

UNIVERSIDAD DE CUENCA



FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

“SISTEMA DE TELEMEDICINA EN GASTROENTEROLOGÍA”

Tesis previa a la obtención del
título de Ingeniero en Electrónica y Telecomunicaciones

Autores:

Marco Stalin Bacuilima Oleas

CI. 0104162763

Juan Rosendo Molina León

CI. 0104826458

Director:

Dr. Carlos Villie Morocho Zurita

CI. 0300930328

Cuenca - Ecuador

2016

Resumen

El presente trabajo describe la implementación y diseño de un sistema de telemedicina, que almacena historias clínicas y estudios gastroenterológicos, cuyo nombre es *SAPHIENS*.

Este sistema está compuesto por: Hardware, parte encargada de la captura de imágenes y videos de un equipo endoscópico; llamado como la interfaz hombre-máquina. Software, responsable de la visualización y almacenamiento de los datos del paciente.

El capítulo uno da una breve introducción sobre la parte médica, específicamente sobre el sistema digestivo, analizando la incorporación de las telecomunicaciones en la medicina.

El diseño y ensamblaje de la interfaz hombre-máquina se detalla en el capítulo dos mientras que en el tres se describe el diseño e implementación de la base de datos y software del sistema.

Las herramientas utilizadas para el desarrollo del sistema se especifican dentro del capítulo cuatro. Finalmente se detallan los resultados y conclusiones obtenidos en la realización del sistema *SAPHIENS*.

Palabras clave: Telemedicina, Gastroenterología, Endoscopio, Arduino, JavaFx, Capturador de video, MySQL.

Bacuilima Marco - Molina Juan

Abstract

The present work, describes the implementation and design of a telemedicine system for gastroenterological studies and medical record storing. We decided to name it **SAPHIENS**.

It is made of: Hardware, responsible of the image and video acquisition from an endoscopic system; called the man-machine interface. Software, for patient data management and visualization.

The first chapter, gives a small introduction about medicine, more specifically the digestive system, analyzing the incorporation of the telecommunications in the medicine.

The assembly and design of the man machine interface is detailed in chapter number two, while chapter number three describes the implementation and design of the system software and database.

The tools that were used in the development of this system are detailed in the chapter number four.

Finally, we specify the conclusions and results obtained with **SAPHIENS** system.

Keywords: Telemedicine, Gastroenterology, Endoscope, Arduino, JavaFX, Video capturer, MySQL.

Bacuilima Marco - Molina Juan

Índice general

Resumen	2
Abstract	4
Índice general	6
Índice de figuras	11
Índice de tablas	14
1. Introducción y estado del arte	25
1.1. Fisiología	25
1.1.1. Fisiología del sistema disgestivo	26
1.1.2. Gastroenterología	26
1.1.3. Sistema digestivo	27
1.1.4. Pruebas diagnósticas	27
1.2. Equipos de adquisición de imágenes gastroenterológicas	28
1.2.1. Endoscopio	28
1.3. Telemedicina	29
1.3.1. Definición de la telemedicina	29
1.3.2. Objetivos de la telemedicina	30
1.3.3. Limitantes de la telemedicina	31
1.3.4. Retos de la telemedicina	32
1.3.5. La telemedicina en el Ecuador	32
1.3.6. Cronología de hitos importantes de la telemedicina a nivel mundial	34
1.4. Sistemas de telemedicina en gastroenterología	36
1.5. Planteamiento del problema	40
1.6. Objetivos generales y específicos	41
1.6.1. Objetivo general	41
1.6.2. Objetivos específicos	42
1.7. Metodología a utilizar	42



2. Adquisición de los datos	43
2.1. Interfaz hombre-máquina para clasificar las imágenes de interés	43
2.1.1. Salidas de video del equipo endoscópico	44
2.1.1.1. Conector de salida <i>Y/C</i>	46
2.1.1.2. Terminal de salida de video compuesto	47
2.2. Dispositivo capturador de video	47
2.2.1. Convertidor analógico-digital (<i>ADC</i>)	47
2.2.2. Adquisición del equipo capturador de video	49
2.2.3. Estándar de video NTSC	50
2.2.4. Estándar de video PAL	51
2.2.5. Elementos electrónicos del dispositivo capturador de video	52
2.3. Función del módulo arduino	54
2.3.1. Programa para el control del disparo de captura de video	57
2.3.2. Dispositivo pedal	59
2.4. Construcción y montaje del dispositivo	60
2.4.1. Parte superior	61
2.4.2. Parte inferior	61
2.4.3. Parte lateral	62
2.4.4. Parte posterior	63
2.4.5. Parte anterior	64
2.5. Distribución interna de la parte física de SAPHIENS	64
3. Sistema de telemedicina para gastroenterología	69
3.1. Manejo de la información	70
3.2. Almacenamiento	71
3.2.1. Diseño y creación de la base de datos	71
3.2.2. Esquema de la base de datos	72
3.3. Características técnicas para la creación del software	74
3.4. Visualización	75
3.4.1. Software–interfaz gráfica	75
3.4.2. Visualización de las imágenes y videos	76
3.4.2.1. Características de las imágenes y videos obtenidos	76



3.5.	Herramientas relevantes para la visualización de imágenes y videos	77
3.5.1.	Reconocimiento de cámaras y puertos	77
3.5.2.	Sistema de archivos base de datos CIE10	81
3.5.3.	Biopsias	81
3.5.4.	Informes	81
3.6.	Posible uso de esta plataforma en telemedicina.	83
4.	Herramientas de hardware y software utilizadas	85
4.1.	Herramientas de hardware	85
4.1.1.	Arduino	85
4.1.2.	Capturador de video	86
4.2.	Herramientas de software	87
4.2.1.	Netbeans	87
4.2.2.	JavaFX	88
4.2.2.1.	Características	89
4.2.3.	Scene Builder	89
5.	Resultados	91
6.	Conclusiones	97
6.1.	Conclusiones	97
A.	Capturas del sistema	101
B.	Base de datos	105
C.	Artículo	107
D.	Exámen gastroenterológico	113
E.	Código fuente para generación de informes	115
F.	Código fuente del software SIM 1.1	121
G.	STSIVA XXI, 2016	195
H.	Manual de Usuario	197



Bibliografía

200



UNIVERSIDAD DE CUENCA
desde 1867

Índice de figuras

1.1. Esquema del funcionamiento del endoscopio [1]	28
2.1. Interfaz hombre máquina. Elaboración Propia	44
2.2. Equipo endoscopio Olympus [2]	45
2.3. Conectores de salida de video. Elaboración propia	46
2.4. Conector de salida Y/C. Elaboración propia	46
2.5. Conector de video compuesto. Elaboración propia	47
2.6. Digitalización de una señal a partir de un convertidor de 3 bits [3]	49
2.7. CCD (dispositivo – chip de carga acoplada). (1) Píxeles individuales recogen la información de brillo. (2) Registros verticales que llevan esta información al (3) registro horizontal; a continuación, envía la señal de vídeo en bruto (4) a un amplificador de salida. Los píxeles se leen una fila a la vez en el área de transferencia de carga horizontal [4]	51
2.8. Dispositivos capturadores de video. Elaboración propia	52
2.9. Electrónica Interna de los dispositivos capturadores de video. Elaboración propia	53
2.10. Presentación del armazón del dispositivo físico de SAP-HIENS . Elaboración propia	55
2.11. Circuito pull-up para el control de disparo de captura de video [5]	56
2.12. Arduino micro [6]	57
2.13. Dispositivo pedal [7]	60
2.14. Cable de audio–jack mono. Elaboración propia	60
2.15. Parte superior de la caja de montaje del dispositivo. Elaboración propia	61



2.16. Parte inferior de la caja de montaje del dispositivo. Elaboración propia	62
2.17. Parte lateral de la caja de montaje del dispositivo. Ela- boración propia	63
2.18. Parte posterior de la caja de montaje del dispositivo. Elaboración propia	63
2.19. Parte anterior de la caja de montaje del dispositivo. Elaboración propia	64
2.20. Montaje del dispositivo físico	65
2.21. Montaje del dispositivo físico	65
2.22. Montaje del dispositivo físico	66
2.23. Montaje del dispositivo físico	67
2.24. Montaje del dispositivo físico	68
3.1. Diagrama de flujo de la información del paciente. Ela- boración propia	71
3.2. Diagrama de flujo de la información del Médico espe- cialista. Elaboración propia	72
3.3. Imagen obtenida por el software. Elaboración propia .	77
3.4. Módulo de biopsias. Elaboración propia	82
3.5. Diagrama de bloques del funcionamiento de SAPHIENS . Elaboración propia	84
4.1. Arduino ATmega 328 [8]	86
A.1. Interfaz Gráfica del sistema	102
A.2. Herramientas relevantes	103
A.3. Base de datos <i>CIE10</i>	104
B.1. Diagrama de la base de datos	106
F.1. splash.fxml	155
F.2. principal.fxml	156
F.3. acercaDe.fxml	161
F.4. pedidos.fxml	163
F.5. pedidoHecho.fxml	168
F.6. imagen.fxml	175
H.1. Ventana principal de SIM 1.1	197



H.2. Ventana donde muestra los estudios. 198
H.3. Información de un estudio realizado. 199



UNIVERSIDAD DE CUENCA
desde 1867

Índice de tablas

4.1. Especificaciones encore, [9]	87
---	----



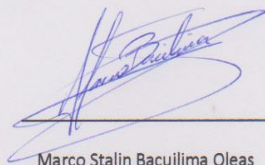
UNIVERSIDAD DE CUENCA
desde 1867



Universidad de Cuenca
Clausula de derechos de autor

Yo, Marco Stalin Bacuilima Oleas, autor de la tesis "*Sistema de telemedicina en gastroenterología*", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de *Ingeniero en Electrónica y Telecomunicaciones*. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.

Cuenca, Octubre 2016.



Marco Stalin Bacuilima Oleas

C.I: 0104162763



Universidad de Cuenca
Clausula de derechos de autor

Yo, Juan Rosendo Molina León, autor de la tesis "*Sistema de telemedicina en gastroenterología*", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de *Ingeniero en Electrónica y Telecomunicaciones*. El uso que la Universidad de Cuenca hiciera de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor.

Cuenca, Octubre 2016.

Juan Rosendo Molina León

C.I: 0104826458



Universidad de Cuenca
Clausula de propiedad intelectual

Yo, Marco Stalin Bacuilima Oleas, autor de la tesis "*Sistema de telemedicina en gastroenterología*", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, Octubre 2016

A handwritten signature in blue ink, appearing to read 'Marco Stalin Bacuilima Oleas', written over a horizontal line.

Marco Stalin Bacuilima Oleas

C.I: 0104162763



Universidad de Cuenca
Clausula de propiedad intelectual

Yo, Juan Rosendo Molina León, autor de la tesis "*Sistema de telemedicina en gastroenterología*", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, Octubre 2016

A handwritten signature in blue ink, appearing to read "Juan Rosendo Molina León", written over a horizontal line.

Juan Rosendo Molina León

C.I.: 0104826458

CERTIFICO

Que el presente proyecto de tesis: “Sistema de telemedicina en gastroenterología” fue dirigido por mi persona.

Dr. Carlos Villie Morocho Zurita
C.I. 0300930328

Agradecimientos

Mi agradecimiento va dirigido a mi familia, por el apoyo que tuve de ellos en todo momento y en todos estos años mientras crecía profesionalmente. Sin olvidar a esa persona que ha estado presente en los momentos más difíciles de culminación de esta carrera, que me ha ayudado en todo lo que podía y que siempre me dió ánimos para seguir hacia adelante, esa persona es mi esposa que siempre confió en mí y que nunca la defraudaré. Agradezco también a mi compañero de tesis Juan Molina que muchas veces se puso con todo el trabajo al hombro para poder realizar un proyecto de calidad y sobretodo distinto a lo normal.

Marco

Sin agradecimientos.

Juan

Dedicatoria

Esta tesis va dedicada con todo el cariño para mi madre y a mi esposa, que soñaban con que cumpliera este sueño tan anhelado. Todo el esfuerzo y trabajo duro que he dado en esta carrera ha sido para poder verlas contentas.

Marco

Sin dedicatoria.

Juan

Capítulo 1

Introducción y estado del arte

Este capítulo aborda conceptos de la medicina y como ha ido cambiando constantemente con la ayuda de la tecnología y la telemedicina. Los conceptos que se presentan van desde la definición de la fisiología, la gastroenterología, el sistema digestivo, tipos de pruebas diagnósticas y equipos utilizados para la adquisición de imágenes. Finalmente, se define la telemedicina, sus objetivos, retos, los tipos de sistemas utilizados, los cambios producidos en el Ecuador por la telemedicina y una cronología de hitos importantes ocurridos en esta área.

1.1. Fisiología

La Biología es la ciencia que se ocupa de los seres vivos, conocida también como la ciencia de la vida, la misma que está dividida en: *anatomía*, *química orgánica*, *botánica*, *zoología*, y en *fisiología* [10].

La *anatomía* describe la organización del cuerpo, la *botánica* y *zoología* describe las semejanzas, diferencias y clasificación de seres vivos, la *química orgánica* estudia la interacción entre sus partes, componentes y elementos que lo componen.

La fisiología, que es la parte de interés para el presente trabajo de



titulación, se dedica a estudiar el funcionamiento del cuerpo y sus partes, como los órganos, tejidos y células [11].

Las funciones de los órganos tienen como objetivo la conservación del individuo, las mismas que se dividen en funciones nutricionales, de relación y de reproducción [12]. Entre las funciones de nutrición comprende la digestión, que es la que se aborda como fisiología del sistema digestivo.

1.1.1. Fisiología del sistema digestivo

La digestión empieza desde la boca. El bolo alimenticio formado al masticar y al mezclarse con la saliva, viaja por la faringe, el esófago, hasta llegar al estómago, lugar donde se procesa hasta convertirse en una mezcla denominada quimo [13]. A la salida del estómago, el tubo digestivo se prolonga al intestino delgado. Donde recibe secreciones de las glándulas intestinales, la bilis y los jugos del páncreas. La gran cantidad de enzimas que contienen estas secreciones van degradando y transformando los alimentos en sustancias más simples.

Luego continúa por el intestino grueso, aquí se produce una absorción de líquidos para formar una masa más sólida. Termina en el esfínter anal, donde los restos no digeridos de los alimentos son evacuados.

1.1.2. Gastroenterología

Estudia el funcionamiento del sistema digestivo y por ende sus enfermedades. Se ocupa de los órganos y de las glándulas de este sistema, como del hígado, estómago, páncreas e intestino delgado. La proctología (estudio del ano y del recto) y la hepatología (estudio del hígado)



son dos especialidades muy importantes en la gastroenterología [14].

Esta rama se apoya del estudio clínico del paciente en métodos de laboratorio y en pruebas de imagenología, de esta manera se comprende más del funcionamiento del sistema digestivo.

Entonces, para tener una mayor comprensión de la función que cumple el equipo que se diseña y construye en este tema de tesis, es necesario revisar algunos conceptos básicos de la gastroenterología.

1.1.3. Sistema digestivo

El *sistema digestivo* está formado por la boca, el esófago, el estómago, el intestino delgado, el intestino grueso, el recto y el ano, conocido como el tracto digestivo. También está conformado por otros órganos como la lengua, las glándulas salivales, el páncreas, el hígado y vesícula biliar, que ayudan al cuerpo a transformar y absorber los alimentos [15].

1.1.4. Pruebas diagnósticas

Muchas enfermedades relacionadas con el sistema digestivo pueden ser diagnosticadas solo examinando al paciente, pero existen algunas que se diagnostican mediante pruebas ya sean de imágenes o de laboratorio.

Las pruebas en imagenología se dividen entre invasivas y no invasivas, las de interés para este tema son las invasivas ya que el equipo a usar es un endoscopio, el mismo que debe ingresar al intestino para obtener imágenes y así dar un diagnóstico [16].

1.2. Equipos de adquisición de imágenes gastroenterológicas

El estudio de la patología intestinal se puede realizar con una gran variedad de técnicas de imagen que permitan explorar el tracto digestivo de forma rápida y precisa. Es importante que el gastroenterólogo conozca las distintas modalidades.

Los equipos más utilizados son: los equipos de ultrasonido de abdomen, los de resonancia magnética, los de arteriografía selectiva, y el endoscopio, el mismo que es de interés para esta tesis [17].

1.2.1. Endoscopio

Es un equipo médico que realiza pruebas diagnósticas del tracto digestivo, para poder observar en una pantalla las imágenes del tubo digestivo a medida que se va introduciendo por la boca una sonda flexible que tiene una pequeña cámara y una luz (ver Figura 1.1), hasta llegar al estómago [18].

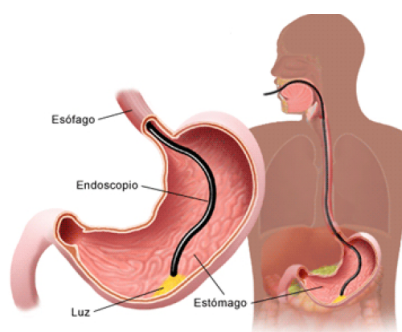


Figura 1.1: Esquema del funcionamiento del endoscopio [1]

La endoscopia se utiliza a menudo para examinar y tratar partes del tubo digestivo, dependiendo del sector que se va a visualizar mediante el endoscopio, el examen suele cambiarse de nombre, como ejemplo



el examen de endoscopía que visualiza el intestino delgado, así también puede ser la sigmoidoscopia, la broncoscopia, la cistoscopia, entre otros [19].

Como se ha visto, existen hoy múltiples modalidades de imagen para explorar la patología del intestino. La más ventajosa y utilizada es la endoscopia, ya que las imágenes son tomadas directamente y se puede observar el estado del estómago. Teniendo ya un grado de conocimiento acerca del funcionamiento del sistema digestivo, también de cómo funciona un endoscopio, y al ser un examen muy común en el Ecuador, se enmarca la necesidad de optimizar este tipo de equipos.

1.3. Telemedicina

La telemedicina es tan antigua como la primera llamada telefónica, existen antecedentes que se lograron transmitir electrocardiogramas desde barcos por la década de los 60. Hoy se cuenta con diagnósticos remotos en prácticamente todas las especialidades médicas.

La telemedicina suele utilizarse en circunstancias de difícil acceso, como en el área militar, para andinistas, en estaciones especiales, en atención remota para mujeres embarazadas con riesgo de aborto, para personas de tercera edad que no pueden movilizarse [20].

1.3.1. Definición de la telemedicina

La telemedicina se define como el servicio de la medicina a largas distancias. Puede decirse que la telemedicina es desde una discusión telefónica entre dos profesionales hasta una consulta, diagnóstico o



cirugía mediante la más alta tecnología existente en tiempo real [20].

Así cómo evoluciona la telemedicina, también se debe modificar su definición e ir ampliando las categorías, tecnologías, métodos de continuo desarrollo.

1.3.2. Objetivos de la telemedicina

Los objetivos de la telemedicina en el transcurso del tiempo han ido creciendo al igual que su definición. Según [21] los objetivos son :

- Prevenir, alertar, supervisar y controlar la expansión de enfermedades transmisibles y no transmisibles, mejorando la vigilancia epidemiológica.
- Contribuir a la integración del sistema de salud y la universalidad de los servicios de salud con calidad, eficiencia y equidad para beneficio prioritario de las poblaciones excluidas y dispersas.
- Promover la colaboración entre gobiernos, planificadores, profesionales de la salud, sociedad civil organizada y comunidades locales para crear un sistema de información y atención de salud fiable, y con calidad; fomentando así la capacitación, educación e investigación para la prevención y control de enfermedades.
- Agilizar la atención en salud, definiendo en tiempo real conductas a seguir (afinar los diagnósticos de los médicos en áreas rurales).
- Adelantar campañas preventivas y de tamizaje en la población.
- Justificar remisiones de pacientes o evitarlas si pueden ser de manejo del nivel del sitio de referencia a fin de no efectuar desplazamientos innecesarios.



- Facilitar diagnósticos más oportunos y tratamientos menos costosos por la oportunidad de una detección temprana de la enfermedad.

Estas actividades están en diversos niveles de desarrollo en algunos países, sin embargo, hoy se puede encontrar que la telemedicina se usa básicamente en dos áreas de trabajo: la práctica médica y la educación.

1.3.3. Limitantes de la telemedicina

Para este año (2016), se ha desarrollado la tecnología de forma exponencial en el campo de la medicina. Ahora se cuenta con telepresencia, realidad virtual, hologramas, robots con tacto, realidad aumentada, entre otros.

Aunque la medicina ha evolucionado rápidamente, aún existen limitantes e inconvenientes, como la disminución en la relación paciente-doctor por no existir contacto real entre ellos. En este punto también entra la confianza del paciente hacia el doctor, y se refleja en la preocupación de la privacidad y confidencialidad de la información, como la filmación de algunos procedimientos y su uso subsiguiente con propósitos educativos [22].

También se disminuye la relación entre los profesionales de la salud, este tiene el mismo efecto que las redes sociales, al pasar la mayoría “conectados” con el mundo nos alejamos del medio, eliminando así las relaciones personales.

La dificultad en el desarrollo de protocolos puede ser una fuerte limitante, y debería existir un mayor control de cómo se realiza una cita médica mediante la telemedicina.



1.3.4. Retos de la telemedicina

Con la telemedicina se desea alcanzar a:

- Tener servicios complementarios e instantáneos a la atención de un especialista (obtención de una segunda opinión).
- Tener servicios complementarios e instantáneos a la atención de un especialista (obtención de una segunda opinión).
- Educar de forma remota para los estudiantes de las escuelas de enfermería y medicina, y otras áreas de salud.
- Información electrónica y servicios de archivo digital de exámenes radiológicos, ecografías y otros.

Todo esto se traduce en una disminución del tiempo entre la toma de exámenes y la obtención de resultados, o entre la atención y el diagnóstico certero del especialista, quien no debe viajar o el paciente no tiene que ir a examinarse, y reducir así costos de tiempo y dinero [22].

1.3.5. La telemedicina en el Ecuador

El Programa Nacional de Telemedicina/Telesalud, se enmarca dentro de los objetivos del Plan Nacional del Buen Vivir. Este plan tiene como objetivo principal mejorar la atención de salud a nivel hospitalario de segundo y tercer nivel por medio de las telecomunicaciones. Se contribuye a que la salud llegue a cualquier lugar del país y sin costo a toda la población ecuatoriana, mediante consultas clínicas y de especialidad a distancia. De esta forma se promueve a programas de gestión, capacitación, consulta bibliográfica, así como promoción, prevención, investigación e interculturalidad, para garantizar los principios de universalidad, equidad, calidad y eficiencia del Sistema en



su Red Pública Integral de Salud.

El Programa Nacional de Telemedicina/Telesalud. En el año 2009 se llevó a cabo gracias al liderazgo del Ministerio de Salud Pública (MSP), a través del Proceso de Ciencia y Tecnología (PCYT) y la decidida participación y cooperación de diversas instituciones públicas y privadas; viabilizando la propuesta mediante el desarrollo de proyectos que escalonadamente darán cobertura a las veinte y cuatro provincias del país y con la suscripción de convenios interinstitucionales entre el MSP, MINTEL, Secretaría Nacional de Planificación y Desarrollo (SENPLADES), Secretaría Nacional de Telecomunicaciones (SENATEL), Fuerzas Armadas del Ecuador (FAE), Universidades, entre otras.

De esa manera, se aplican las tecnologías de la información y las comunicaciones (TICs), logrando así optimizar los recursos existentes y logrando mayor competitividad con los demás países. La telemedicina es un nuevo concepto que ha encontrado su desarrollo con el progreso alcanzado en las tecnologías de la información y el conocimiento, y debe ser conocido por los profesionales de la salud.

Para el año 2010 se inicia el proyecto “Expansión del Programa de Telemedicina a Nivel Nacional”, y tiene un tiempo de ejecución hasta el 2015. El objetivo de este proyecto es tener un mayor alcance en la telemedicina específicamente en la región amazónica, y centros de mando y apoyo en la región sierra. También la de organizar los servicios médicos hospitalarios y ambulatorios a través de sistemas de comunicación. En el Ecuador hay una gran demanda de servicios de salud, eso produce demora en la atención y en la dotación de servicios



médicos, sobre todo en las zonas alejadas. La solución a este problema es la telemedicina, ayudando a que existan en dichas zonas consultas clínicas a distancia, programas de gestión, promoción, prevención, investigación e interculturalidad [23].

Las metas a cumplir por el Ministerio de Salud con este proyecto son las siguientes:

- Veinte unidades de salud incorporadas al Programa Nacional Telemedicina/Telesalud al año 2015.
- Ocho unidades de salud equipadas con equipo informático y con equipos de Telemedicina fija y móvil instalados al 2015. (Proyecto Expansión del Programa de Telemedicina a Nivel Nacional).
- Veinte unidades de salud con conectividad a Internet hasta el año 2015.

1.3.6. Cronología de hitos importantes de la telemedicina a nivel mundial

Los hitos importantes se detallan a continuación¹:

- **1900:** Intentos para desarrollar equipos, en Australia, para transmitir radiografías a través del telégrafo.
- **1924:** Aparece en la revista Radio News, un artículo titulado "Doctor por Radio", el cual abarcó la portada y se describe el esquema de la circuitería necesaria para lograrlo.
- **1950:** Científicos de la NASA desarrollaron un sistema de asistencia médica, que les permitía vigilar constantemente las funciones fisiológicas de los astronautas en el espacio.

¹La cronología es tomada desde [24].



