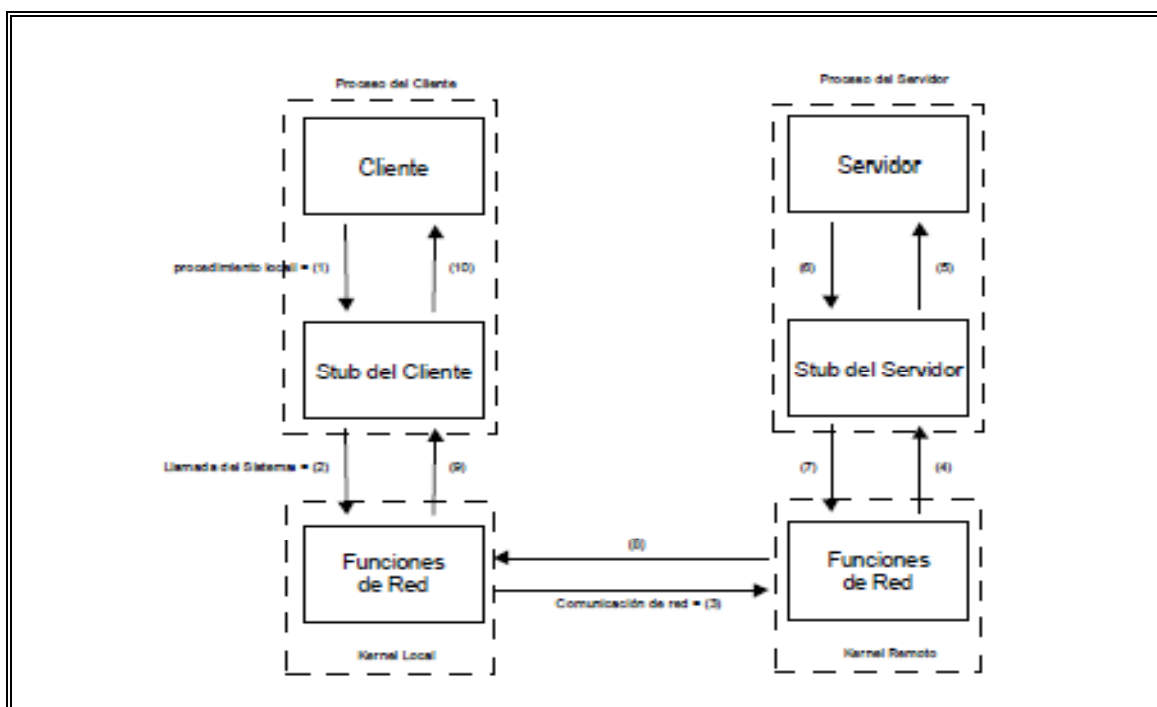


Figura 2-5. Modelo RPC

1. El cliente llama a un procedimiento local llamado “stub” del cliente, el cual aparenta ser el procedimiento servidor que el cliente desea llamar. El propósito del “stub” del cliente es empaquetar los argumentos del procedimiento remoto, adecuarlos a algún formato estándar y construir uno o varios mensajes de red. El empaquetamiento de los argumentos del procedimiento remoto en mensajes de red se conoce como “marshaling”.
2. Estos mensajes son enviados por el “stub” del cliente al sistema remoto, lo cual requiere una llamada del sistema.
3. Los mensajes son transferidos al sistema remoto empleando protocolos con o sin conexión.
4. Un procedimiento “stub” del servidor espera en el sistema remoto la solicitud del cliente. Desempaqueta los argumentos de los mensajes de red y si es necesario realiza alguna conversión.



5. El “stub” del servidor realiza la llamada al procedimiento local que realmente invoca la función del servidor y le pasa los argumentos transferidos a través de la red desde el “stub” del cliente.
6. Cuando el procedimiento del servidor termina, éste le regresa el control al “stub” del servidor devolviendo los resultados obtenidos.
7. El “stub” del servidor adecua el formato de tales resultados, si es necesario, y los empaqueta en mensajes de red para ser devueltos al “stub” del cliente.
8. Los mensajes son transmitidos al “stub” del cliente.
9. El “stub” del cliente lee los mensajes recibidos.
10. Luego de posiblemente convertir los valores de retorno, el “stub” del cliente retorna finalmente dichos resultados a la función del cliente haciendo parecer un retorno normal de función.

El concepto de llamada a procedimiento remoto permite ocultar en los “stubs” todos los detalles del código correspondiente a la comunicación a través de la red. Esto permite que los desarrolladores de programas de aplicación no se preocupen por detalles tales como “sockets” y ordenamiento de bytes. Uno de los objetivos de RPC es facilitar el desarrollo de aplicaciones.

Según el modelo OSI, RPC cae en algún lado entre la capa de transporte y aplicación. Típicamente se considera parte de la capa de presentación. Debido a que RPC le “oculta” a la capa de aplicación los detalles de la red, usualmente incluye una especificación de algún formato estándar para el intercambio de argumentos y resultados entre el cliente y el servidor.

Las implementaciones de RPC más populares son:

- La desarrollada por Sun Microsystem denominada ONC-RCP (Open Network Computing, ONC-RCP), distribuida con casi todos los sistemas UNIX.
- La desarrollada por Microsoft en línea con el Ambiente de Computación Distribuida (DCE, Distributed Computing Enviroment) definido por la Fundación de Software Abierto (OSF, Open Software Foundation). Incluida en los sistemas operativos Windows.

### **2.5.1 Pase de parámetros**

El pase de parámetros puede no ser transparente. Para el caso donde los parámetros son pasados por valor la implementación es simple. Sin embargo, se presentan problemas cuando se intenta pasar parámetros por referencia. Es por esto que típicamente las implementaciones solo le permiten al cliente pasar argumentos por valor. Para cada procedimiento remoto se definen específicamente los argumentos de entrada y los valores de retorno.

### **2.5.2 Enlace**

Esto se refiere a que el cliente contacte al sistema remoto apropiado para la ejecución de un procedimiento específico. Existen dos componentes a enlazar:

- El host remoto, y
- El proceso servidor deseado en dicho host.

Diferentes técnicas son empleadas para lograr este objetivo.

### **2.5.3 Protocolo de transporte**

Algunas implementaciones de RPC emplean un único protocolo de transporte mientras otras permiten su selección. La mayoría de las implementaciones dicen ser independientes del protocolo, sin embargo éstas sólo soportan uno o dos protocolos.

Específicamente, la implementación ONC y DCE soportan TCP y UDP como protocolos de transporte.

### **2.5.4 Semántica de los RPC**

Cuando se realiza una llamada a un procedimiento local, uno no se pregunta cuantas veces el procedimiento se ejecutó. Si un procedimiento retorna entonces se ejecutó exactamente una vez. Sin embargo, si se considera un procedimiento remoto del cual no se ha obtenido respuesta después de un intervalo de tiempo, no se puede tener certeza si se ha ejecutado. Si el servidor experimenta un error de ejecución, por ejemplo, antes de que el

“stub” del lado del servidor se realice la llamada, entonces ésta no se habrá ejecutado. Si el servidor experimenta un error de ejecución después de haber devuelto su resultado al “stub”, se habrá ejecutado una vez. Si el cliente retransmite una solicitud por no haber recibido una respuesta del servidor, se confunden aún más las cosas. Es posible que la primera solicitud haya sufrido un retraso en algún lugar de la red y finalmente, haya sido ejecutada al igual que la solicitud retransmitida.

Existe tres tipos de semánticas de RPC: exactamente una vez, cuando mucho una vez, y al menos una vez.

### **2.5.5 Representación de los Datos**

Empleando llamadas locales no existen problemas de incompatibilidad de la data puesto que el formato binario de todos los tipos de datos es el mismo. Pero cuando el cliente y el servidor se ejecutan en sistemas de arquitecturas diferentes es necesario una conversión de datos. Todas las implementaciones manejan este aspecto definiendo uno o más formatos estándar para los tipos de datos soportados por la implementación. La representación estandarizada de datos empleada por ONC-RPC es XDR, la cual impone un ordenamiento “big-endian” y un tamaño mínimo de 32 bits para cualquier campo. Dicha implementación utiliza tipificación implícita, es decir que sólo el valor de la variable es transmitida a través de la red.

### **2.5.6 Desempeño y seguridad**

Usualmente el desempeño empleando RPC disminuye entre 10 y 100 veces en comparación con un procedimiento local. Sin embargo, RPC debe ser visto como una herramienta que reduce el tiempo de desarrollo de aplicaciones distribuidas.

Con RPC existen los mismos problemas de seguridad relacionados con la ejecución remota de comandos.

## **2.6 Estudio de las Herramientas**

### **2.6.1 Java**

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrollados por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre diciembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java aún no lo es).

#### **2.6.1.1 Características de Java**

##### **Simple**

Es un lenguaje sencillo de aprender. Su sintaxis es la de C++ “simplificada”. Los creadores de Java partieron de la sintaxis de C++ y trataron de eliminar de este todo lo que resultase complicado o fuente de errores en este lenguaje. La herencia múltiple, la aritmética de punteros, por la gestión de memoria dinámica (que en Java se elimina de modo transparente para el programador gracias al recogedor basura) son ejemplos de "tareas complicadas" de C++ y en Java se han eliminado poco simplificado.

## **Orientado a Objetos**

Posiblemente sea el lenguaje más orientado a objetos de todos los existentes; en Java todo, a excepción de los tipos fundamentales de variables (int, char, long...) es un objeto.

## **Distribuido**

Java está muy orientado al trabajo en red, soportando protocolos como TCP/IP, UDP, HTTP y FTP. Por otro lado el uso de estos protocolos es bastante sencillo comparandolo con otros lenguajes que los soportan.

## **Robusto**

El compilador Java detecta muchos errores que otros compiladores solo detectarían en tiempo de ejecución o incluso nunca. A esclarecer así por ejemplo " if(a=b) then ... " o " int i; h=i\*2; " son dos ejemplos en los que el compilador Java no nos dejaría compilar este código; sin embargo un compilador C compilaría el código y generaría un ejecutable que ejecutaría esta sentencia sin dar ningún tipo de error).

## **Portable**

En Java no hay aspectos dependientes de la implementación, todas las implementaciones de Java siguen los mismos estándares en cuanto a tamaño y almacenamiento de los datos. Esto no ocurre así en C++, por ejemplo. En éste un entero, por ejemplo, puede tener un tamaño de 16, 32 o más bits, siendo la única limitación que el entero sea mayor que un short y menor que un long int. Así mismo C++ bajo UNIX almacena los datos en formato little endian, mientras que bajo Windows lo hace en big endian. Java lo hace siempre en little edian para evitar confusiones.

## **Arquitectura Neutral**

El código generado por el compilador Java es independiente de la arquitectura: podría ejecutarse en un entorno UNIX, Mac o Windows. El motivo de esto es que el que realmente ejecuta el código generado por el compilador no

es el procesador del ordenador directamente, sino que este se ejecuta mediante una máquina virtual. Esto permite que los Applets de una web pueda ejecutarlos cualquier máquina que se conecte a ella independientemente de que sistema operativo emplee (siempre y cuando el ordenador en cuestión tenga instalada una máquina virtual de Java).

### **Rendimiento medio**

Actualmente la velocidad de procesado del código Java es semejante a la de C++, hay ciertos pruebas estándares de comparación (benchmarks) en las que Java gana a C++ y viceversa.

Esto es así gracias al uso de compiladores just in time, compiladores que traduce los bytecodes de Java en código para una determinada CPU, que no precisa de la máquina virtual para ser ejecutado, y guardan el resultado de dicha conversión, volviéndolo a llamar en caso de volverlo a necesitar, con lo que se evita la sobrecarga de trabajo asociada a la interpretación del bytecode.

No obstante por norma general el programa Java consume bastante más memoria que el programa C++, ya que no sólo ha de cargar en memoria los recursos necesario para la ejecución del programa, sino que además debe simular un sistema operativo y hardware virtuales (la máquina virtual). Por otro lado la programación gráfica empleando las librerías Swing es más lenta que el uso de componentes nativos en las interfaces de usuario.

En general en Java se ha sacrificado el rendimiento para facilitar la programación y sobre todo para conseguir la característica de neutralidad arquitectural, si bien es cierto que los avances en las máquinas virtuales remedian cada vez más estas decisiones de diseño.

### **Multithread**

Soporta de modo nativo los threads, sin necesidad del uso de de librerías específicas (como es el caso de C++). Esto le permite además que cada Thread de una aplicación

java pueda correr en una CPU distinta, si la aplicación se ejecuta en una máquina que posee varias CPU. Las aplicaciones de C++ no son capaces de distribuir, de modo transparente para el programador, la carga entre varias CPU.

### **2.6.2 JasperReports**

Un punto muy importante de los sistemas se presenta a la hora de mostrar la información resultante de los procesos de las aplicaciones y/o del día a día. Principalmente, cuando esto implica tomar decisiones comerciales o gerenciales. Dicha información. Por lo general, está almacenada en bases de datos o, en su defecto, en archivos planos. Es posible realizar consultas a las bases de datos, leer estos archivos, diseñar y codificar ventanas o interfaces de usuarios, para interactuar con esa información. Pero este proceso se torna complicado cuando es necesario sumarle la funcionalidad de impresión.

La “customización” rápida de la información que estamos mostrando, o bien cuando la aplicación tiene múltiples interfaces. Es decir, cuando el usuario puede utilizarla como un cliente Windows o desde un navegador. A partir de estos puntos, toma importancia la generación de reportes.

Tanto para obtener dinamismo en las consultas, como para lograr múltiples vistas e incluso, para facilitar el mantenimiento posterior y su extensibilidad. Para Java, durante un tiempo, éste fue uno de los puntos más débiles del lenguaje. Hoy en día, existen diversas librerías y herramientas dedicadas para la rápida generación de reportes, JasperReports es una de las más conocidas e interesantes, que combinada con herramientas para el diseño, facilita y agiliza la generación, la pre visualización y la impresión de los reportes.

JasperReports es una poderosa herramienta para generar reportes en Java, con la habilidad de producir contenido completo para la pantalla, directo para impresora o en diferentes formatos de archivo (PDF, XLS, CSV y XML entre otros).

Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones web, para generar contenido dinámico.

Su propósito principal es ayudar a crear documentos de tipo páginas, preparados para imprimir en una forma simple y flexible.

JasperReports se usa comúnmente con iReport, un *front-end* gráfico de código abierto para la edición de informes. Se encuentra bajo licencia libre GNU, por lo que es Software libre.

### **2.6.3 Google Web Toolkit (GWT)**

#### **La emergencia de AJAX**

A inicios del siglo, las compañías no veían lucrativo el uso de la tecnología web, se encontraban en el dilema de saber si era o no la tecnología del futuro, pero algo fundamental empezó a tomar forma surgiendo tecnologías como la creación de HTML limpio, cascade style sheets (CSS's) para páginas elegantes con estilo y la adición de JavaScript.

Esta evolución de la “nueva web” se centró en el dominio de los fundamentos básicos de la tecnología web, antes de añadir otra capa a la arquitectura web. En lugar de construir las cosas viejas de nuevo con la tecnología nueva, los desarrolladores innovaron con la tecnología antigua. Las personas sin conocimientos técnicos disfrutaban con las aplicaciones web anteriores, y en algunos casos, estas aplicaciones se desempeñaban mejor que aplicaciones de escritorio. Con la incorporación de los documentos CSS junto a las páginas web dieron una mayor facilidad de uso, por ende más gente se iba adaptando e integrando a las funcionalidades que brindaban las aplicaciones web.

Surgieron páginas web como son los blogs y redes sociales en donde la administración de estos sitios resultaba muy compleja, por ejemplo, un usuario ingresaba a su cuenta con su nombre de usuario y contraseña, luego de ellos se cargaba una nueva página con toda la información que ellos tenían configurado dentro de su perfil, cualquier actividad que se realizaba dentro de su entorno implicaba nuevas páginas a cargarse dando como resultado aplicaciones web que dejaban mucho que desear.



Esta recarga de páginas constante poco a poco iba desapareciendo con el surgimiento de las tecnologías de Javascript, esto permitía que las solicitudes hacia el servidor se lo realicen de manera asíncrona, actualizando las páginas sin necesidad de recargarlas nuevamente. El uso de Javascript dio un gran apoyo a los programadores ya que ellos podían poner mayor lógica del lado del cliente, reduciendo la carga en el servidor, todos estas operaciones parecían en un principio “trucos” en el desarrollo de aplicaciones web que más tarde la mayoría de las personas usarían y sería conocida a estas técnicas como **Asynchronous JavaScript y XML (XML)**

El uso de técnicas AJAX para la creación de aplicaciones web provocó un importante cambio en la arquitectura de aplicaciones web. Las interfaces ya no se generaban a través de plantillas en el servidor, sino todo a partir de Javascript en el lado del cliente, permitiendo que la aplicación en su lado sea casi independiente del servidor.

A medida que la adopción de AJAX incrementaba, la construcción de aplicaciones mas grandes era necesaria, lo cual implicaba un aumento en la complejidad, así como también en herramientas de desarrollo potentes y el uso de la orientación a objetos. Sin embargo, AJAX evoluciono a partir de tecnologías que en un inicio no era para la construcción de aplicaciones grandes y no había herramientas especializadas en ello.

Han ido surgiendo marcos de trabajo, los cuales proporcionaban una base para la creación de aplicaciones como es el caso de Dojo, proyecto impulsado por la comunidad, que lo único que hace es incorporar bibliotecas que permitían el desarrollo de aplicaciones AJAX. En definitiva, AJAX estaba siendo usado para algo que nunca fue diseñado a hacer.

### **Replanteamiento de las Aplicaciones Web**

La escritura de toda la lógica de las aplicaciones web en Javascript resultaba difícil para los desarrolladores programar sistemas grandes y complejos, cosa que no resulta igual en una aplicación de escritorio ya que se podía elegir entre distintos lenguajes de programación.

Muchas empresas tras conocer el éxito y las limitaciones de AJAX, proponían nuevas herramientas como una alternativa, brindando una experiencia en la navegación de los sistemas web más enriquecedor, las aplicaciones creadas con estas tecnologías eran conocidas como **Aplicaciones Dinámicas de Internet (ARI)**.

En las secciones siguientes se describen varias alternativas al uso de AJAX para la construcción de **RIA's (Rich Internet Applications)**. El tiempo dirá si alguno de estos logrará convertirse en el estándar de facto para este tipo de aplicaciones. Cabe recalcar que estas alternativas en esencia son parecidas a AJAX por lo que vale la pena decir que AJAX de manera indirecta nunca dejara de ser la base para el desarrollo de este tipo de sistemas.

### *Adobe Flash y Flex*

Adobe Flash ha sido el plugin para navegadores más exitoso desde el año 98 e instalado por miles de personas que dispones de un browser. Su triunfo se debe a que el plugin y los archivos SWF son relativamente pequeños, además de poseer herramientas muy poderosas de desarrollo para la construcción de este tipo de archivos. Estos ficheros pueden disponer de animaciones rápidas y gráficos de diseño amigable.

Flash ha sido usado ampliamente como una herramienta para juegos y desplegar anuncios, ha incorporado un elemento en el desarrollo de aplicaciones web conocido como **Adobe Flex**, el cual es un kit de desarrollo para la creación de aplicaciones basadas en Flash. Flex usa **MXML (Multimedia eXtensible Markup Language)** y **ActioScript**, junto con un entorno de desarrollo integrado IDE, para crear aplicaciones que recopilan hasta archivos SWF.

Si bien el entorno de desarrollo Flash está orientado hacia el uso por los diseñadores gráficos, Flex está dirigido a desarrolladores de software y proporciona servicios de datos tales como la interacción remota y soluciones de mensajería para comunicarse con una aplicación Java 2 Enterprise Edition (J2EE) servidor.

### ***Microsoft Silverlight***

Microsoft tiene una larga historia de apoyo al desarrollo de aplicaciones cliente para su sistema operativo Windows. Sus herramientas y la documentación siempre han sido muy buenas, y siempre han hecho un esfuerzo de avanzar en la tecnología web. Sin embargo, muchos de sus esfuerzos han sido en contraste con el carácter democrático de la web. Microsoft inicialmente decidió eliminar el Java Runtime Environment (JRE) de su distribución de Internet Explorer por lo que no se era compatible con los applets de Java y en su lugar proporcionaron la capacidad de integrar Objetos ActiveX, que utilizan su propia tecnología Component Object Model (COM).

Microsoft aprendió de ese error y ahora ofrece Silverlight, en base a .NET, como un plugin para muchos navegadores. Silverlight permite a los desarrolladores construir RIA's mediante el desarrollo de técnicas más avanzadas de JavaScript. Sin embargo, esta solución, que se beneficia de buenas herramientas del Framework de .NET y del apoyo de Microsoft, no tiene una amplia distribución y es controlado por Microsoft, en lugar de una organización de estándares.

### ***Java FX***

Java disfruta desde los inicios de una amplia distribución en los navegadores con el uso de los applets de Java. Estos applets de Java utiliza el Java Runtime Environment del navegador para proporcionar funcionalidad de programación más avanzada que HTML y Javascript podrían proporcionar. Sin embargo, los applets de Java se ganaron una reputación mala debido a la carga lenta e interfaces feas en comparación con las descargas rápidas de Flash y sus interfaces impresionantes. Además, debido a que Microsoft no brindó compatibilidad en el Internet Explorer implicó una caída de los applets.

Como lo demuestra el éxito de Flash y AJAX, los desarrolladores quieren más funcionalidad de programación en el navegador, y con Microsoft entrando en juego

con Silverlight, el momento puede ser adecuado para Java para tener éxito en un segundo intento de aplicaciones basadas en un navegador.

Sun Microsystems anunció el Java FX en el JavaOne el 05-2007, pocos días después de que Microsoft anunció Silverlight 1.1. La familia de tecnologías Java FX ayudan a crear aplicaciones RIA. Java FX se compone de una secuencia de comandos y un módulo móvil que se ejecutan en un estándar.

### **Ingeniería de Software para el AJAX – Google Web Toolkit (GWT)**

Google lanzó su toolkit sin hacer mucha algarabía de su nuevo producto, aunque no fue aceptada universalmente como un gran producto de Google, hubo muchos comentarios a favor y en contra, pero en general el producto se tuvo una buena aceptación. Para la mayoría de desarrolladores parecían un poco confundidos por colocar el kit de herramientas en la web, se necesitaba un nuevo enfoque de las aplicaciones web, si uno se toma el tiempo para analizar a profundidad, es probable que se encuentre al **GOOGLE WEB TOOLKIT** como una herramienta muy poderosa para crear aplicaciones AJAX.

Hay muchos grandes bibliotecas de JavaScript que ayudan a construir grandes aplicaciones AJAX. Sin embargo, GWT fundamentalmente difiere de estas bibliotecas, proporcionando una gran cantidad de herramientas de ingeniería de software de Java para el uso de aplicaciones AJAX, en lugar de proporcionar una biblioteca rica en características (aunque su biblioteca tiene muchas características).

También puede incluir las bibliotecas de JavaScript, y su aplicación compila hasta ser distribuido como una aplicación AJAX utilizando estándares web, y no requiere ningún nuevo plugin. Para el navegador aparece como cualquier otra aplicación AJAX, pero para el desarrollador es como construir una aplicación de escritorio.

La mejora en la flexibilidad y la productividad de desarrollo beneficia a los desarrolladores en el tiempo de programación y para el usuario se le presenta como una aplicación sólida con buena usabilidad. El proceso de desarrollo de las aplicaciones AJAX puede aprovechar de las herramientas de software de alta calidad de ingeniería tales como JUnit para el desarrollo basado en pruebas e IDE's como Eclipse que proporcionan un

rendimiento superior de depuración y control en tiempo de compilación de errores sobre la marcha.

GWT contiene muchas herramientas que ayuden a la creación de aplicaciones AJAX, incluyendo una biblioteca de interfaz de usuario de Widgets y Panels, las bibliotecas para realizar comunicaciones asíncronas a través de HTTP hacia el servidor o llamadas a procedimientos remotos (RPC), herramientas para interoperar con otras aplicaciones Web utilizando JavaScript, JSON y XML, y el acceso a un desarrollo maduro para la ingeniería de software.

### **Construir Rich Interfaces con Widgets y Panels**

GWT proporciona una biblioteca de widgets y paneles que se pueden utilizar en el código de Java de la aplicación AJAX. Están construidas con HTML usando Java Script para controlar los eventos, cuando la aplicación se compila a JavaScript, el navegador los ejecuta al igual que una página web tradicional, sin necesidad de un solo plugin o incluso el JRE.

Los widgets en GWT le dan un control mediante programación de usuarios bien definida. Algunos de los widgets cumplen el estándar de etiquetas HTML como **img** como para las imágenes y los anclajes de los enlaces, junto con los controles de aplicación, tales como botones y cajas de carga de archivos. También se pueden usar widgets más complejos, como el widget de árbol para mostrar un control de árbol que normalmente se ven en las aplicaciones de escritorio. GWT también incluye paneles los cuales facilitan el diseño de la aplicación web, las cuales siguen reglas estrictas para que la presentación sea la misma en los distintos navegadores.

### **Obtener el Mejor Rendimiento con Comunicación Asíncrona**

Las aplicaciones AJAX mejoran el rendimiento del servidor web manteniendo el estado de la aplicación y así aliviar el servidor de esta tarea. Esto significa que en lugar de cargar una nueva página web para cada acción, las aplicaciones AJAX pueden ejecutar varias acciones en conjunto en el lado del cliente y enviar datos como un lote sólo cuando sea necesario. También pueden cargar los datos gradualmente a medida

que el usuario navega a través de la interfaz en lugar de cargar el conjunto de datos cuando se carga la página web.

GWT ofrece una librería HTTP para enviar y recuperar datos de un servidor de forma asincrónica, así como una implementación de RPC que se conecta a un servlet de Java y hace la invocación de métodos en el servidor tan fácil como hacer una llamada al método local.

### **Interoperación a través de estándares web y Servicios Web**

La guía no pretende ser un marco para construir aplicaciones, sino que pretende ser un conjunto de herramientas que mejoren la capacidad de hacer aplicaciones AJAX. Parte de lograr esto consiste en aprovechar los estándares web que existen actualmente y los servicios web. En lugar de comprometerse con una tecnología, las herramientas de GWT le permiten interoperar con cualquier tecnología.

En particular, GWT proporciona bibliotecas de JSON y XML que permiten traducir datos entre XML o JSON y sus objetos de la librería correspondiente. GWT proporciona una manera de conectar la disposición de sus aplicaciones y widgets a archivos CSS para aprovechar esta tecnología de gran estilo y su riqueza de conocimientos de diseño gráfico. El kit de herramientas le permite integrarse con otras existentes bibliotecas de JavaScript, proporcionando el código JavaScript Native Interface (JSNI), una forma de interactuar directamente con JavaScript desde Java y viceversa.

### **Desarrollo rápido usando herramientas Java**

Quizá el mejor truco de GWT es su capacidad para llevar la ingeniería de software en el desarrollo de aplicaciones AJAX, sin necesidad de instalar un nuevo plugin o JRE en el lado del cliente. Se puede utilizar todas las herramientas de desarrollo utilizados en la creación de aplicaciones Java para crear también aplicaciones AJAX con GWT. Estas herramientas incluyen entornos de desarrollo como Eclipse, que permite escribir el código y obtener información inmediata acerca de los errores, compilar el código para capturar errores antes de ejecutar la aplicación, y depurar el

código al correr la aplicación y las variables de inspección que se ejecutan. Se puede aprovechar de JUnit para llevar a cabo el desarrollo de pruebas de sesiones, donde se escribe la prueba antes de escribir el código, y Ant para automatizar la recopilación e implementación.

### **Vista General de los paquetes de GWT**

A pesar de las ventajas principales de GWT de usar el lenguaje Java y de utilizar sus herramientas para crear aplicaciones web con la tecnología estándar, vale la pena dar a relucir la considerable cantidad de paquetes Java que dispone y con los cuales se hará uso para crear sistemas web.

Los paquetes de la biblioteca se encuentran en el archivo **gwt-user.jar**, que se encuentra en el directorio de instalación de GWT. Si no está familiarizado con un archivo *JAR*, piense en él como si se tratase de un archivo *ZIP*, es decir este tipo de archivos contienen en su interior un conjunto de archivos de clase compilados.

Al momento de compilar el proyecto, no es necesario incluir este archivo *JAR* dentro del classpath puesto que el proceso de compilación lo incluye. Además, al momento de subir el proyecto a un servidor web o de aplicaciones no es necesario agregar la librería ya que al compilarse este código es transformado a un lenguaje nativo que fácilmente es interpretado por la mayoría de navegadores web.

En la figura siguiente se puede observar la estructura del paquete *gwt-user.jar*, este se divide en 5 categorías: *user interface*, *server calls*, *data formats*, *JRE Emulation*, and *utility*.

La categoría *user interface* es probablemente la librería más utilizada en el desarrollo de aplicaciones.

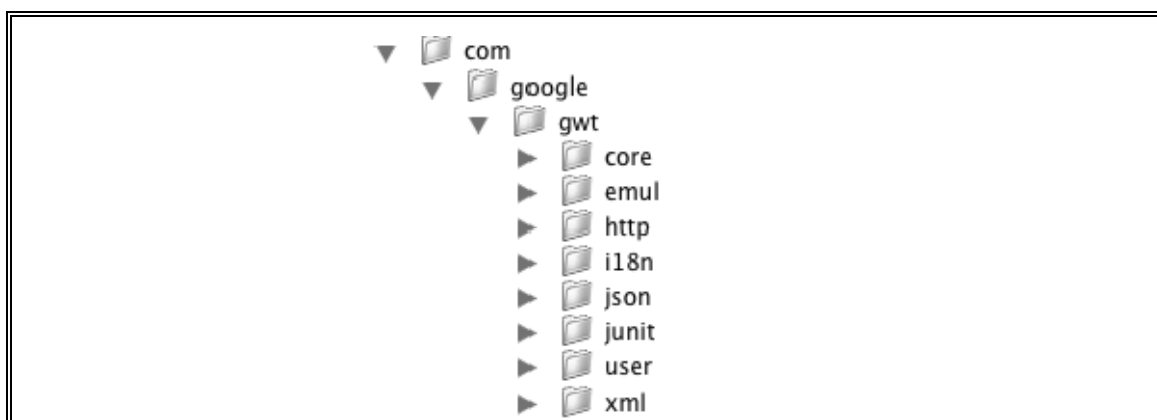


Figura 2-6: Estructura del paquete gwt-user.jar

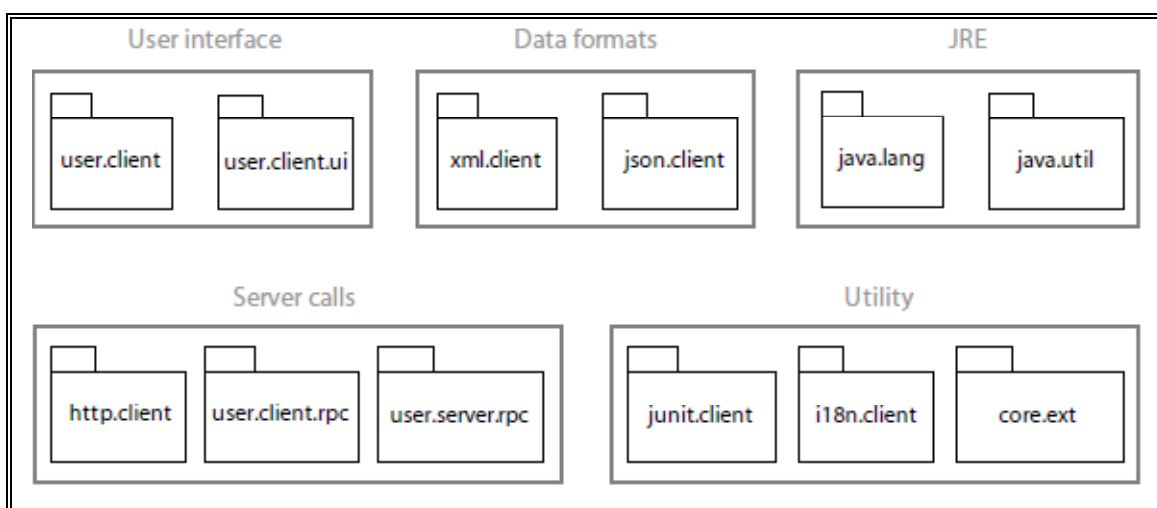


Figura2-7: División de las librerías de GWT.

### **com.google.gwt.user.client**

Este paquete proporciona un acceso básico a la API de un navegador web de forma dinámica. La clase DOM en este paquete permite la edición directa de un documento web a través de su objeto modelo del documento. Los paneles y widgets de GWT, forman parte de la siguiente librería en esta misma categoría, y son usados para la construcción de ellos mismos. También se puede utilizar esta librería para la construcción de widgets personalizados.



### **com.google.gwt.user.client.ui**

En este paquete se prescribe un marco de trabajo para la construcción de las interfaces de usuario muy similar al diseño con Java's Abstract Toolkit (AWT)<sup>5</sup>. Este paquete hace un gran trabajo al ayudar a crear y utilizar dinámicamente componentes reutilizables de interfaz de usuario.

Una segunda categoría de paquetes, *server calls*, ayuda a hacer llamadas remotas a un servidor. Estos paquetes contienen clases que ayudan a las aplicaciones GWT a comunicarse con el servidor.

### **com.google.gwt.http.client**

Este paquete HTTP crea una potente independencia del navegador alrededor del objeto XMLHttpRequest, que las aplicaciones AJAX utilizan para la comunicación asíncrona.

### **com.google.gwt.user.client.rpc y com.google.gwt.user.server.rpc**

Estos paquetes de aplicaciones dan la capacidad de comunicarse con el servidor a través de llamadas a procedimiento remoto. Esta biblioteca utiliza la comunicación RPC llamadas de retransmisión desde una interfaz basada en Java a nuestro servidor sin que nos obligue a pensar en los detalles del protocolo. También se utiliza desde el navegador el objeto JavaScript XMLHttpRequest en su comunicación.

Estas dos categorías de paquetes cubren las dos principales áreas de funcionalidad, interfaces de usuario dinámicas y servidor de comunicaciones asíncronas, necesarios en una biblioteca para apoyar una solicitud Ajax. Si lo que desea es crear una interfaz dinámica, sólo es necesario la primera serie de bibliotecas, si sólo requieren una comunicación asíncrona del servidor, sólo es necesario el segundo conjunto de bibliotecas.

---

<sup>5</sup> Se trata de una biblioteca de clases Java para el desarrollo de Interfaces de Usuario Gráficas. La versión del AWT que Sun proporciona con el JDK se desarrolló en sólo dos meses y es la parte más débil de todo lo que representa Java como lenguaje. El entorno que ofrece es demasiado simple, no se han tenido en cuenta las ideas de entornos gráficos novedosos, sino que se ha ahondado en estructuras orientadas a eventos, llenas de callbacks y sin soporte alguno del entorno para la construcción gráfica

El compilador de GWT sabe que bibliotecas utilizar en la aplicación y es lo suficientemente inteligente como para compilar el código necesario para activar JavaScript. Esto hace al código de la aplicación lo más pequeño posible al distribuirlo. Esta es otra ventaja de utilizar el kit de herramientas en lugar de otros marcos JavaScript, no tendría la capacidad para reducir el tamaño de las bibliotecas con el mismo nivel de granularidad.

La tercera categoría de paquetes proporciona a su aplicación la capacidad para analizar y construir formatos de datos.

`com.google.gwt.xml.client`

Este paquete le da a su aplicación XML la capacidad para analizar un documento XML e iterar sobre su contenido a través de un modelo de objeto. También puede usar este modelo de objeto y generar un documento XML. El uso de este paquete en el paquete HTTP le permite enviar y recibir documentos XML desde y hacia un servidor.

`com.google.gwt.json.client`

JSON es similar al paquete de XML, excepto al formato. El paquete es JSON en lugar de XML. JSON es un formato de datos jerárquica similar a XML, pero más ligero de peso (más pequeño). Esta biblioteca le permite generar y analizar los datos JSON.

La categoría *JRE Emulation* contiene un subconjunto de la biblioteca de Java Runtime Environment para ayudar en las tareas de programación comunes, incluyendo los tipos básicos de Java, las excepciones y las colecciones. El lenguaje Java viene con muchas bibliotecas que ayudan en la construcción de muchos tipos de aplicaciones. En lugar de crear más bibliotecas nuevas, GWT emula dos de las bibliotecas más utilizadas en el JRE. Emulando a estas bibliotecas se tiene la ventaja de ofrecer clases que los desarrolladores ya pueden estar familiarizados. Se considera el término emulación, ya que las clases JRE no se utilizan cuando una aplicación GWT se ha implementado, lo que el compilador de GWT hace es un reemplazo del uso de las clases JRE con el equivalente a código JavaScript.

**java.lang**

Este paquete contiene las clases básicas de Java como *Integer* y *String*.

**java.util**

El paquete *utils* proporciona muchas de las clases de colección básica disponible con Java, incluyendo *maps*, *lists* y *sets*.

La última categoría de paquetes es un conjunto de clases de utilidades diversas, pero importantes.

**com.google.gwt.junit.client**

Este paquete JUnit proporciona clases para ayudar a escribir las pruebas unitarias para su aplicación utilizando JUnit.

**com.google.gwt.i18n.client**

GWT ofrece apoyo a la internacionalización, y este paquete ofrece clases para ayudar a acceder a ella.

**com.google.gwt.core.ext**

Este paquete de extensión proporciona clases que ayudan a extender el compilador GWT.

Estos paquetes son completamente opcionales. Usted puede elegir cualquier combinación de paquetes para utilizar en la aplicación, y el compilador de GWT se asegura de que sólo se utilice el código que está incluido en la aplicación compilada. El desarrollador no está limitado a usar sólo los paquetes que se entregan con el kit de herramientas, también puede incluir bibliotecas de terceros para su uso con GWT para extender la funcionalidad de su aplicación. Por ejemplo, la librería Ext GWT y Smart GWT usados en el desarrollo del sistema Ge@ y de los cuales se hablarán en secciones posteriores.

## **Comunicación con el servidor**

Todas las aplicaciones GWT se ejecutan como código JavaScript en el navegador web del usuario final. Sin embargo, con frecuencia se desea crear algo más que un aplicación de lado del cliente independiente. La aplicación deberá comunicarse con un servidor web, para el envío de solicitudes y recepción de actualizaciones.

En las aplicaciones web tradicionales la comunicación con el servidor requería la carga de una nueva página HTML. En contraste, las aplicaciones desarrolladas con AJAX se cargan toda la interfaz gráfica para el cliente y realiza llamadas asíncronas al servidor para enviar y recibir solo los datos en sí, permitiendo una mayor fluidez y sensibilidad, aumentando así el ancho de banda y liberando de carga al servidor.

## **Escogiendo una estrategia de implementación**

GWT ofrece diferentes maneras de interactuar con el servidor, la elección de ellas dependerá del formato de datos que se utilizan en última instancia.

### **Remote Procedure Calls (GWT RPC)**

Si se está ejecutando Java en el servidor y se está creando una interfaz para la lógica de la aplicación de negocio del lado del servidor, GWT RPC es probablemente su mejor opción. GWT RPC es un mecanismo para pasar objetos Java desde y hacia un servidor a través de HTTP estándar. Se puede utilizar el marco de GWT RPC para realizar llamadas de forma transparente a los servlets Java y GWT cuida los detalles de bajo nivel como la serialización de objetos.

### **¿Qué es GWT RPC?**

El framework GWT RPC hace que sea fácil para el cliente y los componentes del servidor de la aplicación web el intercambio de objetos Java a través de HTTP. El código de servidor que se invoca desde el cliente se refiere a menudo como un servicio. La implementación de un servicio GWT RPC se basa en la arquitectura de servlet conocido de Java. Dentro del código de cliente, va a utilizar una clase generada automáticamente proxy para realizar llamadas al servicio. GWT se

encargará de la serialización de los objetos Java que se envían y regresan, los argumentos de las llamadas al método y el valor de retorno.

Es importante recalcar que los servicios GWT RPC no son los mismos que los servicios web basados en SOAP o REST. Estos simplemente son tomados como un método de peso ligero para la transferencia de datos entre el servidor y la aplicación GWT en el cliente.

### Componentes de JAVA en el mecanismo de GWT RPC

Cuando se hace uso de los GWT RPC, se ven envueltos tres elementos en la llamada a los procedimientos remotos que se encuentran ejecutándose en el servidor.

- El servicio que corre en el servidor (el método al cual se llama).
- El código del cliente que invoca el servicio.
- Y los objetos Java que se pasan entre el servidor y el cliente.

Tanto el servidor y el cliente tiene la capacidad de serializar y deserializar los datos para los objetos que se puede pasar entre ellos como texto normal.

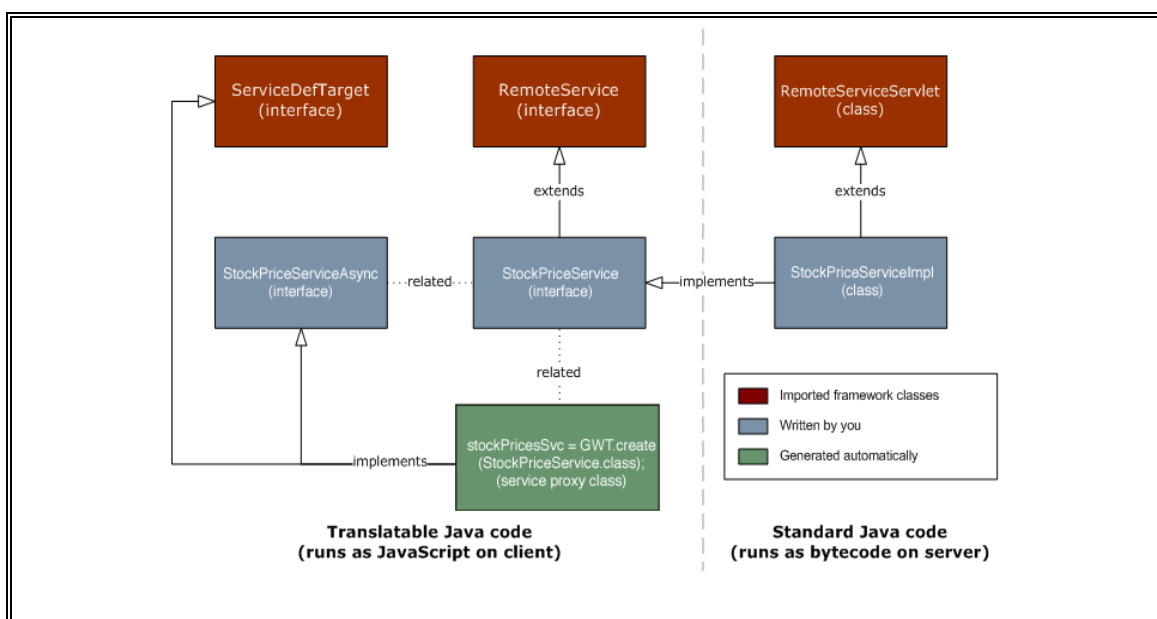


Figura 2-8: Mecanismo de GWT RPC

Para definir un GWT RPC es necesario tres componentes:

- Definir una interface para el servicio este extiende de la clase RemoteService y lista todos los métodos RPC implementados.
- Se crea una clase que extiende de la clase RemoteServiceImpl e implementa la interface que se haya creado.
- Se define una interfaz asíncrona al servicio para ser llamado desde el lado del cliente.

Una implementación del servicio debe extender RemoteServiceServlet y debe implementar la interfaz de servicio asociado. Tenga en cuenta que la implementación del servicio no implementa la versión asíncrona de la interfaz de servicio. Todas las implementaciones de servicios son en última instancia, un servlet, pero en lugar de extender HttpServlet se extiende RemoteServiceServlet. RemoteServiceServlet maneja automáticamente la serialización de los datos pasados entre el cliente y el servidor e invoca el método previsto en la implementación del servicio.

### **Recuperación de datos a través de JSON**

Si la aplicación se comunica con un servidor que no puede albergar servlets Java, o uno que ya utiliza otro formato de datos como JSON o XML, se puede hacer peticiones HTTP para recuperar los datos. GWT ofrece clases genéricas HTTP que se pueden utilizar para construir la petición, y las clases de JSON y XML de cliente se pueden utilizar para procesar la respuesta. También puede utilizar los tipos de recubrimiento para convertir objetos de JavaScript a objetos de Java que se puede interactuar con el IDE, mientras se encuentre en vías de desarrollo.

### **¿Qué es JSON?**

JSON es un lenguaje universal e independiente para el formato de los datos, es similar al XML. Mientras que XML utiliza etiquetas, JSON se basa en la notación literal de objetos de JavaScript. Por lo tanto, el formato es más simple que XML. En

general, los datos codificados en JSON son menos detallados que los datos equivalentes en XML por ello de su rápida descarga.

### **Haciendo solicitudes de Cross-Site para JSONP**

Si usted está creando una aplicación mashup<sup>6</sup> que hay que utilizar los datos de uno o más servidores web remotos, usted tendrá que evitar las restricciones de acceso a SOP (política del mismo origen). GWT dispone de librerías que permitan el acceso a datos de distintos servidores y poderlos cargar en nuestra propia aplicación.

En definitiva y de manera resumida, GWT ofrece un marco de trabajo amplio en donde el desarrollador tiene la capacidad de experimentar y crear grandes aplicaciones, además que tiene una nueva manera de ver la programación orientada a la web.

GWT brinda un mundo de opciones para el desarrollo de aplicaciones web haciendo uso extensivo de lo que es AJAX combinado con las ventajas de utilizar llamadas asíncronas al servidor de las tres maneras diferentes que se detallaron anteriormente.

El desarrollador no necesita tener un gran conocimiento de lo que es AJAX, lenguaje JSON y XML, puesto que la programación realizada con GWT es JAVA puro, es decir lo único que se necesita saber es conocimientos de JAVA y con eso el usuario está listo para empezar a desarrollar aplicaciones con un nuevo enfoque.

### **Instalación del GWT Framework en sistemas operativos Windows**

En la siguiente sección se va a detallar el proceso de instalación del marco de trabajo GWT en el IDE's Netbeans 6.x

Instalación de GWT en Netbeans 6.x

Para la instalación de GWT en Netbeans son necesarios los siguientes requisitos:

---

<sup>6</sup> Un mashup en informática es una aplicación web, que resulta de la fusión o mezcla de otros servicios que ya han sido creados, esto se lleva a cabo por medio de una API o RSS que se conecta a los otros servicios y le trae a la nueva aplicación toda la información necesaria.

- Netbeans IDE, Java Bundle, versión 6.x
- Java SE Development Kit (JDK) o la actualización JDK 6. Puede descargar la versión más reciente de JDK en <http://java.sun.com/javase/downloads>.
- Acceso a la web.

Si se cumple con todos estos requerimientos, se procede a la instalación de GWT el cual difiere en algunos aspectos, en Netbeans no se dispone de un plugin que incluya el SDK de GWT al momento de instalarlo, por lo que es necesario descargárselo junto con el plugin, el proceso que se sigue se detalla a continuación:

La instalación del plugin se lo puede realizar de dos maneras:

- a. Descargándose el componente directamente desde el IDE Netbeans.
  1. Dando clic en **Tools → Plugins**.

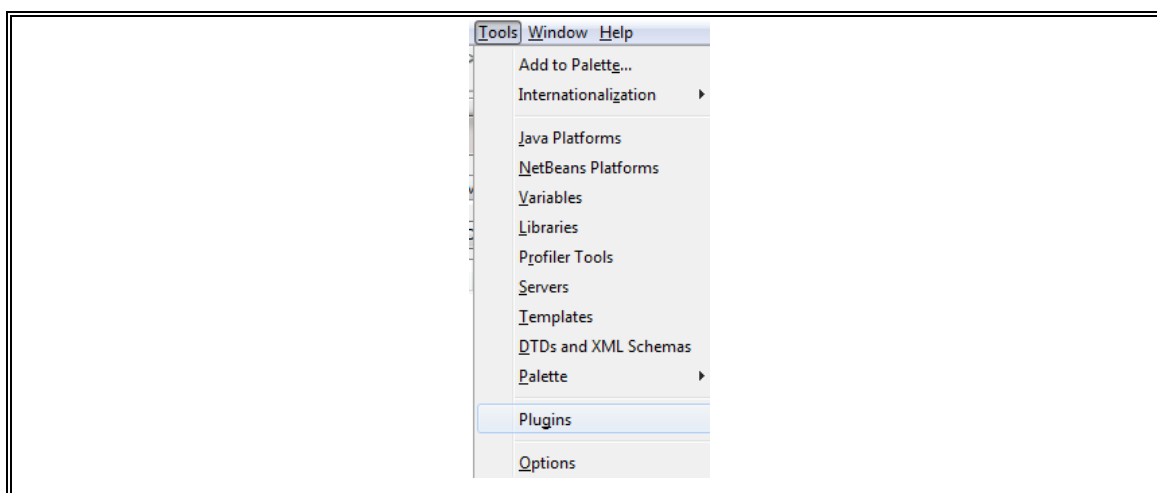


Figura 2-10: Menú Tools (Plugins).

2. La ventana que se abre es la que gestiona los plugins instalados y disponibles para Netbeans.



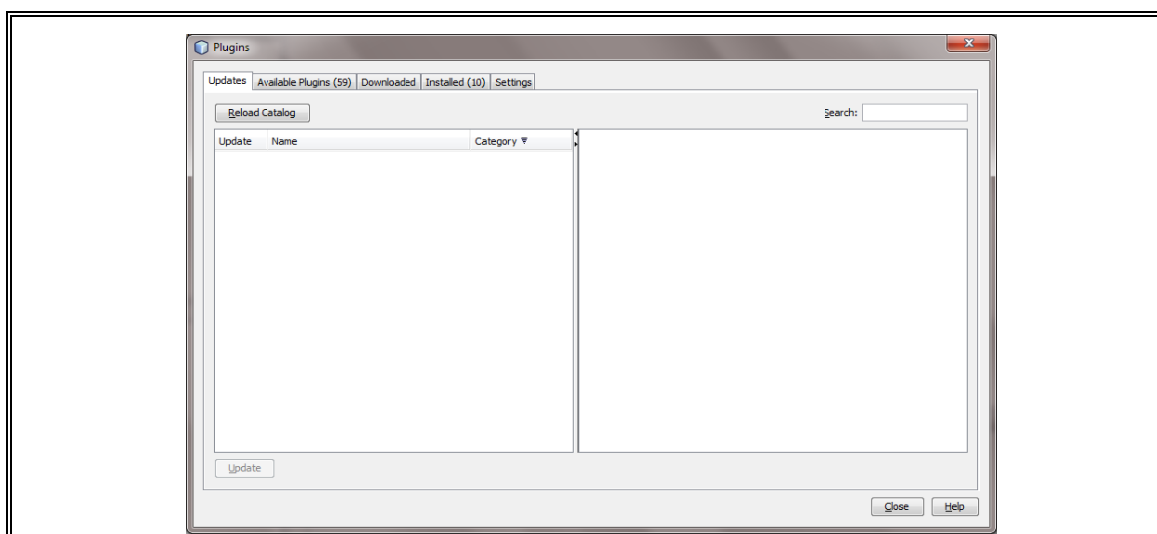


Figura 2-11: Gestión de Plugins.

3. En **search** se escribe “GWT” y a continuación se presenta el plugin para ser instalado como se observa en la figura:

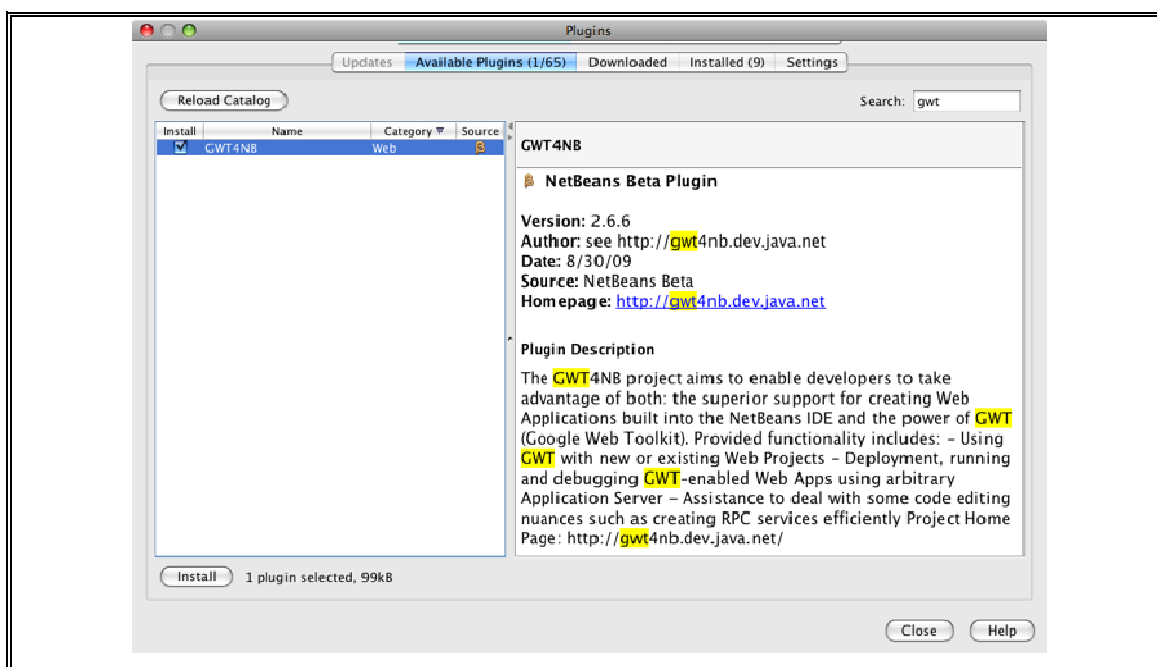


Figura 2-12: Instalación del plugin GWT.

Se selecciona el plugin GWT4NB y se procede a pulsar en Install.

- b. Otra manera de instalar el plugin es descargarlo directamente desde el repositorio, para ello accedemos a la siguiente dirección:<https://gwt4nb.dev.java.net/servlets/ProjectDocumentList?folderID=7081&expandFolder=7081&folderID=0>

Y seleccionamos la versión que sea compatible con el SDK que se detallara más adelante la instalación, en este caso se seleccionó la última versión disponible al momento de escribir esta tesis.



Version	Release Date	Size	Compatible SDKs
GWT4NBV2.6.21	2010 at 7:57:26 PM	196.42 kB	NetBeans 6.7.1 or 6.8, GWT 1.5, 1.6, 1.7 or 2.0, Java 5 or 6
GWT4NBV2.6.22	Stable <a href="#">kore</a> on Saturday, April 10, 2010 at 8:57:56 AM	202.65 kB	NetBeans 6.7.1, 6.8 or 6.9, GWT 1.5, 1.6, 1.7 or 2.0, Java 5 or 6
GWT4NBV2.6.23	Stable <a href="#">kore</a> on Sunday, March 7, 2010 at 3:44:19 PM	204.76 kB	NetBeans 6.7.1, 6.8 or 6.9, GWT 1.5, 1.6, 1.7 or 2.0, Java 5 or 6
GWT4NBV2.6.23	Stable <a href="#">kore</a> on Sunday, May 30, 2010 at 10:02:14 AM	204.76 kB	NetBeans 6.7.1, 6.8 or 6.9, GWT 1.5, 1.6, 1.7 or 2.0, Java 5 or 6
GWT4NBV2.6.3	Stable <a href="#">kore</a> on Thursday, February 25, 2010 at 7:58:22 PM	97 kB	NetBeans 6.7.1, GWT 1.5, 1.6 or 1.7, Java 5 or 6
GWT4NBV2.6.4	Stable <a href="#">kore</a> on Thursday, February 25, 2010 at 7:58:30 PM	98.37 kB	NetBeans 6.7.1, GWT 1.5, 1.6 or 1.7, Java 5 or 6
GWT4NBV2.6.5	Stable <a href="#">kore</a> on Thursday, February 25, 2010 at 7:58:49 PM	98.39 kB	NetBeans 6.7.1, GWT 1.5, 1.6 or 1.7, Java 5 or 6
GWT4NBV2.6.6	Stable <a href="#">kore</a> on Thursday, February 25, 2010 at 7:58:58 PM	98.07 kB	NetBeans 6.7.1, GWT 1.5, 1.6 or 1.7, Java 5 or 6
GWT4NBV2.6.7	Stable <a href="#">kore</a> on Thursday, February 25, 2010 at 7:59:13 PM	100.97 kB	NetBeans 6.7.1, GWT 1.5, 1.6 or 1.7, Java 5 or 6
GWT4NBV2.6.8	Stable <a href="#">kore</a> on Thursday, February 25, 2010 at 7:56:10 PM	101.31 kB	NetBeans 6.7.1, GWT 1.5, 1.6 or 1.7, Java 5 or 6
GWT4NBV2.6.9	Stable <a href="#">kore</a> on Friday, December 11, 2009 at 8:59:45 AM	102.39 kB	NetBeans 6.7.1, GWT 1.5, 1.6, 1.7 or 2.0ms2, Java 5 or 6
GWT4NBV2.7.4	Stable <a href="#">kore</a> on Friday, April 30, 2010 at 9:26:55 PM	222.3 kB	NetBeans 6.7.1, 6.8 or 6.9, GWT 1.5, 1.6, 1.7 or 2.0, Java 5 or 6
GWT4NBV2.7.5	Stable <a href="#">kore</a> on Monday, June 7, 2010 at 8:18:14 PM	221.72 kB	NetBeans 6.7.1, 6.8 or 6.9, GWT 1.5, 1.6, 1.7, 2.0 or 2.1, Java 5 or 6
GWT4NBV2.8.0	Stable <a href="#">kore</a> on Monday, June 7, 2010 at 8:18:23 PM	236.97 kB	NetBeans 6.7.1, 6.8 or 6.9, GWT 1.5, 1.6, 1.7, 2.0 or 2.1, Java 5 or 6, next for NetBeans auto update center

Figura 2-13. Descarga del plugin desde el sitio web directamente.

- c. Una vez descargada accedemos a la ventana de Gestión de Plugins de Netbeans y nos dirigimos a la pestaña **Downloaded**.

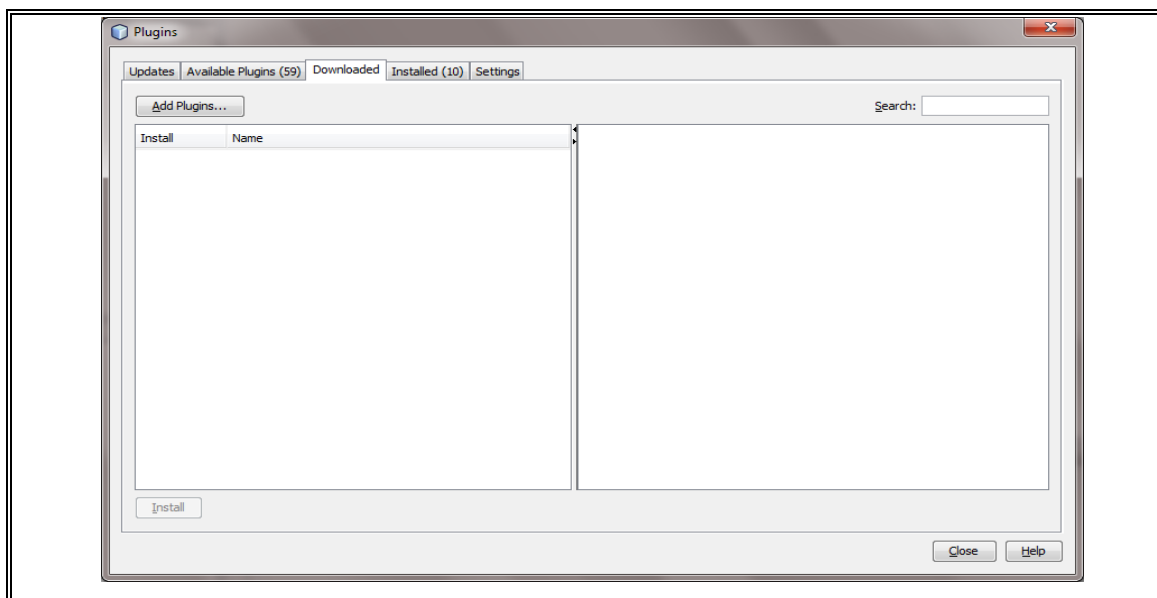


Figura 2-14. Instalación del plugin GWT (2da manera).

Pulsamos el botón **Add Plugins** y nos dirigimos al directorio en donde hayamos descargado el plugin y lo seleccionamos, inmediatamente se carga en la ventana y se pulsa en **Install**.

Una vez descargado el plugin de GWT por cualquiera de las dos maneras procedemos a descargarnos el SDK de GWT, para ellos se sigue los siguientes pasos:

- a. Nos dirigimos a la siguiente página web:  
<http://code.google.com/webtoolkit/download.html>.

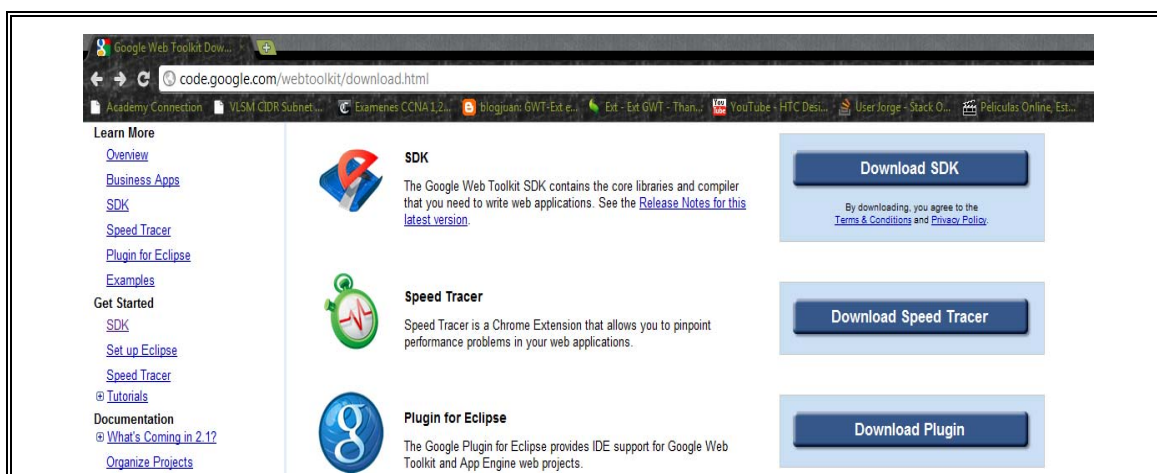


Figura 2-15: Descarga del SDK de GWT (2da manera).

- b. Una vez descargado el SDK, se procede a crear un nuevo proyecto. Se escoge el tipo de proyecto Web Application:

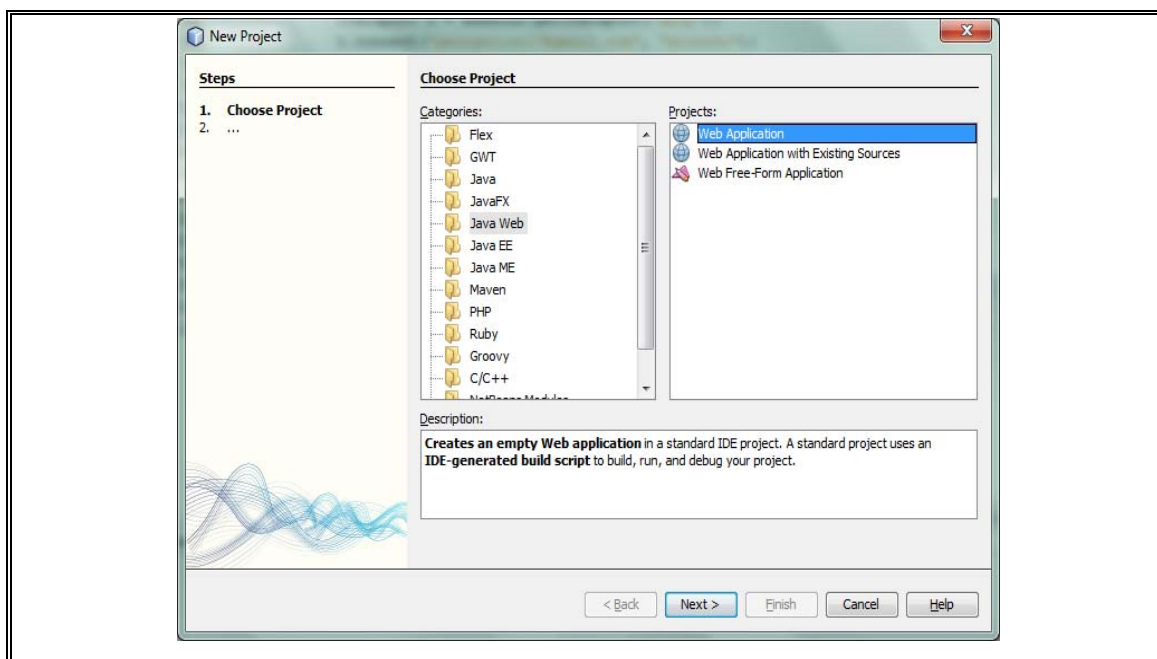


Figura 2-16: Creación de un proyecto GWT (1er Paso)

- c. Se selecciona el nombre del proyecto, el servidor en que se va a ejecutar el sistema, en este caso para el desarrollo de Ge@ se seleccionó el servidor

Glassfish, del cual se hablara en siguientes secciones. Una vez que nos encontramos en el paso 4 aparece el framework de GWT que se ha instalado se seleccione ese:

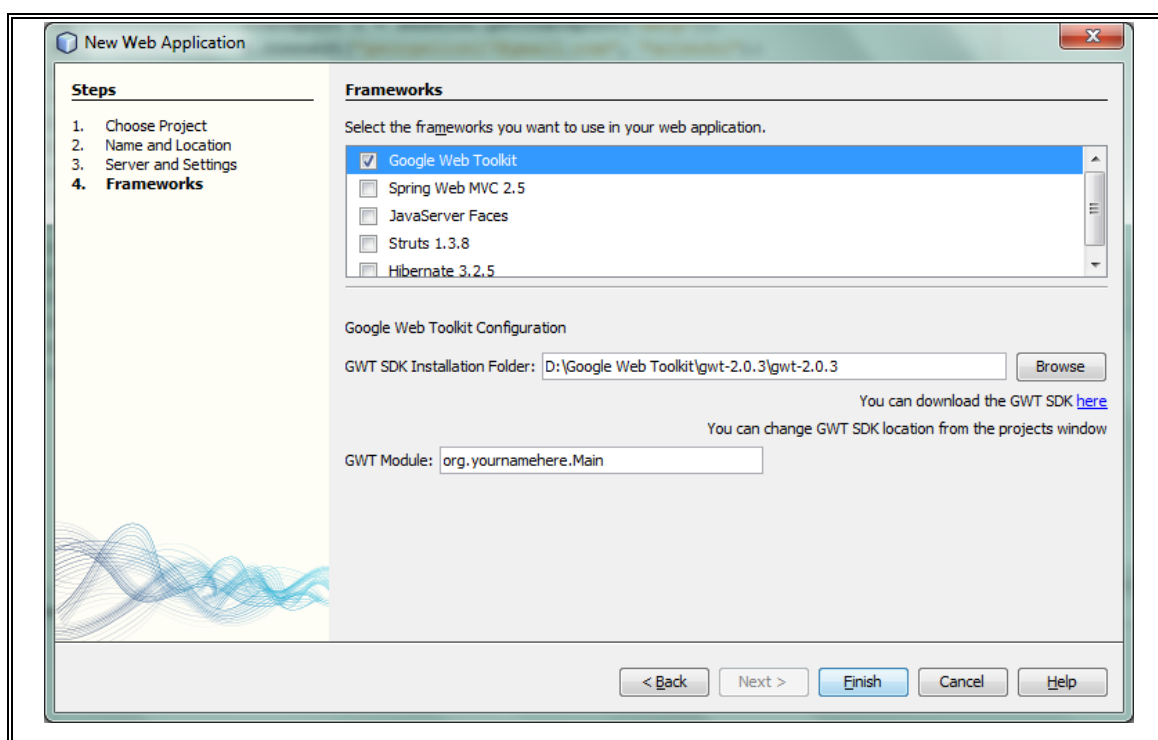


Figura 2-17. Creación de un proyecto GWT (4to paso).

Como se puede observar en la parte inferior de la figura 2.5.14 vemos que es necesario especificar el path en donde se encuentra localizado el GWT SDK que se descargó con anterioridad en el paso a). Luego se procede dar un nombre al módulo GWT. Una vez especificado todos estos parámetros esenciales se procede a pulsar Finalizar (Finish).

Se genera una estructura de ficheros que se pueden visualizar en la siguiente gráfica.

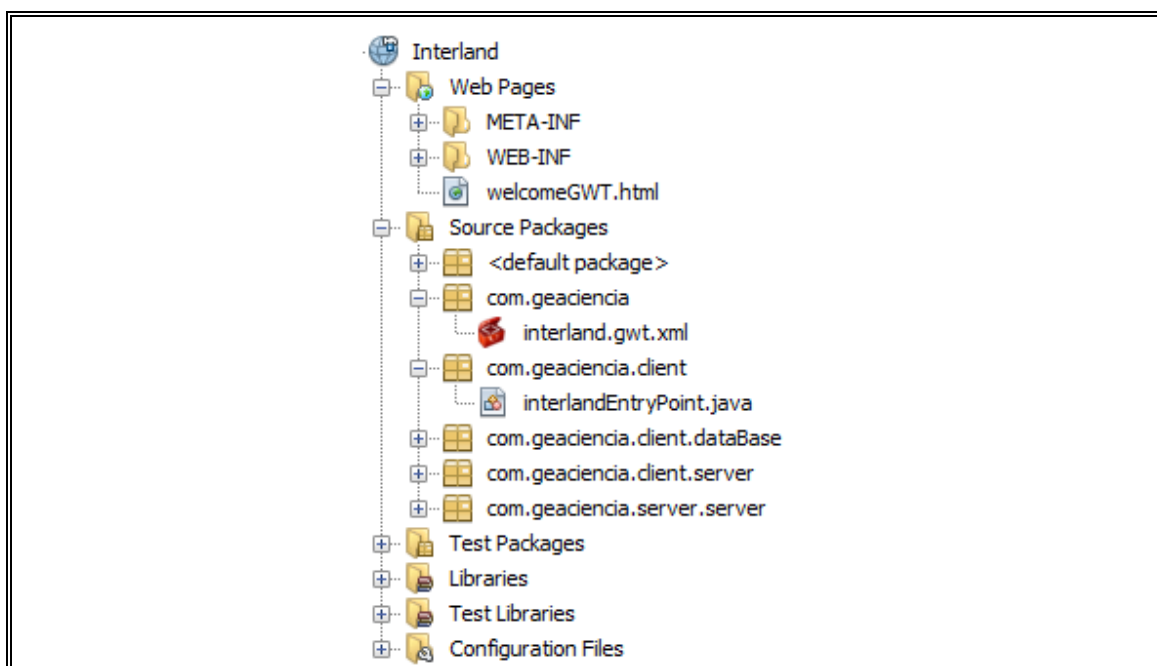


Figura 2-18: Estructura de un proyecto GWT en Netbeans 6.x. }

## 2.6.4 Ext GWT

Ext GWT de Sencha lleva a GWT a un siguiente nivel, proporcionando widgets de alto rendimiento, rico en características y en el diseños de sus plantillas, gráficos avanzados, data loaders, stores, accesibilidad y mucho más.

Es una d las formas más veloces y poderosas de crear aplicaciones Java basadas en web.

### 2.6.4.1 Widgets de alto rendimiento para interfaz de usuario

Ext GWT ofrece una completa biblioteca de widgets de alto rendimiento que son totalmente temáticos y personalizables. Estos incluyen los árboles de datos y grillas combinadas con árboles, listas, formularios, menús, barras de herramientas, paneles y ventanas.

### 2.6.4.2 Templates y Layout Managers

Cuenta con plantillas y manejadores de disposición que ayudan a organizar la visualización de datos y contenidos, Ext GWT incluye plantillas con características

avanzadas tales como matrices de autorrelleno. Además, un administrador de diseño flexible facilita a controlar la visualización de los componentes, incluso para las interfaces de usuario más complejas.

#### **2.6.4.3 Accesibilidad**

Todos los componentes de Ext GWT cumplen con las especificaciones técnicas de la Sección 508 de ARIA (Accessible Rich Internet Applications) e incluyen un tema de alto contraste y soporte de ventanas de alto contraste. Además, la APIFocusManager permite a que la aplicación completa pueda ser navegada a través del teclado.

#### **2.6.4.4 Características Adicionales:**

- Soporte completo de temas con el estándar CSS.
- Bien diseñado, fuente coherente y código plenamente documentado.
- Solución GWT nativa, sin JavaScript externo o bibliotecas de terceros.
- Soporte completo a procedimientos remotos utilizando GWT RPC, JSON y XML.
- Soporte para características Java 1.5
- Licencias de software comercial y de código abierto disponibles.

#### **2.6.4.5 Compatibilidad con Navegadores**

Ext GWT soporta la mayoría de navegadores como:

- Internet Explorer 6+
- FireFox 1.5+ (PC, Mac)
- Safari 3+
- Chrome 3+
- Opera 9+ (PC, Mac)



Figura 2-19: Navegadores compatibles

### 2.6.4.6 Requerimientos Del Sistema

#### Versión de GWT:

Se necesita GWT 1.5 o versiones superiores, las mismas que se pueden descargar en: <http://code.google.com/webtoolkit/download.html>.

#### Versión de Java:

JDK 1.5 o superior, que se pueden encontrar en: <http://java.sun.com/javase/downloads/index.jsp>

### 2.6.4.7 Configuración de Ext Gwt en Netbeans

1. Extraer el EXT GWT SDK descargado, por ejemplo **gxt-2.1.1-gwt2.zip**.
2. Abrir la guía de proyectos GWT en Netbeans.
3. Crear una carpeta con el nombre lib en la guía para crear un proyecto GWT. Para hacer esto, se debe señalar **Archivo | Nuevo Archivo... | Otro** en **Categorías | Carpeta** in **Tipo de Archivos**; pulsar **Next**; colocar el nombre **lib** y elegir **Terminar**.

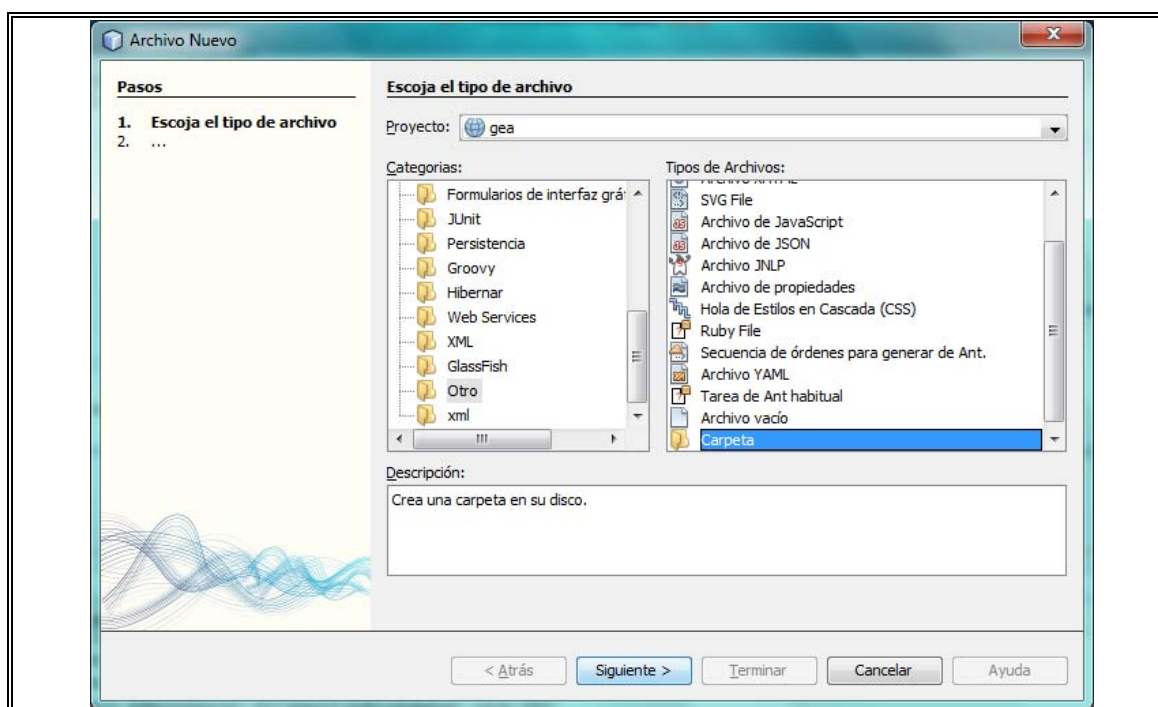


Figura 2-20: Crear carpeta Lib



4. Copiar el archivo **gxt.jar** desde la carpeta gxt-2.1.1 y pegarlo en la carpeta lib creada en el paso anterior.

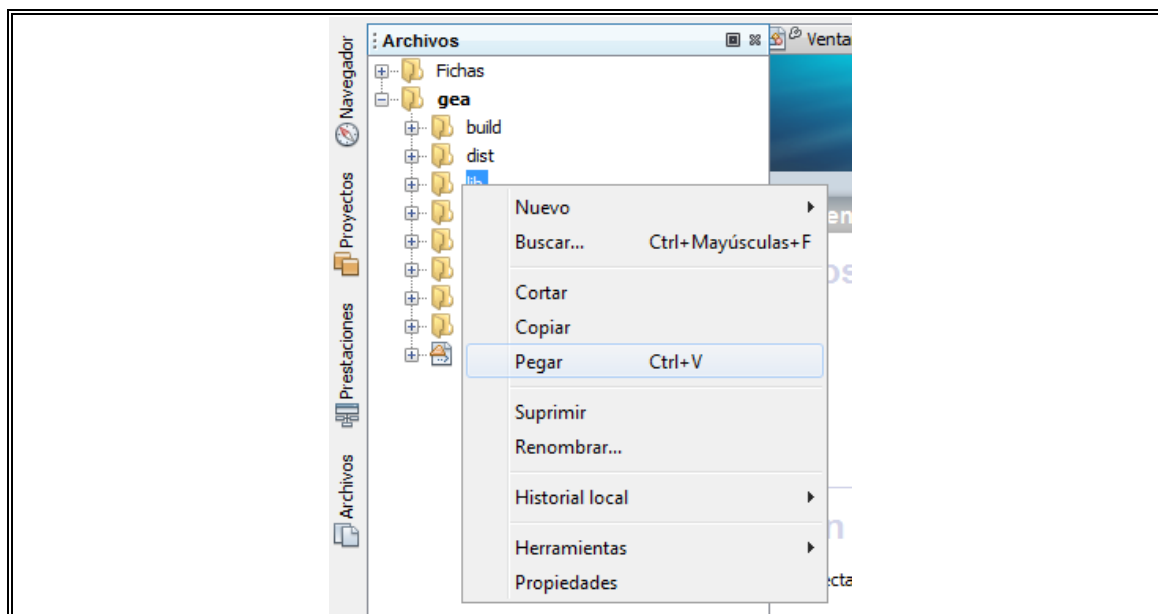


Figura 2-21: Copiar archivo gxt.jar

5. Añadir el archivo gxt.jar en la librería del proyecto

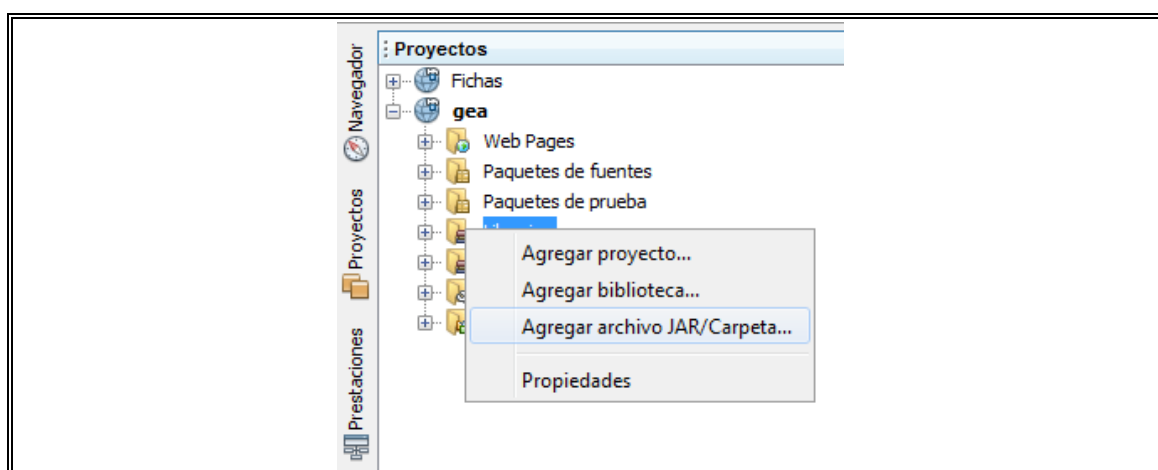


Figura 2-22: Copiar archivo gxt.jar en Librería del Proyecto

6. De la misma manera debemos copiar la carpeta **resources** de la carpeta **gxt-2.2.1** y pegarla en la carpeta **web** del proyecto

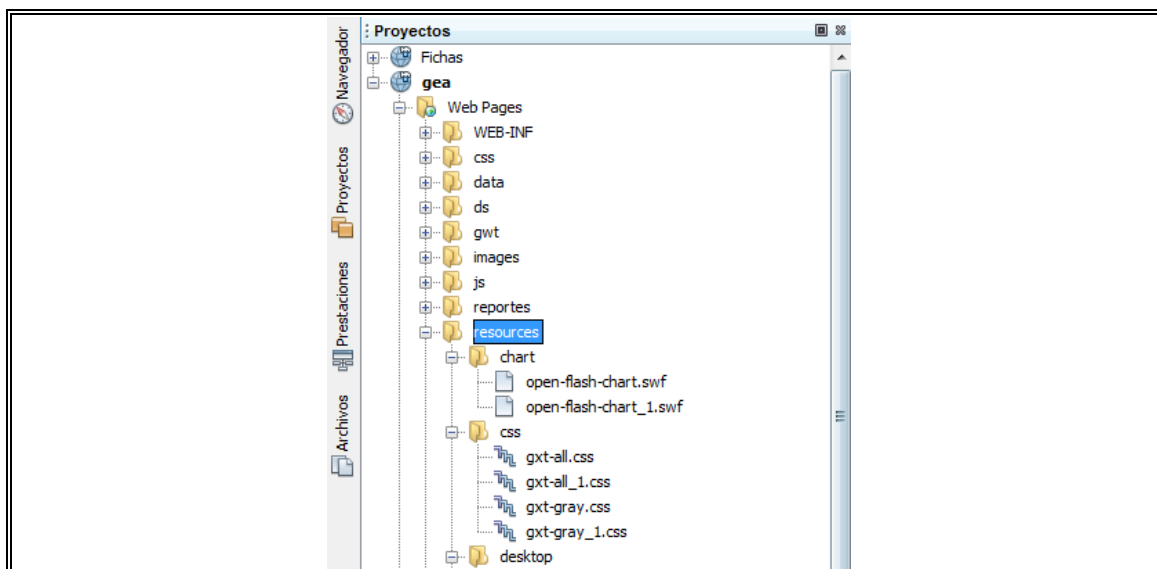


Figura 2-23: Copiar archivo gxt.jar en la Resources

7. Modificar el modulo GWT. Incluir la línea **<inherits name="com.extjs.gxt.ui.GXT"/>** bajo **<module/>** en el archivo **Main.gwt.xml**.

El código completo debe quedar como sigue:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE module PUBLIC "-//Google Inc.//DTD Google Web Toolkit 1.7.0//EN" "http://google-
web-toolkit.googlecode.com/svn/tags/1.7.0/distro-source/core/src/gwt-module.dtd">

<module>
  <inherits name="com.google.gwt.user.User"/>

  <inherits name="com.extjs.gxt.ui.GXT"/>
  <entry-point class="com.blogspot.gwtguide.client.MainEntryPoint"/>
  <!-- modify the entry point class package and name according to your entry point class -->

</module>
```

8. Modificar el archivo **welcomeGWT.html**. Añadir **Add** `<link rel="stylesheet" type="text/css" href="resources/css/gxt-all.css" />` luego de la sección **<head>**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta name='gwt:module' content='com.packtpub.gwt=com.packtpub.gwt'>
    <title>gwt
    <link rel="stylesheet" type="text/css" href="resources/css/gxt-all.css" />
  </head>
  <body>
    <script type="text/javascript" src="com.packtpub.gwt/com.packtpub.gwt.nocache.js">
  </body>
</html>
```

De esta manera estamos listos para trabajar con Ext Gwt.

### 2.6.5 Smart GWT

#### ¿Porque Smart GWT?

Smart GWT ayuda a construir y mantener aplicaciones con una mayor usabilidad, portabilidad, aplicaciones web con una rápida eficiencia, impulsado por una pila abierta de componentes y servicios.

#### Más que solo Widgets: Una arquitectura completa

Smart GWT proporciona una arquitectura de aplicaciones de extremo a extremo, a partir de componentes de interfaz de usuario para el manejo de transacciones en el servidor.

En la página de Isomorphic Software se dispone de un showcase o una aplicación de muestra de todos los componentes por los que está conformado Smart GWT y con ayudas de cómo construir una aplicación. Los ejemplos incluidos con Smart GWT demuestran la simplicidad que sólo puede lograrse mediante un marco que aborda las preocupaciones tanto del servidor y del lado del cliente para ofrecer soluciones arquitectónicas a nivel mundial óptimo.

Los componentes de la interfaz de usuario de Smart GWT están cuidadosamente diseñados para maximizar la capacidad de respuesta y minimizar la carga del servidor, y los componentes del servidor están diseñados alrededor de los requisitos de interfaces de usuario de alta productividad.

Incluso si se adopta sólo una parte de la solución Smart GWT, se beneficia de una arquitectura que tiene en cuenta todo el problema que se desea resolver, no sólo una parte de ella. Cada punto de integración en la plataforma Smart GWT ha sido diseñado con una clara comprensión de los requisitos que se deben cumplir, y las soluciones integradas de Smart GWT proporcionan un modelo que permita el cumplimiento de estos requisitos.

## **Elimina la prueba y depuración de los navegadores**

Smart GWT proporciona un limpio y claro enfoque orientado a objetos para el desarrollo de la interfaz de usuario que protege de los errores del navegador y demás peculiaridades.

Incluso si se desea crear estilos únicos para la interfaz de usuario, Smart GWT simplifica su uso ya que solo se requiere conocer aspectos básicos de diseño sin tener que lidiar con inconsistencias por el navegador. Smart GWT tiene sus componentes orientados a dar a las APIs una mejor flexibilidad y permitir el uso de cualquier método independiente del navegador en que se ejecute, por lo que el desarrollador no tiene que preocuparse de ello. Esto hace que Smart GWT sea una herramienta que garantiza el desarrollo de aplicaciones evitando las inconsistencias que se puedan presentar en los distintos navegadores existentes.

## **Smart GWT como una solución completa**

Smart GWT ofrece un marco de trabajo completo para aplicaciones empresariales, todo lo necesario para la creación de todas las funciones y de aplicaciones robustas y de alta productividad.

Tener marcos de trabajo de distintas fuentes puede ocasionar problemas en el desarrollo de aplicaciones, además de los problemas de interoperabilidad que la mayoría de los navegadores pueda presentar. En definitiva, si se desea construir y mantener un marco de interfaz de usuario para una determinada empresa, Smart GWT puede ser de gran utilidad y brindar un alto nivel competitivo en el desarrollo de aplicaciones rápidas y robustas.

## **Arquitectura abierta y flexible**

Ya que Smart GWT está construido enteramente con tecnologías de web estándar, la integración con el contenido de la web, aplicaciones, portales y portlets resulta sencilla. Se pueden desarrollar aplicaciones con tecnologías de punta de última generación a partir de cero, o se puede actualizar las aplicaciones existentes y portales web a su propio ritmo.

Finalmente, Smart GWT es una herramienta completamente extensible que se guía bajo los estándares de la construcción de sistemas web. Inclusive si el desarrollador no dispone de herramientas dentro del marco de trabajo, puede el mismo desarrollar y construir los componentes que sean necesarios gracias a su arquitectura abierta.

## Arquitectura

La arquitectura de Smart GWT se extiende por el cliente y el servidor, permitiendo que las aplicaciones de Internet sofisticadas (RIA) se comuniquen de manera transparente con sus datos y niveles de servicio.

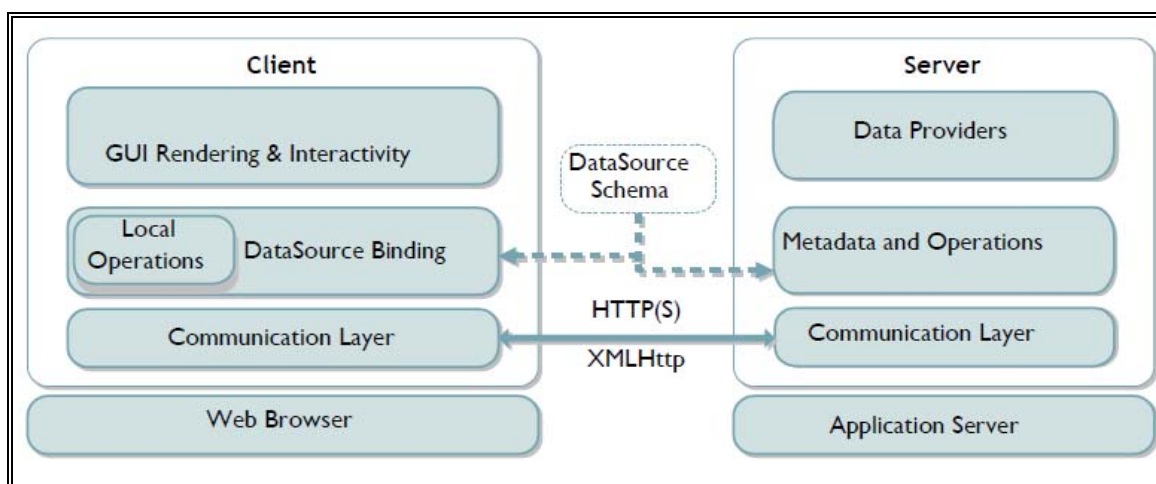


Figura 2-24: Arquitectura de Smart GWT

Dentro del navegador web, Smart GWT proporciona una pila profunda de servicios y componentes para aplicaciones ricas en HTML5 / Ajax. Para aquellos que utilizan un servidor basado en Java, Smart GWT ofrece un marco en el servidor que se pueden añadir a cualquier aplicación web existente de Java.

Los componentes del cliente o servidor tienen un concepto basado en la compartición de fuentes de datos, que describen los objetos de negocio en la aplicación. Al trabajar a partir de una definición única y compartida de modelo de datos, los componentes tanto del servidor como del lado del cliente se coordinan de cerca para ofrecer una funcionalidad muy sofisticada que cualquier otra solución basada en una arquitectura cliente/servidor pueda ofrecer.

Por ejemplo, las reglas de validación son declaradas en una fuente de datos –estas reglas se aplica luego en los componentes AJAX de Smart GWT tanto del lado del cliente como el servidor. De igual manera, el conjunto de operaciones válidas se declara en un origen de datos. Esta simple declaración de controles define el comportamiento en la interfaz del usuario así como la verificación de la seguridad en el cumplimiento de las reglas de negocio en el servidor.

Los DataSources o fuentes de datos se pueden derivar de la ejecución en marcha o como un proceso por lotes, de fuentes de metadatos ya existentes, tales como anotaciones Java Beans y XML Schema, reduciendo aún más la redundancia de todo el sistema.

Este concepto de origen de datos como una responsabilidad compartida entre los componentes del lado del cliente se pueden o no utilizar con los componentes del servidor Smart GWT. Sin embargo, si se opta por no utilizar estos componentes en el lado del servidor, el desarrollador debe mantener y configurar su propia fuente de datos.

Finalmente, cabe señalar que Smart GWT no obliga a que se adopte una arquitectura completa. El desarrollador puede optar por integrarse con las capas y componentes que sean adecuados para los sistemas o aplicaciones que estén desarrollando.