- **1951:** Primera demostración que abarca varios Estados de los Estados Unidos, usando líneas dedicadas y estudios de televisión.
- **1955:** En Montreal, el Dr. Albert Jutras realiza teleradiología, a fin de evitar las altas dosis de radiación que incidían en las fluoroscopías. Se hizo uso de un interfono convencional.
- **1959:** Se consiguen transmitir, por primera vez, imágenes radiológicas a través de la línea telefónica.
- **1959:** En Nebraska, Cecil Wittson comienza sus primeros cursos de teleeducación y telesiquiatría, entre su Hospital y el del Estado, en Norfolk, Virginia, a 180 km de distancia.
- **1971:** Se inicia la era de los satélites, en especial el ATS (lanzado en 1966), con el fin de mejorar las prestaciones de una comunidad de nativos de Alaska.
- **1972:** Inicio de STARPAHC, programa de asistencia médica para nativos de Papago, Arizona. Se realizó electrocardiografía y radiología, y se transmitió por medio de microondas.
- **1975:** Finaliza STARPAHC, el cual fue adaptado de un programa de atención médica para astronautas por la compañía Lockheed.
- **1986:** Se realiza, en Noruega, la primera videoconferencia entre médicos.
- **1988:** La Nasa lanza el programa Space Bridge, a fin de colaborar con Armenia y Ufa (en esa época pertenecientes a la Unión Soviética). Armenia fue devastada por un terremoto. Las conexiones se hicieron usando video en una dirección y voz y fax bidireccionales, entre el centro médico de Yereván, Armenia, y cuatro



hospitales en los Estados Unidos. Se extendió posteriormente el programa a la ciudad de Ufá, para socorrer a los quemados en un terrible accidente de tren.

- **1991:** En la cátedra UNESCO de telemedicina, Catai, se realiza la primera cuantificación de *ADN* a distancia en el mundo, aplicado al análisis de imagen de factores pronósticos en el cáncer de mama.
- **1995:** La clínica Mayo pone en marcha una conexión permanente con el hospital Real de Ammán, en Jordania. Se realizan consultas diarias entre un médico hachemita y otros de los Estados Unidos. El médico hachemita presentaba, como si se tratase de una sesión clínica del hospital, a los pacientes, de forma sucesiva, en directo a los médicos americanos, quienes preguntaban o pedían al médico jordano que indagara, a su vez, en los enfermos, sobre sus dolencias. En otros casos, eran interpretaciones de radiografías o problemas dermatológicos.
- **2001:** Un doctor, en New York, elimina la vesícula enferma de un paciente de 68 años en Estrasburgo, Francia, por medio de un brazo robot. Se observa el vínculo de la Telemedicina con incipientes avances de la Robótica.
- **2003:** Comienza el proyecto de telemedicina en la Antártica (Proyecto Argonauta), dirigido por la Universidad de Chile.

1.4. Sistemas de telemedicina en gastroenterología

Según [25], los sistemas de telemedicina varían dependiendo de la tecnología utilizada, del lugar geográfico a aplicar, y hasta de la si-



tuación económica y política del país a utilizar. Por lo anterior, según este artículo en el mundo de la telemedicina se divide en siete principios de desarrollo de sistemas de telemedicina, los mismos que deben cumplirse para la telemedicina en gastroenterología. En la siguiente sección se explican los principios mencionados en esta revista.

1. Los sistemas en telemedicina deben desarrollarse pragmáticamente, buscando que, tanto las necesidades de los profesionales en salud y de la población de un área geográfica en especial, se solventen con el desarrollo de sistemas de telemedicina. Para esto, se debe ser más sensible en el proceso de identificar al personal médico y clínico interesado en expandir los servicios en salud a través de la telemedicina, brindarles las herramientas para su ejecución y convertirlos en los impulsores. Adicionalmente, se debe analizar con profundidad los beneficios o aplicaciones reales que deben brindar los sistemas de telemedicina, empezando con sistemas de beneficios claros a una gran cantidad de profesionales en salud y progresivamente beneficiar áreas específicas.
2. Los sistemas deben ser manejados y poseídos por los profesionales en salud. Ha sido importante para el desarrollo de la telemedicina, la valoración del personal médico como la base para la proliferación de su imagen e importancia. Adicionalmente, estos sistemas deben ser adquiridos y controlados por ellos mismos, buscando el mejoramiento continuo, valoración por parte de la población y mayor efectividad.
3. La administración y el soporte de los sistemas deben seguir las buenas prácticas administrativas. La utilización de la telemedicina no debe contar con intensos controles burocráticos por parte de los centros hospitalarios.



4. La tecnología debe ser lo más amigable posible. Los sistemas de telemedicina deben ser desarrollados para integrarse a los ambientes de trabajo, no ser divisiones de éste. El futuro en el desarrollo de sistemas en telemedicina está enfocado al uso del Internet y tecnologías en información que faciliten el ingreso y la interacción con el personal médico.
5. Los usuarios de los sistemas de telemedicina deben estar altamente entrenados y soportados, técnicamente y profesionalmente. Para esto es necesario la adecuada presentación y capacitación, en donde el personal pueda reconocer y estudiar el equipamiento utilizado, observar y analizar su funcionamiento, identificar sus beneficios y limitantes. Una vez el personal médico ve a la telemedicina como un componente de su ambiente de trabajo, la utilización de ésta se volverá una constante.
6. Los sistemas de telemedicina deben ser evaluados y sostenidos clínicamente y económicamente. En la actualidad estos sistemas deben ser analizados desde el punto de vista del costo beneficio, evaluando todos los resultados tanto económicos como clínicos provenientes del sistema, dentro de un marco de tiempo razonable haciendo uso apropiado de la inversión final. El desarrollo de lo anterior brinda a la telemedicina ser sostenible financieramente, favoreciendo su utilización en el tiempo y asegurando la financiación de la tecnología, el personal implicado y el soporte técnico.
7. La información sobre el desarrollo de sistemas en telemedicina debe ser compartida. Los diversos estudios en el área y los beneficios reales y potenciales de la telemedicina han impulsado la creación de páginas de internet, organizaciones e instituciones en-



focados en el desarrollo de programas complejos en telemedicina; igualmente la literatura en el área ha tenido un avance significativo, sumado con una mejor divulgación entre diversas revistas generales y especializadas.

Con la tecnología que tiene ya un gran avance, gran cantidad de especialidades médicas se han beneficiado con sistemas en telemedicina, estos sistemas pueden ser desde la asistencia de pacientes a grandes distancias hasta el intercambio de información entre médicos especialistas. Se mencionan varios sistemas [25]:

Tele-diagnosis/consulta: Consulta y diagnóstico a distancia de un paciente por un médico mediante video.

Tele-seguimiento: Seguimiento a distancia de la evolución del paciente. Este servicio es adecuado para pacientes de tercera edad, dificultad del paciente a transportarse, o pacientes crónicos.

Tele-educación (e-learning): Forma mediante la cual varios médicos pueden discutir un caso clínico, evitando viajes innecesarios y que estudiantes puedan observar en directo como se realiza una determinada operación como si estuvieran en la sala de operaciones, al mismo tiempo que tienen las explicaciones sobre la intervención. Con ello está garantizado que los estudiantes reciban el contenido por especialistas de un alto nivel científico y metodológico.

Teleendoscopía: Se desarrolla bajo altos estándares de tecnologías en telecomunicación. La calidad de las imágenes, videos y la velocidad de transmisión, son factores críticos en su desarrollo.



Hay una gran variedad de sistemas de telemedicina que se pueden mencionar como la teleoftalmología, la telepatología, telepediatría, telepsiquiatría, teleoncología, entre otros. Aunque no son de interés para el tema de tesis, se las menciona por su grado de importancia en la telemedicina.

1.5. Planteamiento del problema

En la actualidad, los equipos médicos son una herramienta esencial en el área de la salud, tanto para diagnóstico como para ayuda terapéutica. Debido a la gran cantidad de dispositivos, se clasifican en equipos de diagnóstico y terapéuticos.

Los equipos de diagnóstico son los que dan información del estado del paciente mediante estudios e imágenes que podrían ser procesadas en forma automática e interpretadas por un profesional de la medicina. Entre estos equipos de diagnóstico, existen dispositivos médicos centrados en la adquisición de información del aparato digestivo. Tales instrumentos son de gran ayuda en la especialidad de gastroenterología. Estos dispositivos han evolucionado hasta ser incorporados en un sistema con toda la gestión de los estudios realizados a los pacientes, incluyendo el procesamiento de señales e imágenes, el diagnóstico, tratamiento y el seguimiento del paciente.

Estos equipos modernos tienen costos elevados en comparación con equipos anteriores o desactualizados, los cuales carecen de capacidades de almacenamiento y gestión de la información.

Los médicos especialistas que realizan este tipo de estudios necesitan tener la información de sus pacientes siempre disponible y por tanto,



se ven obligados a adquirir nuevos equipos que les facilite el trabajo, u optar por la instalación de un sistema de hardware y software que permita al equipo realizar la adquisición de imágenes, videos y gestionar a través del mismo la información clínica de los pacientes.

Frente a la problemática descrita, se desarrolla el sistema **SAPHIENS** con componentes de hardware y software que pueda ser incorporado fácilmente a los equipos médicos existentes, permitiendo adicionalmente la gestión del registro médico del paciente en formato electrónico. El sistema a desarrollar, brinda al médico la posibilidad de subir los historiales clínicos de los pacientes a un servidor web para compartir dicha información.

Médicos especialistas y/o estudiantes ubicados en otras localidades pueden analizar estos historiales permitiendo así realizar un estudio del caso para dar el diagnóstico y tratamiento de manera óptima y rápida. De esta manera se desarrolla un sistema telemédico en gastroenterología, para lo cual es necesario diseñar un registro de historias clínicas del paciente [26].

1.6. Objetivos generales y específicos

1.6.1. Objetivo general

Desarrollar e implementar un sistema que permita gestionar la información capturada desde de un equipo de gastroenterología, de tal manera que el médico especialista gestione de manera efectiva la información, brindando así un mejor servicio a sus pacientes.



1.6.2. Objetivos específicos

1. Desarrollar un dispositivo que sirva de interfaz entre el endoscopio y el computador a fin de poder extraer partes del video que sean de interés para el especialista, para diagnóstico, seguimiento y tratamiento del paciente.
2. Desarrollar un software que permita procesar las imágenes extraídas del examen de gastroenterología con el endoscopio para fines de planificación y seguimiento de biopsias realizadas al paciente.
3. Crear un banco de historias clínicas, en la que permita a cualquier usuario con autorización, acceder a la información desde cualquier localidad mediante la web.

1.7. Metodología a utilizar

La metodología para el desarrollo de este proyecto comprende de las siguientes fases:

- **Adquisición de datos:** Diseñar el dispositivo electrónico (hardware) que permita adquirir las imágenes y videos de forma digital a partir del equipo de gastroenterología. Dicha información se la transfiere al ordenador vía USB para su respectivo tratamiento.
- **Almacenamiento de Datos:** Diseño y creación de la base de datos dentro de un servidor local y remoto.
- **Software del sistema *SAPHIENS*:** Diseño y creación de la interfaz gráfica para la visualización y gestión de la información.

Capítulo 2

Adquisición de los datos

En este capítulo se plantea el desarrollo de la interfaz hombre-máquina, capaz de obtener y digitalizar las imágenes y videos de un estudio realizado y enviar las mismas al ordenador del médico especialista.

Se describe las salidas y tipos de señal de video que se obtiene de los equipos endoscópicos. La digitalización de la señal de video analógica mediante los dispositivos capturadores de video (*encore* y *easycap*), la electrónica interna de estos dispositivos. Finalmente el ensamblaje de la interfaz hombre-máquina de **SAPHIENS**.

De acuerdo a lo mencionado se mostrará paso a paso el desarrollo de la interfaz hombre-máquina.

2.1. Interfaz hombre-máquina para clasificar las imágenes de interés

En la Figura 2.1 se observa el esquema de conexión entre el equipo gastroenterológico y el ordenador.

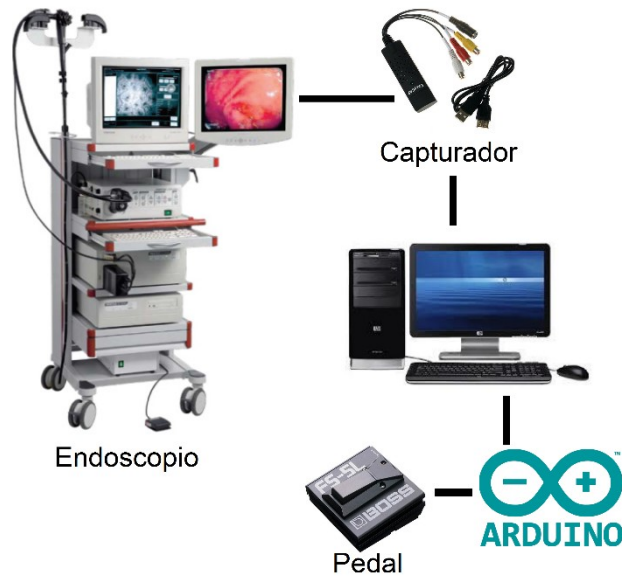


Figura 2.1: Interfaz hombre máquina. Elaboración Propia

En la Figura 2.1 se observa cinco dispositivos. El dispositivo endoscopio proporciona una señal de video analógica la cual se convierte en una señal digital mediante el capturador de video, que a su vez lo envía al ordenador. El módulo arduino controla el disparo para la captura de video e imágenes y la comunicación serial. El disparo de captura se realiza mediante un dispositivo pedal. Durante el diseño del hardware debe tomarse en cuenta la facilidad de uso para el médico especialista y la interfaz de comunicación entre el equipo médico y el ordenador.

2.1.1. Salidas de video del equipo endoscópico

En el mercado existen varias marcas de equipos endoscópicos entre estas se tiene: *Mediglobe*, *Linvatec*, *Europe Division Sales*, *Laborie Belgium*, *Evis*, *Olympus*, entre otros. **SAPHIENS** fue probado con un equipo de la marca *Olympus*. Este se observa en la Figura 2.2.

EVIS EXERA III OLYMPUS



Figura 2.2: Equipo endoscopio Olympus [2]

La mayoría de los equipos endoscópicos cuentan con varias salidas de video, una de estas se conecta al monitor del equipo, mediante el cual el médico especialista puede observar el estado del paciente. Se utiliza una de estas salidas para obtener la señal analógica de video y realizar un tratamiento de señal sobre esta.

En la Figura 2.3 se observa las salidas de video que tiene el dispositivo médico, al ser salidas analógicas se pueden adaptar a un cable *VGA*, utilizando un conector *VGA* o adaptándolo a un conector de salida de video *Y/C* o *Compuesto* dentro del dispositivo médico [27]. De esta manera se obtiene el video del dispositivo gastroenterológico.

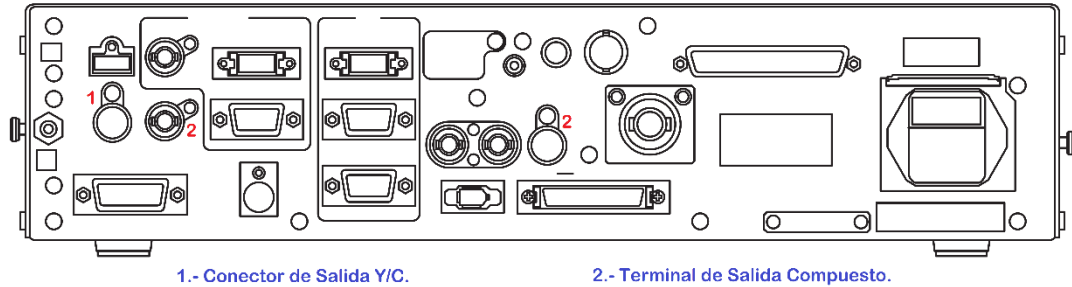


Figura 2.3: Conectores de salida de video. Elaboración propia

A continuación, se describirán las salidas de video con las que cuenta el equipo.

2.1.1.1. Conector de salida Y/C

Esta es una salida de señal analógica conocida como *S – video* o *Y/C – video*, nombrado de esta manera por transmitir información de la luminancia (*Y*), y la crominancia (*C*). Esta señal analógica transmite únicamente el video, para obtener el audio del dispositivo se requiere un cable de audio, por esta característica la señal de video es de una mejor calidad respecto a la del Video Compuesto cuya señal transmite el audio y video a la par [27]. En la Figura 2.4 se puede observar el conector de salida Y/C.



Figura 2.4: Conector de salida Y/C. Elaboración propia

2.1.1.2. Terminal de salida de video compuesto

De manera similar que la señal analógica Y/C este tipo de señal transmite la información de la iluminancia y la crominancia, con la diferencia de que en la señal compuesta se comprime toda la información que se tiene dentro de una señal Y/C . Por este motivo la calidad de video es menor. En la Figura 2.5 se puede observar el conector de video compuesto.

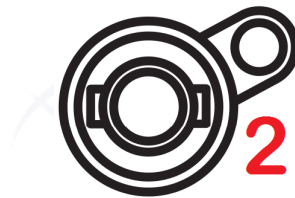


Figura 2.5: Conector de video compuesto. Elaboración propia

2.2. Dispositivo capturador de video

El dispositivo capturador, permite transformar la señal analógica de video en una señal digital para entender de mejor manera esta etapa se explicará la conversión analógica-digital.

2.2.1. Convertidor analógico-digital (ADC)

Una señal analógica es continua en el tiempo al contrario de una señal digital que es discreta en el tiempo, una señal analógica lleva en si cada parte de la información con la cual fue creada, en cambio una señal digital se encuentra formada por la mayor cantidad de partes de dicha información.

Para realizar la conversión de una señal analógica en una señal digital, se deben seguir dos etapas: *cuantificación* y *muestreo* [28].



En la primera etapa se muestrea la señal analógica y a cada valor obtenido se asigna un valor que depende del número de bits del *ADC*. El valor cuantificado se codifica en binario, que de igual manera depende del número de bits del *ADC* [29]. A la cantidad de bits del *ADC* se le conoce como resolución, la mayoría de capturadores de video tienen una resolución de 8 bits con arquitecturas de memoria flash [28] [30]. La frecuencia de muestreo para el proceso se realiza de acuerdo con la tasa de Nyquist la cual nos dice que como mínimo debe ser el doble de la frecuencia máxima de la señal analógica de entrada para la reproducción correcta de la señal de entrada [28].

A menudo se define la resolución de la cuantización a partir del margen de entrada del convertidor, mediante la Ecuación 2.1a. Podemos definir el bit menos significativo dependiendo del número de bits de resolución [3].

$$1LSB = q = (Margen)/(2^N) \quad (2.1a)$$

En la Figura 2.6 se muestra la gráfica de un convertidor de 3 bits, la frecuencia de la señal entrante es de $1KHz$, con un voltaje medio de $5V$, y de $10V$ de cresta a cresta. Al tratarse de convertidor con una resolución de 3 bits, tenemos $2^N = 2^3 = 8$, es decir un total de 8 estados. En código binario esto representa los valores desde 000 hasta 111. Para saber intervalo de cuantización usamos la Ecuación 2.1a. Con lo que tenemos [3]:

$$q = 10/8 = 1,25V \quad (2.2a)$$

De esta manera se digitalizará la señal de video para ser tratado

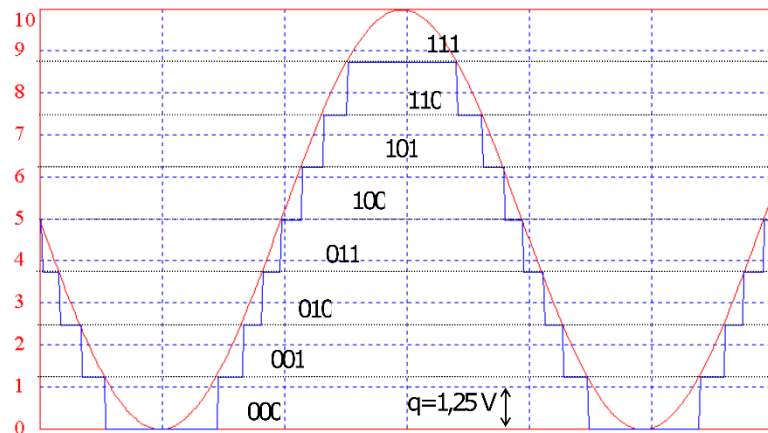


Figura 2.6: Digitalización de una señal a partir de un convertidor de 3 bits [3]

dentro de un ordenador.

Los capturadores ordinarios pueden capturar audio y video, con una resolución de 720 x 480 a 30 *fps* en el estándar NTSC (National Television System Committee) [31]. En el estándar PAL (Phase Alternating Line) se tiene 720 x 576 a 25*fps* [32].

2.2.2. Adquisición del equipo capturador de video

El primer proceso a realizar es la obtención de la señal de video, para esto se debe escoger un dispositivo compatible con la plataforma a usar en este caso Windows, además la señal digital de video debe tener una resolución adecuada con la finalidad de visualizar el video lo más claro posible.

Dentro de los equipos capturadores de video se usaron dos dispositivos que se tienen dentro del país estos son el dispositivo capturador Encore y el dispositivo capturador EasyCap.

Cada uno de estos dispositivos tienen características similares,



estas son [33]:

- Conexión USB 2.0.
- Compatible con los estándares de video NTSC y PAL.
- Entrada de video: RCA compuesto, S-Video.
- Entrada de audio: Estéreo (RCA).
- Resolución NTSC: 720 x 480 a 30fps.
- Resolución PAL: 720 x 576 a 25fps.

2.2.3. Estándar de video NTSC

El Comité Nacional de Estándares de Televisión (NTSC) trató diversas especificaciones técnicas de reproducción y recepción de la imagen, esto garantizó la compatibilidad entre todas las cámaras de televisión y el aparato de TV en todo el país y con las de cualquier otra nación que comparte el sistema NTSC: Japón, México, Canadá y gran parte de América del Sur. A principios de la década de 1950 NTSC cambió estas normas ligeramente para acomodar la adición de color a la señal de televisión [4].

Mientras que la grabación de una película funciona a una velocidad de 24 cuadros por segundo, la velocidad de fotogramas de vídeo NTSC es de 30 fotogramas por segundo. La diferencia principal, sin embargo, es cómo se crean y reproducen esas imágenes. Las películas se realizan por medio de un proceso mecánico y con fotoquímica, pero el estándar NTSC utiliza un proceso electrónico conocido como exploración entrelazada.

Para realizar este proceso el estándar NTSC utiliza un chip CCD (dispositivo de carga acoplada) Figura 2.7 el chip CCD es un sensor de estado sólido (medición ya sea de 1/4", 1/3", 1/2", o 2/3" de diámetro) compuesto por cientos de miles a millones de fotodiodos sensibles a la luz llamados píxeles, que están unidos entre sí (acoplados) y alineados en filas horizontales estrechas [4]. Como con la película, la creación de la imagen comienza con la lente de la recolección de la luz de la escena y de enfoque en el plano de imagen. En el estándar NTSC, el plano de la imagen, o de destino, es la cara del chip CCD [4].

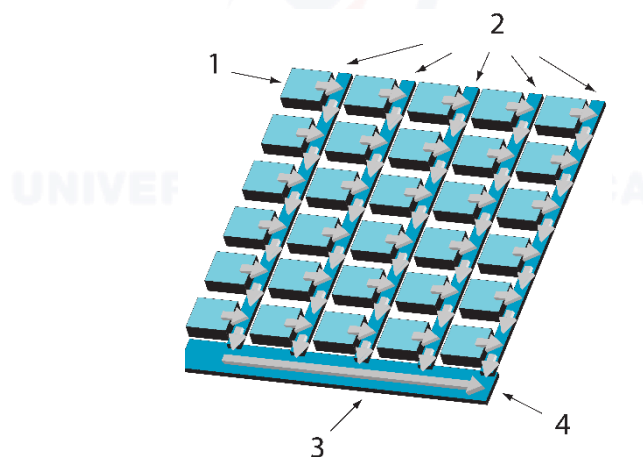


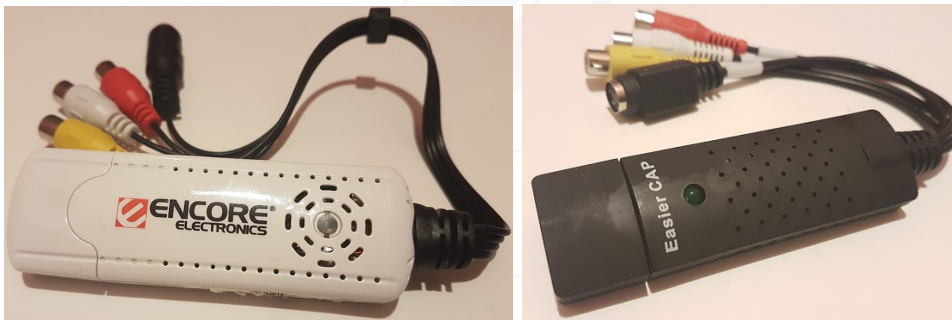
Figura 2.7: CCD (dispositivo – chip de carga acoplada). (1) Píxeles individuales recogen la información de brillo. (2) Registros verticales que llevan esta información al (3) registro horizontal; a continuación, envía la señal de video en bruto (4) a un amplificador de salida. Los píxeles se leen una fila a la vez en el área de transferencia de carga horizontal [4]

2.2.4. Estándar de video PAL

PAL (Phase Alternation Line - Línea de Fase Alternada), este estándar se desarrolló en 1960 y es mayormente usado en el continente europeo, así como en países asiáticos, africanos y algunos países americanos y en Australia. *PAL* es un sistema de codificación que se utiliza en la transmisión analógica de señales

de televisión a color, este estándar tiene una resolución de 720 x 576 píxeles, y 25 fps (frames por segundo), es decir emite 625 líneas a través de una serie de ráfagas producidas por electrones sobre la pantalla del televisor. El estándar *PAL* es completamente incompatible con el estándar NTSC que se estudió con anterioridad, por las características antes mencionadas [34].

En las Figuras 2.8a y 2.8b, se muestran los dispositivos capturadores de video *encore* y *easy cap*, respectivamente.



(a) Capturador de video Encore

(b) Capturador de video EasyCap

Figura 2.8: Dispositivos capturadores de video. Elaboración propia

2.2.5. Elementos electrónicos del dispositivo capturador de video

En la Figura 2.9 se puede observar la electrónica interna de los dispositivos capturadores de video *EasyCap* y *Encore*.

La Figura 2.9 muestra que la electrónica interna de ambos capturadores de video es similar, a continuación, se describe cada uno de los componentes:

1. **Chip UTV007** [31].- Este chip contiene el procesador de entrada, el controlador para el tratamiento de señal, y el

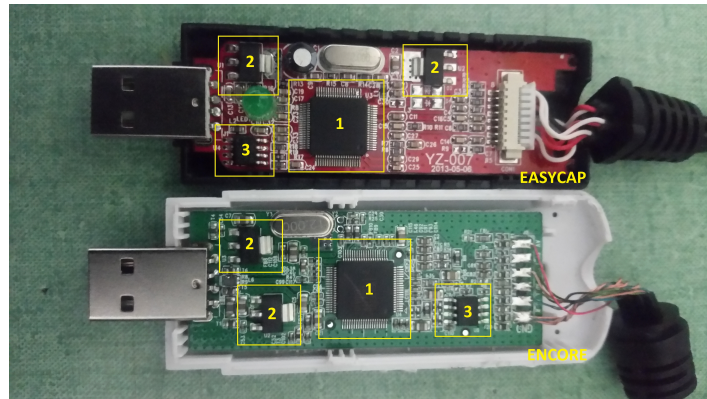


Figura 2.9: Electrónica Interna de los dispositivos capturadores de video. Elaboración propia

decodificador de vídeo. Contiene además el capturador de audio y el controlador para la salida USB.

- Se sincroniza perfectamente el video en tiempo real.
- Funciona de manera correcta con señales débiles.
- El chip es compatible con todos los filtros para el brillo, contraste, tono, saturación y nitidez.
- El ajuste de nitidez es ideal para reducir el ruido, con alguna pérdida de nitidez.
- Contiene una entrada de audio estéreo, de modo que cualquier audio grabado es siempre sincronizado con el vídeo.
- El chip puede trabajar con el formato S-video y Video Compuesto.

2. **Regulador de voltaje AMS1117/Sa1117bh-1.8 [35].-** La serie de reguladores de voltaje ajustable y fijos AMS1117, están diseñados para proporcionar una corriente de salida de hasta 1A y operar hasta con 1V diferencial de entrada a salida. La tensión de abandono del dispositivo se garantiza



como máxima de $1.3V$, ver Anexo ??.

Características

- Voltajes ajustables o fijos de $1.5V$, $1.8V$, $2.5V$, $2.85V$, $3.3V$ y $5.0V$.
- Corriente de salida de $1A$.
- Opera con un voltaje de abonado de hasta $1V$.

Aplicaciones

- Cargadores de baterías.
- Regulador Lineal de $5V$ a $3,3V$.
- Reguladores de voltaje conmutados.
- Reguladores lineales de alta eficiencia.

3. **Memoria eprom serie M24C02 [36].-** Es un dispositivo de memoria programable y borrable eléctricamente compatible con I2C (*EEPROM*), ver Anexo ??.

2.3. Función del módulo arduino

La función del *Arduino* en *SAPHIENS* es de detectar el momento en que es presionado el pedal, dependiendo del número de veces que se ha presionado, el *Arduino* responderá comunicándose con el software de manera serial y enviando la cadena de caracteres “*imagen*” si se ha presionado una sola vez, o envía “*video*” se ha sido presionado dos veces el pedal. La manera en que está programado el *Arduino* da la posibilidad de iniciar con la grabación de un video, y mientras se sigue grabando poder

tomar fotografías.

Teniendo la idea clara de lo que se desea crear, se inicia con el diseño de la parte física del dispositivo **SAPHIENS**. Se utiliza el pin 2 del *Arduino* para conectar el pedal, utilizando cable multipilar se hace una conexión *Pull Up*. Se utiliza el pin 2 ya que este se lo puede configurar para que sea de entrada o de salida digital.

Una conexión *Pull Up* nos ayuda para que no existan falsos estados, es decir, que cuando el pedal no está presionado llegue un voltaje de 5V (*Estado alto*) al pin 2 del *Arduino*, y cuando se presione llegue un voltaje de 0V (*Estado bajo*). Como detalle, se activó el pin 13 y se conectó un led para mejorar la presentación y darle un mejor estilo Figura 2.10.



Figura 2.10: Presentación del armazón del dispositivo físico de **SAPHIENS**. Elaboración propia

En la Figura 2.11 se puede ver el resultado de la conexión *pull up* entre el *Arduino* y el pedal que está simbolizado por el pulsante. Se puede observar en la Figura 2.11 que el pin dos del *Arduino* está conectado a la resistencia de *10 Kohms* y al pedal (repre-

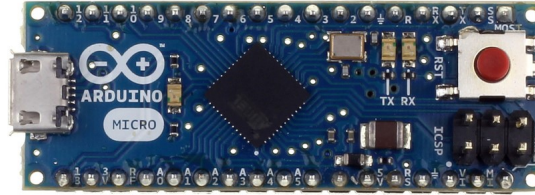


Figura 2.12: Arduino micro [6]

2.3.1. Programa para el control del disparo de captura de video

El software del dispositivo es un lenguaje estándar y un firmware que se ejecuta en el dispositivo, usa un lenguaje que simplifica $C++$, en un IDE basado en el procesamiento. El dispositivo es compatible con las plataformas *Flash*, *Processing*, *Max MSP*, y *MATLAB* [37].

Para realizar el programa de una manera sencilla se utilizó una librería externa (*ClickButton.h*) que contiene métodos pre programados realizados para el control de pulsantes, de esta manera se puede controlar el tiempo de pulso y los rebotes ocasionados por el resorte de los pulsantes normales. Otra de las ventajas del Arduino Uno, es que viene con una librería interna para la comunicación serial lo que simplifica totalmente la configuración del *ICSP* (Programación Serial en Circuito) dejando únicamente los métodos de lectura y escritura para la comunicación serial.

El Dispositivo Arduino funciona conjuntamente con el dispositivo pedal, al recibir dos pulsos seguidos el programa enviará mediante comunicación serial la palabra video, al recibir un pulso dentro de un tiempo de mayor duración enviará la palabra



“Foto”.

```
#include "ClickButton.h"
//Incluye una libreria externa para el control de pulsantes
const int buttonPin1 = 2;
//Agrega el pin dos para ser usado con el pulsante
ClickButton button1(buttonPin1, LOW, CLICKBTN_PULLUP);
//Configura el pulsante como PULLUP
int numclicks = 0;
//Contador de pulsos.
void setup()
{
  pinMode(13, OUTPUT);
  //Configura el pin 13 como salida.
  digitalWrite(13, HIGH);
  Serial.begin(9600);
  //Inicio de la comunicación serial.
  button1.debounceTime = 20;
  // Temporizador de supresión de rebotes en ms
  button1.multiclickTime = 250;
  // Límite de tiempo para los pulsos
  button1.longClickTime = 1000;
  // Tiempo hasta estabilizar el puslo.
}
void loop()
{
  // Actualiza el estado del pulsante
  button1.Update();
  if (button1.clicks != 0) {
    numclicks = button1.clicks;
  }else{
    numclicks = 0;
  }
  if(numclicks==1){
    Serial.print("Foto");
  }
  if(numclicks==2){
    Serial.print("Video");
  }
}
```

```
}  
}
```

Se puede ver que se inicia el código agregando una librería para el control del pulsante, esto facilita para conocer el número de veces que se ha pulsado el pedal, para ello se configura el pin 2 como pullup para ser usado como conexión con dicho pedal; Se agrega la variable 'numclicks' donde se guardará el número de veces que se presione el pedal a la vez. En el método setup, se coloca que el pin 13 este activo, esto significa que habrá un voltaje de 5V a la salida de este pin y se utiliza para colocar un led para visualización. Se inicia la comunicación serial y se da tiempos para que cada determinado momento lea el puerto, y para la lectura del pedal. Ya todo listo se hace el metodo loop que es el que se repite mientras está encendido el *Arduino*, aquí se hace la condición de que si fue presionado el pedal una vez que envíe de manera serial la información "Foto", y si se presiona 2 veces que envíe "Video", caso contrario no envíe nada. En el código se encuentra comentado para su mayor entendimiento.

2.3.2. Dispositivo pedal

El pedal usado para el diseño del dispositivo hardware es el FS-5U se puede observar en la Figura 2.13, este es un pedal diseñado para controlar el cambio de frecuencia de una guitarra eléctrica. Es de construcción metálica por lo cual es resistente al uso. Su funcionalidad dentro de **SAPHIENS** es el control de disparo de la captura de las imágenes y videos.

La conexión entre este dispositivo y el arduino uno se realiza mediante un cable de audio blindado con plug de micrófono mono-estéreo como se observa en la Figura 2.14.



Figura 2.13: Dispositivo pedal [7]



Figura 2.14: Cable de audio-jack mono. Elaboración propia

2.4. Construcción y montaje del dispositivo

Para montar el dispositivo final, se decidió colocar cada uno de los dispositivos antes mencionados dentro de un armazón con la finalidad de tener un solo elemento de hardware y proteger cada uno de ellos, para esto se realizó el diseño del armazón en un software conocido como *Solidworks* [38]. Es un sistema de diseño CAD en 3D [39]. De esta manera se realizó el diseño de la caja. En el Anexo ?? se observa el diseño completo de todas las partes mencionadas en esta sección, a continuación, se describirá cada una de las partes de esta.

2.4.1. Parte superior

En la Figura 2.15 observamos la parte superior de la caja de montaje, en esta tapa, se tiene dos partes importantes, estas son las cejillas de acople, que sirven para fijar la caja junto con sus partes laterales y los agujeros para los tornillos, mediante los cuales se dará una mayor firmeza.

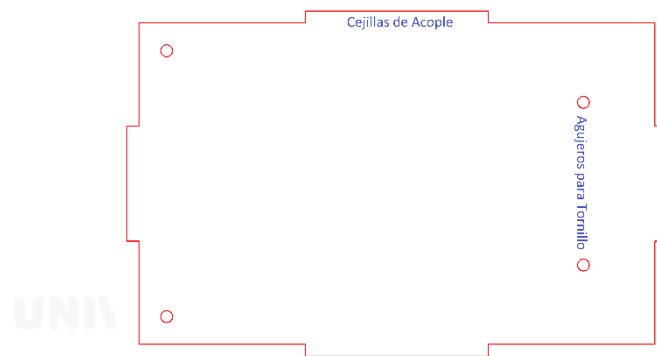


Figura 2.15: Parte superior de la caja de montaje del dispositivo. Elaboración propia

2.4.2. Parte inferior

Al igual que en la parte superior, en la parte inferior se tiene cejillas de acople y los agujeros para los tornillos, pero en la Figura 2.16, podemos observar que el diseño es distinto, con la finalidad de dar una mayor rigidez al momento del armado bajo este criterio las cejillas de la parte superior se acoplan de manera fácil, mientras que las cejillas de la parte inferior se acoplarán de una manera firme dando una mayor seguridad al equipo. Los agujeros de los tornillos de la parte inferior se colocan a distancias iguales que en la superior.

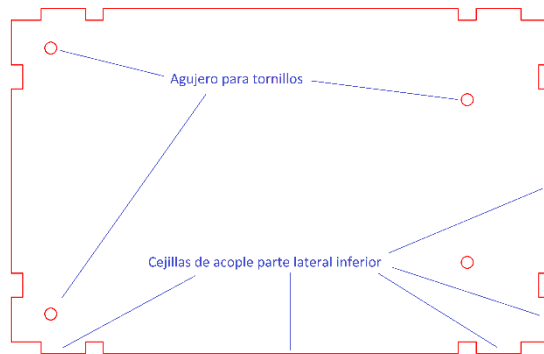


Figura 2.16: Parte inferior de la caja de montaje del dispositivo. Elaboración propia

2.4.3. Parte lateral

Para realizar un acople de las partes de una mejor manera la parte inferior está formada de las siguientes partes:

- Entrada superior de cejillas. - Para dar soporte a la tapa superior de la caja.
- Entrada inferior de cejillas. – Acople de seguridad para la parte inferior.
- Entrada de separación intermedia. - Esta parte permite la separación entre el dispositivo Arduino y el dispositivo capturador de video.
- Espacio para Tuerca. - Espacio para colocar la tuerca que permite de unión entre la tapa frontal y posterior que se explicaran posteriormente.
- Espacio para tornillo de acople. – Espacio para colocar el tornillo de cierre entre las tapas de la caja.

Cada una de estas partes se puede observar en la Figura 2.17.

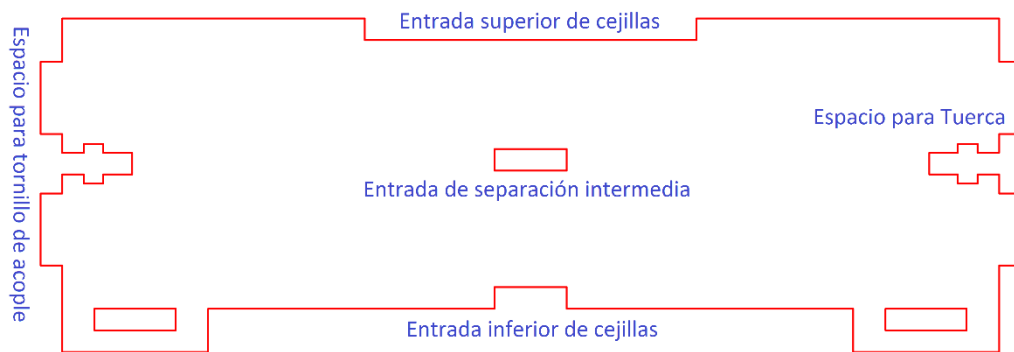


Figura 2.17: Parte lateral de la caja de montaje del dispositivo. Elaboración propia

2.4.4. Parte posterior

En la parte posterior se encuentran cada uno de los conectores de salida los mismos que se conectarán al computador. En este caso son los conectores USB del dispositivo capturador de video y del dispositivo arduino como se observa en la Figura 2.18. Cada una de estas salidas, fue diseñada para que se ajusten de manera perfecta con el dispositivo, de esta manera se asegura que el dispositivo no se mueva dentro de la caja y a partir de esto pueda dañarse con el tiempo.

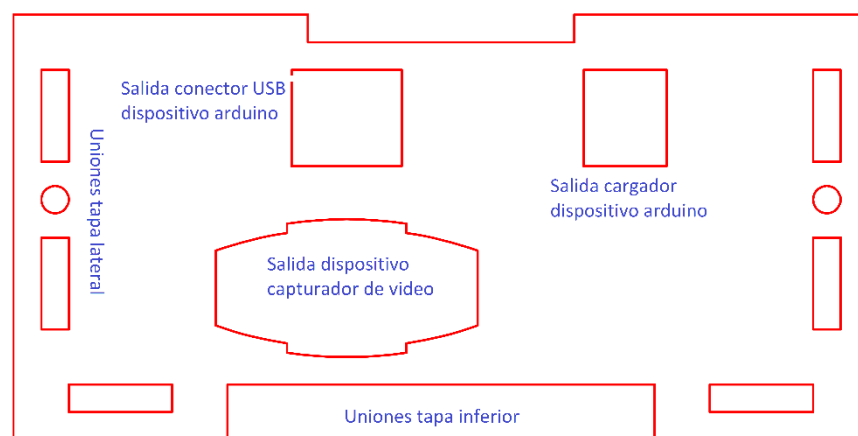


Figura 2.18: Parte posterior de la caja de montaje del dispositivo. Elaboración propia

2.4.5. Parte anterior

En la parte anterior de la caja se tiene las salidas de los conectores de audio y video del dispositivo encore y el plug de entrada de audio mediante el cual se conecta el dispositivo pedal como se observa en la Figura 2.19. Se debe tener en cuenta que dentro del proyecto simplemente se usa el conector de salida de video y el plug de audio debido a las necesidades del mismo.

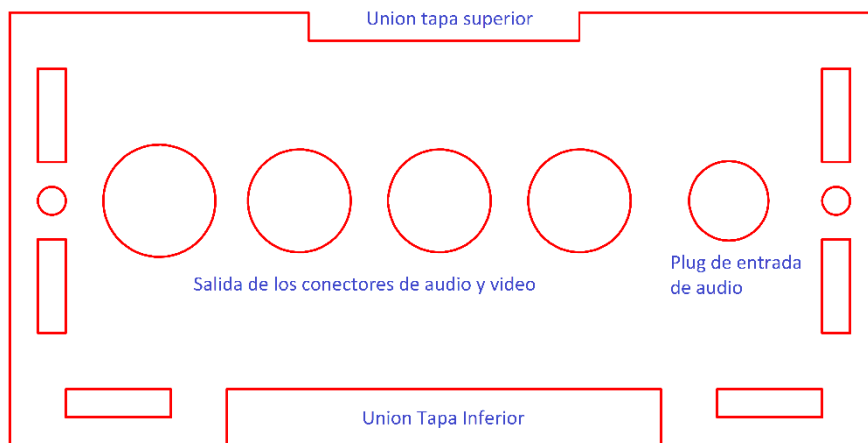


Figura 2.19: Parte anterior de la caja de montaje del dispositivo. Elaboración propia

2.5. Distribución interna de la parte física de *SAPHIENS*

Para la construcción de la parte física de SAPHIENS, se inicia con los cortes mediante láser, se utilizó el material MDF de 5 mm. Ya con las partes cortadas, se colocan los plugs de entrada tanto del capturador de video como el plug para el pedal, se muestra el resultado en la Figura 2.20a.

En el Arduino se coloca un led para indicar el encendido del dispositivo físico, en este caso se utilizó un led de color azul. Aquí



(a) Conexión de plug. Elaboración propia (b) Conexión del indicador de encendido. Elaboración propia

Figura 2.20: Montaje del dispositivo físico

se puede ver que la razón por utilizar el pin 13 para el led es porque el pin de tierra (GND) está a lado y así solo se lo debe colocar en el lugar especificado como está en la Figura 2.20b.

En el prototipo se coloca directamente el circuito pull up, y mediante cable multipar se conecta al plug del pedal. En la Figura ?? se muestra la conexión.



Figura 2.21: Montaje del dispositivo físico

Mediante las cejillas que tiene cada pieza del armazón del dispositivo, a la base se coloca uno de los lados, también se coloca la

pieza triangular que sirve como tope para el capturador de video. También se coloca en la base la pieza rectangular que sirve para que repose el capturador. En la Figura ?? se puede ver la unión de dos de los lados del armazón.

Ahora se coloca la primera pieza donde se conectó los plugs del capturador y el plug del pedal. Se puede observar en la Figura 2.22a que ya se tiene conectado a la caja el capturador y el arduino.



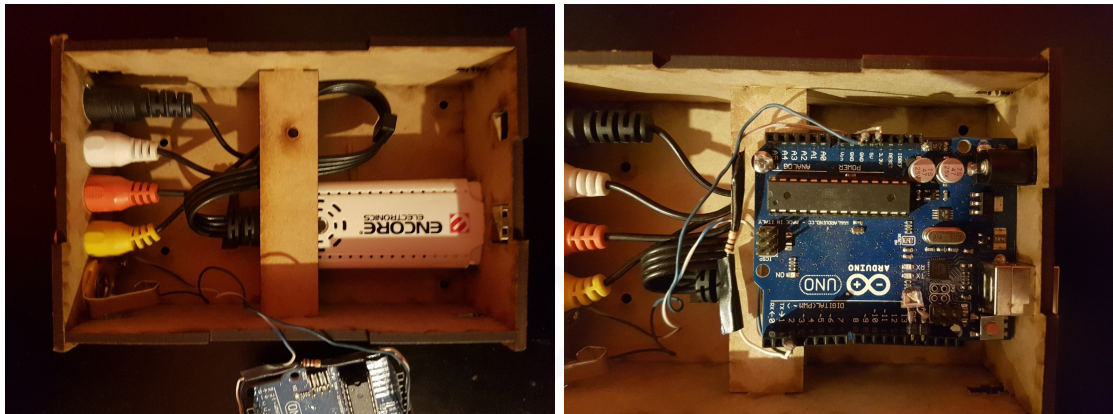
(a) Montaje del dispositivo capturador de video. Elaboración propia (b) Soportes para el dispositivo capturador de video. Elaboración propia

Figura 2.22: Montaje del dispositivo físico

Se coloca el lado donde irá la salida del puerto usb del capturador, al colocar este lado se puede ver en la Figura 2.22b el funcionamiento de la base y el triángulo colocado casi al inicio del ensamblaje.

Encima del capturador se coloca la otra pieza rectangular, esta cumple la función que ser la base donde descansará el Arduino. En la siguiente Figura 2.23a también muestra ya colocado un

lado más, mostrando así que las piezas aunque tengan mucho espacio, están aseguradas. La pieza rectangular tiene un agujero donde se asegurará el Arduino.



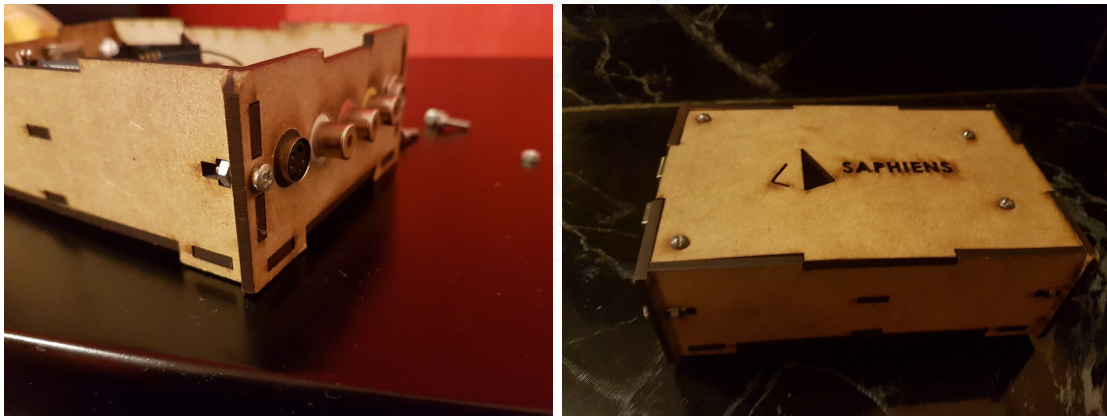
(a) Base del módulo arduino. Elaboración propia (b) Montaje del módulo arduino. Elaboración propia

Figura 2.23: Montaje del dispositivo físico

En la siguiente Figura 2.23b se puede observar que el Arduino va encima del capturador de video, y como se dijo anteriormente, se tiene el agujero en la pieza rectangular para colocar un tornillo y así que quede inmóvil las piezas del dispositivo.

Ahora para asegurar que el dispositivo *SAPHIENS* no pueda desarmarse de manera fácil, se colocan tornillos pequeños como se muestra en la Figura 2.24a.

Y como última parte se coloca la parte de encima de la caja, terminando así el ensamblaje del dispositivo físico de *SAPHIENS*, mostrándose el acabado en la Figura 2.24b.



(a) Aseguramiento final del armazón con pernos y tuercas. Elaboración propia

(b) Acabado final. Elaboración propia

Figura 2.24: Montaje del dispositivo físico

Capítulo 3

Sistema de telemedicina para gastroenterología

Dentro de este capítulo se describe el proceso de almacenamiento y visualización de los datos del paciente. Para el almacenamiento se utiliza el servidor de aplicaciones *WampServer* [40]. El cual contiene herramientas como: *MySQL*, *Apache*, *PHP* entre otros, la combinación de todas estas herramientas dentro de un solo sistema hace más fácil la creación, montaje y mantenimiento de las bases de datos.

Se describe la creación de la ficha electrónica del paciente, entendiéndose por esto el diseño y creación del software utilizado como interfaz gráfica de **SAPHIENS**, este software se desarrolló en el lenguaje *JAVA*.

Una de las principales características de este software, es el uso de *JAVAFX* [41]. Este producto permite crear aplicaciones para una amplia gama de dispositivos, se pueden crear aplicaciones móviles, aplicaciones web, de escritorio entre otras.

El desarrollo de la interfaz se realizó en conjunto con la herra-



mienta SceneBuilder, este software fue diseñado para la creación y ubicación de componentes, como botones, cajas de texto, etc. En conjunto con las hojas de estilo CSS, permiten dar una mejor apariencia al producto final. Las hojas de estilo antes mencionadas permiten personalizar a cada uno de los componentes utilizados dentro del software, cambiando su apariencia, incluyendo animaciones, con la finalidad de dar una mejor presentación.

3.1. Manejo de la información

Con la finalidad de mejorar el diagnóstico de los pacientes que se realizan un estudio de gastroenterología, se pretende dar un libre acceso a los resultados y análisis obtenidos de este tipo de estudios. Para ello la información obtenida será almacenada dentro de un servidor remoto en la web. Estos datos servirán de ayuda para médicos y estudiantes que realicen estudios similares en diferentes centros médicos.