### **Integración de Smart GWT con GWT en NetBeans 6.x**

El proceso de integración del Smart GWT con el framework GWT es sencillo, a continuación se detallan los pasos necesarios:

1. Se procede a descargar la librería de Smart GWT de la siguiente página.  
<http://code.google.com/p/smartgwt/downloads/detail?name=smartgwt-2.4.zip&can=2&q=>

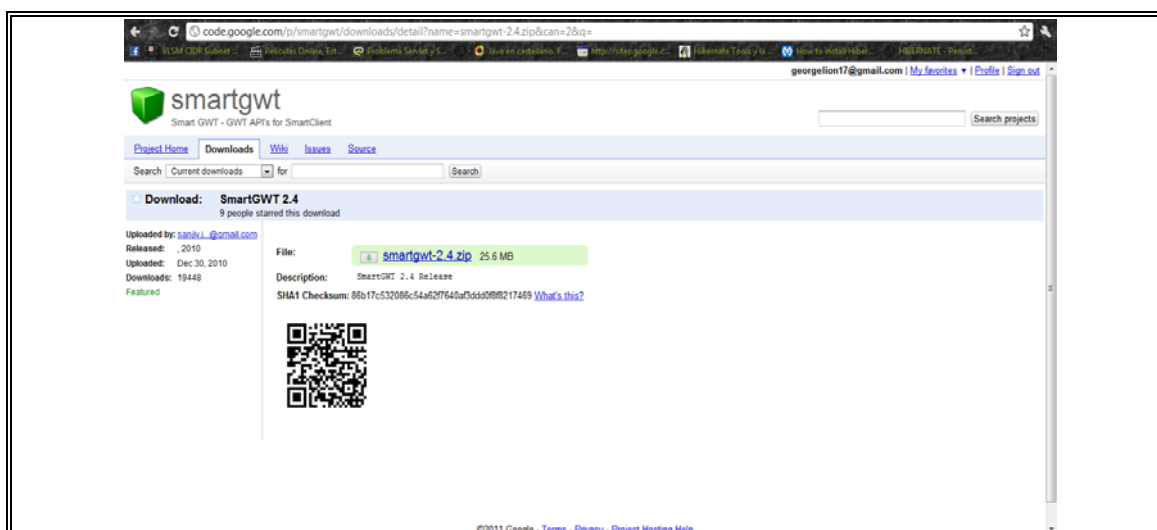


Figura 2-25: Descarga de páginas web.

2. Una vez descargada la librería se procede a añadir al proyecto; se procede a crear un proyecto nuevo del tipo GWT como se especificó en la sección de instalación de GWT.
3. Ya creado el proyecto se va a las propiedades del proyecto con el fin de añadir las librerías descargadas, para ello se da clic derecho sobre el proyecto y se selecciona Propiedades.

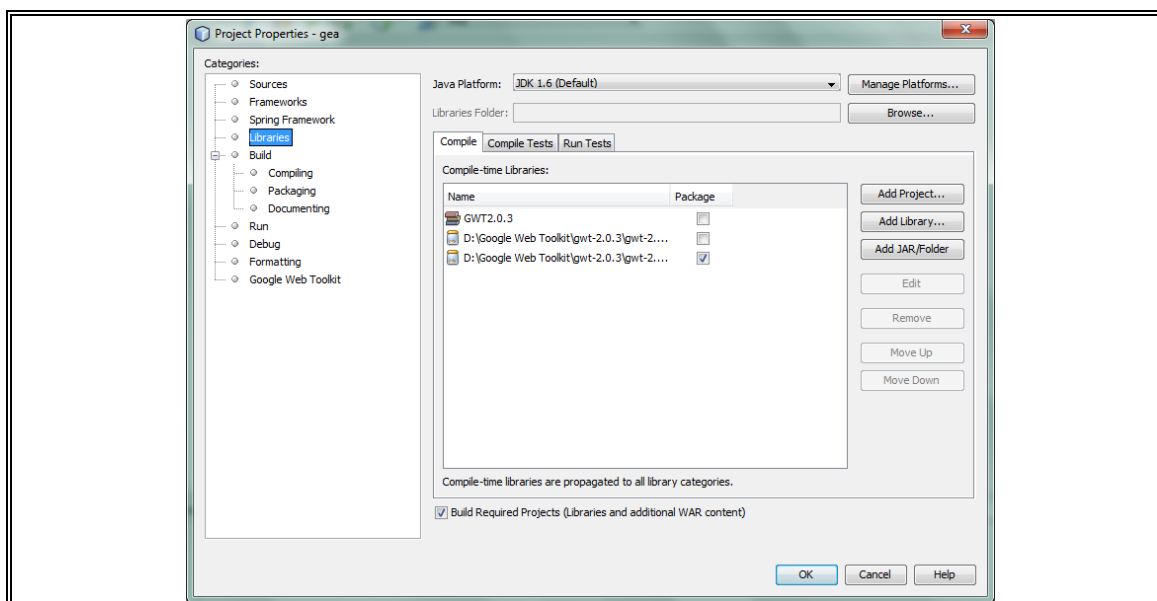


Figura 2-26: Propiedades del Proyecto Ge@



4. En esta ventana se procede a agregar una nueva librería pulsando el botón Add Library o Añadir librería.
5. Se pulsa el botón Create (Crear) y aparecerá la siguiente ventana:

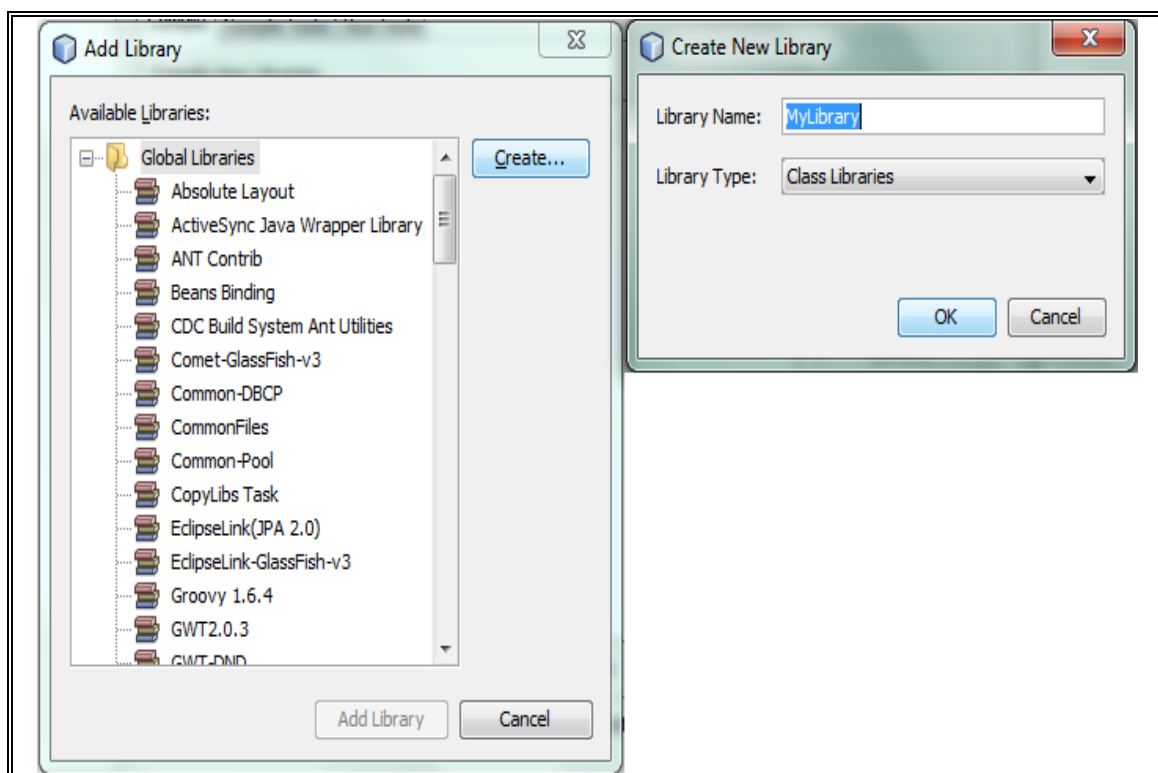


Figura 2-27: Creación de una nueva librería

Se procede a ingresar el nombre de la librería y se escoge como tipo librería de clases. En este caso, se escogió los siguientes valores:

**Library Name:** SmartGWT

**Library Type:** Class Libraries

- Después de pulsar OK, se procede a buscar la ubicación de las librerías descargadas con el fin de añadirlas de manera permanente a NetBeans y por ende al proyecto a desarrollarse.

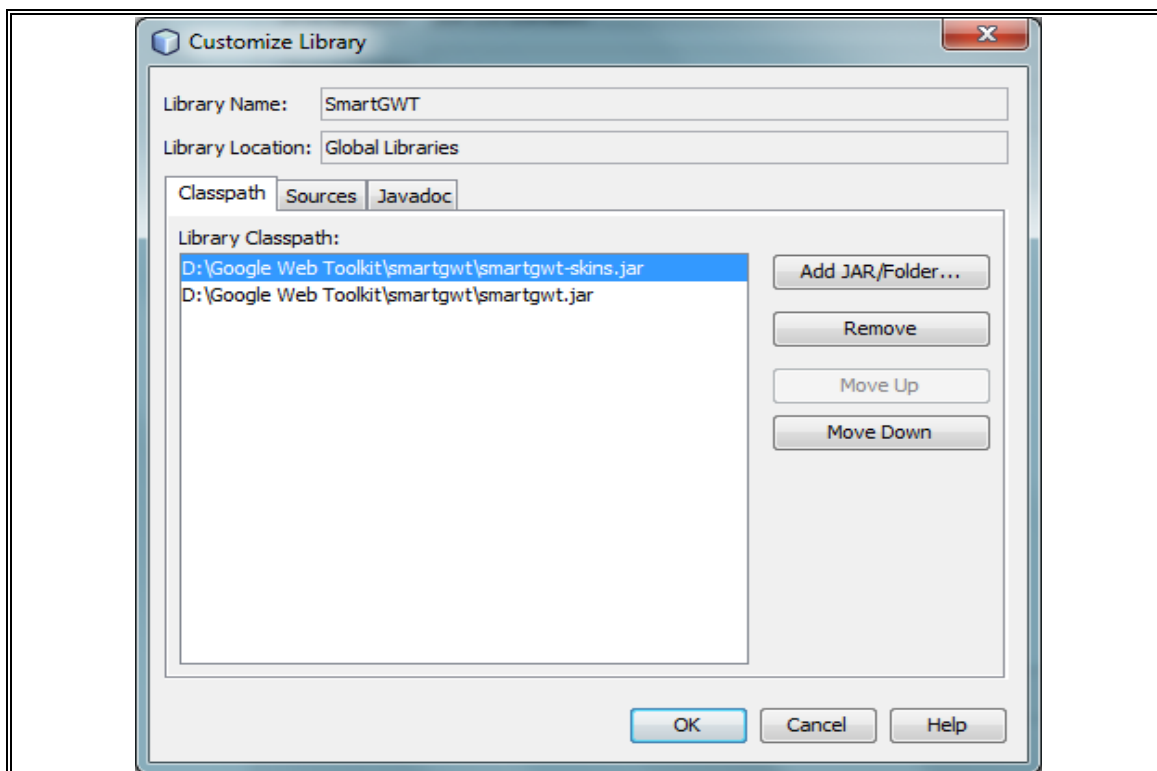


Figura 2-28: Creación de la librería Smart GWT.

Se pulsa ADD JAR/FOLDER y se busca las librerías Smart GWT en la ubicación donde se las haya descargado. Una vez agregados los archivos JAR se pulsa OK.

7. Una vez creada la librería y añadida al proyecto se tendrá la ventana siguiente:

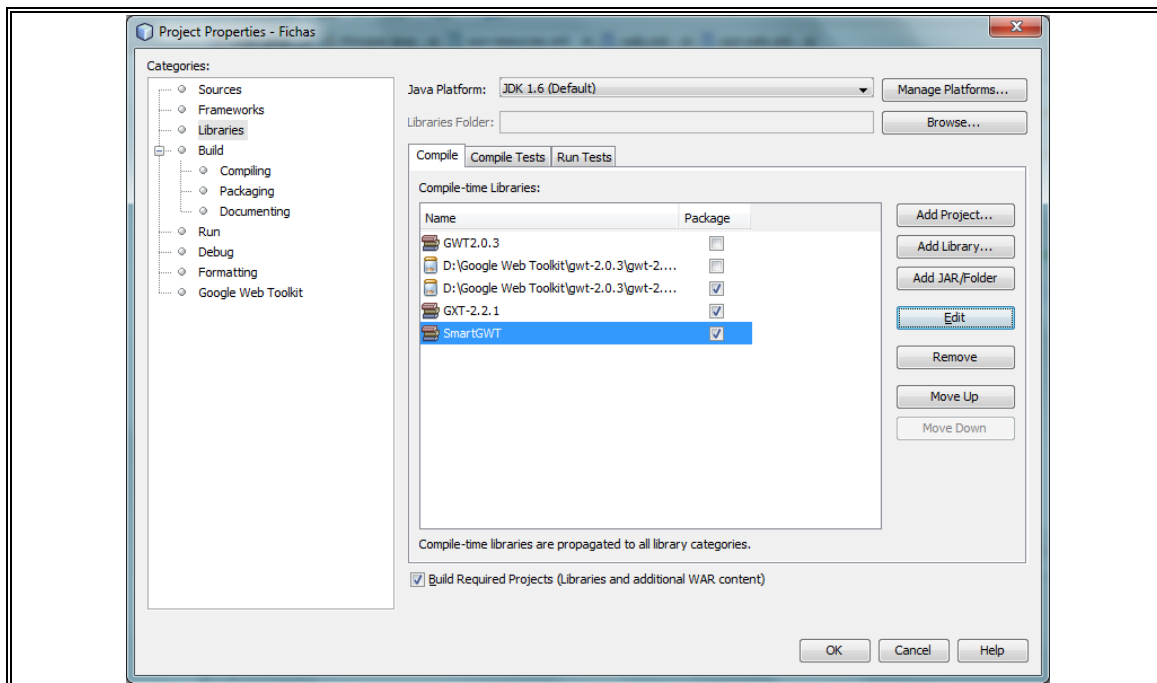


Figura 2-29: Creación de la librería Smart GWT (Propiedades de Proyecto)

8. Una vez añadida la librería al proyecto, se procede a ingresar los archivos CSS y las imágenes necesarias, con el fin de tomar los estilos y formas de Smart GWT. Para ello, cuando se descargó la librería de Smart GWT vino junto con la descarga un proyecto de muestra, el path específico en donde se encuentra es **smartgwt\samples>Showcase\war**, dentro de este directorio se encuentran tres carpetas que hay que añadir al proyecto, que son: CSS, DATA, DS, GWT, IMAGES, JS, SHOWCASE, SOURCE, SOURCEGEN.

Ahora el proceso de copia se puede hacer de dos maneras, directamente dentro de NetBeans en el árbol navegador de proyectos o por el explorador de Windows. Tomando la primera opción, nos dirigimos al panel de navegador de NetBeans y expandimos el árbol la rama Web Pages, damos clic derecho y pulsamos Pegar.

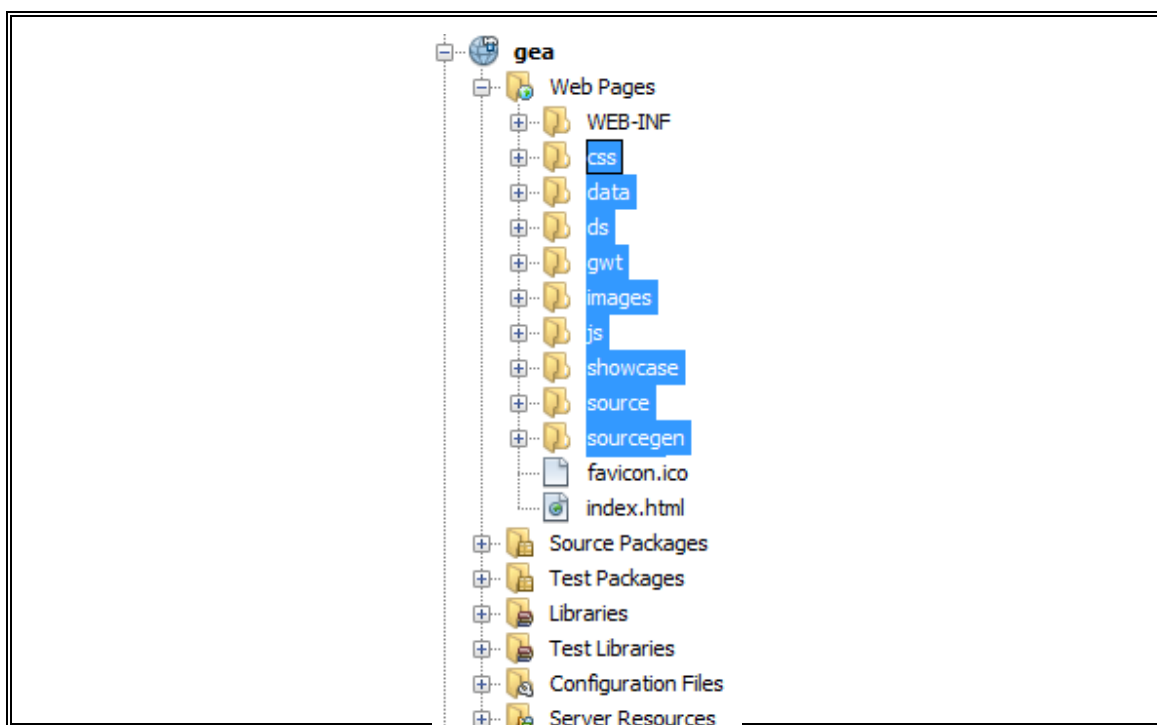


Figura 2-30: Se añade los estilos, formas e imágenes necesarias para Smart GWT.

9. Una vez añadido tanto las carpetas y la librerías de Smart GWT, ahora lo que queda por realizar es hacerlos referencia desde el proyecto, para ello de la imagen anterior se procede a abrir el archivo index.html, este archivo es la página inicial de nuestro sistema Ge@, es aquí donde se va a cargar todo el código JavaScript compilado y generado por GWT, en caso que se utilice otras librerías como en este caso Smart GWT o Ext GWT, se debe añadir una referencia a ellos por medio de esta página HTML. Las líneas que se deben de agregar se pueden observar en la imagen inferior.

```

76 <!--includiendo el SC Core API-->
77 <script src="showcase/sc/modules/ISC_Core.js"></script>
78
79 <!--includiendo el SmartClient -->
80 <script type="text/javascript">document.getElementById('loadingMsg').innerHTML = 'Cargando Componentes UI...';</script>
81 <script src='showcase/sc/modules/ISC_Foundation.js'></script>
82 <script src='showcase/sc/modules/ISC_Containers.js'></script>
83 <script src='showcase/sc/modules/ISC_Grids.js'></script>
84 <script src='showcase/sc/modules/ISC_Forms.js'></script>
85 <script src='showcase/sc/modules/ISC_RichTextEditor.js'></script>
86 <script src='showcase/sc/modules/ISC_Calendar.js'></script>
87
88 <script type="text/javascript">document.getElementById('loadingMsg').innerHTML = 'Cargando Data API...';</script>
89 <script src='showcase/sc/modules/ISC_DataBinding.js'></script>
90

```

Figura 2-31: Inclusión de Core API y SmartClient de Smart GWT

```

107 | <!--Cargando Tema-->
108 | <script type="text/javascript">document.getElementById('loadingMsg').innerHTML = 'Cargando tema...';</script>
109 | <script type="text/javascript">
110 | document.write("<"+<script src=showcase/sc/skins/Enterprise/load_skin.js?isc_version=7.1.js><"+</script>");
111 | </script>
...

```

Figura 2-32: Inclusión del tema Enterprise de Smart GWT.

Por último, hay que agregar una entrada de la librería Smart GWT al proyecto para ello se procede a abrir el archivo XML que se generó al crear el proyecto GWT. Y se añade lo siguiente:

```

<inherits name="com.smartgwt.SmartGwtNoScript"/>
<inherits name="com.smartgwt.tools.SmartGwtTools"/>
<inherits name="com.smartclient.theme.enterprise.EnterpriseResources"/>
<inherits name="com.smartclient.theme.enterpriseblue.EnterpriseBlueResources"/>
<inherits name="com.smartclient.theme.graphite.GraphiteResources"/>

```

Figura 2-33: Inclusión de Smart GWT al proyecto

## 2.6.6 Servidor de Aplicaciones

Un servidor de aplicaciones es un software que proporciona aplicaciones a los equipos o dispositivos cliente, por lo general a través de Internet y utilizando el protocolo http. Es un servidor en una red de computadoras que ejecuta ciertas aplicaciones. Los servidores de aplicación se distinguen de los servidores web por el uso extensivo del contenido dinámico y por su frecuente integración con bases de datos.

Además, un **servidor de aplicaciones** es un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente, es un producto basado en un componente que se encuentra en el plano medio de la arquitectura central de un servidor. Proporciona servicios de 'middleware', es decir, trabaja como un intermediario para la seguridad y el mantenimiento, además de proveer acceso a los datos.

Un servidor de aplicación maneja la mayoría de las transacciones relacionadas con la lógica y el acceso a los datos de la aplicación (esto se solía llamar 'centralización', hace algún tiempo). La ventaja principal de un servidor de aplicaciones es la facilidad para desarrollarlas, puesto que éstas no necesitan ser programadas y en cambio, se arman a

partir de módulos provistos por el servidor de aplicaciones. Por ejemplo, un wiki es un servidor de aplicación que permite a los usuarios crear contenido dinámico a partir del ensamble de los artículos.

**El término servidor de aplicaciones se aplica a todas las plataformas**, y hay muchas variaciones sobre el tema, por lo que resulta un poco ambiguo. El término se utiliza para referirse a los servidores de aplicaciones basadas en Web, como el control de las plataformas de comercio electrónico integrado, sistemas de gestión de contenido de sitios Web y asistentes o constructores de sitios de Internet. Por esta razón, algunos los llaman también ‘servidor web’.

Uno de los ejemplos destacados es el de Sun Microsystems, plataforma J2EE. Los servidores de aplicaciones Java se basan en la Plataforma Java <sup>TM</sup> 2, Enterprise Edition (J2EE <sup>TM</sup>), del cual se hablara con más detalle en la siguiente sección puesto que es el servidor con el que opera Ge@.

### **Beneficios de un Servidor de Aplicaciones**

- Agrupación de recursos (por ejemplo, agrupación de conexiones de base de datos y agrupación de objetos)
- Administración de transacciones distribuida
- Comunicación asincrónica de programa, normalmente a través de colas de mensajes
- Un modelo de activación de objetos oportuno
- Interfaces de servicios Web XML automáticas para tener acceso a objetos de empresa
- Servicios de detección de errores y estado de las aplicaciones
- Seguridad integrada
- Centralización y disminución de la complejidad en el desarrollo de aplicaciones.

### **Ventajas de los servidores de aplicaciones**

- *Integridad de datos y códigos:* al estar centralizada en una o un pequeño número de máquinas servidoras, las actualizaciones están garantizadas para todos sus usuarios. No hay riesgos de versiones viejas.
- *Configuración centralizada:* los cambios en la configuración de la aplicación, como mover el servidor de base de datos o la configuración del sistema, pueden ser hechos centralmente.
- *Seguridad:* se consideran más seguras.
- *Performance:* limitando el tráfico de la red solamente al tráfico de la capa de presentación, es percibido como un modelo cliente/servidor que mejora la performance de grandes aplicaciones.

### 2.6.6.1 GlassFish Server Application

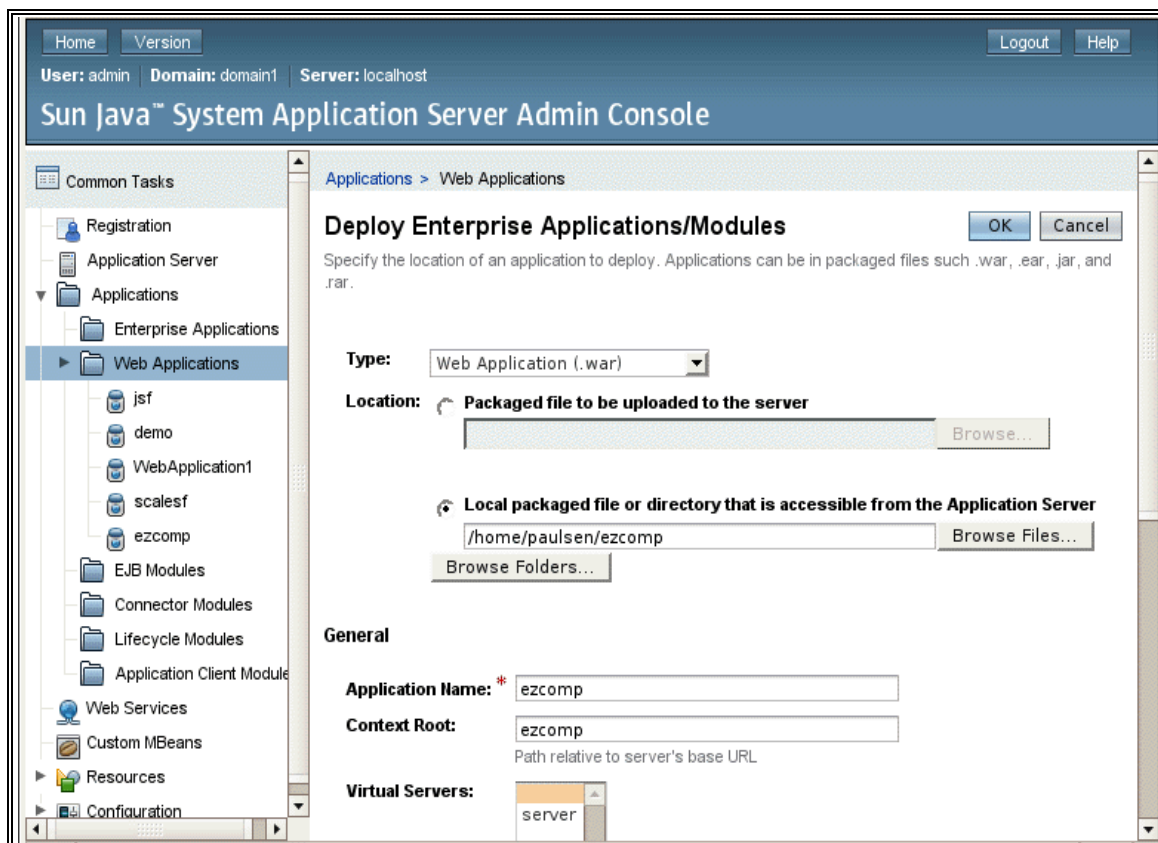


Figura 2-34. Consola de Administración de GlassFish

GlassFish es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, compañía adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. La versión comercial es denominada Oracle GlassFish Enterprise Server (antes Sun GlassFish Enterprise Server). Es gratuito y de código libre, se distribuye bajo un licenciamiento dual a través de la licencia CDDL y la GNU GPL.

GlassFish está basado en el código fuente donado por Sun y Oracle Corporation, éste último proporcionó el módulo de persistencia TopLink. GlassFish tiene como base al servidor Sun Java System Application Server de Oracle Corporation, un derivado de Apache Tomcat, y que usa un componente adicional llamado Grizzly que usa Java NIO para escalabilidad y velocidad.



El término Glassfish, traducido al español sería algo parecido como “Pez de Cristal”, es el nombre de un pez que realmente existe y vive en el agua dulce; su cuerpo es transparente, por lo que sus huesos son visibles. El hecho de este nombre se debe a la transparencia que brinda GlassFish a todos los usuarios, a su arquitectura completa.

Glassfish es un servidor de aplicaciones que implementa la plataforma JavaEE6, por lo que soporta las últimas versiones de tecnologías como: JSP, JSF, Servlets, EJBs, Java API para Servicios Web (JAX-WS), Arquitectura Java para Enlaces XML (JAXB), Metadatos de Servicios Web para la Plataforma Java, y muchas otras tecnologías.

Glassfish además de ser un servidor de aplicaciones, es una comunidad de usuarios, que descargan y utilizan libremente Glassfish, también existen partners que contribuyen agregándole más características importantes a Glassfish. Además ingenieros y beta testers que desarrollan código y prueban las versiones liberadas para eliminar todo fallo que se encuentre, y muchos otros miembros.

La comunidad fue lanzada en el año 2005 en java.net. Al igual que el pez original, la Comunidad Glassfish es transparente en cuanto a términos de entrega de código fuente, discusiones de ingeniería, agendas, datos de descarga, etc. Se puede tener acceso a todo esto, además se puede formar parte de todo el proceso detrás de la comunidad Glassfish.

Ya mencionamos que Glassfish implementa la Plataforma JavaEE6, y que es totalmente compatible con ésta, pero... ¿Qué es JavaEE? Resumiendo, Java Enterprise Edition (JEE) es esencialmente una forma estándar de desarrollar Aplicaciones Java Empresariales que sean portables, esto quiere decir, que puedan ser utilizadas en más de un servidor sin importar el fabricante, sin necesidad de hacerles cambio alguno. Existe una versión de Glassfish que incluye soporte comercial por parte de Sun Microsystems, ésta versión está enfocada a empresas, universidades, etc.

## Características de GlassFish

### **Diseño Modular**

Todo el código de GlassFish fue separado en módulos para brindar mayor flexibilidad y rendimiento. Esta arquitectura modular está implementada sobre el estándar OSGi, lo que permite reusar módulos del mismo contenedor o de terceros.

Este cambio en el diseño permite descargar e instalar solamente los módulos que requieran las aplicaciones que se vayan a desplegar. Las actualizaciones pueden llevarse a cabo sin reinstalar todo el sistema. Este cambio permitió disminuir los tiempos de arranque, el consumo de memoria y espacio en disco.

### **Herramientas de actualización integrada**

La herramienta de actualización está integrada a la consola de administración de GlassFish. Esta herramienta brinda actualizaciones automáticas, y facilita la gestión de los componentes y aplicaciones relacionadas para extender las capacidades del contenedor.

### **Lenguajes de Scripting**

GlassFish permite utilizar varios lenguajes de scripting, pudiendo así desplegar aplicaciones más allá de la tecnología Java. Entre los lenguajes de scripting soportados:

- Ruby y Rails: un lenguaje de scripting y un framework para desarrollar aplicaciones web.
- Grails: un framework de aplicaciones web basado en Groovy y que complementa el desarrollo web Java.
- Phobos: un framework liviano de aplicaciones web que permite usar JavaScript, de manera que los desarrolladores pueden usar el mismo lenguaje en el cliente y en el servidor.
- jMaki: un framework para crear aplicaciones web Ajax.

El soporte para estos lenguajes se provee a través de componentes disponibles en la herramienta de actualización.

### **Extensiones Web**

GlassFish brinda soporte para la capa web de la especificación Java EE, implementado los siguientes estándares:

- Java Persistence API 1.0
- Java Servlet 2.5
- JavaServer™ Faces 1.2
- Java Server Pages™ 2.1 (JSP™ 2.1)
- Java Server Pages Standard Tag Library (JSTL) 1.
- Streaming API for XML (StAX)

Además, GlassFish trae varias extensiones, entre las cuales se encuentra Comet, REST, Ajax, Metro, Roller y Grizzly.

¿Por qué optar por GlassFish?

En el mundo del desarrollo de aplicaciones web en Java, existen muchas herramientas que permiten crear paginas utilizando JSP (como el popular Tomcat de Apache), pero algunas veces necesitamos ir más a fondo en la construcción de una aplicación web y utilizar una mejor versión como la empresarial de Java (J2EE) para la construcción de aplicaciones que respondan a una arquitectura más especializada; y se puede notar que un simple servidor web queda corto para las funcionalidades que ofrece GlassFish, si no que necesitamos un Servidor de Aplicaciones J2EE, que soporte los componentes para la creación de aplicaciones web empresariales utilizando los estándares sugeridos dentro del J2EE.

De estos Servidores de Aplicaciones para J2EE, existen de todos los tamaños, sabores y colores; pero el soportado oficialmente por Sun Microsystem es el Glassfish el cual viene en una versión comercial llamada el Sun GlassFish Enterprise Server y en una versión de código abierto llamada simplemente Glassfish, la cual posee dos licencias la CDDL y el GPL.

Así que por tal razón se optó por Glassfish al ser un servidor de aplicaciones para Java en su versión J2EE, que permite la creación de aplicaciones empresariales con componentes web, transaccionales y de persistencia.

#### Historia de GlassFish

- 6 de junio de 2005 - Primer lanzamiento del proyecto.
- 4 de mayo de 2006 - Primera versión que soporta la especificación Java EE 5.
- 8 de mayo de 2007 - el proyecto Project SailFin se anuncia en JavaOne como un subproyecto bajo GlassFish. SailFin es un proyecto que añade nuevas funcionalidades, como el servlet de Session Initiation Protocol (SIP).
- 17 de septiembre de 2007 - Aparece la versión 2.1 (también conocida como Sun Java System Application Server 9.1) con capacidades de clúster y nuevas características de interconexión entre servicios web.
- 21 de enero de 2009 - Sun Microsystems y la comunidad lanza GlassFish 2.1 (Sun GlassFish Enterprise Server 2.1), el que sirve como la base para el proyecto SailFin SIP AppServer Project (también conocido como Sun Communication Application Server).
- 10 de diciembre 2009 - Se libera la versión 3 de GlassFish. Tomando como origen el hecho de ser la implementación Java EE de referencia, éste es el primer servidor de aplicaciones que implementa completamente Java EE 6 JSR 316. Sin embargo, JSR 316 ha sido aprobado con ciertas reservas. En esta versión GlassFish añade nuevas características para migrar fácilmente desde Tomcat a GlassFish. Otras nuevas características importantes son la modularidad, tiempo mejorado de inicio, despliegues desde plugins en NetBeans y Eclipse y preservación de sesiones a través de redespliegues.

El 25 de marzo de 2010, poco después de la adquisición de Sun Microsystems, Oracle emitió una hoja de ruta para las versiones 3.0.1, 3.1, 3.2 y 4.0 que versarán en temas como clustering, virtualización e integración con coherencia y otras tecnologías de Oracle. La comunidad "open source" permanece de todas formas sin cambios.

## Descarga de GlassFish V3.1

El proceso de descarga de GlassFish para Linux resulta sencillo, se debe de descargar un archivo de formato SH, y el proceso de instalación es similar a la instalación de un programa en Windows, por medio de un wizard.

1. Se ingresa a la siguiente dirección, <http://glassfish.java.net/downloads/3.1-final.html>

Welcome to GlassFish Server Open Source Edition 3.1, the industry's first application server to support the Java Platform, Enterprise Edition 6 (Java EE 6) standard with production-ready features such as clustering and centralized administration. Its overall flexibility and ease of use reduce cost by providing improved developer productivity, simplified application architecture, and dynamic upgrade support.

**What is GlassFish Server 3.1?**  
GlassFish Server 3.1 is the successor of the earlier 3.0.x releases featuring a modular, flexible runtime based on the OSGi standard now with **fully-featured clustering** with centralized administration of multiple clusters and high availability of stateful components. Developers also benefit from the simplified programming model and productivity improvements offered by Java EE 6. Because GlassFish Server 3.1 uses a microkernel architecture based on OSGi, developers can begin with the Java EE 6 Web Profile and use the Update Center to easily upgrade to the full Java EE 6 platform!  
GlassFish Server 3.1 is also the fastest open source application server offering advanced features such as application versioning, application-scoped resources, and great development tool support from NetBeans 7.0, Eclipse and other popular IDEs.

After installing and using GlassFish 3.1, let us know what you think and, of course, feel free to [get involved!](#)

**What is the License?**  
GlassFish Community Distributions are available under a Dual License consisting of the [Common Development and Distribution License \(CDDL\) v1.0](#) and [GNU General Public License \(GPL\) v2](#). See the [third-party readme](#) and [webprofile-third-party-readme](#) for external components included in GlassFish Server 3.1 and their associated licenses.

**How do I get GlassFish Server 3.1?**  
**GlassFish Server Open Source Edition 3.1**

Distribution	Windows [1]	Size (MB)	Linux / Unix / Mac [2]	Size (MB)	Zip archive [3]	Size (MB)
GlassFish Server 3.1 Open Source Edition Full Platform	<a href="#">glassfish-3.1-windows.exe</a> (EN)	52	<a href="#">glassfish-3.1-unix.sh</a> (EN)	52	<a href="#">glassfish-3.1.zip</a> (EN)	78
	<a href="#">glassfish-3.1-windows-ml.exe</a> (multilingual)	61	<a href="#">glassfish-3.1-unix-ml.sh</a> (multilingual)	61	<a href="#">glassfish-3.1-ml.zip</a> (multilingual)	93
GlassFish Server 3.1 Open Source Edition Web Profile	<a href="#">glassfish-3.1-web-windows.exe</a> (EN)	32	<a href="#">glassfish-3.1-web-unix.sh</a> (EN)	32	<a href="#">glassfish-3.1-web.zip</a> (EN)	44
	<a href="#">glassfish-3.1-web-windows-ml.exe</a> (multilingual)	38	<a href="#">glassfish-3.1-web-unix-ml.sh</a> (multilingual)	35	<a href="#">glassfish-3.1-web-ml.zip</a> (multilingual)	38

Figura 2-35: Descarga de GlassFish V3.1

Se selecciona el enlace que nos especifique una versión de GlassFish para Linux y que sea Multilenguaje.

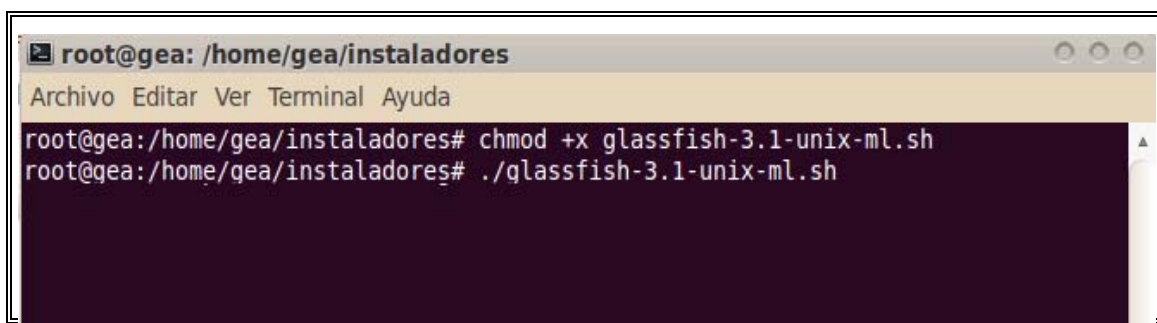
La versión que se toma es la siguiente:

**Distribución:** GlassFish Server 3.1 Open Source Edition Full Platform.

**Linux/Unix/Mac:** glassfish-3.1-unix-ml.sh (multilingual).

## Instalación de GlassFish V3.1

Con el archivo descargado de GlassFish procedemos a guardarlo en una ubicación que nos sea conocida ya que por medio de la terminal se le va a proceder a cambiar los permisos, esto con el objetivo de hacerle al archivo de tipo ejecutable.

A screenshot of a terminal window titled "root@gea: /home/gea/instaladores". The window has a menu bar with "Archivo", "Editar", "Ver", "Terminal", and "Ayuda". The terminal shows two lines of command execution: "root@gea:/home/gea/instaladores# chmod +x glassfish-3.1-unix-ml.sh" and "root@gea:/home/gea/instaladores# ./glassfish-3.1-unix-ml.sh". The background of the terminal is dark purple.

```
root@gea: /home/gea/instaladores
Archivo Editar Ver Terminal Ayuda
root@gea:/home/gea/instaladores# chmod +x glassfish-3.1-unix-ml.sh
root@gea:/home/gea/instaladores# ./glassfish-3.1-unix-ml.sh
```

Figura 2-36. Cambio de permisos al archivo SH

Una vez cambiado los permisos se procede a ejecutar el archivo, eso es lo que hace la segunda línea, luego de ejecutar dicha sentencia, inmediatamente se carga el wizard de GlassFish.

## 2.6.7 MySql

### Panorámica del sistema de gestión de base de datos MySQL

MySQL, el sistema de gestión de bases de datos SQL Open Source más popular, lo desarrolla, distribuye y soporta MySQL AB. MySQL AB es una compañía comercial, fundada por los desarrolladores de MySQL. Es una compañía Open Source de segunda generación que une los valores y metodología Open Source con un exitoso modelo de negocio.

### MySQL es un sistema de gestión de bases de datos relacionales

Una base de datos relacional almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad. La parte SQL de "MySQL" se refiere a "Structured Query Language". SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL. El estándar SQL ha evolucionado desde 1986 y existen varias versiones. En este manual, "SQL-92" se refiere al estándar del 1992, "SQL:1999" se refiere a la versión del 1999, y "SQL:2003" se refiere a la versión actual del estándar. Usamos la frase "el estándar SQL" para referirnos a la versión actual de SQL.

**MySQL software es Open Source.**

Open Source significa que es posible para cualquiera usar y modificar el software. Cualquiera puede bajar el software MySQL desde internet y usarlo sin pagar nada. Si lo desea, puede estudiar el código fuente y cambiarlo para adaptarlo a sus necesidades. El software MySQL usa la licencia GPL (GNU General Public License), <http://www.fsf.org/licenses/>, para definir lo que puede y no puede hacer con el software en diferentes situaciones. Si no se encuentra cómodo con la GPL o necesita añadir código MySQL en una aplicación comercial, puede comprarnos una licencia comercial. Consulte la Introducción a las Licencias MySQL para más información (<http://www.mysql.com/company/legal/licensing/>).

**El servidor de base de datos MySQL es muy rápido, fiable y fácil de usar.**

Si esto es lo que está buscando, debería probarlo. El servidor MySQL también tiene una serie de características prácticas desarrolladas en cooperación con los usuarios. Puede encontrar comparaciones de rendimiento de MySQL Server con otros sistemas de gestión de bases de datos en nuestra página de comparativas de rendimiento.

MySQL Server se desarrolló originalmente para tratar grandes bases de datos mucho más rápido que soluciones existentes y ha sido usado con éxito en entornos de producción de alto rendimiento durante varios años. MySQL Server ofrece hoy en día una gran cantidad de funciones. Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet

**MySQL Server trabaja en entornos cliente/servidor o incrustados**

El software de bases de datos MySQL es un sistema cliente/servidor que consiste en un servidor SQL multi-threaded que trabaja con diferentes backends, programas y bibliotecas cliente, herramientas administrativas y un amplio abanico de interfaces de programación para aplicaciones (APIs).

También proporcionamos el MySQL Server como biblioteca incrustada multi-threaded que puede linkar en su aplicación para obtener un producto más pequeño, rápido y fácil de administrar.

## **Una gran cantidad de software de contribuciones está disponible para MySQL**

Es muy posible que su aplicación o lenguaje favorito soporte el servidor de base de datos MySQL.

La forma oficial de pronunciar "MySQL" es "My Ess Que Ell" (no "my sicuel"), pero no importa si lo pronuncia como "my sicuel" o de alguna otra forma.

## **Las principales características de MySQL**

La siguiente lista describe algunas de las características más importantes del software de base de datos MySQL.

### Interioridades y portabilidad

- Escrito en C y en C++
- Probado con un amplio rango de compiladores diferentes
- Funciona en diferentes plataformas.
- Usa GNU Automake, Autoconf, y Libtool para portabilidad.
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Proporciona sistemas de almacenamiento transaccionales y no transaccionales.
- Un sistema de reserva de memoria muy rápido basado en threads.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Tablas hash en memoria, que son usadas como tablas temporales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (linkado) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.



- Registros de longitud fija y longitud variable.

#### Sentencias y funciones

- Soporte completo para las cláusulas SQL `GROUP BY` y `ORDER BY`. Soporte de funciones de agrupación (`COUNT()`, `COUNT(DISTINCT ...)`, `AVG()`, `STD()`, `SUM()`, `MAX()`, `MIN()`, y `GROUP_CONCAT()`).
- Soporte para `LEFT OUTER JOIN` y `RIGHT OUTER JOIN` cumpliendo estándares de sintaxis SQL y ODBC.
- Soporte para alias en tablas y columnas como lo requiere el estándar SQL.
- `DELETE`, `INSERT`, `REPLACE`, y `UPDATE` devuelven el número de filas que han cambiado (han sido afectadas). Es posible devolver el número de filas que serían afectadas usando un flag al conectar con el servidor.
- El comando específico de MySQL `SHOW` puede usarse para obtener información acerca de la base de datos, el motor de base de datos, tablas e índices. El comando `EXPLAIN` puede usarse para determinar cómo el optimizador resuelve una consulta.
- Los nombres de funciones no colisionan con los nombres de tabla o columna. Por ejemplo, `ABS` es un nombre válido de columna. La única restricción es que para una llamada a una función, no se permiten espacios entre el nombre de función y el '(' a continuación.
- Puede mezclar tablas de distintas bases de datos en la misma.

#### Seguridad

- Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está cifrado cuando se conecta con un servidor.

#### Conectividad

- La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity). Por ejemplo, puede usar MS Access para conectar al servidor

MySQL. Los clientes pueden ejecutarse en Windows o Unix. El código fuente de MyODBC está disponible. Todas las funciones para ODBC 2.5 están soportadas, así como muchas otras.

- La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix.

#### Clientes y herramientas

- MySQL server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas. Estos comandos están disponibles a través de la línea de comandos y el cliente *mysqlcheck*. MySQL también incluye *myisamchk*, una utilidad de línea de comandos muy rápida para efectuar estas operaciones en tablas *MYISAM*.
- Todos los programas MySQL pueden invocarse con las opciones `--help o -?` para obtener asistencia en línea.

## **CAPITULO 3**

## **2 DESARROLLO**

### **3.1 Modelamiento del negocio**

HABITierra es una organización privada con fines sociales y científicos, dedicada al trabajo con comunidades, organizaciones sociales, profesionales e institucionales centrada en temas de desarrollo territorial sostenible y gobernabilidad local.

Para ello cuenta con locales y profesionales repartidos en varios lugares y ámbitos del saber humano, respectivamente. Las oficinas donde generalmente operan están ubicadas en las ciudades de Quito, Cuenca y Machala, siendo la primera la sede principal. Debido al modelo de negocio al que está sujeta la Fundación, se cumplen además con talleres de trabajo en los sitios involucrados con los proyectos de gestión, planificación y gobernabilidad territorial.

HABITierra para brindar una gestión, planificación y gobernabilidad de territorios, ha creado un producto con el cual ha venido trabajando y enriqueciéndose de información de manera continua, y que ha sido de gran ayuda para la generación de un sistema web y que la Fundación toma como base para seguir ofertando a demás instituciones que se unan al trabajo que ofrece. Esta herramienta es conocida como Ge@, de la cual ya se explicó a breves rasgos al inicio de este documento, pero que ahora se complementará con información de cómo está estructurado.

Antes de empezar detallando a Ge@ en su totalidad, se va a explicar de manera general como la empresa realiza su proceso de negocio desde el momento en que un cliente solicita que se le brinde resultados sobre un proyecto y en base a esto una asesoría, basándonos en la Metodología de RUP y haciendo uso del modelamiento teórico de Ge@.

El siguiente diagrama representa los diferentes subsistemas en los que se ha dividido la empresa a nivel de abstracción:

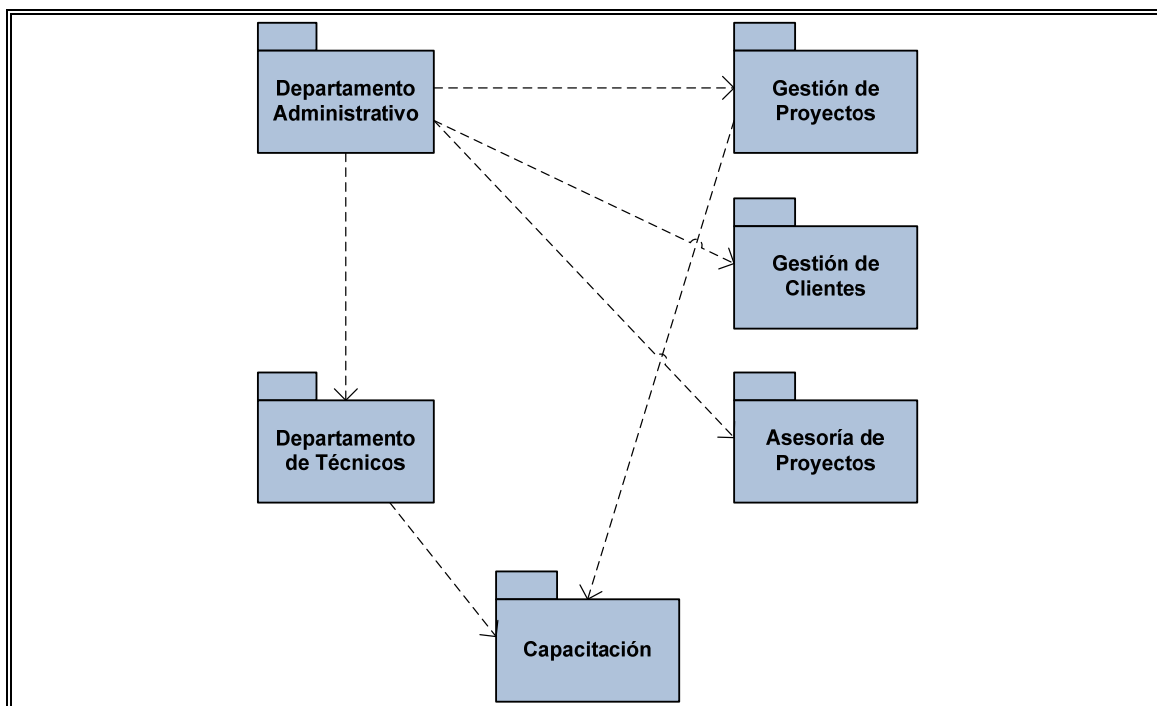


Figura 3-1: Diagrama de los subsistemas de la empresa a nivel de abstracción.

### 3.1.1 Actores y trabajadores

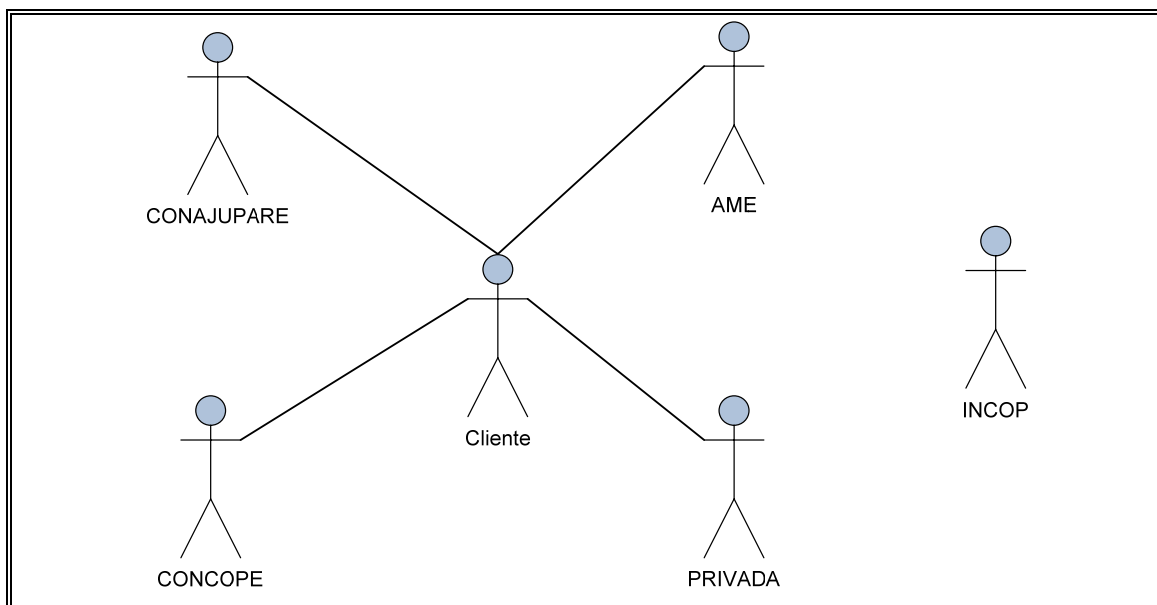


Figura 3-2: Identificación de Actores

Los actores son las entidades individuales que hacen uso de la Herramienta web para la gestión, planificación y gobernabilidad territorial. Los actores que se identifican en el modelo de negocio son:

➤ **Cliente:**

El cliente es el actor encargado de alimentar de información al sistema web con la finalidad de recibir algún resultado que le sirva de apoyo para el desarrollo y avance de su proyecto. Son los principales beneficiarios de la herramienta web. El cliente se clasifica en dos tipos de entidades:

○ **Organizaciones Públicas:**

HABITierra como fundación al brindar soluciones territoriales tiene vínculos directos con instituciones públicas a diferente nivel como son:

**CONAJUPARE:** Consejo Nacional de Juntas Parroquiales Rurales del Ecuador.

**AME:** Asociación de Municipalidades Ecuatorianas.

**CONCOPE:** Consorcio de Consejos Provinciales del Ecuador.

○ **Organizaciones Privadas:**

Debido a la concepción que se maneja en la Fundación de llevar la realidad a sistemas dentro de sistemas, a futuro se puede ofrecer soluciones a distintos ámbitos fuera de lo que es el Ordenamiento Territorial como puede ser el modelamiento de Sistemas Urbanos, Riesgos Bioculturales y Análisis de Recursos y Riesgos, se podría tener clientes no solo a instituciones públicas sino a instituciones privadas.

➤ **Instituto Nacional de Contratación Pública - INCOP**

La fundación HABITierra tiene una dependencia muy directa de un actor externo como es el INCOP, puesto que es uno de los principales medios por los cuales se llega al cliente.

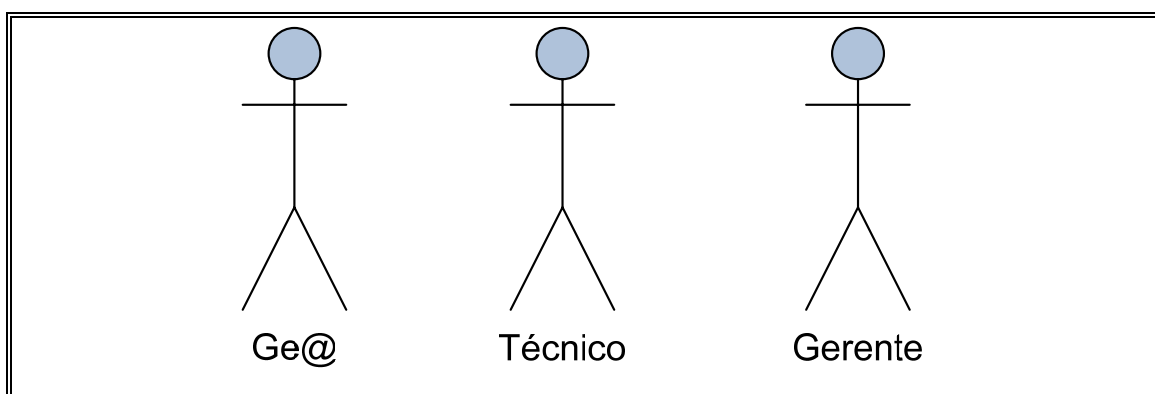


Figura 3-3: Identificación de Trabajadores (Stakeholder)

Los trabajadores o stakeholder, son las personas u organizaciones que están directamente envueltas en la elaboración o tomas de decisiones claves a cerca de la funcionalidad y propiedades del sistema. Para este sistema los trabajadores son:

➤ **Técnico:**

Es un profesional con especialización en las ramas de economía, sociología, ambiental, informática y territorial que forman el grupo de trabajo de la fundación HABITierra. Son los responsables de los sistemas que intervienen en la gestión, planificación y gobernabilidad de territorios.

➤ **Gerente:**

Es la persona que dirige y lidera al grupo de trabajo y toma las decisiones en base a los datos alimentados a la herramienta web. Brinda asesoría y soluciones a los clientes que estén interesados en conocer los resultados.

➤ **Ge@:**

Como se puede ver Ge@ entra como una entidad más dentro del modelo de negocio y dentro del dominio de la organización, por esta razón se lo toma como un trabajador. Ge@ es el medio por el cual los clientes, técnicos y gerentes van a poder coordinar su trabajo y de esta manera llevar a cabo la planificación, gestión y gobierno de territorios.

### Interacción entre actores y trabajadores

Existe una interacción entre los actores y trabajadores, de los cuales se desglosa un conjunto de casos de uso del negocio que se detallan en la siguiente sección, en la imagen inferior se tiene un esquema de esta interacción.

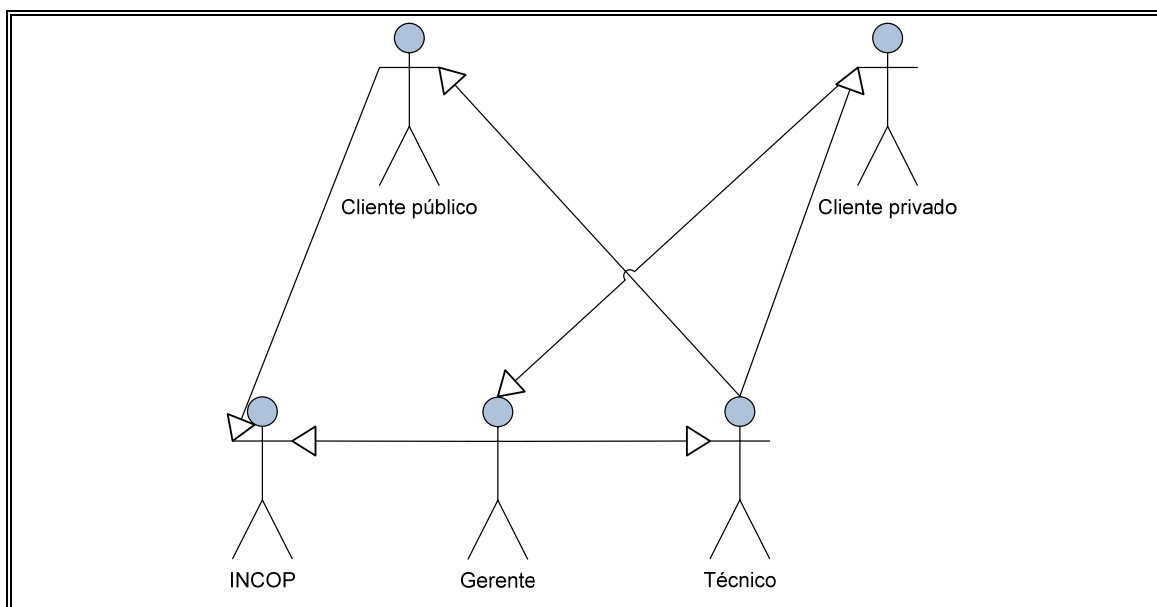


Figura 3-4: Diagrama de actores y trabajadores de negocio (1)

Los actores y trabajadores empiezan a interactuar cuando un cliente sube una oferta al INCOP o se contacta directamente con el Gerente, en el primer caso realizará una



propuesta que deberá ser la ganadora para comunicarse con el cliente; en el segundo caso interactúan directamente. En caso de que se llegue a un acuerdo entre el Cliente y Gerente, es aquí en donde entra en juego Ge@ y es lo que se puede observar en el gráfico inferior.

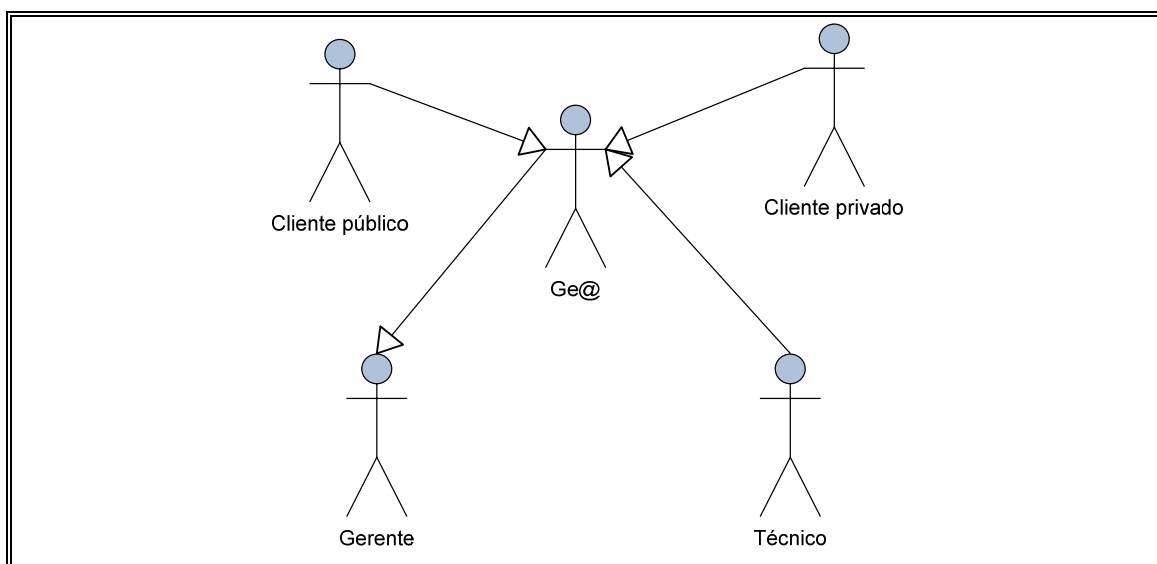


Figura 3-5: Diagrama de actores y trabajadores de negocio (2)

Una vez que el proyecto está en marcha y el gerente ha dialogado con su equipo de trabajo (técnicos), éstos empiezan a alimentar con información a la herramienta web y de ser necesario algunos de ellos capacitan a los clientes para que también aporten con esta tarea, esta capacitación se puede ver en la interacción que existe entre los técnicos y clientes en la figura 3-4.

El Gerente con su grupo de asociados pueden ir revisando cómo se da el avance de los proyectos, revisando tareas, indicadores y gráficos a los que únicamente ellos tienen acceso, es por ello del sentido de la flecha que se tiene entre Ge@ y el Gerente. En esta instancia los gerentes tienen contacto mayormente con sus técnicos y asociados.

Cuando los clientes están familiarizados con la herramienta, conocen su funcionalidad, pueden empezar a requerir mayor información para tomar decisiones, lo que los puede llevar a pedir otro tipo de asesoría adicional, para ello deben contactarse con el gerente que es quién puede brindarles soluciones certeras y personalizadas.

### 3.1.2 Caso de uso del negocio

A partir de la interacción que existe entre los actores y trabajadores, tenemos los siguientes casos de uso:

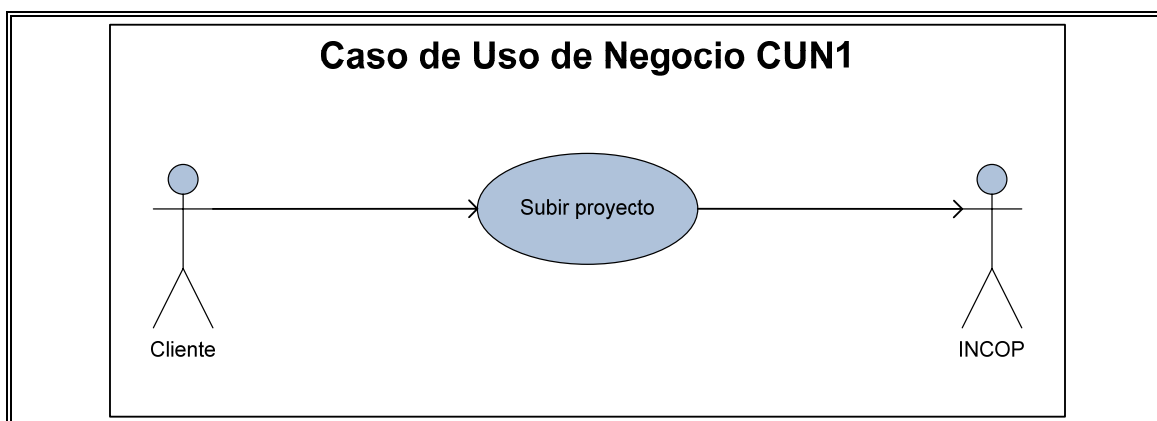


Figura 3-6: Caso de uso de negocio: Subir Proyecto

Caso de uso de negocio CUN1: Subir Proyecto
<p><b><u>Actor principal:</u></b>            Cliente.</p>
<p><b><u>Actores o trabajadores involucrado e intereses:</u></b>            INCOP: Es quién recibe una solicitud por parte del Cliente.</p>
<p><b><u>Precondiciones:</u></b>            La solicitud debe estar en un formato específico.</p>
<p><b><u>Poscondiciones:</u></b>            El cliente espera que se encuentren ofertas.</p>
<p><b><u>Flujo Básico:</u></b></p> <ol style="list-style-type: none"> <li>1. El cliente redacta una petición y la envía al portal de compras públicas.</li> <li>2. El portal la recibe para un posterior análisis si todo está en orden.</li> </ol>

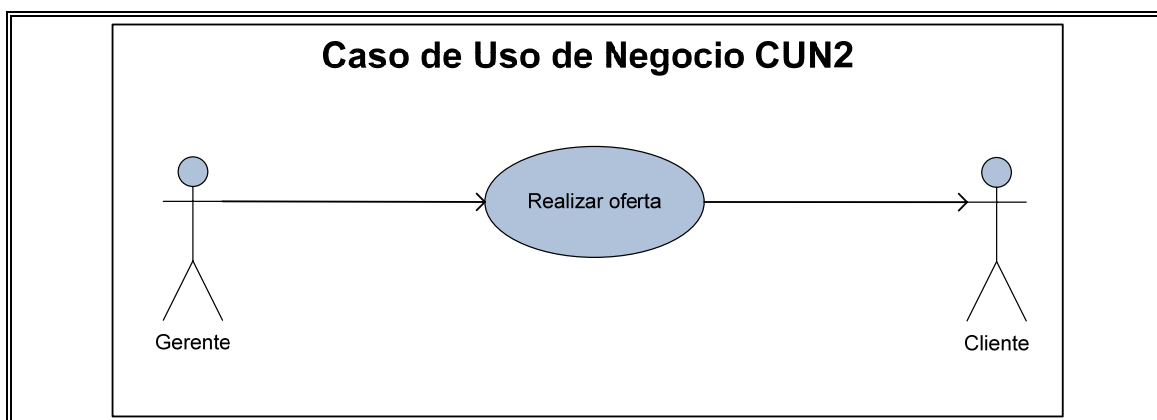


Figura 3-7: Caso de uso de negocio: Realizar oferta

### Caso de uso de negocio CUN2: Oferta Adjudicada

#### **Actor principal:**

Gerente.

#### **Actor o trabajador involucrado:**

INCOP: Mediante este portal el gerente se ha adjudicado el proyecto

Cliente: Acuerda la forma de trabajar y cumplir los requisitos.

#### **Precondiciones:**

Gerente ha presentado oferta que ha cumplido los requisitos y ha ganado.

#### **Poscondiciones:**

Cumplir con los contratos.

#### **Flujo Básico:**

1. La fundación sube una oferta respondiendo a la solicitud del cliente.
2. El INCOP aprueba y da como ganadora a la propuesta.
3. Se contactan el cliente y gerente para realizar el proyecto.

#### **Flujos Alternativos:**

El cliente es privado y se comunica directamente con el gerente.

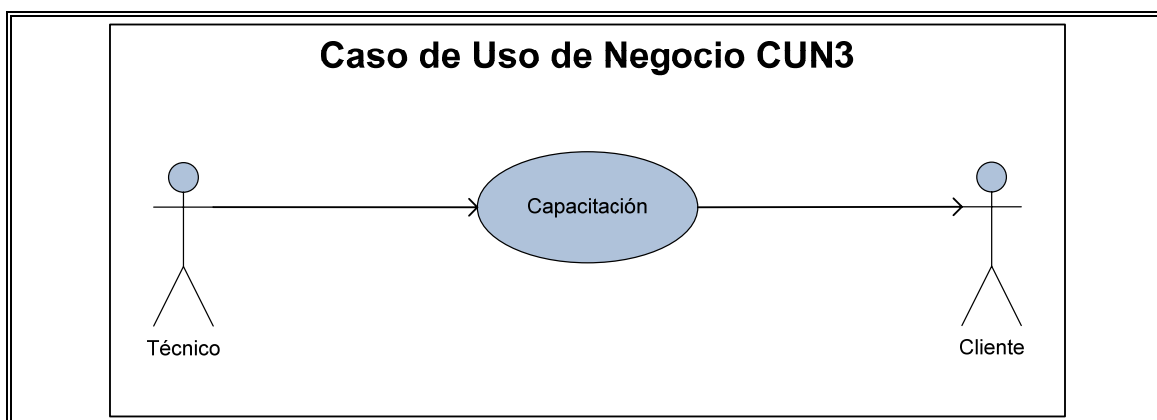


Figura 3-8: Caso de uso de negocio: Capacitación

#### Caso de uso de negocio CUN3: Realizar Oferta

**Actor principal:**

Técnico.

**Actor o trabajador involucrado:**

Cliente: recibe capacitación.

**Poscondiciones:**

Cliente o personal del cliente está en la capacidad de operar y colaborar en el proyecto.

**Flujo Básico:**

1. Se eligen técnicos idóneos.
2. Técnicos conocen operación del sistema y conocen necesidades y objetivos del cliente.
3. Los técnicos capacitan a los clientes.
4. Clientes colaboran proporcionando y dotando de información a los técnicos y directamente a los sistemas informáticos que ahora conocen y operan sin inconveniente.

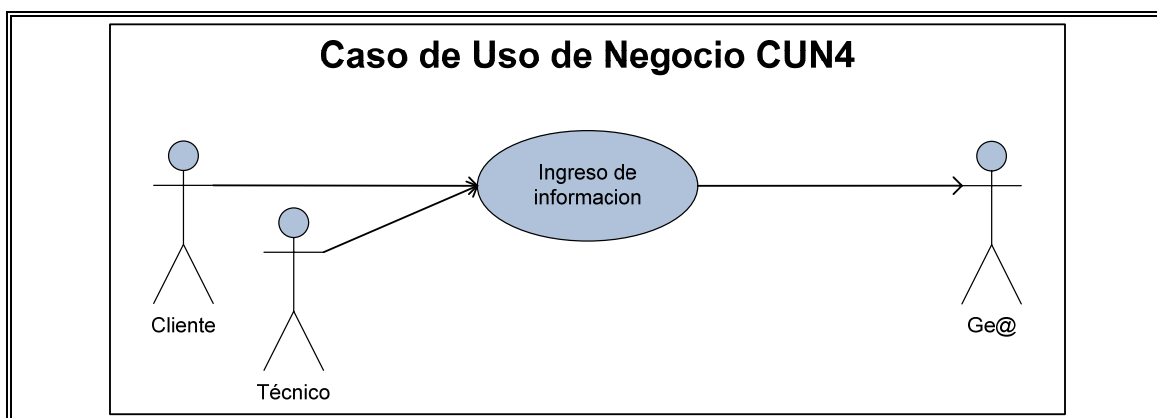


Figura 3-9: Caso de uso de negocio: Ingreso de información

#### **Caso de uso de negocio CUN4: Ingreso de Información**

##### **Actor principal:**

Ge@.

##### **Actor o trabajador involucrado:**

Cliente: realiza la acción de ingresar datos a la herramienta.

Técnico: realiza la acción de ingresar datos a la herramienta.

##### **Poscondiciones:**

Cliente o personal del cliente está en la capacidad de operar la herramienta.

##### **Flujo Básico:**

1. Se toma la información que se ha recopilado.
2. Se procede al ingreso de la información al sistema.
3. Tanto el técnico o cliente deben de conocer el funcionamiento informático y teórico de Ge@.

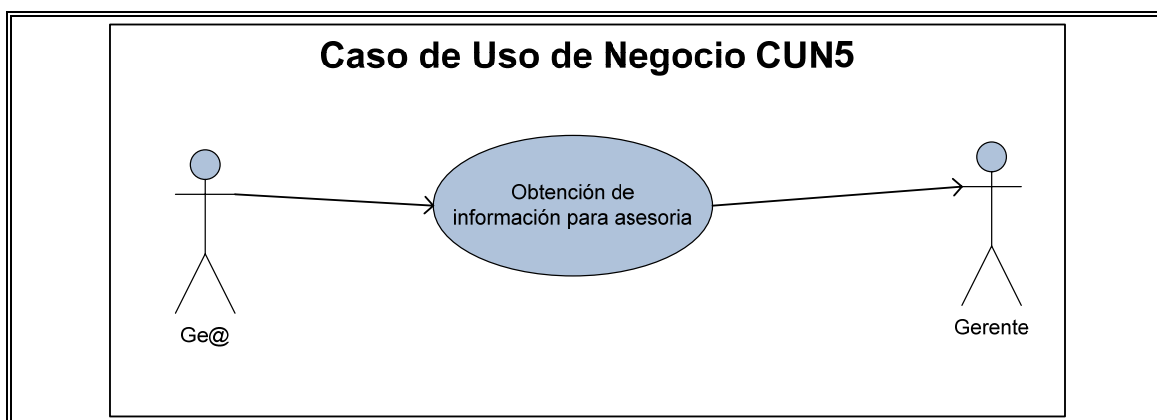


Figura 3-10: Caso de uso de negocio: Obtención de información para asesoría.

#### **Caso de uso de negocio CUN5: Ingreso de Información**

##### **Actor principal:**

Gerente.

##### **Actor o trabajador involucrado:**

Ge@: brinda la información necesaria y con un formato entendible para el gerente.

##### **Precondiciones:**

El gerente debe saber el uso de la herramienta.

##### **Poscondiciones:**

Ge@ debe de estar configurado de tal manera que el Gerente no tenga problemas en interpretar la información.

##### **Flujo Básico:**

1. Se genera reportes por parte de Ge@.
2. Esta información es receptada por el Gerente
3. El gerente interpreta la información para luego brindar la asesoría.

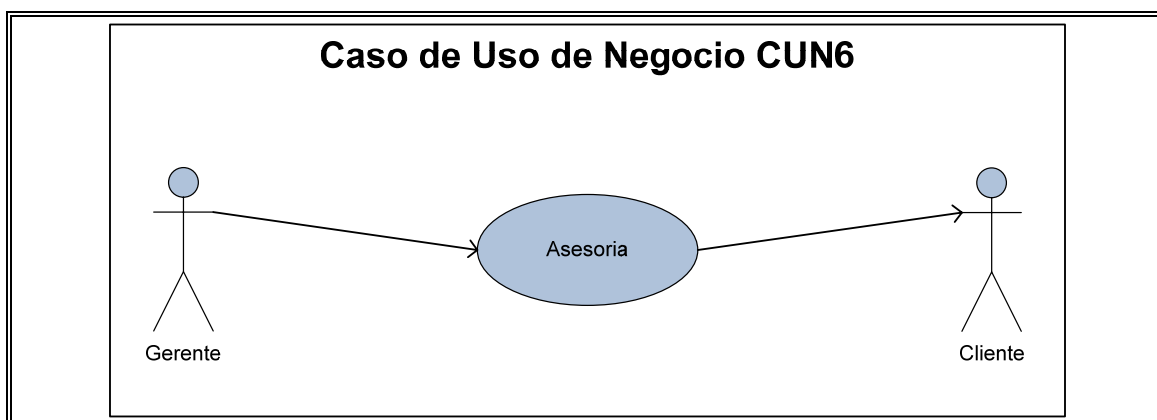


Figura 3-11: Caso de uso de negocio: Asesoría

#### Caso de uso CUN4: Asesoría sobre el proyecto

**Actor principal:**

Gerente.

**Personal involucrado e intereses:**

Cliente: Requiere mayor información.

**Precondiciones:**

El cliente ha manejado el sistema y ha proporcionado información relevante.

**Poscondiciones:**

El gerente proporciona asesoría necesaria.

**Flujo Básico:**

1. Cliente entiende y maneja correctamente las herramientas.
2. Gerente conoce el proyecto y sus avances.
3. Cliente solicita ayuda específica y personalizada.
4. Gerente proporciona servicios que pueden ser adicionales a los acordados.

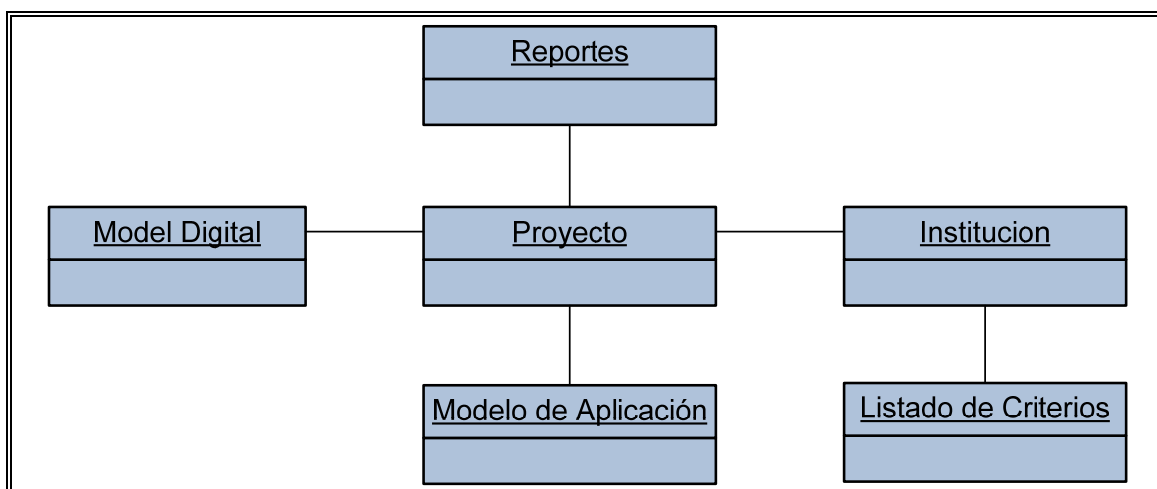


Figura 3-12: Modelo de Dominio

### 3.1.3 Visión, ambiente y terminología de Ge@

Ge@ es una herramienta modelizado teóricamente y digitalmente; y programado informáticamente como sistema, sirve para intervenir en la planificación, gestión y gobernabilidad de sistemas territoriales de forma científica, democrática y sostenible.

Se ejecuta a través de 5 modelos de aplicación: 1) Ordenamiento Territorial, 2) Sistemas Urbanos Complejos, 3) Gestión de Regiones Bioculturales, 4) Recursos y Riesgos Territoriales, 5) Gobernabilidad Territorial.

La unidad operativa de Ge@ es el sistema territorial, es por eso que la herramienta web está enfocada precisamente a esa área es decir a los modelos de aplicación 1) y 5), permitiendo un crecimiento a futuro del resto de modelos de aplicación gracias a la manera en que está desarrollada.

Los modelos del sistema territorial están centrados en 5 subsistemas (socio-cultural, ecológico-territorial, físico-espacial, económico-productivo y político-administrativo) y de la misma manera por elementos que interactúan para estructurar los subsistemas.



## **Sistema**

Un sistema en general consiste en una totalidad organizada, por la interacción de sus partes (subsistemas y elementos básicamente) y contiene propiedades que ninguna de las partes la posee. En este sentido el sistema operativo de Ge@, se estructura y funciona por la interacción de 7 subsistemas fundamentales:

- 1) Principios
- 2) Modelación de Sistemas
- 3) Sistema Territorial
- 4) Modelos de Aplicación
- 5) Diseño de momentos y procesos
- 6) Desarrollo de procesos y
- 7) Capitalización del conocimiento.

Los subsistemas a su vez, se estructuran y funcionan por la interacción de sus elementos. El sistema operativo, hace que Ge@ sea una gran herramienta para interactuar con el sistema territorial. En este sentido Ge@ tiene, modelos, momentos, procesos y herramientas para trabajar de forma interactiva con el equipo técnico y con el sistema, así su propósito fundamental es *interactuar con un sistema territorial para cambiarlo*.

### **3.1.3.1 Principios**

#### **3.1.3.1.1 Científicos**

Podemos afirmar que la ciencia y específicamente su método científico es lo que permite investigar, resolver problemas y tener como producto final el conocimiento científico y en consecuencia el aprendizaje constante. En este sentido es necesario e indispensable tratar a todas las acciones y procesos que se realicen con Ge@ mediante el método científico.

El conocimiento científico debe ser traducido en instrumentos para que mejoren y desarrollen la capacidad comunitaria de monitorear y evaluar los procesos de planificación, gestión y gobernabilidad que se ejecutan en sus territorios.

### **3.1.3.1.2 Democráticos**

La democracia la asumimos como una forma dinámica de *cooperación humana* (cambia, se adapta y evoluciona), basada en principios universales y locales de sostenibilidad, para alcanzar el desarrollo territorial y cognitivo de nuestro sistema global. Esto, a partir de sistemas locales y producto de procesos de planificación, gestión y gobernabilidad-participativos, científicos y creativos.

La cooperación humana, entendida como la capacidad de trabajar juntos por propósitos comunitarios. Priorizar la necesidades, intereses y aspiraciones comunitarias a las personales, familiares y grupales para lograr la equidad social y territorial.

### **3.1.3.1.3 Sostenibles**

Comprender los principios de organización de la naturaleza y utilizarlos en el desarrollo territorial: procesos y leyes de la naturaleza.

Nada debe afectar la capacidad innata de la naturaleza de sustentar la vida. Entrar en la transición sostenible a nivel cultural, ecológico, urbano, económico y político de los sistemas territoriales.

### **3.1.3.1.4 Desarrollo**

Un modelo de desarrollo hegemónico<sup>7</sup> no tiene sentido y no es coherente con la naturaleza ni con los 13.500 millones de años de evolución, por la inmensa variedad de ecosistemas, geo sistemas y sistemas culturales existentes en todos los lugares del planeta.

El desarrollo de procesos y la creación de objetos, debe basarse en las leyes naturales y adecuarse a la evolución y creación (manifiesta en las diversidades físicas, biológicas y culturales).

---

<sup>7</sup> Se refiere al dominio que ejerce una entidad sobre otras del mismo tipo, en este caso a un modelo que domine.

Participación, aprendizaje y empoderamiento de la población en el desarrollo, es la clave para lograr la identidad. En este sentido la participación debe transitar hacia autogobernarse, el aprendizaje hacia el conocimiento del funcionamiento de la naturaleza universal y el empoderamiento a intervenir en las decisiones trascendentales.

### 3.1.3.1.5 Conocimiento

La esencia del desarrollo del conocimiento está en la interacción de teorías, métodos y herramientas (T-M-H's). El conocimiento fluye a través de la red de comunicaciones y es transmitido de generación a generación. Los valores, las creencias y las formas de relacionarse con el entorno son resultados de un proceso de *adquisición, readecuación y reformulación* del conocimiento lo que de hecho permite la sostenibilidad del sistema cognitivo.

El conocimiento es el capital fundamental que tiene la especie humana para su desarrollo. Su capitalización –a través de interacciones que producen sinergias<sup>8</sup>- es la clave para desarrollar aptitudes, actitudes y conciencia frente al desarrollo territorial sostenible, así la participación social se vuelve creativa.

La aplicación de Ge@ nos debe llevar a lograr un conocimiento científico debidamente verificado, cotejado y experimentado, a esto podemos llamarlo rigor científico.

### 3.1.3.2 Modelación de Sistemas

El sistemismo explica la formación, mantenimiento, reparación o desintegración de una cosa compleja concreta de cualquier tipo (sistema) como son: *naturales, sociales, técnicos, conceptuales y semióticos*<sup>9</sup> y de varias combinaciones. También existen o se pueden construir otras formas de clasificar sistemas como por ejemplo: *sistemas vivos, complejos, adaptativos, caóticos o emergentes*.

---

<sup>8</sup>Integración de elementos que da como resultado algo más grande que la simple suma de éstos, es decir, cuando dos o más elementos se unen sinérgicamente crean un resultado que aprovecha y maximiza las cualidades de cada uno de los elementos

<sup>9</sup>Teoría general y ciencia que estudia los signos, sus relaciones y su significado.

Consecuentes con esta forma de pensamiento este modelo (o forma) de trabajar con la realidad (sistemas) despliega siete componentes para crear, analizar, comprender, y gestionar sistemas: *Composición (Cs)*, *Estructura (Es)*, *Mecanismo (Ms)*, *Límite (Ls)*, *Entorno (Ns)*, *Propiedades (Ps)* y *Significado (Ss)*.

Todos los sistemas existentes en el universo pueden ser abordados con el Ms-Ge@. Es necesario señalar que solo los sistemas en donde somos parte los humanos, contienen el componente *Significado (Ss)*.

#### **Ms-Ge@**

$$U(s) = \{C(s), E(s), M(s), L(s), N(s), P(s), S(s)\}$$

En donde:

(Us) = Universo del Sistema

(Cs) = Composición

(Es) = Estructura

(Ms) = Mecanismo

(Ls) = Límite

(Ns) = Entorno

(Ps) = Propiedades

(Ss) = Significado

Ms-Ge@ postula que cualquier sistema *s* puede ser modelado, en cualquier instante dado: en donde el universo de un sistema dado (*Us*) es igual a un conjunto compuesto por los componentes mencionados.

#### **3.1.3.2.1 Composición (Cs)**

Colección de partes, componentes, elementos o el nombre que mejor se ajuste al sistema a tratar. En dependencia del tamaño y complejidad del sistema, sus partes pueden ser clasificadas en niveles para un fácil manejo, por ejemplo en subsistemas de primer, segundo o *n* nivel. O puede ser: sistema, subsistemas y elementos. Este tipo de agrupación y clasificación queda al criterio del equipo o del responsable

#### **3.1.3.2.1.1 Subsistemas**

Los subsistemas en realidad son sistemas dentro de uno mayor; estos están compuestos por elementos que interactúan entre sí. Por ejemplo en un sistema territorial podemos determinar cinco subsistemas: socio-cultural, ecológico-territorial, físico-espacial, económico-productivo y político-administrativo.

#### **3.1.3.2.1.2 Elementos**

Son los componentes concretos y necesarios del subsistema para que este exista como tal. Por ejemplo, en el subsistema ecológico-territorial, sus elementos son: clima, paisaje, flora, fauna, sustrato geológico, hidrología y a lo mejor otros que caractericen de mejor forma dicho subsistema.

#### **3.1.3.2.1.3 Elementos Clave**

Como una forma de iniciar el análisis del sistema procedemos a identificar los elementos claves del mismo. Aunque estos elementos se clarifican con el análisis de interacciones, podemos hacer una aproximación y en base al conocimiento empírico del sistema identificar los elementos que más influyen en el comportamiento de los otros, (por ejemplo a través de un mayor número de interacciones fuertes).

Los elementos clave permiten entender y comprender el funcionamiento de un sistema, en función de los temas a tratar y de la profundidad de análisis requerida.

#### **3.1.3.2.1.4 Variables**

Son los elementos que tienen la propiedad de cambiar significativamente con las influencias del entorno y/o del resto de elementos. La importancia de las variables es muy alta porque son con las cuales debemos trabajar para cambiar el sistema y también porque son las que determinan el equilibrio dinámico del subsistema y del sistema. En el ejemplo del subsistema ecológico-territorial por ejemplo, el *clima* es una variable fundamental.

### **3.1.3.2.2 Estructura (Es)**

La estructura de un sistema se refiere al conjunto de interacciones (interrelaciones, cooperación, complementariedad e interdependencia) entre los componentes de un sistema o de estos con los componentes del entorno (Ns). En este sentido la estructura de un sistema es dinámica, porque a través de las interacciones se producen fenómenos y procesos de realimentación para mantener en equilibrio dicha estructura.

#### **3.1.3.2.2.1 Interacciones**

Cada elemento de un sistema contiene en esencia *energía, materia, información y/o conocimiento* (EMIC), al interactuar este con otro, envía parte o la totalidad de su contenido y luego recibe ese contenido cambiado, en un tiempo (t) determinado. Esto hace que se produzcan cambios permanentes en los elementos y consecuentemente en todo el sistema.

Las interacciones generan procesos al interior de los sistemas y éstos a su vez resultados o productos, que se pueden monitorear, medir y evaluar. Constituyen la esencia de la dinámica de los sistemas y al mismo tiempo son las que mantienen el equilibrio dinámico y la cohesión de las totalidades.

Las interacciones toman mayor importancia al considerar que las propiedades de los subsistemas o elementos de un sistema solo se pueden comprender en términos de su actividad –de su interacción con el entorno y con otros subsistemas o elementos del mismo sistema-. El debilitamiento de las interacciones provoca la descomposición del sistema.

#### **3.1.3.2.2.2 Fenómenos**

Los fenómenos los concebimos como los cambios temporales que experimentan los elementos, productos/resultados, objetos o propiedades de los sistemas y que podemos percibirlos con los sentidos. Son procesos que permiten observar cómo pasa EMIC de un elemento a otro; En síntesis los fenómenos que manifiestan los sistemas, permiten observar sus cambios.

### **3.1.3.2.3 Realimentación**

La realimentación mantiene un sistema en funcionamiento constante durante un cierto tiempo. En rigor la realimentación mantiene funcionando a las interacciones (flujos EMIC) y a los procesos y provoca en gran medida el crecimiento, decrecimiento, regulación, estabilidad o destrucción de un sistema.

La realimentación existe de dos tipos: positiva (auto reforzadora) y negativa (auto reguladora). La primera produce crecimiento o decrecimiento y la segunda regulación y estabilidad. En consecuencia, manejar la realimentación negativa es necesario para alcanzar el equilibrio dinámico del sistema a pesar de las influencias del entorno (Ns) y de las interacciones internas.

Un bucle de realimentación establece interacciones causales entre los elementos que componen el bucle. Manifiesta como la interacción inicial se propaga por todos los elementos, hasta que el último realimenta al primero. Lo importante está en establecer la dirección positiva (+) o negativa (-) de la interacción entre dos elementos del bucle.

La determinación y análisis de la realimentación es la puerta de entrada para la intervención en el sistema.

### **3.1.3.2.3 Mecanismo (Ms)**

El mecanismo se refiere a los procesos que se desarrollan dentro de un sistema y son los responsables de la forma en que se comporta dicho sistema, esto bajo un patrón de organización y de funciones específicas. El mecanismo explica cómo funciona y cómo se comporta un sistema en su totalidad o en sus niveles.

#### **3.1.3.2.3.1 Procesos**

Los procesos en un sistema provocan su crecimiento, cambio, desarrollo y evolución y nos muestran flujos, movimiento y tiempo. Dichos procesos son creados por las interacciones; es decir por los flujos EMIC entre los componentes

de un sistema y su propósito es organizar las actividades y tareas necesarias, para que el proceso se desarrolle de forma armónica, gradual y ordenada.

El manejo de procesos, es clave y se lo realiza a través del desarrollo de los siguientes ítems: *propósito, métodos-herramientas-recursos, actividades, resultados, indicadores, riesgos y formas de aprendizaje.*

Los procesos generan productos y resultados, los mismos que en interacción (sinergia) con otros productos y resultados de otros procesos (dentro de un mismo sistema o del entorno) hacen que dicho sistema se desarrolle y evolucione.

#### **3.1.3.2.3.2 Patrón de Organización**

Es la configuración de las interacciones entre los elementos de un sistema; es decir es el orden que emerge de dichas interacciones y que posibilita que la estructura del sistema esté organizada armoniosamente, y construida en varios niveles. En definitiva el orden (o patrón) hace que todos los elementos “trabajen juntos” de un modo coherente.

El patrón de organización se manifiesta en la configuración de las interacciones existentes entre los componentes del sistema. Este patrón se encarna en la estructura del sistema.

#### **3.1.3.2.3.3 Función**

Es la parte del mecanismo que posibilita comprender mejor la realidad o una parte de ella (como lo es un sistema), porque manifiesta el papel fundamental que desempeña el sistema dentro de la realidad. Encontrar las funciones específicas de un sistema (procesos que le son peculiares), ayuda a descubrir el mecanismo.

#### **3.1.3.2.4 Límites (Ls)**

Excepto el universo, ningún sistema es infinito, en consecuencia posee ciertos límites que lo configuran y lo definen, es decir lo ponen en su justa medida para ser comprendidos e intervenidos.



#### **3.1.3.2.4.1 Fronteras Físicas**

Cotidianamente “ponemos” límites físicos a los objetos; es decir los delimitamos. Pero también es posible delimitar físicamente a casi todo tipo de sistema, porque requieren de un pedazo de corteza terrestre para desplegar sus “actividades”. Las fronteras físicas pueden delimitarse artificialmente, por ejemplo las fronteras entre países, o los límites cantonales.

#### **3.1.3.2.4.2 Problemática**

Otra forma de definir límites, delimitar o configurar un sistema complejo es determinar la problemática que se va a tratar de dicho sistema. Por ejemplo de un sistema territorial podemos abordar problemáticas como: *ordenamiento territorial, desarrollo organizacional, dinámica financiera, asentamientos humanos, medioambiente, etcétera*. Es configurada por la persona o el equipo responsable.

#### **3.1.3.2.4.3 Horizonte de Sucesos**

Todo sistema tiene un “campo de influencia”, es decir, su funcionamiento incide sobre otros sistemas y lugares. Esto es importante determinar porque da un primer acercamiento y conocimiento sobre el entorno del sistema. Por ejemplo, si bien un municipio opera sobre un territorio definido, sus prácticas, su política, etcétera, pueden influir en otros municipios, pero también la superposición de espacios entre organizaciones sociales trae consigo conflictos.

#### **3.1.3.2.4.4 Marco Conceptual**

El marco conceptual, determina una forma específica de abordar la problemática del sistema. Este marco conceptual determina como entender la realidad, como concebir la convivencia humana o como entender el desarrollo local, la sostenibilidad, etcétera. Por ejemplo para tratar el ordenamiento territorial podemos asumir un marco conceptual que priorice la equidad, la sostenibilidad y la solidaridad, otros pueden priorizar el mercado, la competencia y la segregación.

### **3.1.3.2.4.5 Niveles**

Prácticamente todos los sistemas están estructurados por niveles mayores que contienen a otros menores. En este sentido podemos escoger que nivel o niveles de un sistema se va a llegar con nuestra intervención. Por ejemplo un nivel en el sistema territorial sería trabajar en la dimensión de la vivienda para cierto grupo social, con un diseño, uso de materiales y de energía que no afecte al medioambiente, que sea económica y accesible y que se articule a una política de vivienda.

### **3.1.3.2.5 Entorno (Ns)**

El entorno de un sistema se lo puede concebir como un “sistema mayor” con el cual está en constante interacción y que en cierta medida influye en el sistema “menor” y provoca cambios.

Este entorno tiene su propia estructura y dinámica (interacciones entre sus subsistemas, elementos, variables y claves) y lo que más nos preocupa: su dinámica. El entorno influye en el sistema a través de las interacciones que se desarrollan y en dependencia de su fuerza de influencia fuerte, media o débil.

Operativamente hemos denominado factores a los componentes del entorno y realizamos una clasificación geográfica del entorno: global, regional, nacional y local.

#### **3.1.3.2.5.1 Subsistemas.**

Son los factores generales

#### **3.1.3.2.5.2 Elementos.**

Son los factores específicos

#### **3.1.3.2.5.3 Variables.**

Factores cambiantes

#### **3.1.3.2.5.4 Claves.**

Factores muy influyentes

### 3.1.3.2.6 Propiedades (Ps)

La propiedad (o atributo) de un sistema -subsistema, elemento o nivel- se refiere a las características cualitativas que emergen de su funcionamiento como totalidad y que proporcionan identidad y conducta, es decir que la propiedad lo diferencia de otros sistemas similares, por ejemplo: Salitre, capital montubia del Ecuador.

#### 3.1.3.2.6.1 Equilibrio Dinámico

Circular en bicicleta es el ejemplo práctico del equilibrio dinámico y esto sucede en todos los sistemas vivos. En este sentido, se debe lograr que cada componente de un sistema esté funcionando en equilibrio con el resto de componentes y el todo con su entorno.

#### 3.1.3.2.6.2 Estabilidad Evolutiva

Esta es una propiedad del sistema como totalidad y quiere decir que para que evolucione debe contar con cierta estabilidad o también que está estable pero en movimiento. Es similar a volar un avión con piloto automático. La estabilidad es sinónimo de madurez y de que el sistema ha sido bien planificado y construido y está siendo usado, transformado, gestionado y gobernado de forma sostenible.

#### 3.1.3.2.6.3 Vulnerabilidad

Todo sistema contiene ciertos grados de vulnerabilidad en su interacción con el entorno y esto se manifiesta en la posibilidad de ocurrencia de riesgos. Bajo este concepto podemos decir que un sistema es **fuerte, medio o débil**. Por ejemplo un sistema territorial ecológicamente frágil no soporta en el corto plazo actividades insostenibles.

#### 3.1.3.2.6.4 Flexibilidad

La flexibilidad se traduce como la capacidad del sistema de adecuarse a la incertidumbre y a la diversidad del entorno o de cambiar en función de éstos. Por ejemplo una política de estado sobre vivienda que no se adecúe a la diversidad cultural y territorial es rígida, en consecuencia tiene mucho riesgo de “quebrarse”.

#### **3.1.3.2.6.5 Complejidad**

La complejidad es una propiedad –sobre todo- de los sistemas vivos y esta se define como el grado de interdependencia que contengan los componentes del sistema, también por la cantidad de los mismos y por el número de interacciones que produzcan. En esta categoría están los sistemas organizacionales y territoriales que nos ocupan.

#### **3.1.3.2.6.6 Emergencia**

La emergencia es una propiedad de los sistemas complejos y tienen la propiedad de estar produciendo continuamente novedad, pero al mismo tiempo son propiedades que emergen del funcionamiento del sistema como totalidad, es decir son propiedades que sus partes carecen.

#### **3.1.3.2.7 Significado (Ss)**

El significado lo entendemos como el valor que los seres humanos damos a un sistema o a sus componentes, en función de nuestras necesidades, intereses y aspiraciones.

##### **3.1.3.2.7.1 Sentido**

¿Tiene sentido la existencia de tal o cual sistema dentro de una comunidad humana?, es decir proporciona dicho sistema un fuerte sentido de comunidad e identidad colectiva.

##### **3.1.3.2.7.2 Propósito**

Es la identificación de la razón de ser del sistema dentro de la comunidad. El propósito puede estar estructurado por objetivos, metas y estrategias.

##### **3.1.3.2.7.3 Valores**

La valoración del sistema se produce en función de que éste apoye la satisfacción de necesidades, la promoción de intereses y la consecución de aspiraciones a nivel comunitario.

#### **3.1.3.2.7.4 Conceptos**

En el proceso de significación que se desarrolla entre la comunidad y el sistema específico, intervienen también los conceptos que manejemos, es decir los valores y pensamientos sobre aspectos generales y específicos de dicho sistema.

#### **3.1.3.3 Sistema Territorial**

Ge@ ha sido creado para trabajar específicamente en sistemas territoriales, en consecuencia es necesario anotar algunos principios y características importantes del concepto de sistema territorial. Es necesario señalar que –a partir de estos principios y características- debemos “construir” el territorio a trabajar con la aplicación del modelo de sistemas Ge@. El sistema territorial es una totalidad indivisible, cuyo propósito fundamental es el desarrollo de la especie humana y de la vida.

Territorialmente el sistema abarca el área de asentamiento humano más su área de influencia (horizonte de sucesos), la misma está en función de la diversidad territorial y de la diversidad cultural.

De la interacción dinámica territorio-cultura se despliegan cinco subsistemas: *socio-cultural*, *ecológico-territorial*, *físico-espacial*, *económico-productivo* y *político-administrativo*, la interacción de éstos producen sinergias que conforman y caracterizan el sistema.

Es un sistema altamente complejo y abierto, debido a la inmensidad de elementos y sus interacciones, además de estar ligado al medioambiente para su desarrollo, a través del intercambio de materia, información y conocimiento.

El sistema territorial se convierte en la unidad para la planificación, gestión y gobernabilidad del desarrollo territorial.

Cada subsistema contiene una serie de elementos que la conforman y le otorgan identidad. Las interacciones entre dimensiones y elementos producen diversos tipos de

sistemas territoriales a nivel global. Todas estas interacciones parten de dos dimensiones mayores: naturaleza y cultura.

### 3.1.3.3.1 Socio-Cultural

Es un subsistema fundamental, porque tiene la capacidad de interactuar a través del conocimiento con la naturaleza, e incidir de forma razonada, pero generalmente insostenible.

Las interrelaciones y la cooperación humana, configuran las estructuras sociales; y, los significados y valores compartidos configuran la cultura. La diversidad socio-cultural se produce por las diferentes formas de interrelacionarse, cooperar y otorgar valores y significados, con un condicionante fundamental: el medio ambiente.

Las diferencias entre sociedades y culturas o entre grupos humanos e inclusive entre individuos, se manifiestan en dos triadas: necesidades, intereses y aspiraciones; y, en actitudes, aptitudes y conciencia; de su heterogeneidad u homogeneidad dependen los conflictos o la cooperación.



Figura 3-13: Elementos del subsistema socio-cultural

### 3.1.3.3.2 Ecológico-Territorial

Esta dimensión se refiere al espacio geográfico que sustenta las actividades humanas y a todos los fenómenos naturales. Esta contiene elementos que difieren de un territorio a otro, como por ejemplo, el clima, sustrato geológico, topografía, hidrografía, flora y fauna. Estos elementos en interacción configuran un ecosistema, que a su vez interactúa con la dimensión socio-cultural y configuran el medioambiente, dentro del cual se desarrollan una serie de procesos humanos para la producción infinita de artefactos, que influyen en la sostenibilidad medioambiental.

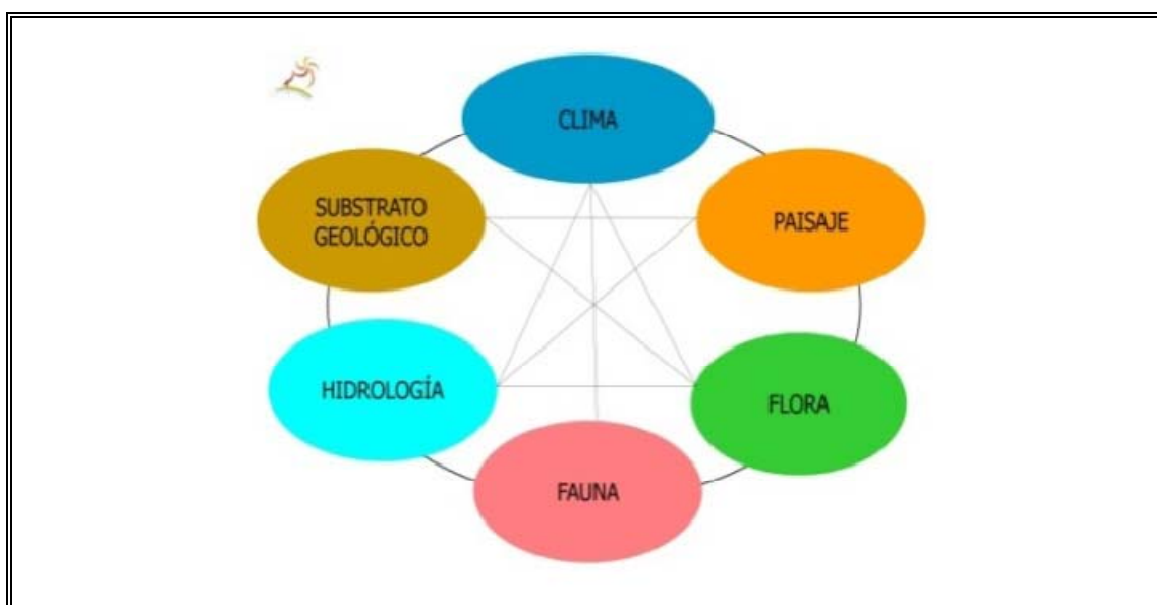


Figura 3-14: Elementos del subsistema ecológico-territorial

### 3.1.3.3.3 Físico-Espacial

Este subsistema es producto de la interacción de los dos anteriores, y se manifiesta plenamente en los asentamientos humanos (aldeas, pueblos, ciudades, y otras), en donde existe una especie de “jardín de artefactos”. En síntesis son todos los artefactos físicos que producen los espacios (ambiente artificial) para desarrollar las actividades humanas. Los asentamientos humanos tienen miles de años de desarrollo y evolución, logrando así un equilibrio dinámico con la naturaleza.

Este equilibrio (entre territorio y cultura) ha experimentado constantemente transformaciones provocadas por “revoluciones” técnicas, sociales y culturales que han funcionado como movimientos telúricos.

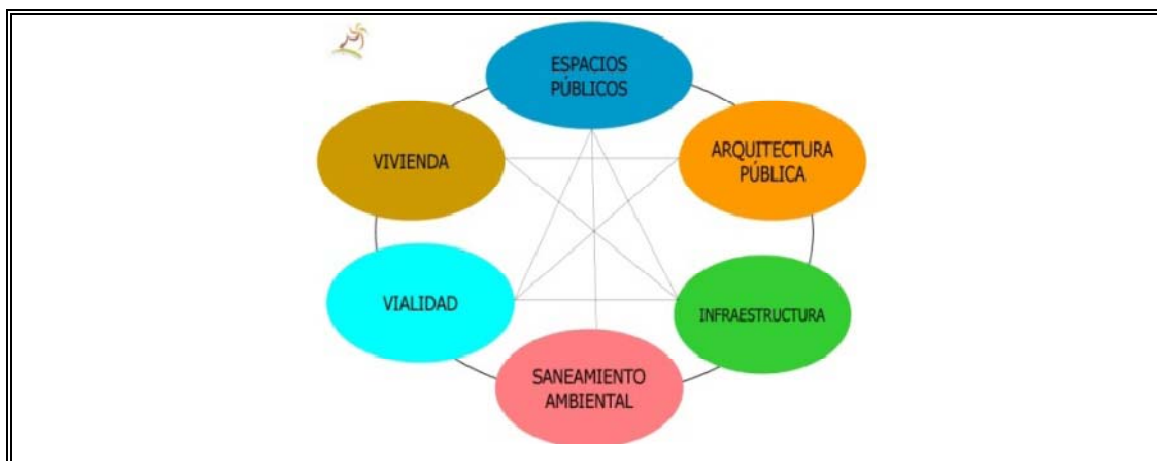


Figura 3-15: Elementos del subsistema físico-espacial

#### 3.1.3.3.4 Económico-Productivo

Este subsistema se refiere a los procesos que los seres humanos ejecutamos para producir e intercambiar bienes y servicios y que provocan una dinámica económica. Esta dimensión es el “motor” de un sistema territorial, consecuentemente lo caracteriza. Por ejemplo, se suele decir, ciudades o pueblos industriales, agrícolas, artesanales, turísticos, universitarios, etcétera, éstas características se convierten en los elementos de esta dimensión.

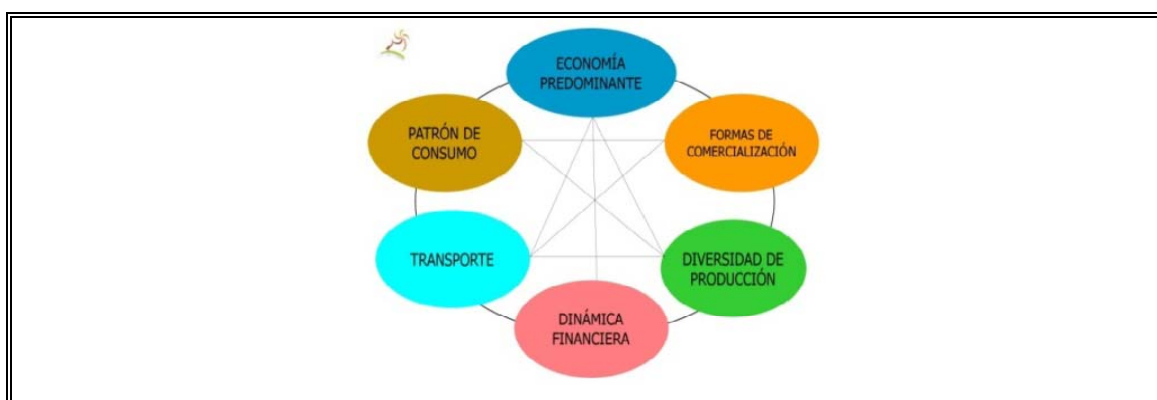


Figura 3-16: Elementos del subsistema económico-productivo



### 3.1.3.3.5 Político-Administrativo

Su propósito o razón de ser es la organización de todo el sistema territorial, organización codificada en normas, leyes, reglamentos o principios y manifiesta en comportamientos y actitudes humanas. Todo esto basado en la cooperación humana - que desde hace miles de años ha sido el motor del desarrollo y evolución humanas-. En esta dimensión interactúan diversos elementos de los procesos de planificar, gestionar y gobernar.



Figura 3-17: Elementos del subsistema político-administrativo

### 3.1.3.4 Modelos de Aplicación

Ge@ interviene en la realidad territorial a través de seis modelos de aplicación:

1. Ordenación Territorial
2. Asentamientos Humanos Sostenibles
3. Gestión de Riesgos Territoriales
4. Manejo de Riesgos Territoriales
5. Gestión del Patrimonio Natural y Cultural
6. Gobernabilidad Territorial

Como se explicó en secciones anteriores gracias a la experiencia que ha venido consiguiendo Ge@ con el paso de los años, se ha podido lograr que el sistema vaya abarcando más modelos de aplicación, actualmente se encuentran estructurados y ya desarrollados bajo el dominio de Ge@ tanto el modelo de aplicación de ordenamiento territorial como la de asentamientos humanos.

Estos modelos de aplicación se basan bajo cinco Momentos de Intervención que conforman el POETAH. En las siguientes secciones se realiza una breve descripción de los dos primeros modelos de aplicación.

#### **3.1.3.4.1 Modelo de Ordenamiento Territorial**

Como se dijo con anterioridad se basa bajo 5 momentos, cada momento está compuesto por un determinado número de ítems necesarios para poder manejar un buen manejo territorial POETAH y cada ítem consta de sus respectivos propósitos, recursos, métodos y herramientas, actividades, resultados, indicadores, riesgos y formas de aprendizaje, todo esto se detalla a continuación:

##### **a) INTUIR-IDEAR**

- a. Establecer objetivos, metas y límites del POETAH.
- b. Conformar el equipo de trabajo.
- c. Establecer la unidad de planificación y manejo: el sistema territorial, sus dimensiones y elementos.
- d. Elaborar la referencia teórica-científica, en base al diálogo de saberes.
- e. Elaborar un plan de trabajo.

##### **b) COMPRENDER-APRENDER**

- a. Realizar un diagnóstico técnico integrado del sistema territorial.
- b. Determinar el estado actual de la producción, uso, transformación y administración de espacio.
- c. Establecer la estructura y dinámica del sistema de asentamientos humanos en el territorio.

- d. Analizar la dinámica del entorno.
- e. Valorar los recursos locales.
- f. Evaluar los riesgos territoriales.
- g. Estructurar el sistema de problemas territoriales.

c) CREAR-PROPONER

- a. Construir y evaluar escenarios territoriales
- b. Diseñar la producción, uso, transformación y administración sostenibles del territorio.
- c. Elaborar un sistema de gestión, para la conversación, uso, protección, restauración y desarrollo de recursos locales.
- d. Crear un modelo de asentamientos humanos sostenibles.
- e. Estructurar un sistema de políticas, programas y proyectos de desarrollo territorial (PPP).

d) IMPLEMENTAR-EVALUAR

- a. Implementar un sistema de regulación del ordenamiento territorial.
- b. Instaurar un sistema de indicadores de desarrollo territorial y de participación comunitaria (observación y control).
- c. Implementar un sistema de gestión del POETAH.

e) CAPITALIZAR-REALIMENTAR

- a. Generar una base de datos e información.
- b. Gestionar una base de conocimientos.
- c. Manejar una red de procesos de desarrollo territorial.

#### **3.1.3.4.2 Modelo de Asentamientos Humanos**

De igual manera que en el anterior modelo, este se conforma de los mismos 5 momentos con sus respectivos ítems y cada uno especificando su respectivo propósito, teoría, método, herramienta, actividades, resultados, riesgos, indicadores y formas de aprendizaje que a continuación se detallan:

- a) INTUIR-IDEAR
  - a. Decisiones y compromisos compartidos sobre el asentamiento sostenible.
  
- b) COMPRENDER-APRENDER
  - a. Los recursos locales: potencialidades y riesgos.
  - b. Establecer el equilibrio dinámico entre cultura y territorio.
  - c. Construir escenarios.
  
- c) CREAR-PROPONER
  - a. Planificación sistémica de los asentamientos y sus subsistemas.
  - b. Diseñar espacios urbanos y arquitectónicos.
  - c. Gestión de la tecnología actual.
  - d. Diseñar políticas y estrategias.
  - e. Diseñar programas y proyectos de desarrollo.
  
- d) IMPLEMENTAR-EVALUAR
  - a. Construcción del asentamiento.
  - b. Construcción del capital sinérgico.
  - c. Elaborar un sistema de indicadores.
  - d. Implementar el sistema de buena gobernabilidad del asentamiento.
  
- e) CAPITALIZAR-REALIMENTAR
  - Se aplican los mismos procesos que el resto de modelos.

#### **3.1.3.4.3 Modelo de Gestión de Territoriales**

Este modelo de gestión está en su fase de diseño.

#### **3.1.3.4.4 Modelo de Manejo de Regiones Bioculturales.**

Este modelo está en su fase de experimentación. Su primera aplicación fue con la capitalización del conocimiento indígena sobre la planificación, gestión y gobernabilidad de sus territorios

#### **3.1.3.4.5 Modelo de Gestión del Patrimonio Natural y Construido.**

Este modelo tuvo una fase de experimentación preliminar, con la ejecución de un proyecto sobre Renovación Urbana Comunitaria de los Barrios Histórico-Populares de la ciudad de Cuenca, ahora esa experiencia sirve para el proceso actual de rediseño del modelo.

#### **3.1.3.4.6 Modelo de Gobernabilidad Territorial.**

Este modelo fue el primero en ser aplicado Ge@, con la construcción de un Sistema de Buena Gobernabilidad Local. En la actualidad se sigue empleando como una aplicación de escritorio; actualizando y modelizando la experiencia como apoyo al rediseño del modelo.

### **3.1.3.5 Diseño de Momentos y Procesos**

#### **3.1.3.5.1 Intuir-Idear**

Es el momento en donde se manifiestan las primeras intuiciones de cómo intervenir en el sistema territorial por parte de los diferentes actores y agentes del desarrollo territorial, estas intuiciones evolucionan hacia ideas, las mismas que se materializan en un equipo y un programa de trabajo. Este momento evolucionará y realimentará al resto de momentos.

#### **3.1.3.5.2 Comprender-aprender**

Todos los participantes comprenden los objetivos, metas, teorías, métodos y herramientas del POETAH para luego aprender colectivamente como funciona un sistema territorial particular y su ubicación e interacción con el mismo. Este

componente se materializa en un documento que defina el estado de situación de dicho sistema territorial.

#### **3.1.3.5.3 Crear-proponer**

Con las capacidades adquiridas de los participantes en los dos momentos anteriores, se inicia la creatividad colectiva de propuestas para cambiar el sistema territorial. Este momento se manifiesta en políticas, programas y proyectos territoriales y en sistemas operativos particulares como por ejemplo: sistema de gestión del POETAH, sistema de manejo de recursos, mecanismos para la producción, uso, transformación y administración sostenibles de los espacios o modelo de calidad sensible, etcétera.

#### **3.1.3.5.4 Implementar-evaluar**

En este momento se ejecuta lo creado y propuesto y se verifican los cambios que producen en el sistema territorial. Se materializa este momento en la constatación y evaluación de dichos cambios a través de ciertos mecanismos y especialmente de la aplicación de variables e indicadores estructurados en un sistema específico.

#### **3.1.3.5.5 Capitalizar-realimentar**

La elaboración de un POETAH genera una cierta cantidad de teorías, métodos y herramientas –es decir conocimiento- sobre temas concretos del desarrollo territorial sostenible. La aplicación del mismo produce experiencias y aprendizaje. En este sentido este componente tiene el propósito de capitalizar dicho conocimiento y realimentar los procesos de elaboración e implementación de un POETAH. Así garantizamos que el modelo o prototipo de ordenamiento territorial y también de Ge@ se actualice permanentemente.

### **3.1.3.6 Desarrollo de Procesos**

#### **3.1.3.6.1 Propósito**

Que nos proponemos lograr, es visualizar los objetivos y metas del componente

### **3.1.3.6.2 Teoría**

Es el conjunto de proposiciones acerca del funcionamiento del mundo, el proceso de generación de teorías está lleno de pasión, imaginación y euforia, de verificaciones prácticas de teorías y afinamiento

### **3.1.3.6.3 Método**

Formas de buscar objetivos específicos, es decir un conjunto de procedimientos y técnicas sistemáticas para abordar asuntos o problemas determinados.

### **3.1.3.6.4 Herramientas**

Todo utensilio mental o material que nos permita realizar una tarea, y que nos provoquen nuevas formas de pensar y actuar en el mundo

### **3.1.3.6.5 Actividades**

Conjunto de tareas que nos permiten lograr uno o varios resultados

### **3.1.3.6.6 Resultados**

Productos medibles y verificables, en función del propósito.

### **3.1.3.6.7 Indicadores**

- Cantidad
- Calidad
- Tiempo
- Lugar
- Grupo social

### **3.1.3.6.8 Forma de Aprendizaje**

Este ítem es fundamental para lograr la trascendencia del componente, el desarrollo del conocimiento y transformaciones de Ge@. Paralelamente debemos considerar que este proceso apoya a practicar una disciplina que equivale a seguir un camino de

aprendizaje, en este sentido podemos decir que la disciplina del aprendizaje es un conjunto de técnicas basadas en una teoría e imagen del mundo, cuya práctica exige estudio y concentración. De esta manera nuestra capacidad evoluciona de tal modo que aprendemos a mirar el mundo de otras maneras.

### 3.1.3.6.9 Riesgos

Toda acción humana conlleva un riesgo, que no es posible manejarlo desde el interior del proceso, sino que obedece a supuestos externos. En este sentido debemos planificar los procesos con el factor riesgo.

### 3.1.3.7 Capitalización del Conocimiento

La capitalización del conocimiento se consigue con la utilización de una base de datos, la cual Ge@ ha venido recolectando información desde que nació por primera vez, actualmente se dispone de una base muy rica en información de las comunidades con las que se ha trabajado.

A partir de la base de datos se genera una base de conocimiento la cual realiza cuatro pasos básicos:

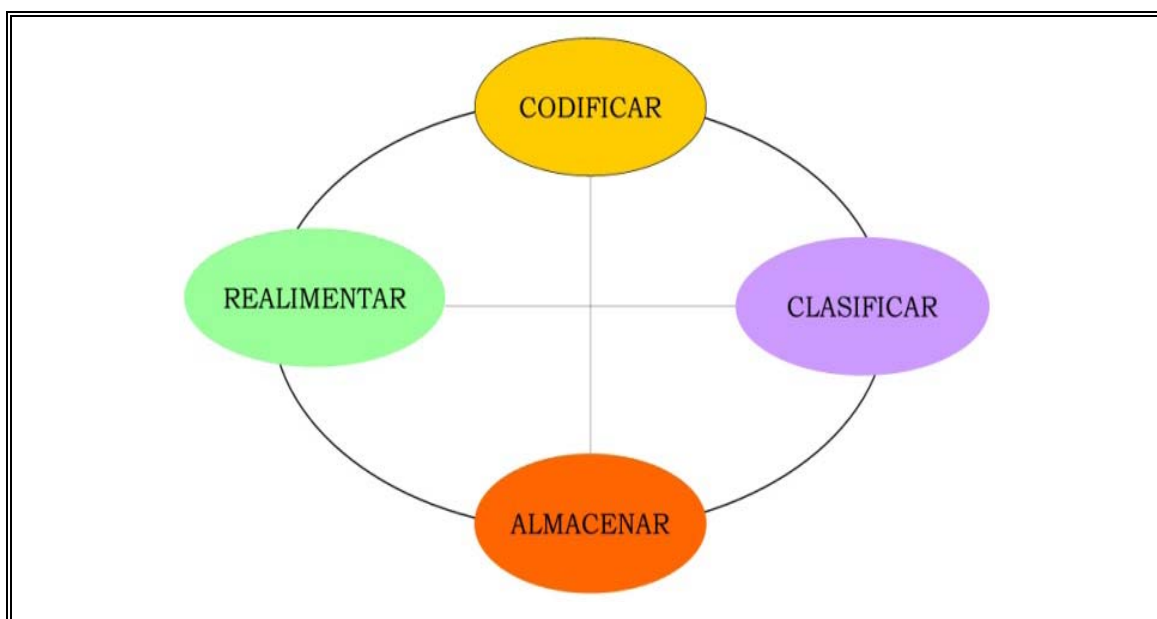


Figura 3-18: Pasos de una Base de Conocimiento



**Codificar.-** el conocimiento humano puede codificarse esencialmente en teorías, métodos y herramientas. En este caso específico lo hemos hecho con cinco ítems más, como lo veremos más adelante.

**Clasificar.-** El conocimiento, las prácticas y las experiencias se clasifican en un primer nivel por las dimensiones de la gobernabilidad: planificar, gestionar y gobernar, y en un segundo nivel por los 26 elementos distribuidos entre las tres dimensiones.

**Almacenar.-** El almacenamiento del conocimiento se ejecuta en la misma estructura de la base, pero de forma digital –que ha sido programada para este proyecto-. El funcionamiento se presenta en un modelo digital está disponible en CD.

**Realimentar.-** La realimentación del conocimiento se da en la medida en que la base se utilice. Así tendremos una actualización y desarrollo permanentes. En la medida en que ingresa más conocimiento codificado, se procede a capitalizarlo en función de las similitudes y diferencias entre el conocimiento existente y el que es introducido.

El propósito fundamental de la base de conocimiento es capitalizar<sup>10</sup> de forma masiva el conocimiento, las creencias y las experiencias humanas sobre temas específicos. En este caso sobre las formas de planificar, gestionar y gobernar sistemas territoriales por parte de las nacionalidades y pueblos indígenas. La capitalización significa “poner en valor” un recurso existente (o un capital) y transformarlo en una inversión para el desarrollo futuro.

Asimismo, la capitalización es, en sí mismo, un proceso de aprendizaje. Aprendemos de los conocimientos acumulados, de las buenas y de las malas prácticas. Para ello necesitamos herramientas de aprendizaje que nos permitan un aprovechamiento pleno de esos conocimientos y prácticas, a través de su sistematización.

---

<sup>10</sup>Es amplificar dos o más conocimientos, prácticas o experiencias sobre un mismo tema, en este sentido NO es la suma, sino la interacción entre conocimientos, prácticas o experiencias. La interacción se ejecuta a través de los ocho ítems de codificación del conocimiento (significado de origen, significado local, propósito, teoría, método, herramienta, lo opuesto y complementario: la dualidad, e interacciones clave), así por ejemplo interactúan dos o más teorías, dos o más métodos, o significados, etc., unificando las similitudes y dejando a las diferencias en suspenso. Así logramos que el conocimiento capitalizado de dos o más personas sea mucho más que simplemente unirlos, entonces el conocimiento se amplifica, porque el nuevo conocimiento (amplificado) es diferente, imposible de lograrlo individualmente.

La base de conocimiento es una herramienta con capacidad para codificar, clasificar, almacenar y realimentar el conocimiento y permitir así desarrollar el proceso de capitalización. Funciona como un archivo interactivo de conocimientos. Se accede para obtener y depositar conocimiento sobre diversos temas de gobernabilidad territorial. Para esto es necesario codificar el conocimiento y las experiencias.

La codificación es la estructuración del conocimiento, las prácticas y las experiencias en códigos o formatos, que posibiliten la interacción con otros diferentes. La codificación esencial del conocimiento es determinar teoría, método y herramienta, componentes que están presentes siempre en toda actividad del conocimiento y que la interacción de éstos genera desarrollo cognitivo. En función del proyecto decidimos aumentar cinco componentes, con el propósito de darle un contexto mayor: significado de origen, significado local, propósito, dualidad e interacciones clave. Así la base de conocimiento presenta una codificación de ocho ítems (componentes) y se desarrolla para cada uno de los 26 elementos del sistema de buena gobernabilidad.

Explicación de los ocho ítems:

**Significado de Origen.-** Las dimensiones, variables y elementos de los sistemas se expresan en palabras, términos, categorías y conceptos, estos tienen su origen en el idioma o lengua que los expresan. El significado de origen nos permite conocer la esencia y las transformaciones que ha experimentado en la dinámica espacio-tiempo. En síntesis es el principio, nacimiento y causa de una cosa, es de donde proviene y evoluciona en el tiempo.

**Significado Local.-** Es el conjunto de creencias locales, que se generan en un territorio específico, al que se lo considera suyo. Estas dan lugar a los valores y significados que las sociedades y culturas otorgan a las dimensiones, variables y elementos de los sistemas, constituyen la funcionalidad y la práctica. En el significado local se expresa la realidad, la riqueza y la identidad.

**Propósito.-** Tiene un significado más amplio que misión, porque tiene que ver con la razón de ser de cada elemento y está en función de los principios, variables e indicadores de una gobernabilidad científica, democrática y sostenible. El propósito no tiene

connotación inmediata y por último el término sugiere un proceso reflexivo. El propósito es la voluntad de hacer o no hacer un proceso dentro de un sistema, que supone la toma de decisiones y en la práctica puede verse como un fin común e individual hacia donde se dirige un proceso o un proyecto.

**Teoría.-** Es el conjunto de proposiciones acerca del funcionamiento de objetos y fenómenos del mundo, en espacio y tiempo específicos, el cual se ha sometido a repetidas verificaciones y se ha ganado cierta confianza. Así el proceso de generación de teorías está lleno de pasión, imaginación y euforia, de verificaciones prácticas de teorías y afinamiento. En síntesis es como vemos el funcionamiento del mundo. En este sentido la cosmovisión indígena juega un papel fundamental. Las teorías están en permanente cambio y dependen en gran medida del desarrollo del conocimiento en un lugar específico y de la memoria de un territorio.

**Método.-** Formas de buscar objetivos específicos. Es decir un conjunto de procedimientos y técnicas sistemáticas para abordar asuntos o problemas determinados. También lo vemos como un procedimiento lógico para poder comprobar teorías o hipótesis, planteadas previamente o en el proceso de planificación.

**Herramienta.-** Todo utensilio mental o material que nos permita realizar una tarea, y que nos provoca nuevas formas de pensar y actuar en el mundo. La herramienta al ser un instrumento para ejecutar procesos y lograr resultados, induce de alguna manera a la estrategia, porque cada herramienta contiene de forma implícita su teoría y método.

**Lo Opuesto y Complementario: La Dualidad.-** Cada componente de un sistema siempre tiene uno opuesto y al mismo tiempo uno complementario. Así el uno no puede existir sin el otro. En la filosofía indígena la dualidad es un principio fundamental y se lo ve como complementario: arriba-abajo, bueno-malo, hombre-mujer..., es decir como la capacidad de reunir dos aspectos distintos pero necesarios, en una misma situación.

#### **2.6.1.8 Caja de Herramientas**

La caja de herramientas es un espacio dentro de Ge@ donde se encuentran clasificadas, almacenadas y codificadas las herramientas utilizadas durante la ejecución de

distintas experiencias y proyectos. Los usuarios al acceder a este componente de Ge@, tendrán una panorámica de las herramientas y podrán enriquecerlas con otras o con cambios en las que están de forma científica a través de sus aspectos, como lo veremos más adelante.

Como herramienta aludimos a todo utensilio material o mental, que nos sirve para elaborar procesos y consecuentemente, obtener productos y resultados. Es substancial señalar que en el proyecto entendimos que toda herramienta contiene una teoría que la sustenta y un método que garantiza su uso correcto.

La caja de herramientas “codifica” a cada herramienta a través de seis aspectos básicos que nos permiten comprender integralmente a las herramientas y también podemos modificarlas o adaptarlas organizad

### 1. Orígenes y evolución

Este aspecto se refiere a tiempo, lugar y persona(s) que la crearon y si ha experimentado cambios o transformaciones con su uso. Aquí se presenta la posibilidad para que los usuarios puedan –a partir de la herramienta original realizar adaptaciones a sus necesidades.

### 2. Teoría/método

Toda herramienta tiene un sustento teórico, es la teoría de la cual emerge, porque herramientas sin teoría pueden funcionar en una situación y en otras fallar pero no sabemos porque. La teoría garantiza que las herramientas sean útiles y transformadoras. Así mismo cada herramienta tiene de forma implícita procedimientos y técnicas para su uso, es decir un método específico, que garantiza su uso adecuado y limita sus riesgos.

### 3. Modo de empleo/uso

Este aspecto se refiere a la forma de usar la herramienta para mejorar su efectividad. Esto generalmente lo conoce la gente que tiene experiencia en el uso. Podemos decir también que se refiere a las habilidades para hacer un buen uso.

#### 4. Cambios/resultados

Que se obtiene aplicando una herramienta, los cambios y resultados deben estar perfectamente claros. Este aspecto determina implícitamente los límites de la herramienta, así como escoger adecuadamente la que necesitamos.

#### 5. Herramientas complementarias

Existen ciertas herramientas complementarias por ejemplo la realimentación con las interacciones o auxiliares como la matriz de NIAs es una auxiliar de diálogo creativo. Estas ayudan para lograr la excelencia en el uso y en los resultados que deben provocar la herramienta escogida.

Si bien no existen herramientas “superiores” a otras, podemos decir que unas son prioritarias en dependencia del tema a tratar y del espacio-tiempo a operar. Así, creemos que el Aprendizaje en Equipo ha sido una herramienta constante y de gran importancia, sobre todo, al inicio: ha propiciado la reflexión y el aprendizaje en las personas del equipo, en función del conocimiento de cada una.

A continuación describimos brevemente –once herramientas básicas– en o que consiste cada herramienta:

**1) *Diálogo Creativo***, posibilita aprender en base a la revisión y cambio de creencias de quienes participan en el proceso y a la sinergia de conocimiento e información de todos los participantes; genera creatividad para tomar decisiones inteligentes y, sobre todo, posibilita el que cada quien sepa lo que tiene que hacer, en función de lo que se trate. A esto se suma la potencialidad de esta herramienta para solucionar problemas y paradojas.

**2) *Base de Conocimiento***, y específicamente su estructura, ha servido para guiar nuestro trabajo de recuperación de la información, para los talleres, diálogos y consultas y en correspondencia nos ha servido para codificar, almacenar y capitalizar el conocimiento.

3) **Consulta**, ha sido una forma alternativa de obtener conocimiento de líderes o grupo de personas sobre temas concretos de la gobernabilidad territorial.

4) **Taller de Trabajo**, lo hemos realizado cuando el diálogo creativo no era pertinente. Los talleres se desarrollaron con trabajos en grupo, sobre temas específicos de planificación, gestión y gobernabilidad.

5) **Realimentación**, herramienta sistémica que nos ha permitido ver con mayor claridad patrones de organización, estructuras, procesos y significados de los sistemas de gobernabilidad y territorialidad. La activación de los ciclos de causa y efecto, los procesos de realimentación tanto reforzadores como compensadores, determinan, finalmente los procesos de cambio de un sistema. La realimentación ha sido fundamental en la formulación y solución de problemas.

6) **Interacción**, es la herramienta que posibilita hacer un análisis y propuestas del flujo de energía, materia, información y conocimiento que experimentan los componentes de los sistemas y las variables de los problemas en un tiempo-espacio determinado. Al mismo tiempo, permite simular los cambios de un sistema a través de sus interacciones. Como parte complementaria de esta herramienta contamos con la Matriz EMIC, que sirve para establecer los contenidos de energía, materia, información y conocimiento que contiene cada componente de un sistema.

7) **Aprendizaje en Equipo**, el aprendizaje humano puede ser individual o colectivo. En equipo es mucho más porque enriquece el conocimiento de todos y también de cada uno. Además la inteligencia, capacidad y conocimiento que se logra, es mucho mayor a la suma del talento intelectual de cada uno. Los seres humanos aprendemos de forma cíclica va de la acción a la reflexión, de la actividad al reposo. En este sentido esta herramienta está diseñada.

8) **Matriz NIAs (de necesidades, intereses y aspiraciones)** es una herramienta muy útil para determinar los conflictos de intereses y de poder. Paralelamente ha permitido conocer el propósito de los participantes en las diferentes acciones del proyecto.

Asumimos que las NIAs constituyen el punto de partida para comprender problemas, formas de gobierno, manejo del poder, etc.

**9) *Perspectivas para la comprensión de sistemas***, esta herramienta ha sido de utilidad para analizar la estructura y dinámica de los sistemas. Así podemos comprender y expresar la dinámica de sistemas a través de: 1) patrón de organización, 2) estructura, 3) proceso y 4) significado.

**10) *Momentos de Intervención***, para cambiar o solucionar problemas de los sistemas, tiene la capacidad de organizar la intervención y garantizar la intervención científica y disciplinada, al mismo tiempo exige aportes y decisiones creativas e inteligentes y promueve el aprendizaje en la práctica.

**11) *Tríada del Conocimiento***, si bien es parte esencial de la base de conocimiento, su uso aislado ha sido permanente porque constituye la esencia del desarrollo del conocimiento. Esta herramienta promueve cuestionarse y cambiar valores, pensamientos y creencias.

## **3.2 Requerimientos**

### **3.2.1 Especificación de requisitos de software**

#### **3.2.1.1 Propósito**

El propósito de la especificación de requisitos de software es el de comunicar de manera precisa las funciones y capacidades que se va a proveer a la Fundación HABITierra, además de las restricciones que se respetarán en el desarrollo del sistema, la estimación previa del tamaño, costo y tiempo, para llevar así de manera correcta la planificación del proyecto. Además imponer una base para evaluación del producto final y para el control de las modificaciones en requerimientos que puedan ser cambiantes.

La audiencia a la que está destinada esta especificación engloba:

- Usuarios.
- Analistas de sistemas y de requerimientos.

- Desarrolladores: diseñadores, programadores.
- Equipo de capacitación de usuario
- Jefe del Proyecto

### 3.2.1.2 Alcance del Sistema

El alcance de la herramienta web para planificar, gestionar y gobernar territorios implica el entendimiento y adopción de la modelación teórica de Ge@ para poder construir el modelo digital que ahora estará orientado a la web. En cuanto a los modelos de aplicación que se desarrollarán, nos centraremos en el Ordenamiento territorial y Gobernabilidad territorial.

### 3.2.1.3 Personal involucrado

Nombre	Pedro Zeas
Rol	Gerente
Categoría profesional	Arquitecto
Responsabilidades	Gerencia de la Fundación HABITierra

Nombre	Ángel Espinoza
Rol	Jefe del Departamento de Informática
Categoría profesional	Ingeniero de Sistemas
Responsabilidades	Directora

Nombre	Fernanda Pérez
Rol	Técnico
Categoría profesional	Ingeniera de Sistemas
Responsabilidades	Coordinación de proyectos Informáticos

Nombre	Cristian Vega
Rol	Técnico
Categoría profesional	Economista
Responsabilidades	Coordinación de Subsistema Económico - Territorial

Nombre	Jeaneth Ramírez
Rol	Técnico
Categoría profesional	Ingeniero Ambiental
Responsabilidades	Coordinación del Subsistema Ambiental



Nombre	Martín Carpio
Rol	Técnico
Categoría profesional	Ingeniero Ambiental
Responsabilidades	Cooperación en el Subsistema Ambiental

Nombre	Isabel Orquera
Rol	Técnico
Categoría profesional	Arquitecto
Responsabilidades	Coordinación del Subsistema Asentamientos Humanos

Nombre	Patricio Carpio
Rol	Técnico
Categoría profesional	Sociólogo
Responsabilidades	Coordinación del Subsistema Socio - Cultural

Nombre	Jorge Espinoza
Rol	Analista, Diseñador, Desarrollador
Categoría profesional	Egresado de Ingeniería de Sistemas
Responsabilidades	Encargado del análisis, diseño y desarrollo del software.

Nombre	Jorge León
Rol	Analista, Diseñador, Desarrollador
Categoría profesional	Egresado de Ingeniería de Sistemas
Responsabilidades	Encargado del análisis, diseño y desarrollo del software.

#### 3.2.1.4 Funcionalidad del sistema

El modelo digital comprenderá los siguientes bloques de trabajo:

- Control de usuarios y Roles
  - Mantenimiento (Ingreso, Actualización).
- Administración de proyectos
  - Mantenimiento (Ingreso, Actualización).
  - Bitácora de proyectos.
- Administración de instituciones
  - Datos de la institución.
  - Ubicación geográfica.

- Administración del sitio
  - Publicidad.
  - Ajuste de parámetros.
- Usar un gestor de contenidos como Joomla que contendrá información básica de la empresa y donde se integrará el sistema Ge@.
- Reportes que permitirán al administrador que sistemas, subsistemas o elementos representan alguna fortaleza o una debilidad en el proyecto que se encuentre desarrollando.

### 3.2.1.5 Características de los usuarios

Tipo de usuario	Gerente
Formación	Educación Superior
Habilidades	Experiencia en Ordenamiento Territorial y buen criterio en la toma de decisiones
Actividades	<ul style="list-style-type: none"> <li>• Supervisar proyectos</li> <li>• Interpretar Resultados</li> <li>• Brindar Asesoría</li> </ul>

Tipo de usuario	Técnico
Formación	Educación Superior
Habilidades	Experiencia en un ámbito específico que aporte a la totalidad de la gobernabilidad, gestión y planificación de Territorios
Actividades	<ul style="list-style-type: none"> <li>• Capacitación a usuarios finales</li> <li>• Alimentar información a la herramienta web</li> </ul>

Tipo de usuario	Usuario Final
Formación	N/A
Habilidades	Representar algún territorio y conocer la herramienta web
Actividades	<ul style="list-style-type: none"> <li>• Alimentar información a la herramienta web</li> </ul>

### 3.2.1.6 Restricciones

Debe contemplarse las implicaciones de los siguientes puntos críticos:

- Sistemas seguros: protección de información, seguridad en los datos, etc.

- Gestión de flujos de trabajo, seguridad de transacciones e intercambio de información
- El sistema debe diseñarse pensando en incorporar a futuro nuevos módulos como modelos de aplicación.
- Usar software libre, como es el caso del lenguaje Java para la programación y MySQL para el almacenamiento y gestión de los datos

A nivel de hardware:

- Dos puertos Gigabit Ethernet integrados
- Tecnología CISCO y 3COM en equipos de red
- Procesador Intel Core i7, 2.4 GHz
- 8 GB RAM
- Disco duro 1 TB
- Servidores con virtualización

A continuación tenemos una lista completa de los requisitos que debe cumplir la herramienta web para la planificación, gestión y gobierno de territorios.

### 3.2.1.7 Requisitos comunes de las interfaces

Los requisitos que involucran entradas y salidas del sistema son detallados a continuación:

#### 3.2.1.7.1 Interfaces de usuario

Número de requisito	RIU1
Nombre de requisito	<i>Uniformidad de Colores</i>
Tipo	Requisito
Complejidad	Baja
Prioridad del requisito	Media/Deseada
Descripción:	Todas las pantallas con las que interactúe el usuario deberán llevar una uniformidad, utilizando colores neutrales que no causen malestar a los usuarios.
Proceso:	Diseño de la interfaces del sistema
Entradas:	Datos obtenidos a partir de imágenes, símbolos que sean representativos para Ge@
Salida:	Interfaces para el usuario

Número de requisito	RIU2
Nombre de requisito	<b><i>Interfaz accesible e intuitiva</i></b>
Tipo	Requisito
Complejidad	Media
Prioridad del requisito	Alta/Esencial.
Descripción:	El manejo de las funcionalidades de la herramienta debe ser lo más intuitivo posible, de manera que sean muy claras las posibles acciones a llevar a cabo y la manera de hacerlas
Proceso:	Estudio de la estructura de los componentes que conformarán la herramienta web incorporando ayudas en línea.
Entradas:	Necesidades de cada usuario.
Salida:	Interfaces para el usuario, manuales en línea.

Número de requisito	RIU3
Nombre de requisito	<b><i>Optimización de interfaces</i></b>
Tipo	Requisito
Complejidad	Media
Prioridad del requisito	Alta/Esencial.
Descripción:	Se debe mostrar la información netamente necesaria dependiendo del usuario y del proceso que se lleve a cabo, mostrando una interfaz distinta para administración y para usuarios comunes.
Proceso:	Identificar rol de usuario
Entradas:	Datos del usuario.
Salida:	Interfaces optimizadas.

Número de requisito	RIU4
Nombre de requisito	<b><i>Coherencia en los datos</i></b>
Tipo	Requisito
Complejidad	Baja
Prioridad del requisito	Media/Deseada.
Descripción:	La información presentada debe estar dispuesta en manera correcta, tomando en cuenta la presentación de caracteres internacionales.
Proceso:	Identificar caracteres internacionales y buscar la manera de presentarlos de manera correcta.
Entradas:	Datos
Salida:	Interfaces compatibles con caracteres internacionales.

Número de requisito	RIU5
Nombre de requisito	<b><i>Consistencia del sistema entre los distintos navegadores</i></b>
Tipo	Requisito
Complejidad	Media
Prioridad del requisito	Media/Deseada.
Descripción:	El sistema web debe visualizarse y manejarse de una forma igual o de la forma más parecida posible en los navegadores mayoritarios (Firefox, Opera, Google Chrome), exceptuando internet explorer que no es compatible con el componente GWT-Graphics
Proceso:	Identificar si son necesarios componentes adicionales al navegador e instalarlos, o evitar hacer uso de dichos elementos.
Entradas:	Información de herramientas a usar, navegadores.
Salida:	Herramienta web compatible con navegadores.

Número de requisito	RIU6
Nombre de requisito	<b><i>Componentes Drag and Drop</i></b>
Tipo	Requisito
Complejidad	Media
Prioridad del requisito	Media/Deseada.
Descripción:	La herramienta web necesita que componente drag and drop
Proceso:	El usuario interactúa con los objetos del sistema ge@ (elementos, subsistemas, sistemas) y para su gestión, creación de interacciones, etcétera, necesita poder arrastrar y soltar mediante el mouse
Entradas:	-
Salida:	Herramienta Web con funcionalidades drag & drop

Número de requisito	RIU7
Nombre de requisito	<b><i>Menús contextuales</i></b>
Tipo	Requisito
Complejidad	Media
Prioridad del requisito	Media/Deseada.
Descripción:	Habilitar la funcionalidad del clic derecho
Proceso:	La herramienta web, usa un ambiente de escritorio, y tiene la necesidad de que el usuario pueda acceder a

	menús contextuales
Entradas:	-
Salida:	Menús contextuales en la herramienta web

### 3.2.1.7.2 Interfaces de Hardware

Número de requisito	RIH1
Nombre de requisito	<i>Acceso a la Web</i>
Tipo	Requisito
Complejidad	Media
Prioridad del requisito	Alta/Esencial.
Descripción:	Los usuarios deben contar con todos los dispositivos de comunicación para el acceso a internet.
Proceso:	-
Entradas:	Equipos o infraestructura para acceder a internet
Salida:	Acceso a la web de todos las personas involucradas con el manejo del sistema

Número de requisito	RIH2
Nombre de requisito	<i>Acceso a los datos</i>
Tipo	Requisito
Complejidad	Media
Prioridad del requisito	Alta/Esencial.
Descripción:	Instalación de un servidor central que contendrá el sistema y los datos
Proceso:	Subir el sistema e instalar la base de datos en el servidor para que se pueda acceder mediante internet.
Entradas:	-
Salida:	-

### 3.2.1.7.3 Requisitos Funcionales

Número de requisito	RF1
Nombre de requisito	<i>Mantenimiento de Usuarios Invitados</i>
Tipo	Requisito
Fuente del requisito	Administrador, usuario común
Prioridad del requisito	Media/Deseada
Proceso:	<p>a. El usuario que no tiene privilegios de administración está en capacidad de crear, modificar y suspender su propia cuenta en la herramienta Web</p> <p>b. El usuario administrador tiene la potestad de crear, modificar, dar de baja o activar a usuarios</p>

	suspendidos. Puede crear usuarios administradores o comunes.
Entradas:	Datos del usuario
Salida:	Datos del usuario almacenados, modificados o puestos en estado de inactivos en la base de datos

Número de requisito	RF2
Nombre de requisito	<b><i>Mantenimiento de Usuarios Administradores</i></b>
Tipo	Requisito
Fuente del requisito	Administrador
Prioridad del requisito	Media/Deseada
Proceso:	Un usuario Administrador puede ingresar, modificar o eliminar cuentas de otro usuario Administrador o sus propios datos.
Entradas:	Datos del usuario administrador
Salida:	Datos del usuario administrador almacenados, modificados o puestos en estado de inactivos en la base de datos

Número de requisito	RF3
Nombre de requisito	<b><i>Mantenimiento de tipo de proyectos</i></b>
Tipo	Requisito
Fuente del requisito	Administrador
Prioridad del requisito	Media/Deseada
Proceso:	El usuario administrador puede habilitar, modificar, ingresar, eliminar tipos de proyecto, recordando que nos estamos refiriendo a los modelos de aplicación de ge@.
Entradas:	Datos del tipo de proyecto (modelo de aplicación)
Salida:	Tipo de proyecto ingresado, modificado o en estado de inactivo almacenado en la base de datos

Número de requisito	RF4
Nombre de requisito	<b><i>Mantenimiento de Proyectos</i></b>
Tipo	Requisito
Fuente del requisito	Administrador, usuario común
Prioridad del requisito	Media/Deseada
Proceso:	<p>a. El administrador puede ingresar, modificar, eliminar un proyecto, inclusive puede compartir proyectos entre diferentes usuarios</p> <p>b. El usuario común tiene la capacidad de modificar, eliminar un proyecto sí y sólo sí él lo creó.</p>

	c. El usuario común, puede ingresar nuevos proyectos e ingresar a proyectos que le han sido compartidos
Entradas:	Datos del proyecto
Salida:	Proyecto ingresado, modificado, eliminado o compartido con otros usuarios almacenado en la base de datos

Número de requisito	RF4.1
Nombre de requisito	<b><i>Mantenimiento de datos de mapa</i></b>
Tipo	Requisito
Fuente del requisito	Invitado, Administrador
Prioridad del requisito	Baja/Opcional
Proceso:	<ul style="list-style-type: none"> <li>• El administrador o usuario tiene la posibilidad de ingresar o modificar puntos que indiquen el lugar en donde se esté llevando a cabo el proyecto, este registro es opcional.</li> <li>• El usuario que no es propietario del proyecto no puede realizar esta operación.</li> </ul>
Entradas:	Datos de ubicación de marcadores.
Salida:	Ubicación de marcador en mapa.

Número de requisito	RF4.2
Nombre de requisito	<b><i>Compartir proyectos</i></b>
Tipo	Requisito
Fuente del requisito	Invitado, Administrador
Prioridad del requisito	Media/Deseada
Proceso:	<ul style="list-style-type: none"> <li>• El administrador o invitado puede como no compartir un proyecto con otros usuarios invitados o administradores de una misma u otra institución.</li> <li>• Aquellos usuarios a los cuales se les ha compartido un proyecto anteriormente no pueden realizar esta operación.</li> </ul>
Entradas:	Datos del tipo de institución, de la institución y de usuarios
Salida:	Proyectos compartidos con usuarios.



Número de requisito	RF5
Nombre de requisito	<b><i>Mantenimiento de Tipo de Instituciones</i></b>
Tipo	Requisito
Fuente del requisito	Administrador
Prioridad del requisito	Alta/Esencial
Proceso:	El administrador puede ingresar, modificar y desactivar tipos de instituciones, como se describió en secciones anteriores estas pueden ser públicas o privadas. Estas están relacionadas directamente con el mantenimiento de instituciones.
Entradas:	Datos del tipo de institución
Salida:	Registro, modificación o desactivación del tipo de institución.

Número de requisito	RF6
Nombre de requisito	<b><i>Mantenimiento de Instituciones</i></b>
Tipo	Requisito
Fuente del requisito	Administrador
Prioridad del requisito	Alta/Esencial
Proceso:	El administrador puede registrar, modificar o desactivar una institución, cabe recalcar que se encuentra directamente ligada con el usuario.
Entradas:	Datos de la institución.
Salida:	Institución modificada, desactivada o registrada como un nuevo ítem.

Número de requisito	RF7
Nombre de requisito	<b><i>Mantenimiento de administración</i></b>
Tipo	Requisito
Fuente del requisito	Administrador
Prioridad del requisito	Media/Deseada
Proceso:	El administrador puede editar un mensaje de publicidad así como desactivarlo o activarlo.
Entradas:	Mensaje de publicidad
Salida:	Visualización en la interfaz tanto del administrador como usuario.

Los siguientes requisitos tratan sobre Ge@, considerándolo como una herramienta web; como se describió en la sección anterior inicialmente Ge@ se conforma de 9 elementos, los cuales por lo general llevan la realidad a un modelamiento de objetos dentro de objetos con sus respectivas relaciones y causas-razones, a partir de ellos surgen los siguientes requisitos:

Número de requisito	RF8
Nombre de requisito	<b><i>Mantenimiento de objetos</i></b>
Tipo	Requisito
Fuente del requisito	Administrador o Invitado
Prioridad del requisito	Alta/Esencial
Proceso:	El administrador o invitado puede modelar la realidad construyendo objetos que son óvalos que representan principios, dimensiones, elementos, etc. A estos objetos los puede modificar su tamaño, color y el texto interno, así como también de ser posible su eliminación.
Entradas:	Datos del ovalo.
Salida:	Visualización del objeto.

Número de requisito	RF9
Nombre de requisito	<b><i>Mantenimiento de interacciones</i></b>
Tipo	Requisito
Fuente del requisito	Administrador o Invitado
Prioridad del requisito	Alta/Esencial
Proceso:	El administrador o invitado pueden crear, modificar o eliminar las relaciones que existen entre los objetos. Cada relación representa un valor el cual me dice si la interacción ente estos dos elementos esta Fuerte, Media, Débil o Nula. Junto con otro valor que me indica si esta Excelente, Bien, Regular, Malo o Pésimo. Sumado el valor de todas las relaciones este indicara el estado de un conjunto de objetos los cuales representan un sistema.
Entradas:	Objeto Inicial, Objeto Final, ¿Cómo está?, ¿Cómo debería ser?
Salida:	Interacción entre dos objetos

Número de requisito	RF10.1
Nombre de requisito	<b><i>Mantenimiento de interacciones internas</i></b>
Tipo	Requisito
Fuente del requisito	Administrador o Invitado
Prioridad del requisito	Alta/Esencial
Proceso:	El administrador o invitado puede crear, modificar o eliminar una interacción interna, se considera interna cuando este se genera con objetos que se encuentran en un mismo nivel conformando un mismo sistema.
Entradas:	Objeto Inicial, Objeto Final, ¿Cómo está?, ¿Cómo debería ser?
Salida:	Interacción entre dos objetos del mismo sistema

Número de requisito	RF10.2
Nombre de requisito	<b><i>Mantenimiento de interacciones externas</i></b>
Tipo	Requisito
Fuente del requisito	Administrador o Invitado
Prioridad del requisito	Alta/Esencial
Proceso:	El administrador o invitado puede crear, modificar o eliminar una interacción externa, se considera externa cuando este se genera con objetos que se encuentran fuera del mismo nivel en el que se encuentran los objetos de otros sistemas.
Entradas:	Objeto Inicial, Objeto Final, ¿Cómo está?, ¿Cómo debería ser?
Salida:	Interacción entre dos objetos que no se encuentran en un mismo sistema.

Número de requisito	RF11
Nombre de requisito	<b><i>Subir y bajar archivos</i></b>
Tipo	Requisito
Fuente del requisito	Administrador o Invitado
Prioridad del requisito	Media/Deseada
Proceso:	Toda descripción que se detalla dentro de Ge@ ya sea una descripción de un objeto, especificación de variables indicadores, de los procesos de momentos de intervención, T-M-H, en general a todo elemento que se le pueda añadir texto, es posible que para reforzar esto se puedan adjuntar archivos o imágenes y que sea posible la descarga del mismo.
Entradas:	Archivo o Imagen en cualquier formato.

Salida:	Archivo o Imagen subido/descargado al/desde servidor, adjunto a cualquier elemento de Ge@.
---------	--

Número de requisito	RF12
Nombre de requisito	<b><i>Mantenimiento Causas y Razones</i></b>
Tipo	Requisito
Fuente del requisito	Administrador o Invitado
Prioridad del requisito	Alta/Esencial
Proceso:	Cada interacción que exista entre los objetos tiene una causa y una razón de su creación, es por eso que se le considera como un requisito en el cual se establezca, modifique o elimine las causas y razones. Cada Causa y Razón está conformado por una prioridad la cual indica un orden a seguir en caso de haber un listado de las mismas.
Entradas:	Descripción y prioridad de la causa o razón
Salida:	Creación, modificación o eliminación de la causa o razón de una determinada interacción.

Número de requisito	RF13
Nombre de requisito	<b><i>Mantenimiento de Teoría-Métodos-Herramientas</i></b>
Tipo	Requisito
Fuente del requisito	Administrador o Invitado
Prioridad del requisito	Media/Deseada
Proceso:	Cada objeto que se cree dentro de los principios sistémicos o esenciales deben de tener su T-M-H que son nada más que una descripción. A esta descripción se la puede crear, modificar o eliminar.
Entradas:	Información sobre Teorías, Métodos, Herramientas. Indicar el objeto a añadir la T-M-H.
Salida:	Registro, modificación o eliminación de los datos de TMH.

Número de requisito	RF14
Nombre de requisito	<b><i>Mantenimiento de Variables e Indicadores</i></b>
Tipo	Requisito
Fuente del requisito	Administrador o Invitado
Prioridad del requisito	Media/Deseada
Proceso:	Todo objeto es decir ya sea un principio, ejercicio de interacciones, modelización, prototipos, etc. tienen la posibilidad de agregar, modificar o eliminar sus variables e indicadores.

Entradas:	Información sobre sus variables e indicadores, indicar el objeto al cual se le añade las variables e indicadores.
Salida:	Registro, modificación o eliminación de los datos que conforman las variables e indicadores.

Número de requisito	RF15
Nombre de requisito	<b><i>Mantenimiento de Base de Conocimiento Capitalizada</i></b>
Tipo	Requisito
Fuente del requisito	Administrador o Invitado
Prioridad del requisito	Media/Deseada
Proceso:	De igual manera que con TMH o VI, el administrador o invitado puede crear, modificar o eliminar la base de conocimiento capitalizada con la que dispone Ge@ por cada objeto que se encuentre creado dentro del sistema, está conformado por 8 elementos que son: 1) Significados de Origen, 2) Significados de Local, 3) Propósito, 4) Teoría, 5) Método, 6) Herramienta, 7) Interacciones Clave, 8) Lo Opuesto y complementario: La Dualidad.  Cada elemento es una simple descripción textual.
Entradas:	Información sobre los 8 elementos del BCC, objeto a crear su BCC.
Salida:	Registro, modificación o eliminación de los datos de la BCC.

Número de requisito	RF16
Nombre de requisito	<b><i>Mantenimiento de Momentos de Intervención</i></b>
Tipo	Requisito
Fuente del requisito	Administrador o Invitado
Prioridad del requisito	Media/Deseada
Proceso:	El administrador o invitado puede crear, modificar o eliminar la información sobre los momentos de intervención que dispone Ge@, cada momento de intervención es una descripción sobre un objeto.
Entradas:	Información sobre los Momentos de Intervención, objeto al cual se le añade.
Salida:	Registro, modificación o eliminación de la información de los momentos de intervención.

Número de requisito	RF16.1
Nombre de requisito	<b><i>Mantenimiento de Procesos de Momentos de Intervención</i></b>
Tipo	Requisito
Fuente del requisito	Administrador o Invitado
Prioridad del requisito	Media/Deseada
Proceso:	El administrador o invitado puede crear, modificar o eliminar información sobre los procesos que se encuentran incluidos por cada objeto dentro de momentos de intervención que dispone Ge@, la descripción es similar a los objetos presentes dentro momentos de intervención.
Entradas:	Información sobre los Procesos dentro Momentos de Intervención, objeto al cual se le añade.
Salida:	Registro, modificación o eliminación de la información de procesos de momentos de intervención.

Número de requisito	RF17
Nombre de requisito	<b><i>Generar Estado del Objeto(Sistema, subsistema, elemento)</i></b>
Tipo	Requisito
Fuente del requisito	Administrador, usuario común
Prioridad del requisito	Alta/Esencial
Proceso:	El usuario necesita conocer el estado del objeto que está modelando, conocer cómo están las interacciones(tanto internas como externas) y los elementos en el nivel actual, mediante valores representativos como totales y promedios
Entradas:	Nivel actual, sistema actual, interacciones del sistema
Salida:	Resumen numérico, estadística de cómo está el sistema en el nivel actual.

Número de requisito	RF18
Nombre de requisito	<b><i>Generar Sumas del sistema</i></b>
Tipo	Requisito
Fuente del requisito	Administrador
Prioridad del requisito	Alta/Esencial
Proceso:	El usuario Administrador tiene la capacidad de

	revisar de manera gráfica cómo se encuentra el sistema, mediante un gráfico de barras que proporciona información de las interacciones entre los diferentes objetos que se comunican e interactúan en el nivel actual
Entradas:	Nivel actual, sistema actual, interacciones del sistema
Salida:	Resumen estadístico mediante gráfico de barras, de los objetos (sistemas, subsistemas, elementos) en el nivel actual.

Número de requisito	RF19
Nombre de requisito	<b><i>Generar Reportes del sistema</i></b>
Tipo	Requisito
Fuente del requisito	Administrador
Prioridad del requisito	Alta/Esencial
Proceso:	El usuario Administrador no sólo necesita los datos en la herramienta web, también necesita descargarlos o imprimirlos para tener un respaldo físico para sus análisis, es por esto que necesita exportar los reportes con información estadística mediante gráficos de los sistemas que está modelando.
Entradas:	Identificador del sistema/ subsistema que está modelando, datos del usuario
Salida:	Resumen estadístico mediante gráfico de barras, de los objetos(sistemas, subsistemas, elementos) en el nivel actual.

Número de requisito	RF20
Nombre de requisito	<b><i>Mantenimiento de Prototipos</i></b>
Tipo	Requisito
Fuente del requisito	Administrador, Usuario común
Prioridad del requisito	Media/Deseada
Proceso:	El usuario puede modificar un prototipo existente o crear un nuevo prototipo en su proyecto.
Entradas:	Datos del prototipo
Salida:	Datos del prototipo nuevo o modificado ingresados

	en la base de datos
Número de requisito	RF20.1
Nombre de requisito	<b><i>Mantenimiento de Variables de Prototipos</i></b>
Tipo	Requisito
Fuente del requisito	Administrador, Usuario común
Prioridad del requisito	Media/Deseada
Proceso:	El usuario puede ingresar, modifica o eliminar variables al diagrama sistémico
Entradas:	Datos de la variable
Salida:	Variable de prototipos ingresada, modificada o eliminada de la base de datos

Número de requisito	RF20.2
Nombre de requisito	<b><i>Mantenimiento de Procesos de Prototipos</i></b>
Tipo	Requisito
Fuente del requisito	Administrador, Usuario común
Prioridad del requisito	Media/Deseada
Proceso:	Ingresar, modificar o eliminar procesos de solución dentro de los 5 grupos de soluciones (intuir-idear, aprender-comprender, crear-proponer, implementar-evaluar, capitalizar-realimentar)
Entradas:	Nombre del proceso y duración del mismo
Salida:	Datos del Proceso de solución nuevo, modificado o eliminando almacenados en la base de datos

Número de requisito	RF21
Nombre de requisito	<b><i>Mantenimiento Realimentación</i></b>
Tipo	Requisito
Fuente del requisito	Administrador, Usuario común
Prioridad del requisito	Media/Deseada
Proceso:	En el diagrama sistémico de las variables se puede especificar cuál es el efecto de una variable sobre otra, pudiendo especificar si este es ayuda a su crecimiento o tiene un efecto compensador, mediante signos + y -. El usuario necesita colocar uno de estos signos y especificar si este efecto produce una demora en la



	otra variable mediante la colocación de un reloj.
Entradas:	VARIABLES QUE INTERACTÚAN
Salida:	REALIMENTACIÓN POSITIVA, NEGATIVA O CON DEMORA ALMACENADA EN LA BASE DE DATOS

Número de requisito	RF22
Nombre de requisito	<b><i>Mantenimiento de la Base de Conocimiento Capitalizada</i></b>
Tipo	Requisito
Fuente del requisito	Administrador, Usuario común
Prioridad del requisito	Media/Deseada
Proceso:	El usuario puede ingresar, modificar los significados de origen, significados de local, propósito, teorías, métodos, herramientas, interacciones claves, dualidad correspondientes a los elementos de los tres bloques de la base de conocimiento (gestionar, gobernar, planificar)
Entradas:	Datos de significados de origen, significados de local, propósito, teorías, métodos, herramientas, interacciones claves, dualidad
Salida:	Datos de la base de conocimientos ingresados o modificados en la base de datos.

Número de requisito	RF22.1
Nombre de requisito	<b><i>Importar base de conocimiento capitalizada</i></b>
Tipo	Requisito
Fuente del requisito	Administrador, Usuario común
Prioridad del requisito	Media/Deseada
Proceso:	Ciertos elementos de los prototipos pueden tener vínculo con elementos de la modelización digital, por ende pueden compartir los mismos conceptos de la base de conocimientos, es por ello que el usuario tiene la necesidad de importar la base de conocimientos de un sistema a otro.
Entradas:	Base de conocimiento a importar
Salida:	Base de conocimiento importada

Número de requisito	RF22.2
Nombre de requisito	<b><i>Mantenimiento caja de herramientas</i></b>
Tipo	Requisito
Fuente del requisito	Administrador, Usuario común
Prioridad del requisito	Media/Deseada
Proceso:	El usuario puede crear, modificar o eliminar ítems de la caja de herramientas
Entradas:	Datos de ítem de la caja de herramientas
Salida:	Ítem nuevo, modificado o desactivado de la caja de herramientas almacenado en la base de datos

### 3.2.1.7.3 Requisitos no funcionales

#### 3.2.1.7.3.1 Requisitos de Rendimiento

Número de requisito	RNF1
Nombre de requisito	<b><i>Tiempo de respuesta</i></b>
Tipo	Requisito
Fuente del requisito	Administrador, Invitado
Prioridad del requisito	Alta/Esencial
Descripción:	El tiempo de respuesta de la aplicación a cada función solicitada por el usuario no debe ser superior a los 10 segundos.

Número de requisito	RNF2
Nombre de requisito	<b><i>Número de terminales</i></b>
Tipo	Requisito
Fuente del requisito	Administrador
Prioridad del requisito	Baja/Opcional
Descripción:	El número de terminales no es fijo, puesto que por cada proyecto que se ejecute el número de usuarios puede variar ya que el sistema puede ser accedido de la web, resulta difícil determinar exactamente el número.

Número de requisito	RNF3
Nombre de requisito	<b><i>Usuarios Simultáneos</i></b>
Tipo	Restricción
Fuente del requisito	Administrador
Prioridad del requisito	Alta/Esencial
Descripción:	El sistema debe permitir acceso simultáneo, puesto que es usado por varios usuarios a través

	de la Web. El número estimado de usuarios simultáneos, en horas de alto trabajo es entre 15 y 30 personas.
--	--

Número de requisito	RNF4
Nombre de requisito	<b><i>Cantidad de registros almacenados</i></b>
Tipo	Restricción
Fuente del requisito	Ingeniera de software, Analistas
Prioridad del requisito	Media/Deseada
Descripción:	El almacenamiento de datos será abundante, los registros almacenados estarán en el orden de los miles y en caso de ser usado por muchos años podría llegar a los millones.

### 3.2.1.7.3.2 Requisitos de Seguridad

Número de requisito	RS1
Nombre de requisito	<b><i>Roles de usuario</i></b>
Tipo	Restricción
Fuente del requisito	Administrador
Prioridad del requisito	Alta/Esencial
Descripción	La seguridad que debe ser implementada en el Sistema debe manejar roles y privilegios de usuario.

Número de requisito	RS2
Nombre de requisito	<b><i>Seguridad en las transacciones</i></b>
Tipo	Restricción
Fuente del requisito	Administrador
Prioridad del requisito	Alta/Esencial
Descripción	Los datos que maneja Ge@ son muy importantes, por lo que hay que evitar al máximo pérdidas e intromisiones entre usuarios que no comparten proyectos.

Número de requisito	RS2.1
Nombre de requisito	<b><i>Encriptación de datos</i></b>
Tipo	Restricción
Fuente del requisito	Administrador
Prioridad del requisito	Alta/Esencial
Descripción	Los datos viajan a través de la red, por ende deben poseer políticas de seguridad como la encriptación para que personas ajenas al sistema no hagan mal uso de estos datos

Número de requisito	RS3
Nombre de requisito	<i>Comunicación entre módulos</i>
Tipo	Requerimiento
Fuente del requisito	Administrador
Prioridad del requisito	Media/Deseada
Descripción	Se debe distinguir qué módulos pueden interactuar entre sí y que módulos son independientes y que usuarios pueden manejarlos.

Número de requisito	RS5
Nombre de requisito	<i>Respaldo de los datos</i>
Tipo	Requerimiento
Fuente del requisito	Administrador
Prioridad del requisito	Alta/Esencial
Descripción	Debido a la importancia de los datos, los mismos deben tener respaldados.

### 3.2.1.7.3.3 Requisitos de Fiabilidad

Número de requisito	RS2
Nombre de requisito	<i>Incidentes permisibles</i>
Tipo	Restricción
Fuente del requisito	Administrador
Prioridad del requisito	Alta/Esencial
Descripción	Es difícil determinar cuándo exactamente va a fallar el sistema, pero por lo general esto se debe a problemas de conexión con el servidor y acceso a la base de datos, se podría decir que el número de fallos así como la frecuencia son bajos.

### 3.2.1.7.3.4 Requisitos de Disponibilidad

Número de requisito	RD1
Nombre de requisito	<i>Disponibilidad del sistema</i>
Tipo	Requisito
Fuente del requisito	Administrador, Invitado
Prioridad del requisito	Media/Deseable
Descripción	El sistema al estar orientado a la web, implica que puede estar disponible a cualquier momento, por ende pensamos que el sistema debe estar utilizable el 95% del tiempo.

### 3.2.1.7.3.5 Requisitos de Mantenibilidad

Número de requisito	RM1
Nombre de requisito	<b><i>Mantenimiento de Roles de Usuario</i></b>
Tipo	Requisito
Fuente del requisito	Administración
Prioridad del requisito	Alta/Esencial
Descripción	El sistema debe permitir que el administrador tenga la posibilidad de modificar los roles de usuario y cambios de permiso.

Número de requisito	RM3
Nombre de requisito	<b><i>Mantenimiento de módulos del sistema</i></b>
Tipo	Requisito
Fuente del requisito	Desarrolladores
Prioridad del requisito	Alta/Esencial
Descripción	El sistema debe permitir que el administrador tenga la posibilidad de agregar nuevos módulos al sistema.

Número de requisito	RM4
Nombre de requisito	<b><i>Mantenimiento de los datos</i></b>
Tipo	Requisito
Fuente del requisito	Administrador
Prioridad del requisito	Alta/Esencial
Descripción	El administrador debe realizar un respaldo, verificar la consistencia, integridad de los datos periódicamente.

Número de requisito	RM5
Nombre de requisito	<b><i>Mantenimiento del sistema</i></b>
Tipo	Requisito
Fuente del requisito	Desarrolladores
Prioridad del requisito	Alta/Esencial
Descripción	Cambios en el código del sistema y tecnología.

**3.2.1.7.3.6 Requisitos de Portabilidad**

Número de requisito	RP1
Nombre de requisito	<i>Portabilidad del sistema</i>
Tipo	Restricción
Fuente del requisito	Administrador
Prioridad del requisito	Alta/Esencial
Descripción	El sistema es portable en cuanto a los navegadores, pero la mayoría de componentes dependen del servidor, un cambio de este implicaría instalar bases de datos, lenguajes de programación, servidores web, plataformas de desarrollo.

### 3.2.2 Casos de uso del sistema

En la siguiente sección se va a explicar los casos de uso que se presentan dentro de la herramienta web para la planificación, gestión y gobernabilidad territorial.

#### 3.2.2.1 Gestión de usuarios

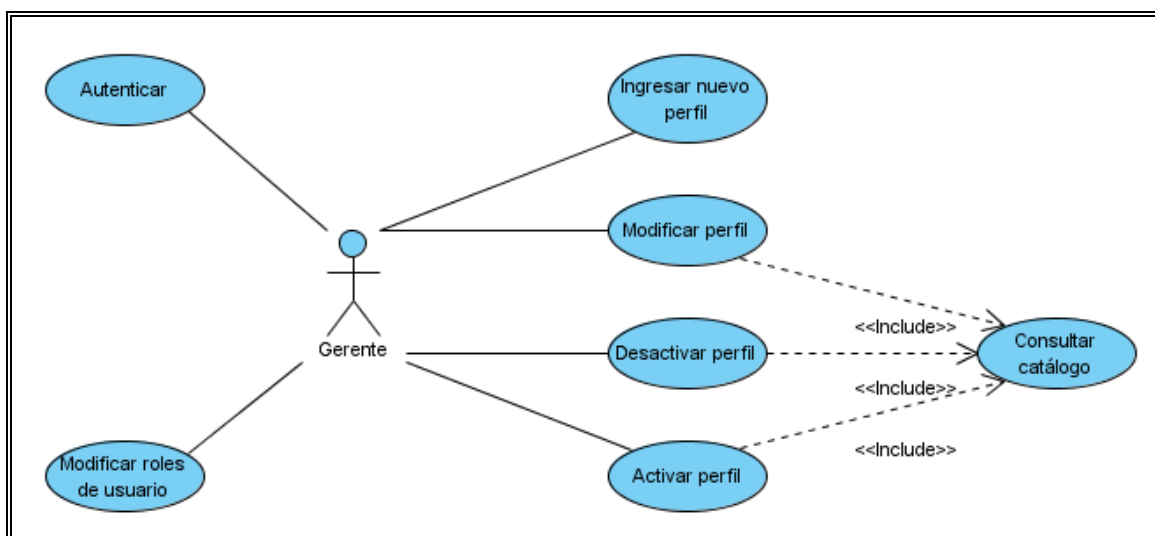


Figura 3-19: Mantenimiento de usuario administrador (Casos de uso)

#### Caso de uso CU1: Ingresar nuevo perfil

##### **Descripción:**

El caso de uso lo ejecuta el GERENTE. Es utilizado cuando un nuevo usuario se incorpora al sistema y desea registrarse por primera vez.

##### **Precondiciones:**

El gerente tiene acceso al sistema y sabe su uso.

##### **Poscondiciones:**

Una vez que el Gerente haya creado el nuevo perfil este quedará almacenado en la base de datos.

##### **Flujo Básico:**

1. Gerente hace Login correctamente
2. El gerente entra a la administración del sistema.

3. El gerente accede al módulo de usuarios.
4. El gerente ingresa los datos del nuevo usuario.
5. Determina el rol de usuario.
6. Los datos se almacenan en la base de datos.

#### **Caso de uso CU2: Modificar perfil**

##### **Descripción:**

El caso de uso lo ejecuta el GERENTE. Es utilizado cuando se desea modificar los datos generales del perfil del administrador o de algún otro usuario ya sea este otro Gerente, Técnico o Usuario común.

##### **Precondiciones:**

El gerente tiene acceso al sistema y sabe su uso.

##### **Poscondiciones:**

Una vez que el Gerente haya modificado el perfil este quedará almacenado en la base de datos.

##### **Flujo Básico:**

1. Gerente hace Login correctamente
2. El gerente entra a la administración del sistema.
3. El gerente accede al módulo de usuarios.
4. El gerente busca al usuario a modificar los datos.
5. Modifica los datos.
6. Los datos modificados se guardan.

#### **Caso de uso CU3: Desactivar perfil**

##### **Descripción:**

El caso de uso lo ejecuta el GERENTE. Es utilizado cuando se desea bloquear el acceso a la herramienta web, a un usuario determinado.

##### **Precondiciones:**



El gerente tiene acceso al sistema y sabe su uso.

**Poscondiciones:**

Una vez que el Gerente haya desactivado al usuario, este no podrá entrar al sistema

**Flujo Básico:**

1. Gerente hace Login correctamente
2. El gerente entra a la administración del sistema.
3. El gerente accede al módulo de usuarios.
4. El gerente busca al usuario a desactivar.
5. Se desactiva al usuario.

**Caso de uso CU4: Activar perfil**

**Descripción:**

El caso de uso lo ejecuta el GERENTE. Cuando un usuario ha desactivado su cuenta o fue el GERENTE quien lo efectuó, entonces es únicamente el GERENTE la persona que puede activarlo nuevamente.

**Precondiciones:**

El gerente tiene acceso al sistema y sabe su uso.

**Poscondiciones:**

El gerente va a poder acceder al perfil.

**Flujo Básico:**

1. Gerente hace Login correctamente
2. El gerente entra a la administración del sistema.
3. El gerente accede al módulo de usuarios.
4. El gerente busca al usuario para activarlo.
5. Activa la cuenta de usuario.

**Caso de uso CU5: Autenticar****Descripción:**

El caso de uso lo ejecuta el GERENTE. Al momento de ingresar al sistema, se solicita que el usuario se autentique para mantener una integridad y seguridad en los datos.

**Precondiciones:**

El gerente tiene acceso al sistema y sabe su uso, y conoce sus datos de login.

**Poscondiciones:**

El gerente accede a su perfil.

**Flujo Básico:**

1. El Gerente ingresa el nombre de usuario.
2. El Gerente ingresa la contraseña.
3. Si los datos son correctos el usuario ingresa al perfil.
4. En caso de tener un error, este dependerá si la cuenta esta desactivada o si los datos ingresados son correctos.
5. En caso de que todo resultado correcto, el gerente ingresa al sistema.

**Caso de uso CU6: Consultar catálogo****Descripción:**

El caso de uso lo ejecuta el GERENTE. Este es un caso de uso necesario ya que cualquier modificación, o activación/desactivación implica una búsqueda en el catálogo de usuarios para seleccionar al usuario correcto.

**Precondiciones:**

El gerente debe disponer de un listado de todos los usuarios.

**Poscondiciones:**

Ninguna.

**Flujo Básico:**

1. Gerente hace Login correctamente

2. El gerente entra a la administración del sistema.
3. El gerente accede al módulo de usuarios.
4. Se lista el conjunto de usuarios registrado en el sistema y se realiza una búsqueda de acuerdo al usuario al que se le desea ejecutar una operación.

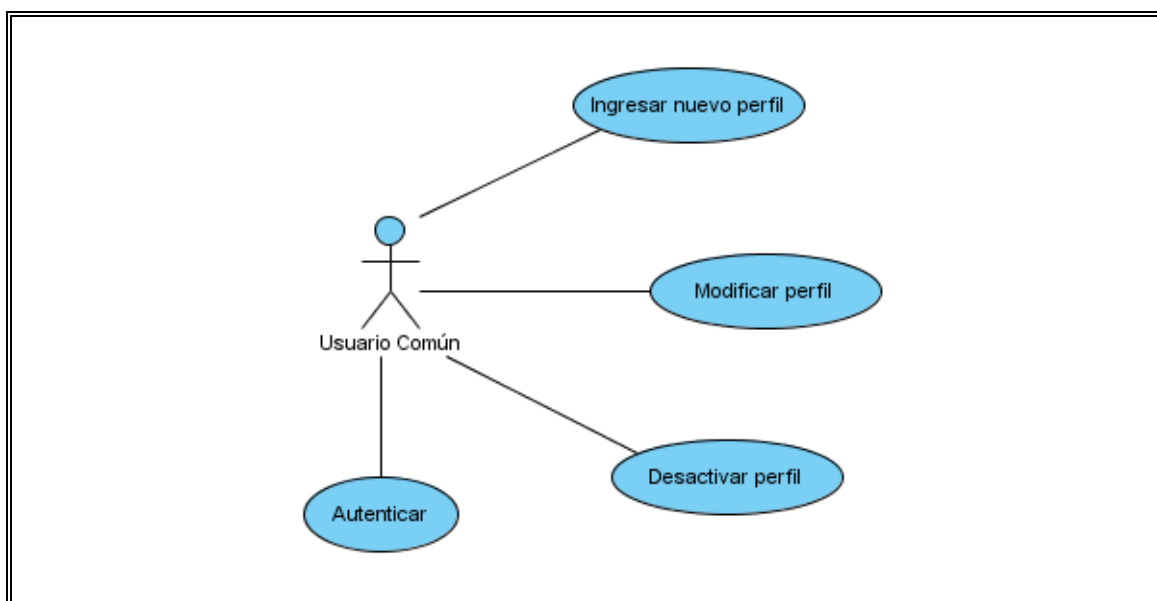


Figura 3-20: Mantenimiento de usuario común (Casos de uso)

#### **Caso de uso CU7: Ingresar nuevo perfil**

##### **Descripción:**

El caso de uso lo ejecuta el USUARIO COMÚN. Es utilizado cuando un nuevo usuario se incorpora al sistema y desea registrarse por primera vez.

##### **Precondiciones:**

El usuario común debe ingresar a la página inicial del sistema para registrarse.

##### **Poscondiciones:**

Una vez que el usuario común haya creado su cuenta este quedará almacenado en la base de datos.

##### **Flujo Básico:**

1. El usuario común ingresa al sitio web.
2. Ingresa a la página de registro de usuario.
3. Ingresa sus datos de cuenta necesarios para el registro.
4. El gerente ingresa los datos del nuevo usuario.
5. Los datos se almacenan en la base de datos con el rol de usuario invitado.

#### **Caso de uso CU8: Modificar perfil**

##### **Descripción:**

El caso de uso lo ejecuta el USUARIO COMÚN y es cuando desea editar los datos de su cuenta.

##### **Precondiciones:**

El usuario común tiene acceso al sistema y conoce su uso.

##### **Poscondiciones:**

Una vez que el usuario común haya modificado el perfil este quedará almacenado en la base de datos.

##### **Flujo Básico:**

1. El usuario común hace Login correctamente
2. Entra a editar su perfil.
3. Modifica los campos necesarios.
4. Guarda los cambios.

#### **Caso de uso CU9: Desactivar perfil**

##### **Descripción:**

El caso de uso lo ejecuta el USUARIO COMÚN cuando desea cerrar su cuenta. La desactivación no indica una eliminación permanente del perfil.

##### **Precondiciones:**

El usuario común tiene acceso al sistema y sabe su uso.

**Poscondiciones:**

Una vez que el usuario común haya desactivado su cuenta, este no podrá entrar al sistema

**Flujo Básico:**

1. El usuario común hace Login correctamente
2. Desactiva su cuenta.
3. Decide si desea o no cerrar su cuenta.
4. En caso de optar por cerrar su cuenta, este cambio se refleja en la base de datos.
5. El usuario común ya no puede acceder a su cuenta.

**Caso de uso CU10: Autenticar****Descripción:**

El caso de uso lo ejecuta el USUARIO COMÚN cuando desea entrar a operar en el sistema.

**Precondiciones:**

El usuario común tiene acceso al sistema, sabe su uso y debe conocer sus datos de login.

**Poscondiciones:**

Ninguno.

**Flujo Básico:**

1. El usuario común ingresa el nombre de usuario.
2. El usuario común ingresa la contraseña.
3. Si los datos son correctos el usuario ingresa al perfil.
4. En caso de tener un error, este dependerá si la cuenta esta desactivada o si los datos ingresados son correctos.
5. En caso de que todo resulto correcto, el usuario común ingresa al sistema.

### 3.2.2.2 Gestión de proyectos

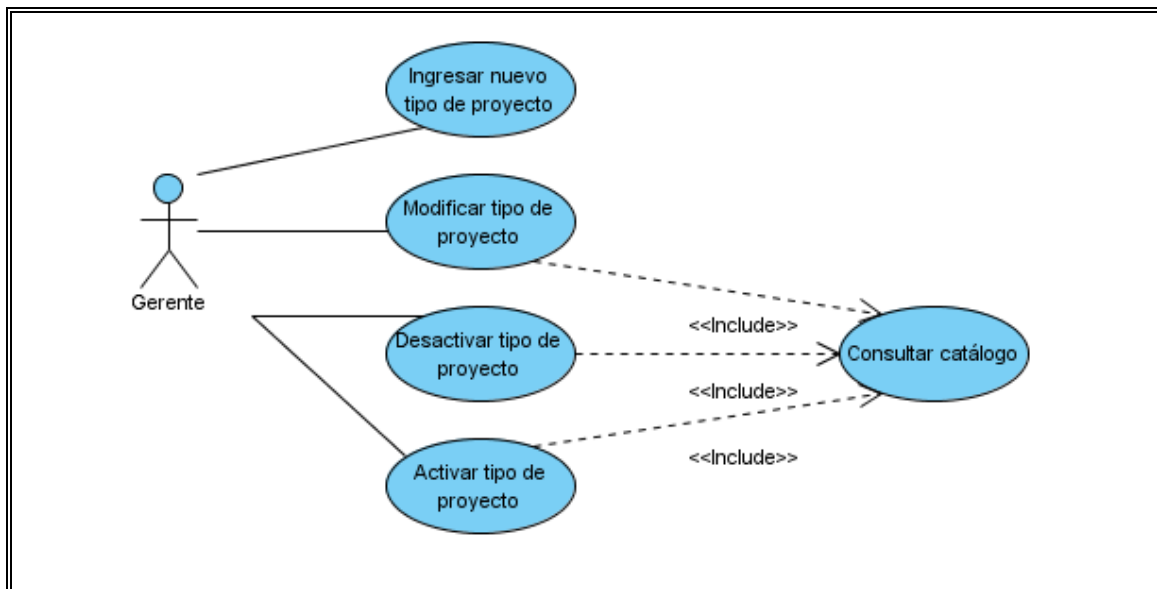


Figura 3-21. Mantenimiento de tipos de proyectos (Casos de Uso)

#### Caso de uso CU11: Ingresar nuevo tipo de proyecto

##### **Descripción:**

El caso de uso lo ejecuta el GERENTE cuando se obtiene un nuevo tipo de proyecto que está relacionado con los Modelos de Aplicación de Ge@.

##### **Precondiciones:**

El Gerente conoce los Modelos de Aplicación y el uso de la herramienta web.

##### **Poscondiciones:**

El nuevo tipo de proyecto se ingresa en la base de datos.

##### **Flujo Básico:**

1. El gerente hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. El gerente ingresa el nombre y descripción del nuevo tipo de proyecto.
4. Guarda el nuevo tipo de proyecto.
5. Los datos se almacenan en la base de datos.

**Caso de uso CU12: Modificar tipo de proyecto****Descripción:**

El caso de uso lo ejecuta el GERENTE y es cuando desea editar los datos de un tipo de proyecto.

**Precondiciones:**

El gerente conoce el sistema y sus módulos.

**Poscondiciones:**

Una vez que el gerente haya modificado los datos del tipo de proyecto este quedará almacenado en la base de datos.

**Flujo Básico:**

1. El gerente hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. Busca el tipo de proyecto a modificar.
4. Modifica el nombre y/o descripción del tipo de proyecto.
5. Guarda los cambios.
6. Los datos se actualizan en la base de datos.

**Caso de uso CU13: Desactivar tipo de proyecto****Descripción:**

El caso de uso lo ejecuta el GERENTE cuando desea desactivar temporalmente algún tipo de proyecto (modelo de aplicación) que no esté implementada, actualmente existen tres modelos implementados.

**Precondiciones:**

El gerente tiene acceso al módulo de proyectos del sistema y domina su manejo.

**Poscondiciones:**

Una vez que el gerente haya desactivado el tipo de proyecto no se podrá generar ningún proyecto de esta índole.

**Flujo Básico:**

1. El gerente hace Login correctamente
2. Ingresa al módulo de proyectos.
3. Consulta el tipo de proyecto a desactivar.
4. Desactiva el proyecto.
5. El cambio se actualiza en la base de datos.

**Caso de uso CU14: Activar tipo de proyecto****Descripción:**

El caso de uso lo ejecuta el GERENTE cuando desea activar un tipo de proyecto que haya sido desactivado con anterioridad.

**Precondiciones:**

El gerente debe tener acceso al sistema y de disponer de los permisos necesarios para entrar al módulo de proyecto.

**Poscondiciones:**

La activación se reflejara en la base de datos.

**Flujo Básico:**

1. El gerente hace Login correctamente.
2. Ingresa al módulo de proyectos.
3. Selecciona el tipo de proyecto a activar.
4. Activa el tipo de proyecto.
5. El cambio se actualiza en la base.

**Caso de uso CU15: Consultar catálogo****Descripción:**

El caso de uso lo ejecuta el GERENTE cuando desea desactivar/activar o modificar un tipo de proyecto.



**Precondiciones:**

El gerente debe disponer de un listado de todos los tipos de proyectos del sistema.

**Poscondiciones:**

Ninguna.

**Flujo Básico:**

1. El gerente hace Login correctamente.
2. Ingresa al módulo de proyectos.
3. Lista los tipos de proyectos a realizar una operación.
4. Cualquier actividad actualizara los datos en la base.