En el capítulo anterior se indicó el proceso para obtener las imágenes y videos de los estudios de gastroenterología realizados. A partir de este punto se realizará el proceso de almacenamiento y visualización de la información. Se puede observar el proceso que sigue la información dentro de los diagramas de flujo de la Figura 3.1 y Figura 3.2.

Los procesos más importantes de los diagramas de flujo son el almacenamiento y visualización de los datos.

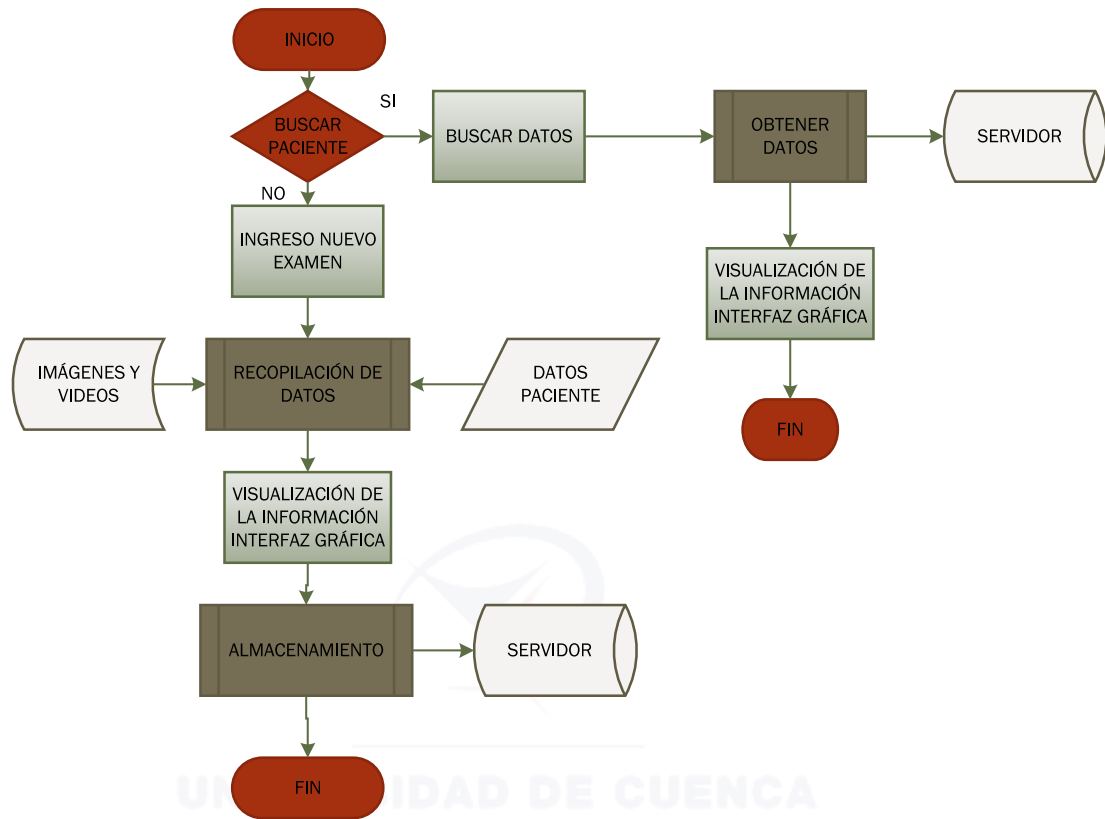


Figura 3.1: Diagrama de flujo de la información del paciente. Elaboración propia

3.2. Almacenamiento

3.2.1. Diseño y creación de la base de datos

Para el diseño de la base de datos se utilizó el modelo relacional. Es importante el diseño de la base de datos y las relaciones que se tenga entre cada tabla, permitiendo de esta manera una fácil modificación de la misma en un futuro. En el diseño de base de datos debe evitarse la redundancia de información en las columnas para mejorar el mantenimiento.

Las ventajas de utilizar el modelo relacional para la creación de la base de datos son las siguientes [42]:

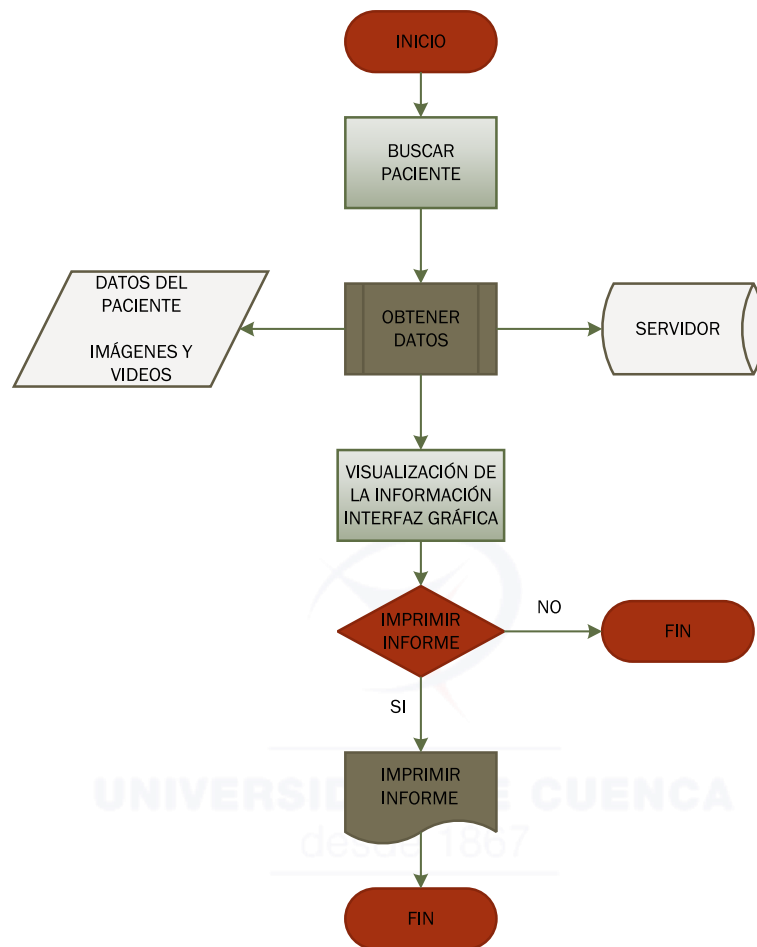


Figura 3.2: Diagrama de flujo de la información del Médico especialista. Elaboración propia

- Numerosas metodologías de diseño de base de datos.
- Manejo del lenguaje de consulta estándar SQL (structured query language - lenguaje de consulta estructurado).
- Compatible con la mayoría de los sistemas de gestión de bases de datos comerciales modernos.

3.2.2. Esquema de la base de datos

El esquema de la base de datos se puede observar en el Anexo B. La base de datos está compuesta por once tablas en total, la más importante de estas es la tabla de Exámenes, que almacena la



información de todo el estudio realizado. Las imágenes y videos no se almacenan dentro de la base de datos, puesto que al hacer esto la velocidad de procesamiento de las consultas aumentaría considerablemente [43].

Para realizar el almacenamiento de las imágenes y videos de manera ya sea remota o local, se guarda el path o directorio dentro del cual serán ubicadas. Al momento de realizar la visualización de la información se obtendrá el path correspondiente a cada paciente y se obtendrá su respectiva imagen o video.

La tabla *paciente* guarda la información básica del paciente, esta información no estará disponible al momento de consultar los estudios para un análisis de los mismos. El médico encargado de realizar el examen será el único usuario en tener acceso a esta información. La tabla *exámenes* es dependiente de las tablas:

- *Lista_DiagnosticoRemison*
- *Lista_EmpresaAfiliada*
- *Lista_Estudios*
- *Lista_Premedicacion*
- *Lista_Doctores_Remitentes*
- *Lista_Doctores_Encargados*
- *paciente*

La última a su vez de la tabla *procedencias*.



La relación entre la tabla *exámenes* con las mencionadas anteriormente es de uno a muchos [42]. Al igual que la relación entre la tabla *paciente* y *procedencias* y la tabla *doctores* con las tablas *Lista_Doctores_Remitentes* y *Lista_Doctores_Encargados*.

La tabla de *claveusuarios* junto con su dependiente *usuarios*, regulan el ingreso y el acceso a la información. Las tablas: *Lista_DiagnosticoRemison*, *Lista_EmpresaAfiada*, *Lista_Estudios*, *Lista_Premedicacion*, *Lista_Doctores_Remitentes*, *Lista_Doctores_Encargados*, almacenan la información relevante para el examen realizado. Finalmente, la tabla *doctores* almacena los nombres de todos los doctores ingresados en la base de datos para una búsqueda rápida de los mismos.

3.3. Características técnicas para la creación del software

El lenguaje *JAVA* es multiplataforma, por este motivo el software puede funcionar tanto en *Windows*, *Linux* y *Mac*, dependiendo de la versión de la máquina virtual o *JRE* que se tenga en cada sistema este funcionará de una manera distinta en cuanto a la visualización [44]. Las características del sistema operativo, programas y plataformas utilizados dentro de los cuales se creó el software son:

1. Sistema Operativo:

- Versión. - Windows 7.
- Tipo de Sistema. - 64 bits.
- RAM. - 2 GB.



2. Máquina Virtual JRE:

- Versión. – 1.7.

3. Servidor de Aplicaciones WampServer:

- Versión WampServer 2.2.
- Versión PHP 5.4.3.
- Versión Servidor Apache 2.4.2.
- Versión MySQL 5.5.24

4. Netbeans:

- Versión 8.1 – Windows.
- Versión SceneBuilder 2.2.

3.4. Visualización

La visualización es muy importante debido a que mediante esta se interactuará directamente con el usuario final en este caso el médico especialista. En las siguientes subsecciones se describen los procesos más importantes de la visualización.

3.4.1. Software–interfaz gráfica

Para la creación de la interfaz del software se utilizó el programa *Netbeans* [45]. Este permite crear una variedad de aplicaciones tanto web como de escritorio. El tipo de aplicación implementado para el software se conoce como *JavaFx Application*. Este tipo de aplicación, permite la combinación de código *xml* y estilos *css* [46]. Para la creación de las vistas o ventanas. El modelo utilizado para el desarrollo de la interfaz se conoce como modelo



vista controlador *MVC* [47]. La interfaz gráfica del software se observa en el Anexo A.1.

La interfaz gráfica del software se divide en cuatro módulos: *búsqueda de la información, visualización de la información, visualización de imágenes y videos y biopsias.*

- La búsqueda de la información del paciente se realiza mediante sentencias *SQL* [42]. Estas sentencias acceden a la base de datos ya sea local o dentro del servidor para obtener la información solicitada de cada paciente.
- La visualización de la información obtenida ya sea desde la base de datos o proporcionada por el paciente es mostrada dentro de los campos de la interfaz gráfica.

El módulo de visualización de las imágenes y videos, y el módulo de biopsias son relevantes dentro del funcionamiento del software, por este motivo se profundizará en ellos.

3.4.2. Visualización de las imágenes y videos

La señal de video obtenida por el capturador descrito en el Capítulo 2 es recibida en tiempo real al momento de iniciar este módulo.

3.4.2.1. Características de las imágenes y videos obtenidos

Dentro de la descripción técnica del capturador de video que se mencionó en el Capítulo 2, la resolución de las imágenes y videos

obtenidos es de 720 X 480 pixeles. Es de gran importancia conocer el espacio ocupado por cada imagen o video capturados más aún si se realizan exámenes diarios del paciente.

Luego de realizarse varias capturas de distintas imágenes provenientes de un endoscopio (ver Figura 3.3). El tamaño promedio de una imagen es de 26 KB a 36 KB. En cuanto al tamaño de un video se realizaron pruebas capturando videos de 1 minuto de duración, en promedio el tamaño de un video capturado es de 3.5 MB a 6.5 MB por minuto. Las extensiones de las imágenes obtenidas son *.jpg* y de los videos *.mov*.

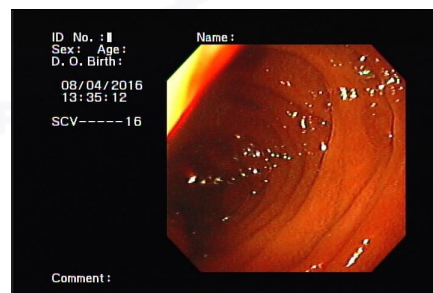


Figura 3.3: Imagen obtenida por el software. Elaboración propia

3.5. Herramientas relevantes para la visualización de imágenes y videos

En el Anexo A.2, se observa la ubicación de las herramientas más relevantes. Estas son:

3.5.1. Reconocimiento de cámaras y puertos

Para el control de las cámaras se utilizó la librería *webcam-capture-0.3.10-RC7* [48]. Esta librería lista las cámaras que se encuentran conectadas al computador, el capturador al enviar



una señal digital de video es reconocido como una cámara web. Al tener una lista es posible saber si el dispositivo capturador se encuentra conectado mediante el nombre del mismo. Los dispositivos capturadores *encore* con los cuales se realizaron la mayoría de las pruebas son reconocidos por defecto con el nombre *TridVid Capture 0*.

Mediante el código java mostrado a continuación, se controla si el dispositivo se encuentra conectado desde el inicio del software, de no estar conectado se visualizara dentro del combobox el mensaje de “*Desconectado*”.

```
List<Webcam> camaras = Webcam.getWebcams();
    if (camaras.size() > 0) {
        if (camaras.size() != numeroCamaras) {
            numCam = 0;
            int num = 0;
            comboCamaras.getSelectionModel().clearSelection();
            comboCamaras.getItems().clear();
            comboCamaras.setEditable(true);
            listaCamaras = FXCollections.observableArrayList();
            for (int i = 0; i < camaras.size(); i++) {
                String valor = String.valueOf(camaras.get(i));
                valor = valor.replace("Webcam ", "");
                listaCamaras.add(valor);
                String nuevoValor = valor.substring(0, 15);
                if (String.valueOf(nuevoValor).equals("TridVid
                Capture 0")) {
                    numCam = i;
                    num = i;
                }
            }
            comboCamaras.setItems(listaCamaras);
            comboCamaras.getSelectionModel().select(num);
            numeroCamaras = camaras.size();
            //            comboCamaras.setEditable(false);
        }
    }
```



```
    }  
    } else if (!comboCamaras.getValue().equals("Desconectado")) {  
        comboCamaras.setEditable(true);  
        comboCamaras.getItems().clear();  
        comboCamaras.getSelectionModel().clearSelection();  
        comboCamaras.getItems().add("Desconectado");  
        comboCamaras.getSelectionModel().select(0);  
        numeroCamaras = 0;  
    }  
    try {  
        Thread.sleep(200);  
    } catch (InterruptedException ex) {  
        Logger.getLogger(HiloCamarasPuertos.class.getName())  
            .log(Level.SEVERE, null, ex);  
    }  
}
```

Es importante saber que el software reconoce tanto las cámaras web conectadas al computador como el capturador de video, ambas dispositivos funcionan de manera correcta. En el segundo menú desplegable, se visualizan los dispositivos conectamos mediante comunicación serial al computador.

Para el control de puertos mediante comunicación serial, se utilizó la librería *JSSC-2.8.0*. Esta librería proporciona herramientas de software para las comunicaciones serial en *JAVA*. Al igual que la librería anterior permite saber si se encuentra conectado o no el dispositivo capturador. El código fuente utilizado para el control de puertos es el siguiente:

```
detectPort();  
    if (portList.size() > 0) {  
        if (portList.size() != numeroPuertos) {  
            comboPuertos.getSelectionModel().clearSelection();  
            comboPuertos.getItems().clear();  
            comboPuertos.setEditable(true);  
        }  
    }  
}
```



CAPÍTULO 3. SISTEMA DE TELEMEDICINA PARA GASTROENTEROLOGÍA

```
        comboPuertos.setItems(portList);
        comboPuertos.getSelectionModel().select(0);
        numeroPuertos = portList.size();
    }
} else if (!comboPuertos.getValue().equals("Desconectado")) {
    comboPuertos.getSelectionModel().clearSelection();
    comboPuertos.getItems().clear();
    comboPuertos.setEditable(true);
    comboPuertos.getItems().add("Desconectado");
    comboPuertos.getSelectionModel().select(0);
    numeroPuertos = 0;
}
try {
    Thread.sleep(200);
} catch (InterruptedException ex) {
    Logger.getLogger(HiloPuertos.class.getName())
        .log(Level.SEVERE, null, ex);
}
}

private void detectPort() {
    portList = FXCollections.observableArrayList();
    String[] serialPortNames = SerialPortList.getPortNames();
    for (String name : serialPortNames) {
        portList.add(name);
    }
}
}
```

De esta manera es posible saber si se tiene errores en el dispositivo capturador de video o el disparador de captura de video de **SAPHIENS**, puesto que se mostrará una notificación si no se detecta la conexión de los dispositivos.



3.5.2. Sistema de archivos base de datos CIE10

La base *CIE10* se implementó dentro del software para facilitar el diagnóstico del estudio realizado al paciente. Mediante esta base de archivos el médico especialista tiene la opción de obtener diagnósticos para un caso de estudio específicamente de gastroenterología. En el Anexo [A.3](#), se observa la distribución de los diagnósticos *CIE10*.

La base está compuesta en su inicio de veinte y un carpetas de estudios generales que se divide en un subdirectorío por cada especialidad a tratar dentro de las cuales se encuentran los estudios para cada caso dentro de la rama de gastroenterología.

3.5.3. Biopsias

El software cuenta con un módulo dedicado a obtener los puntos dentro de los cuales el médico especialista debe obtener muestras para realizar una biopsia. Estos puntos no son guardados dentro de la base de datos, debido a que solo se necesitan dichos puntos para generar el informe de la biopsia en ese momento. Por este motivo se almacenan dentro de la memoria del computador hasta realizar su impresión y no se utiliza espacio innecesario dentro del servidor remoto o local. De no realizar la impresión de estos puntos, la información se perderá. En la Figura [3.4](#) se observa este módulo.

3.5.4. Informes

El software también permite la gestión las citas médicas del paciente e informes para ayudar al gastroenterólogo durante los

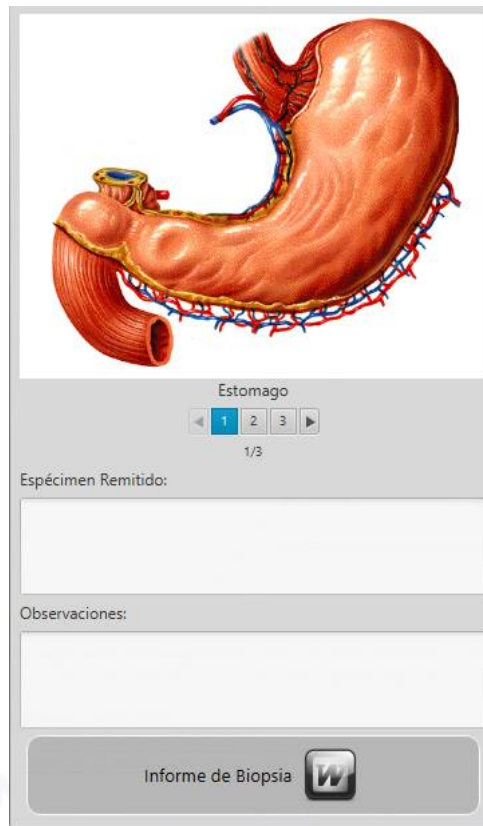


Figura 3.4: Módulo de biopsias. Elaboración propia

procedimientos de diagnóstico o tratamientos. Asociado a una cita en particular puede ser imágenes que el personal médico puede visualizar y extraer para su incorporación en el informe médico.

El informe incluye la información de la cita, así como la información demográfica sobre el paciente. Además, el personal médico puede pegar las imágenes relevantes seleccionadas usando la interfaz de usuario del sistema. Para obtener las imágenes relevantes el médico especialista selecciona cada una de ella para su incorporación dentro del informe final. La estructura final del informe se observa en el Anexo D.



Para generar el informe con las imágenes seleccionadas se utiliza la librería *java2word-3.3* [49]. Esta librería permite la generación de documentos de texto de código Java. El código relevante para la generación de los informes se puede observar en el Anexo E.

3.6. Posible uso de esta plataforma en telemedicina.

Una aplicación de la telemedicina se basa en la incorporación de tecnologías informáticas y de comunicación para llevar a cabo el diagnóstico médico y el tratamiento de los pacientes. En un típico servicio de telemedicina el Historial Clínico se transmite de igual manera que las señales y las imágenes adquiridas de un paciente.

Esta información puede ser analizada por médicos expertos ubicados en lugares lejanos del paciente, y pese a la distancia se puede obtener un informe detallado del caso clínico del paciente. Este informe junto con los diagnósticos y procedimientos de tratamiento se pueden enviar a la ubicación del paciente. Un diagrama que describe una posible aplicación de la telemedicina en gastroenterología se muestra en la Figura 3.5.

En la parte izquierda del diagrama se muestra la ubicación del paciente. En este punto se realiza el Historial Clínico del Paciente y se obtienen las imágenes y datos para almacenar en la base de datos. Esta información es luego transferida a un servidor remoto (servidor de la nube) a través del internet.

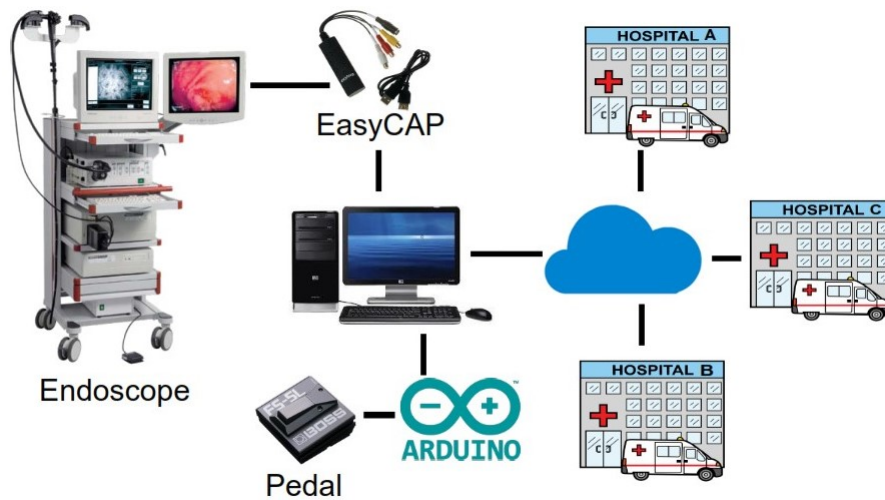


Figura 3.5: Diagrama de bloques del funcionamiento de *SAPHIENS*. Elaboración propia

En el momento en que la información se sube al servidor remoto, la información ya está disponible para su consulta y edición por parte del personal médico autorizado situado en varios centros hospitalarios remotos.

De este modo la experiencia médica puede ser utilizada de manera óptima para la obtención de un mejor diagnóstico o tratamiento para el paciente. *SAPHIENS* permite capturar y seleccionar el vídeo e imágenes de equipos gastroenterológicos, de acuerdo a la evaluación de diagnóstico del especialista. La información se almacena en un servidor de base de datos local dentro de una red de uso local.

Capítulo 4

Herramientas de hardware y software utilizadas

Aquí se dará la explicación del funcionamiento de los dispositivos de hardware que se utilizaron para la creación del proyecto, se habla también acerca de las librerías utilizadas para el control de la cámara del endoscopio. De la misma manera se dá una breve explicación del funcionamiento de los programas que se utilizan.

4.1. Herramientas de hardware

4.1.1. Arduino

Massimo Banzi, David Cuartielles y Gianluca Martino desarrollan *Arduino*, ahora este dispositivo es conocido en todo el mundo por expertos y aficionados a la electrónica. Tiene gran acogida y es por eso que ahora existe una gran cantidad de modelos y shields que aumenta las funcionalidades de esta placa electrónica.

Está conformado por pines de entrada y salida digitales, protocolos de comunicación, pines analógicos, botón reset, un conector usb para la conexión con la computadora y alimentación externa.

El cerebro de esta placa es un microcontrolador *ATMEL* (*ATmega 328*), aquí es donde se guarda el programa que el usuario cree para que tenga el funcionamiento deseado. Consta también conectores para la conexión con periféricos de entrada y salida como podría ser teclados, sensores, motores, entre otros [50].

Por su bajo costo, fácil de programar, y sobre todo porque es

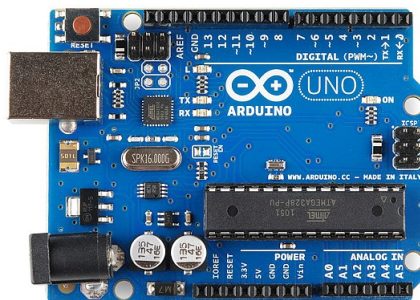


Figura 4.1: Arduino ATmega 328 [8]

hardware libre, se escogió este dispositivo para la construcción del sistema *SAPHIENS* que se explica en esta tesis. Se escogió la placa Arduino Uno, y cuenta con las siguientes características:

- 2 *KB* de SRAM.
- 1 *KB* de memoria EEPROM.
- Comunicación serial UART TTL 5V.
- Comunicación I2C y SPI.

4.1.2. Capturador de video

El capturador utilizado para este proyecto es el de marca ENCORE, modelo ENMVG. Es un dispositivo para la captura de audio y video analógico. Para que funcione solo hay que conectar el video analógico que se desea capturar y conectarle a la



computadora, viene con un programa que obtiene tanto el video como el sonido del dispositivo. Para el caso del sistema de telemedicina, se conecta de la misma manera, pero ahora se ejecuta la aplicación realizada, y mediante la librería *webcam-capture-0.3.10-RC7.jar* que se explica en este capítulo, se captura la imagen. Las especificaciones se encuentran en la Tabla [51]:

Interfaz del Dispositivo:	<i>USB 2.0, backward compatible con USB 1.x</i>
Dimensiones	93 x 30 x 15 mm
Video	Formato de la Señal: NTSC and PAL Entrada: Composite RCA and S-Video Resolución: NTSC: 720x480, 352x480, 352x240 PAL: 720x576, 352x576, 352x288 Convertidor AD: 10-bit Analog-to-Digital Converter Formato de Archivos: AVI, MPEG-1(VCD), MPEG-2 (DVD), MPEG-4** (XVID) and WMV**
Audio	Canales de Audio: Stereo Entrada: L/R RCA Convertidor AD: 16-bit Analog-to-Digital Converter Frecuencia de Muestreo: 44.1KHz / 48KHz Frecuencia de Datos de Audio: 128k 192k bit/second

Tabla 4.1: Especificaciones encore, [9]

4.2. Herramientas de software

4.2.1. Netbeans

Creado por Sun Microsystems, es un proyecto exitoso de código abierto y consta de dos productos: el Netbeans IDE y Netbeans Platform. Para este tema de tesis se utiliza el Netbeans IDE por



lo que hablaremos de este entorno de desarrollo [52].

Netbeans IDE es una herramienta donde se puede escribir, compilar, depurar y ejecutar programas, sirve para varios lenguajes de programación, entre ellos Java que es el lenguaje que se utiliza para la generación del software.

Consta con Java Swing que simplifica el desarrollo de aplicaciones para escritorio, pero los componentes de Swing tienen un estilo difícil de modificar, por lo que se buscaron otras herramientas y la de fácil aprendizaje y simplicidad en usar fue Scene Builder que sirve con Netbeans pero es necesario de JavaFX que se explica a continuación.

4.2.2. JavaFX

Utilizado para la creación de aplicaciones con características web, las aplicaciones creadas con JavaFx pueden ser ejecutadas en una amplia variedad de dispositivos, como de escritorio, web, tv, consolas de videojuegos y hasta para celulares. Para poder ser ejecutado solo necesita de que el dispositivo tenga instalado *JRE* (Java Runtime Environment) que permite la ejecución de programas Java [53].

JavaFX amplía la potencia de Java permitiendo a los desarrolladores utilizar cualquier biblioteca de Java en aplicaciones JavaFX. De esta forma, los desarrolladores pueden ampliar sus capacidades en Java y utilizar la tecnología de presentación que JavaFX proporciona para crear atractivo visual.



4.2.2.1. Características

Permite crear animación en las aplicaciones, pueden tener sonido y video. Al ser un modelo MVC (Modelo Vista Controlador), permite el flujo de trabajo adecuado entre diseñador y desarrollador.

La interfaz de usuario se crea en un archivo *FXML*, el lenguaje es parecido a código *HTML*, pero *ORACLE* ha desarrollado el programa Scene Builder que se puede crear estos archivos con una interfaz gráfica. Parecido a la programación web, usa de archivos *CSS* para dar estilo a las ventanas, también puede darse algunos estilos mediante Scene Builder pero por buenas prácticas de programación se utiliza los archivos CSS [53].

4.2.3. Scene Builder

Con esta aplicación se puede construir la interfaz gráfica de un programa en Java de forma sencilla, nos evita entonces crear la vista del proyecto mediante código. Además, nos ayuda con los eventos que tiene cada componente del proyecto [53].

Estas herramientas son suficientes para la creación del diseño y desarrollo del programa para el sistema de telemedicina. Hay que tener presente que para el uso de la base de datos se utiliza también el programa Wampserver que ya fue mencionado en capítulos anteriores, también es necesario mencionar el uso de algunas librerías para mejorar la capacidad del programa, las mismas que son explicadas a continuación:

- **Webcam-capture-0.3.10:** Librería utilizada para la obtención de imágenes de cámaras web [54]. Lo que se usa de esta librería es:



- Reconocimiento de cualquier cámara que se encuentre conectada en la computadora.
 - Visualización de las imágenes que la cámara seleccionada está obteniendo.
 - Ayuda con el guardado de la imagen, es decir, toma fotografías.
- **Controlsfx-8.40.10:** Librería muy útil para mejorar los controles con los que cuenta la aplicación Scene Builder, lo que hace es agregarles más funciones y mejores aspectos a los controles [55].
 - **Java2word-3.3:** Con esta librería se generan los informes médicos en un formato muy común que es “.doc”. Se utiliza este formato ya que se hace de forma simple una plantilla para el informe y se puede cambiar rápidamente el diseño sin necesidad de cambiar algo en el programa [49].

Capítulo 5

Resultados

Al terminar de realizar la primera versión del dispositivo SAP-HIENS, los resultados son positivos y negativos. Se presentan si con los dispositivos escogidos se tuvieron buenos resultados, cuales son las fallas y problemas que se presentaron y que se ha obtenido al realizar este tema de tesis.

La funcionalidad del software demostró ser amigable para el usuario, ganando así aceptación por un pequeño número de doctores que vieron el sistema en funcionamiento; tres de cuatro doctores estaban dispuestos a utilizar el sistema.

Reducción de tiempo para el médico especialista al momento de realizar un estudio de gastroenterología.

La actualización del equipo endoscópico utilizado para pruebas del sistema, culminó de manera exitosa, puesto que las ventajas en cuanto al almacenamiento de la información en una base de datos y el control de historias clínicas funcionó de manera correcta.



El primer capturador de video utilizado para la construcción del dispositivo SAPHIENS fue el dispositivo ENCORE. Este capturador funcionó sin dificultad alguna, siendo compatible con el software desarrollado y obteniendo las imágenes y videos de manera eficiente. Al utilizar el capturador de video EASY-CAP, se tuvieron inconvenientes de compatibilidad con las librerías utilizadas en el sistema SAPHIENS, por este motivo no se visualizaba de manera correcta el video dentro del software de SAPHIENS.

El tiempo de respuesta del disparador para la captura de video controlado por el dispositivo pedal, dio problemas en sus inicios debido al tiempo programado para detectar cada pulsación dentro del módulo Arduino. Al reducir el tiempo de espera dentro de la programación se corrigió este problema

La visualización del video dentro de la interfaz gráfica se realiza en tiempo real, resultado esperado al utilizar el dispositivo capturador de video. La desventaja obtenida se da al momento de guardar el video puesto que es necesario configurar la cantidad de frames por segundo (fps), dependiendo las características del ordenador para la reproducción del video.

Datos almacenados en la base de datos de los estudios realizados de manera correcta a un dominio dentro de la web. La desventaja que se tiene es el almacenamiento de las imágenes y videos debido a que el espacio de almacenamiento es reducido en comparación a la cantidad de videos e imágenes que se obtendrán, por este motivo se almacena los mismos de manera local o dentro de un servidor remoto mas no en la base de datos.



Problemas en el tiempo de respuesta del software dependiendo de las características del ordenador en el que fue instalado. Dichas características no cumplían las especificaciones descritas dentro del capítulo 2.

Mediante la creación de un programa que accede a la base de datos dentro del dominio web se pudo obtener la información de los estudios realizados y presentar los mismos de manera correcta. De esta manera se puede compartir la información con médicos y estudiantes ubicados en distintas localidades.

La atención al paciente en cuanto a tiempo realizado por estudio y diagnósticos presentados se dio de manera favorable debido a que el médico especialista al tener almacenada la información de cada paciente observo la evolución del mismo.

Reducido gasto por parte del médico especialista al actualizar el equipo endoscópico que puede seguir trabajando de manera eficiente, mediante una actualización de bajo costo.

Finalmente, realización de una publicación para un artículo del sistema SAPHIENS para STSIVA (SYMPOSIUM ON SIGNAL PROCESSING, IMAGES AND ARTIFICIAL VISION). Ver Anexo [G](#).

El Arduino escogido funciona correctamente en el sistema, por lo que para otras versiones se puede colocar otro tipo de Arduino más compacto y con menos características ya que no es necesario de todo su potencial. Al ser Arduino un dispositivo de hardware



libre, se puede también obtener el esquema electrónico y configurarlo para que sea funcional para SAPHIENS, aunque se necesita de mayor trabajo en la construcción.

El sistema SAPHIENS tiene las siguientes características:

- Creación del historial clínico para un ilimitado número de pacientes.
- El sistema se conecta con el equipo médico, en este caso un endoscopio, así se puede obtener imágenes cuando el doctor lo requiera mediante un pedal conectado al sistema que será el que avisará cuando debe tomar una foto. Como un extra, también puede generar video del examen que se está realizando.
- En cada cita médica se guarda toda la información necesaria para llevar dicho seguimiento, además de guardar las imágenes obtenidas por el endoscopio en su respectiva cita, de la misma manera guarda los videos.
- EL sistema es intuitivo y de fácil manejo. Tiene también métodos de búsquedas avanzadas que cuando se escribe parte del texto a buscar, ya sea el nombre del paciente, el examen o la cita, encuentra los resultados más parecidos.
- Realiza informes del examen realizado para el paciente, de esa manera el paciente puede tener mayor conocimiento de los resultados obtenidos de la cita realizada. También hace informes para las biopsias, que mediante imágenes se observa donde se hizo las respectivas biopsias.

- Guarda la información de la clínica u hospital para los informes.
- Notificación del sistema cuando está conectado el endoscopio y visualización en pantalla.
- Cuando se busca un paciente específico, también muestra todas las citas realizadas de ese paciente.
- Existe una sección donde se puede subir un examen a un servidor en la nube, de esa manera cualquier otra persona puede ver el examen si consta con el sistema de visualización de exámenes realizados. Hay que recalcar que la información del examen que se sube a la nube no tiene información del paciente como nombre o número de cédula, cuidando así la privacidad.
- A demás, un sistema más sencillo que es para mostrar los exámenes que se encuentran en la nube, así estudiantes o doctores de cualquier lugar del mundo que cuenten con el sistema y con internet, puedan ver los exámenes realizados y mejorar los conocimientos o tener una segunda opinión acerca de dicho examen.
- Todo un sistema que es de fácil creación y de muy bajo costo.

Capítulo 6

Conclusiones

6.1. Conclusiones

Finalizado el proyecto de tesis y de los objetivos planteados, se han llegado a se tienen las siguientes conclusiones:

- La utilización de dispositivos electrónicos desarrollados permite la creación de un sistema preciso y de bajo costo en comparación con el desarrollo de las tarjetas electrónicas del mismo que representan un costo en cuanto a precios y tiempo.
- La resolución de video obtenida mediante el capturador, visualiza de manera correcta los estudios realizados, pero con el avance de la tecnología los nuevos capturadores de video tienen resoluciones de alta calidad. Se plantea la actualización de SAPHIENS para obtener imágenes y videos de alta calidad que mejoraría el análisis de los estudios realizados.
- El software tiene una buena funcionalidad para médicos especialistas en gastroenterología, pero tiene la opción de expandirse a mayores áreas mediante actualizaciones del software, por lo que en futuras versiones del producto se puede



contar con un sistema más general para la ayuda de cualquier médico.

- La atención al paciente final mejora debido a que el estudio y el informe del mismo se puede realizar en un solo proceso, guardando también la información obtenida por el examen y así poder llevar un seguimiento del paciente por parte del médico. El tiempo de la cita también disminuye mejorando el servicio y atendiendo a más pacientes por día.
- Una herramienta útil para los estudiantes de medicina que quieran ver resultados reales, sobretodo ver las opiniones de médicos que tienen años de experiencia.
- La gestión y almacenamiento de historias clínicas mejora el proceso de ingreso y evaluación de los pacientes, permitiendo de esta manera mejorar futuros resultados en estudios similares. Muy importante también para ver la evolución que ha tenido un paciente.
- AL hacer al sistema más amigable para el usuario, se dificulta más el desarrollo del software, ya que este tiene que estar pendiente a cambios inesperados, como cuando se desconecta el dispositivo del software, o cuando no se llena de manera correcta los datos de un paciente, también cuando no tuvo una conexión con la base de datos, entre otros.
- Facilita al médico de manera considerable el trabajo que debe realizar ya que cada vez que regrese un paciente, tendrá que preguntarle toda la información y detalles de citas anteriores, perdiendo así tiempo y pacientes.
- Enviar una imagen a un servidor web, es más complicado



mediante programación java, es por eso que se aconseja usar un servidor que trabaje también con java, o buscar otras herramientas de desarrollo que facilitan la programación en la web.

- javaFX, mediante archivos FXML, da mayor libertad al diseñar la interfaz gráfica, obteniendo mejores resultados que creando en Java Swing.
- El sistema SAPHIENS cuenta con cie10 específicamente para diagnósticos en el área de gastroenterología, no cabe duda que puede ser aumentado para otras áreas de atención.
- Se logró crear un proyecto de bajo costo pero con muy buenas especificaciones para el manejo y control de la información de pacientes para un doctor especialista en Gastroenterología.

Anexo A

Capturas del sistema



UNIVERSIDAD DE CUENCA
desde 1867



ANEXO A. CAPTURAS DEL SISTEMA

The screenshot displays a medical system interface with the following components:

- Top Bar:** Includes navigation icons (home, info), a date field set to 04/08/2016, and a 'Puentes' dropdown menu currently set to 'COM4'. There are also buttons for 'Imágenes', 'Videos', and 'Biopsia'.
- Patient Information:** A section for 'Paciente' with fields for 'Número de Registro' (014-0000000002), 'H.C.' (0104826458), 'Procedencia' (26 años), and 'Fecha Nacimiento' (07/11/1989). The patient name is 'JUAN ROSENDO MOLINA LEÓN'. There are buttons for 'Nueva Cita' and 'Citas Existentes'.
- Endoscopy Details:** A section for 'Estudio' with fields for 'Estudio' (ENDOSCOPIA), 'Empresa' (CUENCA), 'Médico Encargado' (Dr. Victor Hugo Minga), 'Médico Remitente' (Dr. Victor Hugo Minga), 'Premedicación' (Xilocaina Spray + Propofol), 'Diag. Remisión' (Epigastrología), and 'C.B.' (od. Beneficiari).
- Diagnosis and Report:** A section for 'Informe' with a 'Diagnóstico' field containing 'HERNIA HIATAL' and 'GASTROPATIA ENANTEMATOSA'. The report text reads: 'ESOFAGO.- Calibre y trayecto normal con mucosa normal a la tinción electrónica no se observa mucosa metaplasia con cambio mucosa a 37 cms de ADS Y con cardias a 39 cms de ADS'. There are fields for 'Fecha Creación' (2016/08/04) and 'Fecha de Modificación' (2016/08/04).
- Video Feed:** A large central window showing a live endoscopy video. Below it are fields for 'ID No.', 'Sex', 'Age', 'D.O. Birth', and 'SCV'. A 'Comment' field is also present. A timeline at the bottom of the video shows frames 11 through 20, with frame 15 selected.
- Bottom Panel:** Contains search bars for 'Paciente' and 'Empresa'. The 'Paciente' search shows 'CÉDULA' and 'PACIENTE' with search results for 'Marco Bacuilima' and 'JUAN ROSENDO MOLINA...'. The 'Empresa' search shows 'ESTUDIO' and 'EMPRESA' with search results for 'IESS' and 'CUENCA'.
- Right Side:** Includes buttons for 'Iniciar Cámara', 'Informe', 'Nueva Búsqueda...', and 'Guardar Cita'. A system tray at the bottom right shows the time 12:54 and date 04/08/2016.

Figura A.1: Interfaz Gráfica del sistema

Sistema de Informes Médicos (SIM)

Fecha: 08/10/2016

Puertos: Desconectado

Imágenes Videos Biopsia

Cámaras: TOSHIBA Web Camera - CAMARAS Y PUERTOS

Paciente: Nombre del Paciente

Número de Registro: 014-000000003 H.C.: Lugar

N° Seg./C.I.: Cédula Procedencia: 0 años Sexo: MASCULINO

Fecha Nacimiento: 0 años

Nueva Cita Citas Existentes

Estudio: Estudio a realizarse Empresa:

Médico Encarga... Médico Encargado de la Cita

Médico Remiten... Nombre del Médico Remitente

Premedicación: Premedicación dada Cama:

Diag. Remisión: Diagnóstico por remisión C.B.: od. Beneficiari

Informe: Diagnóstico: CIE10

INICIO DE CAPTURA **INFORME**

Iniciar Cámara Informar

Nueva Cita Guardar Cita

Nuevo Paciente

Buscar Paciente...

CÉDULA PACIENTE

0104162763 Marco Bacuilima

0104926458 JUAN ROSENDO MOLIN...

Buscar Cita...

ESTUDIO EMPRESA

Endoscopia IESS

ENDOSCOPIA UCUENCA

ENDOSCOPIA CUENCA

ENDOSCOPIA CUENCA

ENDOSCOPIA CUENCA

ENDOSCOPIA CUENCA

ENDOSCOPIA CUENCA

ENDOSCOPIA CUENCA

Figura A.2: Herramientas relevantes

Enfermedades infecciosas intestinales	01 Ciertas enfermedades infecciosas y Parasitarias
Tuberculosis	02 Tumores [neoplasias]
Ciertas zoonosis bacterianas	03 Enfermedades de la sangre y de los órganos hematopoyéticos, y ciertos trastornos que afectan el mecanismo de la inmunidad
Otras enfermedades bacterianas	04 Enfermedades endocrinas, nutricionales y metabólicas
Infecciones con modo de transmisión predominantemente sexual	05 Trastornos mentales y del comportamiento
Otras enfermedades debidas a espiroquetas	06 Enfermedades del sistema nervioso
Otras enfermedades causadas por clamidias	07 Enfermedades del ojo y sus anexos
Rickettsiosis	08 Enfermedades del oído y de la apófisis mastoides
Infecciones virales del sistema nervioso central	09 Enfermedades del sistema circulatorio
Fiebres virales transmitidas por artrópodos y fiebres virales hemorrágicas	10 Enfermedades del sistema respiratorio
Infecciones virales caracterizadas por lesiones de la piel y de las membranas mucosas	B65 Esquistosomiasis [bilharziasis]
Hepatitis viral	B66 Otras infecciones debidas a tremátodos
Enfermedad por virus de la inmunodeficiencia humana (VIH)	B67 Equinocosis
Otras enfermedades virales	B68 Teniasis
Micosis	B69 Cisticercosis
Enfermedades debidas a protozoarios	B70 Difilobotriasis y esparganosis
Helminthiasis	B71 Otras infecciones debidas a cestodos
Pediculosis. Acanthias y otras manifestaciones	B72 Dracontiasis
Secuelas de enfermedades infecciosas y parasitarias	B73 Oncoercosis
Bacterias, virus y otros agentes infecciosos	B74 Filariasis
Otras enfermedades infecciosas	B75 Triquinosis
	B76 Anquilostomiasis y necatoriasis
	B77 Ascariasis
	B78 Strongiloidiasis
	B79 Tricuriasis
	B80 Enterobiasis
	B81 Otras helmintiasis intestinales, no clasificadas en otra parte
	B82 Parasitosis intestinales, sin otra especificación
	B83 Otras helmintiasis

Figura A.3: Base de datos CIE10

Anexo B

Base de datos



UNIVERSIDAD DE CUENCA
desde 1867

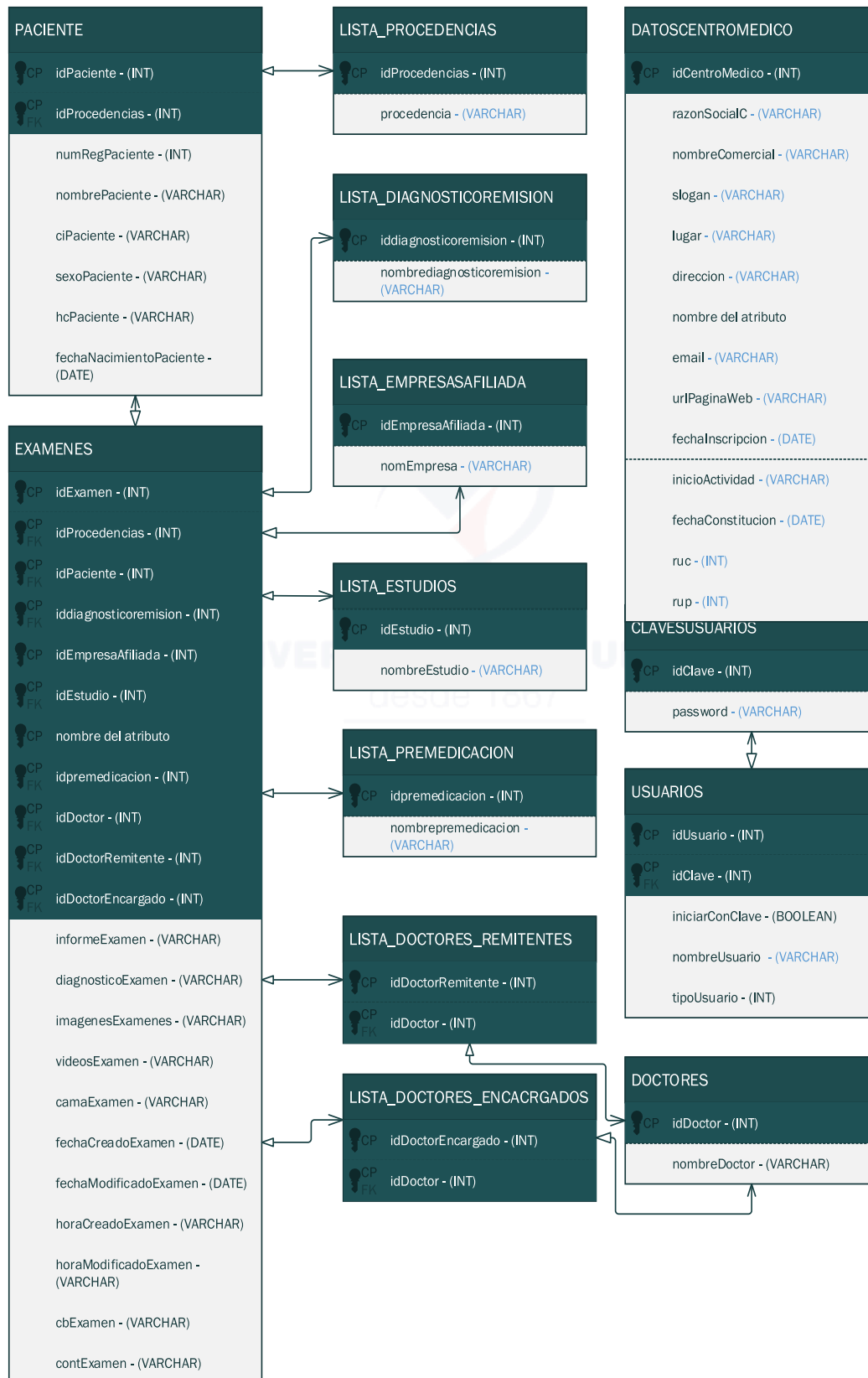


Figura B.1: Diagrama de la base de datos

Anexo C

Artículo



UNIVERSIDAD DE CUENCA
desde 1867

Video and Imaging Gastroenterological Medical Equipment Oriented to Telemedicine

Juan Molina
and Marco Bacuilima
Departamento de Eléctrica,
Electrónica y Telecomunicaciones,
Universidad de Cuenca
Cuenca, Ecuador
juan.molinal@ucuenca.ec
marco.bacuilimao@ucuenca.ec

Villie Morocho
Departamento de Ciencias
de la Computación,
Universidad de Cuenca
Cuenca, Ecuador
vmorochoz@gmail.com

Ruben Medina
and Alexandra La Cruz
Investigador Prometeo,
Universidad de Cuenca
Cuenca, Ecuador
rmedinaenator@gmail.com
alacruz@usb.ve

Abstract

This paper describes a system for management of clinical information in gastroenterology. The system consists of two blocks (hardware and software), both developed during the investigation. The hardware interface is connected to endoscopy equipment for video and image acquisition. Then, the patient electronic health record (EHR) is created using a software designed for including the relevant images and video sequences selected by the medical staff. The patient EHR is stored locally as well as in a remote server where authorized users can review and eventually edit the information within a telemedicine protocol.

1. Introduction

Gastrointestinal endoscopy is a low cost therapeutic and diagnostic tool where a thin and flexible tube known as endoscope enables visualization of the gastrointestinal track. The endoscopy services used the required equipment for performing the image acquisition including a light source, video camera and also tools for taking tissue samples or for cutting polyps [8]. During the examination the gastroenterologist performs visualization of different organs such as the esophagus, stomach, and duodenum, and then he has to select the relevant images for including in the medical report [13]. For completing this task they usually have basic and limited tools that tend to be time consuming. These tools lack of the appropriate systems for handling the patient electronic health record (EHR) according to the standard clinical recommendations. In a typical telemedicine application the EHR is transmitted via Internet, because it is a reliable way that is available in almost all over the world. A remote server is used, due to the need of supporting the

information access all the time, until it is received in the selected remote medical center. The designed protocol implies not only the mailing of information from a medical center to another, but also the system allows that the remote medical center could send information to any other center. This assures system connectivity among all medical centers [2].