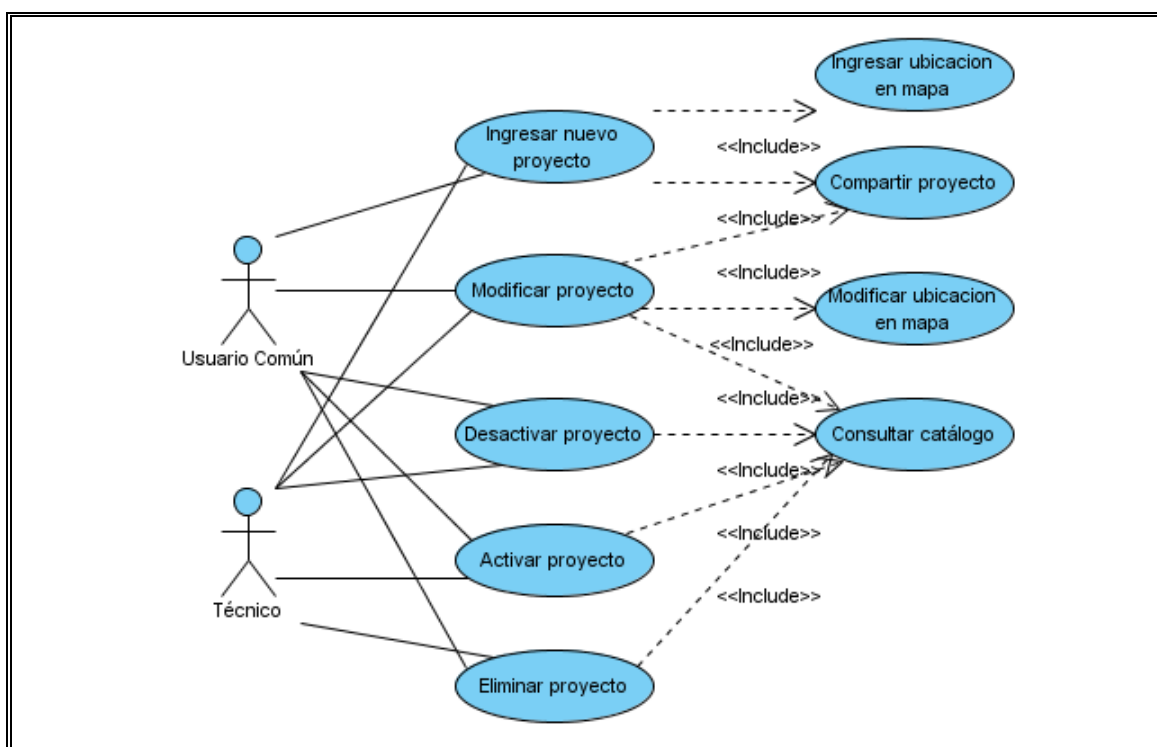


Figura 3-22: Mantenimiento de proyectos (casos de uso)

**Caso de uso CU16: Ingresar nuevo proyecto****Descripción:**

El caso de uso lo ejecuta el TÉCNICO o USUARIO COMÚN cuando se obtiene un nuevo proyecto a resolver y brindar asesoría.

**Precondiciones:**

El técnicoo usuario común conoce los datos del proyecto, y el uso del sistema.

**Poscondiciones:**

El nuevo proyecto se registra en la base de datos.

**Flujo Básico:**

1. El gerente hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. El gerente ingresa los datos del nuevo proyecto.
4. Opcionalmente puede agregar puntos en el mapa o compartir proyectos.
5. Guarda el nuevo proyecto.
6. Los datos se almacenan en la base de datos.

**Caso de uso CU17: Ingresar ubicación en el mapa.****Descripción:**

El caso de uso lo ejecuta el TÉCNICO o USUARIO COMÚN cuando desea indicar en un mapa la zona en donde se está llevando a cabo un determinado proyecto, este caso de uso no es necesario que se ejecute, ya que puede haber el caso de que exista o no dicha ubicación.

**Precondiciones:**

El gerente o usuario común conoce la zona en donde se desarrolla el proyecto y conoce el sistema.

**Poscondiciones:**

Se configura zonas de desarrollo del proyecto dentro de un mapa, esta información se

registra en la base de datos.

**Flujo Básico:**

1. El técnico o usuario común hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. Busca el proyecto a modificar.
4. En el mapa se colocan los puntos que indican la zona de desarrollo.
5. Guarda los cambios.
6. Los datos se actualizan en la base de datos.

**Caso de uso CU18: Compartir proyecto**

**Descripción:**

El caso de uso lo ejecuta el USUARIO COMÚN O TÉCNICO cuando desea compartir un determinado proyecto con otros usuarios o técnicos de una misma o distinta institución.

**Precondiciones:**

El usuario común o técnico tiene acceso al módulo de proyectos del sistema y domina su manejo, conoce a los usuarios que va a compartir el proyecto. En caso que se esté modificando el compartir proyectos siempre se podrá hacer si el técnico o usuario común son los propietarios.

**Poscondiciones:**

El usuario al cual se le haya compartido el proyecto no podrá configurarlo más que únicamente la persona que lo creó. Los usuarios a los cuales se les haya compartido se registrarán en la base de datos.

**Flujo Básico:**

1. El usuario común o técnico hace Login correctamente
2. Ingresa al módulo de proyectos.
3. Consulta el proyecto a compartir.
4. Selecciona el tipo de institución y la institución en donde se encuentran los usuarios a cual compartir.

5. Se selecciona los usuarios a compartir.
6. Se Guarda los cambios.
7. Se registra en la base de datos los usuarios a los cuales se les comparte el proyecto.

#### **Caso de uso CU19: Modificar proyecto**

##### **Descripción:**

El caso de uso lo ejecuta el TÉCNICO o USUARIO COMÚN cuando desea modificar los datos con los que fue configurado un proyecto al momento de su creación.

##### **Precondiciones:**

El técnico o usuario común deben ser los propietarios del proyecto ya que son ellos los únicos que pueden editarlos, además debe conocer el manejo del módulo de proyectos.

##### **Poscondiciones:**

Los cambios indicaran una actualización en la base de datos.

##### **Flujo Básico:**

1. El técnico o usuario común hacen Login correctamente.
2. Ingresa al módulo de proyectos.
3. Busca el proyecto a modificar.
4. Seleccionan el proyecto a modificar, ven que sea de su autoría.
5. Modifican los datos.
6. El cambio se actualiza en la base.

#### **Caso de uso CU20: Modificar datos de mapa**

##### **Descripción:**

El caso de uso lo ejecuta el TÉCNICO o USUARIO COMÚN cuando desea modificar los datos de ubicación en el mapa en donde se encuentre desarrollando el proyecto, únicamente si son propietarios.

##### **Precondiciones:**

El usuario común o técnico debe conocer la zona en donde se desarrolla el proyecto y el

manejo del sistema. Debe ser propietario del proyecto para editar los datos del mapa.

**Poscondiciones:**

Los cambios se reflejarán en la base de datos.

**Flujo Básico:**

1. El usuario común o técnico hacen Login correctamente.
2. Ingresa al módulo de proyectos.
3. Buscan el proyecto al cual se le desea editar los datos del mapa.
4. Editan los puntos del mapa.
5. Guarda los cambios.
6. Los datos se refrescaran también en la base de datos.

**Caso de uso CU21: Consultar catálogo**

**Descripción:**

El caso de uso lo ejecuta tanto el TÉCNICO como el USUARIO COMÚN cuando desea modificar los datos del proyecto, del mapa; cuando se desea compartir el proyecto, activarlo/desactivarlo o eliminarlo.

**Precondiciones:**

El gerente debe disponer de un listado de todos los proyectos del sistema.

**Poscondiciones:**

Estar seguro de los cambios que vaya a realizar, ya que los cambios se actualizarán en la base de datos.

**Flujo Básico:**

1. El usuario común o técnico hacen Login correctamente.
2. Ingresa al módulo de proyectos.
3. Listan los proyectos.
4. Seleccionan el proyecto a realizar una operación.
5. Cualquier actividad actualizara los datos en la base.

**Caso de uso CU22: Desactivar proyecto****Descripción:**

El caso de uso lo ejecuta tanto el TÉCNICO como el USUARIO COMÚN cuando desea desactivar un proyecto.

**Precondiciones:**

El gerente debe conocer el sistema y estar seguro de desactivar un proyecto. Debe ser propietario del proyecto.

**Poscondiciones:**

Si se desactivo un proyecto puede ser activado, esta no es una desactivación permanente.

**Flujo Básico:**

1. El usuario común o técnico hacen Login correctamente.
2. Ingresa al módulo de proyectos.
3. Buscan el proyecto.
4. Desactivan el proyecto.
5. La desactivación se refleja en la base de datos.

**Caso de uso CU23: Activar proyecto****Descripción:**

El caso de uso lo ejecuta tanto el TÉCNICO como el USUARIO COMÚN cuando desea activar un proyecto.

**Precondiciones:**

El gerente debe conocer el sistema y estar seguro de activar un proyecto.

**Poscondiciones:**

Una vez que se active el proyecto todos los usuarios que contenían este proyecto van a poder acceder a él.

**Flujo Básico:**

1. El usuario común o técnico hacen Login correctamente.

2. Ingresar al módulo de proyectos.
3. Buscar el proyecto.
4. Activar el proyecto.
5. La activación se refleja en la base de datos.

#### **Caso de uso CU24: Eliminar proyecto**

##### **Descripción:**

El caso de uso lo ejecuta tanto el TÉCNICO como el USUARIO COMÚN cuando desea eliminar un proyecto.

##### **Precondiciones:**

El gerente debe conocer el sistema y estar seguro de eliminar un proyecto, además de ser el propietario, no cualquier usuario puede llevar a cabo esta operación.

##### **Poscondiciones:**

Si se eliminó un proyecto este no puede ser recuperado, es una eliminación permanente.

##### **Flujo Básico:**

1. El usuario común o técnico hacen Login correctamente.
2. Ingresar al módulo de proyectos.
3. Buscar el proyecto.
4. Eliminar el proyecto.
5. Se confirma la eliminación.
6. El proyecto es eliminado de la base de datos.

### 3.2.2.3 Gestión de instituciones

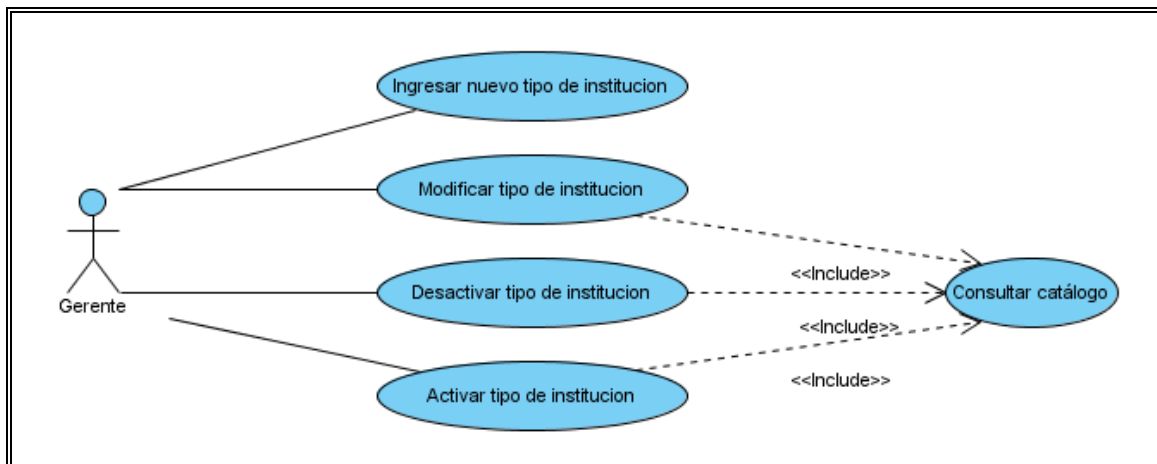


Figura 3-23: Mantenimiento de Tipo de Instituciones (casos de uso)

#### Caso de uso CU25: Ingresar nuevo tipo de institución

##### **Descripción:**

El caso de uso lo ejecuta el GERENTE cuando se obtiene un nuevo tipo de institución ya sea esta una entidad pública o privada.

##### **Precondiciones:**

El Gerente conoce la herramienta web y los datos del nuevo tipo de institución.

##### **Poscondiciones:**

El nuevo tipo de institución se ingresa en la base de datos.

##### **Flujo Básico:**

1. El gerente hace Login correctamente.
2. Se dirige al módulo de instituciones.
3. El gerente ingresa el nombre y descripción del nuevo tipo de institución.
4. Guarda el nuevo tipo de institución.
5. Los datos se almacenan en la base de datos.



**Caso de uso CU26: Modificar tipo de institución****Descripción:**

El caso de uso lo ejecuta el GERENTE y es cuando desea editar los datos de un tipo de institución.

**Precondiciones:**

El gerente conoce el sistema y el módulo de institución, cuenta con los permisos para acceder a la zona de administración.

**Poscondiciones:**

Una vez que el gerente haya modificado los datos del tipo de institución este quedará almacenado en la base de datos.

**Flujo Básico:**

1. El gerente hace Login correctamente.
2. Se dirige al módulo de institución.
3. Busca el tipo de institución a modificar.
4. Modifica el nombre y/o descripción del tipo de institución.
5. Guarda los cambios.
6. Los datos se actualizan en la base de datos.

**Caso de uso CU27: Desactivar tipo de institución****Descripción:**

El caso de uso lo ejecuta el GERENTE cuando desea desactivar temporalmente algún tipo de institución.

**Precondiciones:**

El gerente debe tener acceso al módulo de institución del sistema y dominar su manejo.

**Poscondiciones:**

Una vez que el gerente haya desactivado el tipo de institución no se podrá asignar instituciones relacionadas a este tipo.

**Flujo Básico:**

1. El gerente hace Login correctamente
2. Ingresa al módulo de institución.
3. Consulta el tipo de institución a desactivar.
4. Desactiva la institución.
5. El cambio se actualiza en la base de datos.

**Caso de uso CU28: Activar tipo de institución****Descripción:**

El caso de uso lo ejecuta el GERENTE cuando desea activar un tipo de institución.

**Precondiciones:**

El gerente debe tener acceso al sistema y de disponer de los permisos necesarios para entrar al módulo de institución.

**Poscondiciones:**

La activación se reflejara en la base de datos.

**Flujo Básico:**

1. El gerente hace Login correctamente.
2. Ingresa al módulo de instituciones.
3. Selecciona el tipo de institución a activar.
4. Activa el tipo de institución.
5. El cambio se actualiza en la base.

**Caso de uso CU29: Consultar catálogo****Descripción:**

El caso de uso lo ejecuta el GERENTE cuando desea desactivar/activar o modificar un tipo de institución.

**Precondiciones:**

El gerente debe disponer de un listado de todos los tipos de institución del sistema.

**Poscondiciones:**

Cualquier cambio provocara una actualización en la base de datos.

**Flujo Básico:**

1. El gerente hace Login correctamente.
2. Ingresa al módulo de instituciones.
3. Lista los tipos de instituciones a realizar una operación.
4. Cualquier actividad actualizara los datos en la base.

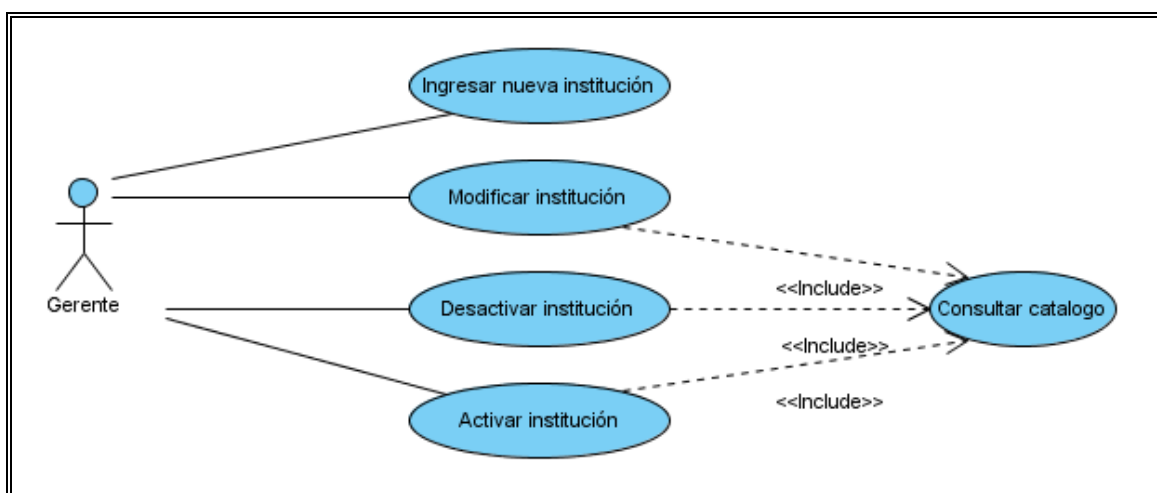


Figura 3-24: Mantenimiento de Instituciones (casos de uso)

**Caso de uso CU30: Ingresar nueva institución****Descripción:**

El caso de uso lo ejecuta el GERENTE cuando se obtiene una nueva institución ya sea esta una entidad pública o privada.

**Precondiciones:**

El Gerente conoce la herramienta web y los datos dela nueva institución.

**Poscondiciones:**

La nueva institución se ingresa en la base de datos.

**Flujo Básico:**

1. El gerente hace Login correctamente.
2. Se dirige al módulo de instituciones.
3. El gerente ingresa el tipo de institución, nombre y descripción de la nueva institución.
4. Guarda la nueva institución.
5. Los datos se almacenan en la base de datos.

**Caso de uso CU31: Modificar institución****Descripción:**

El caso de uso lo ejecuta el GERENTE y es cuando desea editar los datos de una institución.

**Precondiciones:**

El gerente conoce el sistema y el módulo de institución, cuenta con los permisos para acceder a la zona de administración.

**Poscondiciones:**

Una vez que el gerente haya modificado los datos de la institución este quedará almacenado en la base de datos.

**Flujo Básico:**

7. El gerente hace Login correctamente.
8. Se dirige al módulo de institución.
9. Busca la institución a modificar.
1. Modifica el tipo de institución, nombre y/o descripción de la institución.
2. Guarda los cambios.
3. Los datos se actualizan en la base de datos.

**Caso de uso CU32: Desactivar institución****Descripción:**

El caso de uso lo ejecuta el GERENTE cuando desea desactivar temporalmente una institución.

**Precondiciones:**

El gerente debe tener permisos para acceder al módulo de institución del sistema y dominar su manejo.

**Poscondiciones:**

Una vez que el gerente haya desactivado la institución no se podrá asignar usuarios relacionadas a esa institución.

**Flujo Básico:**

1. El gerente hace Login correctamente
2. Ingresa al módulo de institución.
3. Consulta la institución a desactivar.
4. Desactiva la institución.
5. El cambio se actualiza en la base de datos.

**Caso de uso CU33: Activar institución****Descripción:**

El caso de uso lo ejecuta el GERENTE cuando desea activar una institución.

**Precondiciones:**

El gerente debe tener acceso al sistema y de disponer de los permisos necesarios para entrar al módulo de institución.

**Poscondiciones:**

La activación se reflejara en la base de datos.

**Flujo Básico:**

1. El gerente hace Login correctamente.

2. Ingresa al módulo de instituciones.
3. Buscala institución a activar.
4. Activa la institución.
5. El cambio se actualiza en la base.

#### **Caso de uso CU34: Consultar catálogo**

##### **Descripción:**

El caso de uso lo ejecuta el GERENTE cuando desea desactivar/activar o modificar una institución.

##### **Precondiciones:**

El gerente debe disponer de un listado de todas las instituciones del sistema.

##### **Poscondiciones:**

Cualquier cambio provocara una actualización en la base de datos.

##### **Flujo Básico:**

1. El gerente hace Login correctamente.
2. Ingresa al módulo de instituciones.
3. Lista las instituciones a realizar una operación.
4. Cualquier actividad actualizara los datos en la base.

### 3.2.2.4 Gestión de sitio

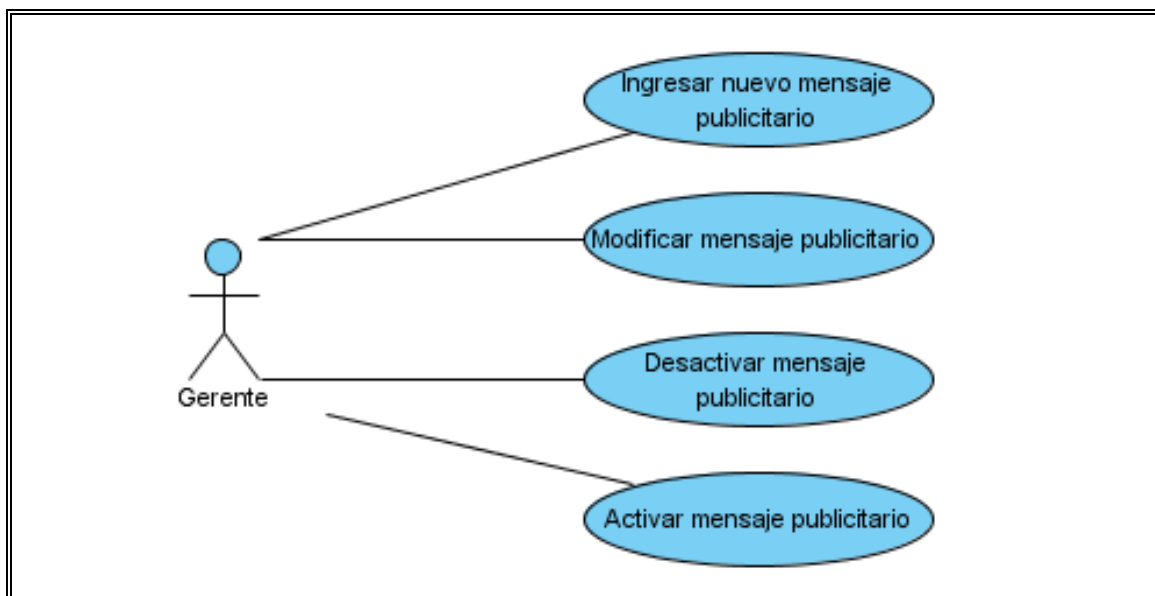


Figura 3-25: Mantenimiento de la parte administrativa (casos de uso)

#### Caso de uso CU35: Ingresar nuevo mensaje publicitario

##### **Descripción:**

El caso de uso lo ejecuta el GERENTE cuando se desea comunicar información al resto de usuarios registrados en la herramienta web.

##### **Precondiciones:**

El Gerente conoce la herramienta web y el mensaje que desea comunicar mediante el mensaje publicitario.

##### **Poscondiciones:**

El mensaje será visualizado por todos los usuarios que inicien sesión luego de que se creó el mensaje.

##### **Flujo Básico:**

1. El gerente hace Login correctamente.
2. Se dirige al módulo de administración.
3. El gerente ingresa el mensaje de publicidad.

4. Guarda el mensaje.
5. La información se almacenan en la base de datos.

#### **Caso de uso CU36: Modificar mensaje publicitario**

##### **Descripción:**

El caso de uso lo ejecuta el GERENTE y es cuando desea editar el mensaje de publicidad.

##### **Precondiciones:**

El gerente conoce el sistema y el módulo de administración, debe contar con los permisos y conocer el mensaje que se desea transmitir.

##### **Poscondiciones:**

Una vez que el gerente haya modificado el mensaje este quedará almacenado en la base de datos y va a ser visualizado por todos los usuarios de la herramienta.

##### **Flujo Básico:**

1. El gerente hace Login correctamente.
2. Se dirige al módulo de administración.
3. Modifica el mensaje publicitario.
4. Guarda los cambios.
5. Los datos se actualizan en la base de datos.

#### **Caso de uso CU37: Desactivar mensaje publicitario**

##### **Descripción:**

El caso de uso lo ejecuta el GERENTE cuando desea desactivar temporalmente el mensaje publicitario.

##### **Precondiciones:**

El gerente debe tener permisos para acceder al módulo de administración del sitio y dominar su manejo y estar seguro de desactivar el mensaje.

##### **Poscondiciones:**



Una vez que el gerente haya desactivado el mensaje este no podrá ser visualizado por el resto de usuarios.

**Flujo Básico:**

1. El gerente hace Login correctamente
2. Ingresa al módulo de administración.
3. Desactiva el mensaje publicitario.
4. El cambio se actualiza en la base de datos.

**Caso de uso CU38: Activar mensaje publicitario**

**Descripción:**

El caso de uso lo ejecuta el GERENTE cuando desea activar el mensaje publicitario.

**Precondiciones:**

El gerente debe tener acceso al sistema y de disponer de los permisos necesarios para entrar al módulo de administración.

**Poscondiciones:**

La activación se reflejara en la base de datos, y en las interfaces de los usuarios que podrán observar el mensaje de publicidad.

**Flujo Básico:**

1. El gerente hace Login correctamente.
2. Ingresa al módulo de administración.
3. Activa el mensaje publicitario.
4. El cambio se actualiza en la base.

3.2.2.5 Modelamiento Digital del Proyecto

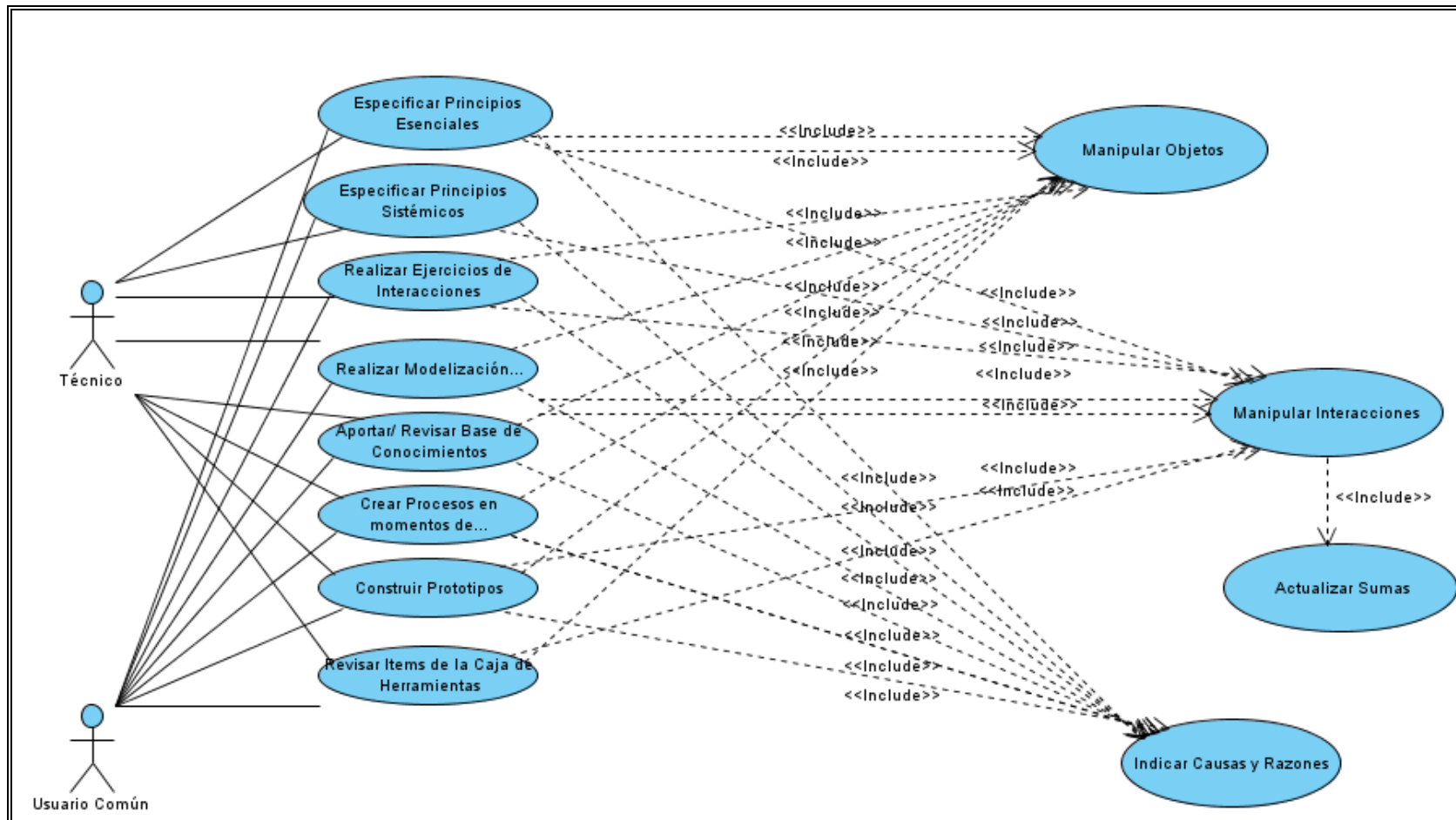


Figura 3-26: Caso de uso de Modelamiento Digital del Proyecto

### Caso de uso CU39: Especificar Principios Esenciales

#### **Descripción:**

El caso de uso lo ejecuta el TÉCNICO o el USUARIO COMÚN cuando se quiere modelar un proyecto.

#### **Precondiciones:**

El usuario tiene conocimiento del modelamiento teórico en el que se basa la herramienta web, debe tener acceso al sistema y haber abierto un proyecto específico

#### **Poscondiciones:**

Las especificaciones de los Principio Esenciales deben almacenarse en la base de datos.

#### **Flujo Básico:**

1. El usuario hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. Ingresa a un proyecto específico.
4. Ingresa al módulo de Principios Esenciales.
5. El usuario ingresa, modifica o elimina los principios esenciales y sus características.
6. Los cambios referentes a Principios Esenciales se almacenan en la base de datos

### Caso de uso CU40: Especificar Principios Sistémicos

#### **Descripción:**

El caso de uso lo ejecuta el TÉCNICO o el USUARIO COMÚN cuando requiere especificar los Principios Sistémicos del un determinado proyecto.

#### **Precondiciones:**

El usuario tiene conocimiento del modelamiento teórico en el que se basa la herramienta web, debe tener acceso al sistema y haber abierto un proyecto específico

#### **Poscondiciones:**

Las especificaciones de los Principio Sistémicos deben almacenarse en la base de datos.

#### **Flujo Básico:**

1. El usuario hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. Ingresa a un proyecto específico.
4. Ingresa al módulo de Principios Sistémicos.
5. El usuario ingresa, modifica o elimina los Principios Sistémicos y sus características
6. Los cambios referentes a Principios Sistémicos se almacenan en la base de datos

#### **Caso de uso CU41: Realizar Ejercicios de Interacciones**

##### **Descripción:**

El caso de uso lo ejecuta el TÉCNICO o el USUARIO COMÚN cuando requiere realizar Ejercicios de Interacciones dentro un determinado proyecto.

##### **Precondiciones:**

El usuario tiene conocimiento del modelamiento teórico en el que se basa la herramienta web, debe tener acceso al sistema y haber abierto un proyecto específico

##### **Poscondiciones:**

Las especificaciones de los Principio Sistémicos deben almacenarse en la base de datos.

##### **Flujo Básico:**

1. El usuario hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. Ingresa a un proyecto específico.
4. Ingresa al módulo de Ejercicios de Interacciones.
5. El usuario ingresa, modifica o elimina Ejercicios de interacciones y sus características
6. Los cambios referentes a Ejercicios de interacciones se almacenan en la base de datos

#### **Caso de uso CU42: Relizar Modelización Digital**

##### **Descripción:**

El caso de uso lo ejecuta el TÉCNICO o el USUARIO COMÚN cuando requiere Modelizar Digitalmente un determinado proyecto.

##### **Precondiciones:**

El usuario tiene conocimiento del modelamiento teórico en el que se basa la herramienta web, debe tener acceso al sistema y haber abierto un proyecto específico

**Poscondiciones:**

Las especificaciones de la Modelización Digital deben almacenarse en la base de datos.

**Flujo Básico:**

1. El usuario hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. Ingresa a un proyecto específico.
4. Ingresa al módulo de Modelización Digital.
5. El usuario ingresa, modifica o elimina los objetos y características que permiten modelar digitalmente al proyecto
6. Los cambios referentes a la Modelización Digital se almacenan en la base de datos
- 7.

**Caso de uso CU43: Aportar a la Base de Conocimientos****Descripción:**

El caso de uso lo ejecuta el TÉCNICO o el USUARIO COMÚN cuando requiere aportar o extraer información de la Base de Conocimientos.

**Precondiciones:**

El usuario tiene conocimiento del modelamiento teórico en el que se basa la herramienta web, debe tener acceso al sistema y haber abierto un proyecto específico

**Poscondiciones:**

Las especificaciones de los Principio Sistémicos deben almacenarse en la base de datos.

**Flujo Básico:**

1. El usuario hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. Ingresa a un proyecto específico.
4. Ingresa al módulo de Base de Conocimientos.
5. El usuario ingresa o modifica información teórica de la Base de Conocimiento Capitalizada
6. Los cambios referentes a la Base de Conocimiento se almacenan en la base de datos

**Caso de uso CU44: Crear Procesos en Momentos de Intervención****Descripción:**

El caso de uso lo ejecuta el TÉCNICO o el USUARIO COMÚN cuando requiere revisar procesos en los Momentos de Intervención.

**Precondiciones:**

El usuario tiene conocimiento del modelamiento teórico en el que se basa la herramienta web, debe tener acceso al sistema.

**Flujo Básico:**

1. El usuario hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. Ingresa a un proyecto específico.
4. Ingresa al módulo de Momentos de Intervención
5. El usuario visualiza información acerca de los Procesos creados en los Momentos de Intervención

**Caso de uso CU45: Construir Prototipos****Descripción:**

El caso de uso lo ejecuta el TÉCNICO o el USUARIO COMÚN cuando requiere manejar prototipos.

**Precondiciones:**

El usuario tiene conocimiento del modelamiento teórico en el que se basa la herramienta web, debe tener acceso al sistema

**Poscondiciones:**

Las especificaciones de los Prototipos y los elementos que intervienen en ellos deben almacenarse en la base de datos.

**Flujo Básico:**

1. El usuario hace Login correctamente.
2. Se dirige al módulo de proyectos.

3. Ingresa a un proyecto específico.
4. Ingresa al módulo de Prototipos.
5. El usuario ingresa, modifica o elimina los prototipos y las características que los componen.
6. Los cambios referentes a la Modelización Digital se almacenan en la base de datos.

#### **Caso de uso CU46: Revisar Ítems de la Caja de Herramientas**

##### **Descripción:**

El caso de uso lo ejecuta el TÉCNICO o el USUARIO COMÚN cuando requiere manipular alguno o todos los 11 ítems de la Caja de Herramientas.

##### **Precondiciones:**

El usuario tiene conocimiento del modelamiento teórico en el que se basa la herramienta web, debe tener acceso al sistema.

##### **Poscondiciones:**

Las modificaciones en los ítems de la Caja de Herramientas los elementos que intervienen en ellos deben almacenarse en la base de datos.

##### **Flujo Básico:**

1. El usuario hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. Ingresa a un proyecto específico.
4. Ingresa al módulo de la Caja de Herramientas.
5. El usuario ingresa, modifica o elimina los elementos que integran a los ítems de la Caja de Herramientas.
6. Los cambios referentes a la Caja de Herramientas se almacenan en la base de datos.

#### **Caso de uso CU47: Manipular Objetos**

##### **Descripción:**

El caso de uso lo ejecuta el TÉCNICO o el USUARIO y es uno de los más comunes, interviene en cada uno de los módulos debido a que el principio de Ge@ es modelar la realidad en objetos sistematizados representados mediante elipses.

**Precondiciones:**

El usuario tiene conocimiento del modelamiento teórico en el que se basa la herramienta web, debe tener acceso al sistema.

**Poscondiciones:**

Las modificaciones en los objetos deben almacenarse en la base de datos.

**Flujo Básico:**

1. El usuario hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. Ingresa a un proyecto específico.
4. Ingresa a alguno de los módulos de modelamiento.
5. El usuario ingresa, modifica o elimina los objetos, sus características gráficas, conceptos que integren al objeto
6. Los cambios referentes al objeto y los elementos involucrados se almacenan en la base de datos.

**Caso de uso CU48: Manipular Interacciones****Descripción:**

El caso de uso lo ejecuta el TÉCNICO o el USUARIO y es junto con el CU47 uno de los más comunes, interviene en cada uno de los módulos debido a que el principio de Ge@ es relacionar a los diferente objetos, para conocer los flujos de información y el estado de los sistemas que se modelan.

**Precondiciones:**

El usuario tiene conocimiento del modelamiento teórico en el que se basa la herramienta web, debe tener acceso al sistema.

**Poscondiciones:**

Las modificaciones en las interacciones deben almacenarse en la base de datos.

**Flujo Básico:**

1. El usuario hace Login correctamente.



2. Se dirige al módulo de proyectos.
3. Ingresa a un proyecto específico.
4. Ingresa a alguno de los módulos de modelamiento.
5. El usuario ingresa, modifica o elimina las interacciones, especifica cómo se encuentra la interacción, cómo debería encontrarse y si la interacción es fuerte.
6. Los cambios referentes al objeto y los elementos involucrados se almacenan en la base de datos.

#### **Caso de uso CU49: Indicar Causas y Razones**

##### **Descripción:**

El caso de uso lo ejecuta el TÉCNICO o el USUARIO y va de la mano con el CU48, consiste en especificar las razones de los porqué que surgen al momento de valorar al estado de las interacciones.

##### **Precondiciones:**

El usuario tiene conocimiento del modelamiento teórico en el que se basa la herramienta web, debe tener acceso al sistema.

##### **Poscondiciones:**

Las modificaciones en la las causas y razones deben almacenarse en la base de datos.

##### **Flujo Básico:**

1. El usuario hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. Ingresa a un proyecto específico.
4. Ingresa a alguno de los módulos de modelamiento.
5. El usuario selecciona una interacción
6. Se especifican las causas y razones por las que se han puesto los valores: como está y como debería estar.
7. Las especificaciones de las causas y razones se almacenan en la base de datos.

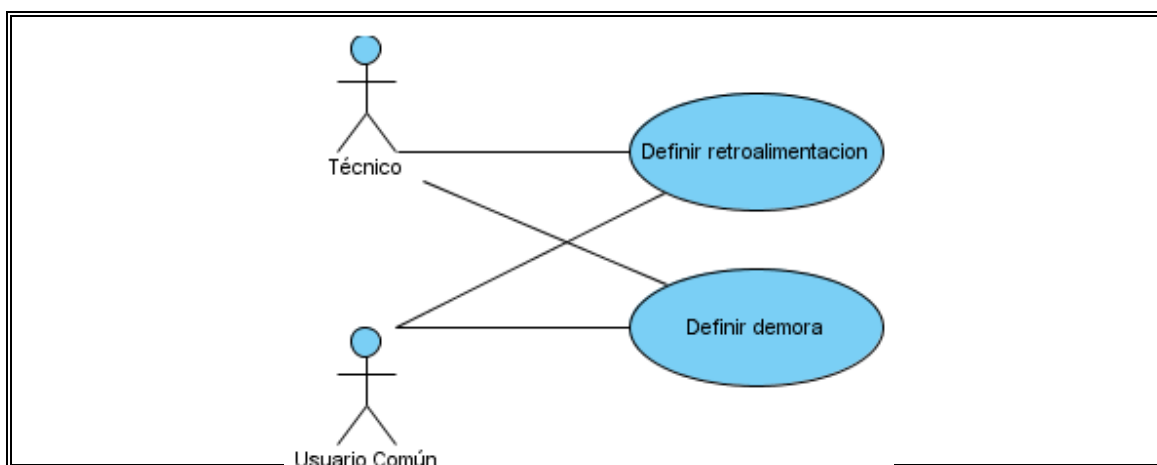


Figura 3-27: Caso de Uso Mantenimiento Realimentación

#### Caso de uso CU51: Mantenimiento de Realimentación

##### **Descripción:**

El caso de uso lo ejecuta el TÉCNICO o el USUARIO, se lo lleva a cabo cuando se está modelando un prototipo y el usuario requiere visualizar el Diagrama Sistémico de las Variables, en este caso los objetos (elipses) representan variables que intentan equilibrar un sistema, las relaciones son conexiones entre las elipses que forman un ciclo por lo que ya no se representa con líneas rectas sino con curvas.

##### **Precondiciones:**

El usuario tiene conocimiento del modelamiento teórico en el que se basa la herramienta web, debe tener acceso al sistema.

##### **Poscondiciones:**

Las modificaciones en las interacciones deben almacenarse en la base de datos.

##### **Flujo Básico:**

1. El usuario hace Login correctamente.
2. Se dirige al módulo de proyectos.
3. Ingresa a un proyecto específico.
4. Ingresa al módulo de Prototipos y selecciona alguno de ellos.
5. El usuario pide visualizar el Diagrama Sistémico de las variables.

6. El usuario ingresa, modifica o elimina los elementos de la realimentación, crea variables e indica si estas influyen para el crecimiento o compensación del sistema, además se indica si esta variable nueva genera una demora en el ciclo.
7. Los cambios referentes al objeto y los elementos involucrados se almacenan en la base de datos.

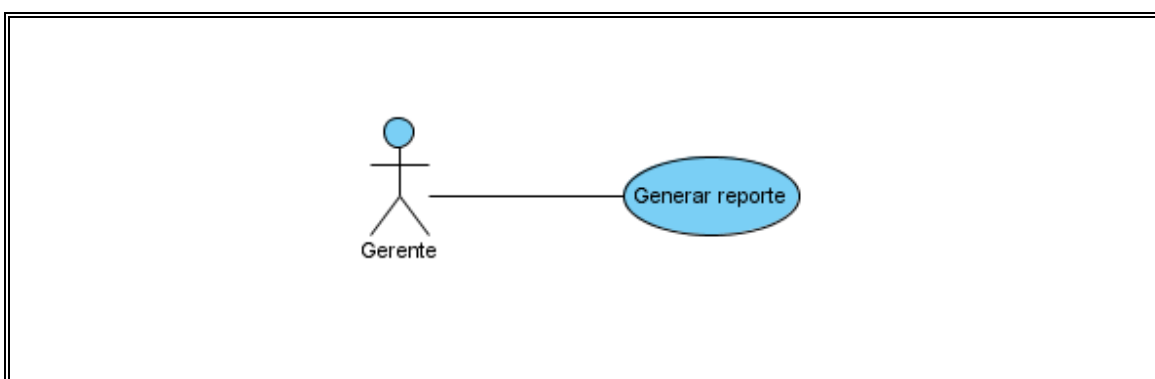


Figura 3-28: Caso de Uso Generar Reporte

### Caso de uso CU52: Generar Reporte

#### **Descripción:**

El caso de uso lo ejecuta el GERENTE y es cuando requiere conocer cómo se encuentran los objetos, sistemas, elementos de los proyectos de los usuarios.

#### **Precondiciones:**

El gerente conoce el sistema y el módulo de administración, debe contar con los permisos y conocer cómo interpretar los datos.

Asegurarse que todos los datos estén actualizados antes de generar el reporte

#### **Poscondiciones:**

-

#### **Flujo Básico:**

1. El usuario hace Login correctamente.

2. Se dirige al módulo de proyectos.
3. Ingresa a un proyecto específico.
4. Ingresa al módulo en el cual quiera generar reportes e interpretar los datos.
5. El gerente pide que se genere un reporte de los objetos e interacciones que intervienen en un determinado sistema
6. El gerente elige en que formato desea descargar el reporte (PDF, Word, Excel)

### 3.2.3 Resumen de los casos de uso

A continuación se presenta una tabla que indica todos los casos de uso encontrados en el desarrollo de la herramienta web para la planificación, gestión y gobernabilidad territorial.

	Notación	Caso de uso	Actor
Gestión de Usuarios	CU1	Ingresar nuevo perfil	Gerente
	CU2	Modificar perfil de usuario	Gerente
	CU3	Desactivar perfil de usuario	Gerente
	CU4	Activar perfil de usuario	Gerente
	CU5	Autenticar	Gerente
	CU6	Consultar catalogo	Gerente
	CU7	Ingresar nuevo perfil	Usuario común
	CU8	Modificar perfil de usuario	Usuario común
	CU9	Desactivar perfil	Usuario común
	CU10	Autenticar	Usuario común
Gestión de proyectos	CU11	Ingresar nuevo tipo de proyecto	Gerente
	CU12	Modificar tipo de proyecto	Gerente
	CU13	Desactivar tipo de proyecto	Gerente
	CU14	Activar tipo de proyecto	Gerente
	CU15	Consultar catalogo	Gerente
	CU16	Ingresar nuevo proyecto	Técnico, Usuario Común
	CU17	Ingresar datos de mapa	Técnico, Usuario Común
	CU18	Compartir proyecto	Técnico, Usuario Común
	CU19	Modificar proyecto	Técnico, Usuario Común
	CU20	Modificar datos de mapa	Técnico, Usuario Común
	CU21	Desactivar proyecto	Técnico, Usuario Común
	CU22	Activar proyecto	Técnico, Usuario Común
	CU23	Eliminar proyecto	Técnico, Usuario Común
	CU24	Consultar catalogo	Técnico, Usuario Común

Gestión de instituciones	CU25	Ingresar nuevo tipo de institución	Gerente
	CU26	Modificar tipo de institución	Gerente
	CU27	Desactivar tipo de institución	Gerente
	CU28	Activar tipo de institución	Gerente
	CU29	Consultar catalogo	Gerente
	CU30	Ingresar nueva institución	Gerente
	CU31	Modificar institución	Gerente
	CU32	Desactivar institución	Gerente
	CU33	Activar institución	Gerente
	CU34	Consultar catalogo	Gerente
Gestión de administración	CU35	Ingresar nuevo mensaje publicitario	Gerente
	CU36	Modificar mensaje publicitario	Gerente
	CU37	Desactivar mensaje publicitario	Gerente
	CU38	Activar mensaje publicitario	Gerente
Modelamiento Digital del Proyecto	CU39	Especificar Principios Esenciales	Técnico, Usuario Común
	CU40	Especificar Principios Sistémicos	Técnico, Usuario Común
	CU41	Realizar Ejercicios de Interacciones	Técnico, Usuario Común
	CU42	Relizar Modelización Digital	Técnico, Usuario Común
	CU43	Aportar a la Base de Conocimientos	Técnico, Usuario Común
	CU44	Crear Procesos en Momentos de Intervención	Técnico, Usuario Común
	CU45	Construir Prototipos	Técnico, Usuario Común
	CU46	Revisar Items de la Caja de Herramientas	Técnico, Usuario Común
	CU47	Manipular Interacciones	Técnico, Usuario Común
	CU48	Manipular Objetos	Técnico, Usuario Común
	CU49	Indicar Causas y Razones	Técnico, Usuario Común
	CU50	Mantenimiento de Realimentación	Técnico, Usuario Común
	CU51	Generar Reporte	Gerente

### 3.2.3 Diagramas de actividades

#### 3.2.3.1 Caso de uso CU1

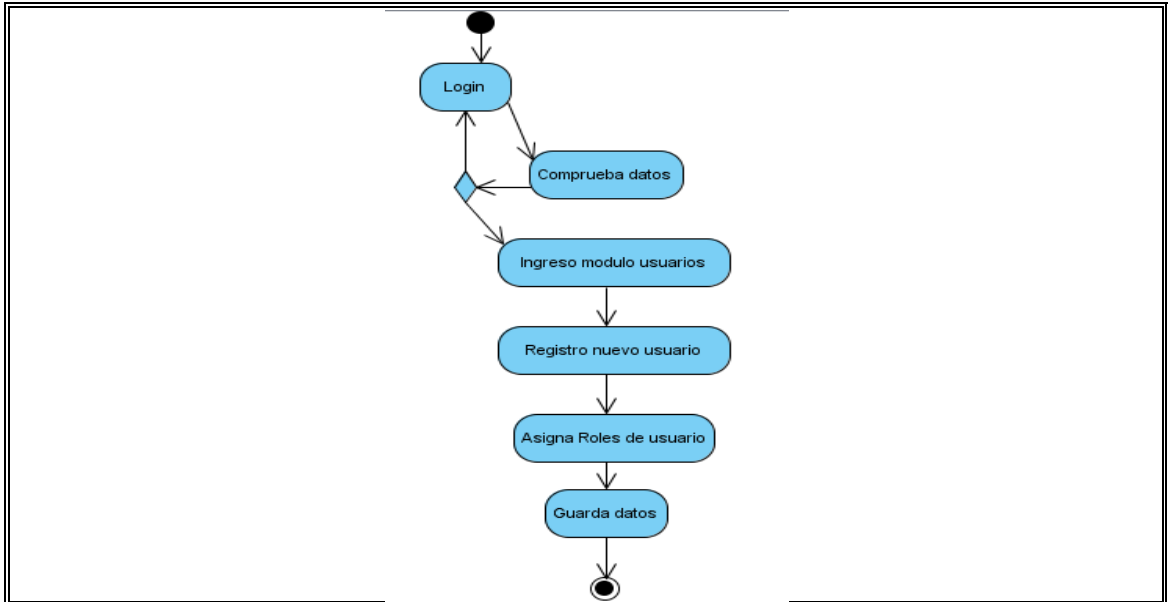


Figura 3-29: Diagrama de actividades del caso de uso CU1

#### 3.2.3.2 Caso de uso CU2

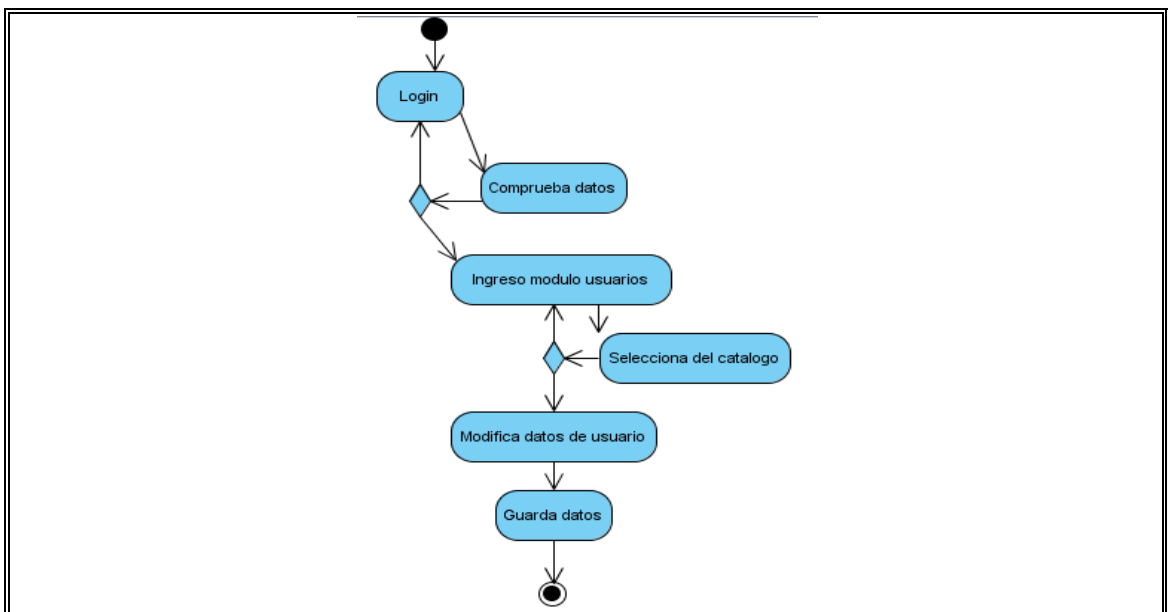


Figura 3-30: Diagrama de actividades del caso de uso CU2

### 3.2.3.3 Caso de uso CU3

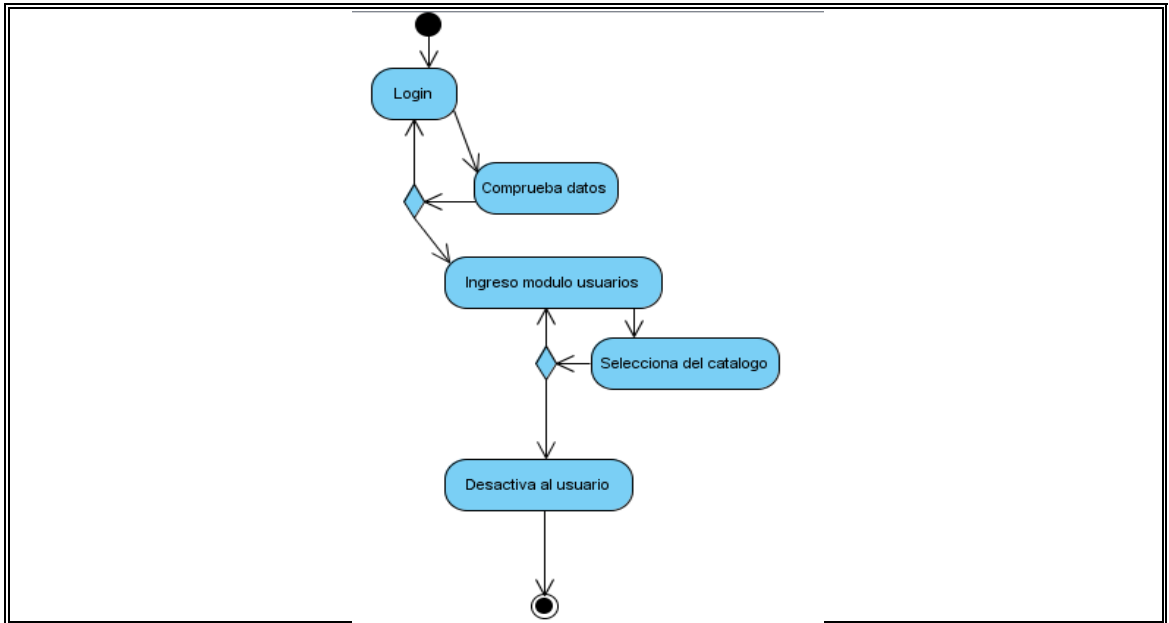


Figura 3-31: Diagrama de actividades del caso de uso CU3

### 3.2.3.4 Caso de uso CU4

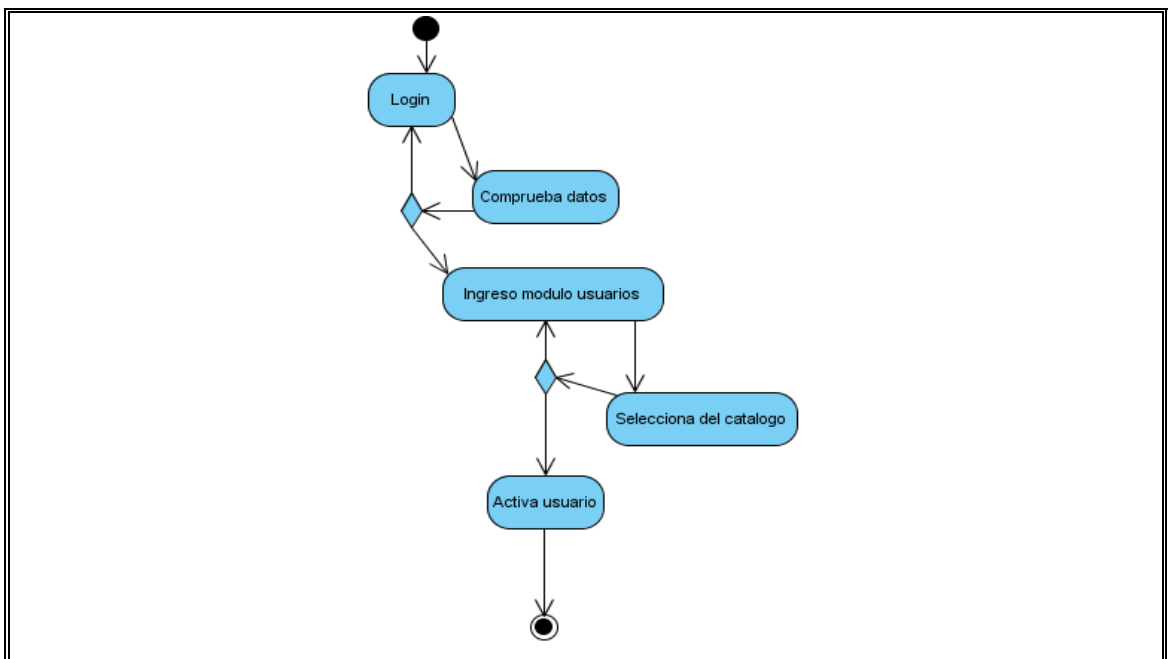


Figura 3-32: Diagrama de actividades del caso de uso CU4

### 3.2.3.5 Caso de uso CU5

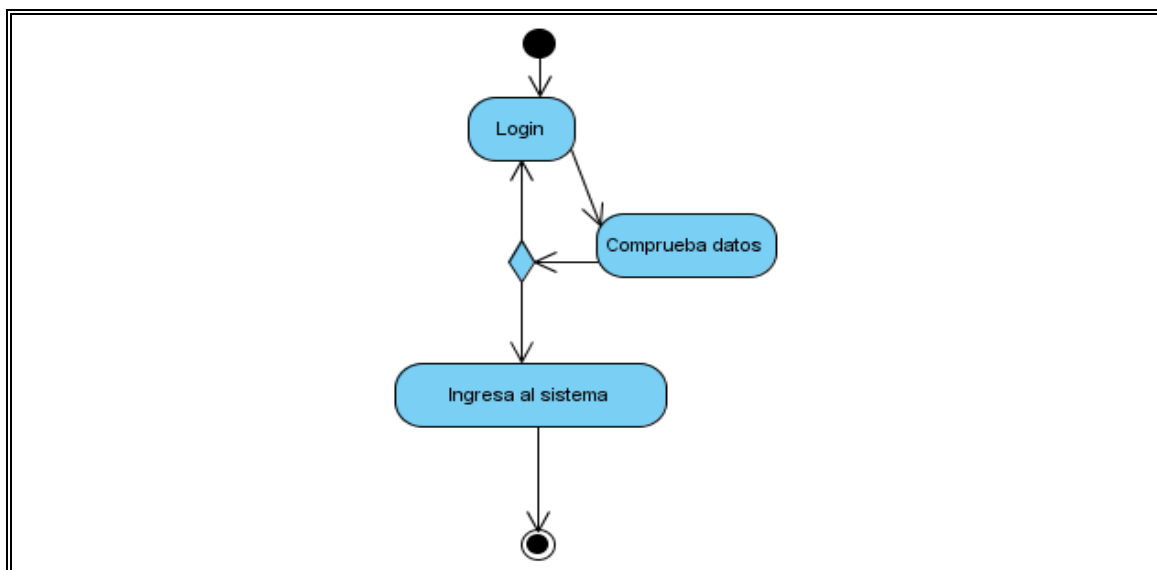


Figura 3-33: Diagrama de actividades del caso de uso CU5

### 3.2.3.6 Caso de uso CU6

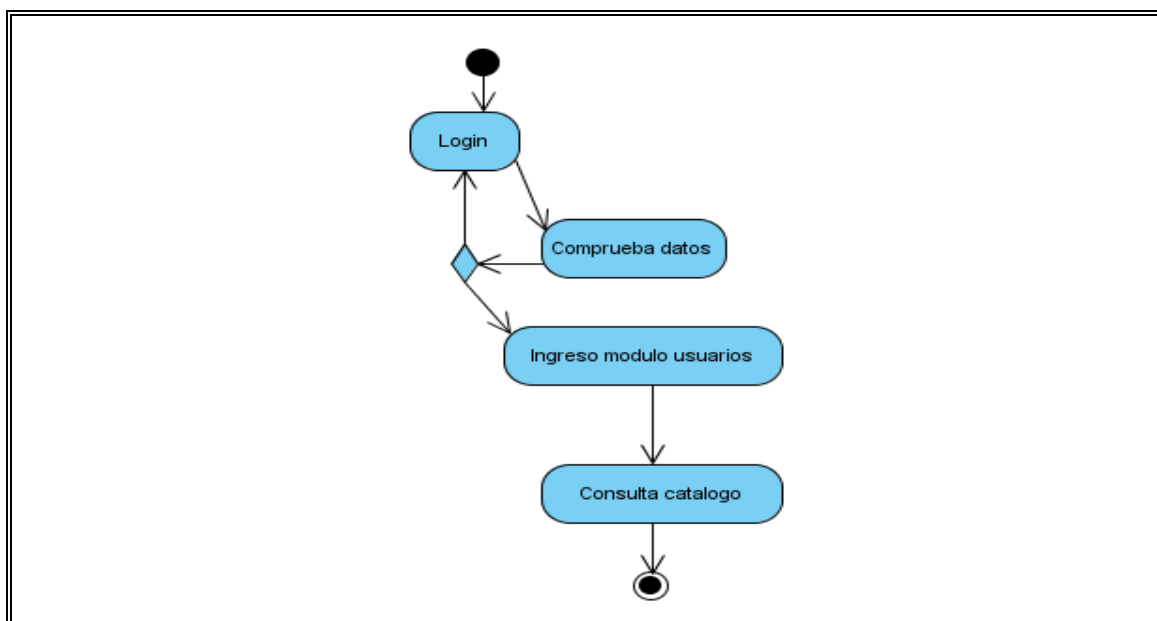


Figura 3-34: Diagrama de actividades del caso de uso CU6



### 3.2.3.7 Caso de uso CU7

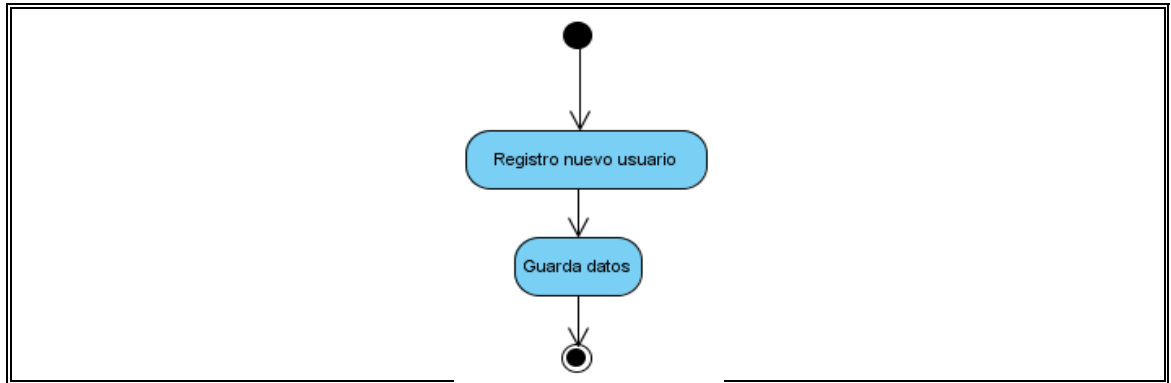


Figura 3-35: Diagrama de actividades del caso de uso CU7

### 3.2.3.8 Caso de uso CU8

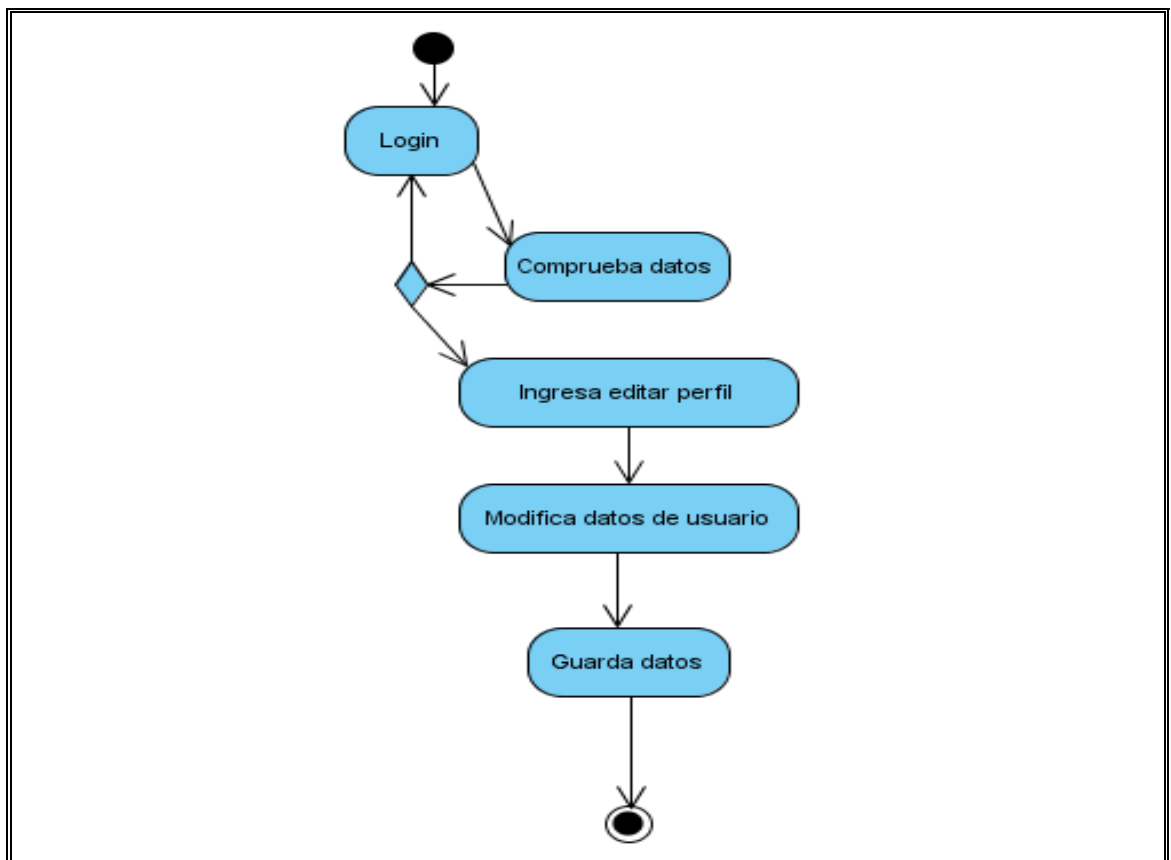


Figura 3-36: Diagrama de actividades del caso de uso CU8

### 3.2.3.9 Caso de uso CU9

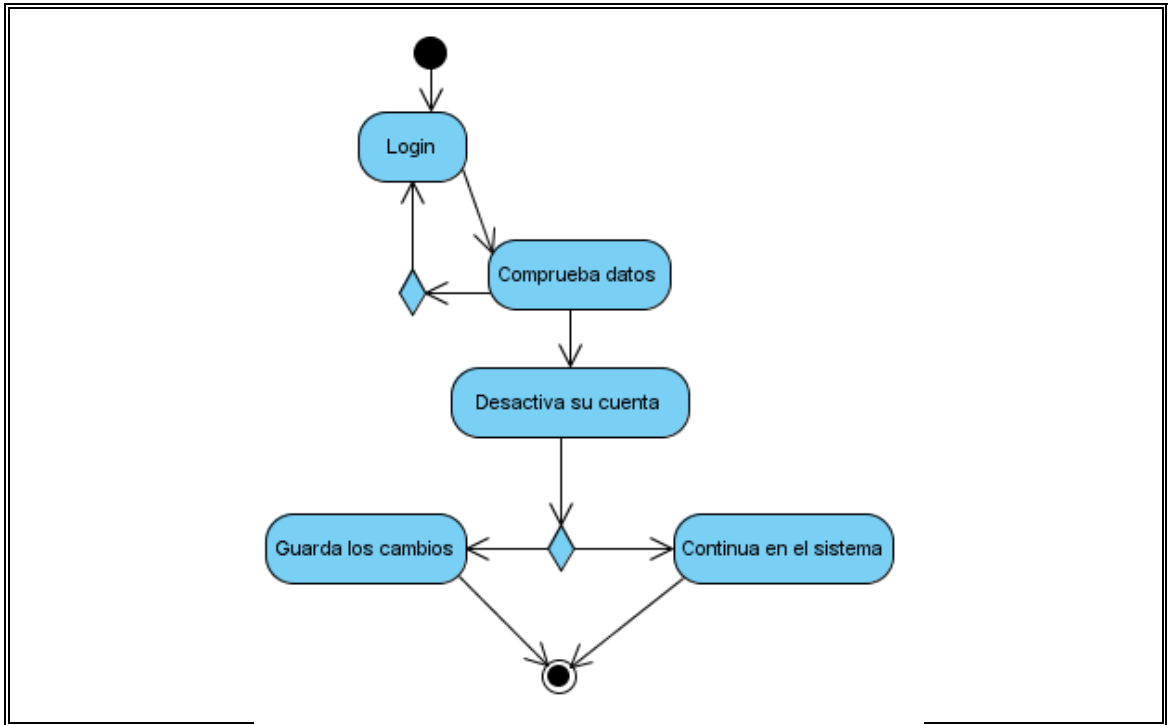


Figura 3-37: Diagrama de actividades del caso de uso CU9

### 3.2.3.10 Caso de uso CU10

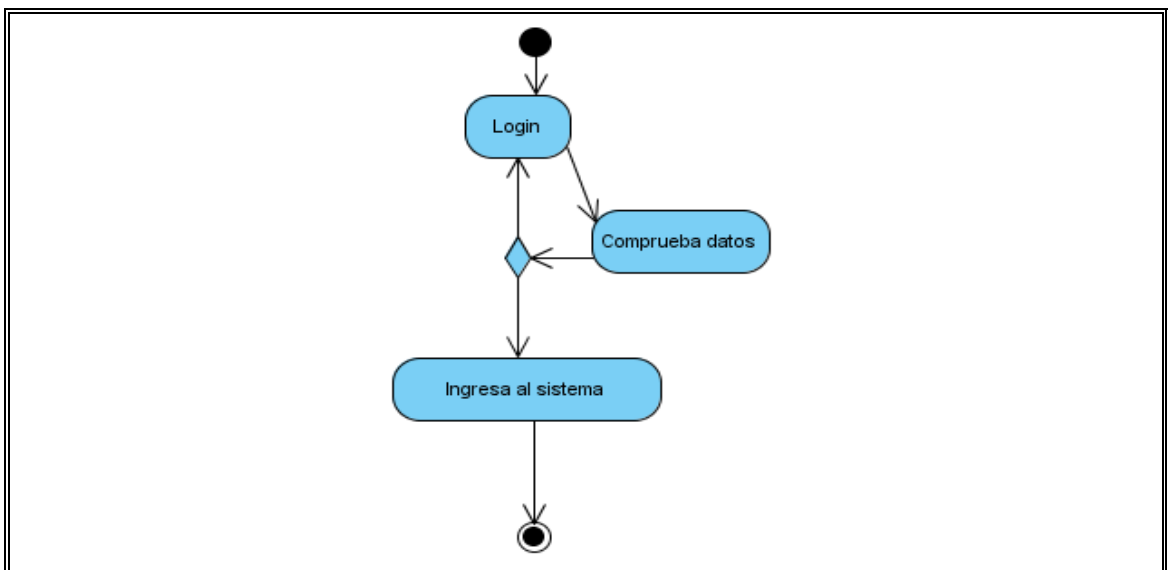


Figura 3-38: Diagrama de actividades del caso de uso CU10

### 3.2.3.11 Caso de uso CU11

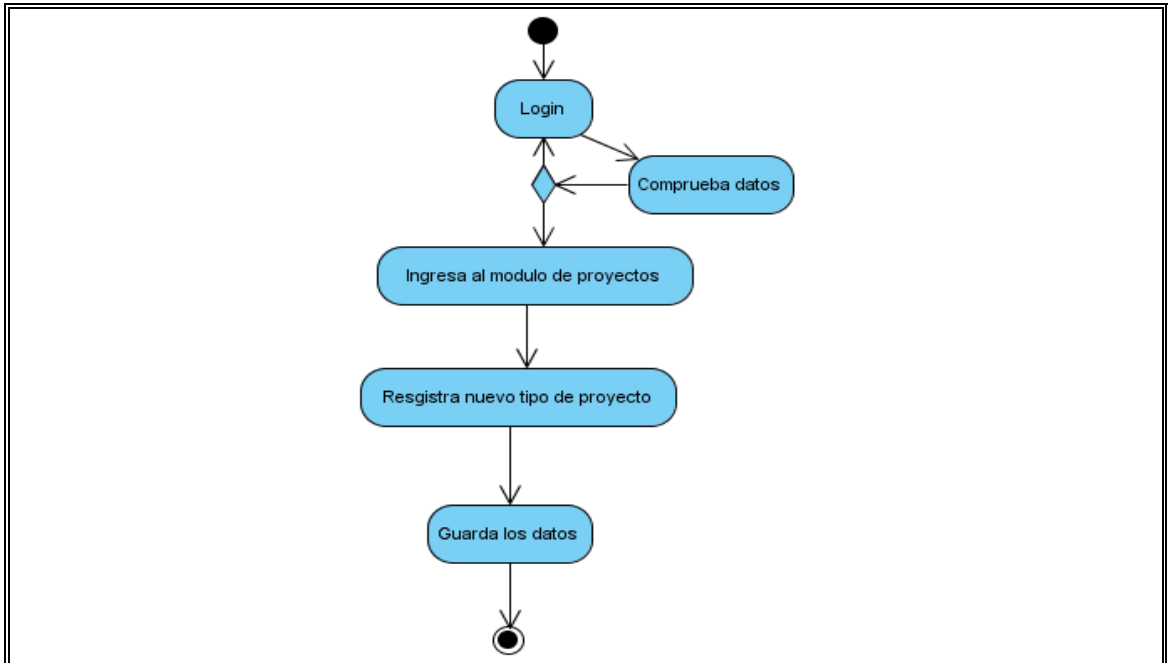


Figura 3-39: Diagrama de actividades del caso de uso CU11

### 3.2.3.12 Caso de uso CU12

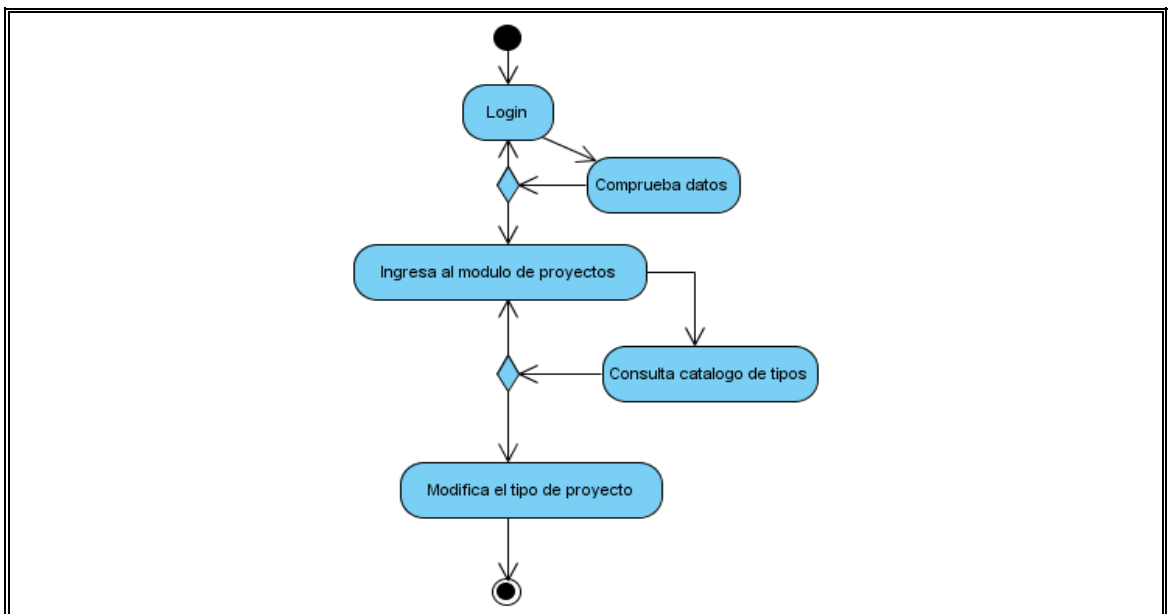


Figura 3-40: Diagrama de actividades del caso de uso CU12

### 3.2.3.13 Caso de uso CU13

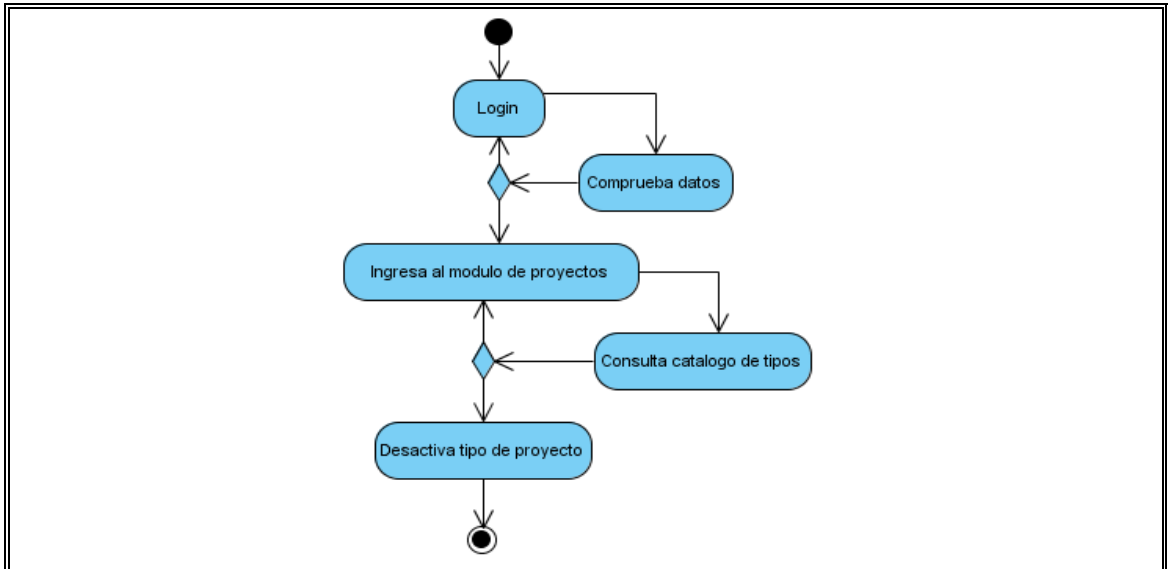


Figura 3-41: Diagrama de actividades del caso de uso CU13

### 3.2.3.14 Caso de uso CU14

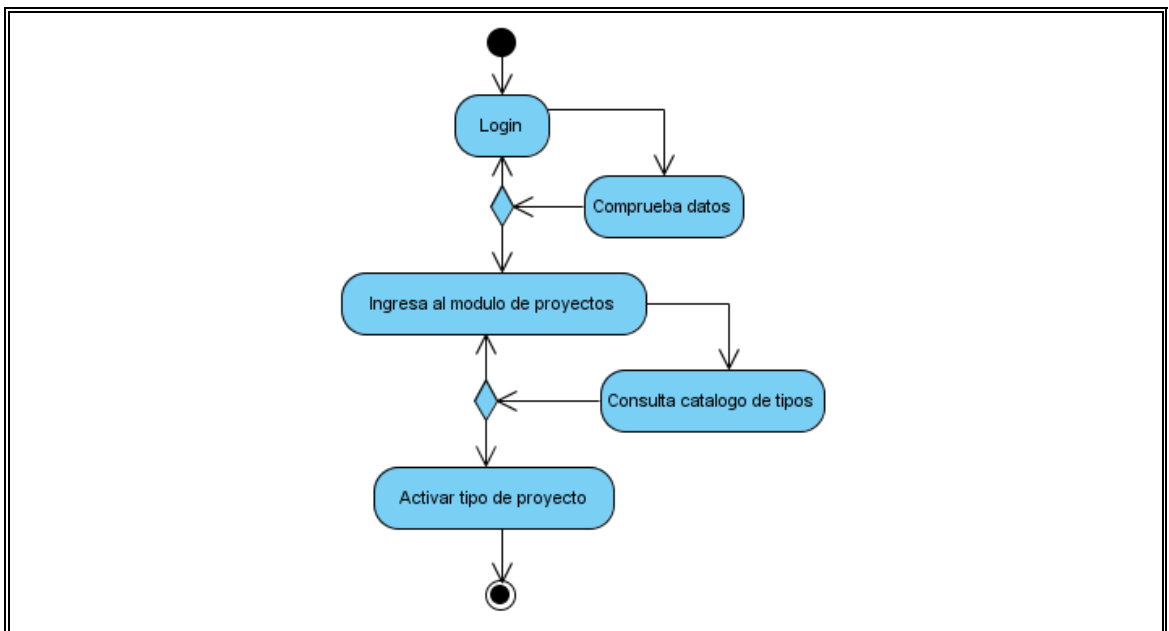


Figura 3-42: Diagrama de actividades del caso de uso CU14

**3.2.3.15 Caso de uso CU15**

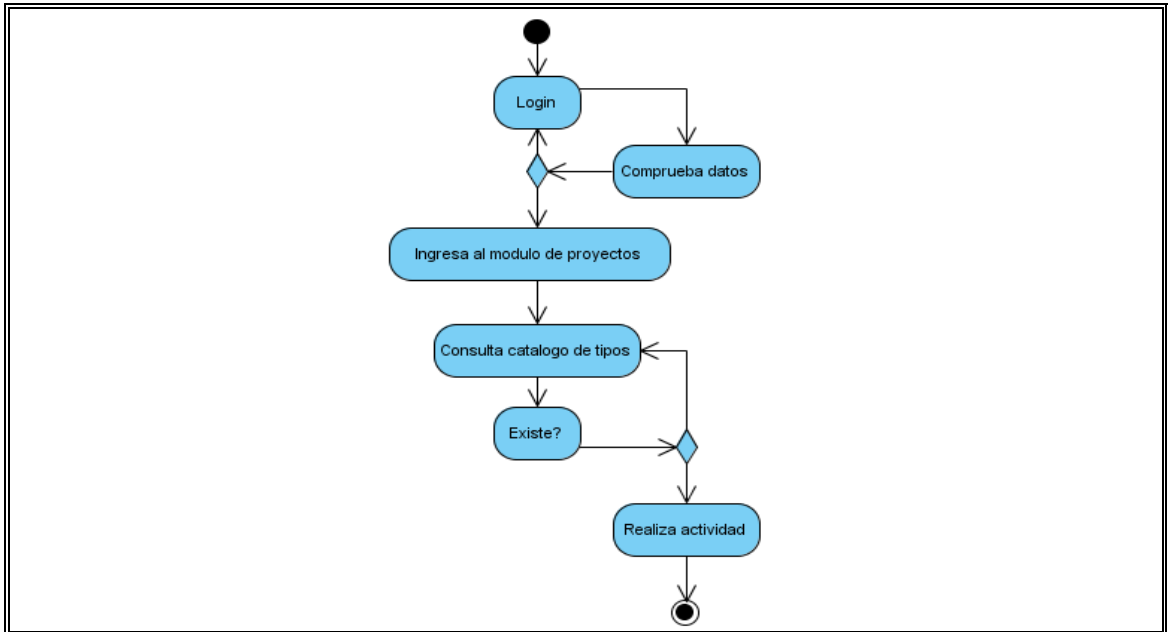


Figura 3-43: Diagrama de actividades del caso de uso CU15

**3.2.3.16 Caso de uso CU16, CU17 y CU18**

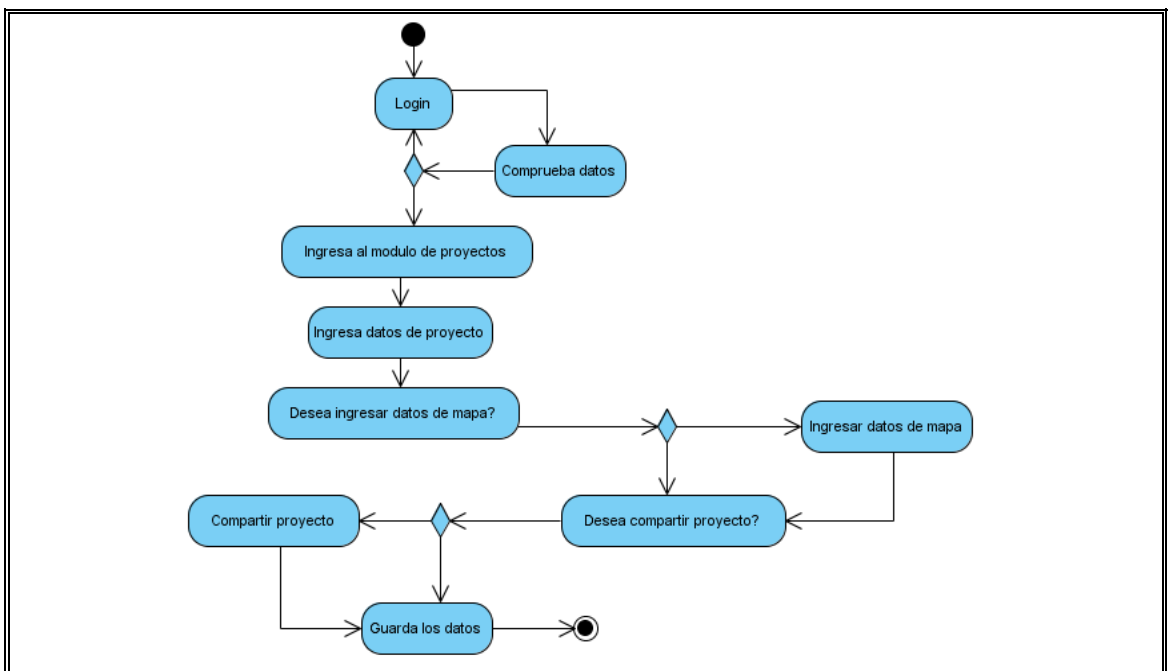


Figura 3-44: Diagrama de actividades del caso de uso CU16

**3.2.3.17 Caso de uso CU19 y CU20**

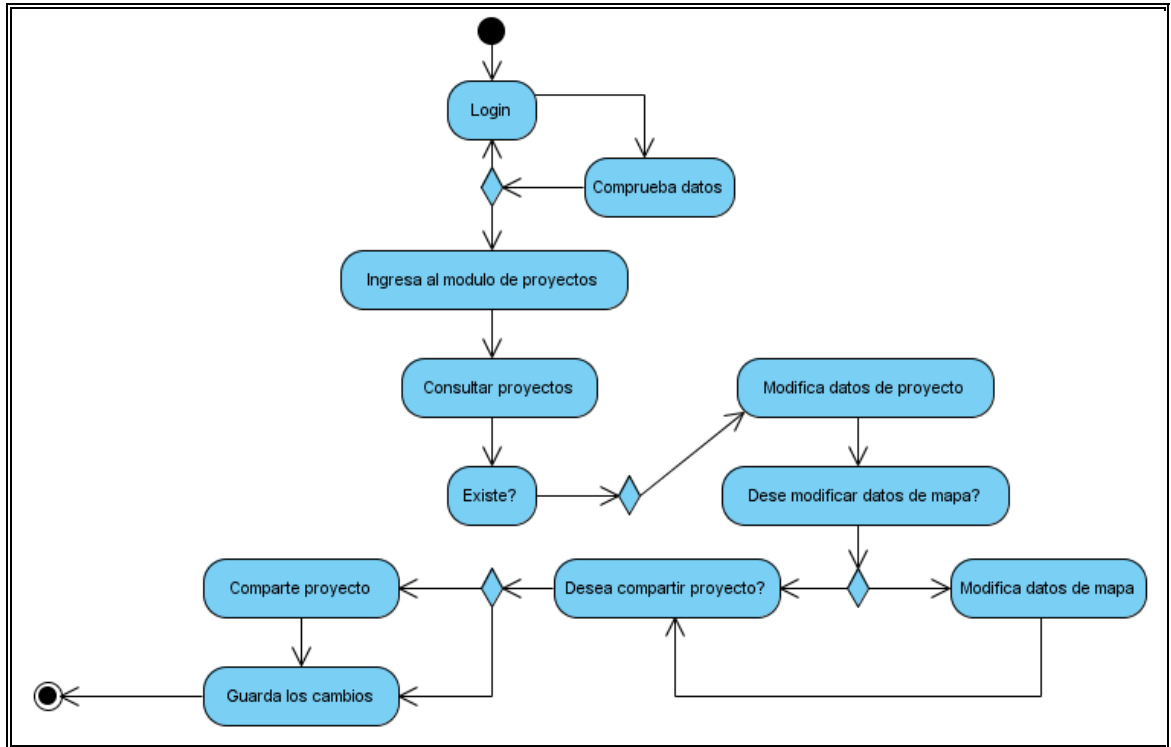


Figura 3-45: Diagrama de actividades del caso de uso CU19

**3.2.3.18 Caso de uso CU21**

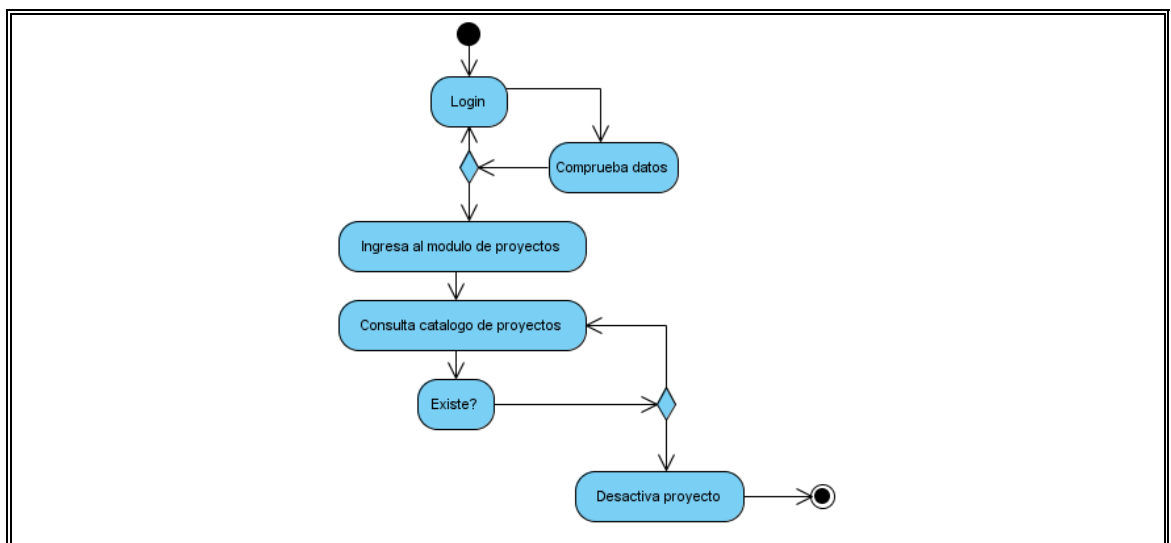


Figura 3-46: Diagrama de actividades del caso de uso CU21

### 3.2.3.19 Caso de uso CU22

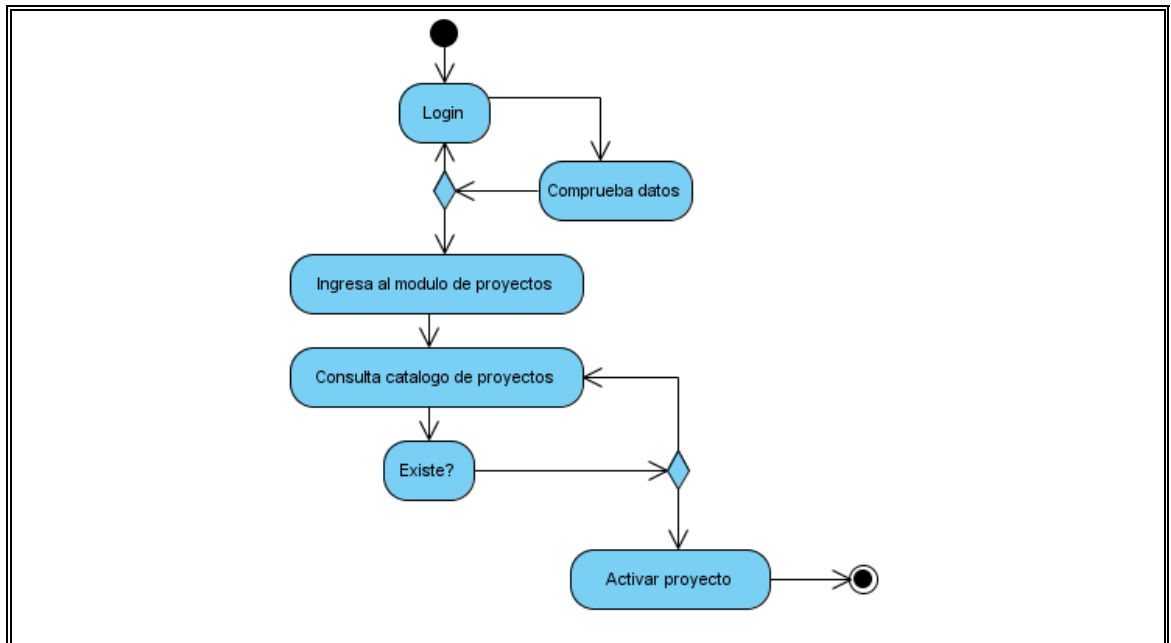


Figura 3-47: Diagrama de actividades del caso de uso CU22

### 3.2.3.20 Caso de uso CU23

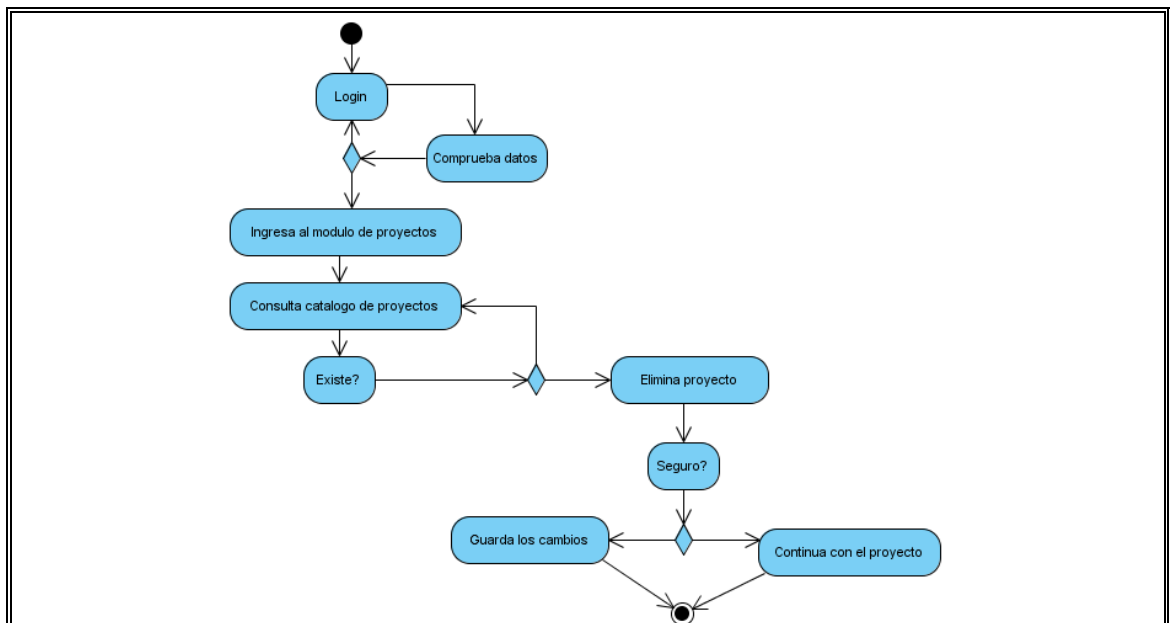


Figura 3-48: Diagrama de actividades del caso de uso CU23

### 3.2.3.21 Caso de uso CU24

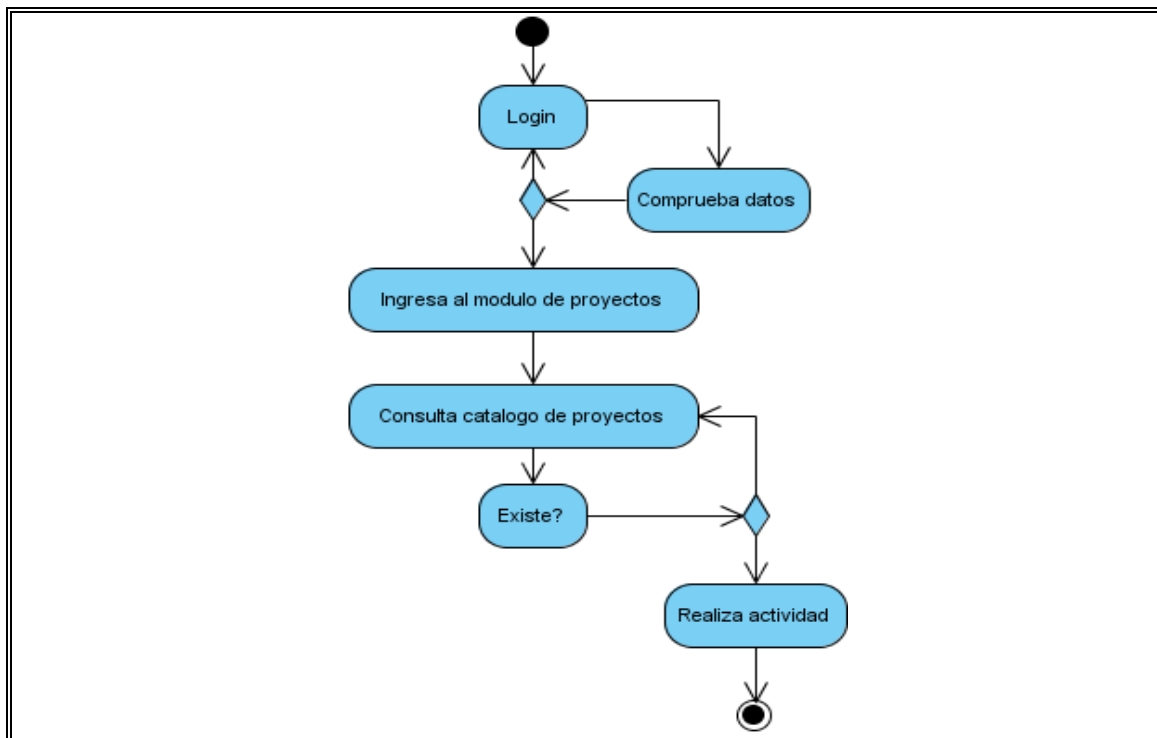


Figura 3-49: Diagrama de actividades del caso de uso CU24

### 3.2.3.22 Caso de uso CU25

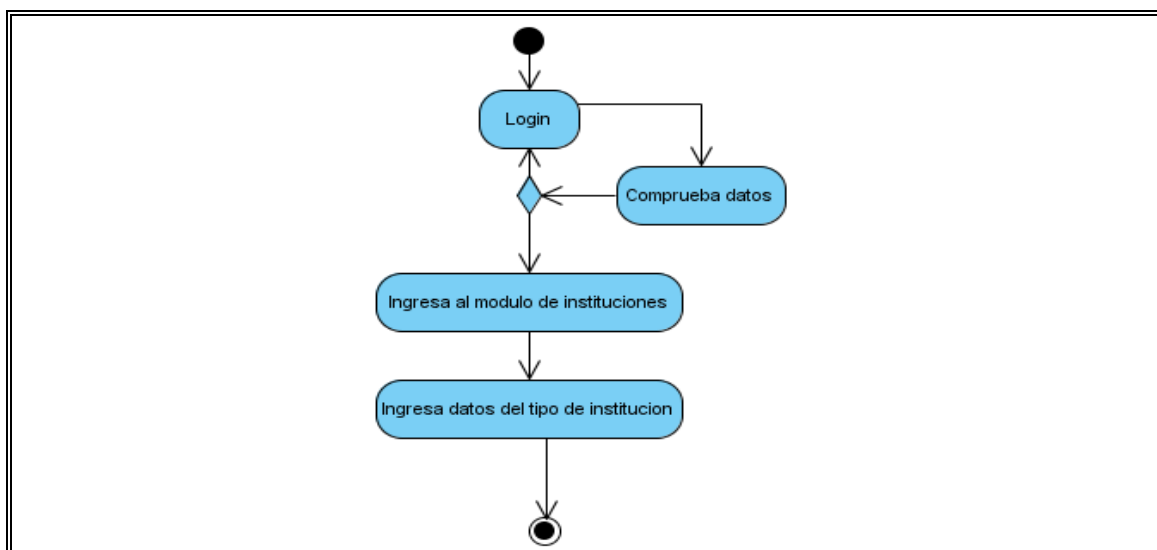


Figura 3-50: Diagrama de actividades del caso de uso CU25



### 3.2.3.23 Caso de uso CU26

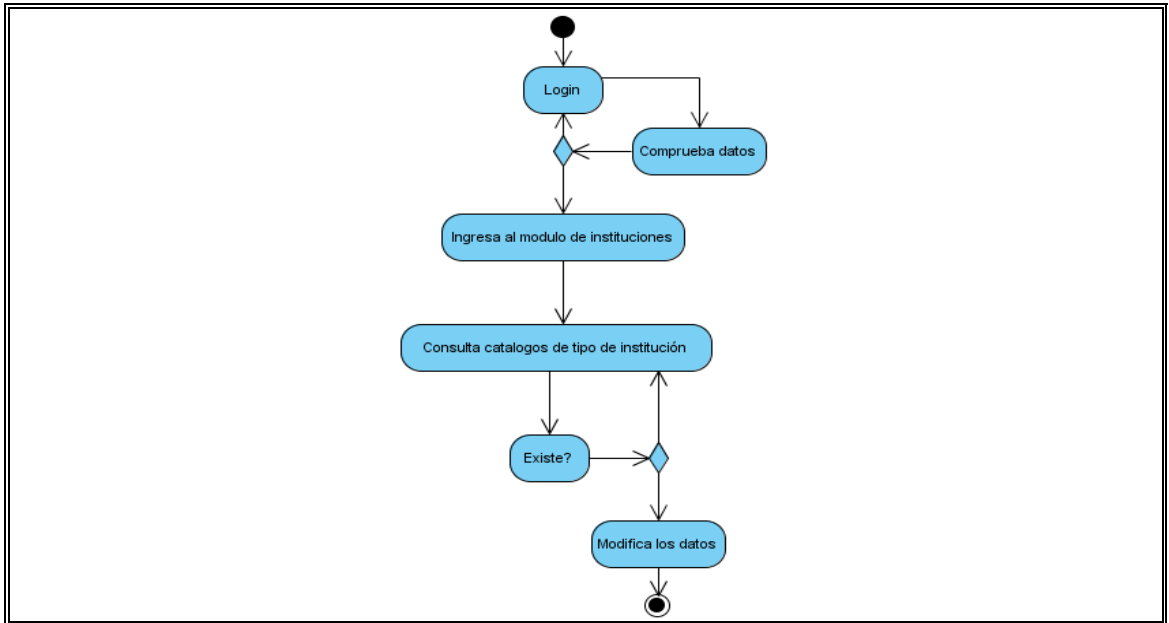


Figura 3-52: Diagrama de actividades del caso de uso CU26

### 3.2.3.24 Caso de uso CU27

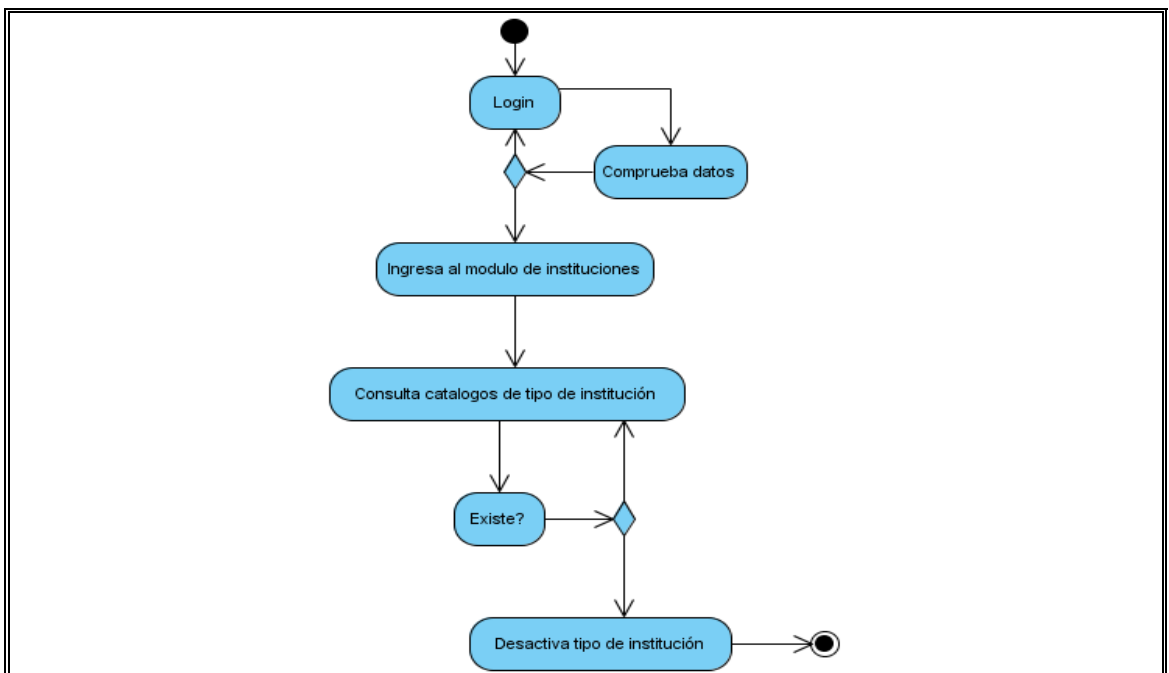


Figura 3-53: Diagrama de actividades del caso de uso CU27

### 3.2.3.25 Caso de uso CU28

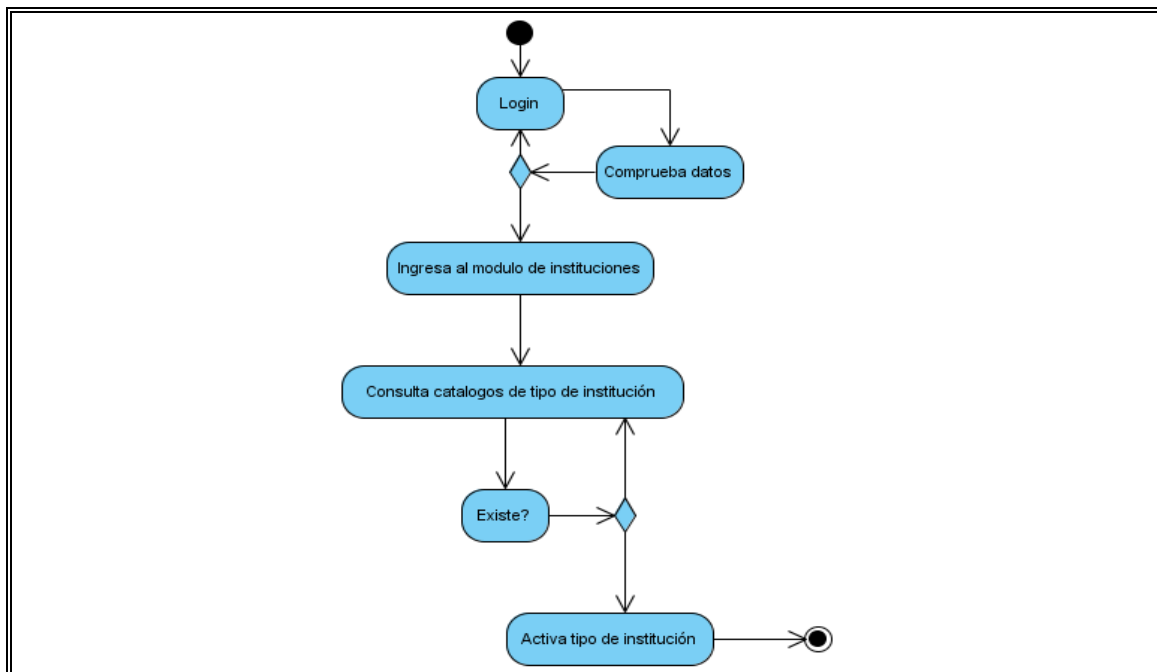


Figura 3-54: Diagrama de actividades del caso de uso CU28

### 3.2.3.26 Caso de uso CU29

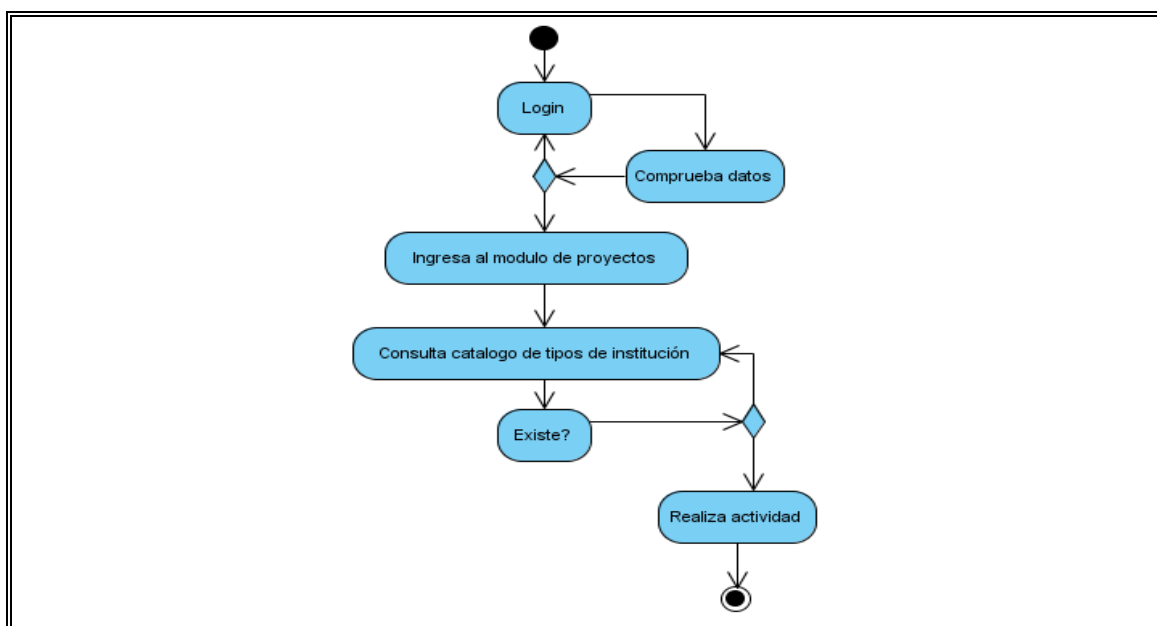


Figura 3-55: Diagrama de actividades del caso de uso CU29

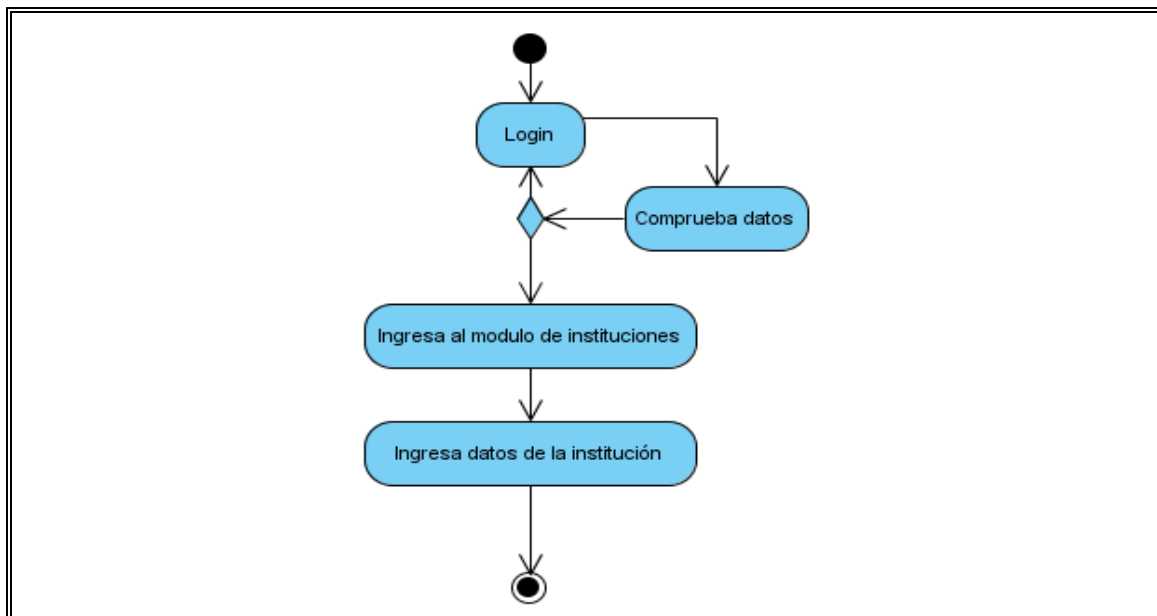
**3.2.3.27 Caso de uso CU30**

Figura 3-56: Diagrama de actividades del caso de uso CU30

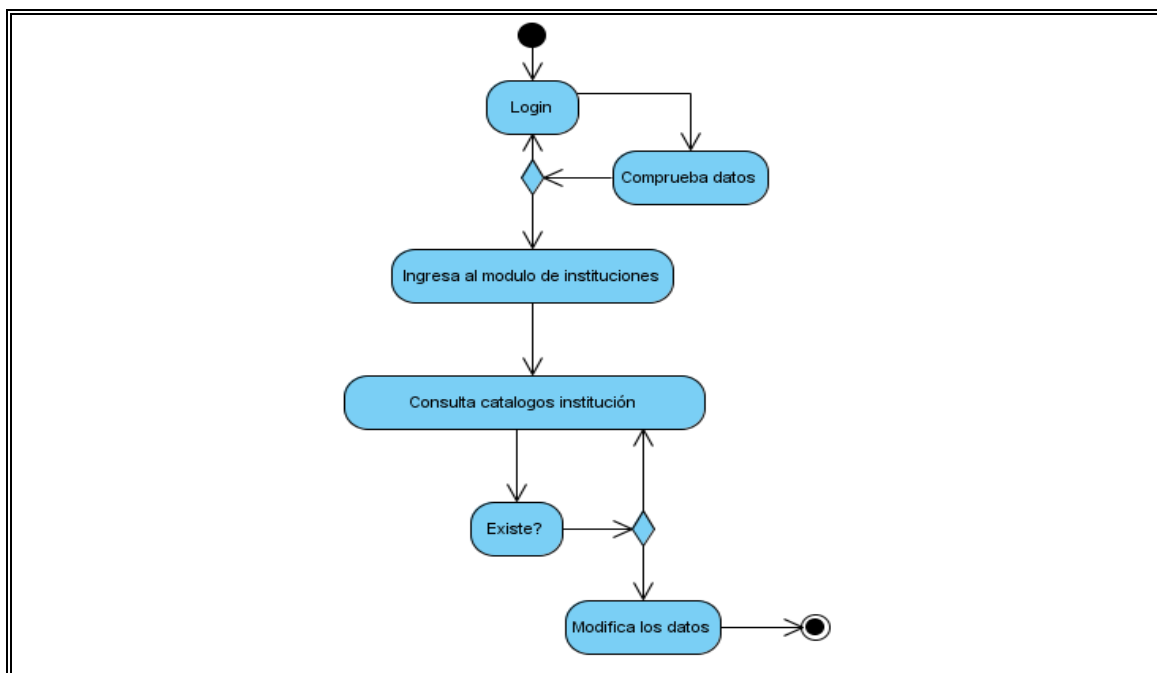
**3.2.3.28 Caso de uso CU31**

Figura 3-57: Diagrama de actividades del caso de uso CU31

### 3.2.3.29 Caso de uso CU32

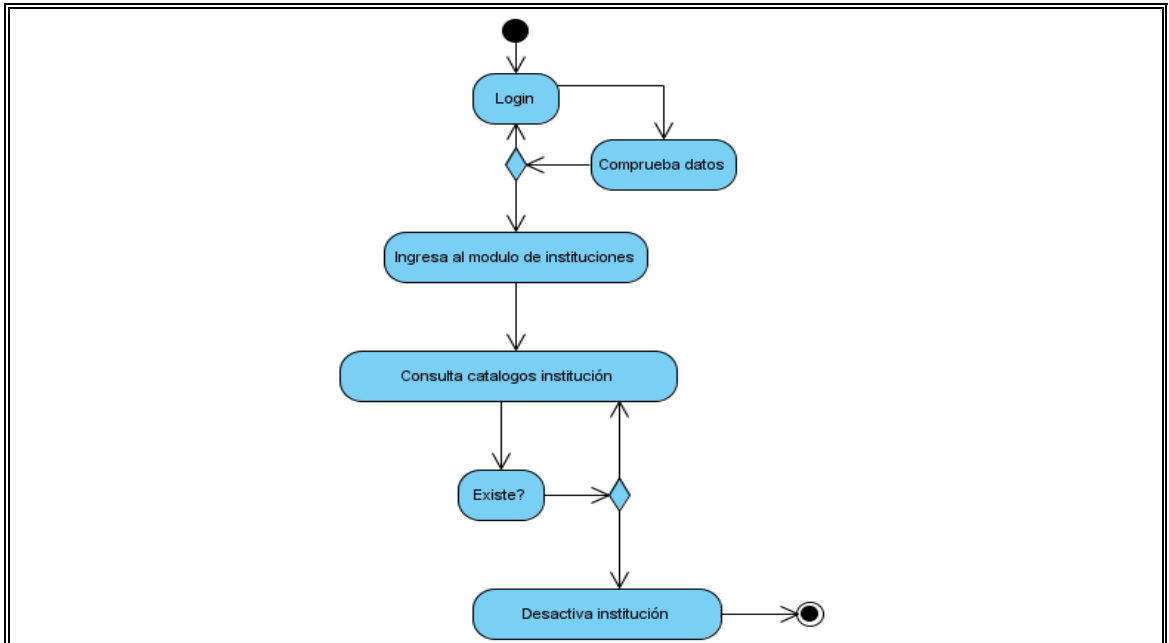


Figura 3-58: Diagrama de actividades del caso de uso CU32

### 3.2.3.30 Caso de uso CU33

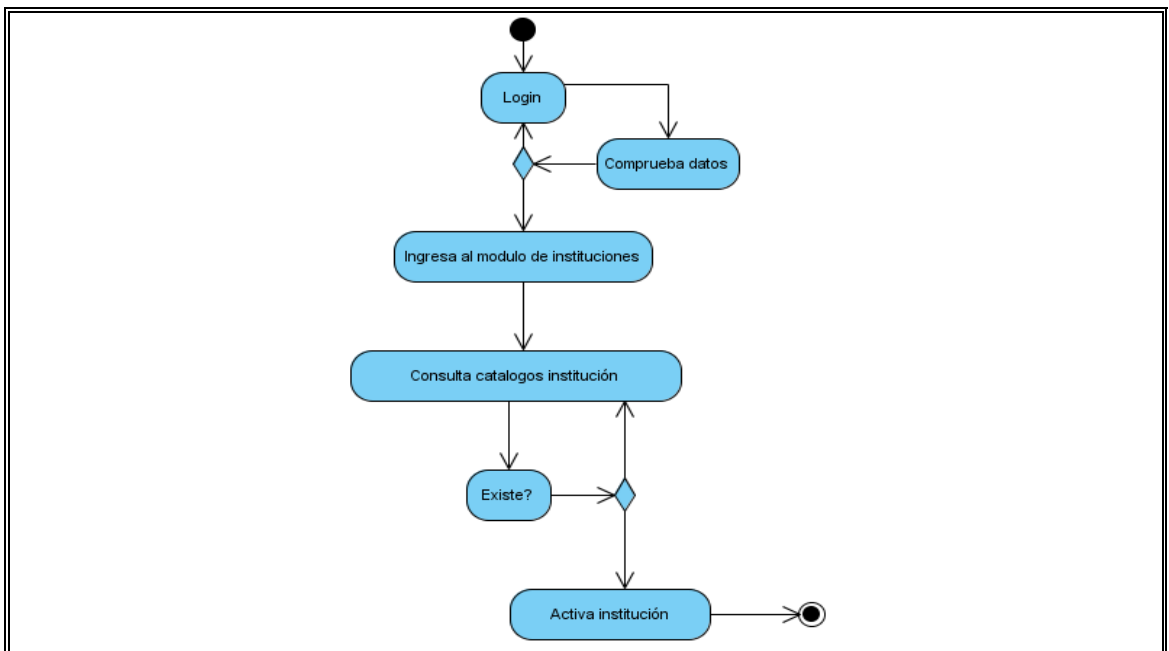


Figura 3-59: Diagrama de actividades del caso de uso CU33

### 3.2.3.31 Caso de uso CU34

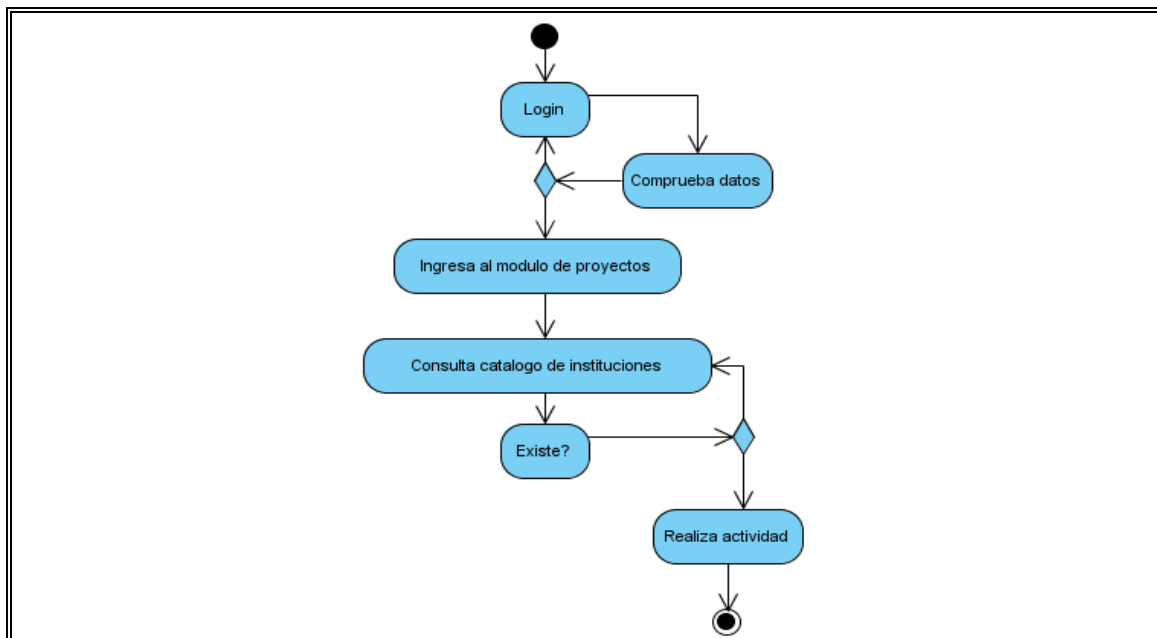


Figura 3-60: Diagrama de actividades del caso de uso CU34

### 3.2.3.32 Caso de uso CU35

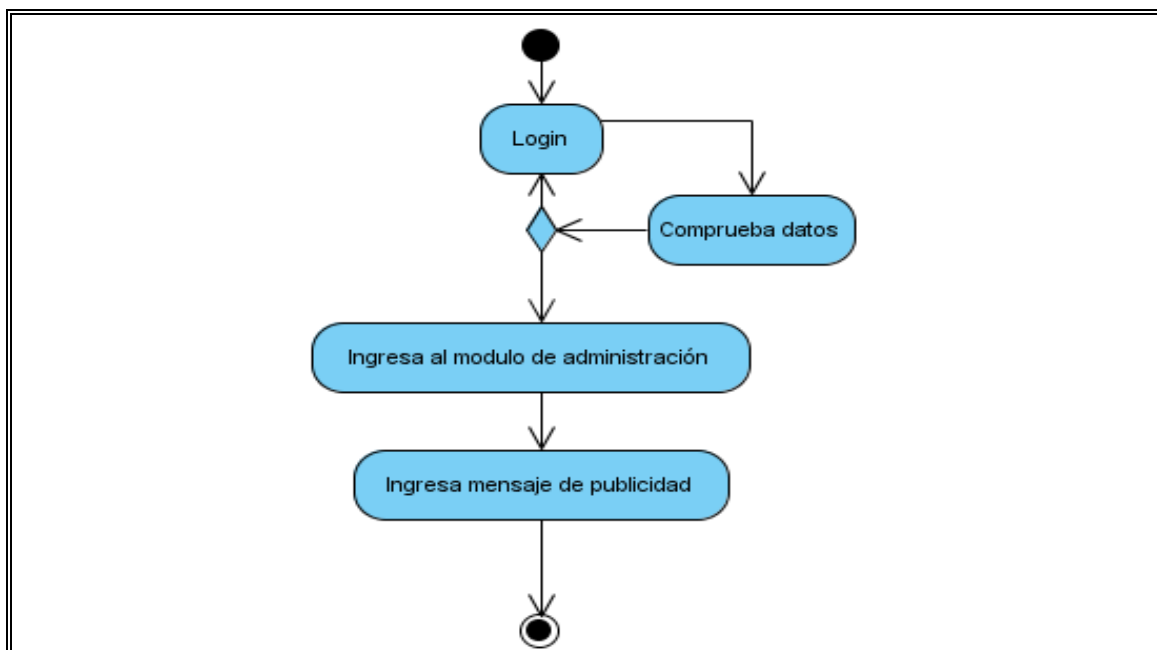


Figura 3-61: Diagrama de actividades del caso de uso CU35

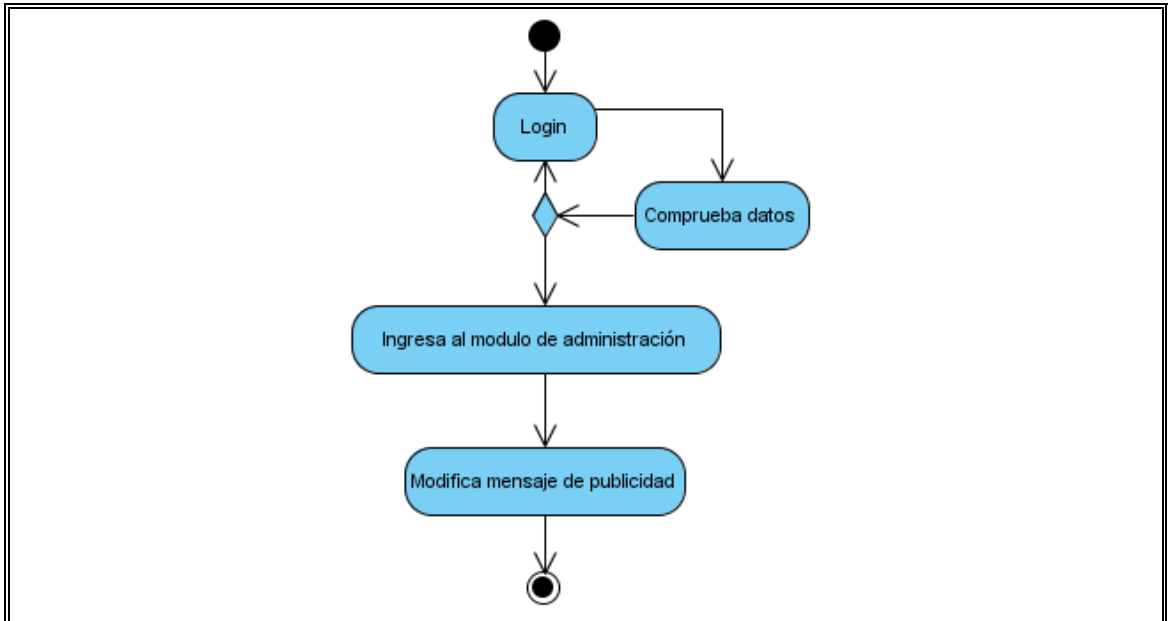
**3.2.3.33 Caso de uso CU36**

Figura 3-62: Diagrama de actividades del caso de uso CU36

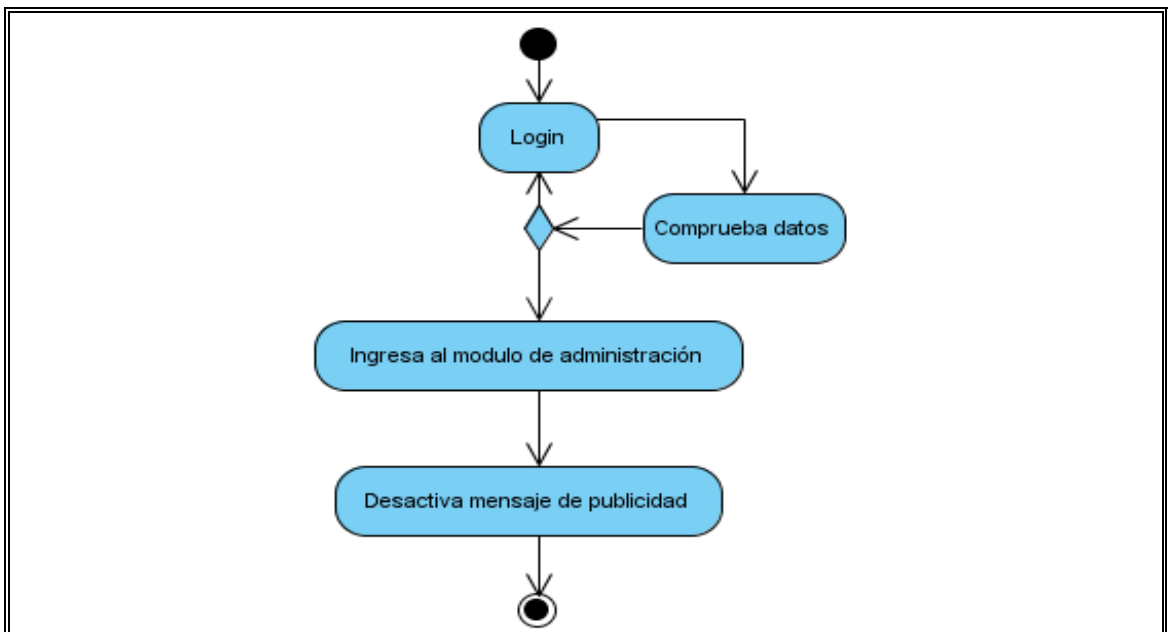
**3.2.3.34 Caso de uso CU37**

Figura 3-63: Diagrama de actividades del caso de uso CU37

### 3.2.3.35 Caso de uso CU38

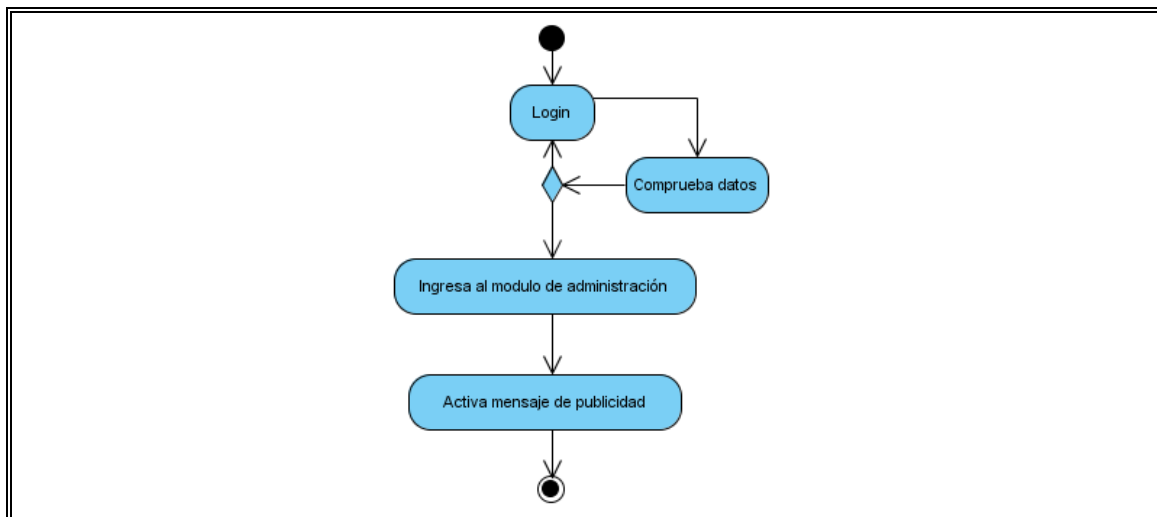


Figura 3-64: Diagrama de actividades del caso de uso CU38

### 3.2.3.36 Casos de uso CU39 – CU46

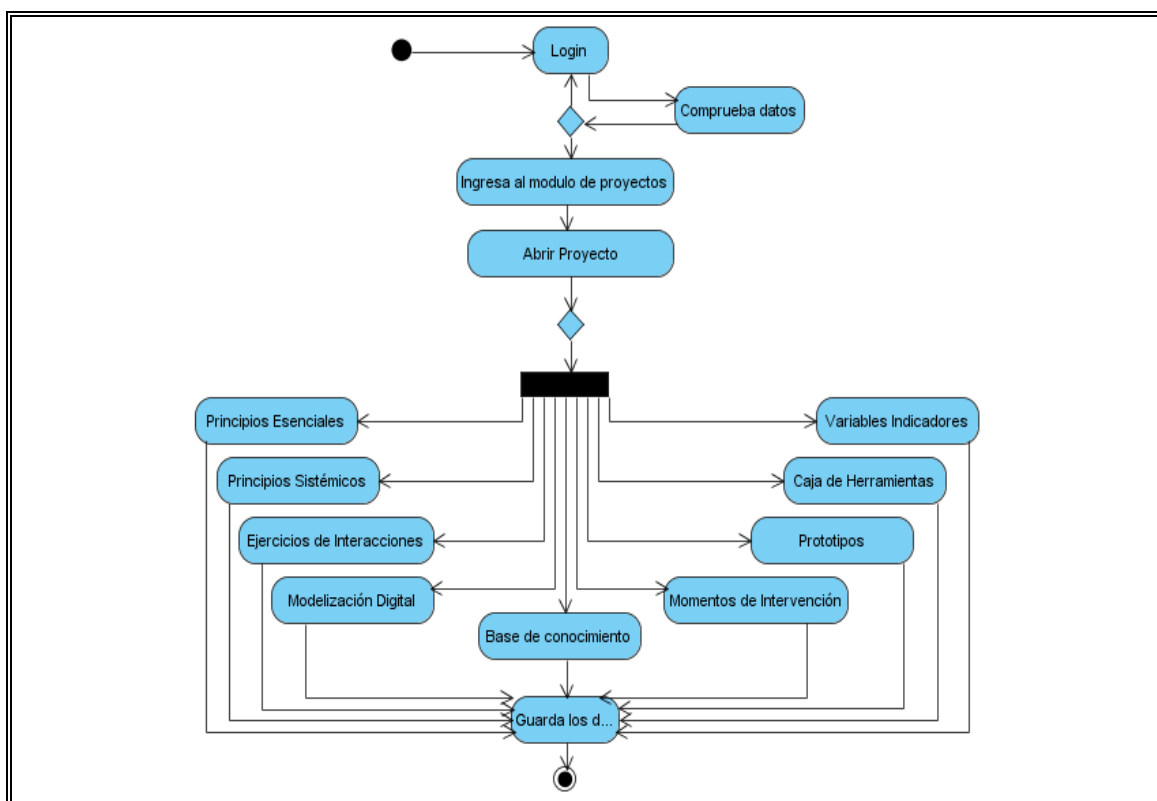


Figura 3-65: Diagrama de actividades del caso de uso CU39-46

## 3.2.3.37 Caso de uso CU47

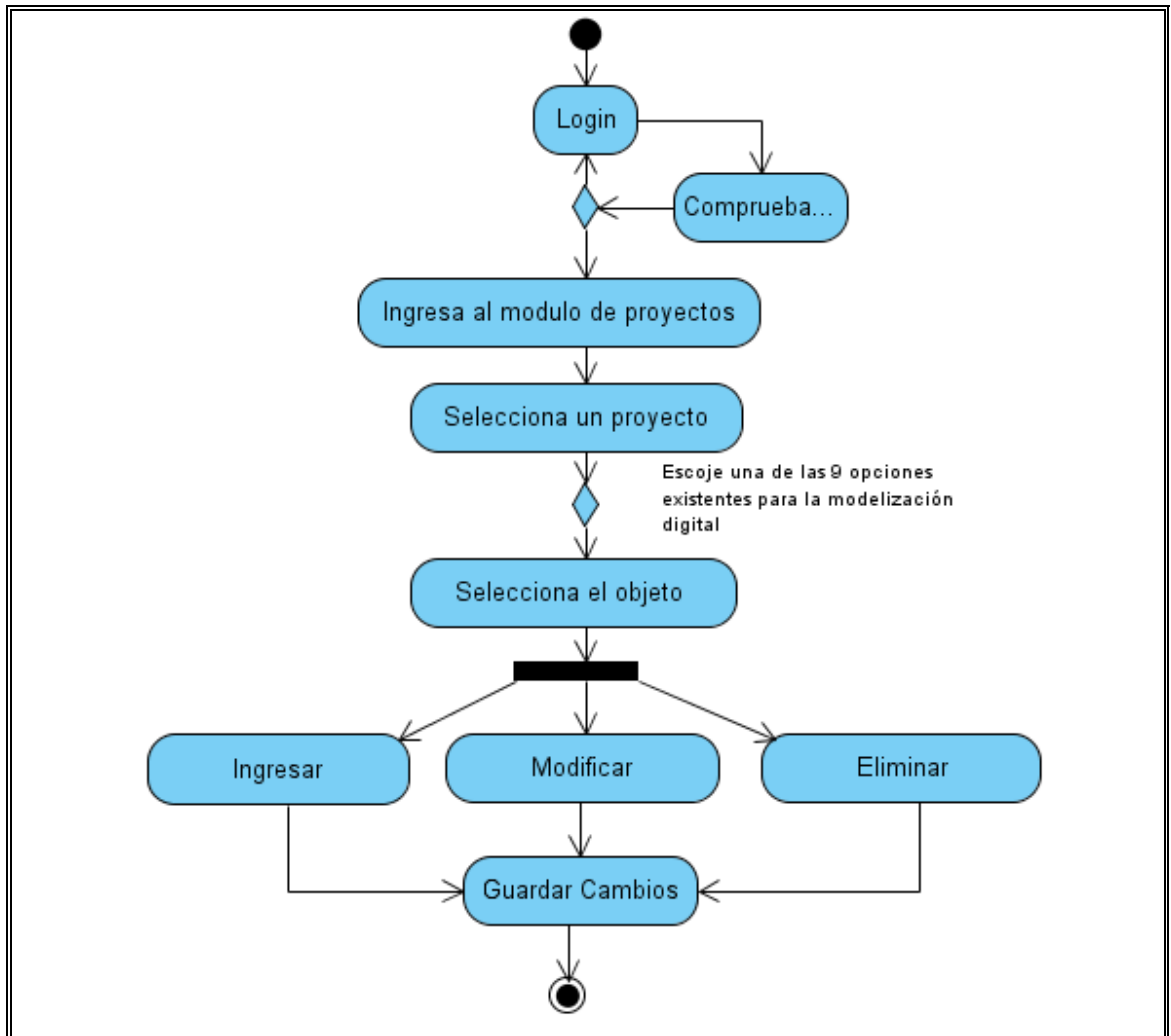


Figura 3-66: Diagrama de actividades del caso de uso CU47



## 3.2.3.38 Caso de uso CU48

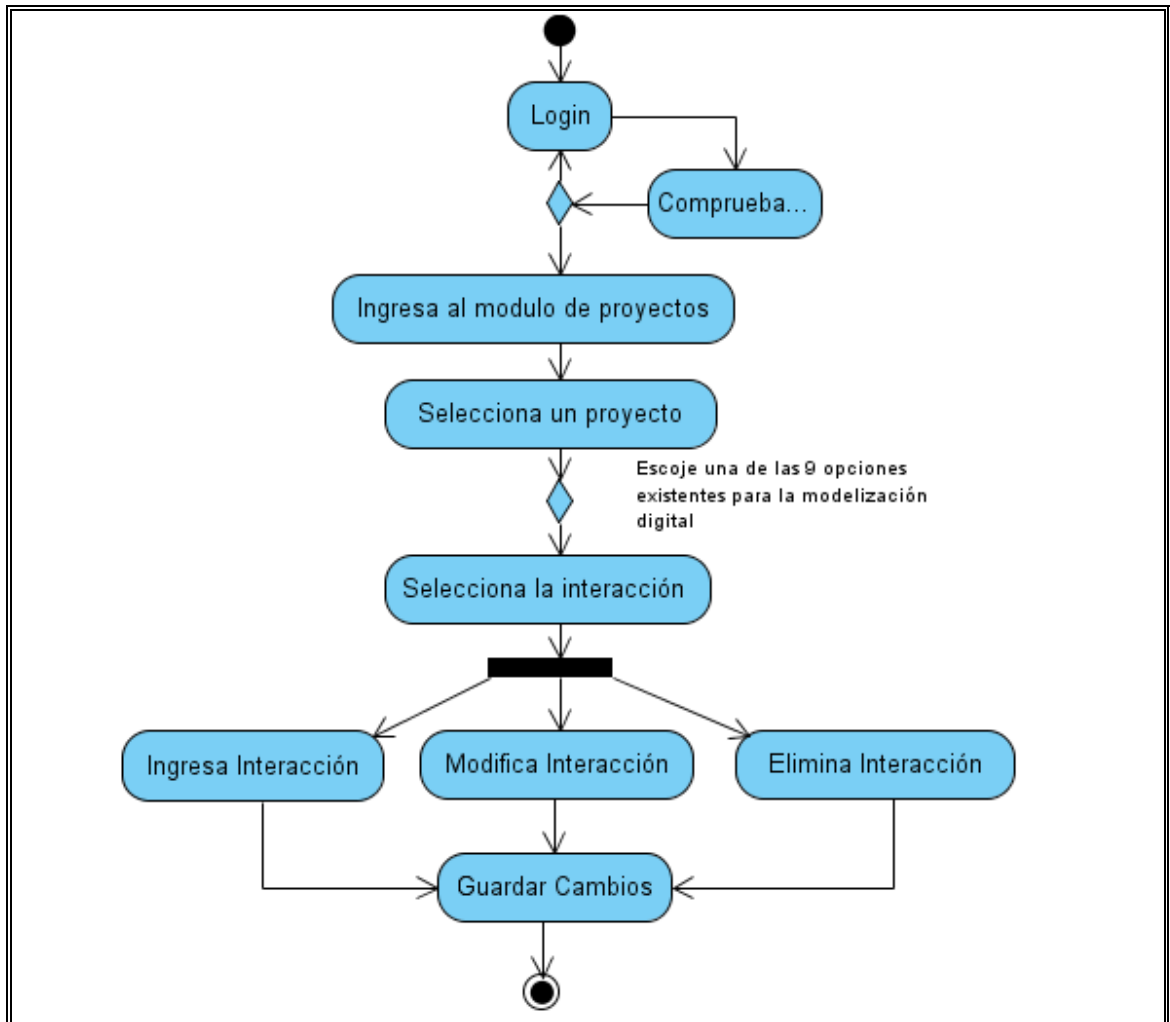


Figura 3-67: Diagrama de actividades del caso de uso 48

## 3.2.3.40 Caso de uso CU49

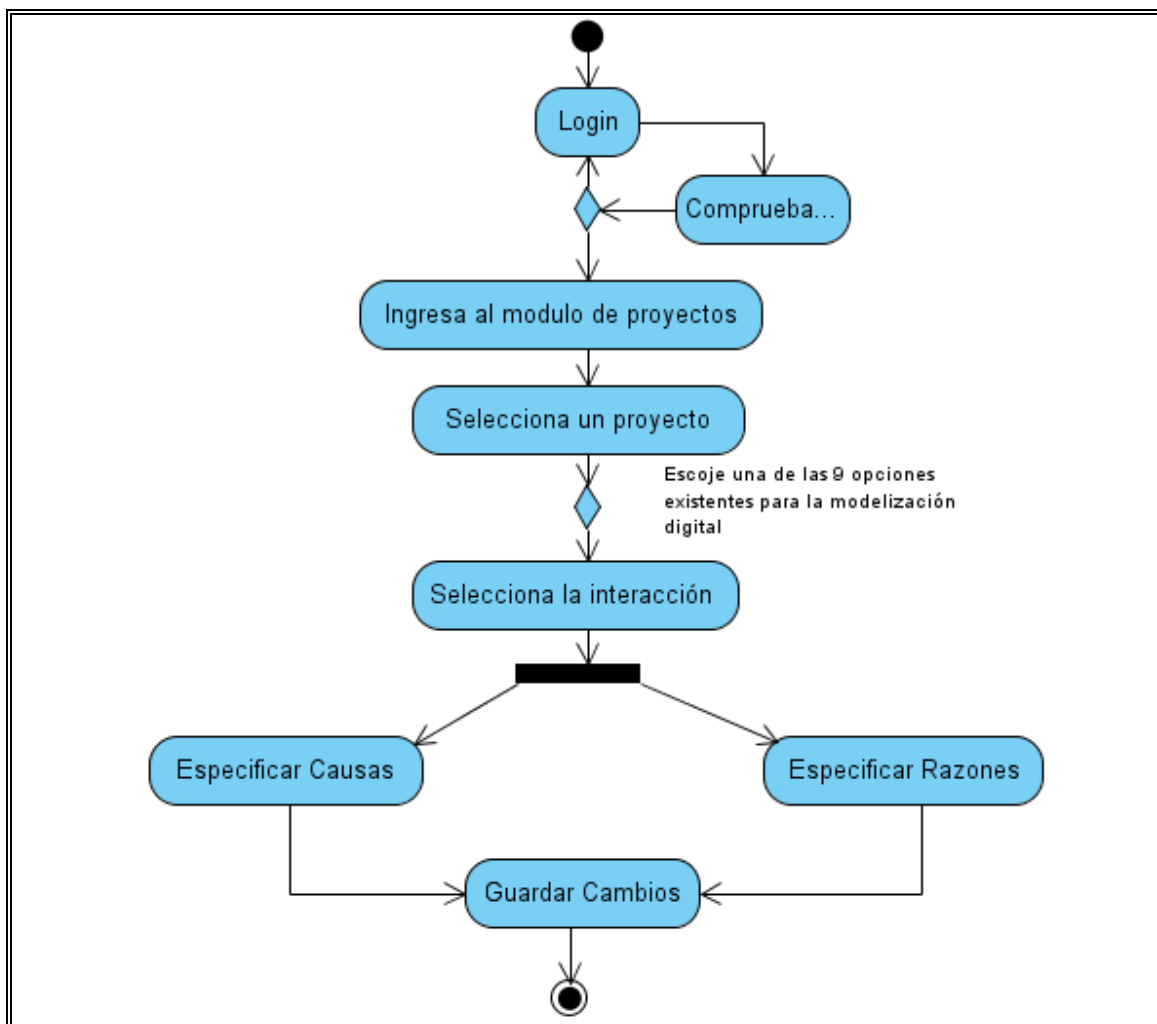


Figura 3-68: Diagrama de actividades del caso de uso CU49

### 3.2.3.41 Caso de uso CU50

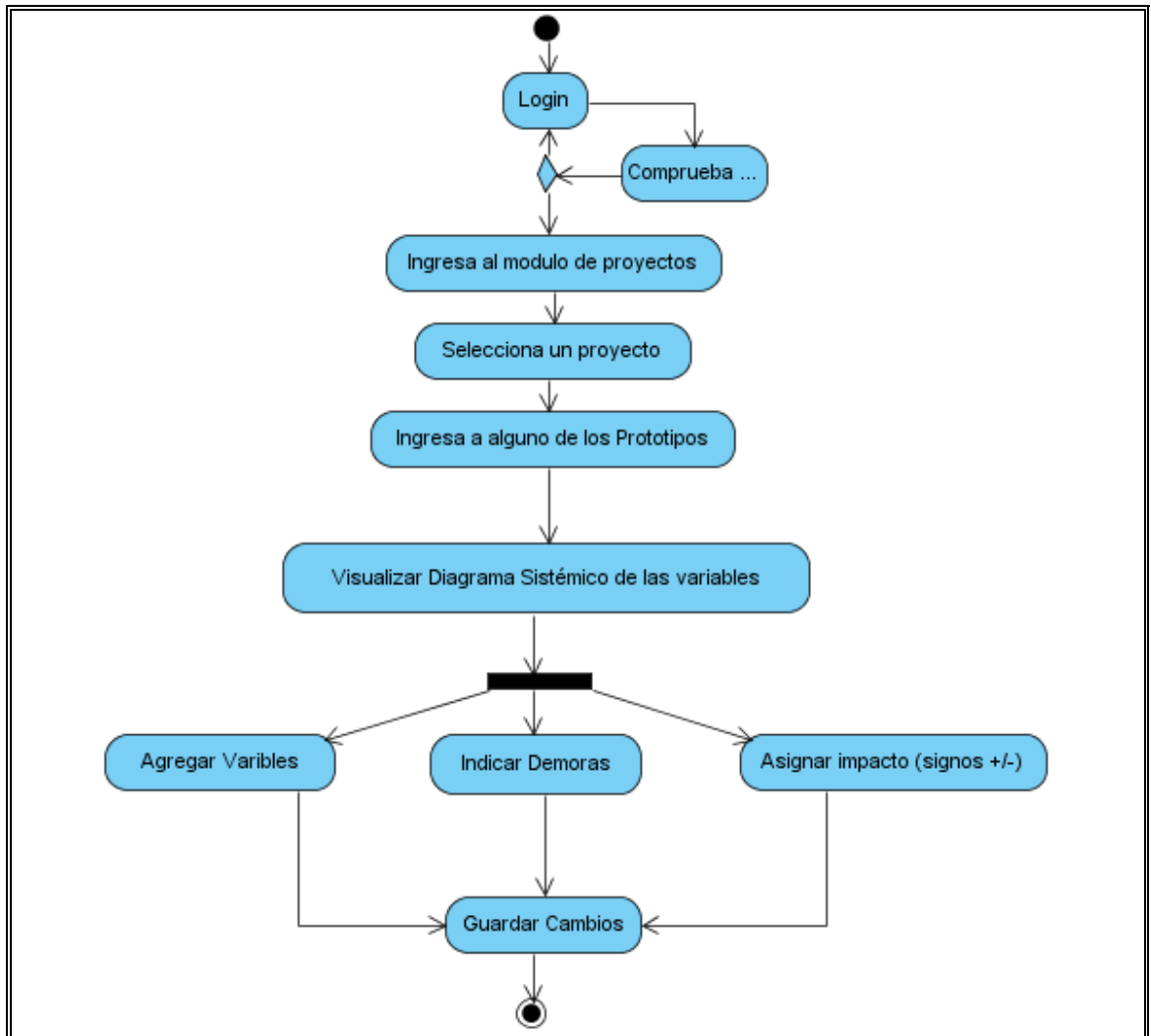


Figura 3-69: Diagrama de actividades del caso de uso CU510

## 3.2.3.42 Caso de uso CU52

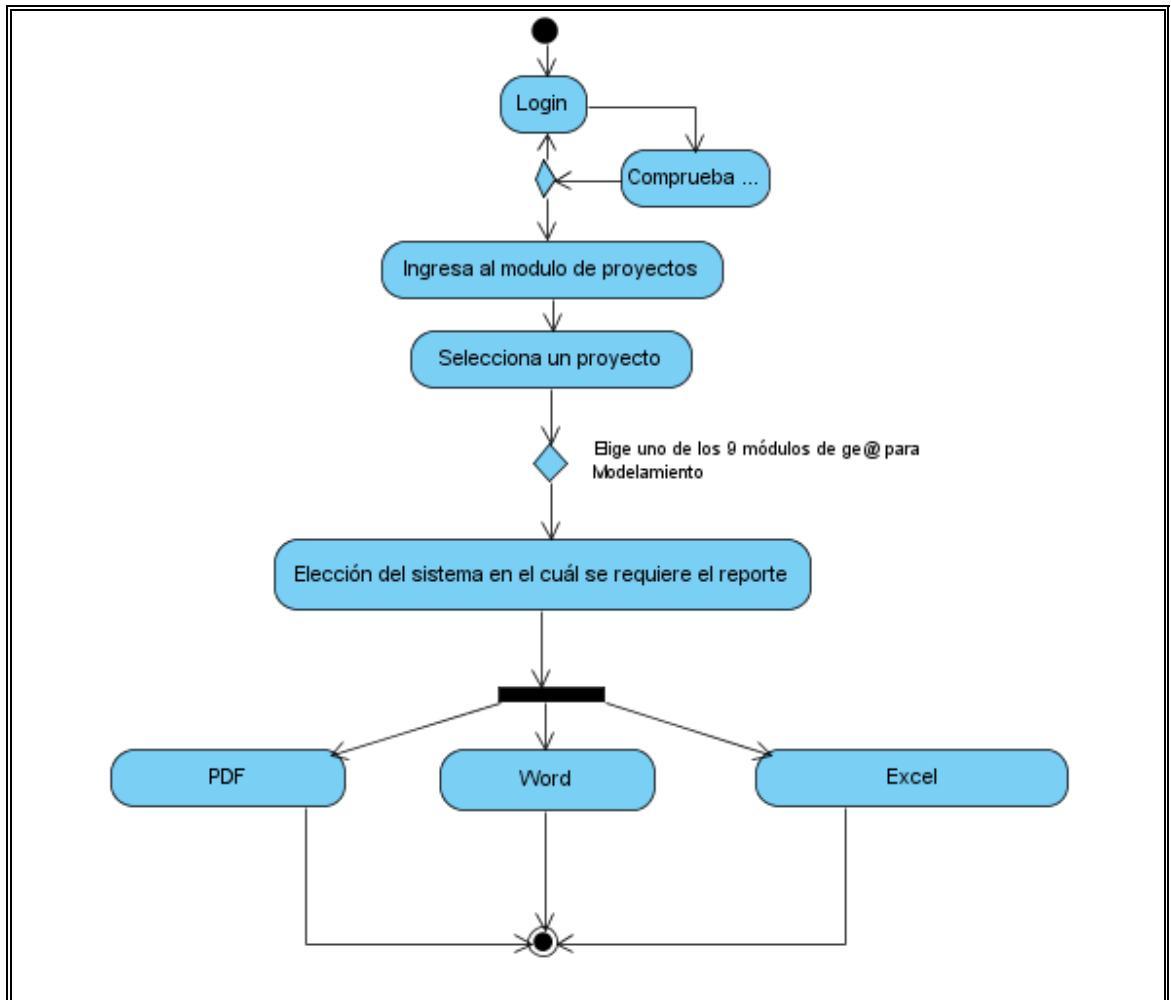


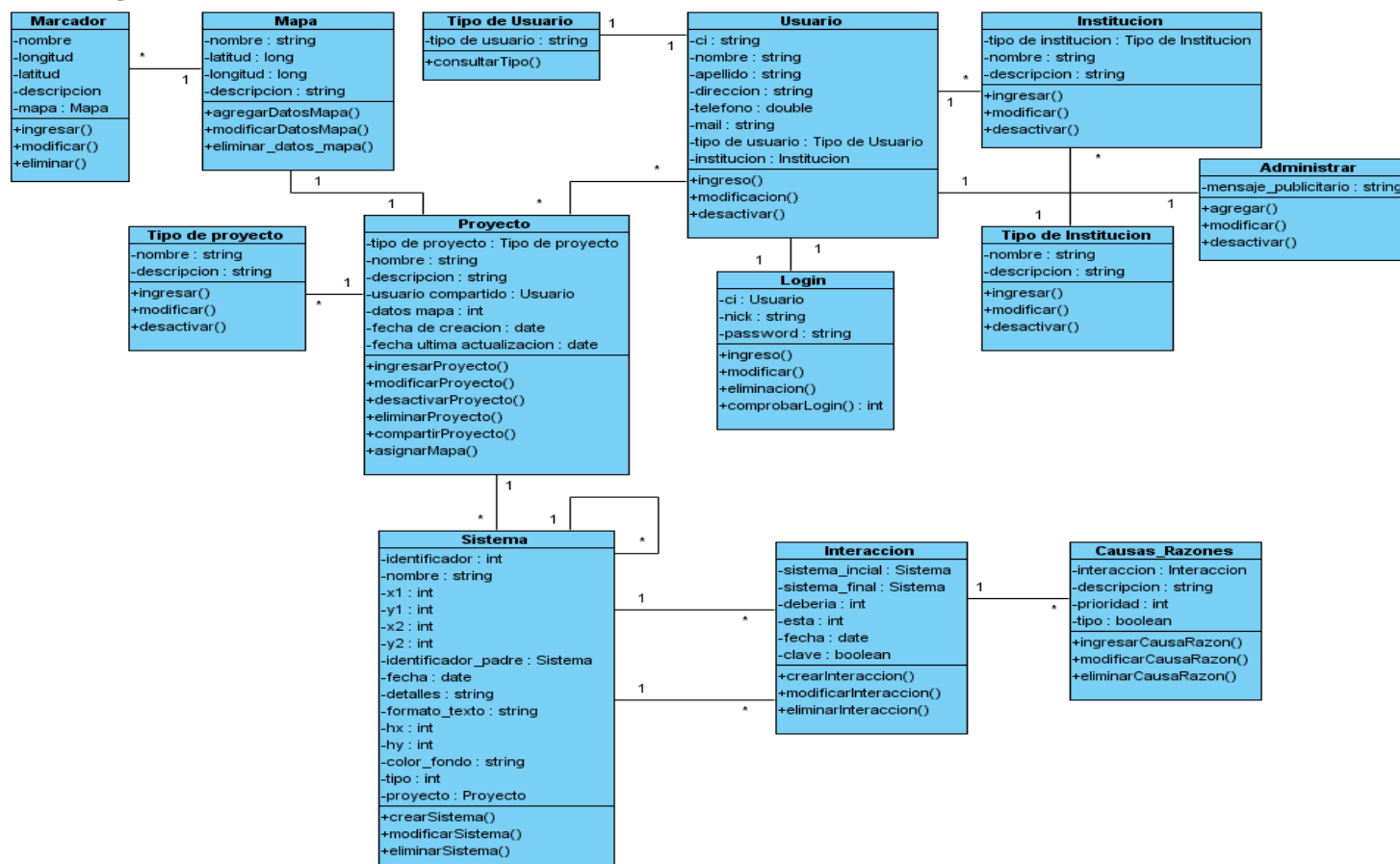
Figura 3-70: Diagrama de actividades del caso de uso CU51

### 3.3 Análisis y diseño

#### 3.3.1 Modelo de diseño

En esta fase se tiene una transición de la etapa de análisis a la de diseño, en donde se trata de definir un dispositivo, un proceso o como en este caso un sistema, con suficientes detalles como para permitir su interpretación y realización física. Para poder entender cómo se presentan las colaboraciones entre los módulos del sistema se va hacer uso del diagrama de clases ya que cada actividad que se realiza en cada módulo del sistema puede ser representada por una clase.

##### 3.3.1.1 Diagrama de Clases



### 3.3.2 Modelo de datos: Diagrama de Entidad-Relación y Diccionario

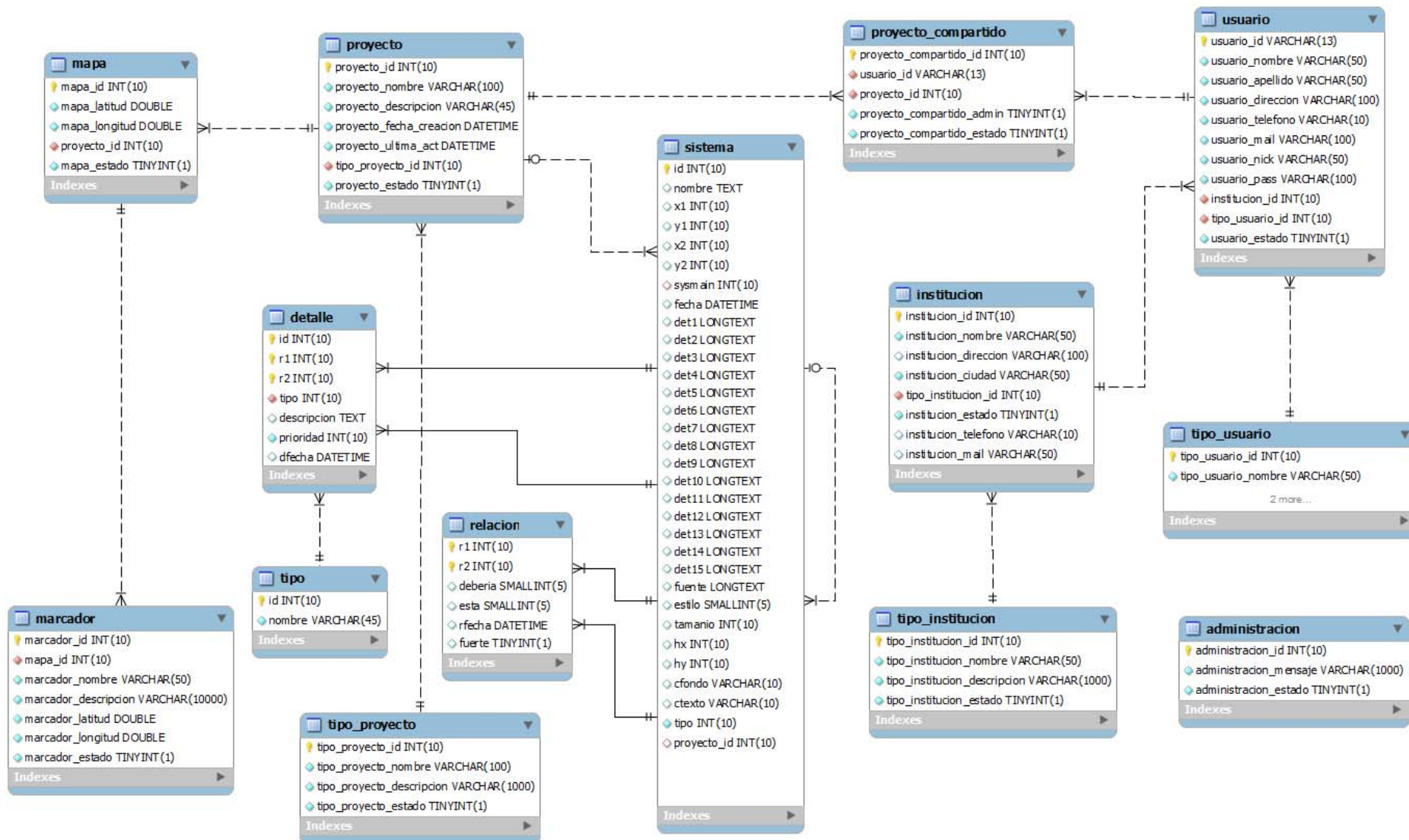


TABLA	CAMPO	TIPO	TABLA RELACIONADA	CAMPO RELACIONADO	DESCRIPCIÓN
<b>tipo_institucion</b>		Entidad			Entidad que almacena información sobre los tipos de institución existentes
	tipo_institucion_id	INT(10) <b>PK</b>			Código del tipo de institución
	tipo_institucion_nombre	VARCHAR(50)			Nombre del tipo de institución
	tipo_institucion_descripcion	VARCHAR(400)			Descripción del tipo de institución
	tipo_institucion_estado	TINYINT(1)			Estado del tipo de institución (1 activo, 0 desactivado/eliminado)
<b>institucion</b>		Entidad			Entidad que almacena información acerca de las instituciones
	institucion_id	INT(10) <b>PK</b>			Código de la institución
	institucion_nombre	VARCHAR(50)			Nombre de la institución
	institucion_direccion	VARCHAR(100)			Dirección de la institución
	institucion_ciudad	VARCHAR(50)			Ciudad sede de la institución
	tipo_institucion_id	INT(10)	tipo_institucion	tipo_institucion_id	Llave foránea que especifica de que tipo es la institución
	institucion_estado	TINYINT(1)			Estado de la institución
	institucion_telefono	VARCHAR(10)			Teléfono de la institución
	institucion_mail	VARCHAR(50)			Mail de la institución

tipo_usuario		Entidad			Entidad que almacena información sobre los roles de usuario del sistema
	tipo_usuario_id	INT(10) <b>PK</b>			Código del tipo de usuario
	tipo_usuario_nombre	VARCHAR(50)			Nombre del tipo de usuario (roles de usuario como administrador, invitado, técnico)
usuario		Entidad			Entidad que almacena todos los usuarios del sistema
	usuario_id	VARCHAR(13) <b>PK</b>			Código del usuario
	usuario_nombre	VARCHAR(50)			Nombre del usuario
	usuario_apellido	VARCHAR(50)			Apellido del usuario
	usuario_direccion	VARCHAR(100)			Dirección del usuario
	usuario_telefono	VARCHAR(10)			Teléfono del usuario
	usuario_mail	VARCHAR(100)			Mail del usuario
	usuario_nick	VARCHAR(50)			Nombre con el cual el usuario ingresa
	usuario_pass	VARCHAR(100)			Contraseña con la que el usuario ingresa a la herramienta web
	institución_id	INT(10) <b>FK</b>	institución	institución_id	Vincula al usuario con la institución a la que pertenece



	tipo_usuario	INT(10) <b>FK</b>	usuario	tipo_usuario_id	Identifica el rol que tiene ese usuario en la herramienta web
	usuario_estado	TINYINT(1)			Estado del tipo de institución (1 activo, 0 desactivado/eliminado)
tipo_proyecto		Entidad			Entidad que almacena información acerca de los tipos de proyectos que pueden desarrollarse
	tipo_proyecto_id	INT(10) <b>PK</b>			Código del tipo de proyecto
	tipo_proyecto_nombre	VARCHAR(100)			Nombre del tipo de proyecto (modelo de aplicación. Ejm: Ordenamiento Territorial)
	tipo_proyecto_descripcion	VARCHAR(100)			Descripción del tipo de proyecto
	tipo_proyecto_estado	TINYINT(100)			Estado (1 activo, 0 desactivado/eliminado)
proyecto		Entidad			Entidad que almacena los proyectos que gestiona la herramienta web
	proyecto_id	INT(10) <b>PK</b>			Código del proyecto
	proyecto_nombre	VARCHAR(100)			Nombre del proyecto
	proyecto_descripcion	VARCHAR(100)			Descripción del proyecto
	proyecto_fecha_creacion	VARCHAR(100)			Fecha de creación del proyecto

	proyecto_ultima_act	DATETIME			Fecha de la última actualización del proyecto
	tipo_proyecto_id	INT(10) <b>FK</b>	tipo_proyecto	tipo_proyecto_id	Identifica el modelo de aplicación del proyecto
	proyecto_estado	TINYINT(1)			Estado del proyecto(1 activo, 0 desactivado/eliminado)
proyecto_compartido		Entidad			Tabla intermedia usada para romper la relación de n:n entre usuarios y proyectos
	proyecto_compartido_id	INT(10) <b>PK</b>			Código de tabla intermedia proyecto_compartido
	usuario_id	VARCHAR(13) <b>FK</b>	usuario	usuario_id	Código del usuario que contiene el proyecto
	proyecto_id	VARCHAR(13) <b>FK</b>	proyecto	proyecto_id	Código del proyecto que está siendo compartido
	proyecto_compartido_admin	TINYINT(1)			Especifica si el usuario es quien creó el proyecto y lo administra
	proyecto_compartido_estado	TINYINT(1)			Estado del proyecto compartido(1 activo, 0 desactivado/eliminado)

mapa		Entidad			Entidad que almacena información acerca de los distintos mapas que se usan en el sistema
	mapa_id	INT(10) <b>PK</b>			Código del mapa
	mapa_latitud	DOUBLE			Latitud del mapa
	mapa_longitud	DOUBLE			Longitud del mapa
	proyecto_id	INT(10)	proyecto	proyecto_id	Campo usado para determinar a qué proyecto pertenece el mapa
	mapa_estado	INT(10)			Estado del mapa(1 activo, 0 desactivado/eliminado)
marcador		Entidad			Entidad que almacena información de los marcadores ingresado en los mapas de proyectos
	marcador_id	INT(10) <b>PK</b>			Código del marcador
	mapa_id	INT(10)	mapa	mapa_id	Especifica a que mapa pertenece el marcador
	marcador_nombre	VARCHAR(50)			Nombre del marcador
	marcador_descripcion	VARCHAR(600)			Descripción del marcador

	marcador_latitud	DOUBLE			Latitud del marcador
	marcador_longitud	DOUBLE			Longitud del marcador
	marcador_estado	INT(10)			(1 activo,0eliminado)
sistema		Entidad			Entidad que almacena información de los distintos objetos :sistemas, subsistemas, elementos (sistemas de n nivel)
	id	INT(10) <b>PK</b>			Código del Sistema
	nombre	VARCHAR(300)			Nombre del Sistema
	x1	INT(10)			Posición en pantalla del extremo izquierdo de la elipse que representa al Sistema
	y1	INT(10)			Posición en pantalla del extremo superior de la elipse que representa al Sistema

	x2	INT(10)			Posición en pantalla del extremo derecho de la elipse que representa al Sistema
	y2	INT(10)			Posición en pantalla del extremo inferior de la elipse que representa al Sistema
	sysmain	INT(10) <b>FK</b>	sistema	id	Permite la recursividad, indica cual es el sistema padre del sistema
	fecha	DATETIME			Fecha de última modificación del Sistema
	det1	LONGTEXT			Descripción teórica de los propósitos de los momentos de intervención / Significados de origen de la Base de Conocimiento
	det2	LONGTEXT			Descripción teórica de las Teorías de los momentos de intervención / Significados de Local de la Base de Conocimiento
	det3	LONGTEXT			Descripción de Métodos de momentos de intervención /

					Propósitos de la Base de Conocimiento
	det4	LONGTEXT			Descripción teórica de las Herramientas de los momentos de intervención / Teorías de la Base de Conocimiento
	det5	LONGTEXT			Descripción teórica de las Actividades de los momentos de intervención / Métodos de origen de la Base de Conocimiento
	det6	LONGTEXT			Descripción teórica de los Resultados de los momentos de intervención /Herramientas de la Base de Conocimiento
	det7	LONGTEXT			Descripción teórica de los Indicadores de los momentos de intervención / Significados de origen de la Base de Conocimiento
	det8	LONGTEXT			Descripción teórica del Aprendizaje de los momentos de intervención

	det9	LONGTEXT			Descripción teórica de los propósitos de los momentos de intervención / Significados de origen de la Base de Conocimiento
	det10	LONGTEXT			Descripción teórica las Interacciones Clave de la Base de Conocimiento
	det11	LONGTEXT			Descripción teórica de la Dualidad de la Base de Conocimiento
	det12	LONGTEXT			Descripción teórica de la Situación del proyecto en la modelización del sistema
	det13	LONGTEXT			Descripción teórica de la Introducción de los Principios Esenciales/ Sistémicos
	det14	LONGTEXT			Descripción teórica de las Variables de los Principios Esenciales/ Sistémicos
	det15	LONGTEXT			Descripción teórica de los Indicadores de los Principios Esenciales/ Sistémicos

	fuelle	VARCHAR(300)			Fuente del texto que se muestra en la elipse que representa al sistema
	estilo	VARCHAR(300)			Estilo del texto que se muestra en la elipse que representa al sistema
	tamano	INT(10)			Tamaño del texto que se muestra en la elipse que representa al sistema
	hx	INT(10)			Altura de la elipse que representa al sistema
	cfondo	VARCHAR(300)			Color de fondo de la elipse que representa al sistema
	ctexto	VARCHAR(300)			Color del texto que se muestra en la elipse que representa al sistema
	tipo	INT(10)			Tipo de sistema (Sistema, Prototipo, Proceso de Solución, Variable de Realimentación)
	proyecto_id	INT(10) <b>FK</b>	proyecto	proyecto_id	Asocia el sistema a un proyecto



relación		Entidad			Entidad que almacena información acerca de las interacciones que se producen entre los distintos sistemas
	r1	INT(10) <b>PK FK</b>	sistema	id	Conforman la clave primaria de la entidad, además son llaves externas que permiten conocer que sistemas intervienen en la interacción
	r2	INT(10) <b>PK FK</b>	sistema	id	
	debería	INT(10)			Especifica cómo debería estar la interacción entre los sistemas
	esta	INT(10)			Indica cómo está estar la interacción entre los sistemas
	rfecha	DATETIME			Fecha de la última actualización
	fuerte				Establece si la interacción entre sistemas es fuerte (0) o no (1)

detalle					Entidad que almacena información que explica el por qué las interacciones entre sistemas tienen asignados sus valores
	id	INT(10) <b>PK</b>			Código del detalle
	r1	INT(10) <b>FK</b>	sistema	id	Especifican los sistemas que intervienen en el detalle
	r2	INT(10) <b>FK</b>	sistema	id	
	tipo	INT(10) <b>FK</b>	tipo	id	Indica si se trata de una causa o una razón
	descripción	TEXT			Descripción textual que explica la razón para los valores está y debería que se establecieron en la tabla interacción
	prioridad	INT(10)			Prioridad del detalle
	dfecha	DATETIME			Fecha de la última modificación del detalle

tipo		Entidad			Entidad que almacena información de detalle
	id	INT(10) <b>PK</b>			Código del tipo
	nombre	VARCHAR(50)			Causa / Razón
administracion					Entidad que almacena información acerca de los mensajes del administrador
	administracion_id	INT(10) <b>PK</b>			Código de administración
	administracion_mensaje	VARCHAR(500)			Mensaje con publicidad, información o avisos por parte del administración a los usuarios
	administracion_estado	TINYINT(1)			Estado del tipo del mensaje de administración(1 se muestra, 0 desactivado/no se muestra)

### 3.3.3 Arquitectura

El sistema Ge@ debido al modo en que opera, debe contar con flexibilidad para trabajar en dos entornos: web e intranet. Esto debido a que muchas de las veces los administradores lo aplican en talleres o charlas para obtener datos que alimenten posteriormente la base de conocimiento y conseguir así capitalizarlo. En tales escenarios generalmente no se cuenta con infraestructura que permita la conexión a internet, por lo que debe ser construido pensando en ocuparlo en pequeñas intranets, en la cual sea sólo necesario instalar la “Herramienta web para la planificación, manejo y ordenamiento territorial” en un ordenador y conectar varios equipos en la misma red interna y así acceder al sistema.

A continuación se presentan dos diagramas en donde se especifican los elementos que conforman los dos tipos de entornos:

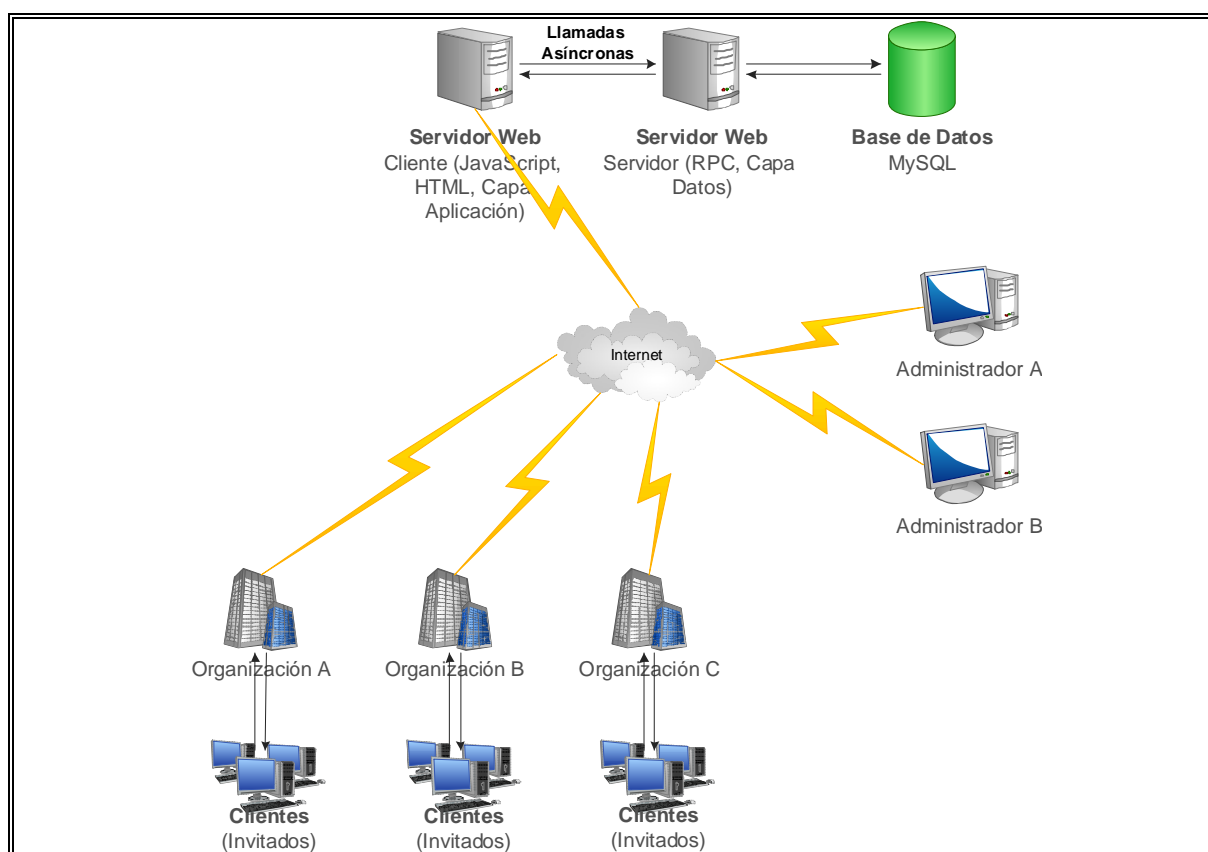


Figura 3-71. Arquitectura 1 (Internet)

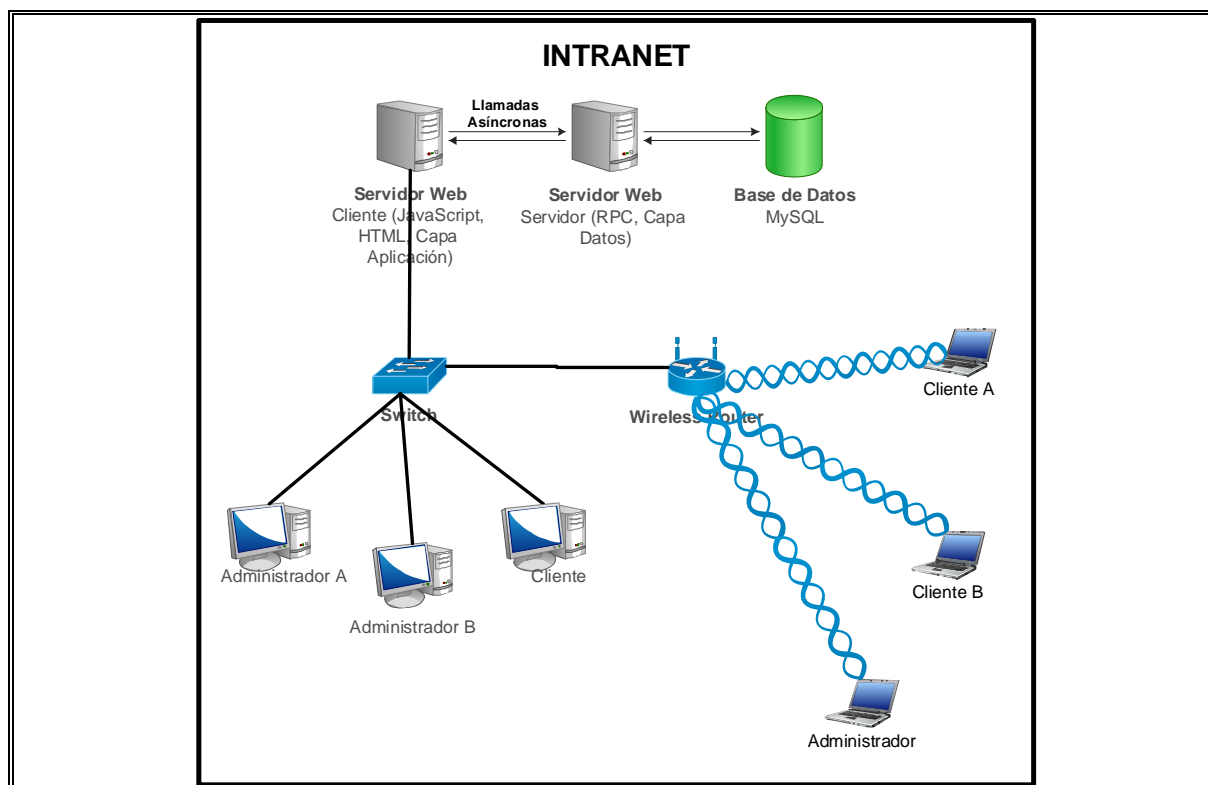


Figura 3-72. Arquitectura 2 (Intranet)

En cualquiera de los dos casos Ge@ se mantiene siempre enfocado a brindar una herramienta tanto para los clientes o administradores del sistema. Como se puede observar en los gráficos, se trata de una arquitectura cliente/servidor centralizada.

La herramienta web al ser una arquitectura cliente/servidor, indica que debe existir una comunicación entre estos dos elementos, la forma en cómo estos elementos se comunican es explicado a continuación y se podrá observar cómo trabaja la arquitectura que se ha propuesto para este sistema web.

El sistema está conformado por tres capas que son:

- **Capa de Aplicación:** Esta capa es la que se encuentra en contacto directo con el cliente ya sea este administrador o usuario común, es aquí donde se genera toda la lógica necesaria para poder crear interfaces en donde los clientes puedan interactuar con el sistema. Esta capa contendrá todo el código HTML y Javascript usado para la construcción de la capa. Toda la interfaz se presentará al lado del cliente en sus navegadores.

- **Capa de Negocio:** Esta capa intermedia contiene la lógica del sistema es decir, se encargará de tomar los datos, trabajar con ellos, pulirlos y ser enviados a la capa de interfaz o aplicación para ser presentados al usuario final. Esta capa se encuentra presente al lado del cliente, es decir la capa de aplicación solicita y presenta datos de acuerdo a esta capa.
- **Capa de datos:** Esta capa se encuentra en el lado del servidor y es la encargada de tomar las peticiones enviadas del lado del cliente, la capa de datos contiene toda la lógica de comunicación con la base de datos, cualquier comunicación con el servidor pasa por esta capa.

La comunicación que existe entre estas capas determinará la manera en que opera la arquitectura del sistema. Basándonos en un proceso simple de registro de datos en la base, vemos que se presentan los siguientes pasos:

1. El usuario ingresa la información que desea guardar en la base de datos sobre la interfaz que puede visualizar en el navegador de su máquina.
2. Una vez que haya ingresado los datos necesarios, el usuario manda a ejecutar el proceso de guardar los datos al generar una acción sobre un componente de la interfaz como por ejemplo un botón, esto genera que toda la información se encapsule en un objeto, el proceso de encapsulación indica que la información está pasando por la capa de negocio, lo que se hace exactamente es colocar todos los datos en un objeto la cual está conformado por campos que almacenarán los datos recolectados de la capa de aplicación. Todo esto se produce en el lado del cliente.
3. Para enviar este objeto a la capa de datos en el lado del servidor, se hace uso de lo que se conoce como llamada a procedimientos remotos asíncronos. La capa de negocio genera un objeto asíncrono que es el encargado de comunicarse con el servidor y el cual empaquetará la información de manera segura para ser enviado por la red. Los RPC Asíncronos liberan de carga el lado del cliente ya que el usuario tranquilamente puede seguir operando sobre la interfaz (capa de aplicación) mientras que de manera interna (capa de negocio) se encapsula y empaqueta la información para ser enviada al servidor.

4. El proceso de envío sobre la red dependerá de la tecnología que se use ya sea en la Intranet o Internet, por lo general mediante paquetes TCP/IP sobre una infraestructura física determinada por el proveedor de servicios.
5. Una vez que la información se haya transmitido por la red, al llegar al lado del servidor se tiene una interfaz que toma el objeto asíncrono enviado, y lo desempaqueta en el objeto original, para poder reconstruir el objeto es necesario que los componentes que generan el objeto y que son elaborados en el lado del cliente se encuentren presentes en el lado del servidor, es decir la capa de negocio se encuentra en ambos lados. De esta manera se puede extraer la información tomada de la capa de aplicación y ser enviada a la base de datos.
6. El proceso de guardar o tomar datos de la base está dado por el servidor web o aplicaciones que se tenga dentro de lado del servidor.
7. Si la llamada al procedimiento remoto contiene un valor de retorno se sigue el mismo proceso pero en sentido opuesto.

En general, la comunicación entre el cliente y el servidor está basado en llamadas a procedimientos remotos asíncronos, cualquier acción entre estos dos lados pasan por una llamada a un procedimiento remoto.

### 3.4 Implementación

El proceso de implementación consiste en definir cómo va a ser construido la herramienta web desde sus interfaces hasta los componentes que lo conformarán, para esta etapa y siguiendo el modelo RUP se va a hacer uso de los siguientes gráficos de UML: Diagrama de Paquetes y Componentes, que nos van a servir de guía para darnos una idea de cómo se implementará el sistema.

La codificación de la arquitectura se puede observar en las siguientes figuras, en donde se puede observar mediante paquetes y clases Java los dos lados tanto del cliente como servidor y las tres capas de la herramienta web.