Several attempts for gastroenterological telemedicine applications have been reported. Puentes et al. [11] describe a first approach for a Telemedicine in Gastroenterology (TMGE) project oriented to utilization of Asynchronous Transfer Mode (ATM) networks, distributed databases, intranet and multimedia tools. In Braga et al. [8] a solution including streaming of video through internet is reported. Their solution also provides tools for exporting exams in several video formats. The system also collects clinical information and shares this EHR during several stages of the therapeutic or diagnostic procedure between authorized medical staff. Graszew and Rakowsky [7] report a systematic qualitative review concerning the telemedicine applications for chronic digestive diseases. The document describes several telemedicine based studies; however, they do not describe the technological tools used for performing the telemedicine application. Zildzic et al. [17] analyzes gastroenterohepatology telemedicine applications within a framework of Picture Archiving and Communication System (PACS) and DICOM standard. They propose to embed this telemedical application within the PACS framework using the DICOM protocol for handling the multimodal image and video sources. The EHR could also be incorporated within the Hospital Information System (HIS) using the HL7 standard. They also propose using the telemedicine applications for tele-education. In Cross R. [1] a review concerning the application of telemedicine for patients with in-

inflammatory bowel disease is presented. An important application is for helping health providers to educate and to monitor patients as well as helping patients to adhere self-care plans. He discusses the application of a telemanagement system in a group of patients. Results found were promising as the patients were satisfied with the system and attained a decrease in disease activity and higher quality of life. Riaz and Atreja [12] describe clinical trials using high-speed Internet and smartphone devices and applications for the treatment of chronic gastrointestinal diseases. They found that even when there are several ongoing projects about this type of application, it is still necessary more research for confirming the utility of these technologies in gastroenterology.

Several contributions concerning telemedicine applications have been previously reported. In Perez et al. [10] a new strategy is proposed for integrating several known and universally accepted medical ontologies and controlled vocabularies, in order to make automatic semantic annotations over DICOM files. In Dugarte et al. [2] the development of a Cardiology Electronic Health Record (CEHR) system is reported. Software consists of a structured algorithm designed under Health Level-7 (HL7) international standards. Novelty of the system is the integration of high resolution ECG (HRECG) signal acquisition and processing tools, patient information management tools and telecardiology tools.

The main contribution of this paper is the developing and implementation of a system for performing the acquisition of images, video sequences and information from standard gastroenterology endoscopic equipment for further incorporation into a patient EHR. This system can improve the management of clinical information and enables a gastroenterology telemedicine application.

2. Method

A system including hardware and software is proposed for handling the patient electronic health record including acquisition of video or images recorded during the gastroenterological examination.

2.1. Hardware Description

The hardware includes an EasyCap frame grabber that is used for performing the video acquisition. This device is controlled by an Arduino microcontroller that allows selecting the frames or clips to be acquired as well as the transferring of this information to a personal computer through an USB port.

2.1.1 EasyCap frame grabber

Two steps need to be executed for converting an analog signal into a digital signal: quantification and sampling. In the quantification process the analog signal is sampled and

each value obtained is assigned a value that depends on the number of bits of the Analog Digital Converter (ADC). The quantized value is coded in binary, which likewise depends on the number of bits of the ADC. The number of bits of the ADC is known as the resolution of the signal; most of video grabbers have a resolution of 8-bit with flash memory architectures. The EasyCap frame grabber captures high quality video with audio and transfer the digital data to the computer using an USB 2.0 interface without requiring sound card. The device supports resolution NTSC: 720 x 480 at 30 fps, and it has a RCA composite video input and a S-video input. Figure 1 shows the EasyCap device and its connectors with video sources and the computer.

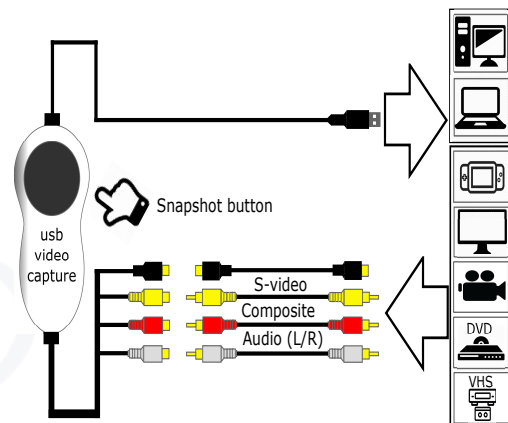


Figure 1. EasyCap frame grabber device and their connectors to video sources and computer

2.2. Arduino microcontroller

Arduino is an open source hardware and software development-oriented for interactive projects [3]. This device will be used to detect the precise moment, selected by the medical doctor, for performing the video acquisition. The medical staff interacts with the system using a pedal where switches are located (see Figure 2). Arduino receives a trigger signal that is used starting and ending the video acquisition. The device software is written using a simplified version of the C programming language. The software is installed as firmware running on the device.

2.3. Software Description

The main software component running in a personal computer, allows the doctor to create an EHR for the patient incorporating the demographic and clinical information as well as a detailed gastroenterological examination report; including images, video clips and anatomical locations for biopsy samples. The patient EHR can be stored locally or even transferred to a remote cloud backup location where this information can be accessed by other authorized medical staff. In this way a gastroenterological telemedicine application is performed. The software application was devel-

oped using open source libraries and performs the transfer of information considering data encrypting.

2.4. Hardware design and construction

In Figure 2 the connection between the target gastroenterology equipment and the developed hardware is shown. The EasyCap device is connected to the video output from the gastroenterology equipment. The video signal is digitalized by this device and transferred to the computer through a USB port. The program running on the Arduino microcontroller reads the pedal switches de-bounds the signal and transfers this information to the host computer using the USB interface. The device used for starting and ending the acquisition is a Boss FS-5U pedal that is normally intended for musical applications. It is a metallic heavy-duty switch that allows easy interaction between the medical staff and the system while working with the endoscopic probes and the patient.



Figure 2. Diagram describing the connection between the host gastroenterology equipment and the developed hardware

3. Results

This application is intended for a gastroenterology telemedicine application. In consequence it is very important to manage the patient EHR with a secure and simple user interface. The development of the software was performed using the *NetBeans* platform [16], that is an open source JAVA Integrated Development Environment (IDE). Within this environment several technologies can be handled. In this case the JavaFX was used for developing Web based Rich Internet Applications (RIAs) [15]. The main window of the software developed using this JAVA IDE is shown in Figure 3. The application requires the following libraries:

- **webcam-capture-0.3.10-RC7:** It acquires and handles video streams from any camera connected to the

computer. In this case the EasyCap frame grabber is considered as a video source. The library is a generic webcam java API [5].

- **java2word-3.3:** it is a library that allows generating MS word documents from JAVA code. In this application this library is used for writing the clinical reports of the patient [6].
- **controlsfx-8.20.8:** it is an open source library providing user interface controls and other tools to complement the core JavaFX distribution [4].
- **jssc-2.8.0:** it is a library providing software tools for serial communications in JAVA.
- **mysql-connector-java-5.1.16-bin:** This library is intended for handling the database server [9].

The management of the created database from web applications is handled by an open source application known as *Wampserver* [14].

3.1. Telemedicine features

A telemedicine application is based in incorporating communication and computer technologies for performing medical diagnosis and treatment of patients. In a typical telemedicine service the patient EHR is transmitted as well as the signals and images acquired from a patient. This information can be analyzed by medical experts located far from the patient where a detailed medical report can be obtained. This report can be send to the patient location involving diagnosing or treatment procedures. A diagram describing a possible telemedicine application in gastroenterology is shown in Figure 4. In the top of the diagram the location of the patient is shown. The electronic health record of the patient is constructed and the signal or video acquisition is performed. This information is then transferred to a remote server (cloud server) through the internet link. The information is now available for consulting and editing by authorized medical staff located in several remote hospital centers. In this way the medical expertise can be optimally used for obtaining a better diagnosis or treatment for the patient. The system allows capturing and selecting video and images from gastroenterology equipment, according the diagnostic evaluation of the specialist. The information is stored in a local database server as a patient EHR and the authorized medical staff can edit the patient information. The system also allows managing the patient medical appointments and reports for helping the gastroenterologist during the diagnosis or treatments procedures. Associated to a particular appointment can be images or video sequences that the medical staff can visualized and extract for incorporation into the medical report. An impor-

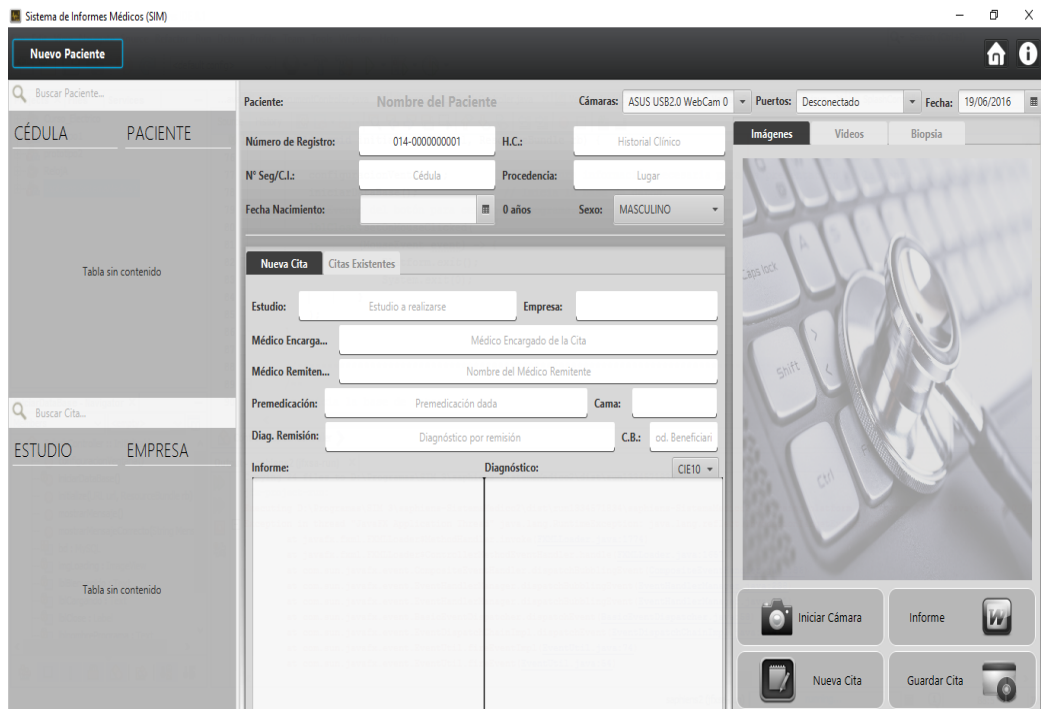


Figure 3. Main window of the software application for the gastroenterology telemedicine system

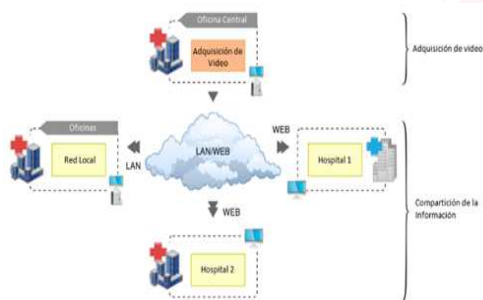


Figure 4. Telemedicine application

tant feature of the system is the generation of gastroenterology reports as shown in Figure 5. The report includes appointment information as well as demographic information about the patient. Additionally the medical staff can paste the relevant images selected using the user interface of the system. The conclusion and recommendations or treatment of the patient are also incorporated in this report.

4. Conclusions

The reported gastroenterology telemedicine system can be used with existent endoscope gastroenterology equipment by incorporating important features such as creation and management of the patient EHR and telemedicine applications. The system includes hardware that is constructed using low cost components and the software is designed using open source libraries. All relevant clinical information is available in the web and can be accessed se-

curely by authorized medical staff that can participate during the diagnostic and therapeutical process. As future research we are working for performing the clinical validation and for complying with the HL7 standard for the EHR.

5. Acknowledgments

This work was mainly supported by the Prometeo Project of the Ministry of Higher Education, Science, Technology and Innovation (SENESCYT) of the Republic of Ecuador.

References

- [1] R. K. Cross. Application of telemedicine for patients with inflammatory bowel disease: review of the literature and future implications. *Journal of Gastrointestinal & Digestive System*, 2011, 2012.
- [2] N. Dugarte, R. Medina, L. Huiracocha, and R. Rojas. Open source cardiology electronic health record development for digicardiac implementation. In *11th International Symposium on Medical Information Processing and Analysis (SIPAIM 2015)*, pages 96810Y–96810Y. International Society for Optics and Photonics, 2015.
- [3] W. Durfee. *Arduino microcontroller guide*. minnesota: University of minnesota, 2011.
- [4] F. experience. Controlsfx. <http://fxexperience.com/controlsfx/>, 2016. [Online; accessed 16-June-2016].
- [5] GitHub. webcam-capture. <http://webcam-capture.sarxos.pl/>, 2015. [Online; accessed 16-June-2016].

Anexo D

Exámen gastroenterológico



Universidad de Cuenca - Sistema de telemedicina en gastroenterología
Director de Tesis: Dr. Villie Morocho
Tesistas: Marco Bacuilima - Juan Molina

N°Registro: 014-0000000002

Enlace:

Paciente: JUAN ROSENDO MOLINA LEÓN

Estudio: ENDOSCOPIA

Sexo: MASCULINO

Edad: 26 años

H.C.: 2

Cama:

N°Seg. / C.I.: 0104826458

CB:

Médico Encargado: Dr. Anónimo

Médico Remitente: Dr. Anónimo

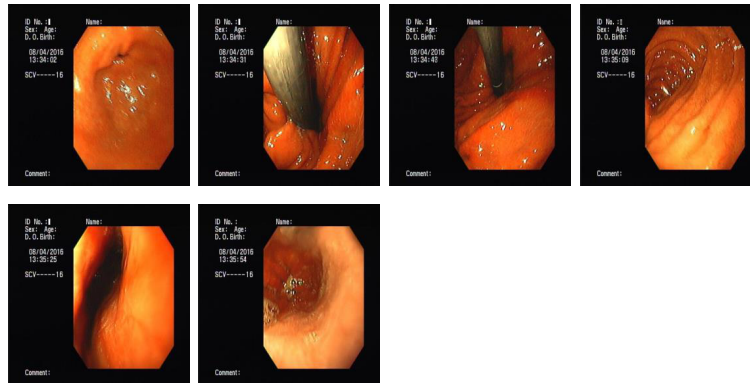
Procedencia: CUENCA

Empresa: CUENCA

Premedicación: Xilocaina Spray + Propofol

Diagnóstico de Remisión: Epigastralgia

Fecha: 2016/08/04



Informe:

ESOFAGO. - Calibre y trayecto normal con mucosa normal a la tinción electrónica no se observa mucosa metaplasia con cambio mucosa a 37 cms de ADS Y con cardias a 39 cms de ADS.

ESTOMAGO. - con contenido claro con cardias incontinente con mucosa de fondo normal con cuerpo con pliegues leves congestivos antro con mucosa enanatematosa lineal, con píloro central franqueable.

BULBO Y DUODENO. - Con calibre, tamaño y elasticidad conservada, con mucosa normal.

Diagnósticos:

HERNIA HIATAL GASTROPATIA ENANTEMATOSA

Comentarios Adicionales:

Atentamente,

Dr. Anónimo

Anexo E

Código fuente para generación de informes

```
package Clases;

import java.io. File ;
import java.io. FileNotFoundException ;
import java.io. FilenameFilter ;
import java.io. PrintWriter ;
import java.util. ArrayList ;
import java.util. List ;
import java.util. logging. Level ;
import java.util. logging. Logger ;
import javafx.geometry. Pos ;
import javafx.util. Duration ;
import org.controlsfx.control. Notifications ;
import static saphiens2. Saphiens2. nombrePrograma ;
import word. utils. Utils ;

import word.w2004. elements. Image ;
public class DocumentoWord {

    List<String> list = new ArrayList<String>();

    public void crearDocumento(String numRegistro, String fecha,
        String enlace, String paciente, String estudio, String sexo,
        String edad, String HC, String cama, String ciPaciente,
        String CB, String drEncargado, String procedencia, String
        drRemitente, String empresa, String premedicacion, String
        diagRemision, String informe, String diagnostico, String
        numCita, int val, ArrayList lista) {

        list.add("@FotoA");
        list.add("@FotoB");
        list.add("@FotoC");
        list.add("@FotoD");
        list.add("@FotoE");
        list.add("@FotoF");
        list.add("@FotoG");
    }
}
```



```
list.add("@FotoH");
list.add("@FotoI");
list.add("@FotoJ");
list.add("@FotoK");
list.add("@FotoL");
list.add("@FotoM");
list.add("@FotoN");
list.add("@FotoO");

PrintWriter writer = null;
File archivo = new File("Informe.doc");
File file;
String path;
FilenameFilter filtro = null;
if (val == 1) {
    file = new File("Pacientes/" + numRegistro + "/Cita_" +
        numCita + "/Fotos");
    path = "Pacientes/" + numRegistro + "/Cita_" + numCita +
        "/Fotos/";
} else {
    file = new File("Biopsias");
    path = "Biopsias/";
}
boolean exists = file.exists();
String[] ficheros = null;

numRegistro = numRegistro.replace("&", "&");
fecha = fecha.replace("&", "&");
enlace = enlace.replace("&", "&");
paciente = paciente.replace("&", "&");
estudio = estudio.replace("&", "&");
sexo = sexo.replace("&", "&");
edad = edad.replace("&", "&");
HC = HC.replace("&", "&");
cama = cama.replace("&", "&");
ciPaciente = ciPaciente.replace("&", "&");
CB = CB.replace("&", "&");
drEncargado = drEncargado.replace("&", "&");
procedencia = procedencia.replace("&", "&");
drRemitente = drRemitente.replace("&", "&");
empresa = empresa.replace("&", "&");
premedicacion = premedicacion.replace("&", "&");
diagRemision = diagRemision.replace("&", "&");
informe = informe.replace("&", "&");
diagnostico = diagnostico.replace("&", "&");

numRegistro = numRegistro.replace("\", \"");
fecha = fecha.replace("\", \"");
enlace = enlace.replace("\", \"");
paciente = paciente.replace("\", \"");
estudio = estudio.replace("\", \"");
sexo = sexo.replace("\", \"");
edad = edad.replace("\", \"");
HC = HC.replace("\", \"");
cama = cama.replace("\", \"");
```




```
ciPaciente = ciPaciente.replace("\", "&quot;");  
CB = CB.replace("\", "&quot;");  
drEncargado = drEncargado.replace("\", "&quot;");  
procedencia = procedencia.replace("\", "&quot;");  
drRemitente = drRemitente.replace("\", "&quot;");  
empresa = empresa.replace("\", "&quot;");  
premedicacion = premedicacion.replace("\", "&quot;");  
diagRemision = diagRemision.replace("\", "&quot;");  
informe = informe.replace("\", "&quot;");  
diagnostico = diagnostico.replace("\", "&quot;");
```

```
numRegistro = numRegistro.replace("'", "&apos;");  
fecha = fecha.replace("'", "&apos;");  
enlace = enlace.replace("'", "&apos;");  
paciente = paciente.replace("'", "&apos;");  
estudio = estudio.replace("'", "&apos;");  
sexo = sexo.replace("'", "&apos;");  
edad = edad.replace("'", "&apos;");  
HC = HC.replace("'", "&apos;");  
cama = cama.replace("'", "&apos;");  
ciPaciente = ciPaciente.replace("'", "&apos;");  
CB = CB.replace("'", "&apos;");  
drEncargado = drEncargado.replace("'", "&apos;");  
procedencia = procedencia.replace("'", "&apos;");  
drRemitente = drRemitente.replace("'", "&apos;");  
empresa = empresa.replace("'", "&apos;");  
premedicacion = premedicacion.replace("'", "&apos;");  
diagRemision = diagRemision.replace("'", "&apos;");  
informe = informe.replace("'", "&apos;");  
diagnostico = diagnostico.replace("'", "&apos;");
```

```
numRegistro = numRegistro.replace("<", "&lt;");  
fecha = fecha.replace("<", "&lt;");  
enlace = enlace.replace("<", "&lt;");  
paciente = paciente.replace("<", "&lt;");  
estudio = estudio.replace("<", "&lt;");  
sexo = sexo.replace("<", "&lt;");  
edad = edad.replace("<", "&lt;");  
HC = HC.replace("<", "&lt;");  
cama = cama.replace("<", "&lt;");  
ciPaciente = ciPaciente.replace("<", "&lt;");  
CB = CB.replace("<", "&lt;");  
drEncargado = drEncargado.replace("<", "&lt;");  
procedencia = procedencia.replace("<", "&lt;");  
drRemitente = drRemitente.replace("<", "&lt;");  
empresa = empresa.replace("<", "&lt;");  
premedicacion = premedicacion.replace("<", "&lt;");  
diagRemision = diagRemision.replace("<", "&lt;");  
informe = informe.replace("<", "&lt;");  
diagnostico = diagnostico.replace("<", "&lt;");
```

```
numRegistro = numRegistro.replace(">", "&gt;");  
fecha = fecha.replace(">", "&gt;");  
enlace = enlace.replace(">", "&gt;");  
paciente = paciente.replace(">", "&gt;");
```



```
estudio = estudio.replace(">", "&gt;");
sexo = sexo.replace(">", "&gt;");
edad = edad.replace(">", "&gt;");
HC = HC.replace(">", "&gt;");
cama = cama.replace(">", "&gt;");
ciPaciente = ciPaciente.replace(">", "&gt;");
CB = CB.replace(">", "&gt;");
drEncargado = drEncargado.replace(">", "&gt;");
procedencia = procedencia.replace(">", "&gt;");
drRemitente = drRemitente.replace(">", "&gt;");
empresa = empresa.replace(">", "&gt;");
premedicacion = premedicacion.replace(">", "&gt;");
diagRemision = diagRemision.replace(">", "&gt;");
informe = informe.replace(">", "&gt;");
diagnostico = diagnostico.replace(">", "&gt;");

try {
    String xmlTemplate;
    xmlTemplate = Utils.readFile("plantilla.doc");
    xmlTemplate = replacePh(xmlTemplate, "@Registro",
        numRegistro);
    xmlTemplate = replacePh(xmlTemplate, "@Fecha", fecha);
    xmlTemplate = replacePh(xmlTemplate, "@Enlace", enlace);
    xmlTemplate = replacePh(xmlTemplate, "@NombrePaciente",
        paciente);
    xmlTemplate = replacePh(xmlTemplate, "@Estudio",
        estudio);
    xmlTemplate = replacePh(xmlTemplate, "@Sexo", sexo);
    xmlTemplate = replacePh(xmlTemplate, "@Edad", edad);
    xmlTemplate = replacePh(xmlTemplate, "@HC", HC);
    xmlTemplate = replacePh(xmlTemplate, "@Cama", cama);
    xmlTemplate = replacePh(xmlTemplate, "@CI", ciPaciente);
    xmlTemplate = replacePh(xmlTemplate, "@CB", CB);
    xmlTemplate = replacePh(xmlTemplate, "@MedicoEncargado",
        drEncargado);
    xmlTemplate = replacePh(xmlTemplate, "@Ciudad",
        procedencia);
    xmlTemplate = replacePh(xmlTemplate, "@MedicoRemitente",
        drRemitente);
    xmlTemplate = replacePh(xmlTemplate, "@Empresa",
        empresa);
    xmlTemplate = replacePh(xmlTemplate, "@Premedicacion",
        premedicacion);
    xmlTemplate = replacePh(xmlTemplate, "@DiagnosticoRem",
        diagRemision);
    xmlTemplate = replacePh(xmlTemplate, "@Informe",
        informe);
    xmlTemplate = replacePh(xmlTemplate, "@Diagnostico",
        diagnostico);
    int contador_imagenes = 0;

    if (exists) {
        if (val == 1) {
            filtro = (File directory, String fileName) ->
                fileName.endsWith(".jpg");
        }
    }
}
```



```
    } else {
        filtro = (File directory, String fileName) ->
            fileName.endsWith(".png");
    }
    ficheros = file.list(filtro);
    for (int i = 0; i < ficheros.length; i++) {
        String a = ficheros[i];
        if (val == 1) {
            if
                (((String.valueOf(lista.get(i))).equals("1"))
                && val == 1) {
                Image img =
                    Image.from_FULL_LOCAL_PATHL(path +
                    a).setHeight("100").setWidth("100");

                xmlTemplate = replacePh(xmlTemplate,
                    list.get(contador_imagenes),
                    img.getContent());

                contador_imagenes = contador_imagenes +
                    1;
            }
        } else {
            Image img = Image.from_FULL_LOCAL_PATHL(path
                + a).setHeight("100").setWidth("100");
            xmlTemplate = replacePh(xmlTemplate,
                list.get(contador_imagenes),
                img.getContent());
            contador_imagenes = contador_imagenes + 1;
        }
    }
}

while (contador_imagenes < 15) {
    xmlTemplate = replacePh(xmlTemplate,
        list.get(contador_imagenes), "");
    contador_imagenes = contador_imagenes + 1;
}

// Crea el archivo final
writer = new PrintWriter(archivo);
String myWord = xmlTemplate;
myWord = myWord.replace("UTF-8", "iso-8859-1");
myWord = myWord.replace(" ", " ");
myWord = myWord.replace("<", "<");
myWord = myWord.replace(">", ">");
writer.println(myWord);
writer.close();
} catch (FileNotFoundException ex) {
    Logger.getLogger(DocumentoWord.class.getName()).log(Level.SEVERE,
        null, ex);
} finally {
    try {
        writer.close();
    }
}
```