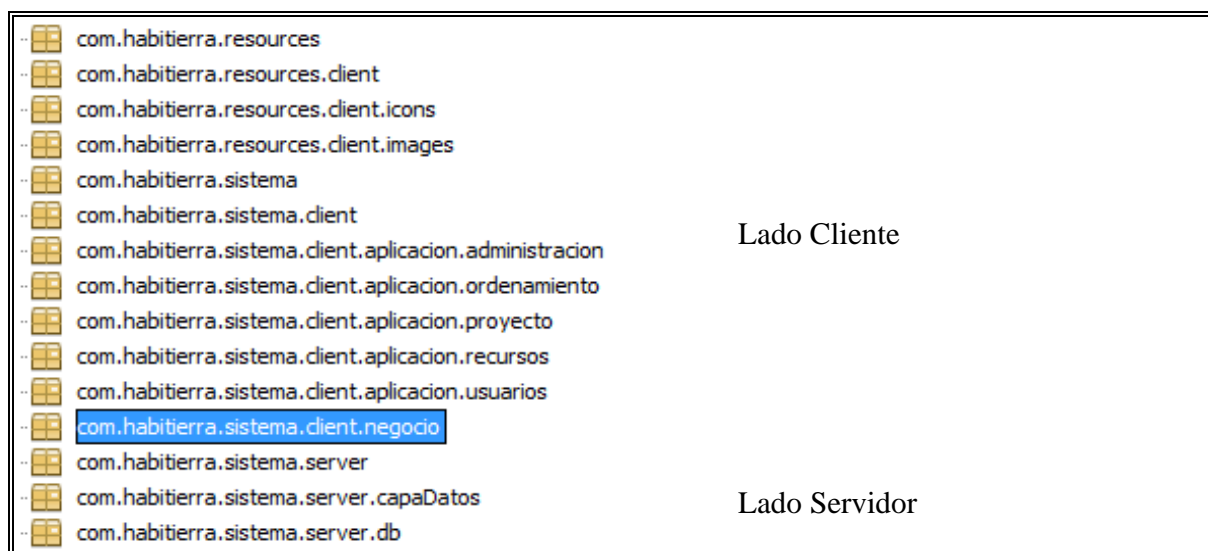


Figura 3-73: Arquitectura cliente/servidor en paquetes Java.

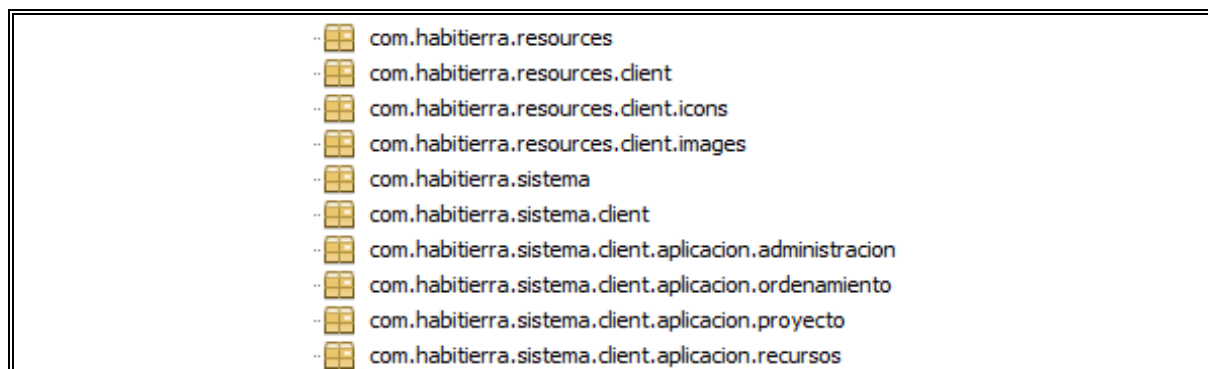


Figura 3-74: Capa de aplicación en paquetes Java.



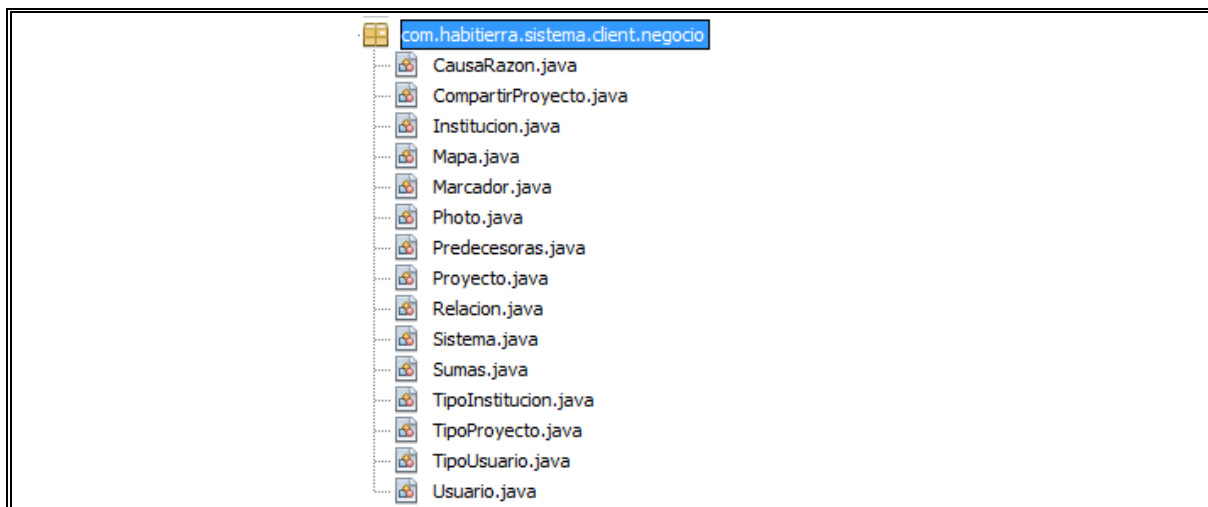


Figura 3-75: Capa de Negocio en clases Java

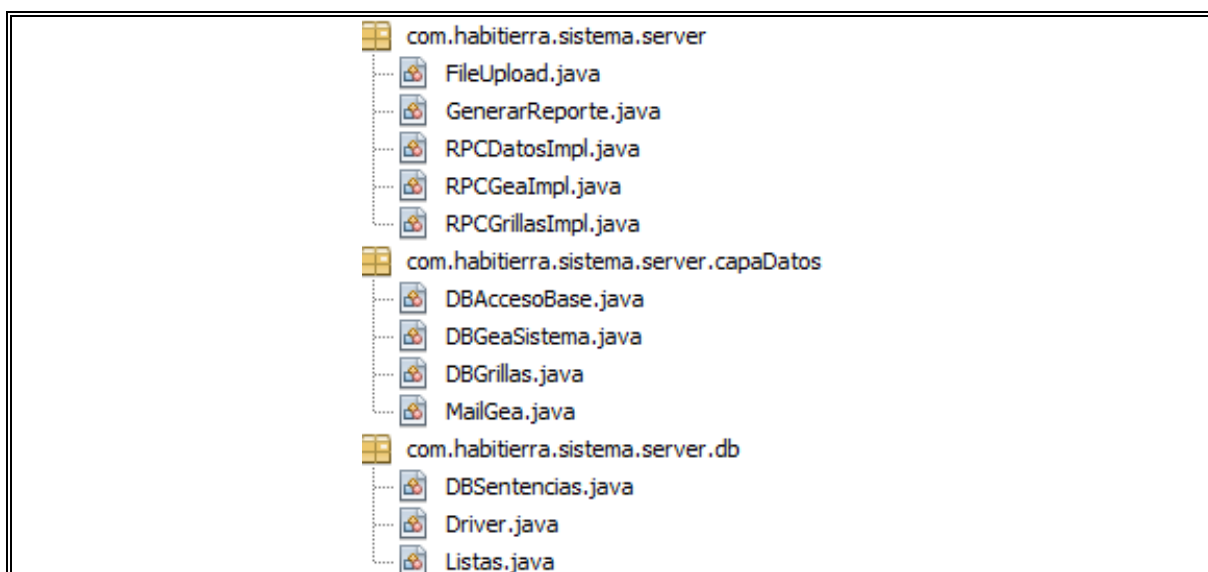


Figura 3-76: Capa de datos en paquetes y clases Java.

### 3.4.1 Prototipo de interfaces

Luego del análisis y diseño efectuado en la fase de análisis y diseño, se procede a generar un prototipo de interfaces del sistema, que de acuerdo al modelo RUP en posteriores iteraciones se las va a ir refinando de acuerdo a la interacción que se tenga con los usuarios que manejarán el sitio web.

#### 3.4.4.1 Prototipo de Login

En primera instancia se tiene la interfaz de login que se usa cuando el usuario desea ingresar al sistema, dispone de dos cuadro de texto uno para ingresar el nombre de usuario y otro para la contraseña y un botón de entrar.

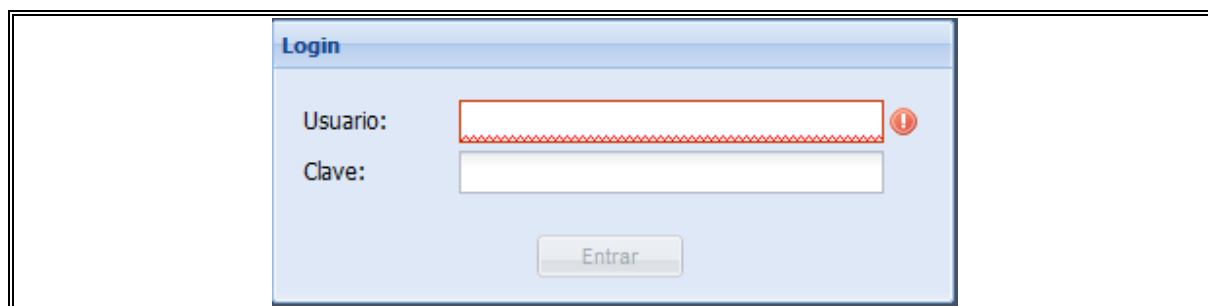


Figura 3-77: Prototipo de interfaz de login. Muestra un formulario de login con los campos 'Usuario:' y 'Clave:', un botón 'Entrar', y un ícono de advertencia en el campo de usuario.

Figura 3-77: Prototipo de interfaz de login

#### 3.4.4.2 Prototipo de Registro de Usuarios

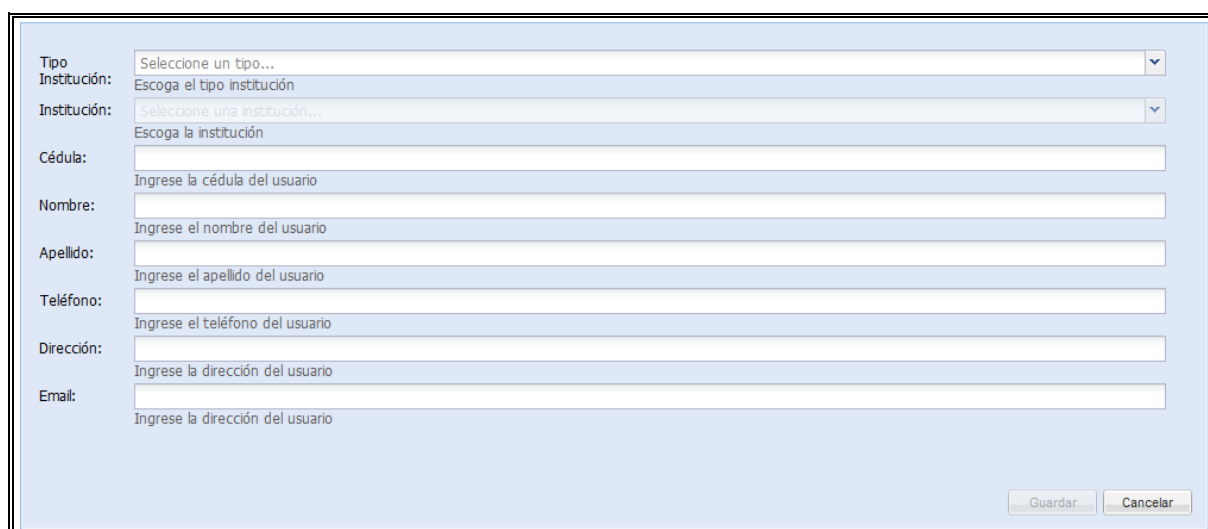


Figura 3-78: Prototipo de interfaz de registro de usuario. Muestra un formulario de registro con campos para Tipo, Institución, Cédula, Nombre, Apellido, Teléfono, Dirección, y Email, y botones 'Guardar' y 'Cancelar'.

Figura 3-78: Prototipo de interfaz de registro de usuario

La interfaz de registro de usuario es sencilla cuenta con dos combo box que permiten asociar un usuario dentro de una determinada institución al seleccionar su tipo y los demás datos necesarios del usuario. Dispone de dos botones, el primero usado para guardar los datos y un segundo botón para cancelar el registro.

### 3.4.4.3 Prototipo de interfaz de administración

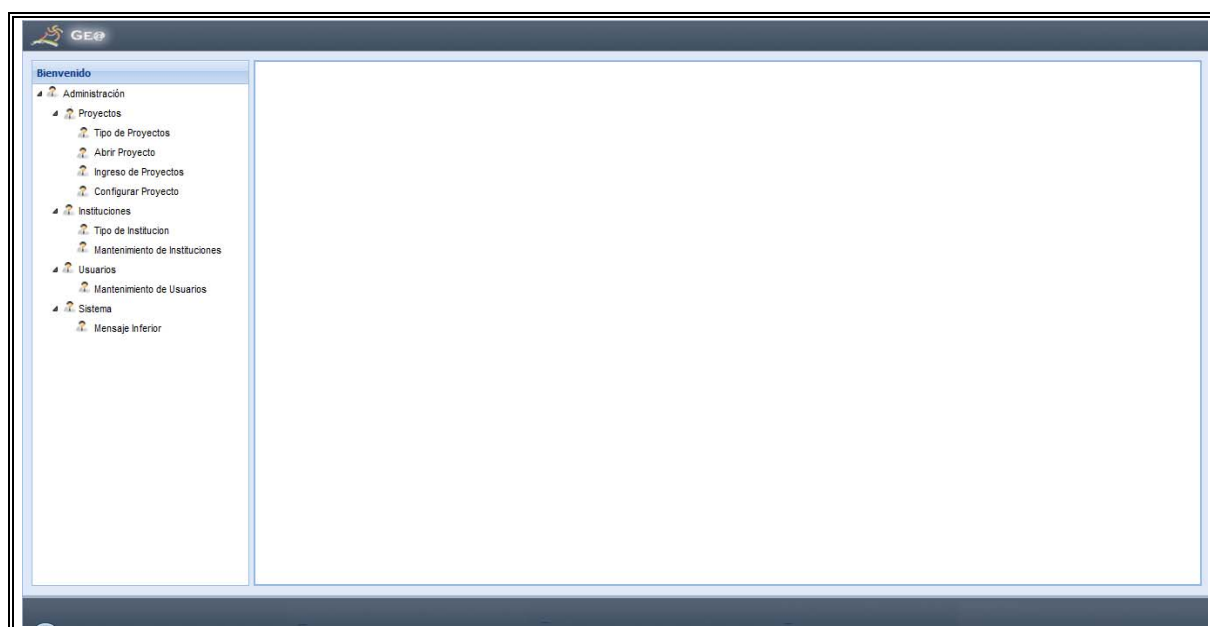


Figura 3-79: Prototipo de interfaz de administración

Esta es la interfaz a la cual accede el administrador luego de hacer login. Es una página simple que muestra un árbol en donde se presentan los módulos del sistema y un área en donde se cargarán las interfaces de ingreso, modificación y eliminación de acuerdo al módulo que se haya seleccionado.

### 3.4.4.4 Prototipo de interfaz de invitado

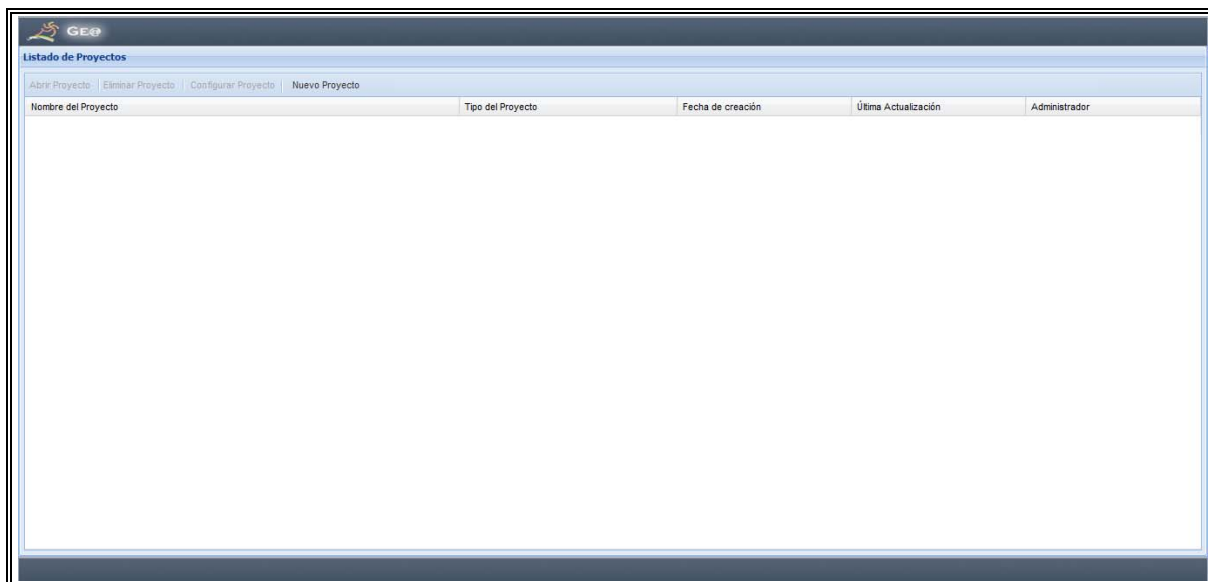


Figura 3-80: Prototipo de interfaz de invitado

El prototipo de interfaz de invitado es diferente al de administrador, en esta pantalla se tiene un listado de los proyectos asociados al usuario y las operaciones básicas que son abrir, eliminar, configurar y crear nuevos proyectos, presentados en la parte superior de la lista como un conjunto de botones.

### 3.4.4.5 Prototipo de interfaz de registro de proyectos

Este prototipo de interfaz de usuario muestra un formulario titulado "Datos del Proyecto". El formulario contiene tres campos de entrada: "Nombre de Proyecto" con el texto de ayuda "Ingrese el nombre del proyecto", "Tipo de Proyecto" con un menú desplegable y el texto de ayuda "Escoga el Tipo de Proyecto", y "Descripción" con un área de texto grande y el texto de ayuda "Ingrese una descripción del proyecto". En la parte inferior derecha del formulario, hay dos botones: "Guardar" y "Cancelar".

Figura 3-81: Prototipo de ingreso de proyectos

El prototipo de interfaz de ingreso de proyectos fue tomado como muestra para el ingreso de instituciones, tipo de proyectos, tipos de institución y administración del sistema, cada uno configurado de acuerdo a los campos que necesitan dentro de cada módulo.

Las mismas interfaces son tomadas para presentar los datos cuando vayan a ser modificados o eliminados.

### 3.4.4.6 Prototipo de interfaz de sistema Ge@

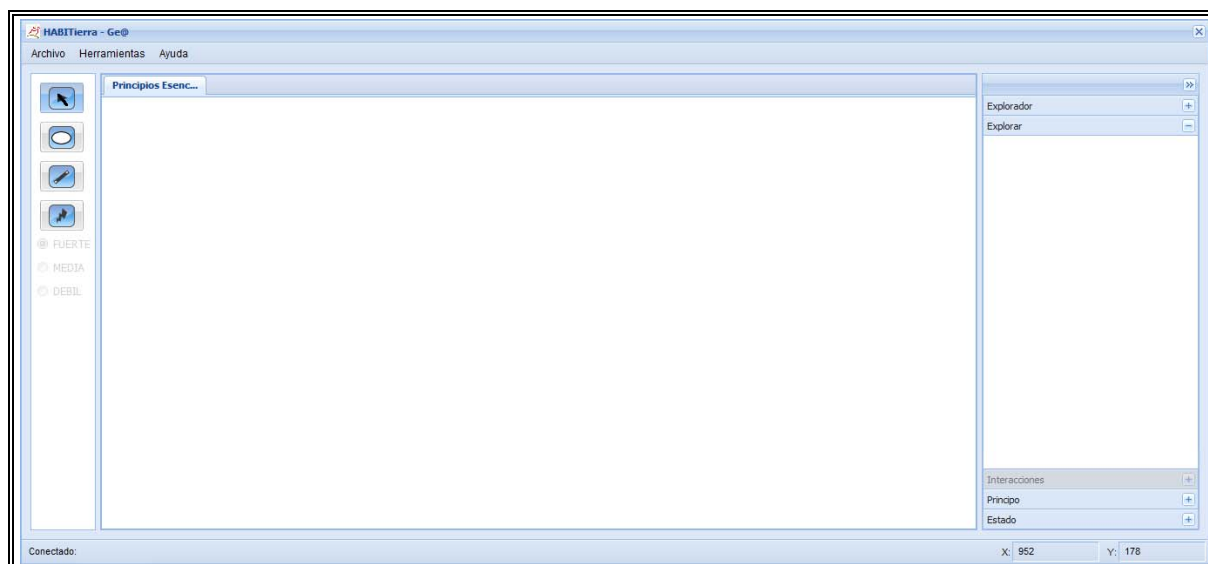


Figura 3-82: Prototipo de interfaz de sistema ge@

Al pulsar abrir proyecto ya sea desde la interfaz de administrador o invitado este llevará a la interfaz de lo que es el modelamiento digital de Ge@, es la parte central de la herramienta web, es aquí donde el usuario va a crear sus sistemas con sus respectivas asociaciones que se tengan en el proyecto con el fin de tener una asesoría por parte del Gerente de la empresa; mediante esta interfaz se podrá gestionar, planificar y gobernar territorios.

Las interfaces que se visualizan en cada una de las secciones anteriores son prototipos y es probable que la mayoría de estas cambien con el transcurso de las siguientes iteraciones del modelo RUP, pero se toman como base para que a partir de estas interfaces, los usuarios tengan idea de cómo estará implementada la herramienta web para la planificación, gestión y gobernabilidad de territorios.

### 3.4.2 Diagrama de Paquetes

El siguiente diagrama es utilizado para mostrar cómo está dividido el sistema en agrupaciones lógicas mostrando las dependencias entre estas agrupaciones. Dado que normalmente un paquete está pensando como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de la herramienta web.

Los paquetes son buenos elementos de gestión es por esta razón que son muy utilizados dentro de esta fase del modelo RUP, cada paquete puede asignarse a un individuo o equipo. A continuación se presenta el diagrama de paquetes de la herramienta web.

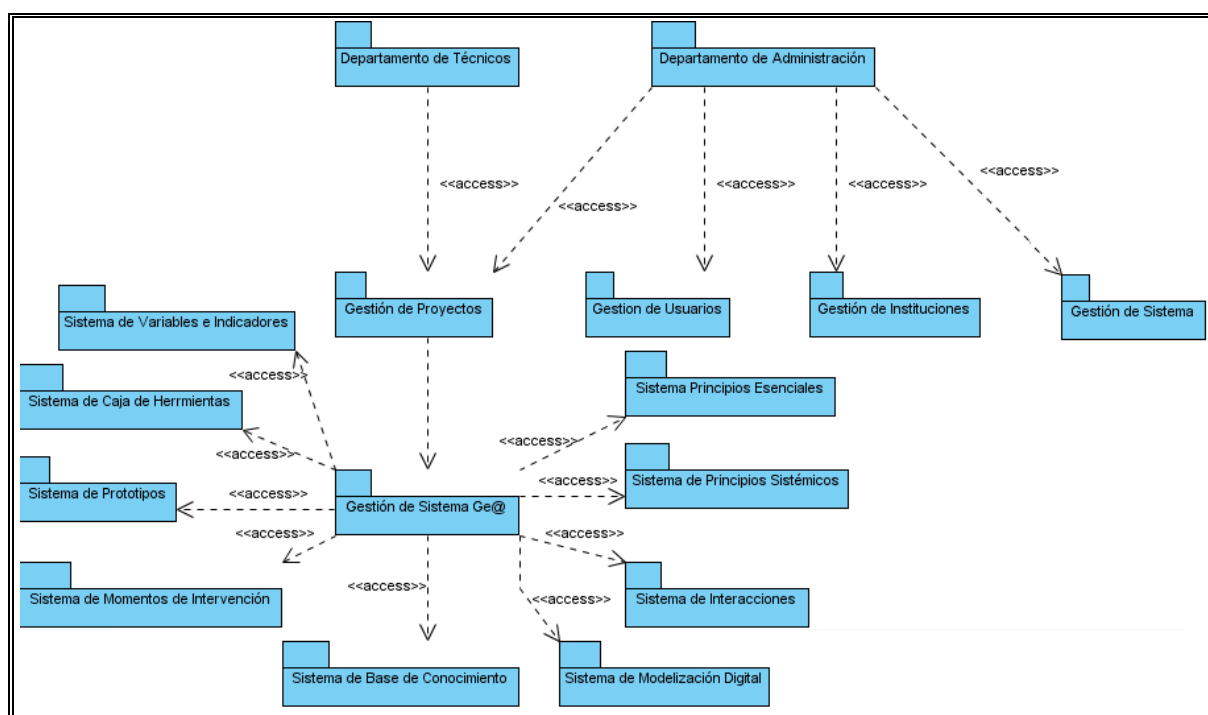


Figura 3-83: Diagrama de Componentes de la “Herramienta Web para la planificación, gestión y gobierno de territorios”

Al observar la figura superior se puede observar el orden jerárquico que se sigue al usar la herramienta web y que módulos y sistemas internos se maneja. En manera resumida se muestra como está estructurado el sistema web.

### 3.4.3 Diagrama de Componentes

El diagrama de componentes representa cómo la herramienta web es dividida en componentes y muestra las dependencias entre estos componentes. Representa los componentes físicos como los módulos, archivos o paquetes. Estos diagramas no incluyen toda la información, ya que por lo general los componentes se realizan por partes y cada diagrama describe un apartado del sistema.

El diagrama de paquetes descrito en la sección anterior es la base para generar el diagrama de componentes, los módulos que se van a describir con estos diagramas son los siguientes

- Gestión de proyectos.
- Gestión de instituciones.
- Gestión de usuarios.
- Gestión de sistema.
- Gestión de Sistema Ge@.

#### 3.4.3.1 Diagrama de Componentes de Gestión de Proyectos

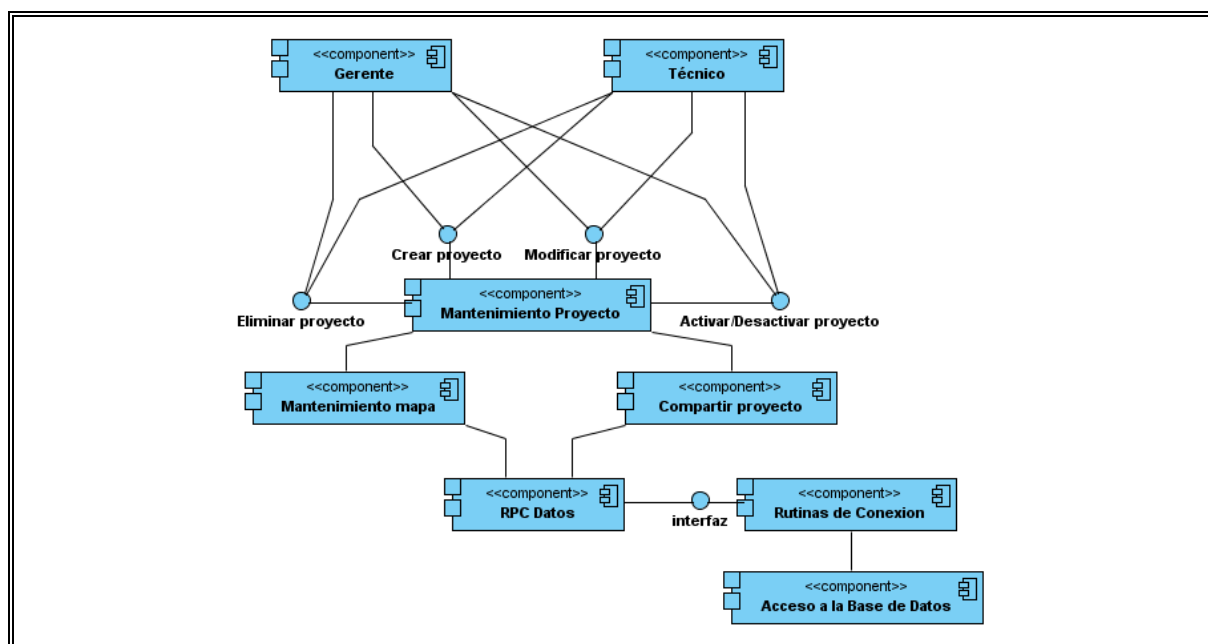


Figura 3-84: Diagrama de componentes del paquete Gestión de Proyectos

### 3.4.3.2 Diagrama de Componentes de Gestión de Instituciones

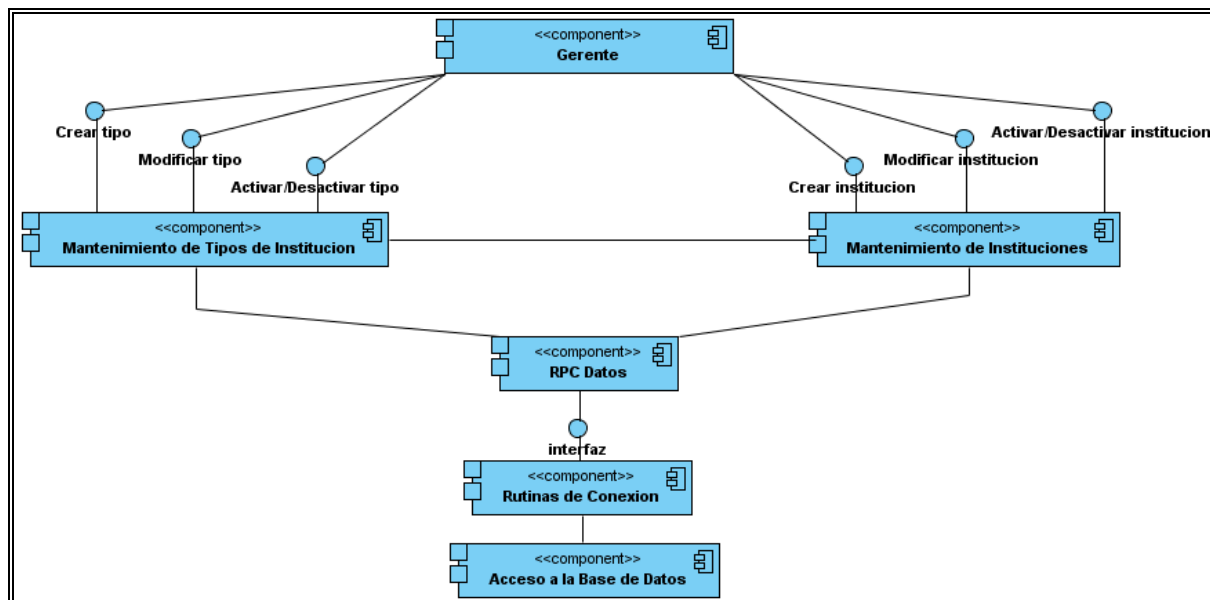


Figura 3-85: Diagrama de componentes del paquete Gestión de Instituciones

### 3.4.3.3 Diagrama de Componentes de Gestión de Usuarios

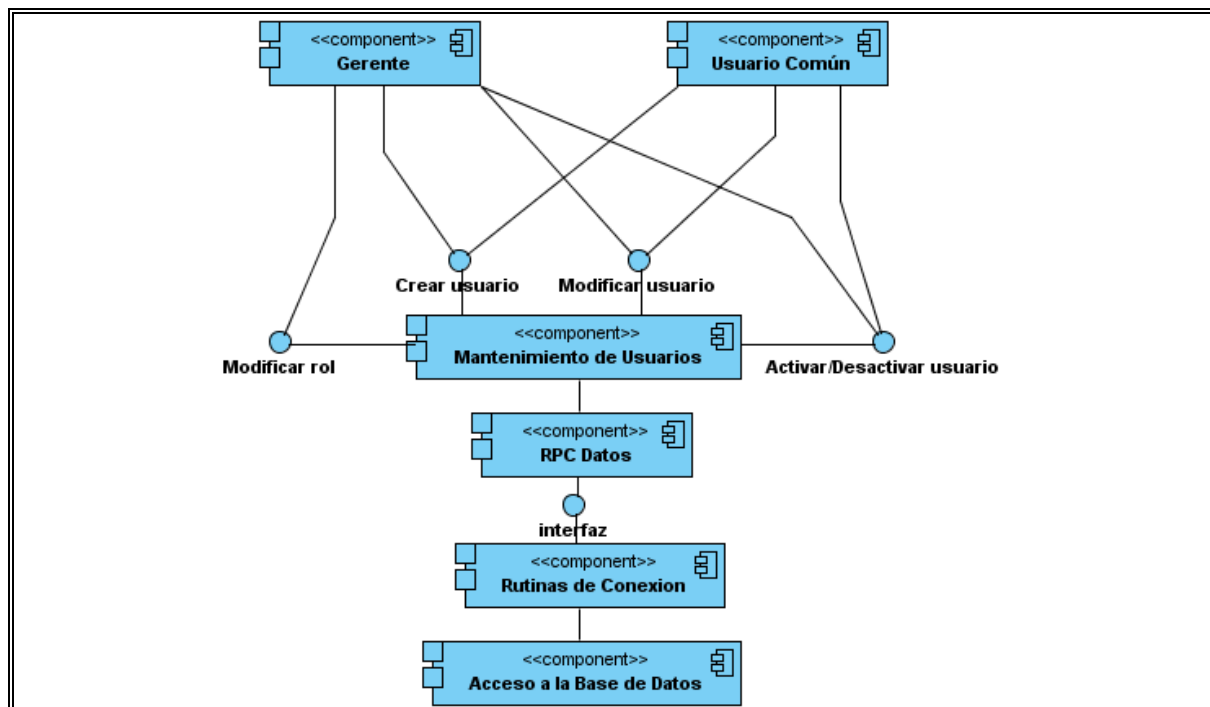


Figura 3-86: Diagrama de componentes del paquete Gestión de Usuarios



### 3.4.3.4 Diagrama de Componentes de Gestión de Sistema

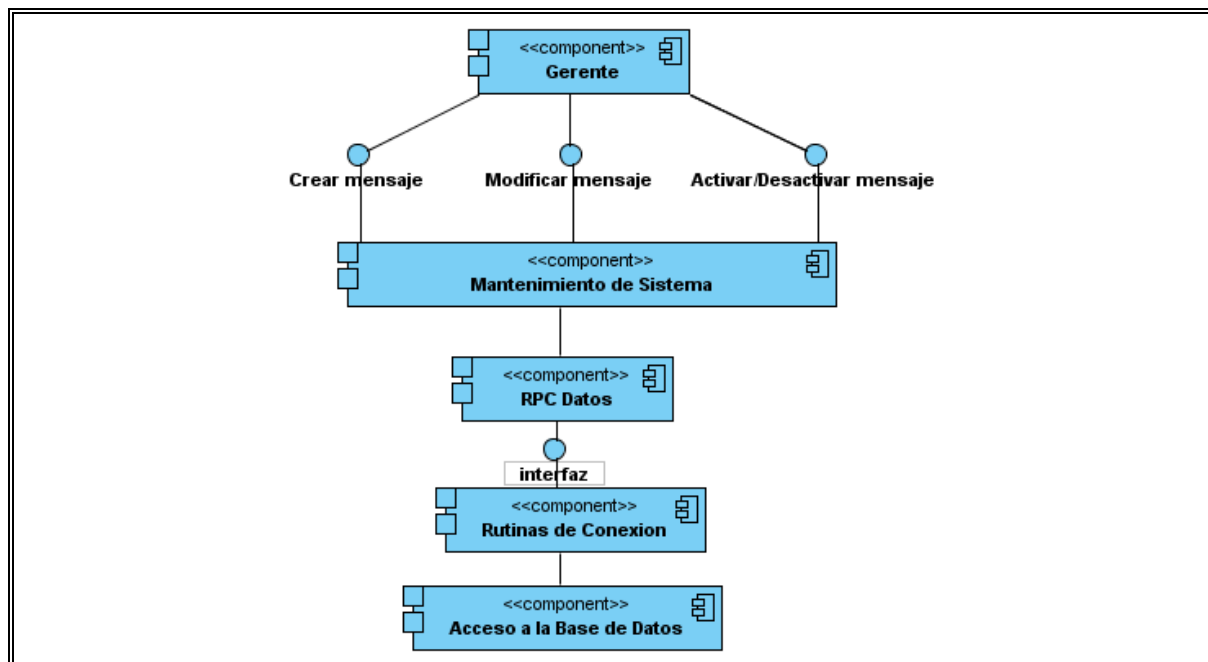


Figura 3-87: Diagrama de componentes del paquete Gestión de Sistema

### 3.4.3.5 Diagrama de Componentes de Gestión de Sistema Ge@

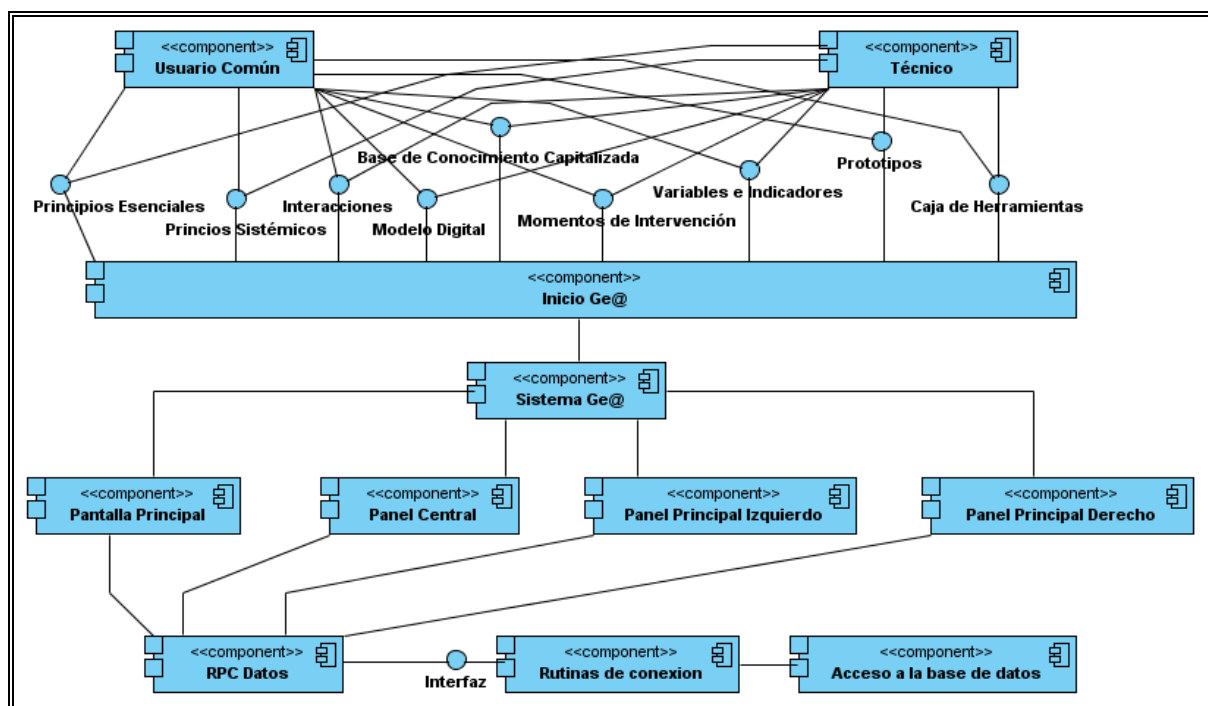


Figura 3-88: Diagrama de componentes del paquete Gestión de Sistema Ge@

### **3.5 Pruebas**

Este artefacto cubre el conjunto de pruebas realizadas los casos de uso más relevantes y frecuentes en la Herramienta Web

Este caso de uso son:

- Gestión de Usuarios .- Desactivar Perfil de Usuario (CU9)
- Gestión de Proyectos.- Ingresar nuevo proyecto (CU16)
- Gestión de Proyectos.- Compartir Proyectos (CU18)
- Gestión de Instituciones.- Modificar Institución (CU31)
- Gestión de Administración.- Modificar Mensaje Publicitario (CU36)
- Modelamiento Digital del Proyecto.- Manipular Objetos(CU48)
- Modelamiento Digital del Proyecto.- Indicar Causas y Razones(CU48)
- Modelamiento Digital del Proyecto.- Mantenimiento de Realimentación(CU50)
- Modelamiento Digital del Proyecto.- Generar Sumas (CU52)
- Modelamiento Digital del Proyecto.- Generar Reporte (CU51)

#### **3.5.1 CU9.- Desactivar Perfil de Usuario**

##### **3.5.1.1 Descripción**

Ingresamos a la herramienta Web como usuario Gerente, accediendo a al módulo de usuarios, el sistema nos mostrará un listado de los mismos y al marcar uno de ellos, se mostrará una interfaz para que llevemos a cabo la modificación de dicho Usuario. Una vez hechas las modificaciones guardar y ver cómo responde el sistema.

##### **3.5.1.2 Condiciones de Ejecución**

Las condiciones de ejecución del caso de prueba son que el usuario gerente, dará de baja al Técnico César Franco, cuyo perfil actualmente se encuentra en estado “Activo”

### 3.5.1.3 Entrada

Los datos ingresados se muestran en la figura:

The screenshot shows a web application interface for user management. On the left, there is a table titled "Mantenimiento de Instituciones" with columns for Cédula, Nombre, Apellido, Institucion, and Estado. The table lists 18 users, with the entry for Cesar Franco highlighted. On the right, there is a form titled "Información de Usuario" with fields for Tipo, Institución, Cédula, Nombre, Apellido, Dirección, Teléfono, E-mail, and Rol de Usuario. Below this is another section titled "Información de cuenta de usuario" with fields for Username, Password, and Confirmar password. At the bottom right, there are three buttons: "Nuevo", "Desactivar", and "Guardar".

Cédula	Nombre	Apellido	Institucion	Estado
8000000008	Nasario	Buele Buele	Parroquias Rurales El Oro	Activo
9000000009	José Luis	Ramirez	Parroquias Rurales El Oro	Activo
0101301828	Pedro	Zeas	Geoinformatica Consultores	Activo
0105219109	Fernanda	Pérez	Geoinformatica Consultores	Activo
0123456789	Angel	Espinoza	Geoinformatica Consultores	Activo
0702629718	Angel	Espinoza	Geoinformatica Consultores	Activo
0100000000	Cristhian	Vega	Geoinformatica Consultores	Activo
0103269718	Tatiana	Montalvan	Geoinformatica Consultores	Activo
0105012751	Andres	Molina	Geoinformatica Consultores	Activo
0200000000	Jeaneth	Ramirez	Geoinformatica Consultores	Activo
0300000000	Cesar	Franco	Geoinformatica Consultores	Activo
0400000000	Isabel	Orquera	Geoinformatica Consultores	Activo
0500000000	Lourdes	Monteros	Geoinformatica Consultores	Activo
0600000000	Johanna	Fernandez	Geoinformatica Consultores	Activo
1104244578	IVAN	ESPINOSA	Geoinformatica Consultores	Activo

**Información de Usuario**

Tipo: PRIVADA  
 Institución: GeoInformatica Consultores  
 Cédula: 0300000000  
 Nombre: Cesar  
 Apellido: Franco  
 Dirección: Machala  
 Teléfono: 12341234  
 E-mail: cesar.franco@geoinformatica.org  
 Rol de Usuario: Invitado

**Información de cuenta de usuario**

Username: cesar.franco  
 Password:   
 Confirmar password:   
 Nuevo Desactivar Guardar

Figura 3-89: Datos de prueba CU2

### 3.5.1.4 Resultado Esperado

La cuenta del Técnico César Franco debe desactivarse

### 3.5.1.5 Evaluación de la prueba

Mantenimiento de Instituciones				
Cédula	Nombre	Apellido	Institucion	Estado ▲
8000000008	Nasario	Buele Buele	Parroquias Rurales El Oro	Activo
9000000009	José Luis	Ramírez	Parroquias Rurales El Oro	Activo
0101301828	Pedro	Zeas	Geoinformatica Consultores	Activo
0105219109	Fernanda	Pérez	Geoinformatica Consultores	Activo
0123456789	Angel	Espinoza	Geoinformatica Consultores	Activo
0702629718	Angel	Espinoza	Geoinformatica Consultores	Activo
0100000000	Cristhian	Vega	Geoinformatica Consultores	Activo
0103269718	Tatiana	Montalvan	Geoinformatica Consultores	Activo
0105012751	Andres	Molina	Geoinformatica Consultores	Activo
0200000000	Jeaneth	Ramirez	Geoinformatica Consultores	Activo
0300000000	Cesar	Franco	Geoinformatica Consultores	Desactivado
0400000000	Isabel	Orquera	Geoinformatica Consultores	Activo
0500000000	Lourdes	Monteros	Geoinformatica Consultores	Activo
0600000000	Johanna	Fernandez	Geoinformatica Consultores	Activo
1104244578	IVAN	ESPINOSA	Geoinformatica Consultores	Activo

Figura 3-90: Resultado de prueba CU2

Prueba superada con éxito

## 3.5.2 CU16.- Ingresar Nuevo Proyecto

### 3.5.2.1 Descripción

Ingresamos a la herramienta Web con un una cuenta de usuario Técnico o Usuario Común, accedemos al módulo de proyectos, elegimos la opción Ingresar Nuevo Proyecto y se mostrará una interfaz para que llevemos a cabo el ingreso del proyecto. Una vez completados los datos generamos el proyecto y vemos como responde el sistema.

### 3.5.2.2 Condiciones de Ejecución

Las condiciones de ejecución del caso de prueba son que se ingresará un nuevo proyecto denominado “Pot El Oro”, se ingresará además un marcador en el mapa denominado “POT EL ORO”.

### 3.5.2.3 Entrada

Los datos ingresados se muestran en la figura:

Bienvenido a la versión final del sistema Ge@. Para mayor información o sugerencias contáctenos a [www.habitierra.org](http://www.habitierra.org) o al 072

Figura 3-91: Datos de prueba CU16

### 3.5.2.4 Resultado Esperado

Se debe generar un proyecto y debe mostrarse en la lista de proyectos

### 3.5.2.5 Evaluación de la prueba

Nombre del Proyecto	Tipo del Proyecto	Fecha de creación	Última Actualización	Usuario	Institución
Proyecto Marco Leon	Ordenamiento Territorial	2/23/11	3/3/11	Marco Leon	Municipio de Loja
Proyecto prueba mapa	Ordenamiento Territorial	3/2/11	3/2/11	Jorge Leon	Parroquias Rurales
Proyecto Prueba Mapas	Ordenamiento Territorial	3/2/11	3/3/11	Jorge Leon	Parroquias Rurales
Proyecto 4	Ordenamiento Territorial	12/15/10	3/3/11	Jorge Leon	Parroquias Rurales
Proyect 6	Ordenamiento Territorial	1/3/11	3/3/11	Jorge Leon	Parroquias Rurales
Proyecto 7	Ordenamiento Territorial	1/4/11	3/3/11	Jorge Leon	Parroquias Rurales
Proyecto de Prueba	Ordenamiento Territorial	2/23/11	3/3/11	Jorge Leon	Parroquias Rurales
Pot El Oro	Ordenamiento Territorial	12/15/10	7/7/11	Jorge Leon	Parroquias Rurales
Carcabon	Ordenamiento Territorial	5/5/11	5/5/11	Luis Robalino Mach	Parroquias Rurales

Figura 3-92: Resultado de prueba CU16

Prueba superada con éxito

### 3.5.3 CU18.- Compartir Proyectos

#### 3.5.3.1 Descripción

Ingresamos a la herramienta Web como usuario común o Técnico, accediendo al de Proyectos, elegimos un proyecto existente, elegimos la opción Compartir Proyecto y el sistema nos mostrará un listado de los usuarios y se elige aquellos con quienes se compartirá el proyecto

#### 3.5.3.2 Condiciones de Ejecución

Las condiciones de ejecución del caso de prueba son que el usuario “Jorge Espinosa” ha creado un proyecto denominado “Pot El Oro” y ha decidido compartirlo con un grupo de usuarios que se encuentran activos y que pueden formar de la misma institución o de otra distinta.

#### 3.5.3.3 Entrada

Los datos ingresados se muestran en la figura:

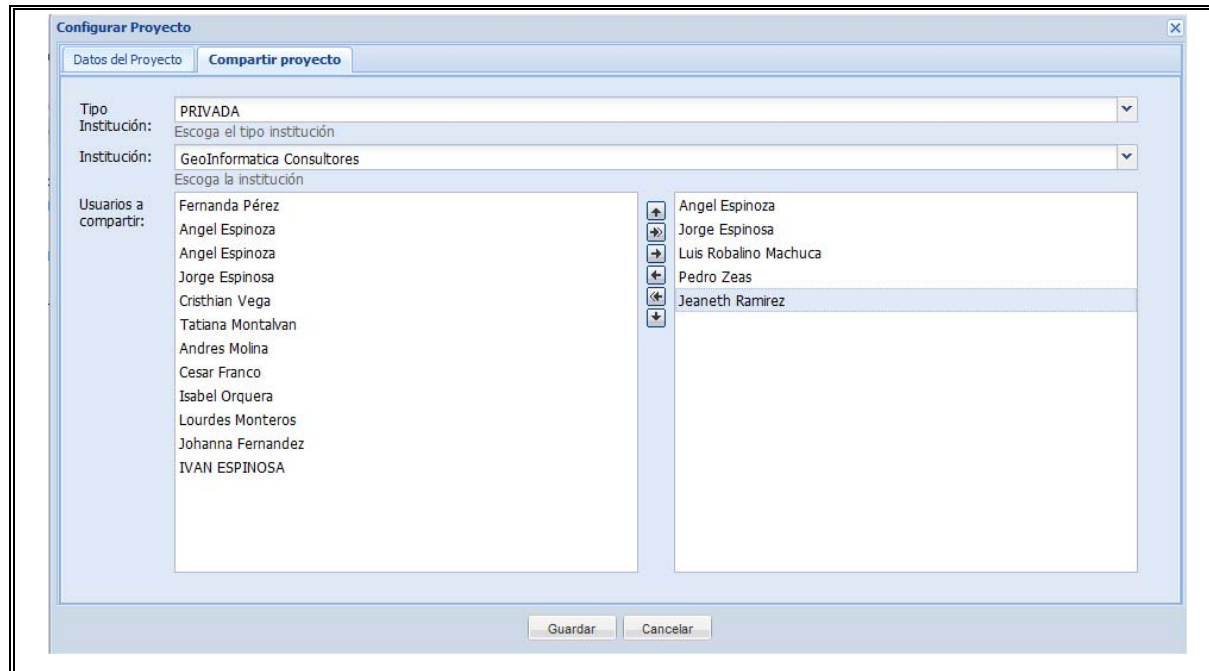


Figura 3-93: Datos de prueba CU18

### **3.5.3.4 Resultado Esperado**

El proyecto “Pot El Oro” compartido entre diversos usuarios.

### **3.5.3.5 Evaluación de la prueba**

Prueba superada con éxito

## **3.5.4 CU31.- Modificar Institución**

### **3.5.4.1 Descripción**

Ingresamos a la herramienta Web como usuario Gerente, accediendo al módulo de Instituciones, el sistema nos mostrará un listado con los nombre de las existentes y al marcar una de ellas se muestra una interfaz para que llevemos a cabo la modificación de dicho Usuario. Una vez dentro intentaremos guardar los nuevos datos y observamos cómo responde el sistema.

### **3.5.4.2 Condiciones de Ejecución**

Las condiciones de ejecución del caso de prueba son que el usuario Gerente se ha dado cuenta de un error en una de las instituciones creadas y necesita cambiar el tipo de organización, pues la Institución “Municipio de Loja” no es parte de la CONAJUPARE sino de la AME

### **3.5.4.3 Entrada**

Los datos ingresados se muestran en la figura:

The screenshot shows a web application interface for managing institutions. On the left, there is a table with the following data:

Institución	Ciudad	Dirección	Tipo de Institución	Estado
Municipio de Loja	Loja	Bolivar entre Jose Antonio y Colon	CONAJUPARE	Activo
Parroquias Rurales El Oro	Machala	Av. Machala	CONAJUPARE	Activo
Junta Cantonal Ibarra	Ibarra	Av. Ibarra	AME	Activo
Geoinformatica Consultores	Cuenca	Pajan y Rio Malacatus	PRIVADA	Activo

On the right, there is a form for editing the selected institution. The form fields are:

- Tipo Institución: AME (dropdown menu)
- Nombre: Municipio de Loja
- Ciudad: Loja
- Dirección: Bolivar entre Jose Antonio y Colon
- Teléfono: 2577292
- E-mail: loja@correo.com

At the bottom right of the form, there are three buttons: 'Nuevo', 'Desactivar', and 'Guardar'. The status bar at the bottom indicates 'Page 1 of 1' and 'Displaying 1 - 4 of 4'.

Figura 3-94: Datos de prueba CU31

### 3.5.4.4 Resultado Esperado

La Institución “Municipio de Loja” debe estar modificada, pertenecer a la “AME”

### 3.5.4.5 Evaluación de la prueba

The screenshot shows the updated table of institutions after the test. The data is as follows:

Institución	Ciudad	Dirección	Tipo de Institución	Estado
Municipio de Loja	Loja	Bolivar entre Jose Antonio y Colon	AME	Activo
Parroquias Rurales El Oro	Machala	Av. Machala	CONAJUPARE	Activo
Junta Cantonal Ibarra	Ibarra	Av. Ibarra	AME	Activo
Geoinformatica Consultores	Cuenca	Pajan y Rio Malacatus	PRIVADA	Activo

Figura 3-95: Resultado de prueba CU31

Prueba superada con éxito



### 3.5.5 CU36.- Modificar Mensaje Publicitario

#### 3.5.5.1 Descripción

Ingresamos a la herramienta Web como usuario Gerente, accediendo a al módulo de Sistema, elegimos la opción Mensaje Inferior el sistema nos mostrará una interfaz para que llevemos a cabo la modificación del mensaje.

#### 3.5.5.2 Condiciones de Ejecución

Las condiciones del caso de prueba son que el usuario Gerente ingresa un nuevo mensaje informativo e intenta guardarlo, espereamos ver la respuesta del sistema cuando el usuario presione el botón Guardar

#### 3.5.5.3 Entrada

Los datos ingresados se muestran en la figura:

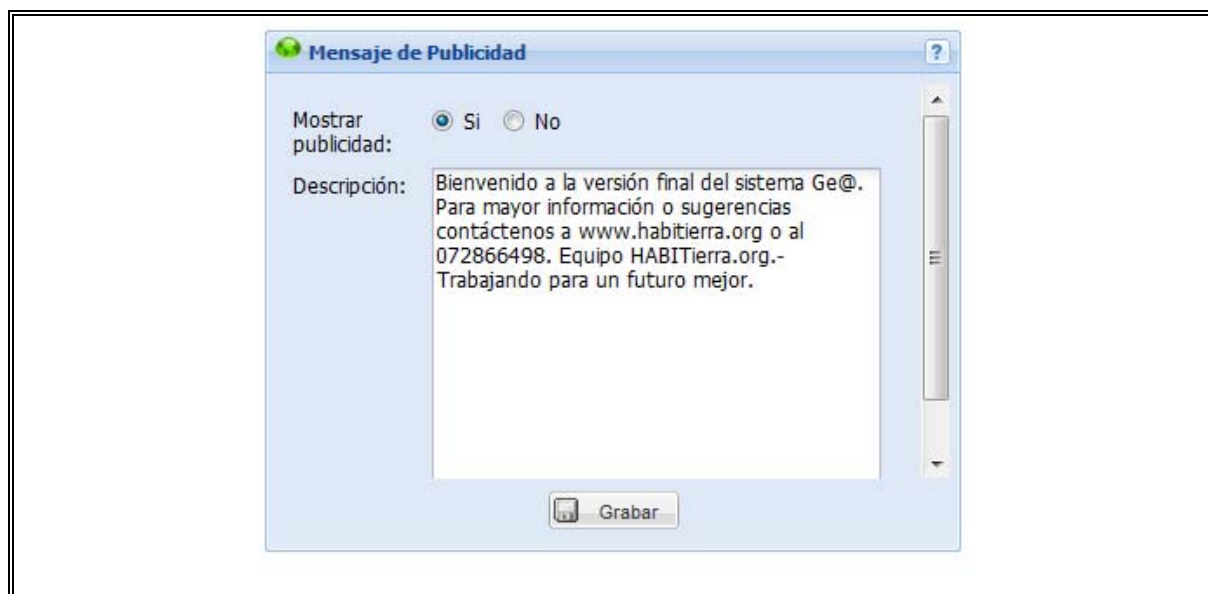


Figura 3-96: Datos de prueba CU36

#### 3.5.5.4 Resultado Esperado

Nuevo mensaje de parte de la Gerencia deslizándose en la parte inferior de la interfaz de usuario.

### 3.5.5.5 Evaluación de la prueba

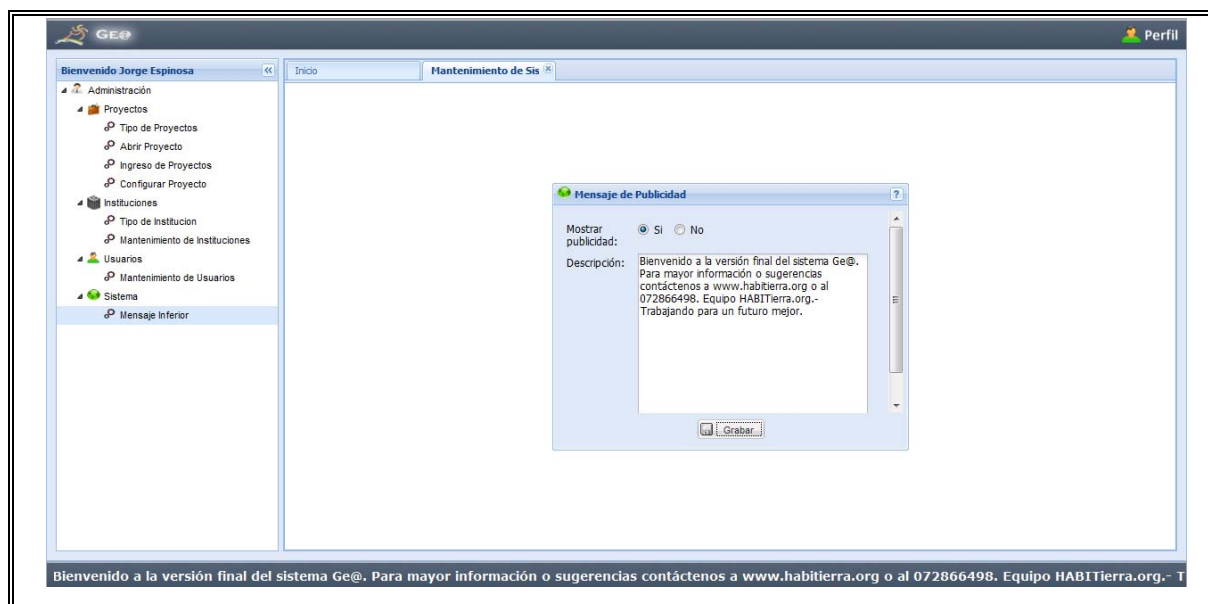


Figura 3-97: Resultado de prueba CU36

Prueba superada con éxito

## 3.5.6 CU48.- Manipular Objetos

### 3.5.6.1 Descripción

Ingresamos a la herramienta Web como usuario Técnico o Usuario común, accediendo a al módulo de proyectos, elegimos alguno existente o creamos uno nuevo, ingresamos a uno de los 9 módulos existentes para el Modelamiento Digital del proyecto, el aquella interfaz seleccionamos el ícono Crear Nueva Dimensión, el sistema nos mostrará una interfaz para que llevemos a cabo el ingreso del nuevo objeto y sus características (Introducción, Interacciones, Archivos,etc). Intentaremos guardar los nuevos datos y veremos cómo responde el sistema.

### 3.5.6.2 Condiciones de Ejecución

Dentro del módulo Principios Esenciales, del proyecto Pot El Oro, creamos un nuevo objeto denominado “Nuevo Principio Esencial”, detallamos una introducción y generaremos 3 interacciones con el resto de Principios Esenciales.

### 3.5.6.3 Entrada

Los datos ingresados se muestran en la figura:

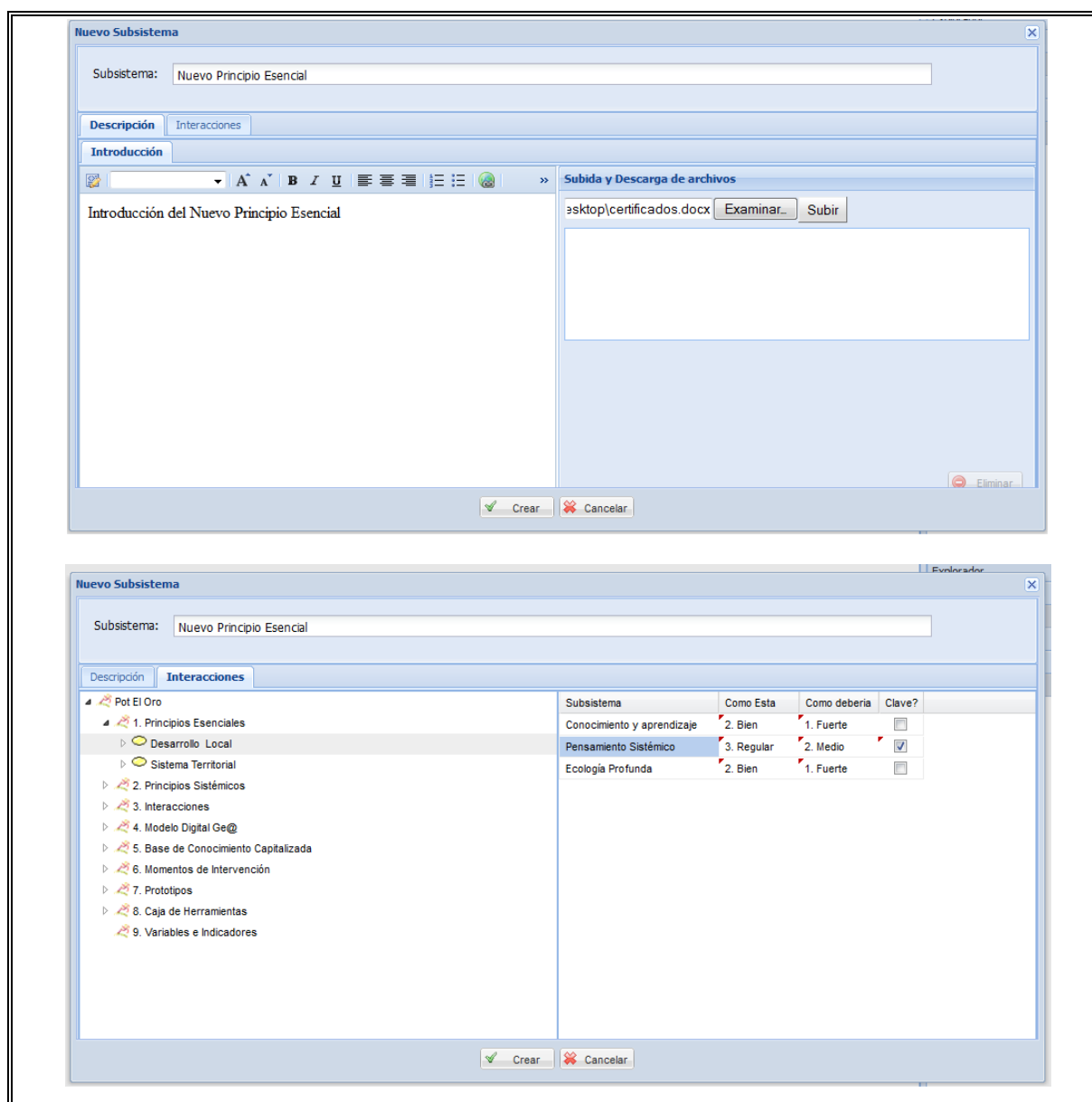


Figura 3-98: Datos de prueba CU48

### 3.5.6.4 Resultado Esperado

El sistema presenta un nuevo Principio Esencial, se alinea y ordena automáticamente con el resto de principios y muestra también las interacciones generadas.

### 3.5.6.4 Evaluación de la prueba

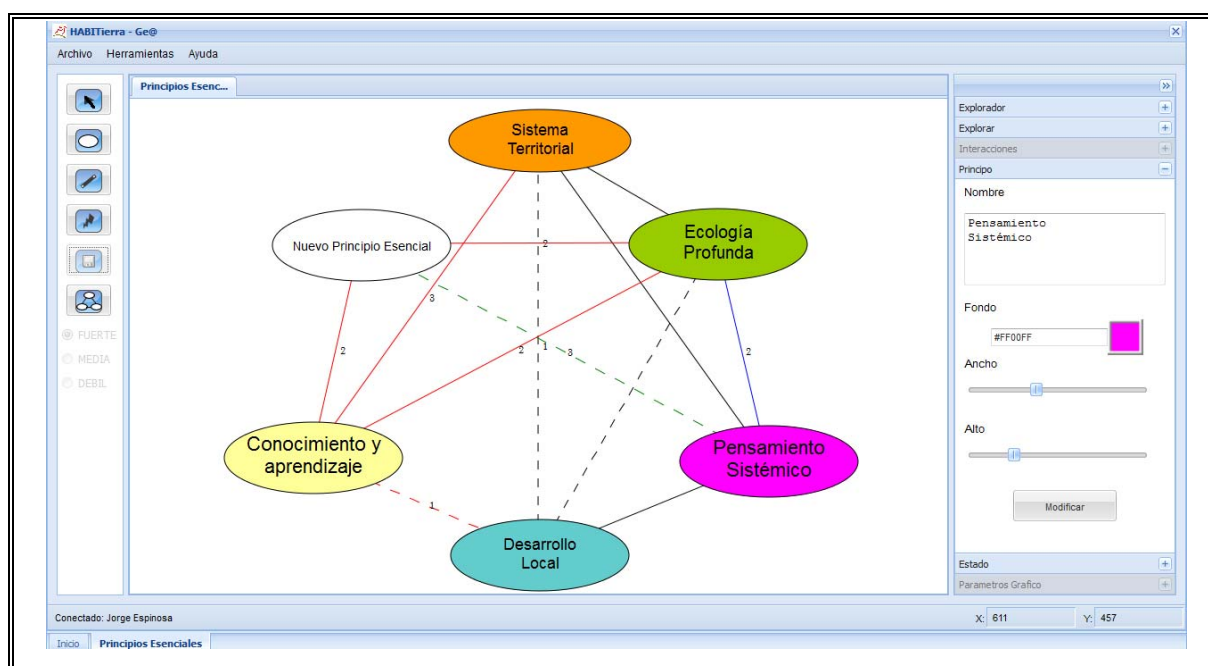


Figura 3-99: Resultado de prueba CU48

Prueba superada con éxito

## 3.5.7 CU49.- Indicar Causas y Razones

### 3.5.7.1 Descripción

Ingresamos a la herramienta Web como usuario Técnico o Usuario común, accediendo a al módulo de proyectos, elegimos alguno existente o creamos uno nuevo, ingresamos a uno de los 9 módulos existentes para el Modelamiento Digital del proyecto, el aquella interfaz seleccionamos una interacción, damos clic derecho y elegimos la opción Causas y Razones. En la interfaz que se muestra llenamos los datos de Causas y Razones, intentaremos guardar los nuevos datos y veremos cómo responde el sistema.

### 3.5.7.2 Condiciones de Ejecución

El usuario accede a la interacción que existe entre los sistemas Ecología Profunda y Nuevo Principio Esencial. Las propiedades de interacción están establecidas así: Está= Bien, Debería Ser = Fuerte. El Usuario explica el porqué debería ser fuerte.

### 3.5.7.3 Entrada

Los datos ingresados se muestran en la figura:

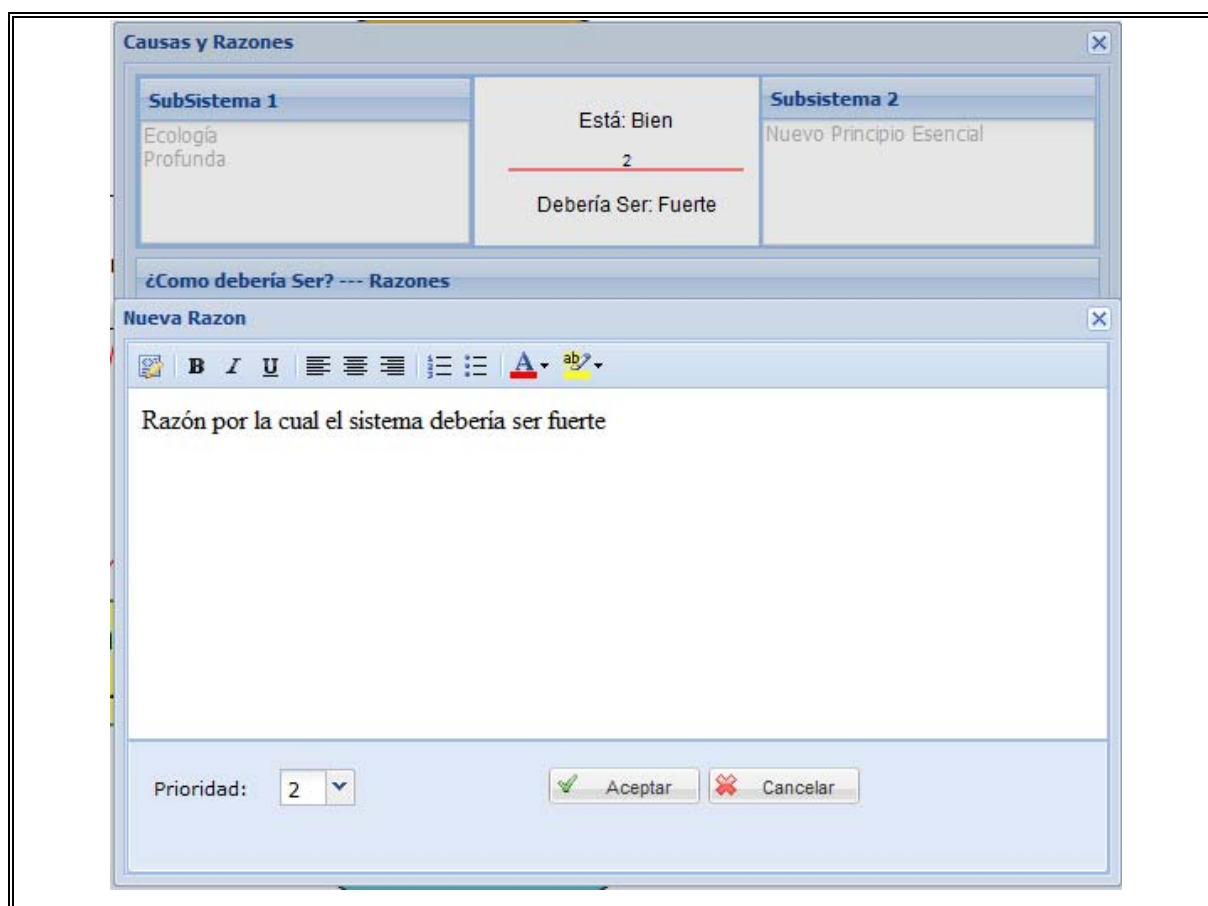


Figura 3-100: Datos de prueba CU49

### 3.5.7.4 Resultado Esperado

Se ingresa la nueva razón.

### 3.5.7.5 Evaluación de la prueba

The screenshot shows a software window titled "Causas y Razones". It contains the following elements:

- SubSistema 1:** A table with one row containing "Ecología Profunda".
- Subsistema 2:** A table with one row containing "Nuevo Principio Esencial".
- Summary:** In the center, it says "Está: Bien" above a red horizontal line with the number "2" below it, and "Debería Ser: Fuerte" below that.
- ¿Como debería Ser? --- Razones:** A table with two columns: "Prioridad" and "Razones". It contains one row with "2" and "Razón por la cual el sistema debería ser fuerte".
- Buttons:** A green button with a plus sign and the text "Nueva Razón" is located below the "Razones" table.
- ¿Como está? --- Causas:** A table with two columns: "Prioridad" and "Causas". It is currently empty.
- Buttons:** A green button with a plus sign and the text "Nueva Causa" is located below the "Causas" table.

Figura 3-101: Resultado de prueba CU49

Prueba superada con éxito

## 3.5.8 CU50.- Mantenimiento de Realimentación

### 3.5.8.1 Descripción

Ingresamos a la herramienta Web como usuario Técnico o Usuario común, accediendo a al módulo de proyectos, elegimos alguno existente o creamos uno nuevo, entre los módulos de Modelamiento Digital del proyecto ingresamos al Módulo de Prototipos, elegimos un Prototipo, en uno de sus Sistemas damos clic derecho y elegimos la opción Diagrama Sistémico de las Variables, damos clic en alguna de las interacciones del Ciclo y procedemos a modificarla. Veremos cómo responde el sistema.

### 3.5.8.2 Condiciones de Ejecución

Nos encontramos en el diagrama de Variables Sistémicas del problema “El centralismo del poder en las autoridades locales” encontrado en un Prototipo. Elegimos una interacción y procedemos a indicar que la misma genera una demora y que aporta a aumentar ya sea al crecimiento o disminución del problema. Vemos cómo responde el sistema

### 3.5.8.3 Entrada

Los datos ingresados se muestran en la figura:

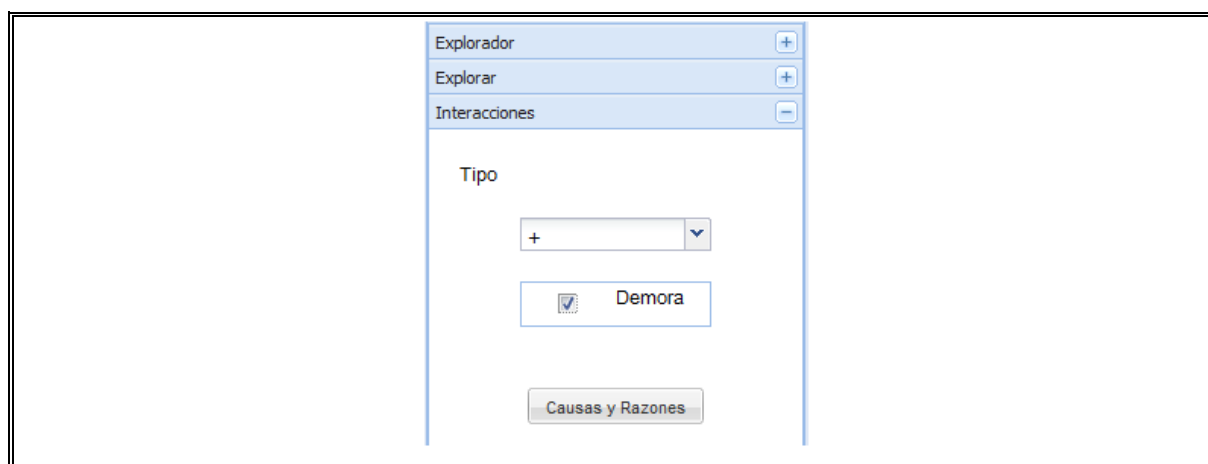


Figura 3-102: Datos de prueba CU50

### 3.5.8.4 Resultado Esperado

De manera automática se deben actualizar estos cambios en el gráfico con los nuevos parámetros configurados.

### 3.5.8.5 Evaluación de la prueba

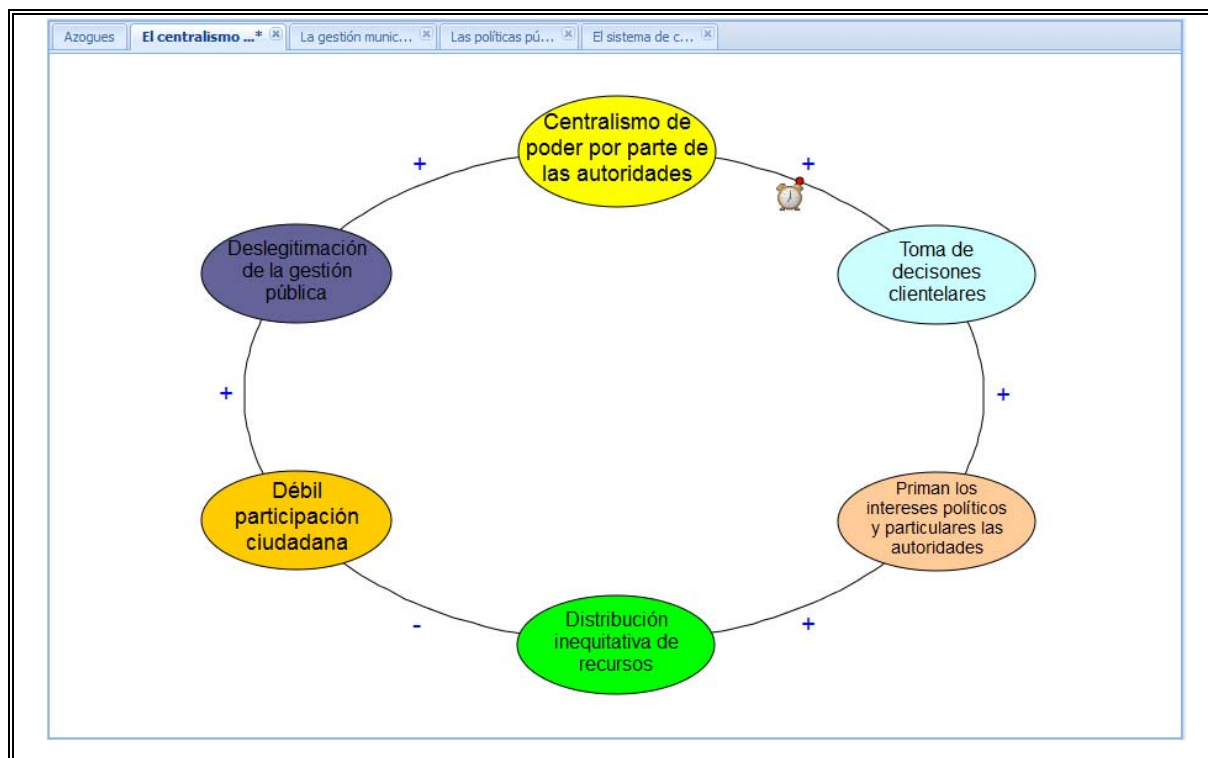


Figura 3-103: Resultado de prueba CU50

Prueba superada con éxito

### 3.5.9 CU52.- Generar Sumas

#### 3.5.9.1 Descripción

Ingresamos a la herramienta Web como usuario Técnico o Usuario común, accediendo a al módulo de proyectos, elegimos alguno existente o creamos uno nuevo, ingresamos a uno de los 9 módulos existentes para el Modelamiento Digital del proyecto, en aquella interfaz desplegamos la opción Explorar del panel de herramientas (parte derecha), damos clic derecho sobre el primer sistema que ahí aparece (para visualizar las sumas en el nivel actual) y elegimos la opción sumas.



### 3.5.9.2 Condiciones de Ejecución

Las condiciones de ejecución del caso de prueba son que el usuario desea conocer cómo están las sumas de los Principios Esenciales del Proyecto Pot El Oro, a través del menú contextual pide ver las sumas al sistema.

### 3.5.9.3 Entrada

Los datos ingresados se muestran en la figura:

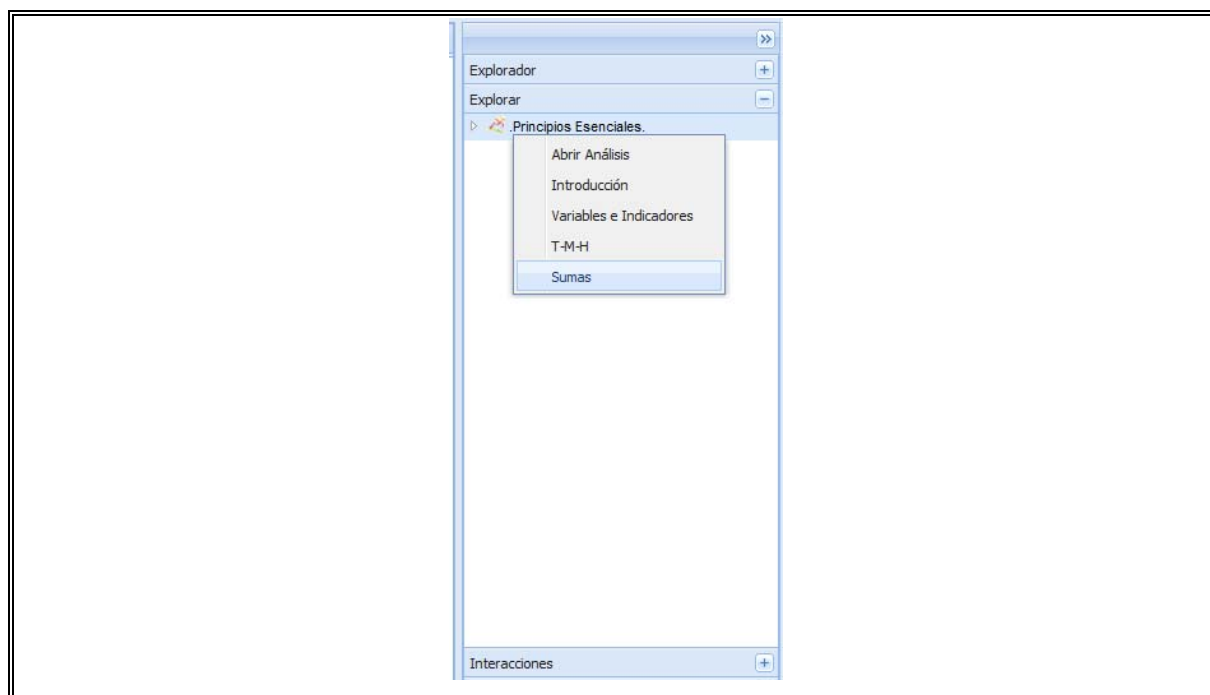


Figura 3-104: Datos de prueba CU52

### 3.5.9.4 Resultado Esperado

Gráfico de barras especificando cómo se encuentran los Principios Esenciales mediante el valor de sus interacciones

### 3.5.9.5 Evaluación de la prueba

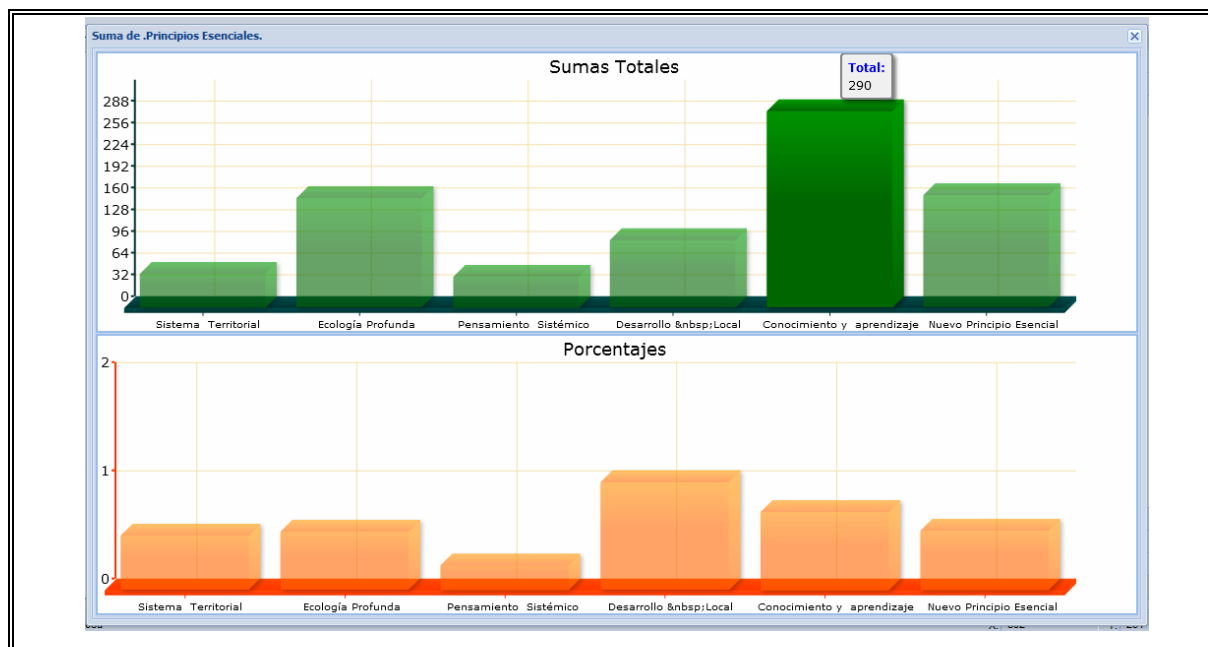


Figura 3-105: Resultado de prueba CU52

Prueba superada con éxito

### 3.5.10 CU51.- Generar Reporte

#### 3.5.10.1 Descripción

Ingresamos a la herramienta Web como usuario Gerente, accediendo a al módulo de proyectos, elegimos alguno existente o creamos uno nuevo, ingresamos a uno de los 9 módulos existentes para el Modelamiento Digital del proyecto, en aquella interfaz desplegamos la opción Estado del panel de herramientas (parte derecha), damos clic en exportar.

#### 3.5.10.2 Condiciones de Ejecución

Las condiciones de ejecución del caso de prueba son que el usuario desea generar un reporte de los Principios Esenciales del Proyecto POT El Oro, elige generar un archivo PDF y pulsa el botón Exportar.

### 3.5.10.3 Entrada

Los datos ingresados se muestran en la figura:

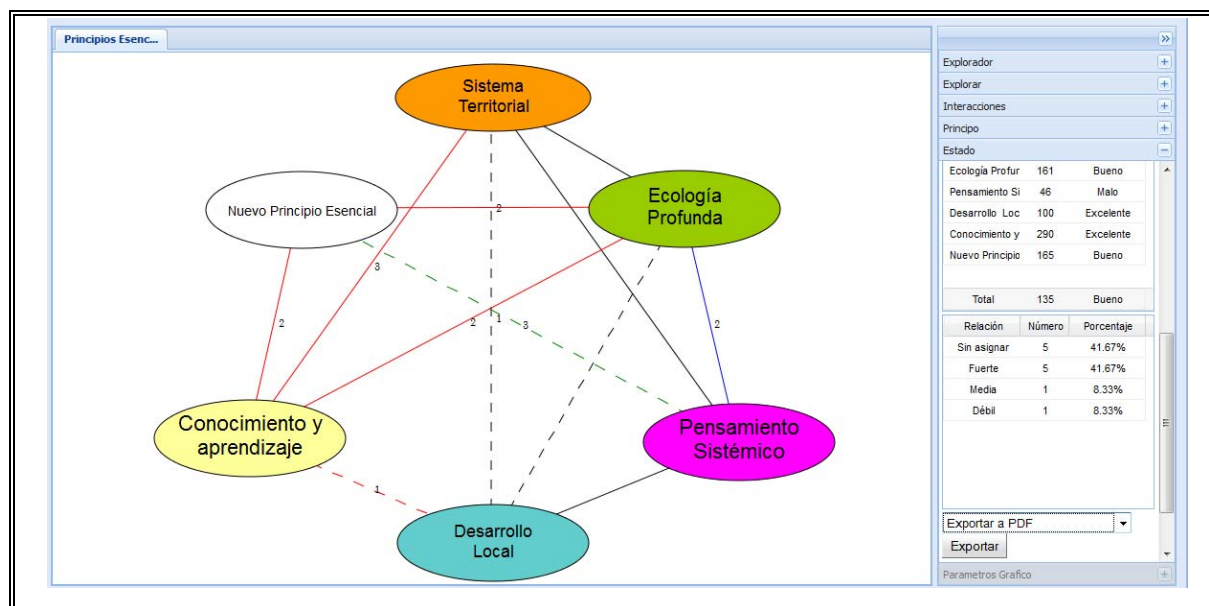


Figura 3-106: Datos de prueba C51

### 3.5.10.4 Resultado Esperado

Reporte en formato PDF de los Principios Esenciales del proyecto Pot El Oro.

### 3.5.10.5 Evaluación de la prueba

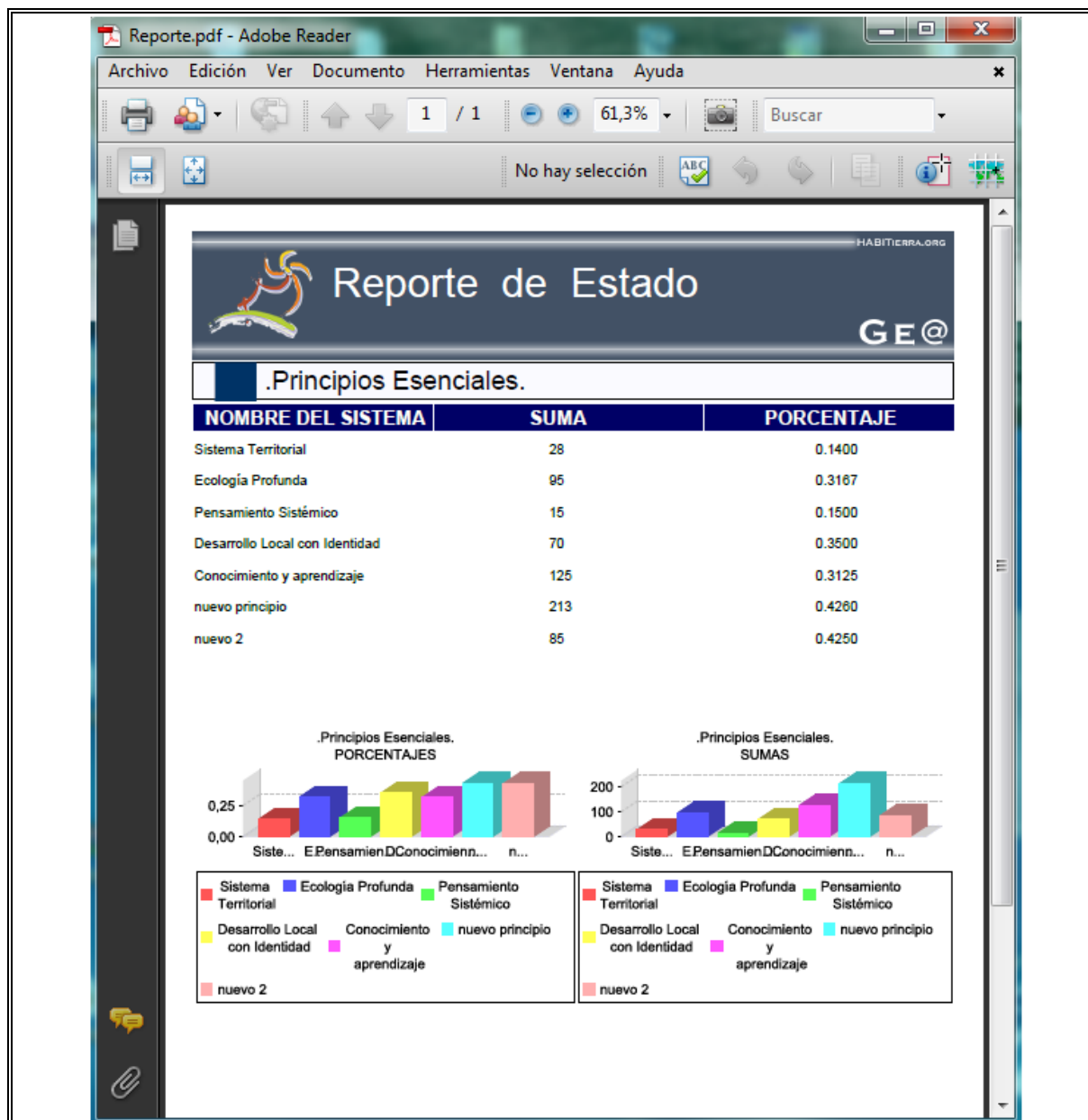


Figura 3-107: Resultado de prueba CU51

Prueba superada con éxito

### 3.6 Despliegue

El diagrama de despliegue es utilizado dentro de la metodología RUP para modelar el hardware utilizado en las implementaciones de la herramienta web y las relaciones entre sus componentes.