ANEXO E. CÓDIGO FUENTE PARA GENERACIÓN DE INFORMES

```
    } catch (Exception e) {
        Notifications notificationBuilder =
            Notifications.create()
                .title(nombrePrograma.getValue())
                .text("Existi un error al abrir el
                    documento de Word \n Verifiqu que no
                    est abierto el documento \n Guarde la
                    cita y vuelva a generar el informe....")
                .hideAfter(Duration.seconds(10))
                .position(Pos.CENTER);
        notificationBuilder.darkStyle();
        notificationBuilder.showError();
    }
}

public String replacePh(String base, String placeholder, String
value) {
    if (!base.contains(placeholder)) {
        System.out.println("### WARN: couldn't find the place
            holder: " + placeholder);
        return base;
    }
    return base.replace(placeholder, value);
}
}
```

UNIVERSIDAD DE CUENCA
desde 1867

Anexo F

Código fuente del software SIM 1.1

hibernate.cfg.xml

```
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate
Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property>
      <property
        name="hibernate.connection.driver_class"></property>
      <property name="hibernate.connection.url"></property>
      <property name="hibernate.connection.username"></property>
      <property name="hibernate.connection.password"></property>
      <property name="hibernate.c3p0.min_size">2</property>
      <property name="hibernate.c3p0.max_size">500</property>
      <property name="hibernate.c3p0.timeout">36000</property>
      <property name="hibernate.c3p0.max_statements">0</property>
    </property>
    name="hibernate.c3p0.idle_test_period">300</property>
  </property>
    name="hibernate.c3p0.acquire_increment">2</property>
  </property>
    <property name="hibernate.dialect"></property>
    <property name="hibernate.hbm2ddl.auto">update</property>
    <property name="show_sql">>false</property>
    <mapping class="Model.Classes.Examenes"/>
  </session-factory>
</hibernate-configuration>
```

AparienciaVentana.java

```
import com.sun.javafx.util.Utils;
import javafx.event.ActionEvent;
import javafx.geometry.Rectangle2D;
import javafx.scene.control.Button;
import javafx.scene.control.Separator;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.HBox;
import javafx.stage.Screen;
import javafx.stage.Stage;
import javafx.stage.StageStyle;
```



```
public class AparienciaVentana {
    Stage stage;
    Button btnMinimizar;
    Button btnMaximizar;
    Button btnCerrar;
    Button btnIcono;
    HBox barraTitulo;
    Separator separadorVertical;
    Separator separadorHorizontal;
    Separator separadorVH;
    Separator separadorHV;
    Rectangle2D backupWindowBounds = null;
    boolean maximized = false;
    boolean resizable = true;
    private double dragOffsetX = 0;
    private double dragOffsetY = 0;
    private double mouseDragOffsetX = 0;
    private double mouseDragOffsetY = 0;
    double stageMinimumWidth = 1280;
    double stageMinimumHeight = 800;
    int minWidth = -1;
    int minHeight = -1;
    public AparienciaVentana(Stage stag, Button btnMin, Button
        btnMax, Button btnClose, HBox barra,
        Separator separadorV, Separator separadorH, Separator
        separadorHV, Separator separadorVH) {
        this.stage = stag;
        stage.setOpacity(0.98);
        stage.initStyle(StageStyle.TRANSPARENT);
        btnMinimizar = btnMin;
        btnMinimizar.setOnAction((ActionEvent event) -> {
            stage.setIconified(true);
        });
        btnMaximizar = btnMax;
        btnMaximizar.setOnAction((ActionEvent event) -> {
            cambiarTamanoVentanaButton();
        });
        btnCerrar = btnClose;
        btnCerrar.setOnAction((ActionEvent event) -> {
            System.exit(0);
        });
        this.barraTitulo = barra;
        barraTitulo.setOnMouseClicked((MouseEvent event) -> {
            try {
                if (event.getClickCount() == 2) {
                    cambiarTamanoVentanaButton();
                }
            } catch (Exception ex) {
            }
        });
        barraTitulo.setOnMouseDragged((MouseEvent event) -> {
            if (maximized) {
                this.maximized = false;
                if (backupWindowBounds != null) {

```



```
        mouseDragOffsetX =
            ((backupWindowBounds.getWidth() *
             event.getSceneX()) / barraTitulo.getWidth());
        mouseDragOffsetY = event.getSceneY();
        this.stage.setWidth(backupWindowBounds.getWidth());
        this.stage.setHeight(backupWindowBounds.getHeight());
    }
} else {
    if (mouseDragOffsetX > barraTitulo.getWidth()) {
        mouseDragOffsetX = barraTitulo.getWidth() - 20;
    }
    stage.setX(event.getScreenX() - mouseDragOffsetX);
    stage.setY(event.getScreenY() - mouseDragOffsetY);
}
});
barraTitulo.setOnMousePressed((MouseEvent event) -> {
    mouseDragOffsetX = event.getSceneX();
    mouseDragOffsetY = event.getSceneY();
});
barraTitulo.setOnMouseReleased((MouseEvent event) -> {
    if (resizable) {
        Screen screen = (Screen)
            Screen.getScreensForRectangle(0, 0, 1.0D,
            1.0D).get(0);
        if (event.getScreenY() == 0) {
            this.maximized = true;
            backupWindowBounds = new
                Rectangle2D(this.stage.getX(),
                    this.stage.getY(), this.stage.getWidth(),
                    this.stage.getHeight());
            this.stage.setX(0);
            this.stage.setY(0);
            this.stage.setWidth(screen.getVisualBounds().getWidth());
            this.stage.setHeight(screen.getVisualBounds().getHeight());
        } else if (event.getScreenX() == 0) {
            this.maximized = true;
            backupWindowBounds = new
                Rectangle2D(this.stage.getX(),
                    this.stage.getY(), this.stage.getWidth(),
                    this.stage.getHeight());
            this.stage.setX(0);
            this.stage.setY(0);
            this.stage.setWidth(screen.getVisualBounds().getWidth()
                / 2);
            this.stage.setHeight(screen.getVisualBounds().getHeight());
        } else if ((event.getScreenX() <
            screen.getVisualBounds().getWidth() + 2) &&
            (event.getScreenX() >
            screen.getVisualBounds().getWidth() - 2)) {
            this.maximized = true;
            backupWindowBounds = new
                Rectangle2D(this.stage.getX(),
                    this.stage.getY(), this.stage.getWidth(),
                    this.stage.getHeight());
        }
    }
});
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
        this.stage.setX(screen.getVisualBounds().getWidth()
            / 2);
        this.stage.setY(0);
        this.stage.setWidth(screen.getVisualBounds().getWidth()
            / 2);
        this.stage.setHeight(screen.getVisualBounds().getHeight());
    }
}
});
this.separadorVertical = separadorV;
separadorVertical.setOnMouseDragged((MouseEvent event) -> {
    if (resizable) {
        final double stageY = Utils.isMac() ? stage.getY() +
            22 : stage.getY();
        final Screen screen =
            Screen.getScreensForRectangle(stage.getX(),
                stageY, 1, 1).get(0);
        Rectangle2D visualBounds = screen.getVisualBounds();
        if (Utils.isMac()) {
            visualBounds = new
                Rectangle2D(visualBounds.getMinX(),
                    visualBounds.getMinY() + 22,
                    visualBounds.getWidth(),
                    visualBounds.getHeight());
        }
        double maxY = Math.min(visualBounds.getMaxY(),
            event.getScreenY() - dragOffsetY);
        stage.setHeight(Math.max(stageMinimumHeight, maxY -
            stageY));
        event.consume();
    }
});
this.separadorHorizontal = separadorH;
separadorHorizontal.setOnMouseDragged((MouseEvent event) -> {
    if (resizable) {
        final double stageY = Utils.isMac() ? stage.getY() +
            22 : stage.getY();
        final Screen screen =
            Screen.getScreensForRectangle(stage.getX(),
                stageY, 1, 1).get(0);
        Rectangle2D visualBounds = screen.getVisualBounds();
        if (Utils.isMac()) {
            visualBounds = new
                Rectangle2D(visualBounds.getMinX(),
                    visualBounds.getMinY() + 22,
                    visualBounds.getWidth(),
                    visualBounds.getHeight());
        }
        double maxX = Math.min(visualBounds.getMaxX(),
            event.getScreenX() + dragOffsetX);
        stage.setWidth(Math.max(stageMinimumWidth, maxX -
            stage.getX()));
        event.consume();
    }
});
});
```




```
this.separadorHV = separatorHV;
separadorHV.setOnMouseDragged((MouseEvent event) -> {
    if (resizable) {
        final double stageY = Utils.isMac() ? stage.getY() +
            22 : stage.getY();
        final Screen screen =
            Screen.getScreensForRectangle(stage.getX(),
                stageY, 1, 1).get(0);
        Rectangle2D visualBounds = screen.getVisualBounds();
        if (Utils.isMac()) {
            visualBounds = new
                Rectangle2D(visualBounds.getMinX(),
                    visualBounds.getMinY() + 22,
                    visualBounds.getWidth(),
                    visualBounds.getHeight());
        }
        double maxX = Math.min(visualBounds.getMaxX(),
            event.getScreenX() + dragOffsetX);
        double maxY = Math.min(visualBounds.getMaxY(),
            event.getScreenY() - dragOffsetY);
        stage.setWidth(Math.max(stageMinimumWidth, maxX -
            stage.getX()));
        stage.setHeight(Math.max(stageMinimumHeight, maxY -
            stage.getY()));
        event.consume();
    }
});
this.separadorVH = separatorVH;
separadorVH.setOnMouseDragged((MouseEvent event) -> {
    if (resizable) {
        if ((minWidth == -1) && (minHeight == -1)) {
            final double stageY = Utils.isMac() ?
                stage.getY() + 22 : stage.getY();
            final Screen screen =
                Screen.getScreensForRectangle(stage.getX(),
                    stageY, 1, 1).get(0);
            Rectangle2D visualBounds =
                screen.getVisualBounds();
            if (Utils.isMac()) {
                visualBounds = new
                    Rectangle2D(visualBounds.getMinX(),
                        visualBounds.getMinY() + 22,
                        visualBounds.getWidth(),
                        visualBounds.getHeight());
            }
            double maxX = Math.min(visualBounds.getMaxX(),
                event.getScreenX() + dragOffsetX);
            double maxY = Math.min(visualBounds.getMaxY(),
                event.getScreenY() - dragOffsetY);
            stage.setWidth(Math.max(stageMinimumWidth, maxX
                - stage.getX()));
            stage.setHeight(Math.max(stageMinimumHeight,
                maxY - stage.getY()));
            event.consume();
        }
    }
});
```



```
    }
  });
}
private void cambiarTamanoVentanaButton() {
  Screen screen = (Screen)
  Screen.getScreensForRectangle(this.stage.getX(),
  this.stage.getY(), 1.0D, 1.0D).get(0);
  if (resizable) {
    if (maximized) {
      this.maximized = false;
      if (backupWindowBounds != null) {
        this.stage.setX(backupWindowBounds.getMinX());
        this.stage.setY(backupWindowBounds.getMinY());
        this.stage.setWidth(backupWindowBounds.getWidth());
        this.stage.setHeight(backupWindowBounds.getHeight());
      }
    } else {
      this.maximized = true;
      backupWindowBounds = new
      Rectangle2D(this.stage.getX(),
      this.stage.getY(), this.stage.getWidth(),
      this.stage.getHeight());
      this.stage.setX(screen.getVisualBounds().getMinX());
      this.stage.setY(screen.getVisualBounds().getMinY());
      this.stage.setWidth(screen.getVisualBounds().getWidth());
      this.stage.setHeight(screen.getVisualBounds().getHeight());
    }
  }
}
public void setResizable(Boolean dato) {
  if (dato == false) {
    btnMaximizar.setDisable(true);
    btnMinimizar.setDisable(true);
    this.resizable = false;
  } else {
    btnMaximizar.setDisable(false);
    btnMinimizar.setDisable(false);
    this.resizable = true;
  }
}
public void setWidth(double valor) {
  stage.setWidth(valor);
}
public void setHeight(double valor) {
  stage.setHeight(valor);
}
public void setMaximized(boolean valor) {
  Screen screen = (Screen) Screen.getScreensForRectangle(0, 0,
  1.0D, 1.0D).get(0);
  if (valor) {
    this.maximized = true;
    backupWindowBounds = new
    Rectangle2D(((screen.getVisualBounds().getWidth() -
    stageMinimumWidth) / 2),
    ((screen.getVisualBounds().getHeight() -
```



```
        stageMinimumHeight) / 2), stageMinimumWidth,
        stageMinimumHeight);
    this.stage.setX(screen.getVisualBounds().getMinX());
    this.stage.setY(screen.getVisualBounds().getMinY());
    this.stage.setWidth(screen.getVisualBounds().getWidth());
    this.stage.setHeight(screen.getVisualBounds().getHeight());
    }
}
public void setMinHeight(double valor) {
    stage.setMinHeight(valor);
}
public void setMinWidth(double valor) {
    stage.setMinWidth(valor);
}
}
```

Archivo.java

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.util.Properties;
import java.util.logging.Level;
import java.util.logging.Logger;
public class Archivo {
    public String leerArchivoProperties(String direccion, String
        nombreArchivo, String campo) {
        Properties informacion = new Properties();
        try {
            try (FileInputStream in = new FileInputStream(direccion
                + "/" + nombreArchivo + ".properties")) {
                informacion.load(in);
                in.close();
                return informacion.getProperty(campo);
            }
        } catch (Exception ex) {
            return null;
        }
    }
    public void escribirArchivoProperties(String direccion, String
        nombreArchivo, String campo, String dato) {
        FileOutputStream out = null;
        Properties info = new Properties();
        try {
            FileInputStream in = new FileInputStream(direccion + "/"
                + nombreArchivo + ".properties");
            info.load(in);
            out = new FileOutputStream(direccion + "/" +
                nombreArchivo + ".properties");
            info.setProperty(campo, dato);
            info.store(out, null);
        } catch (Exception ex) {
            Logger.getLogger(Archivo.class.getName()).log(Level.SEVERE,
                null, ex);
        }
    }
}
```



```
}
```

CRUD.java

```
import java.util.List;
import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.Transaction;
public class CRUD {
    private Session sesion;
    private Transaction tx;
    public Boolean iniciaOperacion() {
        try {
            sesion = HibernateUtil.getSessionFactory().openSession();
            tx = sesion.beginTransaction();
            return true;
        } catch (Exception e) {
            return false;
        }
    }
    private void manejaExcepcion(HibernateException he) throws
        HibernateException {
        tx.rollback();
        throw new HibernateException("Ocurri un error en la capa
            de acceso a datos", he);
    }
    public Boolean guardarObjeto(Object objeto) {
        try {
            iniciaOperacion();
            sesion.persist(objeto);
            tx.commit();
            return true;
        } catch (HibernateException he) {
            manejaExcepcion(he);
            throw he;
        } finally {
            sesion.close();
        }
    }
    public Boolean actualizarObjeto(Object objeto) throws
        HibernateException {
        try {
            iniciaOperacion();
            sesion.update(objeto);
            tx.commit();
            return true;
        } catch (HibernateException he) {
            manejaExcepcion(he);
            throw he;
        } finally {
            sesion.close();
        }
    }
    public Boolean eliminaContacto(Object objeto) throws
        HibernateException {
```



```
        try {
            iniciaOperacion();
            sesion.delete(objeto);
            tx.commit();
            return true;
        } catch (HibernateException he) {
            manejaExcepcion(he);
            throw he;
        } finally {
            sesion.close();
        }
    }
}

public Object obtenerObjeto(Class clase, String idContacto)
throws HibernateException {
    Object contacto = null;
    try {
        iniciaOperacion();
        contacto = (Object) sesion.get(clase, idContacto);
    } finally {
        sesion.close();
    }
    return contacto;
}

public List<Object> obtenerLista(String tabla) throws
HibernateException {
    List<Object> listaContactos = null;
    try {
        iniciaOperacion();
        listaContactos = sesion.createQuery("from " +
            tabla).list();
    } finally {
        sesion.close();
    }
    return listaContactos;
}

public List<Object> obtenerLista(Class clase) throws
HibernateException {
    List<Object> listaObjetos = null;
    try {
        iniciaOperacion();
        listaObjetos = sesion.createQuery("from " +
            clase.getName()).list();
    } finally {
        sesion.close();
    }
    return listaObjetos;
}
}
```

HibernateUtil.java

```
import static Model.Prototipo1.direccion;
import static Model.Prototipo1.nombreArchivo;
import java.util.Properties;
import org.hibernate.HibernateException;
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
import org.hibernate.SessionFactory;
import org.hibernate.cfg.AnnotationConfiguration;
import org.hibernate.cfg.Configuration;
public class HibernateUtil {
    private static SessionFactory sessionFactory;
    static {
        try {
            Properties properties = new Properties();
            properties.setProperty("hibernate.dialect",
                "org.hibernate.dialect.MySQLDialect");
            properties.setProperty("hibernate.connection.driver_class",
                "com.mysql.jdbc.Driver");
            if (Integer.valueOf(new
                Archivo().leerArchivoProperties(direccion,
                nombreArchivo, "tipoConexionDB")) == 0) {
                properties.setProperty("hibernate.connection.url",
                    "jdbc:mysql://"
                    + new
                        Archivo().leerArchivoProperties(direccion,
                        nombreArchivo, "ipDB")
                    + "/" + new
                        Archivo().leerArchivoProperties(direccion,
                        nombreArchivo, "nombreDB"));
            } else {
                properties.setProperty("hibernate.connection.url",
                    "jdbc:mysql://"
                    + new
                        Archivo().leerArchivoProperties(direccion,
                        nombreArchivo, "ipDB")
                    + ":" + new
                        Archivo().leerArchivoProperties(direccion,
                        nombreArchivo, "puertoDB")
                    + "/" + new
                        Archivo().leerArchivoProperties(direccion,
                        nombreArchivo, "nombreDB"));
            }
            properties.setProperty("hibernate.connection.username",
                new Archivo().leerArchivoProperties(direccion,
                nombreArchivo, "usuarioDB"));
            properties.setProperty("hibernate.connection.password",
                new Archivo().leerArchivoProperties(direccion,
                nombreArchivo, "claveDB"));
            Configuration cfg = new
                AnnotationConfiguration().configure();
            cfg.addProperties(properties);
            sessionFactory = cfg.buildSessionFactory();
        } catch (NumberFormatException | HibernateException ex) {
            System.err.println("Error en HibernateUtil.java.");
            throw new ExceptionInInitializerError(ex);
        }
    }
    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}
```



MySQL.java

```
import java.io.Serializable;
import java.sql.*;
public class MySQL implements Serializable {
    public static Connection conect;
    public Statement st;
    String ipServidor;
    String nombreDB;
    String nombre;
    String clave;
    String puerto;
    public MySQL() {
    }
    public MySQL(String ipServidor, String nombreDB, String nombre,
        String clave, String puerto) {
        this.ipServidor = ipServidor;
        this.nombreDB = nombreDB;
        this.nombre = nombre;
        this.clave = clave;
        this.puerto = puerto;
    }
    public MySQL(String ipServidor, String nombreDB, String nombre,
        String clave) {
        this.ipServidor = ipServidor;
        this.nombreDB = nombreDB;
        this.nombre = nombre;
        this.clave = clave;
    }
    public boolean verificarConexionLocal() {
        try {
            conect = DriverManager.getConnection("jdbc:mysql://" +
                ipServidor + ":" + puerto + "/", nombre, clave);
            return true;
        } catch (SQLException ex) {
            return false;
        }
    }
    public boolean verificarConexionRemota() {
        try {
            conect = DriverManager.getConnection("jdbc:mysql://" +
                ipServidor + "/", nombre, clave);
            return true;
        } catch (SQLException ex) {
            return false;
        }
    }
    public boolean verificarConexionLocal(String nombre, String
        clave, String puerto) {
        try {
            conect = DriverManager.getConnection("jdbc:mysql://" +
                ipServidor + ":" + puerto + "/", nombre, clave);
            return true;
        } catch (SQLException ex) {
            return false;
        }
    }
}
```



```
}
public boolean crearDB() {
    try {
        st = conect.createStatement();
        st.executeUpdate("CREATE DATABASE " + nombreDB);
        st.close();
        return true;
    } catch (Exception ex) {
        return false;
    }
}

public boolean crearDB(String nombreDataBase) {
    try {
        st = conect.createStatement();
        st.executeUpdate("CREATE DATABASE " + nombreDataBase);
        st.close();
        return true;
    } catch (SQLException ex) {
        return false;
    }
}

public String getNombreDB() {
    return nombreDB;
}

public void setNombreDB(String nombreDB) {
    this.nombreDB = nombreDB;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public String getClave() {
    return clave;
}

public void setClave(String clave) {
    this.clave = clave;
}

public Boolean borrarDB(String nombreDB) {
    try {
        st = conect.createStatement();
        st.executeUpdate("DROP DATABASE " + nombreDB);
        st.close();
        return true;
    } catch (SQLException ex) {
        return false;
    }
}

public boolean reemplazarDB(String nombreDB) {
    if (borrarDB(nombreDB)) {
        if (crearDB()) {
            return true;
        } else {
            return false;
        }
    }
}
```




```
    }  
  } else {  
    return false;  
  }  
}  
}
```

Prototipo1.java

```
import Model.Classes.Examenes;  
import javafx.application.Application;  
import javafx.stage.Stage;  
public class Prototipo1 extends Application {  
    public static final String direccion = "inf/prop";  
    public static final String nombreArchivo = "ini";  
    public static Examenes examenHecho = new Examenes();  
    public static String imagen="";  
    @Override  
    public void start(Stage stage) throws Exception {  
        new Ventana().abrirVentana("splash", 2);  
        stage.close();  
    }  
    public static void main(String[] args) {  
        launch(args);  
    }  
}
```

Ventana.java

```
import static Model.Prototipo1.direccion;  
import static Model.Prototipo1.nombreArchivo;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Label;  
import javafx.scene.image.Image;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
public class Ventana {  
    public void abrirVentana(String nombreVentana) {  
        try {  
            Parent root =  
                FXMLLoader.load(getClass().getResource("/View/" +  
                    nombreVentana + ".fxml"));  
            Scene scene = new Scene(root);  
            Stage stage = new Stage();  
            stage.setScene(scene);  
            scene.getStylesheets().add(this.getClass().getResource  
                ("/Style/estilo" + new  
                Archivo().leerArchivoProperties(direccion ,  
                nombreArchivo, "estilo" + ".css")  
                .toExternalForm());  
            stage.initStyle(StageStyle.DECORATED);  
        }  
    }  
}
```



```
        stage.getIcons().add(new Image("Images/" + new
            Archivo().leerArchivoProperties(direccion,
                nombreArchivo, "icono")));
        stage.show();
    } catch (Exception ex) {
        System.out.println("Existi un error al abrir la
            ventana " + nombreVentana + ".fxml");
    }
}
}
public void abrirVentana(String nombreVentana, int estilo) {
    try {
        Parent root =
            FXMLLoader.load(getClass().getResource("/View/" +
                nombreVentana + ".fxml"));
        Scene scene = new Scene(root);
        Stage stage = new Stage();
        stage.setScene(scene);
        scene.getStylesheets().add(this.getClass().getResource
            ("/Style/estilo" + new
                Archivo().leerArchivoProperties(direccion,
                    nombreArchivo, "estilo") + ".css").toExternalForm());
        switch (estilo) {
            case 1:
                stage.initStyle(StageStyle.DECORATED);
                break;
            case 2:
                stage.initStyle(StageStyle.UNDECORATED);
                break;
            case 3:
                stage.initStyle(StageStyle.TRANSPARENT);
                break;
            default:
                stage.initStyle(StageStyle.DECORATED);
        }
        stage.getIcons().add(new Image("Images/" + new
            Archivo().leerArchivoProperties(direccion,
                nombreArchivo, "icono")));
        stage.show();
    } catch (Exception ex) {
        System.out.println("Existi un error al abrir la
            ventana " + nombreVentana + ".fxml");
    }
}
}
public void abrirVentana(Stage stage, String nombreVentana, int
    estilo) {
    try {
        Parent root = FXMLLoader.load(getClass().getResource
            ("/View/" + nombreVentana + ".fxml"));
        Scene scene = new Scene(root);
        stage.setScene(scene);
        scene.getStylesheets().add(this.getClass().getResource
            ("/Style/estilo" + new
                Archivo().leerArchivoProperties(direccion,
                    nombreArchivo, "estilo") + ".css").toExternalForm());
        switch (estilo) {
```



```
        case 1:
            stage.initStyle(StageStyle.DECORATED);
            break;
        case 2:
            stage.initStyle(StageStyle.UNDECORATED);
            break;
        case 3:
            stage.initStyle(StageStyle.TRANSPARENT);
            break;
        default:
            stage.initStyle(StageStyle.DECORATED);
    }
    stage.getIcons().add(new Image("Images/" + new
        Archivo().leerArchivoProperties(direccion,
        nombreArchivo, "icono")));
    stage.show();
} catch (Exception ex) {
    System.out.println("Existi un error al abrir la
        ventana " + nombreVentana + ".FXML");
}
}
public void abrirVentana(Stage stage, String nombreVentana,
    StageStyle estilo, Object controlador) {
    try {
        FXMLLoader loader = new
            FXMLLoader(getClass().getResource ("/View/" +
            nombreVentana + ".FXML")); // Se instancia la
            escena.
        loader.setController(controlador);
        Parent root = loader.load();
        Scene scene = new Scene(root);
        stage.setScene(scene);
        scene.getStylesheets().add(this.getClass().getResource
            ("/Style/estilo" + new
            Archivo().leerArchivoProperties(direccion,
            nombreArchivo, "estilo") + ".css").toExternalForm());
        stage.initStyle(estilo);
        stage.getIcons().add(new Image("Images/" + new
            Archivo().leerArchivoProperties(direccion,
            nombreArchivo, "icono")));
        stage.show();
    } catch (Exception ex) {
        System.out.println("Existi un error al abrir la
            ventana " + nombreVentana + ".FXML");
    }
}
public void cerrarVentana(Label objeto) {
    Stage stage;
    stage = (Stage) objeto.getScene().getWindow();
    stage.close();
}
}
```

Config.java