El siguiente diagrama muestra la ubicación del equipo servidor ubicado en la sucursal de cuenca de la fundación HABITierra y como es usado de almacén central de los datos, además de contener la “Herramienta web para la gestión, planificación y gobierno de territorios”.

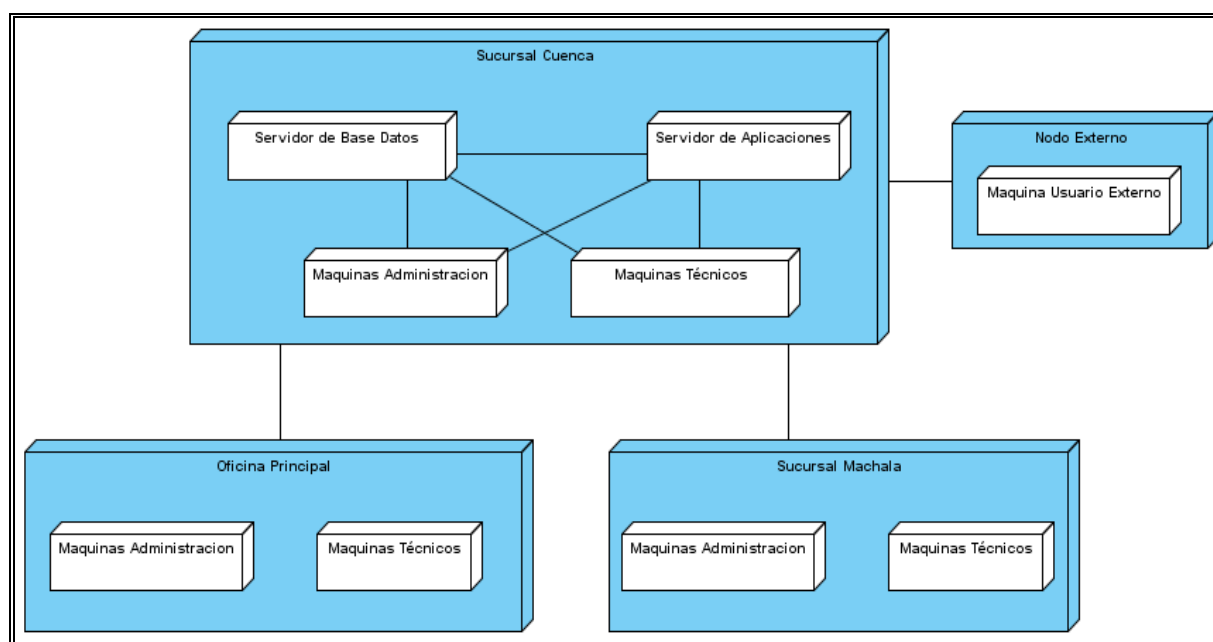


Figura 3-108: Diagrama de despliegue de la “Herramienta web para la planificación, gestión y gobierno de territorios”

#### 3.6.1 Manual de Usuario

Revisar ANEXO A.

## **CAPITULO 4**

#### 4.1 CONCLUSIONES

Al haber culminado con el presente proyecto previo a la obtención del título de Ingeniero en Sistemas, hemos llegado a las siguientes conclusiones:

- La herramienta web para la planificación, gestión y gobierno de territorios, es ahora un gran aporte y una realidad para la Fundación HABITierra, pues mediante este proyecto ha alcanzado un nuevo nivel en cuanto a la capitalización del conocimiento humano, fieles a su filosofía lograda con años de experiencia e investigación y ahora con herramientas que hacen uso de las TI, seguramente les permitirán promover, apoyar y participar en el cambio de los sistemas territoriales hacia una transición de sostenibilidad integral.
- Al tratarse de un sistema orientado a la Web las limitaciones para el acceso de parte de los usuarios son mínimas. Esto no beneficia únicamente a los usuarios finales, sino también a quienes gestionan y administran al sistema, pues no se depende de un único equipo que contenga el sistema instalado sino únicamente de una conexión a internet.
- Mediante el proyecto hemos podido descubrir la existencia de potentes herramientas que nos ayudan a desarrollar aplicaciones web con relativa facilidad a través de componentes que además de ser dinámicos y facilitar tareas al programador siendo reutilizables, mejoran la calidad de interfaz del sistema consiguiendo incluso que los usuarios conciban a la aplicación web como si se tratase de un sistema de escritorio.
- El éxito de cualquier sistema ya sea web o de escritorio radica en elegir de manera correcta y seguir cuidadosamente la metodología de desarrollo. RUP ha conseguido mediante sus artefactos e iteraciones que la herramienta web para la planificación, gestión y gobierno de territorios cumpla con las expectativas deseadas de la Fundación.
- Hemos entendido que el perfil del ingeniero en sistemas debe de abarcar y estar abierto al aprendizaje de otros ámbitos del saber humano, su fin es el dar soluciones ágiles, eficientes y solventes a cualquier problema que le presenten, obligándole así a desenvolverse de manera astuta y receptiva a la vez de tener facilidad de comunicación con profesionales de cualquier nivel y a pesar de que sus conocimientos técnicos sean muy escasos llegar a un entendimiento de sus problemas y necesidades

## 4.2 RECOMENDACIONES

- El nivel de éxito que se tenga al manejar esta herramienta web está en relación directa con el conocimiento del modelo teórico que rige al Sistema, por lo que se recomienda que para explotar las buenas funcionalidades con las que cuenta, se tenga un buen nivel de entendimiento de los paradigmas con que aquí se trabaja.
- Es recomendable usar el navegador de internet Mozilla Firefox para no tener ningún problema de compatibilidad. El sistema funciona correctamente en la mayoría de navegadores exceptuando el Internet Explorer con el cual hay inconvenientes con librerías gráficas, específicamente con GWT-Graphics.
- En lo que respecta a las tecnologías usadas en este proyecto, hemos quedado bastante satisfechos, recomendamos la investigación y uso de nuevas y mejoradas herramientas que cada vez brindan mayores facilidades y posibilidades al desarrollador.



## BIBLIOGRAFIA

Justin Gehtland, Ben Galbraith, Dion Almaer (ISBN—13:978-0976694083). **PRAGMATIC AJAX: A WEB 2.0 PRIMER.** *Se trata de un libro muy práctico que incluye muchos ejemplos reales fáciles de entender a pesar de su complejidad.* Ver más información sobre el libro. (<http://www.pragmaticprogrammer.com/title/ajax/>).

Dave Crane, Eric Pascarello, and Darren James (ISBN-13: 978-1932394610). **AJAX IN ACTION.** *El código de los ejemplos incluidos está muy bien programado, lo que ayuda a crear aplicaciones muy profesionales.* Ver más información sobre el libro (<http://www.manning.com/crane/>).

Bruce Perry (ISBN-13: 978-0596101695). **AJAX HACKS.** *Colección de trucos y pequeñas utilidades listas para copiar+pegar en las aplicaciones reales.* Ver más información sobre el libro (<http://www.oreilly.com/catalog/ajaxhks/>).

W. Richard Stevens (1990). **UNIX NETWORK PROGRAMMING.** Prentice Hall Publisher.

John Bloomer and O'reilly & Associates. (March, 1992). **POWER PROGRAMMING WITH RPC.**

Andrew S. Tanenbaum. (1996). **SISTEMAS OPERATIVOS DISTRIBUIDOS.** Prentice-Hall Publisher.

Ryan Dewsbury (2007). **GOOGLE WEB TOOLKIT APPLICATIONS.** Prentice Hall Publisher.

Arq. Pedro Seas (2002). **GE@.** Habitierra.org

Isomorphic (2010). **SMART GWT™ QUICK START GUIDE.** Isomorphic Software, Inc. Ver más en <http://www.isomorphic.com>.

Sitio Web Programatium. **ENTENDIENDO AJAX.** Ver más información en <http://www.programatium.com/manuales/ajax/entendiendo-ajax.htm>

Sitio Web Wikipedia Español. **AJAX.** Ver más información en <http://es.wikipedia.org/wiki/AJAX>.

Sitio Web Programación Web. **¿QUÉ ES AJAX?** Ver más información en <http://www.programacionweb.net/articulos/articulo/?num=317>.

Sitio Web Wikipedia Español. **RPC.** Ver más información en <http://es.wikipedia.org/wiki/RPC>

Sitio Web Culturación. **¿QUÉ SON LOS RPC?** Ver más información en <http://culturacion.com/2009/04/que-son-los-rpc-llamadas-de-procedimiento-remoto/>

Sitio Web Textos Científicos. **RPC**. Ver más información en <http://www.textoscientificos.com/redes/tcp-ip/servicios-capa-transporte/rpc>

Sitio Web Java Hispano. **BIBLIOTECA DE WIDGETS PARA GWT**. Ver más información en [http://www.javahispano.org/contenidos/es/smartgwt\\_1\\_0\\_\\_biblioteca\\_de\\_widgets\\_para\\_google\\_web\\_toolkit/](http://www.javahispano.org/contenidos/es/smartgwt_1_0__biblioteca_de_widgets_para_google_web_toolkit/)

Sitio Web Hibernate Community Documentation. **LA PRIMERA APLICACIÓN HIBERNATE**. Ver más información en <http://docs.jboss.org/hibernate/core/3.6/reference/es-ES/html/tutorial.html#tutorial-firstapp>

SitioWeb Section508.gov. **SECCIÓN 508 ARIA**. Ver más información en: <http://www.section508.gov/index.cfm?fuseAction=stdsdoc>

SitioWeb MySQL. **PANORÁMICA DEL SISTEMA DE GESTIÓN DE BASE DE DATOS MYSQL**. Ver más información en: <http://dev.mysql.com/doc/refman/5.0/es/what-is.html>

SitioWeb GWT Guide. **SETTING UP EXT GWT IN NETBEANS GWT PROJECT**. Ver más información en: <http://gwt-guide.blogspot.com/2010/05/setting-up-ext-gwt-in-netbeans-gwt.html>

SitioWeb Sencha. **GETTING STARTED**. Ver más información en: <http://dev.sencha.com/deploy/gxt/screencasts/gettingstarted/step5-project/index.html>

SitioWeb Centro de Especialización Profesional y Extensión Universitaria: CEPEU. **COMBINANDO REPORTES EN JAVA**. Ver más información en: [http://www.cepeu.edu.py/LIBROS\\_ELECTRONICOS\\_3/lpcu089%20-%2001.pdf](http://www.cepeu.edu.py/LIBROS_ELECTRONICOS_3/lpcu089%20-%2001.pdf)

# **ANEXOS**

## ANEXO A: MANUAL DE USUARIO

La herramienta web para la gestión, planificación y gobernabilidad territorial se maneja en base a proyectos y sistemas con sus respectivos elementos, interacciones, etc. además de ser una aplicación multiusuarios, se maneja un sistema de logging con el fin de proteger los datos de cada usuario.

La estructura de usuarios se basa en dos tipos:

- Administrador
- Invitado

Cada usuario accede a diferentes áreas dentro del Sistema Ge@, pero las operaciones que efectúa son las mismas con respecto al manejo de sus propios proyectos.

A continuación se especifica con mayor detalle las operaciones que efectúa tanto el Administrador como el usuario Invitado, y que de paso este documento sirva de guía para los nuevos miembros que decidan integrarse al conjunto de usuarios que forman parte del Sistema Ge@.

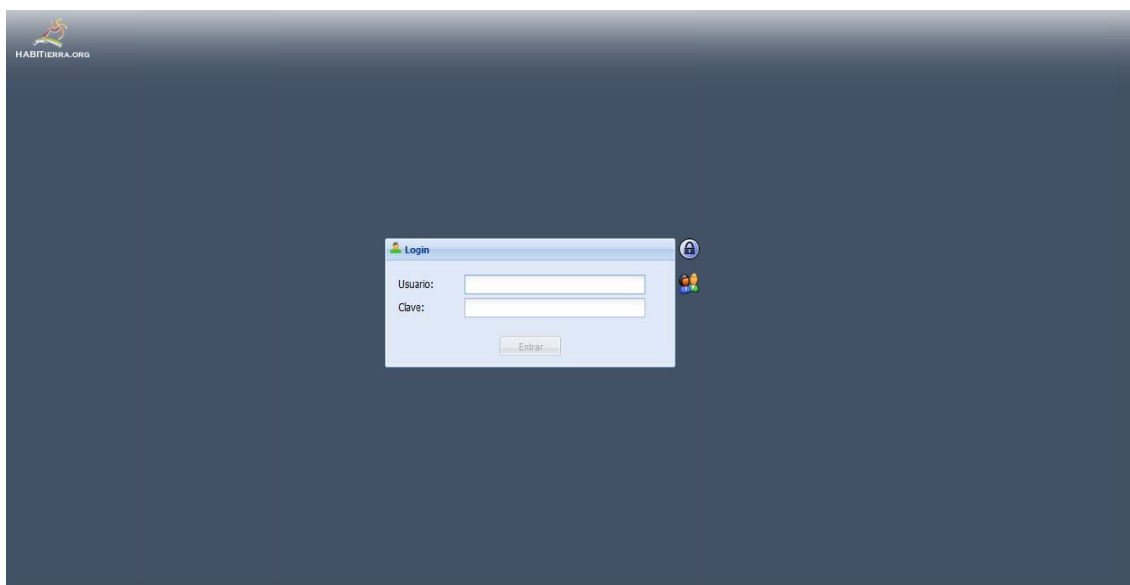
Para ingresar al sistema se lo puede realizar en cualquier navegador web ya sea este Mozilla Firefox, Opera, Safari, Google Chrome, para los navegadores Internet Explorer aún existen problemas de compatibilidad que en un futuro se van a resolver, de preferencia se recomienda a los usuarios que hagan uso del navegador Mozilla Firefox.

En la barra de direcciones del navegador colocamos la siguiente url:  
<http://201.238.150.105:9090/gea/>

En este paso vale la pena aclarar que el sistema pasa por un proceso de carga de todos los elementos que conforman la aplicación web. Si es la primera vez que el sistema se está cargando en su navegador, puede que el proceso tome un número determinado de minutos en cargarse dependiendo de la conexión de internet que disponga, por lo general toma de 1min a 2min, le recomendamos que sea paciente y espere a que todos los elementos se terminen por cargar. Una vez que el sistema se haya cargado, las siguientes veces que se ejecute el proceso será mucho más rápido ya que el sistema hace uso de las cookies del navegador.

## LOGGING

Una vez que haya terminado el proceso de carga, la primera pantalla en aparecer es la ventana de Logging, aquí el usuario ingresa su Username y Password que se le asignaron al momento de registrarse.



En la pantalla de Logging existen dos botones:



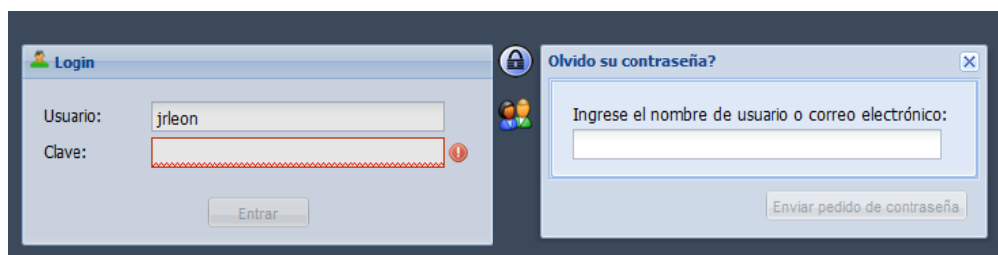
*Registro de Nuevo Usuario*



*¿Olvido su contraseña?*

### Botón “¿Olvido su contraseña?”

Este botón debe ser pulsado únicamente si el usuario no recuerda la contraseña para ingresar al sistema, en este caso se presenta una nueva ventana como se puede visualizar en la siguiente imagen:



En esta nueva ventana, el usuario debe ingresar su correo electrónico o su nombre de usuario, en ambos casos, la administración de Ge@ enviará a su correo un mensaje en el cual se presentará su nueva contraseña y con la cual va a ingresar al sistema.

Es recomendable que una vez que el usuario consiga su nueva contraseña deba inmediatamente modificarla dentro de su perfil, esto por motivos de seguridad.

### **Botón “Registro de nuevo usuario”**

Este botón lleva al usuario a una nueva ventana en donde procederá al ingreso de sus datos personales, el usuario no se debe preocupar si por alguna razón se equivoca ya que estos pueden ser modificados luego dentro del perfil del usuario.



El registro de un nuevo usuario implica el ingreso de los siguientes datos:

- **Tipo Institución:** se refiere al tipo de institución a la cual corresponde la institución en donde opera el usuario.
- **Institución:** es la institución en donde trabaja el usuario a registrarse.
- **Cédula:** se ingresa el número de cédula del usuario.
- **Nombre:** se ingresa el nombre del usuario.
- **Apellido:** se registra el apellido del usuario.
- **Teléfono:** se registra el teléfono del usuario, esto con el objetivo de brindar una posterior asesoría por parte de la administración de Ge@.
- **Dirección:** se ingresa la dirección del usuario.

- **Email:** se registra la dirección de correo electrónico del usuario con el mismo objetivo de brindar una asesoría posterior al usuario además de trabajar con el usuario constantemente.

Todos los datos son necesarios ingresar para poder continuar con el proceso de registro. Luego, se procede a pulsar el botón guardar y el usuario será registrado.

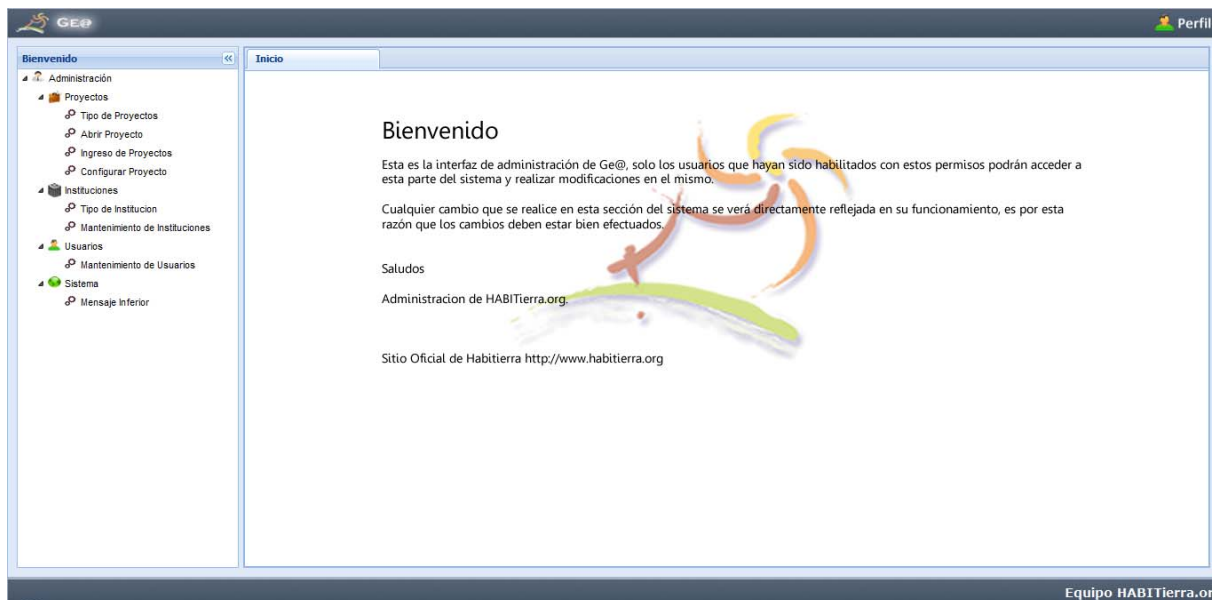
**NOTA:** Como se puede observar el usuario nunca ingresa su nombre de usuario y contraseña esto se debe a que el sistema automáticamente asigna a la primera parte de su correo como nombre de usuario y a su cedula como contraseña. Por ejemplo, si usted ingreso a mail-usuario@correo.com y 0123456789 como cédula sus datos de Logging serán los siguientes:

- **Nombre de Usuario:** mail-usuario
- **Contraseña:** 0123456789

Es importante recalcar que el usuario que se registra de esta manera será un usuario INVITADO.

## ADMINISTRADOR

Este tipo de usuario puede acceder a la página de administración de Ge@, aquí se presenta la pantalla principal del Sistema Ge@ del administrador, esta se divide en 4 áreas.

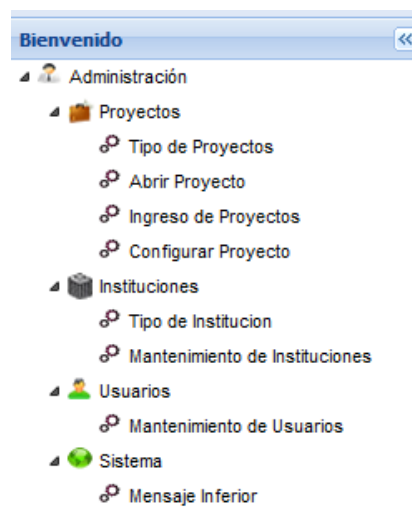


1. Panel Navegador
2. Área de Trabajo
3. Barra de Titulo y Menú Perfil de Usuario.
4. Mensaje de Publicidad.



## Panel Navegador

El área de panel navegador sirve de ayuda al usuario para acceder a los distintos módulos de administración con los que dispone Ge@. Estos modulos se detallarán en las siguientes secciones. Como resumen podemos indicar que la administración se maneja en base a 4 módulos.



- Administración de Proyectos
  
- Administración de Instituciones
  
- Administración de Usuarios
  
- Administración del Sistema.

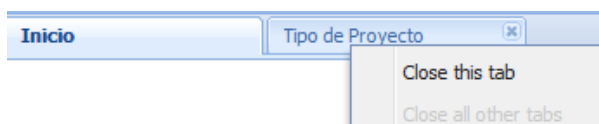
## Área de Trabajo

El área de trabajo es la zona más grande de la pantalla inicial aquí el usuario podrá visualizar todos los módulos que abra a partir del panel de navegador.



El área de trabajo se basa en pestañas la razón de esto es el de facilitarle al usuario para que pueda desplazarse entre los distintos módulos sin tener que cerrarlos. En la imagen se observa la pestaña principal del área de trabajo el cual se le da una bienvenida al usuario administrador.

El usuario puede cerrar cualquier pestaña al dar clic en la x que se presenta en la esquina superior derecha o dando clic derecho sobre la pestaña en este caso se presenta un menú como se puede ver en la imagen:



**Cierra la pestaña**

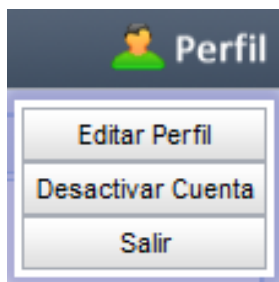
**Cierra todas las pestañas  
menos la seleccionada**

## Barra de Título y Menú de Perfil de Usuario

La barra de título presenta el nombre del sistema web, además de permitir la edición del perfil del usuario como podemos notar en la siguiente imagen.



Al acercar el mouse al botón perfil se presentara un menú, en donde el usuario puede realizar las siguientes operaciones:



Permite la edición de los datos del usuario, los mismos que ingresaron al momento del registro

Permite la desactivación de la cuenta del usuario, es necesario que el usuario esté completamente seguro de desactivar su cuenta ya que la reactivación lo puede realizar únicamente el administrador del sitio.

Permite salir del sistema y regresar a la pantalla de Login

### Edición del Perfil de Usuario

La pantalla de edición de perfil de usuario es similar a la de registro, con la diferencia que el usuario puede modificar también su nombre de usuario y contraseña.

En este caso se presenta unas variaciones, si el usuario no necesita modificar el nombre de usuario y contraseña simplemente no modifica el nick y no ingresa una nueva contraseña, en caso contrario debe registrarlos.

 A screenshot of the "Editar perfil" (Edit Profile) form. The form is titled "Editar perfil" and has a "Perfil" icon in the top right corner. On the left side, there is a placeholder image of a man in a suit with the word "USUARIOS" below it. The form is divided into two main sections: "Datos del Usuario" and "Datos de nombre de usuario y contraseña".
   
 The "Datos del Usuario" section contains the following fields:
 

- Tipo: A dropdown menu with "XXXXXXXXXX" selected.
- Institución: A dropdown menu with "Escoge el tipo institución" as the prompt.
- Institución: A dropdown menu with "XXXXXXXXXX" selected and "Escoge la institución" as the prompt.
- Cédula: A text input field with "XXXXXXXXXX" and the prompt "Ingrese la cédula del usuario".
- Nombre: A text input field with "XXXXXXXX" and the prompt "Ingrese el nombre del usuario".
- Apellido: A text input field with "XXXXXXXX" and the prompt "Ingrese el apellido del usuario".
- Teléfono: A text input field with "0000000" and the prompt "Ingrese el teléfono del usuario".
- Dirección: A text input field with "Xxx, XXXXXXX" and the prompt "Ingrese la dirección del usuario".
- Email: A text input field with "xxxxxxxx@xxxxx.com" and the prompt "Ingrese la dirección del usuario".

 The "Datos de nombre de usuario y contraseña" section contains:
 

- Nombre de usuario: A text input field with "XXXXXXXXX" and the prompt "Ingrese un nombre de usuario".
- Password: A text input field with the prompt "Ingrese una contraseña".
- Confirmar Password: A text input field with the prompt "Por favor, reingrese la contraseña".

 At the bottom right of the form, there are two buttons: "Guardar" and "Cancelar".

## Mensaje de Publicidad

La barra inferior de la página de administración muestra un mensaje que se desplaza de derecha a izquierda de manera constante, esto con la idea de comunicar a los demás usuarios, mensajes que la administración de Ge@ desea informar a su comunidad. Este mensaje puede ser modificado únicamente por el administrador y en el módulo de administración del sistema.

Equipo HABITierra.org

## Módulos del Sistema Ge@

Como se describió en secciones anteriores el administrador dispone de un panel de navegador por donde el usuario va a acceder a los distintos módulos. A continuación se detalla los distintos módulos y su manera de operar.

### Administración de Proyectos

#### Tipo de Proyectos

Esta pestaña permite al administrador el registro, modificación y desactivación de un tipo de proyecto.

Tipo de Proyecto	Estado
Ordenamiento Territorial	Activo
Sistemas Urbanos Complejos	Activo
Regiones Bioculturales	Activo
Recursos y Riesgos	Activo
Gobernabilidad Territorial	Activo
Sistemas Gubernamentales	Desactivado
Jorge Leon	Desactivado
prueba	Desactivado

Nombre:   
Ingrese el nombre del tipo de proyecto

Descripción:   
Ingrese una descripción

Nuevo Desactivar Guardar

Page 1 of 1 Displaying 1 - 8 of 8

**Barra de paginación y recarga de la lista de tipos de institución.**

**Botones Nuevo, Desactivar y Guardar**

La pestaña presenta una tabla que muestra el listado de todos los tipos de proyectos al dar clic en una fila los datos de ese tipo se cargarán en el área de registro de datos. A estos datos se los puede modificar y para que los cambios efectuados se reflejen posteriormente se pulsa el botón **Guardar**. Otra tarea que se puede efectuar es la desactivación y se lo puede realizar al momento de pulsar el botón **Desactivar**.

En caso de que se desee crear un nuevo tipo, al pulsar el botón **Nuevo** se limpiaran los datos del área de registro y se procederá a ingresar los nuevos datos que conformarán el nuevo tipo de institución y una vez terminado se pulsa el botón **Guardar**.

En caso de existir más de 50 registros estos se estructuran en distintas páginas que por medio de la **barra de paginación** se va a poder acceder, cada página tiene un máximo de 50 ítems. Esta barra incluye los botones de avanzar, retroceder, ir a la primera página o ultima y un botón de refresco, este botón permitirá al usuario refrescar la grilla en caso de que los datos de la tabla no se visualicen o se hayan congelado.

### **Abrir Proyecto**

Esta pestaña brinda un listado de todos los proyectos que hayan ingresado todos los usuarios del sistema, tanto los usuarios invitados o demás administradores. En esta pestaña se puede visualizar 6 columnas:

1. El nombre del proyecto.
2. El tipo del proyecto.
3. La fecha de creación.
4. La fecha de su última actualización.
5. El usuario que creó el proyecto.
6. La institución a la cual pertenece el usuario.

Todas las grillas disponen de la barra de paginación para manejar de una mejor manera el gran número de proyectos que pueda tener el sitio.

Nombre del Proyecto	Tipo del Proyecto	Fecha de creación	Última Actualización	Usuario	Institución
Proyecto Marco Leon	Ordenamiento Territorial	2/23/11	3/3/11	Marco Leon	Municipio de Loja
Proyecto prueba mapa	Ordenamiento Territorial	3/2/11	3/2/11	Propio	Parroquias Rurales
Proyecto Prueba Mapas	Ordenamiento Territorial	3/2/11	3/3/11	Propio	Parroquias Rurales
Proyecto 4	Ordenamiento Territorial	12/15/10	3/3/11	Propio	Parroquias Rurales
Proyect 6	Ordenamiento Territorial	1/3/11	3/3/11	Propio	Parroquias Rurales
Proyecto 7	Ordenamiento Territorial	1/4/11	3/3/11	Propio	Parroquias Rurales
Proyecto de Prueba	Ordenamiento Territorial	2/23/11	3/3/11	Propio	Parroquias Rurales
Pot El Oro	Ordenamiento Territorial	12/15/10	5/5/11	Propio	Parroquias Rurales
Carcabon	Ordenamiento Territorial	5/5/11	5/5/11	Luis Robalino Mach	Parroquias Rurales
Barbones	Ordenamiento Territorial	5/5/11	5/5/11	José Luis Gómez Á	Parroquias Rurales
Laiberia	Ordenamiento Territorial	5/5/11	5/5/11	Pedro Lucas del Ro	Parroquias Rurales
Rio Bonto	Ordenamiento Territorial	5/5/11	5/5/11	Francisco Ramírez	Parroquias Rurales
Tendales	Ordenamiento Territorial	5/5/11	5/5/11	Rolando Suarez Ra	Parroquias Rurales
El Paraiso	Ordenamiento Territorial	5/5/11	5/5/11	Santos Hipólito Mac	Parroquias Rurales
La Libertad	Ordenamiento Territorial	5/5/11	5/5/11	Jimmy Castillo Abac	Parroquias Rurales
San Isidro	Ordenamiento Territorial	5/5/11	5/5/11	Manuel Orlando Roi	Parroquias Rurales
El Retiro	Ordenamiento Territorial	5/5/11	5/5/11	Sin Nombre	Parroquias Rurales
El Ingenio	Ordenamiento Territorial	5/5/11	5/5/11	José María Asanza	Parroquias Rurales
Buenavista	Ordenamiento Territorial	5/5/11	5/5/11	Yefferson Sandovz	Parroquias Rurales
Chacras	Ordenamiento Territorial	5/5/11	5/5/11	Sin Nombre	Parroquias Rurales


Page 1 of 2      Displaying 1 - 50 of 66

Abrir Proyecto

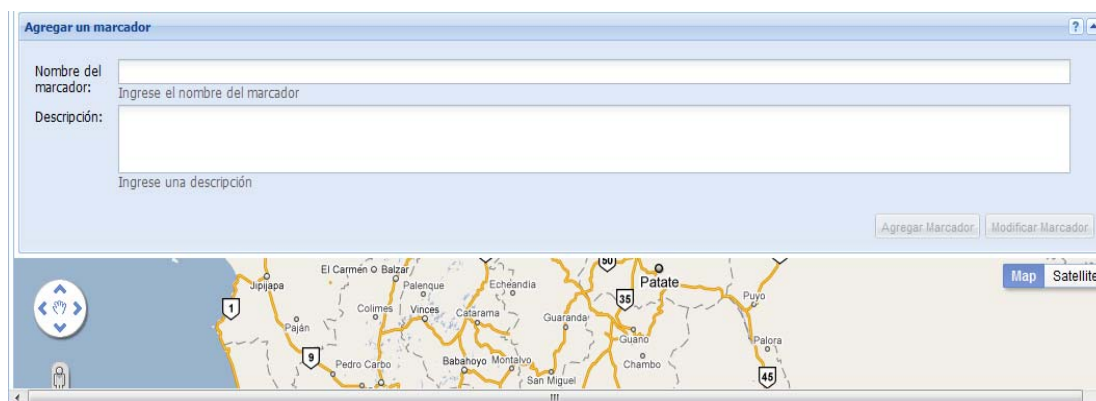
El proceso de abrir un proyecto es muy simple, se selecciona de la lista el proyecto a abrir y se pulsa el botón **Abrir Proyecto** que se encuentra localizado en la esquina inferior de la imagen y se procederá a ejecutar el sistema Ge@.

### Ingreso de Proyectos

El ingreso de proyectos resulta una tarea muy sencilla; se ingresan los datos que se solicitan para cada proyecto y son los que se puede visualizar en la imagen.

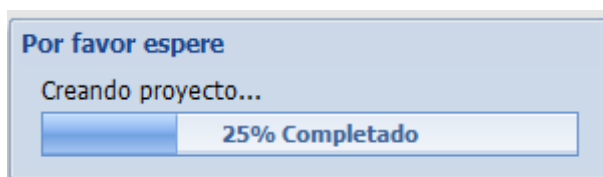
Datos del Proyecto	
Nombre de Proyecto:	<input type="text"/> Ingrese el nombre del proyecto
Tipo de Proyecto:	<input type="text"/> Escoga el Tipo de Proyecto
Descripción:	<input type="text"/> Ingrese una descripción del proyecto
<input type="button" value="Guardar"/> <input type="button" value="Cancelar"/>	
Ubicación en Google Maps	
<input type="text"/>	
	

Además, el usuario puede colocar un punto en el mapa de Google Maps, esto con la idea de que el administrador o invitado pueda reconocer la zona en donde se está llevando a cabo el proyecto. Para realizar esta tarea, el usuario debe expandir la ventana de **Agregar un marcador** y procederá a ingresar los datos necesarios para añadir al marcador y por ende al mapa.



La tarea de agregar un marcador no es obligatoria, es decir, el usuario puede como no agregar el marcador, si lo desea simplemente solo ingresa los datos del proyecto, estos datos pueden modificarse posteriormente.

Cuando haya creado el proyecto se presentara una barra de carga en la cual se le indica que se está configurando el proyecto para ser usado luego.



Para poder compartir un proyecto esto se lo hace desde la pestaña **Configurar Proyecto** que se describe en la siguiente sección.

### **Configurar Proyecto**

Dentro del módulo de proyectos el usuario puede también proceder a modificar o configurar el proyecto, para ello se entra en el panel navegador a la rama configurar proyecto, inmediatamente se carga una pestaña en donde se visualizará un listado de todos los proyectos del sistema, similar a lo que se visualizó al momento de abrir un proyecto.

La diferencia entre estas dos ventanas está en el grupo de botones que se visualizan en la parte inferior, en este caso el usuario puede configurar el sistema, desactivar el proyecto o eliminarlo permanentemente.

Nombre del Proyecto	Tipo del Proyecto	Fecha de creación	Última Actualización	Usuario	Institución	Estado
Proyecto Marco Leon	Ordenamiento Territorial	2/23/11	3/3/11	Marco Leon	Municipio de Loja	Activo
Proyecto prueba mapa	Ordenamiento Territorial	3/2/11	3/2/11	Propio	Parroquias Rurales	Activo
Proyecto Prueba Mapas	Ordenamiento Territorial	3/2/11	3/3/11	Propio	Parroquias Rurales	Activo
Proyecto 4	Ordenamiento Territorial	12/15/10	3/3/11	Propio	Parroquias Rurales	Activo
Proyect 6	Ordenamiento Territorial	1/3/11	3/3/11	Propio	Parroquias Rurales	Activo
Proyecto 7	Ordenamiento Territorial	1/4/11	3/3/11	Propio	Parroquias Rurales	Activo
Proyecto de Prueba	Ordenamiento Territorial	2/23/11	3/3/11	Propio	Parroquias Rurales	Activo
Pot El Oro	Ordenamiento Territorial	12/15/10	5/5/11	Propio	Parroquias Rurales	Activo
Carcabon	Ordenamiento Territorial	5/5/11	5/5/11	Luis Robalino Mach	Parroquias Rurales	Activo
Barbones	Ordenamiento Territorial	5/5/11	5/5/11	José Luis Gómez Á	Parroquias Rurales	Activo
Laiberia	Ordenamiento Territorial	5/5/11	5/5/11	Pedro Lucas del Ro	Parroquias Rurales	Activo
Rio Bonto	Ordenamiento Territorial	5/5/11	5/5/11	Francisco Ramirez	Parroquias Rurales	Activo
Tendales	Ordenamiento Territorial	5/5/11	5/5/11	Rolando Suarez Ra	Parroquias Rurales	Activo
El Paraiso	Ordenamiento Territorial	5/5/11	5/5/11	Santos Hipólito Mac	Parroquias Rurales	Activo
La Libertad	Ordenamiento Territorial	5/5/11	5/5/11	Jimmy Castillo Abac	Parroquias Rurales	Activo
San Isidro	Ordenamiento Territorial	5/5/11	5/5/11	Manuel Orlando Roi	Parroquias Rurales	Activo
El Retiro	Ordenamiento Territorial	5/5/11	5/5/11	Sin Nombre	Parroquias Rurales	Activo
El Ingenio	Ordenamiento Territorial	5/5/11	5/5/11	José María Asanza	Parroquias Rurales	Activo
Buenavista	Ordenamiento Territorial	5/5/11	5/5/11	Yefferson Sandove	Parroquias Rurales	Activo
Chacras	Ordenamiento Territorial	5/5/11	5/5/11	Sin Nombre	Parroquias Rurales	Activo

Page 1 of 2      Displaying 1 - 50 of 66

Configurar Proyecto    Desactivar Proyecto    Eliminar Proyecto

Se maneja bajo el mismo esquema se selecciona el proyecto y se pulsa el botón de acuerdo a la tarea que desea realizar.

### Configurar Proyecto

**Configurar Proyecto**

Datos del Proyecto    Compartir proyecto

Datos del Proyecto

Nombre de Proyecto: Pot El Oro  
Ingrese el nombre del proyecto

Tipo de Proyecto: Ordenamiento Territorial  
Escoga el Tipo de Proyecto

Descripción de Proyecto: Proyecto Prueba  
Ingrese una descripción del proyecto

Ubicación en Google Maps

Agregar un marcador

Nombre del marcador: Ingrese el nombre del marcador

Descripción:

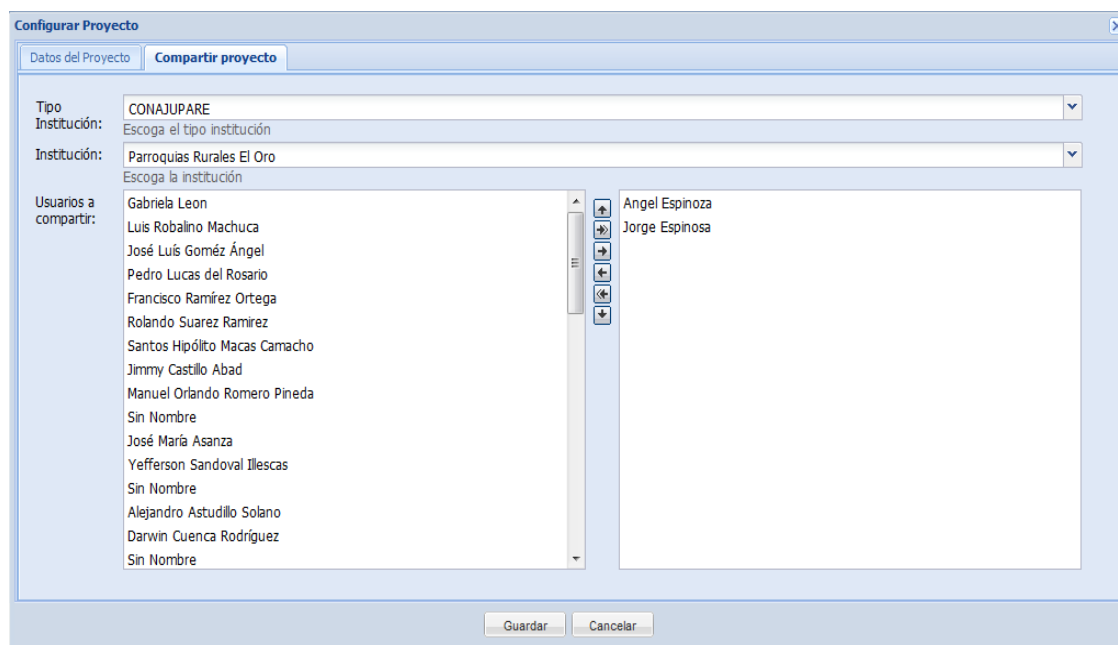
Map    Satellite

Guardar    Cancelar

Al pulsar el **botón configurar proyecto** se carga una nueva ventana, y de manera similar al registro de un nuevo proyecto aquí se cargaran los datos, así como también del mapa en caso



de existir. Lo nuevo que se suma a esta ventana es una nueva pestaña la cual permite al usuario compartir proyectos con otros usuarios de la misma o de distinta institución.

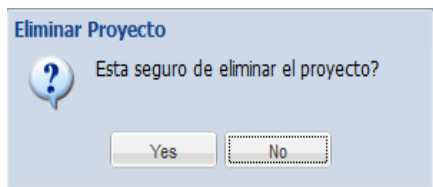


El proceso de compartir un proyecto es sencillo, gracias a que al momento de registrarse un usuario el sistema solicita que el usuario pertenezca a una institución, de acuerdo a ello, los proyectos se pueden compartir ya sea con usuarios de una misma institución o no, basta con seleccionar el tipo de institución y la institución a la cual pertenece el usuario que se le va a compartir el proyecto.

Vemos que se presenta una lista de los usuarios pertenecientes a determinada institución se selecciona los usuarios a compartir y se los arrastra a la segunda lista o por medio de los botones centrales entre las dos listas.

Una vez que se haya terminado la configuración del sistema se procede a pulsar el botón **Guardar**. De igual manera el registro de los datos del mapa o de compartir un proyecto son opcionales, lo que es necesario son los datos del proyecto.

## Desactivar y Eliminar un proyecto




El proceso de desactivación es similar a las anteriores secciones que se han visto se selecciona el proyecto a desactivar y se pulsa el botón de igual manera con el botón de eliminación, la diferencia entre ambas esta que la primera permite una eliminación temporal (se puede volver a activar) mientras que la segunda es permanente, por esa razón al momento de eliminar se le pregunta si está seguro si desea eliminar el proyecto, si está de acuerdo a todo se pulsa el botón SI y el proyecto será eliminado.

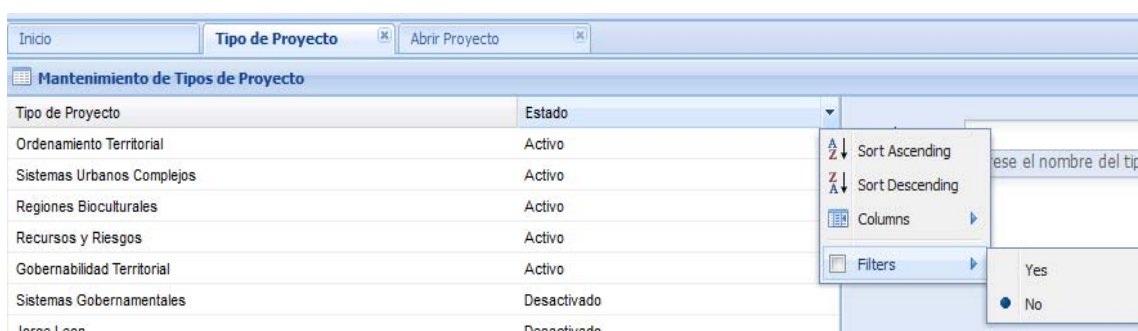
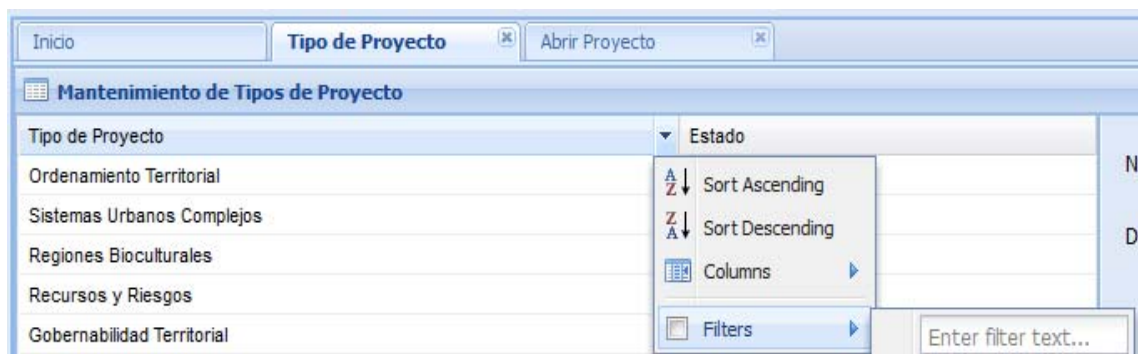
## Búsqueda

La búsqueda de un proyecto o tipo de proyecto se lo puede realizar dentro de su respectiva rama, es decir si el usuario desea buscar un tipo de proyecto lo puede realizar desde su módulo en el panel de navegador, de igual manera para un proyecto ya sea en **abrir un proyecto** o **configurar un proyecto**.

La búsqueda para un tipo de proyecto se la efectúa de la siguiente manera:

1. Por ejemplo, dentro de la pestaña tipo de proyecto, si el usuario desea buscar un tipo de proyecto de acuerdo al nombre se dirige a la columna de Tipo de Proyecto.
2. Da clic en la flecha  se desplaza un menú en donde el usuario puede realizar la búsqueda, se coloca sobre la opción Filters e ingresa el nombre del tipo de proyecto a buscar.

Se puede realizar la búsqueda por tipo de proyecto o por estado, como se puede observar en las siguientes imágenes.



Para el caso de buscar un proyecto se sigue el mismo principio se dirige a la columna por la cual se desea filtrar en este caso el proyecto se puede filtrar de acuerdo a nombre, el tipo del proyecto, fecha de creación, fecha de la última actualización, el nombre del creador del proyecto o de acuerdo a la institución a la que pertenece el usuario.

## Administración de Instituciones

### Tipos de Institución

Esta pestaña permite al administrador el registro, modificación y desactivación de un tipo de institución.

La estructura de la pantalla es la misma del tipo de proyecto por lo que el usuario no puede perderse. En la imagen inferior se puede ver su estructura especificado con una etiqueta que indica la tarea que hace cada componente.

Tipo de Institución	Descripción	Estado
CONAJUPARE	Asociación de juntas provinciales	Activo
AME	Asociación de juntas cantonales	Activo
CONCOPE	Asociación de juntas parroquiales	Activo
PRIVADA	En caso de tratarse de una empresa privada se selecciona e	Activo
OTRA	Cualquier otro tipo no incluido	Activo

Nombre:

Descripción:

Nuevo Desactivar Guardar

Page 1 of 1 Displaying 1 - 5 of 5

#### Barra de paginación

**Botón Nuevo:** Registro de un nuevo tipo de institución

**Botón Desactivar:** Desactivación de un tipo seleccionado

**Botón Guardar:** Es usado ya sea al momento de registrar un nuevo o realizar los cambios efectuados a un determinado tipo.

La **búsqueda** es la misma que para el módulo de proyectos, se selecciona la columna por la cual se desea filtrar en este caso por tipo y estado y se procede al filtrado.

## Mantenimiento de Instituciones

Siguiendo el mismo esquema se tiene el listado de las instituciones con las distintas opciones de filtrado de acuerdo a la columna seleccionada y el área de datos, junto con la barra de paginación y los botones de Nuevo, Desactivar y Guardar

Institución	Ciudad	Dirección	Tipo de Institución	Estado
Municipio de Loja	Loja	Bolivar entre Jose Antonio y Colon	CONAJUPARE	Activo
Parroquias Rurales El Oro	Machala	Av. Machala	CONAJUPARE	Activo
Junta Cantonal Ibarra	Ibarra	Av. Ibarra	AME	Activo
Geoinformatica Consultores	Cuenca	Pajan y Rio Malacatus	PRIVADA	Activo

Formulario de datos:

Tipo Institución: Seleccione un tipo...

Nombre: Ingrese el nombre de la institución

Ciudad: Ingrese la ciudad de la institución

Dirección: Ingrese la dirección de la institución

Teléfono: Ingrese el número telefónico de la institución

E-mail: Ingrese el correo electrónico de la institución

Botones: Nuevo, Desactivar, Guardar

Paginación: Page 1 of 1

Displaying 1 - 4 of 4

Cada componente cumple el mismo objetivo, la diferencia está en el número de columnas que se visualizan en la tabla, lo cual indica que el filtrado se puede realizar ya sea por:

- Institución
- Ciudad
- Tipo de Institución
- Estado

No se considera a la columna dirección como una columna para el filtrado de instituciones.

## Administración de Usuarios

El módulo de administración de usuarios se basa en una sola rama que es el mantenimiento de usuarios en esta ventana el administrador dispone de una lista de todos los usuarios que conforman Ge@ y se puede realizar actividades como

- Ingreso de nuevos usuarios
- Modificación de los datos de un usuario
- Activación/desactivación de la cuenta de un usuario
- Roles de usuario.

Cédula	Nombre	Apellido	Institucion	Estado
0104413761	Marco	Leon	Municipio de Loja	Activo
0104413762	Gabriela	Leon	Parroquias Rurales El Oro	Activo
1000000001	Luis Robalín	Machuca	Parroquias Rurales El Oro	Activo
1000000010	José Luis	Gómez Ángel	Parroquias Rurales El Oro	Activo
1100000011	Pedro Lucas	del Rosario	Parroquias Rurales El Oro	Activo
1200000012	Francisco	Ramírez Ortega	Parroquias Rurales El Oro	Activo
1300000013	Rolando	Suarez Ramirez	Parroquias Rurales El Oro	Activo
1400000014	Santos Hipó	Macas Camacho	Parroquias Rurales El Oro	Activo
1500000015	Jimmy	Castillo Abad	Parroquias Rurales El Oro	Activo
1600000016	Manuel Oria	Romero Pineda	Parroquias Rurales El Oro	Activo
1700000017	Sin	Nombre	Parroquias Rurales El Oro	Activo
1800000018	José María	Asanza	Parroquias Rurales El Oro	Activo
1900000019	Yefferson	Sandoval Illescas	Parroquias Rurales El Oro	Activo
2000000002	Sin	Nombre	Parroquias Rurales El Oro	Activo
2000000020	Alejandro	Astudillo Solano	Parroquias Rurales El Oro	Activo
2100000021	Darwin	Cuenca Rodríguez	Parroquias Rurales El Oro	Activo
2200000022	Sin	Nombre	Parroquias Rurales El Oro	Activo
2300000023	Mauricio	Armijos Armijos	Parroquias Rurales El Oro	Activo
2400000024	Sin	Nombre	Parroquias Rurales El Oro	Activo
2500000025	Gilmer	Ordoñez Murillo	Parroquias Rurales El Oro	Activo
2600000026	José Fernar	Rivas Romero	Parroquias Rurales El Oro	Activo
2700000027	Carmen Jan	Loayza	Parroquias Rurales El Oro	Activo

Área de datos

Roles de Usuario

Barra de

Botones Nuevo,  
Guardar Desactivar

Como se puede ver en la imagen superior el filtrado es únicamente por Nombre, Apellido y Estado del usuario.

Si un usuario desactivo su cuenta por error desde esta pestaña el administrador puede reactivarlo nuevamente.

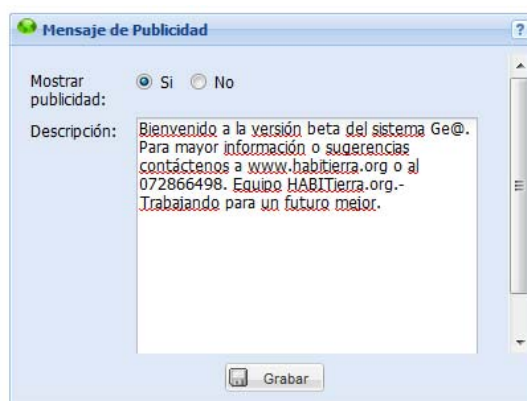
Si el administrador desea modificar el rol de algún usuario desde el campo roles de usuario lo puede realizar, no es necesario que tenga que modificar todos los datos del usuario, solo el rol únicamente. Así como también su nombre de usuario y contraseña.

La mayoría de módulos se basan en la misma estructura de una lista de ítems junto con el área de datos y los tres botones Nuevo, Modificar y Guardar, en algunos casos se incluyen otros botones como es el caso de Proyectos, pero en general, el manejo de los módulos es el mismo.

## Administración del Sistema

### Mensaje Inferior

Como se indicó al inicio del manual el sistema web dispone de una barra inferior que muestra un mensaje, pues desde este módulo el administrador puede modificar el mensaje y mostrar información de interés a la mayoría de usuarios de Ge@.



En esta ventana el administrador puede realizar dos tareas:

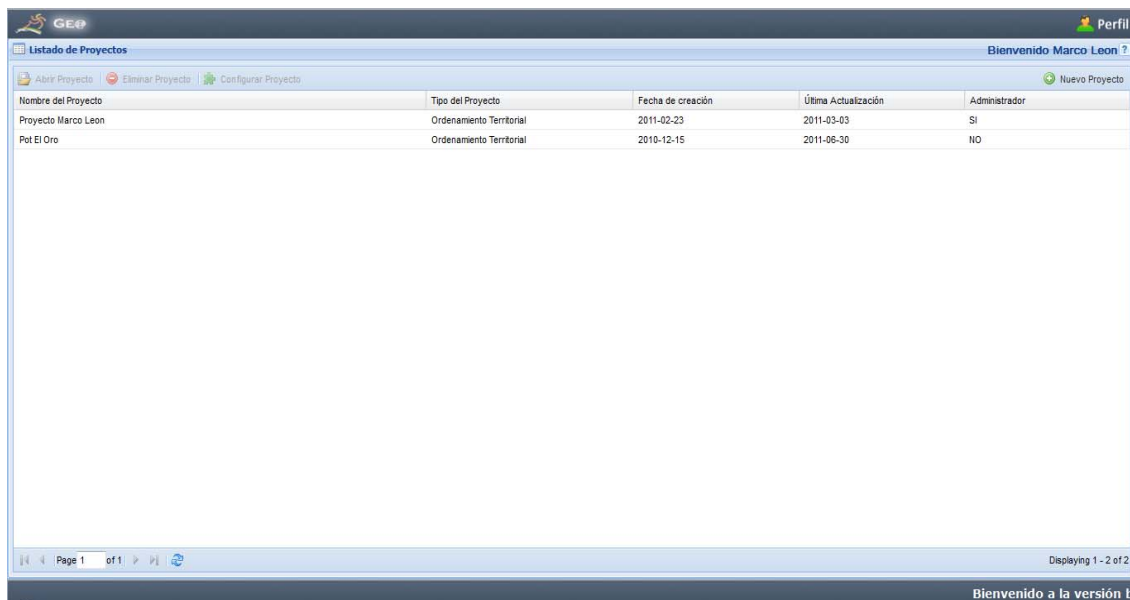
- Desactivar el mensaje inferior.
- Modificar el mensaje.

En cualquiera de los dos casos, el usuario que se encuentra ya dentro del Sistema Ge@ verá reflejada esta acción cuando ingrese nuevamente.

El sistema está adaptado para seguir añadiendo nuevos módulos de administración o añadir nuevas funcionalidades a los módulos existentes.

## INVITADO

La pantalla del usuario invitado varía bastante con la del usuario administrador como se puede observar en la imagen:



En este caso estamos realizando un ejemplo con el usuario Marco León, vemos que la pantalla es diferente a la de un administrador, el usuario invitado únicamente puede observar los proyectos que haya creado o que le hayan sido compartidos.

### Componentes de la Pantalla Inicial del Usuario Invitado

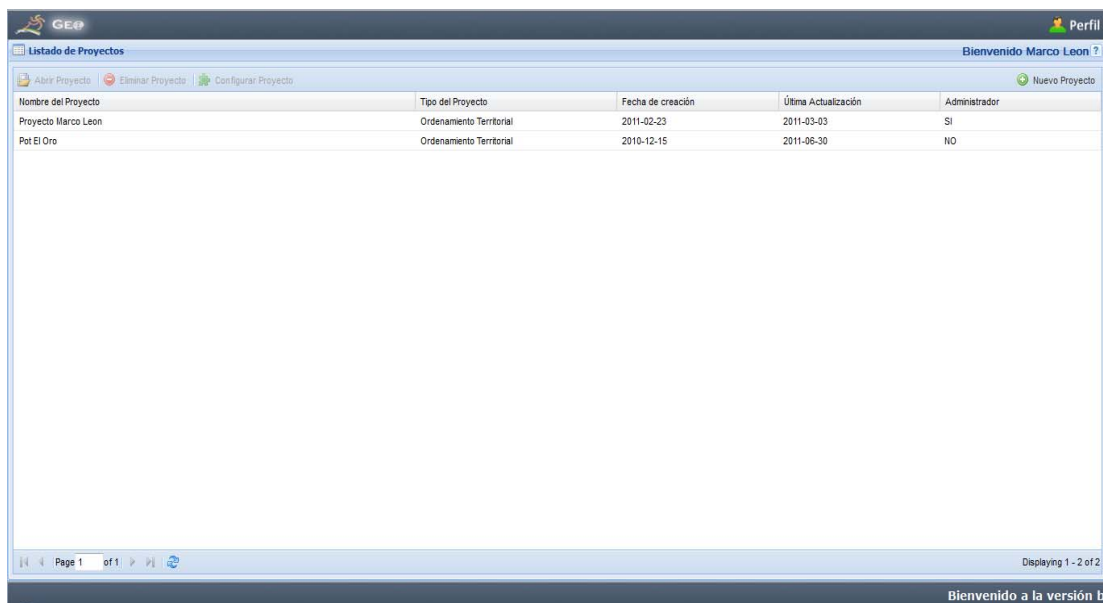
La pantalla se estructura en 3 áreas:

1. Barra de Título
2. Listado de Proyectos
3. Barra de Mensaje Publicitario

Tanto el área 1 y 3 se repiten en el administrador. La primera muestra el título del sitio y el botón para poder acceder al perfil de usuario, y la última presenta el mensaje publicitario que puede ser configurado en la pantalla de administrador en el módulo de sistema.

Los componentes que conforman la pantalla del usuario invitado son los que se pueden observar en la imagen, están listados y detallados en la parte inferior.





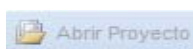
The screenshot shows a web application interface for managing projects. At the top, there is a header with the 'Ge@' logo and a user profile section for 'Marco Leon'. Below the header, there are navigation buttons: 'Abrir Proyecto', 'Eliminar Proyecto', 'Configurar Proyecto', and 'Nuevo Proyecto'. The main content area displays a table with the following data:

Nombre del Proyecto	Tipo del Proyecto	Fecha de creación	Última Actualización	Administrador
Proyecto Marco Leon	Ordenamiento Territorial	2011-02-23	2011-03-03	SI
Pot El Oro	Ordenamiento Territorial	2010-12-15	2011-06-30	NO

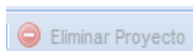
At the bottom of the interface, there is a pagination control showing 'Page 1 of 1' and a status bar indicating 'Displaying 1 - 2 of 2'.

### Botones de Configuración

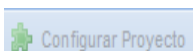
Los botones de configuración son los siguientes:



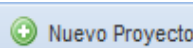
Al seleccionar un proyecto de la lista este botón se activa y permite ingresar al sistema Ge@.



Este botón se encarga de eliminar un proyecto que sea de la autoría del usuario invitado, en caso de ser un proyecto compartido por otro usuario este botón estará desactivado.



Al pulsar este botón el usuario puede editar los datos del proyecto que haya seleccionado, además puede compartir el proyecto con otros usuarios. De igual manera si el proyecto no es autoría del usuario este botón permanecerá desactivado.



Este botón es usado para la creación de un nuevo proyecto, al crear un proyecto este vendría a ser de su autoría y puede llevar a cabo la eliminación y configuración del mismo.

## Listado de Proyectos

El listado de proyectos se basa en una tabla conformada por 5 columnas:

- 1. Nombre del Proyecto:** Se muestra el nombre que se le haya dado al proyecto al momento de crearlo, este nombre puede ser modificado al pulsar el botón Configurar Proyecto siempre y cuando sea propietario del proyecto
- 2. Tipo del Proyecto:** Como se habló con anterioridad actualmente el tipo de sistema con el que se trabaja es Ordenamiento Territorial.
- 3. Fecha de Creación:** Como su nombre lo indica, permite visualizar la fecha en que fue creado el proyecto.
- 4. Fecha de última actualización:** Indica la fecha en que se realizó la última modificación al proyecto.
- 5. Administrador:** Esta columna indica si el proyecto es propiedad del usuario invitado. En este caso el usuario Marco León dispone de dos proyectos uno creado por él mismo con el nombre de “*Proyecto Marco León*” y en la columna Administrador saldrá **SI**, el siguiente proyecto se denomina “*POT El Oro*” y como se ve en la columna aparece escrito **NO**, lo cual indica que ese proyecto ha sido compartido por otro usuario.

Nombre del Proyecto	Tipo del Proyecto	Fecha de creación	Última Actualización	Administrador
Proyecto Marco Leon	Ordenamiento Territorial	2011-02-23	2011-03-03	SI
Pot El Oro	Ordenamiento Territorial	2010-12-15	2011-06-30	NO

## Barra de Paginación

La tabla puede mostrar 50 proyectos, en caso de que un usuario sobrepase esa cantidad el listado continúa en una siguiente página. Lo que permite la barra de paginación es el paso de una página a otra, además de disponer un botón de refresco en caso de que la tabla se congele o en caso de que el usuario haya modificado o creado un proyecto y los cambios no se vean reflejados en la lista.



Muestra el número de la página actual, además de contener los botones de ir al inicio, atrás, siguiente, ir al último

Botón de refresco. Refresca la

Etiqueta que muestra la página actual.

## Operaciones del Usuario Invitado

Las operaciones que puede realizar un usuario invitado están marcadas por los botones de configuración del proyecto. Las acciones que se ejecuta al pulsar cada botón son especificadas a continuación:

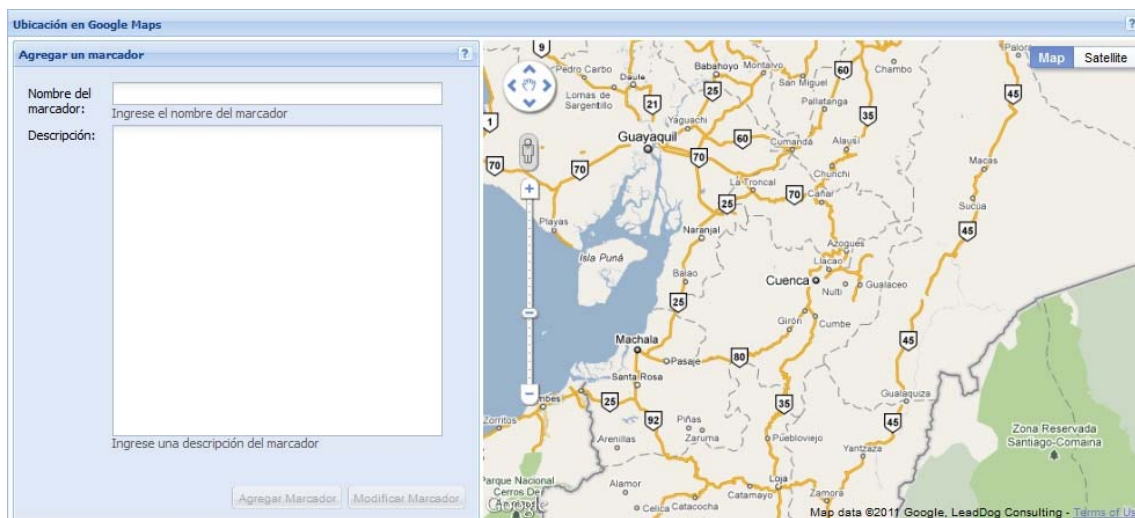
### Creación de nuevo proyecto

### Datos del proyecto

Los datos del proyecto son campos que el usuario obligatoriamente debe de llenarlos para poder crear el proyecto, los campos a llenar son:

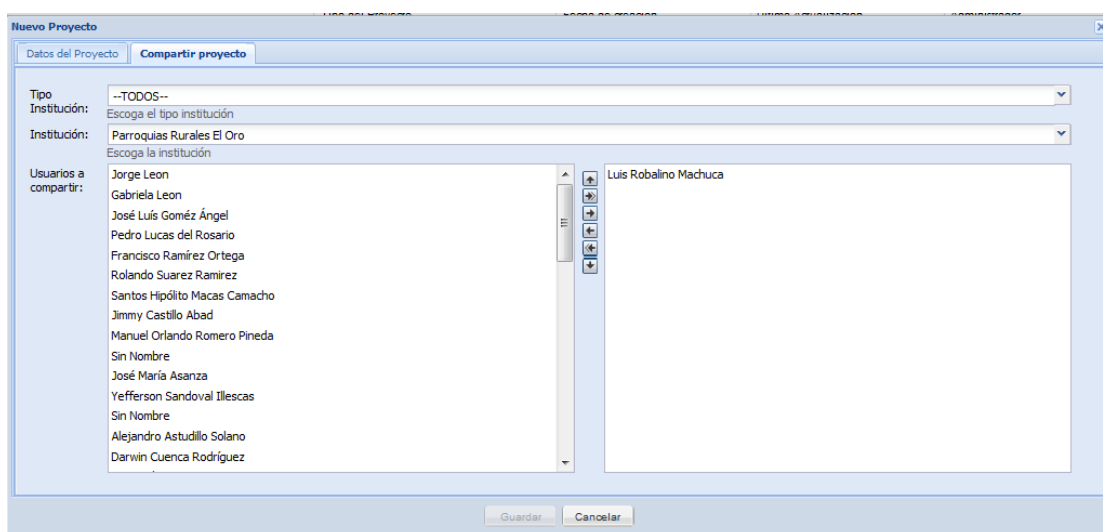
- a. **Nombre del Proyecto**
- b. **Tipo de Proyecto:** son 5 tipos: 1) Ordenamiento Territorial, 2) Sistemas Urbanos Complejos, 3) Regiones Bioculturales, 4) Recursos y Riesgos y 5) Gobernabilidad Territorial. Los tipos que se encuentran implementados son 1, 2 y 5, el resto se encuentran en proceso de desarrollo.
- c. **Descripción del Proyecto.**

## Ubicación en Google Maps



En la parte inferior de la ventana de creación de nuevo proyecto existe la posibilidad de colocar uno o varios puntos en el mapa de Google, con la finalidad de poder indicar la zona en donde se está llevando a cabo el proyecto, para dar una idea general del lugar de desarrollo y así el administrador pueda dar una asesoría adecuada del mismo. La colocación del punto en el mapa no es obligatoria. Para **eliminar un marcador** solo se da clic derecho sobre el marcador a eliminar.

## Compartir Proyectos

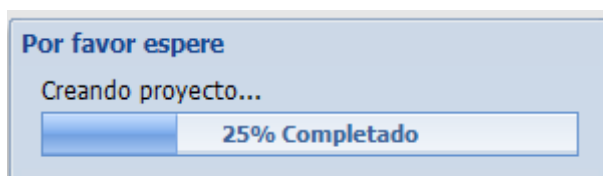


El proceso de compartir un proyecto es sencillo, gracias a que al momento de registrarse un usuario, el sistema solicita que el usuario pertenezca a una institución, de acuerdo a ello, los proyectos se pueden compartir ya sea con usuarios de una misma institución o no, basta con seleccionar el tipo de institución y la institución a la cual pertenece el usuario que se le va a compartir el proyecto.

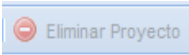
Vemos que se presenta una lista de los usuarios pertenecientes a determinada institución se selecciona los usuarios a compartir y se los arrastra a la segunda lista o por medio de los botones centrales entre las dos listas.

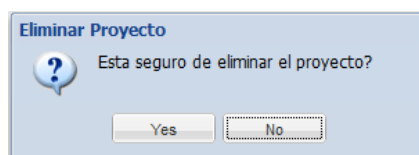
Una vez que se haya terminado el ingreso de los datos del proyecto se procede a pulsar el botón **Guardar**.

Cuando haya creado el proyecto se presentara una barra de carga en la cual se le indica que se está configurando el proyecto para ser usado posteriormente.



### Eliminación de un Proyecto

El proceso de eliminación es sencillo, de la lista de proyectos se selecciona el ítem a ser eliminado y se pulsa el botón , el proceso de eliminación requiere que el usuario este seguro de la operación que va a realizar, por seguridad se le pregunta si la tarea que mando a ejecutar es la correcta.



### Configurar Proyecto

La ventana de configuración es la misma que se presentó para la creación de un nuevo proyecto con la diferencia que aquí se cargan los datos que hayan sido establecidos anteriormente al momento de su creación.

El usuario únicamente lo que debe realizar es la modificación de los campos que realmente necesite cambiar como los datos del proyecto, la ubicación del o los puntos en el mapa y a aquellos usuarios a los que se les haya compartido.

**Configurar Proyecto**

Datos del Proyecto    Compartir proyecto

**Datos del Proyecto**

Nombre de Proyecto: Proyecto Marco Leon  
Ingrese el nombre del proyecto

Tipo de Proyecto: Ordenamiento Territorial  
Escoga el Tipo de Proyecto


Descripción de Proyecto: Proyecto de Marco Leon  
Ingrese una descripción del proyecto

**Ubicación en Google Maps**

Agregar un marcador

Nombre del marcador: Oficina Geo  
Ingrese el nombre del marcador

Descripción: oficina



Guardar    Cancelar

**Configurar Proyecto**

Datos del Proyecto    **Compartir proyecto**

Tipo: CONAJUPARE  
Institución: Escoga el tipo institución

Institución: Municipio de Loja  
Escoga la institución

Usuarios a compartir:

Jorge Espinosa

Guardar    Cancelar

## SISTEMA GE@

### Escritorio Ge@



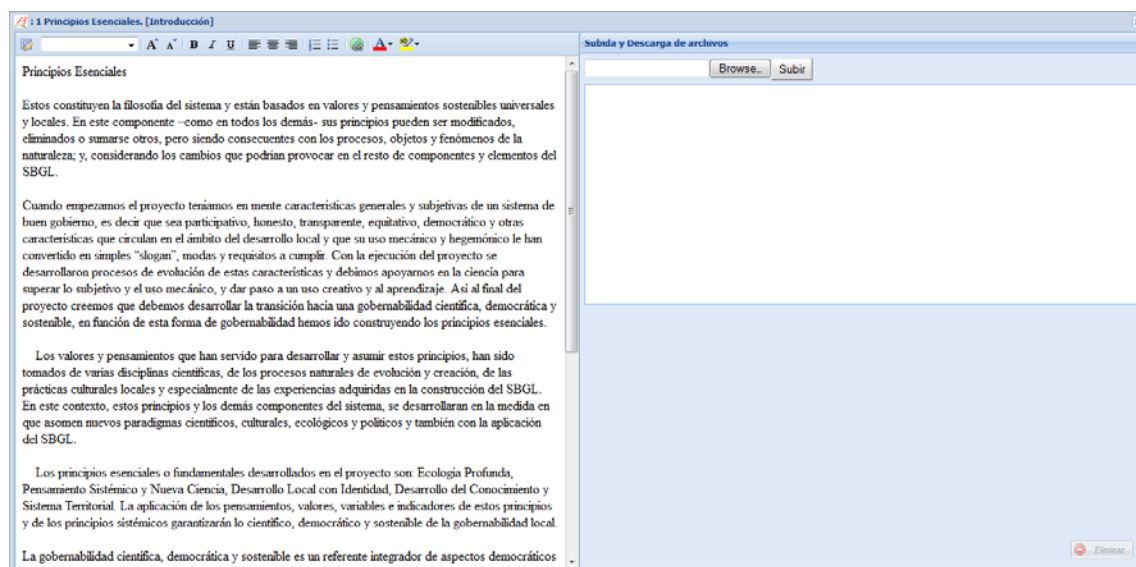
Haciendo analogía al escritorio de Windows, tiene la misma estructura pero en sí no es exactamente un escritorio, ya que un escritorio tiene más funcionalidades que el Escritorio de Ge@, este se estructura con la finalidad de facilitarle al usuario una mejor navegación entre los distintos componentes de Ge@, basándose en un sistema de ventanas.

Por cada nueva ventana abierta se carga un nuevo ítem en la barra inferior y el usuario puede cambiar entre ventanas al solo dar clic el ítem que desea.

### Menú Principal



El menú principal está conformado por 9 componentes que se describen en la imagen. Cada uno de los componentes del sistema tiene una introducción para que pueda tener conocimiento de lo que trata. Cabe recalcar que la estructura de toda introducción de cada componente es similar a lo que se visualiza en el siguiente gráfico.



En la parte izquierda se tiene el contenido de dicha introducción, en la parte derecha se puede agregar cualquier tipo de archivo digital (fotos, videos, documentos de Word, documentos de Excel, etc.) que sirva de apoyo para sustentar el contenido de la parte izquierda.

Cada uno de los nueve componentes generales tiene varias interpretaciones y abre una ventana diferente, para ello hay que hacer clic sobre cada componente, e inmediatamente la acción se verá reflejada en el navegador, a continuación se detalla una tabla de las distintas opciones que se presentan por cada componente.

Componente	Opción	
Principios Esenciales	Introducción	Abrir
Principios Sistémicos	Introducción	Abrir
Interacciones	Introducción	Ejercicios
Modelización Digital	Introducción	Software
Base de Conocimiento Capitalizada	Introducción	Abrir
Momentos de Intervención	Introducción	Abrir
Prototipos	Introducción	Software
Caja de Herramientas	Introducción	Abrir
Variables e Indicadores	Introducción	Abrir



## Estructura del Software

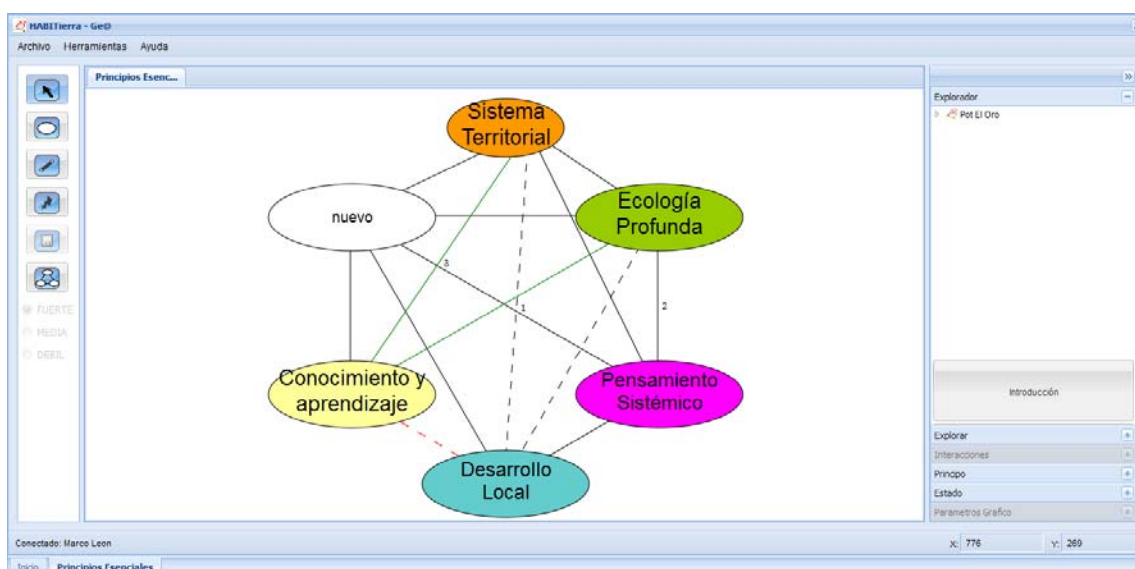
Barra de Título

Área de trabajo

Panel de  
Propiedades

Barra de Menús

Barra de  
Herramientas



Barra de Estado

Objeto

Interacción

### Barra de Herramientas

Las herramientas generales de principios esenciales, principios sistémicos, interacciones, Modelo Digital Ge@ y Prototipos, están en la parte izquierda de la pantalla. Son las siguientes:



Seleccionar Objeto



Nuevo Objeto



Nuevo Interacción



Mostrar/Ocultar propiedades



Guardar



Ordenar Objetos

④ FUERTE

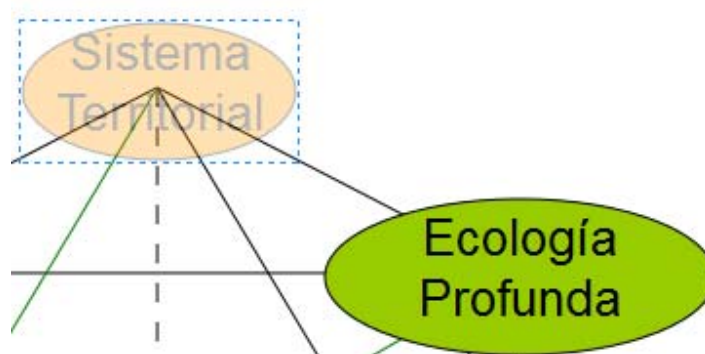
Tipo de interacción

○ MEDIA

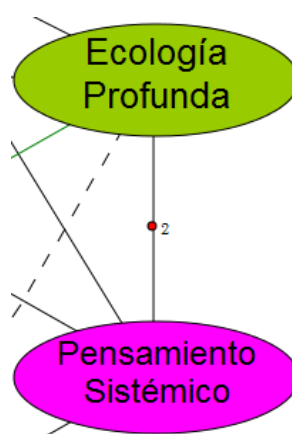
○ DEBIL

**Seleccionar Objeto.**-Sirve para seleccionar un objeto (óvalos que representan principios, dimensiones, elementos, etc.) o una interacción, al seleccionar un objeto se puede ver las opciones de este, cambiar el nombre, color, tipo de letra, tamaño, etc.

Para seleccionar un objeto hay que escoger este botón y con el botón derecho del ratón hacer clic sobre el objeto a señalar y desplazar el objeto, la imagen se tornara como se puede ver en la figura.



Al señalar una interacción se puede definir: ¿Cómo está? ¿Cómo debería ser?, si es una interacción clave, etc. Para señalar una interacción se hace clic sobre la interacción a seleccionar. La interacción señalada se muestra de la siguiente forma.

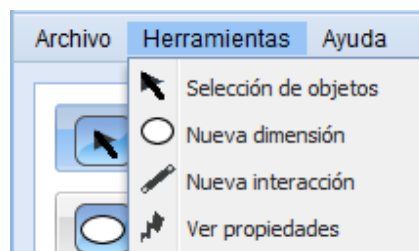


**Nuevo Objeto:**Permite agregar un nuevo objeto (óvalos que representan principios, dimensiones, elementos, etc.), Se lo puede realizar de dos maneras:

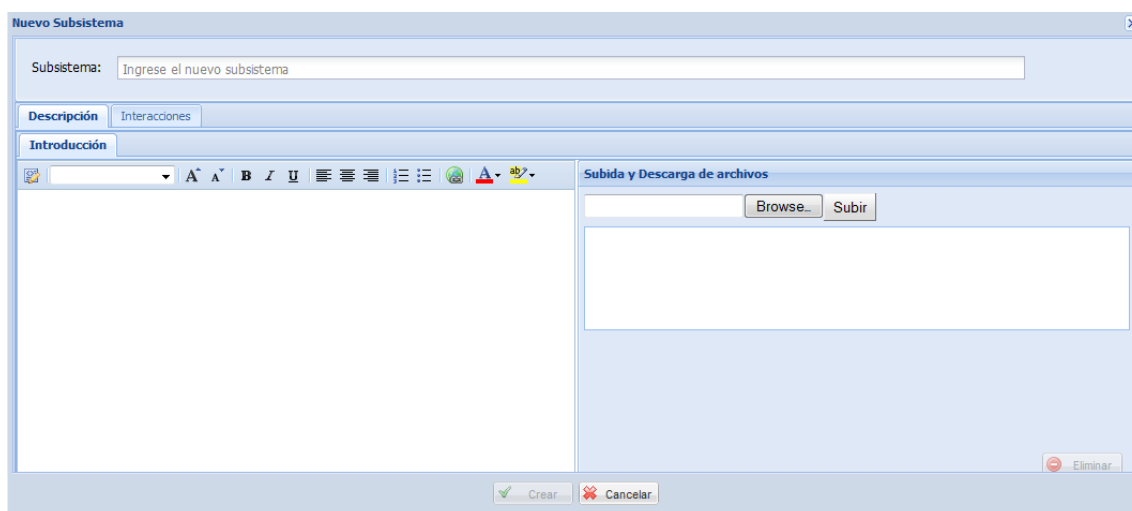
- Por medio del botón que se encuentra en la barra de herramientas



- Por medio de la barra de menús

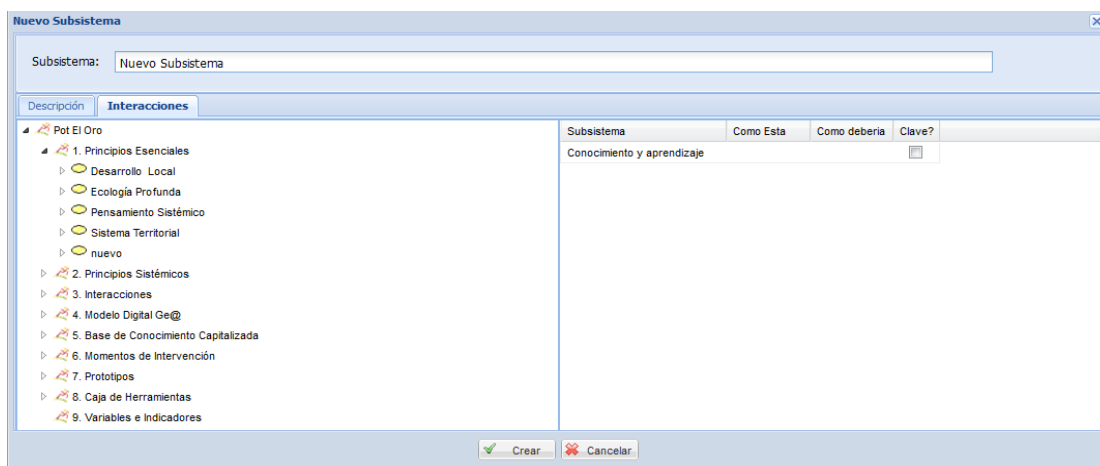


De cualquiera de las dos maneras se abrirá la siguiente ventana. Aquí el usuario puede agregar una introducción, adjuntar un archivo y generar interacciones con otros objetos.



La pestaña descripción varía de acuerdo al componente que haya seleccionado desde el menú inicial de Ge@, pero básicamente se basa en la misma estructura. En la imagen superior se visualiza la ventana para la creación de un nuevo subsistema dentro del componente Principios Esenciales, se puede observar que la pestaña descripción incluye una introducción en la parte derecha izquierda y en la parte derecha permite la subida de cualquier archivo que sirva de refuerzo (similar a la ventana introducción que se presentó en el menú Inicial).

Todas las ventanas de los distintos componentes incluye la pestaña Interacciones, esta pestaña permite relacionar el nuevo objeto creado con cualquier objeto presente en el sistema objeto, el proceso de interacción es similar al de compartir un proyecto



Se selecciona del árbol que se muestra en la parte izquierda y se le arrastra para el lado derecho, inmediatamente el dato se carga en la grilla, aquí el usuario puede añadir parámetros extras a la nueva interacción como son: **¿Cómo está?**, **¿Cómo debería?** Y si se trata de una interacción **Clave**.

**Nueva Interacción.** Permite crear una interacción entre dos objetos, al crear un nuevo objeto las interacciones se crean de acuerdo a lo que se haya seleccionado en la pestaña interacciones, pero si no se ha creado interacción alguna, lo creamos presionando el botón izquierdo del ratón sobre el primer objeto y luego damos clic en el segundo elemento, no es necesario tener presionado el mouse, durante este proceso de realizar una interacción se puede observar una línea que nace desde el objeto inicial. Hay que tener en cuenta que si ya existe la interacción esta no se creara nuevamente y se le indicará que ya existe.

**Mostrar/Ocultar Propiedades.** Muestra u oculta el panel de la derecha que contiene propiedades de un objeto o relación de cada componente y el explorador de Ge@.

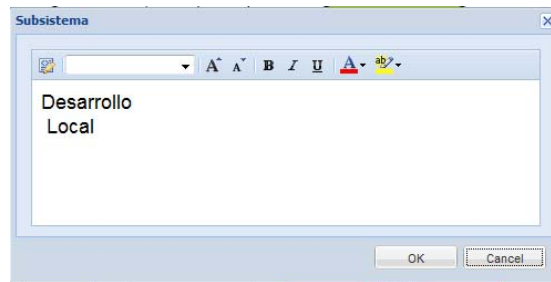
### Propiedades

Las propiedades se encuentran en la parte derecha de la pantalla. Estas son:

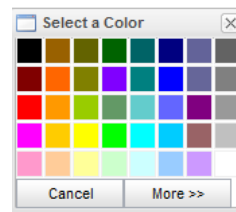
**Propiedades de un objeto:** nombre, alto, ancho, color y tipo de letra.



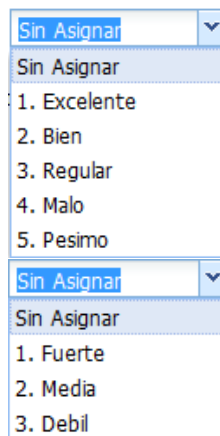
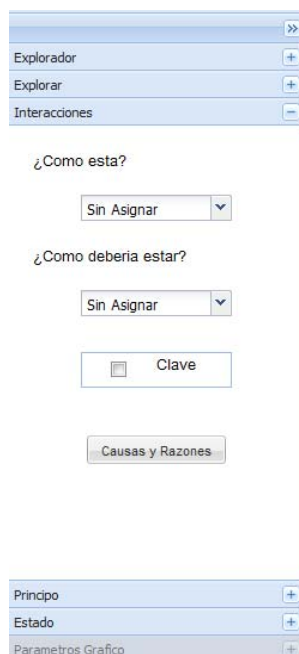
Al seleccionar un objeto su contenido interno se carga en este cuadro de texto, si se desea cambiar el formato del texto al dar clic sobre este recuadro se muestra la siguiente ventana



Para cambiar el fondo del objeto se da clic aquí.



**Propiedades de una interacción:** ¿Cómo está?, ¿Cómo debería ser?, si es interacción clave, causas y razones, tipo de letra.



Al pulsar el boton de Causas y Razones se visualiza la siguiente pantalla:


Una pre-visualización de ¿cómo está? Y ¿Cómo debería ser?


Muestra el nombre del sistema origen

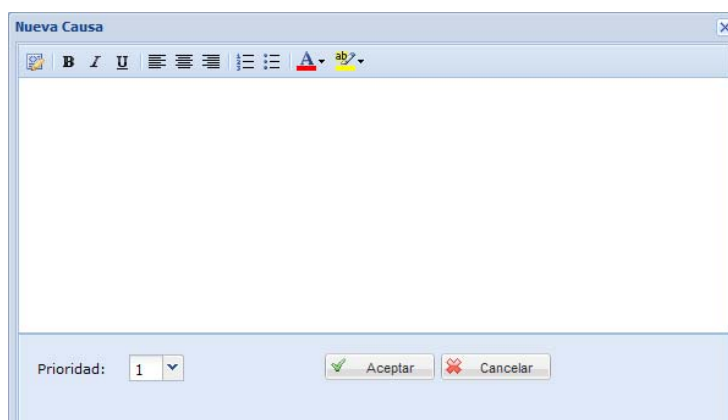
Muestra la lista de **Razones**

Muestra el nombre del sistema final

Muestra la lista de **Causas**

El botón  **Nueva Razón** permite la creacion de nuevas razones, al pulsar este botón aparece una nueva ventana en donde el usuario va a ingresar una breve descripcion de la razón y la prioridad de la misma, todo esto se visualiza en la siguiente figura.

El botón  **Nueva Causa** permite la creación de nuevas causas, al pulsar este botón aparece una nueva ventana en donde el usuario va a ingresar una breve descripcion de la causa y la prioridad de la misma, todo esto se visualiza en la siguiente figura.



**Estado del Componente o Sistema:** Muestra una leyenda de lo que significa el valor y color de la interacción así como también el estado (resultado del análisis sistémico) y el número de interacciones agrupadas por tipo.

En esta propiedad el usuario puede mandar a exportar el estado del sistema con el fin de tener un documento que sea imprimible y que valide el estado del mismo.

Todo esto se puede visualizar en el siguiente grupo de imagen.


Valor	Peso	Equivalencia
1	10	Excelente
2	7	Bien
3	5	Regular
4	3	Malo

Color	Peso	Equivalencia
Rojo	10	Fuerte
Verde	5	Media
Azul	3	Débil

Sistema	Total	Equivalencia
Sistema Territoi	25	Malo
Ecología Profur	0	Excelente
Desarrollo Loc	0	Excelente
Conocimiento y	25	Pésimo

En esta imagen el usuario puede observar la leyenda, esta leyenda puede ocultarse al pulsar el botón que se encuentra en la parte superior que tiene la forma  este permite expandir y ocultar la leyenda.

Sistema	Total	Equivalencia
Sistema Territor	25	Malo
Ecología Profur	0	Excelente
Desarrollo Loc	0	Excelente
Conocimiento y	25	Pésimo
Total	13	Pésimo
Relación	Número	Porcentaje
Sin asignar	11	78.57%
Fuerte	1	7.14%
Media	2	14.29%

En esta imagen el usuario puede observar el ¿Cómo está? Y el ¿Cómo debería ser? De todo el sistema en general. Esta imagen se basa en el componente de Principios Esenciales y como se observa en la imagen el sistema está pésimo, haciendo cuenta entre todos los subsistemas y tiene 1 interacción fuerte y dos medias, el resto no se ha asignado.

Para obtener un reporte de lo que se visualiza en las tablas, se debe seguir bajando en la misma ventana hasta llegar a un botón que ofrece al usuario la posibilidad de exportar estas tablas junto con un gráfico a los siguientes formatos PDF, XLS y DOCX para luego ser impresos en caso de desearlo.



El proceso de obtener el reporte toma su tiempo, ya que el sistema lo que hace es generar el reporte en el formato que solicitó y luego empaquetarlo para ser enviado desde el servidor hasta el usuario final.

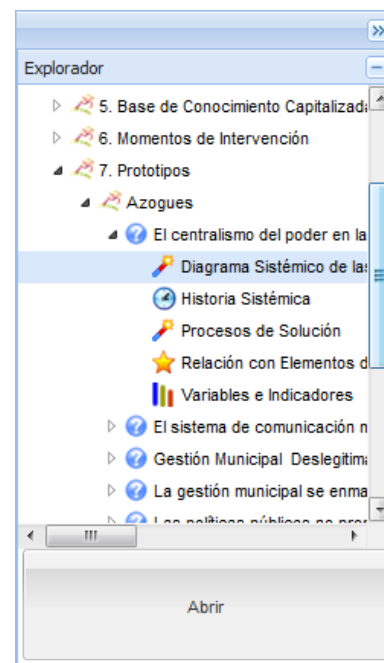
Una vez terminado todo el proceso se le presentará una ventana del navegador en el que esté trabajando el cual le indicará que usted esta descargando un archivo, se acepta esta ventana e inmediatamente el archivo se descargará en su máquina.



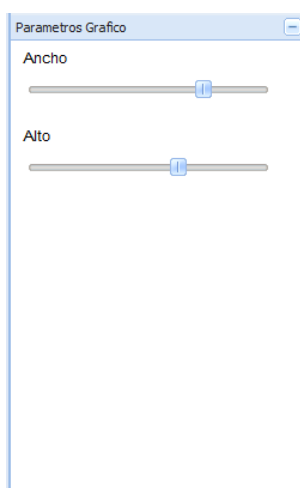



**Explorar:** Muestra en forma jerárquica los objetos que forma un componente del Sistema Ge@.

**Explorador:** Muestra en forma jerárquica (de árbol) todo el Sistema Ge@. Permite acceder directamente a cualquier parte de Ge@ desde este árbol, por ejemplo en la imagen inferior se puede observar el componente Prototipos junto con el botón Abrir que se encuentra en la parte inferior.



**Parámetros Gráfico:** La propiedad parámetros gráfico únicamente se activa cuando se entra al componente Prototipos dentro del diagrama sistémico de las variables de un problema cualquiera. Y permite modificar el ancho y alto de los diagramas.



Si se desea **imprimir** el área de trabajo se lo puede hacer desde el menu archivo la opción **Imprimir** .

## Estructura del Sistema Ge@

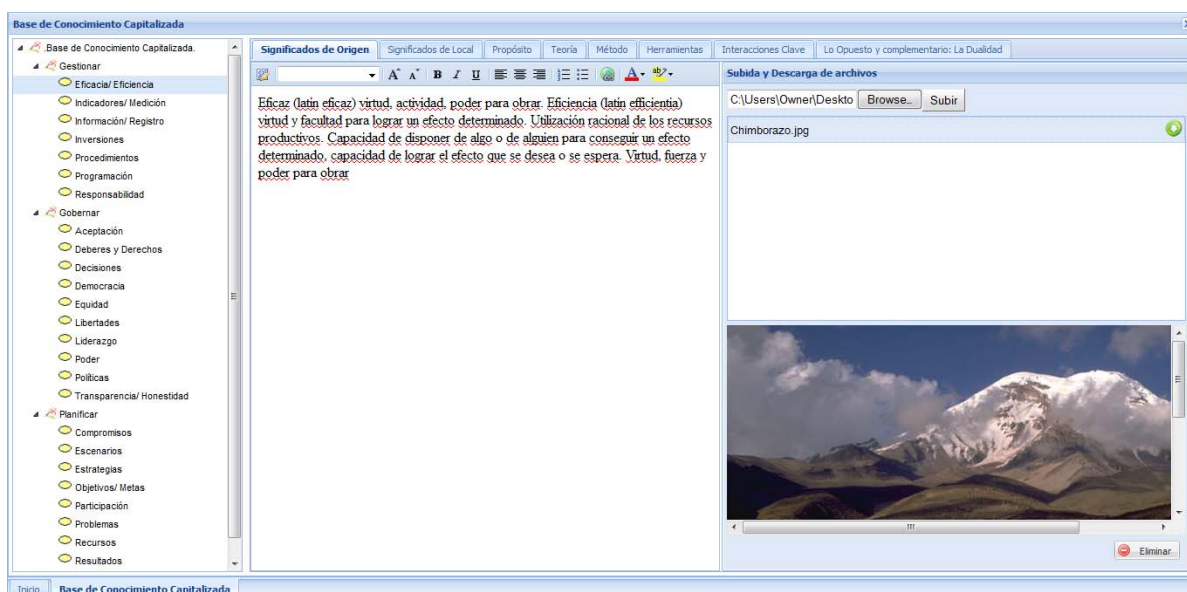
COMPONENTE				
Principios Esenciales	Introducción			
	Principio Esencial	Introducción		
		Variables e Indicadores		
		Teoría-Método-Herramienta		
Principio Sistémico	Introducción			
	Principio Sistémico	Introducción		
		Variables e Indicadores		
		Teoría-Método-Herramienta		
Interacciones	Introducción			
	Ejercicio	Situación		
		Dimensión/Elemento	Introducción	
			Variables e Indicadores	
		Base de Conocimiento (8 elementos)		
		Sumas		
Model Digital Ge@	Introducción			
	Modelo	Situación		
		Dimensión/Elemento	Introducción	
			Variables e Indicadores	
		Base de Conocimiento (8 elementos)		
		Sumas		
Base de Conocimiento Capitalizada	Introducción			
	Dimensión/Elemento	Base de Conocimiento (8 elementos)		
Momentos de Intervención	Introducción			
	Momento	Introducción		
		Teoría-Método-Herramienta		
Prototipos	Introducción			
	Prototipo	Problema	Historia Sistémica	
			Variable e Indicadores	
			Diagrama Sistémico del problema	
			Diagrama Sistémico de la solución posible	
			Relación con los Elementos de Ge@	
Procesos de Solución				
		Relacionarlo con el resto de Problemas		
Caja de Herramientas	Introducción			
	Herramienta	Introducción		
		Ítems (5)		
Variables e Indicadores	Introducción			
	Introducción, Variables, Indicadores de:	Dimensión/Elemento		
		Principios Esenciales		
		Principios Sistémicos		
		Problemas		

## Manejo de ítems y archivos digitales

Esta opción está disponible en la mayoría de componentes que conforman el sistema Ge@, el manejo de estas opciones es similar lo único que cambia son los diferentes ítems que tiene cada componente. (Los ítems son los que se visualizan en la imagen anterior).

Cada elemento del sistema puede tener ítems, estos ítems tienen un cuadro de descripción y un panel de subida y descarga de archivos que respaldan la descripción. A continuación se detallan las opciones que tiene la pantalla de gestión de ítems de los elementos del sistema.

<b>Componente del Sistema Ge@</b>	<b>Ítems del componente</b>	<b>Botón de Búsqueda de archivos a subir</b>	<b>Lista de Archivos Subidos</b>	<b>Botón de descarga</b>
-----------------------------------	-----------------------------	--	----------------------------------	--------------------------



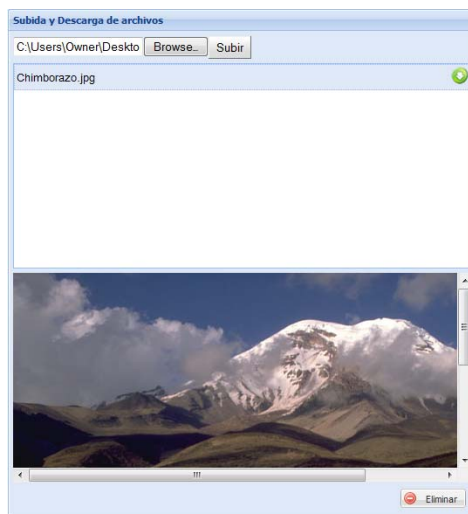
**Elementos del Componente**

**Área de descripción**

**Vista previa del archivo digital**

**Botón Eliminar Archivo Subido**


Cada componente dispone de un panel en el lado derecho junto a la descripción, el cual es usado para la subida y descarga de archivos y que permita respaldar la información descrita en el área de descripción.



El usuario puede subir el archivo al servidor y poder descargarlo desde cualquier máquina en caso que lo desee, como se explicó anteriormente cada ítem dispone de este panel.

Es necesario indicar que únicamente solo los archivos subidos que sean del tipo imagen tendrán una pre-visualización, caso contrario el usuario únicamente podrá descargar el archivo si desea visualizar su contenido.

La subida del archivo se realiza pulsando el botón de búsqueda de archivos, el usuario se le presentara una ventana en donde buscará el fichero que desea subir.

El proceso de descarga es simple, el usuario únicamente tiene que pulsar el siguiente icono  y seleccionar el lugar en donde sea guardar el archivo.

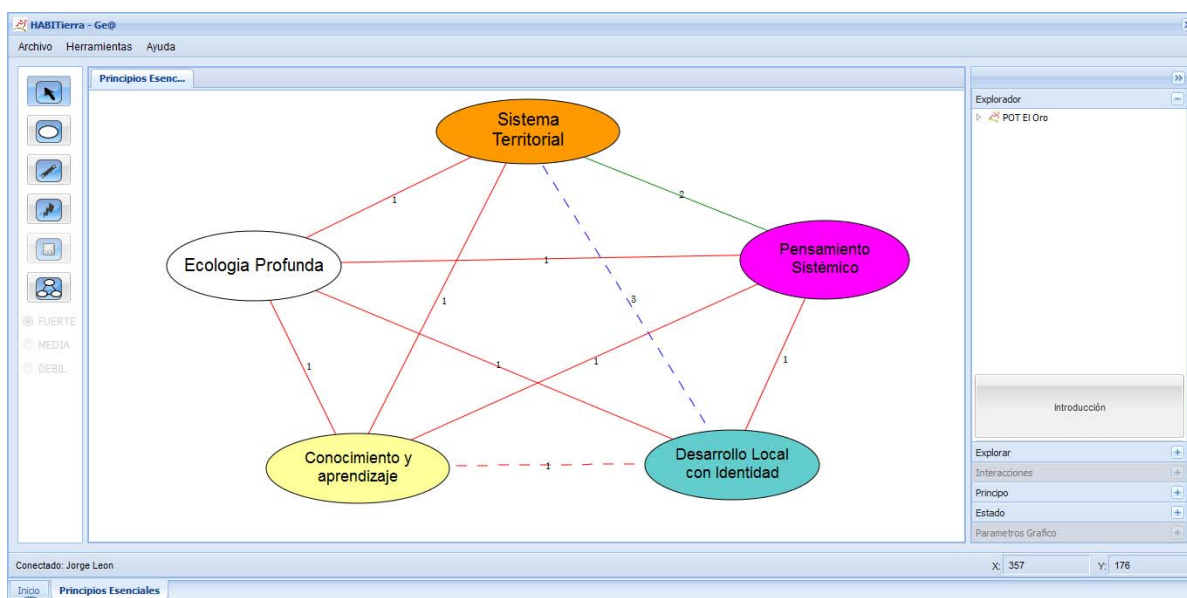
### **Componentes del Sistema Ge@**

Para entrar a cada uno de los componentes de Ge@, se lo puede realizar desde el menú principal de Ge@, que se describió en secciones anteriores. Se tiene nueve opciones 1) Principios Esenciales, 2) Principios Sistémicos, 3) Interacciones, 4) Modelización Digital, 5) Base de Conocimiento Capitalizada, 6) Momentos de Intervención. 7) Prototipos, 8) Caja de Herramientas y 9) Variables e Indicadores.


Cada una de estas opciones se describen en las siguientes secciones.

## 1. Principios Esenciales

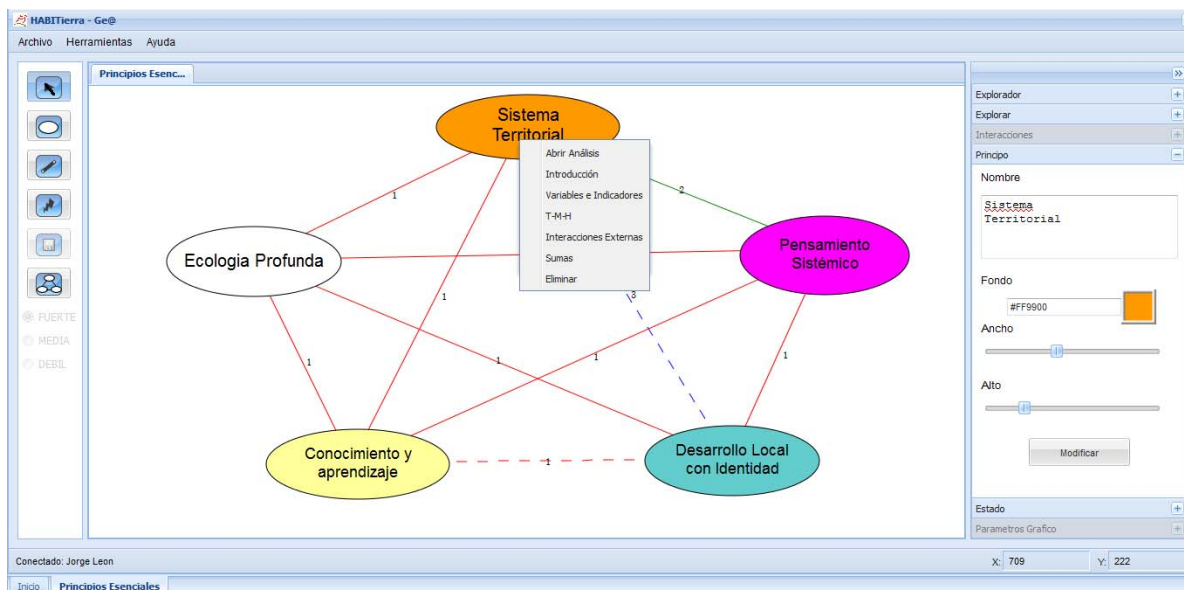
Al abrir se van a mostrar los principios esenciales del Sistema Ge@ de la siguiente manera:



### Opciones de cada Principio

Cada principio esencial tiene varias opciones. Para poder ver las opciones de cada principio hay que escoger la opción seleccionar  de la barra de herramientas de la izquierda; luego hay que señalar el principio (con el botón izquierdo del ratón hacer clic sobre el óvalo del principio), una vez señalado, hacer clic con el botón derecho del ratón y escoger la opción deseada así.

Las opciones disponibles para cada principio esencial son: Abrir Análisis, Introducción, Variables e Indicadores, Teoría-Método-Herramientas (T-M-H), Interacciones Externas, Sumas y Eliminar.

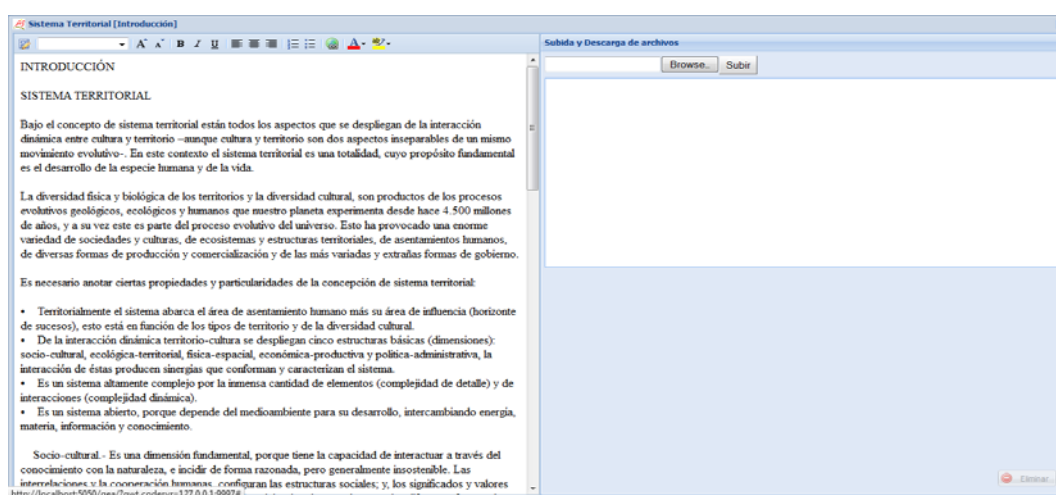


## 1.1 Abrir Análisis

Abre una nueva pestaña para indicar “*sub principios esenciales*” que puede contener el principio esencial escogido. Las opciones de un sub principio son iguales a las de un principio.

## 1.2 Introducción

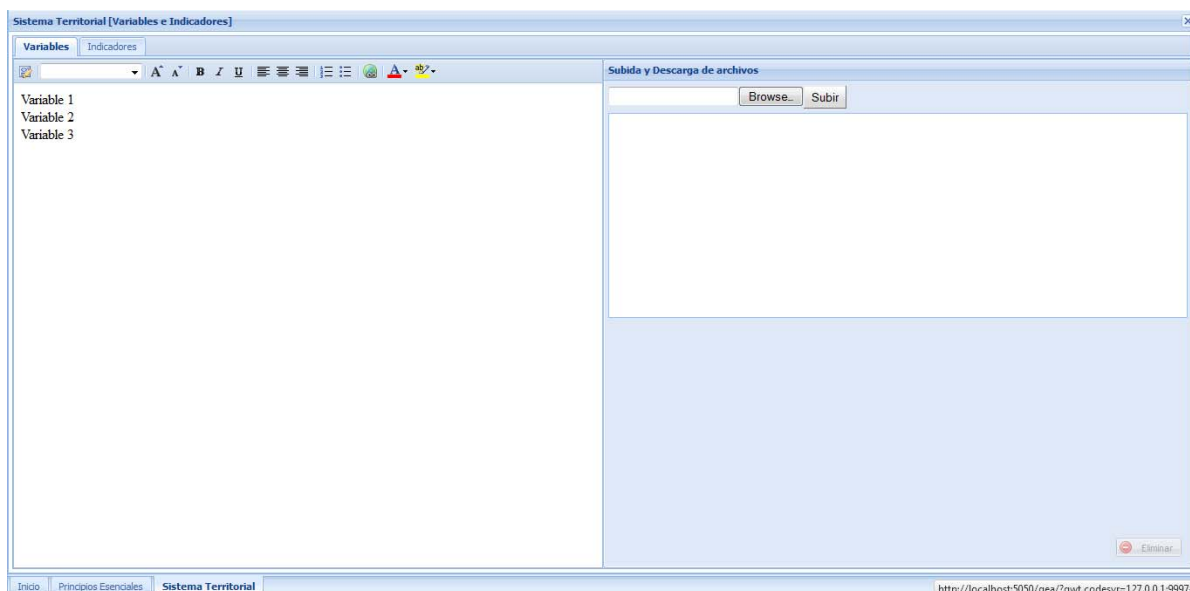
Al escoger la opción Introducción aparecerá la siguiente ventana con una introducción del principio.



Para aprender más sobre cómo manejar las opciones de esta ventana vea el punto Manejo de ítems y archivos digitales.

### 1.3 Variables e Indicadores

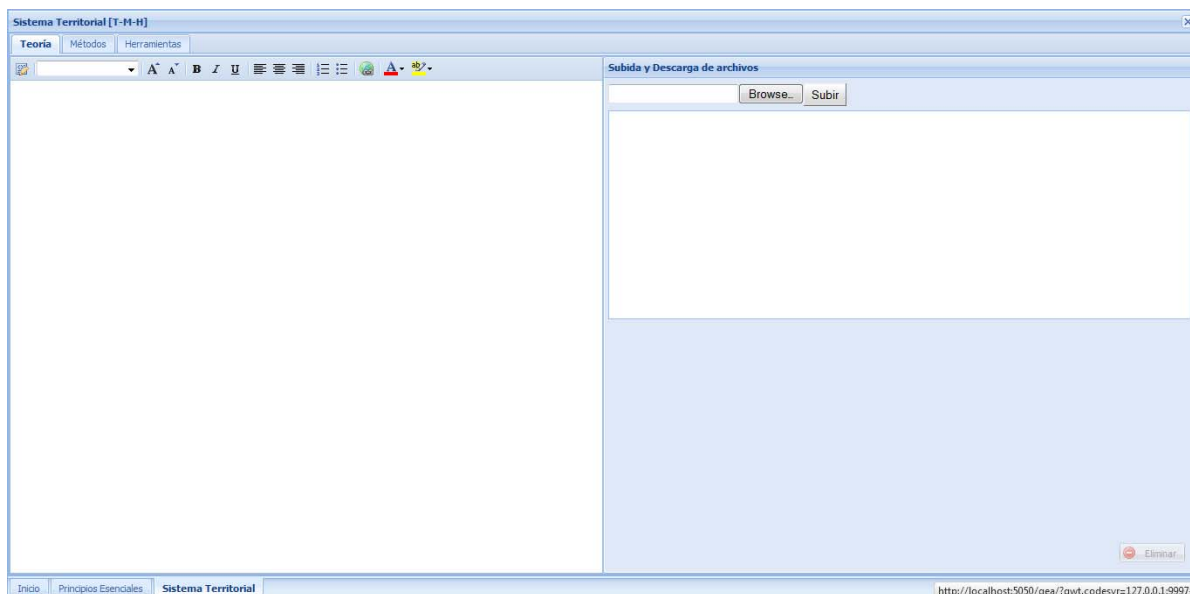
Al escoger la opción Variables e Indicadores aparecerá la siguiente ventana con variables e indicadores sobre el principio esencial seleccionado.



En la parte izquierda se tienen las variables o indicadores y en la parte derecha archivos digitales que ayuden a sustentar estas variables o indicadores. Para ver los indicadores hay que hacer clic sobre la pestaña *Indicadores*. Para ocultar la ventana y grabar los cambios realizados hacer clic en la X en la parte superior, si es que se realizó un cambio aparecerá una ventana que le indicara si desea grabar los cambios, en caso positivo se guarda los cambio y sale, en caso contrario no los graba y sale, y una tercera opción que cancela la salida y continua en la ventana.

### 1.4 Teoría-Método-Herramientas (T-M-H)

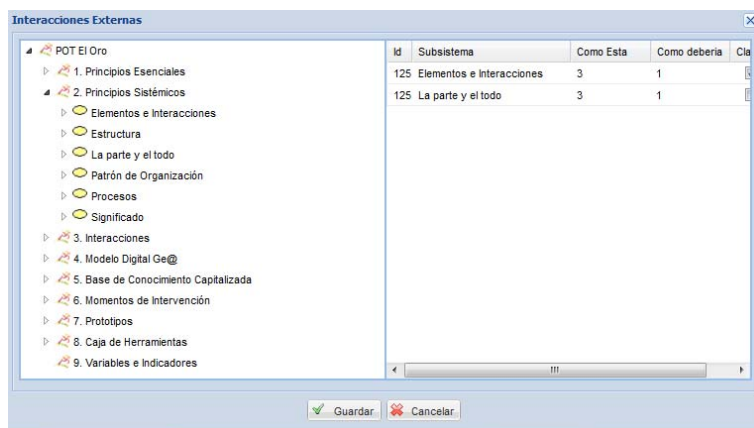
Al escoger la opción Teoría-Método-Herramientas aparecerá la siguiente ventana con la teoría, método y herramienta sobre el principio esencial seleccionado.



En la parte izquierda se tiene la descripción de las teorías, métodos y herramientas y en la parte derecha archivos digitales que ayuden a sustentarlas. Para ver los métodos o herramientas hay que hacer clic sobre las demás pestañas. Para ocultar la ventana y grabar los cambios realizados hacer clic en la X en la parte superior, si es que se realizó un cambio aparecerá una ventana que le indicara si desea grabar los cambios, en caso positivo se guarda los cambio y sale, en caso contrario no los graba y sale, y una tercera opción que cancela la salida y continua en la ventana.

## 1.5 Interacciones Externas

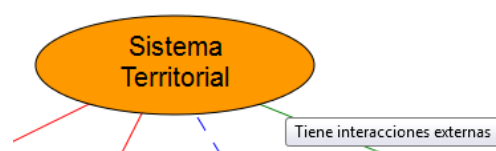
La opción de interacciones externas permite al usuario visualizar a aquellas relaciones que no se pueden observar en la pantalla, aquellas que se han realizado con objetos que no se encuentran en el mismo nivel del sistema.





Esta ventana está conformado por dos áreas la primera es un árbol en donde se puede observar los objetos que se han creado en el Sistema Ge@ y a partir de aquí se seleccionará el que se desee relacionar con el principio. El proceso es simple, se selecciona y arrastra a la tabla. Aquí también se puede visualizar las relaciones externas que se hayan efectuado con anterioridad.

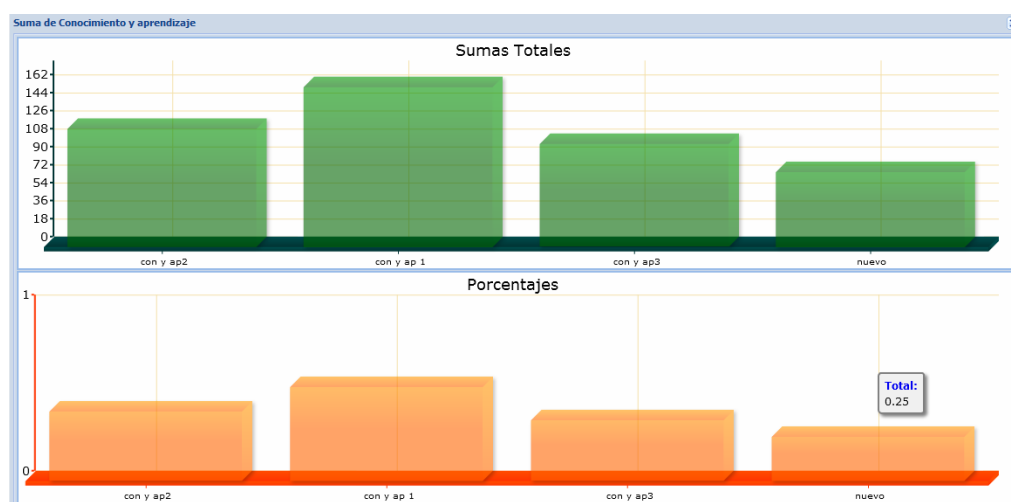
Si un principio contiene interacciones externas estas se puede observar al acercar el cursor del mouse a la elipse e inmediatamente se mostrará un mensaje que indica que el principio contiene interacciones externas.



Todas las interacciones externas si bien no se pueden visualizar en el sistema, estas influyen en el estado de los principios esenciales.

## 1.6 Sumas

Esta opción muestra una gráfica del resultado del análisis sistémico de los subprincipios del principio esencial seleccionado.

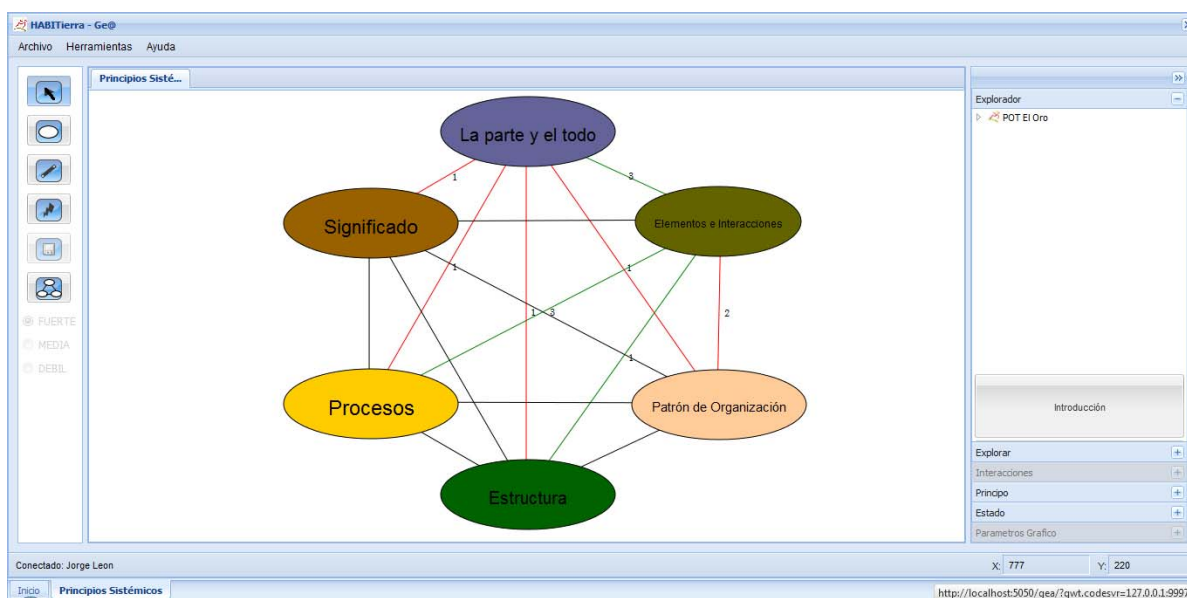


## 1.7 Eliminar


Elimina el principio esencial seleccionado.

## 2. Principios Sistémicos

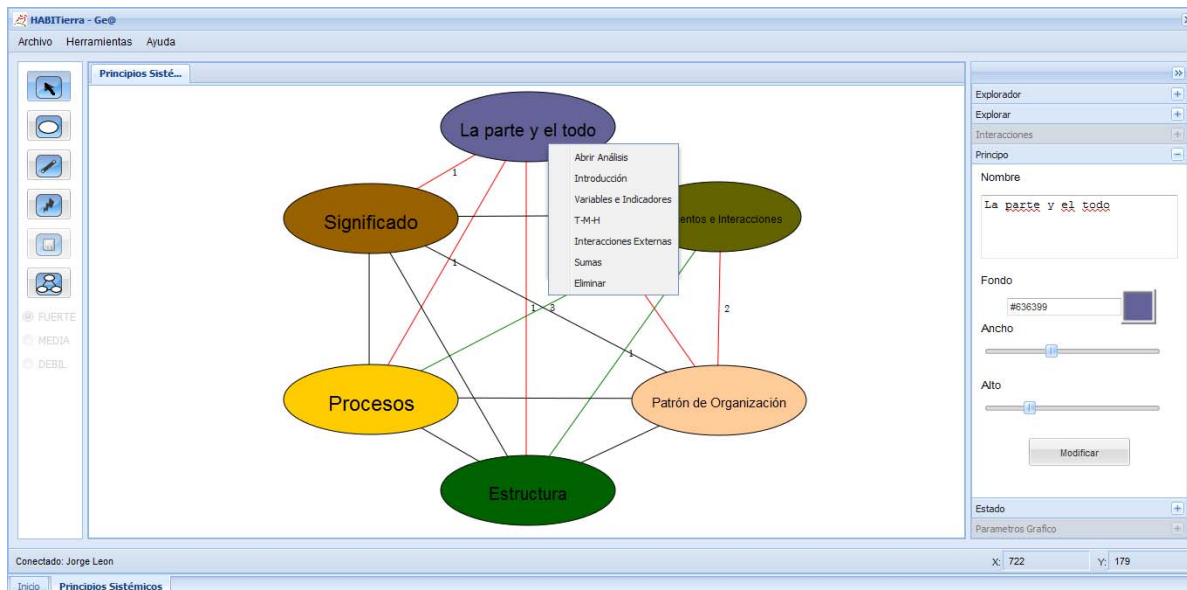
Al abrir se van a mostrar los principios sistémicos del Sistema Ge@ de la siguiente manera:



### Opciones de cada Principio

Cada principio esencial tiene varias opciones. Para poder ver las opciones de cada principio hay que escoger la opción seleccionar  de la barra de herramientas de la izquierda; luego hay que señalar el principio (con el botón izquierdo del ratón hacer clic sobre el óvalo del principio), una vez señalado, hacer clic con el botón derecho del ratón y escoger la opción deseada así.

Las opciones disponibles para cada principio esencial son: Abrir Análisis, Introducción, Variables e Indicadores, Teoría-Método-Herramientas (T-M-H), Interacciones Externas, Sumas y Eliminar.

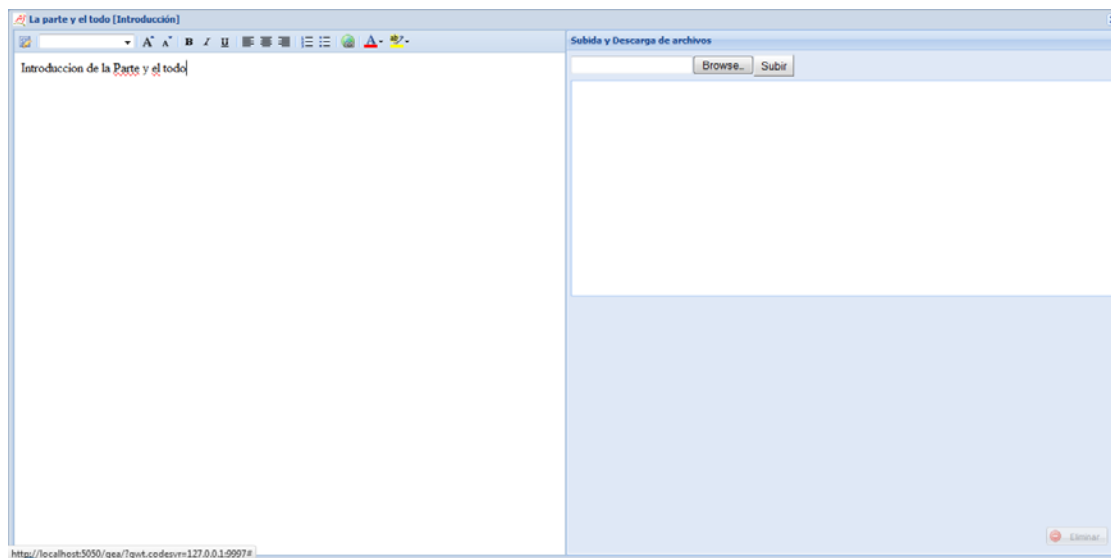


## 2.1 Abrir Análisis

Abre una nueva pestaña para indicar “*sub principios sistémicos*” que puede contener el principio sistémico escogido. Las opciones de un sub principio son iguales a las de un principio sistémico.

## 2.2 Introducción

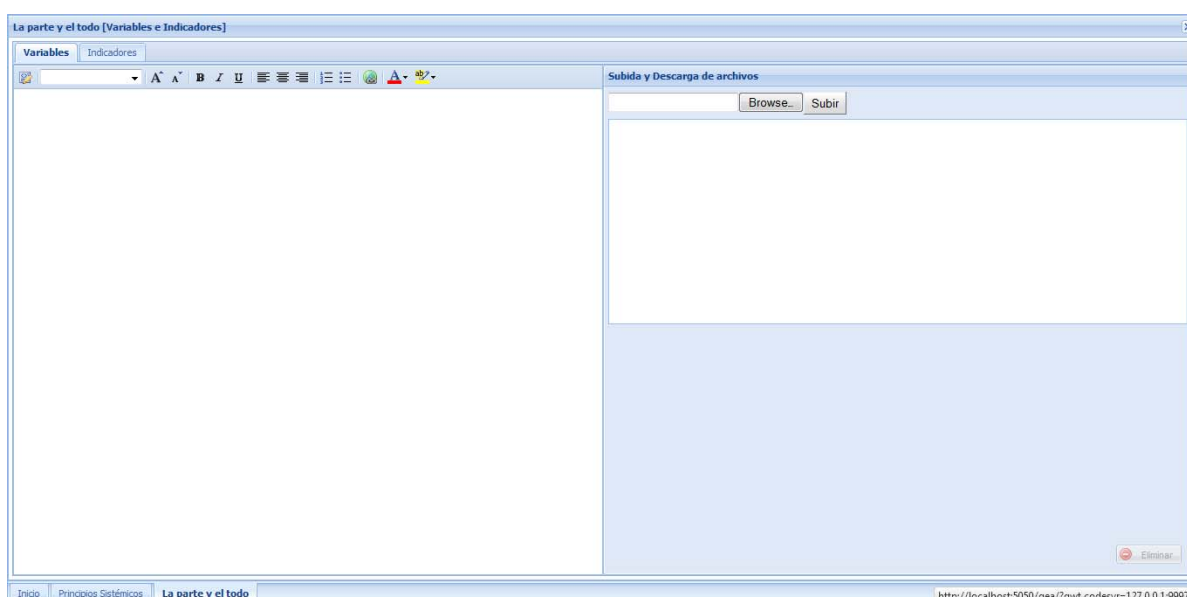
Al escoger la opción Introducción aparecerá la siguiente ventana con una introducción del principio.



Para aprender más sobre cómo manejar las opciones de esta ventana vea el punto Manejo de ítems y archivos digitales.

### 2.3 Variables e Indicadores

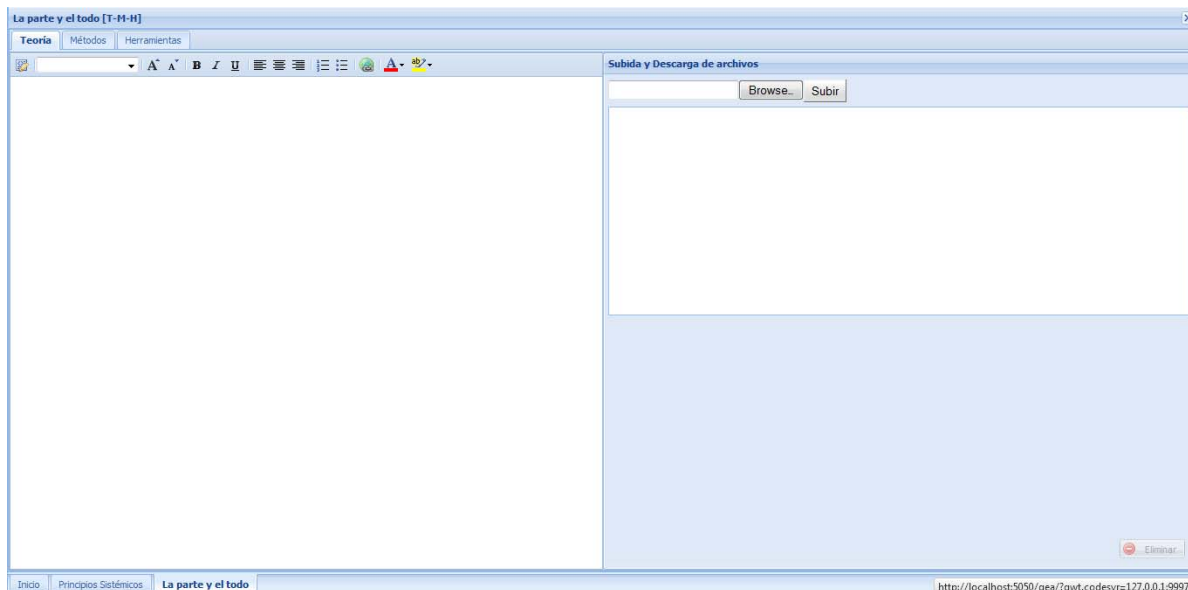
Al escoger la opción Variables e Indicadores aparecerá la siguiente ventana con variables e indicadores sobre el principio sistémico seleccionado.



En la parte izquierda se tienen las variables o indicadores y en la parte derecha archivos digitales que ayuden a sustentar estas variables o indicadores. Para ver los indicadores hay que hacer clic sobre la pestaña *Indicadores*. Para ocultar la ventana y grabar los cambios realizados hacer clic en la X en la parte superior, si es que se realizó un cambio aparecerá una ventana que le indicara si desea grabar los cambios, en caso positivo se guarda los cambio y sale, en caso contrario no los graba y sale, y una tercera opción que cancela la salida y continua en la ventana.

### 2.4 Teoría-Método-Herramientas (T-M-H)

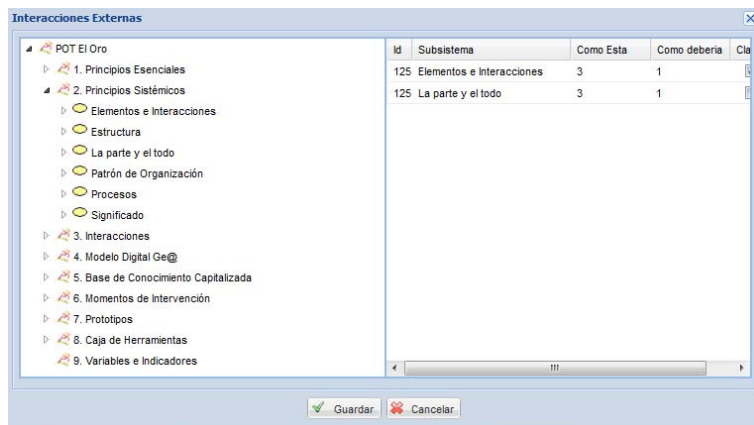
Al escoger la opción Teoría-Método-Herramientas aparecerá la siguiente ventana con la teoría, método y herramienta sobre el principio sistémico seleccionado.



En la parte izquierda se tiene la descripción de las teorías, métodos y herramientas y en la parte derecha archivos digitales que ayuden a sustentarlas. Para ver los métodos o herramientas hay que hacer clic sobre las demás pestañas. Para ocultar la ventana y grabar los cambios realizados hacer clic en la X en la parte superior, si es que se realizó un cambio aparecerá una ventana que le indicara si desea grabar los cambios, en caso positivo se guarda los cambios y sale, en caso contrario no los graba y sale, y una tercera opción que cancela la salida y continúa en la ventana.

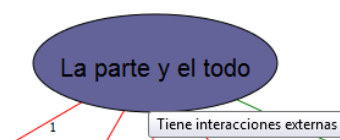
## 2.5 Interacciones Externas

La opción de interacciones externas permite al usuario visualizar a aquellas relaciones que no se pueden observar en la pantalla, aquellas que se han realizado con objetos que no se encuentran en el mismo nivel del sistema.



Esta ventana está conformada por dos áreas la primera es un árbol en donde se puede observar los objetos que se han creado en el Sistema Ge@ y a partir de aquí se seleccionará el que se desee relacionar con el principio. El proceso es simple, se selecciona y arrastra a la tabla. Aquí también se puede visualizar las relaciones externas que se hayan efectuado con anterioridad.

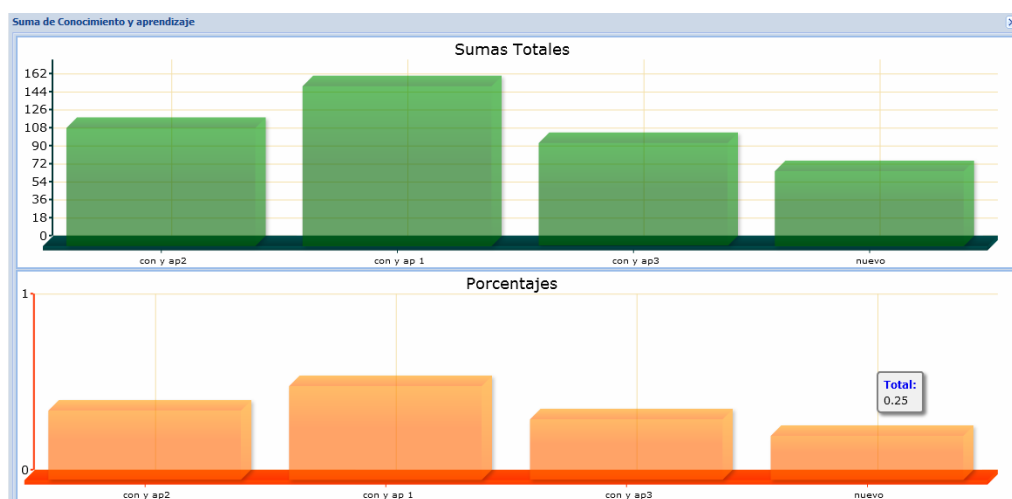
Si un principio contiene interacciones externas estas se puede observar al acercar el cursor del mouse a la elipse e inmediatamente se mostrará un mensaje que indica que el principio contiene interacciones externas.



Todas las interacciones externas si bien no se pueden visualizar en el sistema, influyen en el estado de los principios esenciales.

## 2.6 Sumas

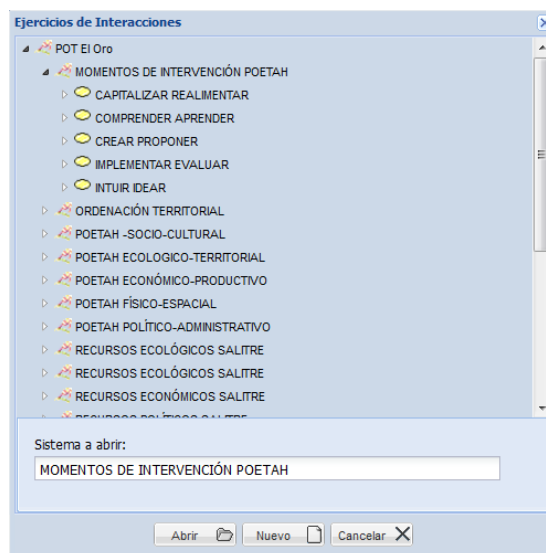
Esta opción muestra una gráfica del resultado del análisis sistémico de los subprincipios del principio esencial seleccionado.



## 2.7 Eliminar

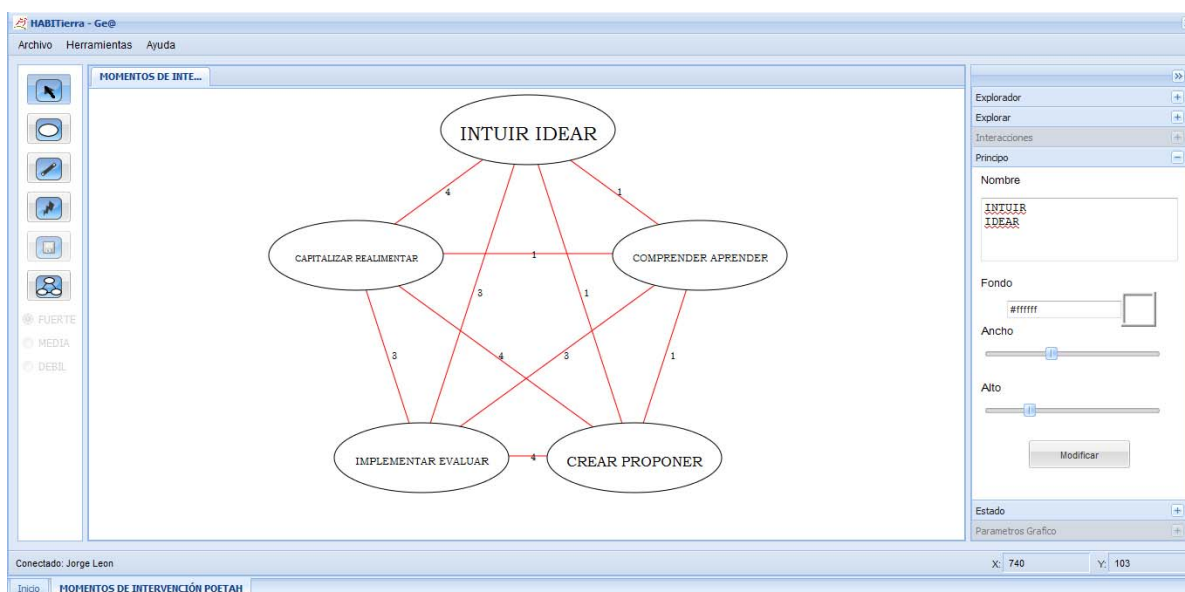
Elimina el principio esencial seleccionado.

### 3. Interacciones



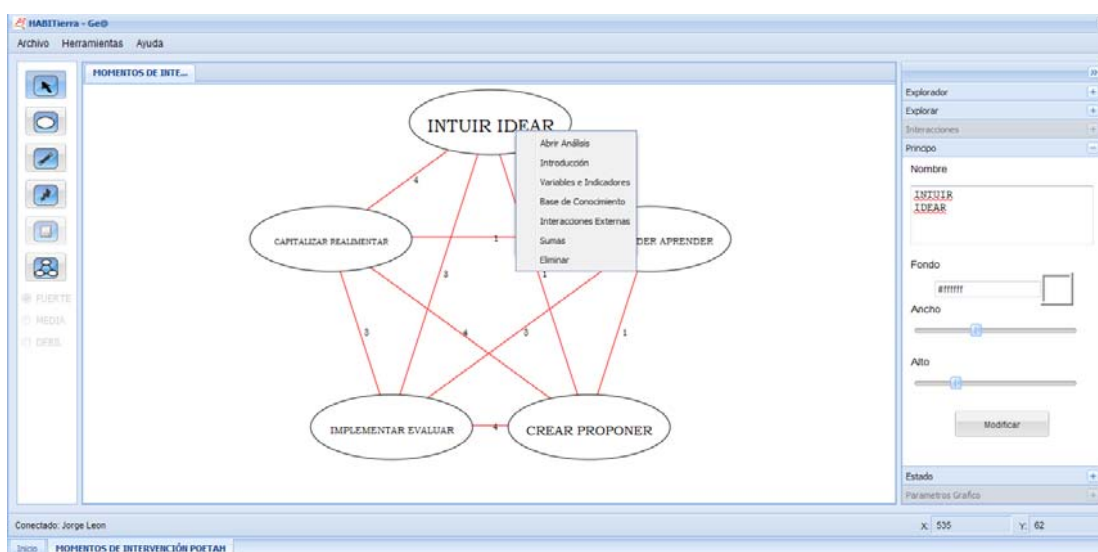
En el menú principal cuando se selecciona la tercera opción de Interacciones se muestra la ventana superior. Aquí se lista todos los ejercicios de interacciones que se hayan creado, es posible crear un nuevo ejercicio o abrir uno existente.

Al abrir uno existente se abre el sistema Ge@, cumple las mismas funcionalidades que en el resto de opciones como es el caso de los principios sistémicos y esenciales.



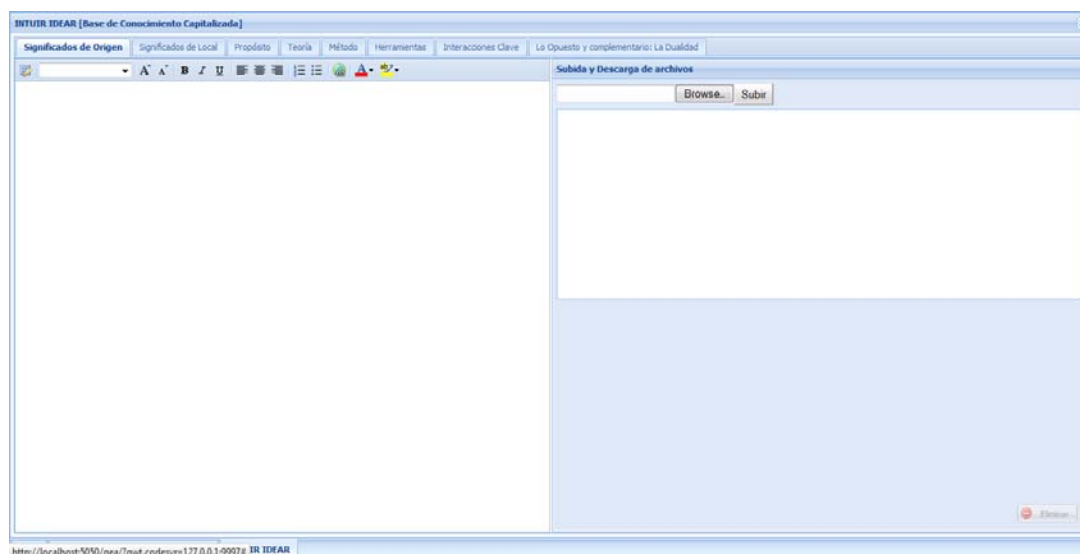
## Opciones de cada Ejercicio de Interacción

Cada elemento creado en Interacciones tiene varias opciones. Las opciones que se tienen son las mismas que en Principios Esenciales y Sistémicos, la única diferencia está en que la opción T-M-H se intercambia por Base de Conocimiento, como se puede visualizar en la siguiente imagen:



Si desea conocer más sobre la tarea que realiza cada opción puede hacerlo en la sección de *Opciones de Principios Esenciales o Sistémicos*.

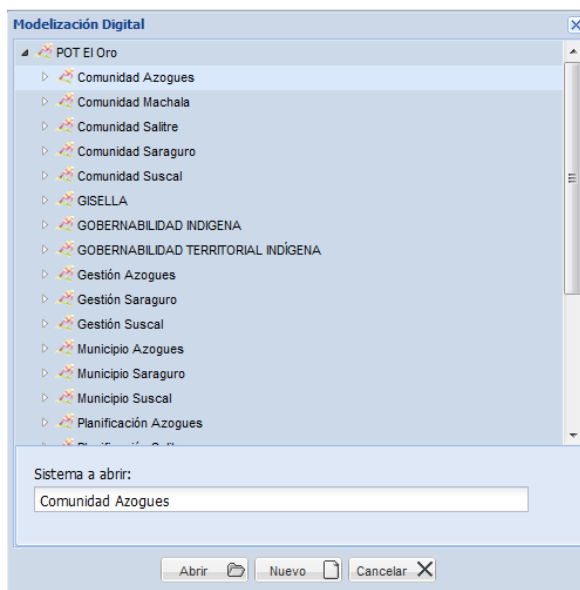
La opción Base de Conocimiento abre la siguiente ventana:



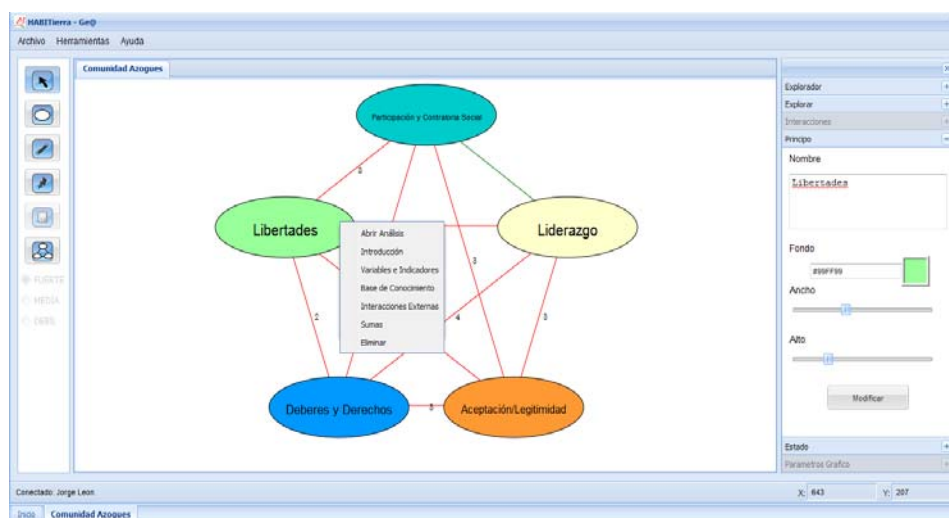


Esta ventana está conformada por nueve ítems, cada ítem dispone de un cuadro de texto y el panel de subida y descarga de archivos. Las acciones que se puede realizar sobre esta ventana son similares al resto de ítems en otros componentes de Ge@.

#### 4. Modelización Digital



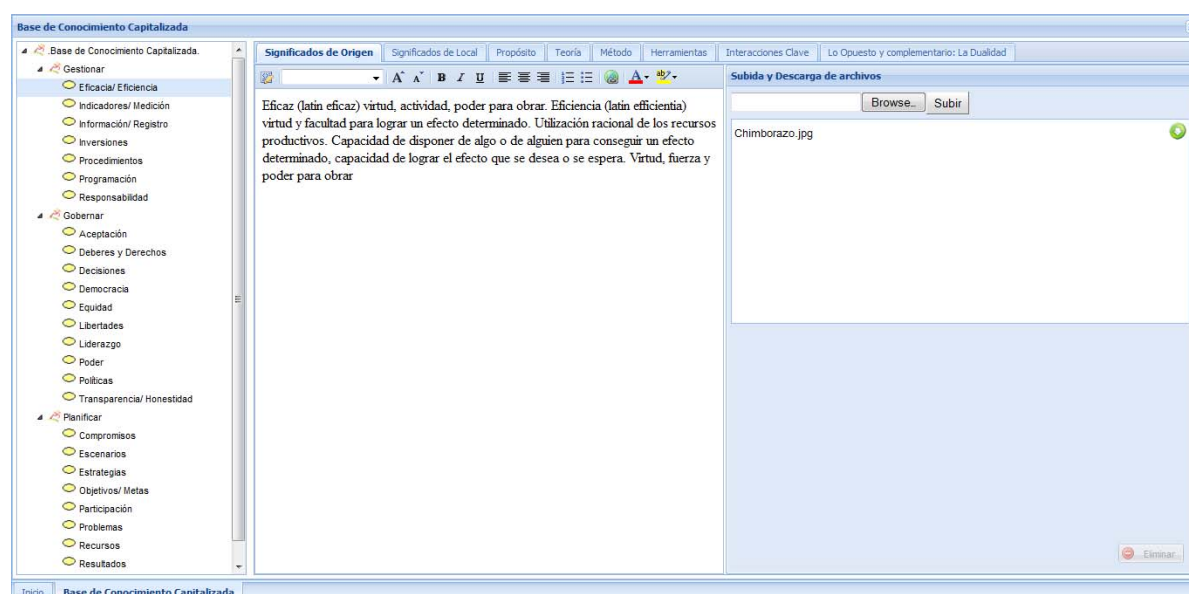
De igual manera que en Interacciones se dispone de una lista de todos los sistemas creados con Modelización Digital, la selección de cualquier sistema abrirá al Sistema Ge@ y las opciones de cada elemento creado en este componente son similares a las de Interacciones, como se puede visualizar en la siguiente imagen:



Estas opciones son Abrir Análisis, Introducción, Variables e Indicadores, Base de Conocimiento, Interacciones Externas, Sumas y Eliminar.

La descripción de cada una de estas opciones fueron descritas en las sección Interacciones, si desea conocer más sobre cada opción revisar la sección “*Opciones de cada Ejercicio de Interacción*”.

## 5. Base de Conocimiento Capitalizada

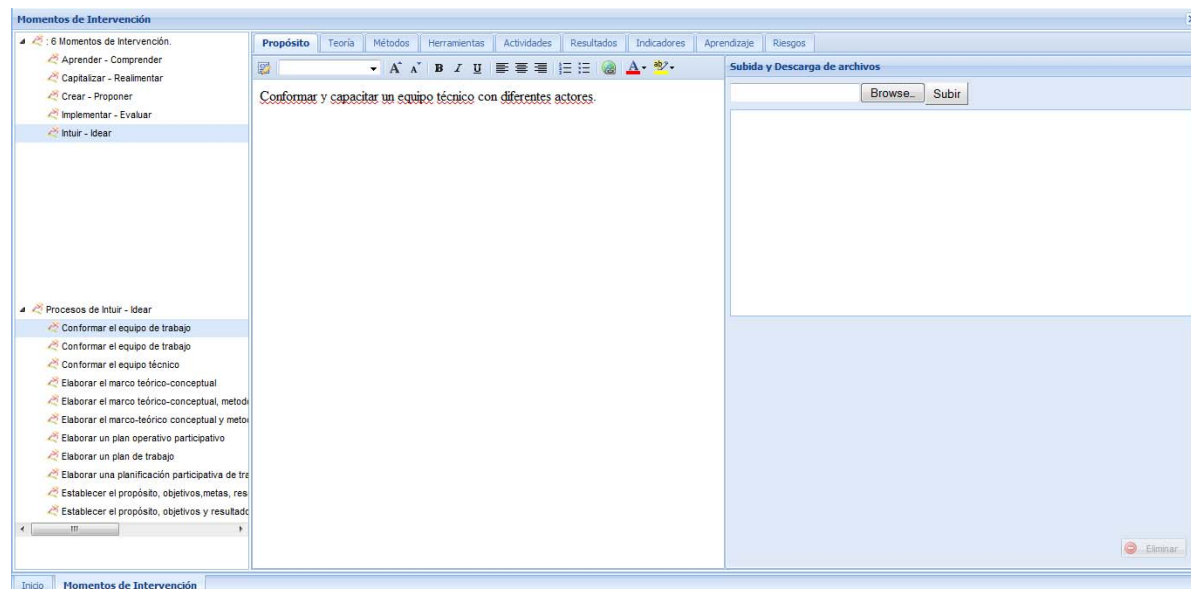


Esta opción es diferente a las anteriores, esta muestra una nueva ventana en donde se pueden visualizar los elementos principales que conforman la Base de Conocimiento Capitalizada.

Cada elemento está compuesto por ocho ítems que se visualizan como pestañas que se observa en la imagen superior.

Cada ítem está compuesto de un área de descripción y subida y descarga de archivos, si se desea conocer cómo se opera esta sección, revisar *Manejo de ítems y archivos digitales*.

## 6. Momentos de Intervención

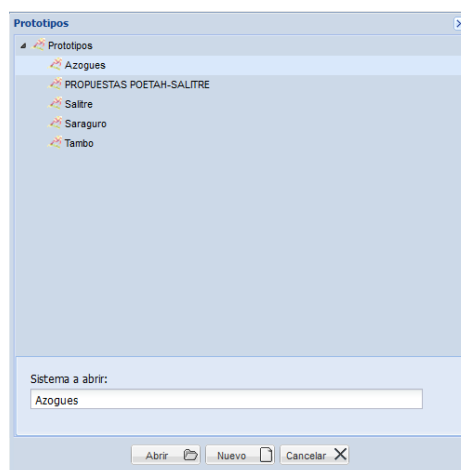


Esta opción es diferente a las anteriores, esta muestra una nueva ventana en donde se pueden visualizar los elementos principales que conforman los Momentos de Intervención.

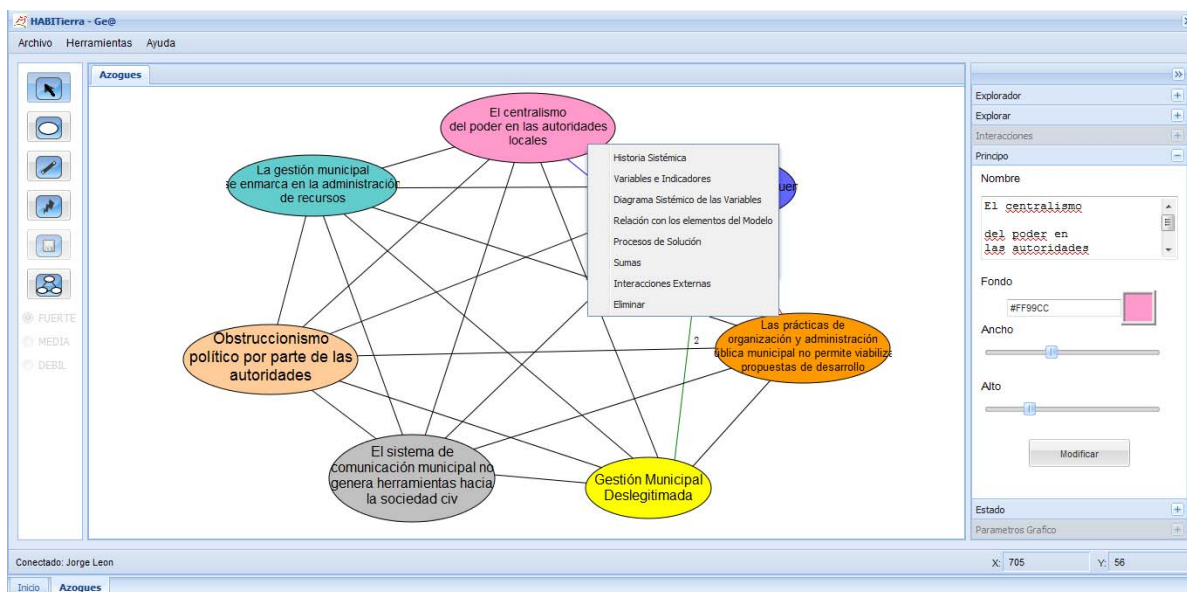
Cada elemento está compuesto por nueve ítems que se visualizan como pestañas que se observa en la imagen superior.

Cada ítem está compuesto de un área de descripción y subida y descarga de archivos, si se desea conocer cómo se opera esta sección, revisar *Manejo de ítems y archivos digitales*.

## 7. Prototipos

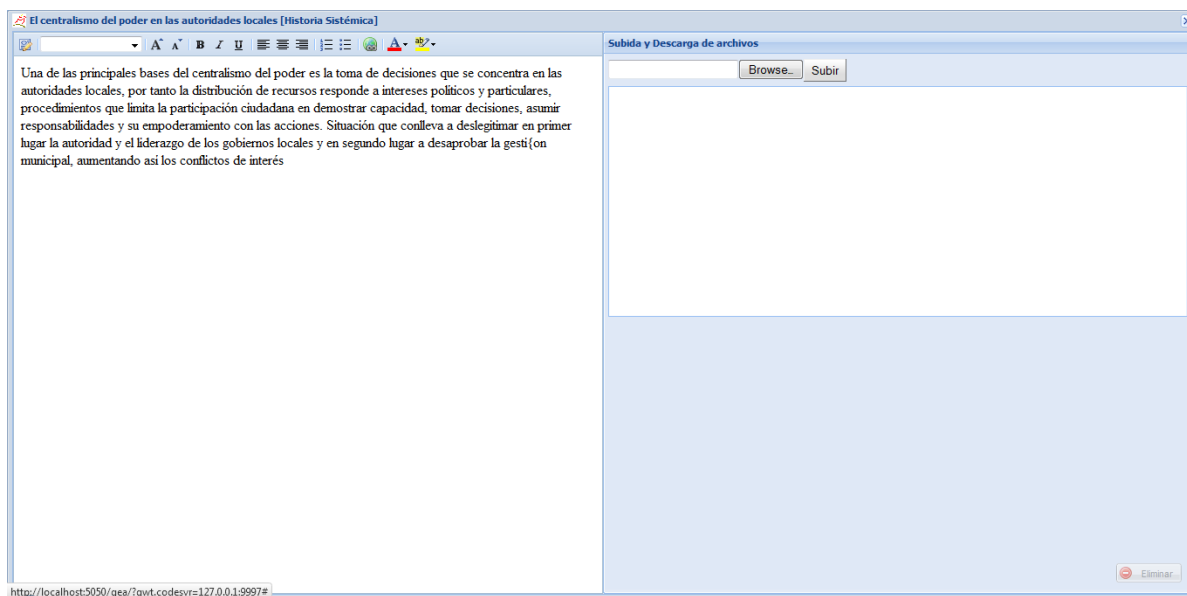


La opción prototipos abre una ventana en donde se lista los prototipos que se hayan creado, se puede crear nuevos prototipos o abrir uno existente. Al seleccionar uno y pulsar el botón abrir se mostrara la ventana del Sistema Ge@ como se puede visualizar en la siguiente imagen.



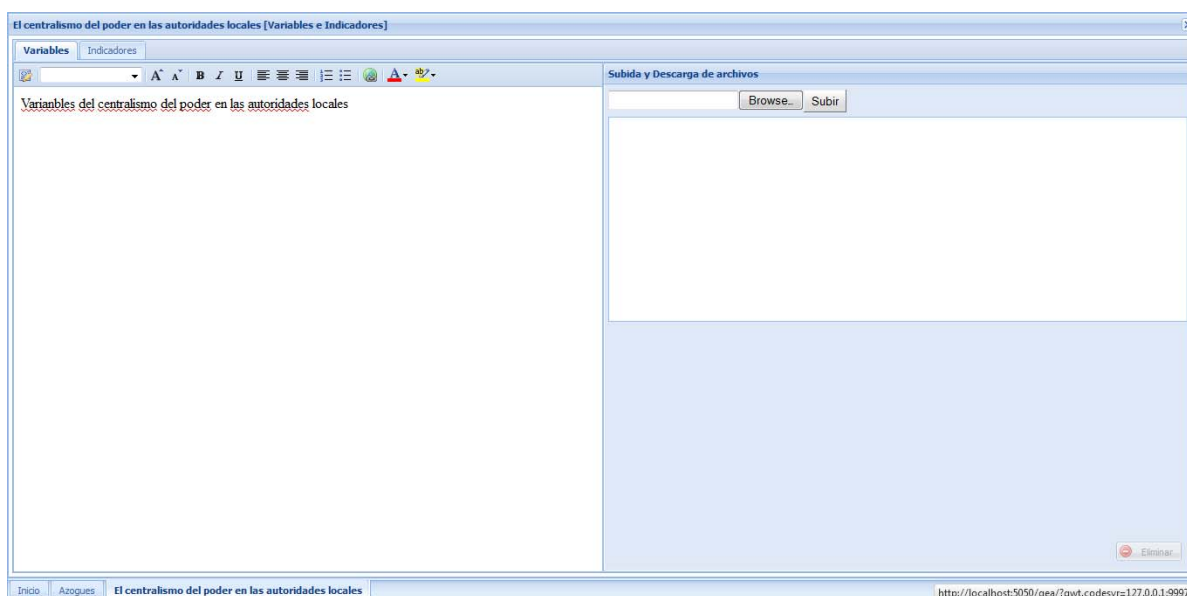
Esta ventana varía de las anteriores ya que dentro de la opción Prototipos se puede realizar distintas operaciones además las opciones de cada elemento que conforma prototipo cambia. Estas opciones son: Historia Sistémica, Variables e Indicadores, Diagrama Sistémico de las variables, Relación con los elementos del Modelo, Procesos de Solución, Sumas, Interacciones Externas y Eliminar.

## 7.1 Historia Sistémica



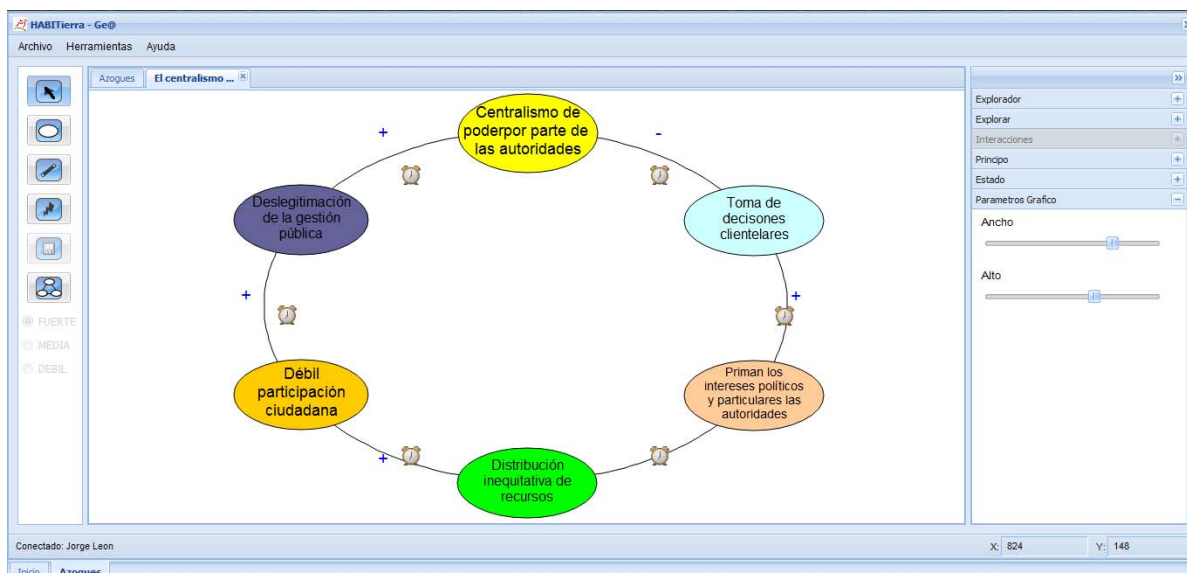
La ventana de Historia Sistémica es similar a la de Introducción de las opciones de Principios Sistémicos o Esenciales. Se tiene la descripción junto con la subida y descarga de archivos más en la sección de *Manejo de Ítems y Archivos Digitales*.

## 7.2 Variables e Indicadores



Similar a la ventana de Principios Esenciales o Sistémicos.

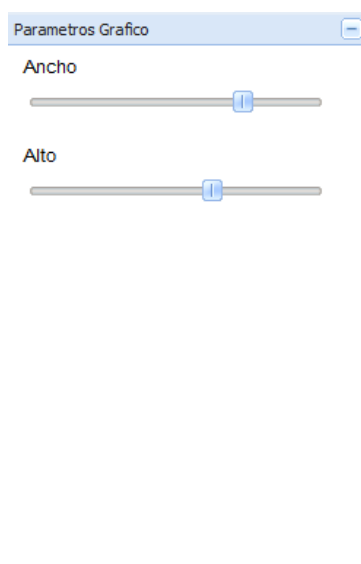
### 7.3 Diagrama Sistémico de las Variables



Esta opción abre una nueva pestaña como se puede observar en la imagen superior, el gráfico que se visualiza es diferente a los que se ha venido observando anteriormente, ya que este demuestra un ciclo a seguir, en donde cada relación indica un tipo y una demora, dentro de las propiedades de la interacción se tiene lo siguiente:

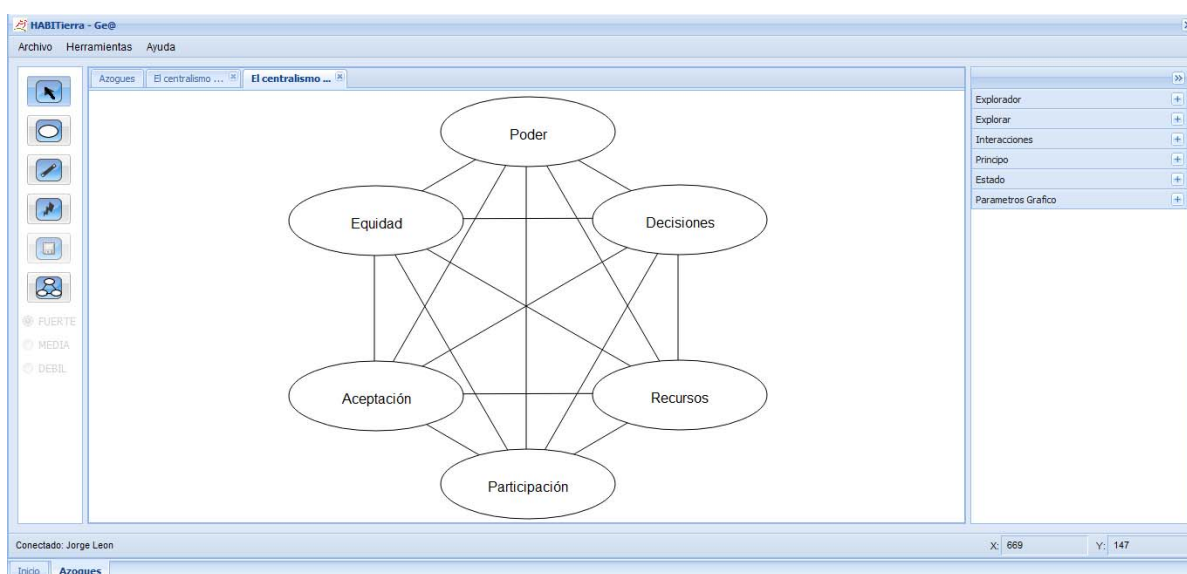
Cada opción se refleja en el diagrama como son el signo + o -, y la imagen de un reloj que indica que existe una demora.

Estos objetos (eclipses) no se pueden desplazar individualmente, siempre mantienen una posición única, si se desea modificar su posición esta se lo hace en conjunto con el resto de objetos en las propiedades de gráfico con el nombre **Parámetros Gráficos** que se puede observar en la siguiente figura.



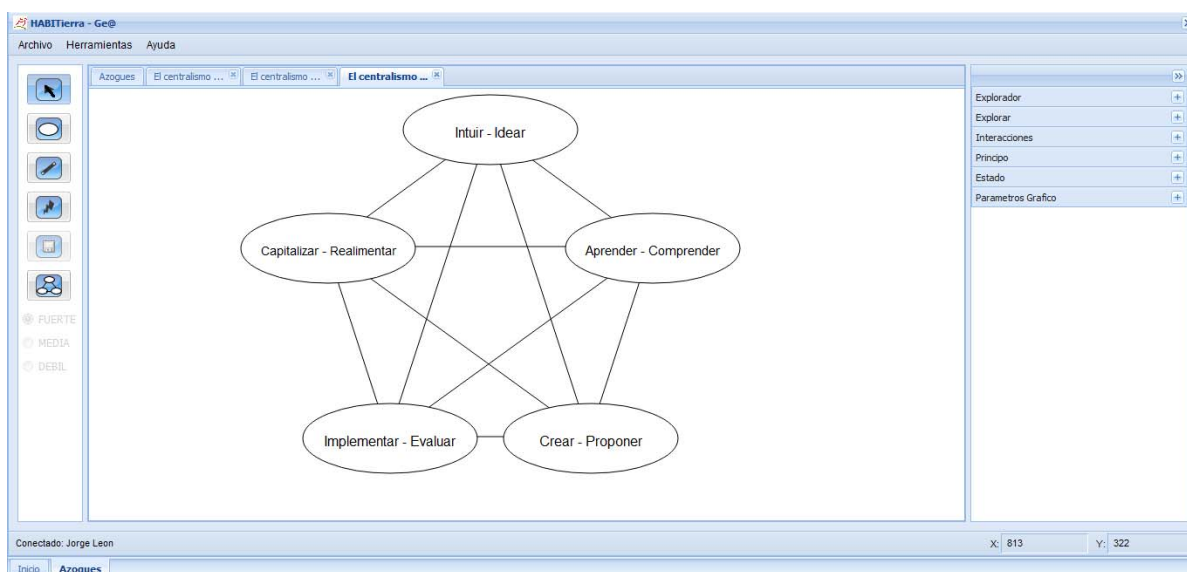
Todo el sistema en conjunto representa un eclipse mayor a la cual se puede modificar su alto o ancho, y el cual indicara el movimiento de cada objeto del sistema.

#### 7.4 Relación con los elementos del Modelo



De igual manera al seleccionar esta opción esta crea una nueva pestaña en donde se visualiza los elementos que conforman la relación que tienen con el Modelo Ge@, no hay mucha diferencia a las anteriores ventanas que se tienen con Principios Sistémicos o Esenciales.

## 7.5 Procesos de Solucion



De igual forma la ventana no cambia, la estructura es la misma que en las anteriores secciones, la manera en que se puede crear más procesos es similar a lo que se tiene en Principios Esenciales o Sistémicos.

## 7.6 Sumas

Es el mismo proceso que se tiene en los anteriores componentes que se han tratado en este manual. Muestra una gráfica del resultado del análisis sistémico de los subprincipios del prototipo seleccionado.

## 7.7 Interacciones Externas

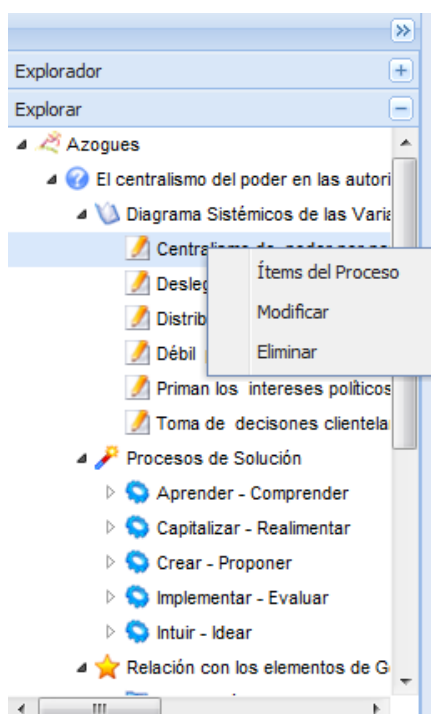
Cumplen el mismo propósito que en los anteriores componentes, permite relacionar con objetos que se encuentran externos al sistema actual.



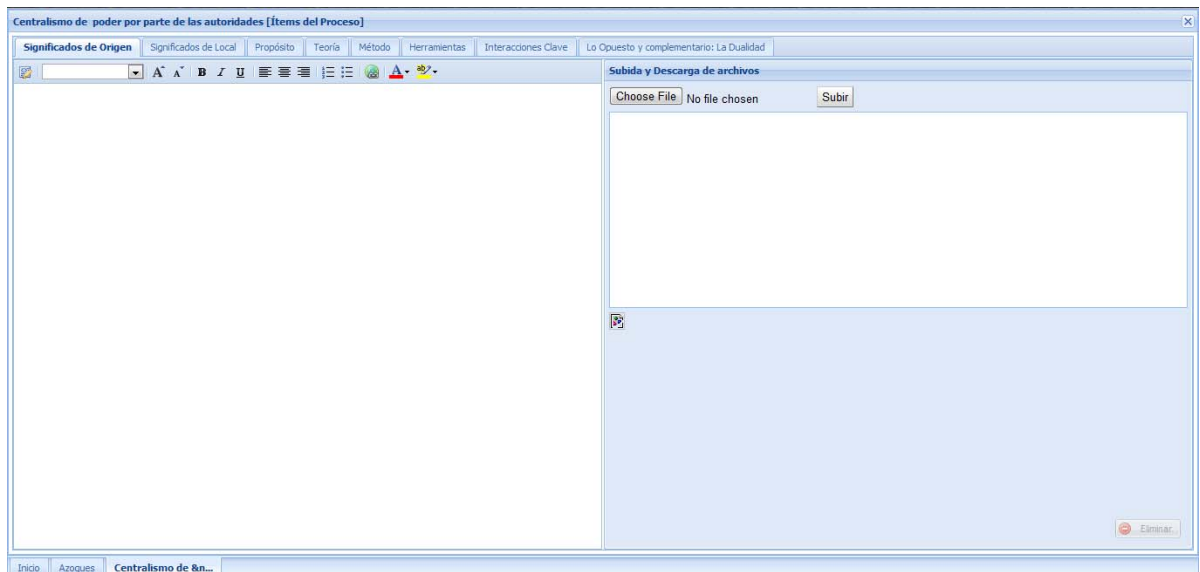
## 7.8 Eliminar

Permite eliminar el problema seleccionado.

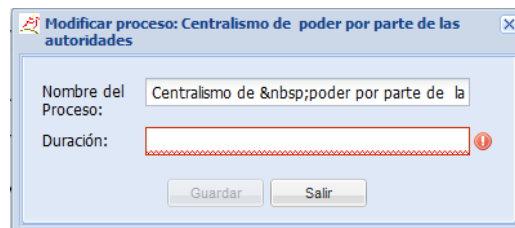
Dentro de la opción Prototipos, desde el árbol Explorar se pueden llevar a cabo distintas tareas, como se observa en la figura los elementos que conforman el diagrama sistémico de las variables disponen de un menú distinto a lo que se ha venido observando se tiene las siguientes opciones Ítems del Proceso, Modificar y Eliminar, que se pueden observar en este grupo de imágenes.



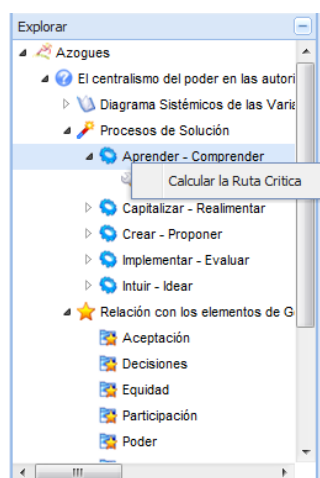
## Ítems del Proceso.



## Modificar



Continuando con el árbol explorador, tenemos una opción diferente para los Procesos de Solución, como se puede observar en la figura:



La opción Cálculo de la Ruta Crítica abre una nueva ventana, en donde el usuario puede determinar que procesos tienen predecesoras y si es que las tiene, se puede determinar la ruta crítica de estos procesos y el tiempo que tomará ejecutarlos.

Cálculo de la Ruta

Listado de procesos existentes para la tarea seleccionada

Tarea a la cual se le desea añadir una predecesora.

Lista de las tareas predecesoras

Para seleccionar las tareas predecesoras se seleccionan de una lista y se las arrastra a la otra lista y con esto se habrán generado las tareas predecesoras.

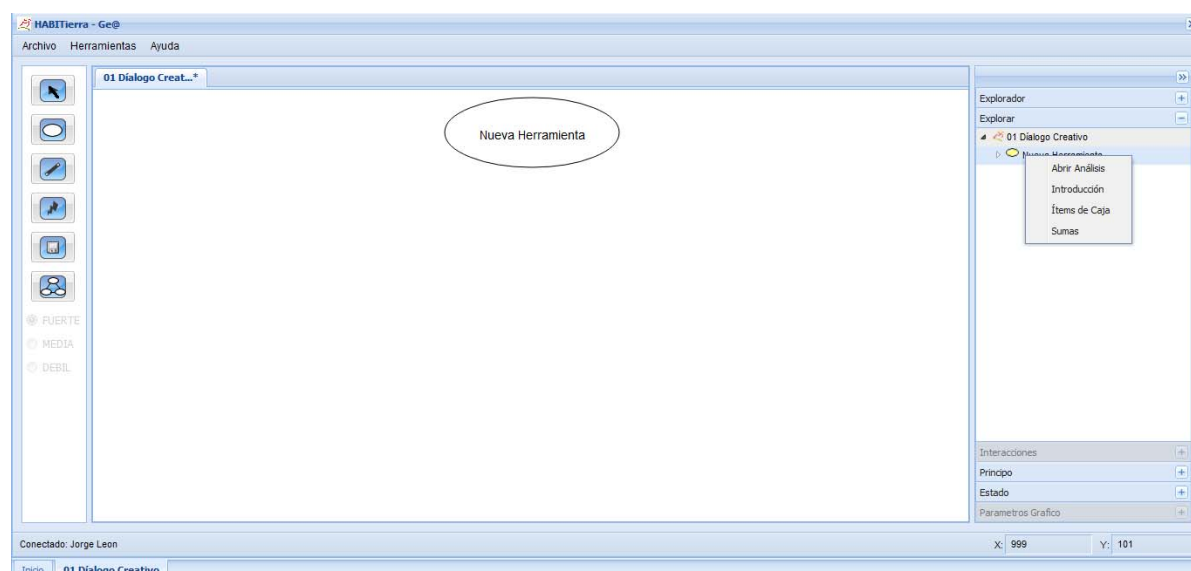
Una vez que se haya terminado de seleccionar aquellas tareas predecesoras, se pulsa el botón de calcular la ruta crítica y se abre la siguiente ventana.

	Pr	Democratizar los espacios...	Distribución eficiente / eficaz...	Elaborar una propuesta de...	Involucrar a la Sociedad civil, en procesos de...	Fin	t
Pr	*						0
Democratizar los espacios de toma de decisión		*	0		0		0
Distribución eficiente / eficaz de recursos			*				0
Elaborar una propuesta de Desarrollo Instituci...			0	*	0		0
Involucrar a la Sociedad civil, en procesos de...					*		0
Fin						*	0
T	0.0	0	0	0	0	0	0

Esta ventana muestra un listado de aquellos procesos que se consideran críticos y muestra el cálculo de la ruta crítica con una especie de matriz, como se puede observar en la figura.

## 8. Caja de Herramientas

La opción de caja de herramientas muestra nuevamente muestra un listado que indica la caja de herramientas de Ge@ cada ítem de esta lista abre nuevamente el Software Ge@, pero las opciones por cada elemento son distintas como se puede observar en la siguiente figura:



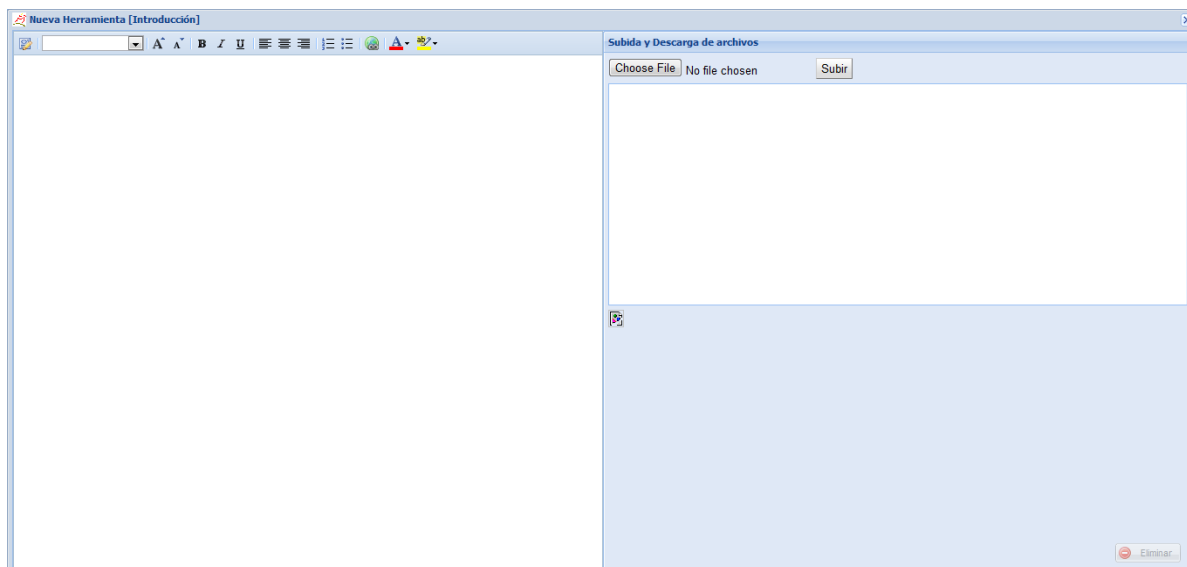
Vemos que las opciones que se presentan son Abrir Análisis, Introducción, Ítems de la caja y Sumas.

### 8.1 Abrir Análisis

Al seleccionar esta opción se abrirá una nueva pestaña en donde se cargarán los elementos con sus respectivas interacciones correspondientes al elemento de la caja de herramientas seleccionado.

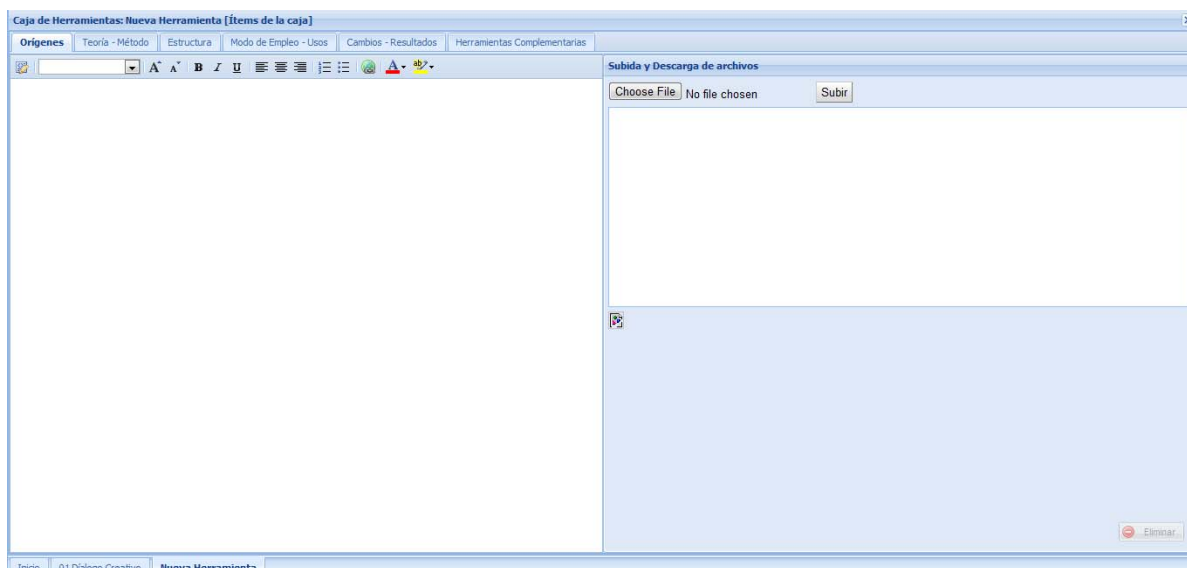
### 8.2 Introducción

La introducción es la misma como se ha venido observando en los distintos componentes de Ge@



### 8.3 Ítems de Caja

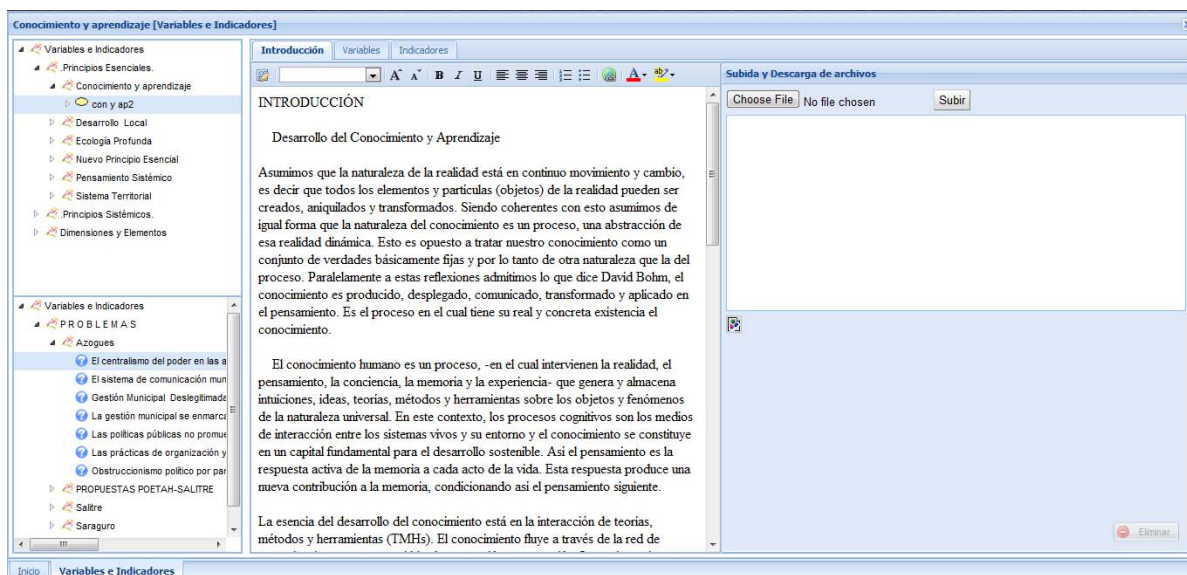
Al seleccionar esta opción se abrirá la siguiente ventana, que en general tiene la misma estructura que las demás, la diferencia se encuentra en los ítems que lo conforman, los cuales son: Orígenes, Teoría-Método, Estructura, Modo de Empleo – Usos, Cambios – Resultados y Herramientas Complementarias.



## 8.4 Sumas

De igual manera las sumas son una gráfica que muestra el estado del sistema, en este caso de los elementos que conforman la herramienta seleccionada.

## 9. Variables e Indicadores



Esta opción del Sistema Ge@, muestra las Variables e Indicadores de Principios Esenciales, Sistémicos, Base de Conocimiento y de los Problemas.