```
import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
@Entity
@Table(name = "Config")
public class Config implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int idConfig;
    private int numeroDespacho;
    public Config() {
    }
    public int getIdConfig() {
        return idConfig;
    }
    public void setIdConfig(int idConfig) {
        this.idConfig = idConfig;
    }
    public int getNumeroDespacho() {
        return numeroDespacho;
    }
    public void setNumeroDespacho(int numeroDespacho) {
        this.numeroDespacho = numeroDespacho;
    }
}
```

Exámenes.java

```
import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
@Entity
@Table(name = "exámenes")
public class Exámenes implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int idExamen;
    private String premedicacion;
    private String diagnosticoRemision;
    private String informe;
    private String diagnostico;
    private String imagenes;
    private String estudio;
    private String doctorEncargado;
    private String fecha;
    private String edadPaciente;
    private String lugarPaciente;
    public Exámenes() {
    }
}
```



```
public int getIdExamen() {
    return idExamen;
}
public void setIdExamen(int idExamen) {
    this.idExamen = idExamen;
}
public String getPremedicacion() {
    return premedicacion;
}
public void setPremedicacion(String premedicacion) {
    this.premedicacion = premedicacion;
}
public String getDiagnosticoRemision() {
    return diagnosticoRemision;
}
public void setDiagnosticoRemision(String diagnosticoRemision) {
    this.diagnosticoRemision = diagnosticoRemision;
}
public String getInforme() {
    return informe;
}
public void setInforme(String informe) {
    this.informe = informe;
}
public String getDiagnostico() {
    return diagnostico;
}
public void setDiagnostico(String diagnostico) {
    this.diagnostico = diagnostico;
}
public String getImagenes() {
    return imagenes;
}
public void setImagenes(String imagenes) {
    this.imagenes = imagenes;
}
public String getEstudio() {
    return estudio;
}
public void setEstudio(String estudio) {
    this.estudio = estudio;
}
public String getDoctorEncargado() {
    return doctorEncargado;
}
public void setDoctorEncargado(String doctorEncargado) {
    this.doctorEncargado = doctorEncargado;
}
public String getFecha() {
    return fecha;
}
public void setFecha(String fecha) {
    this.fecha = fecha;
}
public String getEdadPaciente() {
```



```
        return edadPaciente;
    }
    public void setEdadPaciente(String edadPaciente) {
        this.edadPaciente = edadPaciente;
    }
    public String getLugarPaciente() {
        return lugarPaciente;
    }
    public void setLugarPaciente(String lugarPaciente) {
        this.lugarPaciente = lugarPaciente;
    }
}
```

Gestor.java

```
import Model.CRUD;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.concurrent.Service;
import javafx.concurrent.Task;
import javafx.concurrent.WorkerStateEvent;
import javafx.event.EventHandler;
import javafx.scene.layout.StackPane;
import org.controlsfx.control.MaskerPane;
import org.hibernate.HibernateException;
public class Gestor {
    private ObservableList<Exámenes> listaExámenes =
        FXCollections.observableArrayList();
    public Gestor() {
    }
    public ObservableList<Exámenes> getListaExámenes() {
        return listaExámenes;
    }
    public void actualizarListaExámenes() {
        Service<Integer> service = new Service<Integer>() {
            @Override
            protected Task<Integer> createTask() {
                return new Task<Integer>() {
                    @Override
                    protected Integer call() throws Exception {
                        listaExámenes.clear();
                        try {
                            for (Object prod : new
                                CRUD().obtenerLista(new
                                    Exámenes().getClass())) {
                                listaExámenes.add((Exámenes) prod);
                            }
                        } catch (HibernateException |
                            NumberFormatException e) {
                        }
                        return null;
                    }
                };
            }
        };
    }
};
```



```
        service.start();
    }
    public void esperar(StackPane s) {
        Service<Integer> service = new Service<Integer>() {
            @Override
            protected Task<Integer> createTask() {
                return new Task<Integer>() {
                    @Override
                    protected Integer call() throws Exception {
                        MaskerPane masker = new MaskerPane();
                        s.getChildren().add(masker);
                        masker.setText("Cargando...\n Espere por
                                    favor.");
                        masker.setVisible(true);
                        s.toFront();
                        return null;
                    }
                };
            }
        };
        service.start();
        service.setOnRunning((WorkerStateEvent event) -> {
        });
        service.setOnSucceeded((WorkerStateEvent event) -> {
        });
        service.setOnFailed((WorkerStateEvent event) -> {
        });
    }
    public void noEsperar(StackPane s) {
        s.getChildren().removeAll();
        s.toBack();
    }
    public void actualizarListaExamenes(StackPane s) {
        MaskerPane masker = new MaskerPane();
        s.getChildren().add(masker);
        masker.setText("Cargando...\n Espere por favor.");
        masker.setVisible(true);
        s.toFront();
        Service<Integer> service = new Service<Integer>() {
            @Override
            protected Task<Integer> createTask() {
                return new Task<Integer>() {
                    @Override
                    protected Integer call() throws Exception {
                        listaExamenes.clear();
                        try {
                            for (Object prod : new
                                CRUD().obtenerLista(new
                                    Examenes().getClass())) {
                                listaExamenes.add((Examenes) prod);
                            }
                        } catch (HibernateException |
                            NumberFormatException e) {
                        }
                        return null;
                    }
                };
            }
        };
    }
}
```



```
        }
    };
}
};
service.start();
service.setOnSucceeded(new EventHandler<WorkerStateEvent>() {
    @Override
    public void handle(WorkerStateEvent event) {
        s.toBack();
    }
});
}
public void mantenerConexion() {
    new Thread(() -> {
        while (true) {
            try {
                ((Exámenes) new CRUD().obtenerLista(new
                    Exámenes().getClass().get(0)).getDiagnostico());
                System.out.println("hilo");
                Thread.sleep(20000);
            } catch (HibernateException | InterruptedException
                e) {
            }
        }
    }).start();
}
}
```

Usuario.java

```
import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
@Entity
@Table(name = "Usuarios")
public class Usuario implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int idUsuario;
    private String nombreUsuario;
    private String cargoUsuario;
    private String passwordUsuario;
    private Boolean verDatosEmpresa;
    private Boolean modDatosEmpresa;
    private Boolean verDatosUsuario;
    private Boolean modDatosUsuario;
    private Boolean verInventario;
    private Boolean modInventario;
    private Boolean verPedidos;
    private Boolean hacerPedidos;
    private Boolean modificarPedidos;
    public Usuario() {
```




```
}
public int getIdUsuario() {
    return idUsuario;
}
public void setIdUsuario(int id) {
    this.idUsuario = id;
}
public String getCargoUsuario() {
    return cargoUsuario;
}
public void setCargodUsuario(String cargo) {
    this.cargoUsuario = cargo;
}
public String getNombreUsuario() {
    return nombreUsuario;
}
public void setNombreUsuario(String nombreUsuario) {
    this.nombreUsuario = nombreUsuario;
}
public String getPasswordUsuario() {
    return passwordUsuario;
}
public void setPasswordUsuario(String passwordUsuario) {
    this.passwordUsuario = passwordUsuario;
}
public Boolean getVerDatosEmpresa() {
    return verDatosEmpresa;
}
public void setVerDatosEmpresa(Boolean verDatosEmpresa) {
    this.verDatosEmpresa = verDatosEmpresa;
}
public Boolean getModDatosEmpresa() {
    return modDatosEmpresa;
}
public void setModDatosEmpresa(Boolean modDatosEmpresa) {
    this.modDatosEmpresa = modDatosEmpresa;
}
public Boolean getVerDatosUsuario() {
    return verDatosUsuario;
}
public void setVerDatosUsuario(Boolean verDatosUsuario) {
    this.verDatosUsuario = verDatosUsuario;
}
public Boolean getModDatosUsuario() {
    return modDatosUsuario;
}
public void setModDatosUsuario(Boolean modDatosUsuario) {
    this.modDatosUsuario = modDatosUsuario;
}
public Boolean getVerInventario() {
    return verInventario;
}
public void setVerInventario(Boolean verInventario) {
    this.verInventario = verInventario;
}
}
```



```
public Boolean getModInventario() {
    return modInventario;
}
public void setModInventario(Boolean modInventario) {
    this.modInventario = modInventario;
}
public Boolean getVerPedidos() {
    return verPedidos;
}
public void setVerPedidos(Boolean verPedidos) {
    this.verPedidos = verPedidos;
}
public Boolean getHacerPedidos() {
    return hacerPedidos;
}
public void setHacerPedidos(Boolean hacerPedidos) {
    this.hacerPedidos = hacerPedidos;
}
public Boolean getModificarPedidos() {
    return modificarPedidos;
}
public void setModificarPedidos(Boolean modificarPedidos) {
    this.modificarPedidos = modificarPedidos;
}
}
```

estilo1.css

```
}
* {
    -colorBorde1: rgba(0,89,169,0.6);
    -colorBorde2: rgba(0,0,0,0.1);
    -colorFondo: #2b579a;
}
.cerrarVentana1 {
    -fx-alignment: CENTER;
    -fx-text-fill: white;
    -fx-font-size: 15px;
}
.cerrarVentana1:hover {
    -fx-text-fill: red;
    -fx-font-size: 19px;
    -fx-font-weight: bold;
}
.cerrarVentana1:pressed {
    -fx-text-fill: white;
    -fx-font-size: 19px;
    -fx-font-weight: bold;
}
.popover > .content {
    -fx-background-color:
        radial-gradient(center 1% 160%, radius
            100%, rgba(0,89,169,0.1), rgba(255,255,255,0.9));
}
```



```
.popover > .content > .accordion > .titled-pane > .title >
  .arrow-button > .arrow {
  -fx-background-color: red;
  -fx-pref-height: 34.0 !important;
}
.popover > .border {
  -fx-stroke-width: 0;
  -fx-fill: rgba(240,240,240,0.8) !important;
  -fx-effect: dropshadow(gaussian, lightgray, 6.0, 0.0, 1.0, 1.0);
}
.tabla_datos {
  -fx-control-inner-background: #ffffff;
  -fx-background-color: #ffffff;
  -fx-table-cell-border-color: transparent;
  -fx-table-header-border-color: transparent;
  -fx-padding: 5;
}
.tabla_datos .column-header-background {
  -fx-background-color: transparent;
}
.tabla_datos .column-header, .tabla_datos .filler {
  -fx-size: 12pt;
  -fx-border-width: 0 0 1 0;
  -fx-background-color: transparent;
  -fx-border-color:
    transparent
    transparent
    derive(-fx-base, 80%)
    transparent;
  -fx-border-insets: 0 10 1 0;
}
.tabla_datos .column-header .label {
  -fx-font-size: 10pt;
  -fx-font-family: "Segoe UI Light";
  -fx-text-fill: blue;
  -fx-alignment: CENTER;
  -fx-opacity: 1;
}
.tabla_datos:focused .table-row-cell:filled:focused:selected {
  -fx-background-color: -fx-focus-color;
}
.labelMensaje{
  -fx-background-radius: 10;
  -fx-background-color: red;
  -fx-alignment: CENTER;
  -fx-font-size: 10pt;
}
.fondoVentanaSplash{
  -fx-background-color: -colorFondo;
}
.textFieldLogin{
  -fx-skin: "Model.Style.MetroTextFieldSkin";
}
.text-input:focused{
  -fx-background-color: #5c5c5c, white;
}
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
-fx-text-fill: black;
}
.text-input > .right-button{
    -fx-cursor: default;
    -fx-background-insets: -0.1666665em -0.45em -0.1666665em -0.45em;
}
.text-field > .right-button > .right-button-graphic {
    -fx-shape : "M221.738,305.87316.135,6.161-2.875,2.8631
                -6.135-6.1591-6.263,6.2371-2.864-2.87516.263-6.2381-6.177-
                6.20212.875-2.86316.177,6.20116.244-6.2212.864,2.876L221.738,305.873z";
    -fx-padding: 0.5em 0.5em; /* TODO: put the real number here 18
    18 */
    -fx-background-color: black;
}
.text-input > .right-button:hover {
    -fx-background-color: #dedede;
}
.text-input > .right-button:pressed {
    -fx-background-color: black;
}
.text-input > .right-button:pressed > .right-button-graphic {
    -fx-background-color: white;
}
.text-input:disabled {
    -fx-opacity: 1;
    -fx-background-insets: 0, 2;
    -fx-background-color: #e7e6e6;
    -fx-text-fill: #999999;
}
.item-title.disabled{
    -fx-text-fill: #939393;
}
.passwordFieldLogin {
    -fx-skin: "Model.Style.MetroPasswordFieldSkin";
}
.passwordFieldLogin > .right-button > .right-button-graphic {
    -fx-shape : "M307.688,399.564c0,1.484-1.203,2.688-
                2.688,2.688c-1.484,0-2.688-1.203-2.688-2.688s1.203-2.688,2.688-
                2.688C306.484,396.876,307.688,398.08,307.688,399.564z
                M297.5,
                399h2.5c0,
                0,1.063-4,5-4c3.688,0,5,4,5,4h2.5c0,0-2.063-6.5-
                7.5-6.5C299,392.5,297.5,399,297.5,399z";
    -fx-scale-shape: false;
    -fx-background-color: black;
}
.separador *.line{
    -fx-border-style: solid;
    -fx-border-width: 2;
    -fx-border-color: -colorBorde1;
}
.separador2 *.line{
    -fx-border-style: solid;
    -fx-border-width: 2;
```



```
-fx-border-color: linear-gradient(to top,
    -colorBorde1, -colorBorde2);
}
.backgroundVPrincipal {
    -fx-background-image: url("../Images/abstract.png");
    -fx-background-repeat: repeat;
    -fx-background-color:
        linear-gradient(#38424b 0%, #1f2429 20%, #191d22 100%),
        linear-gradient(#20262b, #191d22),
        radial-gradient(center 50% 50%, radius 100%,
            rgba(114,131,148,0.9), rgba(255,255,255,0));
}
.barraDeTitulo {
    -fx-background-color: linear-gradient(to right,
        -colorBorde1, -colorBorde2);
    -fx-background-radius: 0;
}
.cerrarVentana {
    -fx-background-color: transparent;
    -fx-background-image: url("../Images/close.png");
    -fx-background-position: center;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}
.cerrarVentana:hover {
    -fx-background-color: transparent;
    -fx-background-image: url("../Images/close-hover.png");
    -fx-background-position: center;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}
.maximizarVentana {
    -fx-background-color: transparent;
    -fx-background-image: url("../Images/maximize.png");
    -fx-background-position: center;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}
.maximizarVentana:hover {
    -fx-background-color: transparent;
    -fx-background-image: url("../Images/maximize-hover.png");
    -fx-background-position: center;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}
.minimizarVentana {
    -fx-background-color: transparent;
    -fx-background-image: url("../Images/minimize.png");
    -fx-background-position: center;
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
}
.minimizarVentana:hover {
    -fx-background-color: transparent;
    -fx-background-image: url("../Images/minimize-hover.png");
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
-fx-background-position: center;
-fx-background-repeat: no-repeat;
-fx-background-position:center;
}
.iconoVentana {
-fx-background-color: transparent;
-fx-background-image: url("../Images/logo.png");
-fx-background-position: center;
-fx-background-repeat: no-repeat;
-fx-background-position:center;
}
.tooltip {
-fx-background-color: rgba(80,80,80,0.8);
-fx-background-radius:0;
-fx-text-fill: white;
}
* {
-colorBotonHover: #3b6bb7;
-colorBotonPressed: #0b3b87;
-colorFondo:-colorBorde1;
}
.vbox{
-fx-background-color: -colorFondo;
-fx-padding: 0 ;
}
.separador3 *.line{
-fx-border-style: solid;
-fx-border-width: 2;
-fx-border-color:-colorFondo;
}
.fotoAdministrador {
-fx-shape: "M 50,30 m 0,25 a 1,1 0 0,0 0,-50 a 1,1 0 1,0
0,50";
-fx-border-color: white;
-fx-border-width: 2;
-fx-background-color:transparent;
}
.fotoAdmin {
-fx-effect: dropshadow(three-pass-box, rgba(0,0,0,0.8), 10, 0,
0, 0);
-fx-padding: 10;
-fx-background-color: firebrick;
-fx-background-radius: 5;
}
.fondoFoto{
-fx-background-image: url("../Images/foto.jpg");
-fx-background-position: left;
-fx-background-size: 138 160;
-fx-background-repeat:no-repeat;
-fx-opacity: 0.5;
}
.botonnes {
-fx-background-color: #28559c, #28559c;
-fx-background-radius: 0;
-fx-text-fill: white ;
```



```
-fx-padding: 0 ;
}
.botonInstituto:pressed {
    -fx-background-color: transparent ;
}
.botonInstituto {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/Instituto.png");
    -fx-background-repeat: no-repeat;
    -fx-background-position: 115px center;

    -fx-padding: 0 ;
}
.botonInstituto: hover {
    -fx-background-color: -colorBotonHover;
}
.botonInstituto:pressed {
    -fx-background-color: -colorBotonPressed;
}
.botonActualizar {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-repeat: no-repeat;
    -fx-background-position: 115px center;
    -fx-padding: 0 ;
}
.botonActualizar: hover {
    -fx-background-color: rgba(0,0,0,0.1);
}
.botonActualizar:pressed {
    -fx-background-color: rgba(0,0,0,0.5);
}
.botonDocentes {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/Profesor.png");
    -fx-background-repeat: no-repeat;
    -fx-background-position: 115px center;
    -fx-padding: 0 ;
}
.botonDocentes: hover {
    -fx-background-color: -colorBotonHover;
}
.botonDocentes:pressed {
    -fx-background-color: -colorBotonPressed;
}
```



```
}
.botonAlumnos {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/ico_pedidos.png");
    -fx-background-repeat: no-repeat;
    -fx-background-position: 115px center;
    -fx-padding: 0 ;
}
.botonAlumnos: hover {
    -fx-background-color: -colorBotonHover;
}
.botonAlumnos: pressed {
    -fx-background-color: -colorBotonPressed;
}
.botonAsignaturas {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/Asignatura.png");
    -fx-background-repeat: no-repeat;
    -fx-background-position: 115px center;
    -fx-padding: 0 ;
}
.botonAsignaturas: hover {
    -fx-background-color: -colorBotonHover;
}
.botonAsignaturas: pressed {
    -fx-background-color: -colorBotonPressed;
}
.botonMatriculas {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/Inscripcion.png");
    -fx-background-repeat: no-repeat;
    -fx-background-position: 115px center;
    -fx-padding: 0 ;
}
.botonMatriculas: hover {
    -fx-background-color: -colorBotonHover;
}
.botonMatriculas: pressed {
    -fx-background-color: -colorBotonPressed;
}
.botonCalendario {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/Calendario.png");
```




ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
-fx-background-repeat: no-repeat;
-fx-background-position: 115px center;

-fx-padding: 0 ;
}
.botonCalendario: hover {
    -fx-background-color: -colorBotonHover;
}
.botonCalendario: pressed {
    -fx-background-color: -colorBotonPressed;
}
.botonNotas {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/inventario.png");
    -fx-background-repeat: no-repeat;
    -fx-background-position: 115px center;
    -fx-padding: 0 ;
}
.botonNotas: hover {
    -fx-background-color: -colorBotonHover;
}
.botonNotas: pressed {
    -fx-background-color: -colorBotonPressed;
}
.botonAsistencia {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/asistencia.png");
    -fx-background-repeat: no-repeat;
    -fx-background-position: 115px center;
    -fx-padding: 0 ;
}
.botonAsistencia: hover {
    -fx-background-color: -colorBotonHover;
}
.botonAsistencia: pressed {
    -fx-background-color: -colorBotonPressed;
}
.botonConfiguracion {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/config.png");
    -fx-background-repeat: no-repeat;
    -fx-background-position: 115px center;
    -fx-padding: 0 ;
}
.botonConfiguracion: hover {
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
-fx-background-color: -colorBotonHover;
}
.botonConfiguracion: pressed{
    -fx-background-color: -colorBotonPressed;
}
.botonAcercaDe {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/acercaDe.png");
    -fx-background-repeat: no-repeat;
    -fx-background-position: 115px center;
    -fx-padding: 0 ;
}
.botonAcercaDe: hover {
    -fx-background-color: -colorBotonHover;
}
.botonAcercaDe: pressed{
    -fx-background-color: -colorBotonPressed;
}
}
.botonAyuda {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/help.png");
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
    -fx-padding: 0 ;
}
.botonAyuda: hover {
    -fx-background-color: -colorBotonHover;
}
}
.botonAyuda: pressed{
    -fx-background-color: -colorBotonPressed;
}
}
.botonTwitter {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/twitter.png");
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;

    -fx-padding: 0 ;
}
.botonTwitter: hover {
    -fx-background-color: rgb(94,169,221);
}
.botonTwitter: pressed{
    -fx-background-color: -colorBotonPressed;
}
}
.botonGoogle {
```



```
-fx-background-color: transparent ;
-fx-background-radius: 0;
-fx-padding: 0 ;
-fx-text-fill: white ;
-fx-background-image: url("../Images/g+.png");
-fx-background-repeat: no-repeat;
-fx-background-position: center;
-fx-padding: 0 ;
}
.botonGoogle: hover {
    -fx-background-color: rgb(221,75,57);
}
.botonGoogle: pressed {
    -fx-background-color: -colorBotonPressed;
}
.botonFacebook {
    -fx-background-color: transparent ;
    -fx-background-radius: 0;
    -fx-padding: 0 ;
    -fx-text-fill: white ;
    -fx-background-image: url("../Images/facebook.png");
    -fx-background-repeat: no-repeat;
    -fx-background-position: center;
    -fx-padding: 0 ;
}
.botonFacebook: hover {
    -fx-background-color: rgb(59,89,152);
}
.botonFacebook: pressed {
    -fx-background-color: -colorBotonPressed;
}
.background {
    -fx-background-color:
        linear-gradient(#38424b 0%, #1f2429 20%, #191d22 100%),
        linear-gradient(#20262b, #191d22),
        radial-gradient(center 15% 85%, radius 100%, rgba(0,89,
            169,1), rgba(255,255,255,0.7));
    -fx-background-insets: 0,9 9 8 9,9,10,11;
    -fx-background-radius: 0;
    -fx-padding: 15 15 15 15;
    -fx-font-family: "Helvetica";
    -fx-font-size: 18px;
    -fx-text-fill: #311c09;
    -fx-effect: innershadow( three-pass-box , rgba(0,0,0,1) , 2, 0.0
        , 0 , 1);
}
.panel {
    -fx-background-color:
        linear-gradient(#38424b 0%, #1f2429 20%, #191d22 100%),
        linear-gradient(#20262b, #191d22),
        radial-gradient(center 50% 50%, radius 100%, rgba(200,200,
            200,0.6), rgba(114,131,148,0.1));
    -fx-background-radius: 5;
    -fx-padding: 1 1 1 1;
    -fx-font-family: "Helvetica";
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
-fx-font-size: 18px;
-fx-text-fill: #311c09;
-fx-effect: innershadow( three-pass-box , rgba(0,0,0,1) , 0, 0.0
    , 0 , 1);
}
.botonMenu {
    -fx-background-color: transparent;
    -fx-text-fill: white;
}

.botonMenu:hover {
    -fx-background-color: #3b6bb7;
    -fx-text-fill: white ;
}

.botonMenu:pressed {
    -fx-background-color: #0b3b87;
    -fx-text-fill: white ;
}

.panelInterno{
    -fx-background-color: #ffffff;
}
.tab-header-background {
    -fx-background-color: #0000ff;
}
.tab-pane
{
    -fx-tab-min-width:100px;
    -fx-tab-min-height:40px;
}
.tab-paneConfiguracion
{
    -fx-tab-min-width:120px;
    -fx-tab-min-height:40px;
}
.tab {
    -fx-background-color: transparent;
    -fx-border-color: transparent;
    -fx-border-width: 0;
}
.tab-label {
    -fx-alignment: CENTER;
    -fx-text-fill: white;
    -fx-font-size: 13px;
    -fx-font-weight: bold;
}
.tab:hover .tab-label {
    -fx-alignment: CENTER;
    -fx-text-fill: white;
    -fx-font-weight: bold;
    -fx-font-size: 17px;
}
.tab:selected .tab-label {
    -fx-alignment: CENTER;
```



```
-fx-text-fill: #00ff00;
-fx-font-weight: bold;
-fx-font-size: 17px;
}
.acordeonEmpresa .titled-pane{
-fx-text-fill: rgb(255,255,255);
-fx-inner-border: rgba(0,0,0,0);
-fx-body-color: rgba(0,0,0,0);
-fx-color: rgba(0,0,0,0);
-fx-padding: 0px;
-fx-background-insets: 0px;
-fx-background-radius: 0px;
-fx-border-color: rgba(0,0,0,0);
-fx-box-border: rgb(255,255,255);
}
.acordeonEmpresa .titled-pane > .title{
-fx-padding: 0 0 0 -8;
}
.acordeonEmpresa .titled-pane > .title > .arrow-button .arrow {
-fx-shape: "c";
}
.acordeonEmpresa .titled-pane:hover{
-fx-text-fill: rgb(255,255,255);
-fx-body-color: rgb(59,89,152);
-fx-color: rgba(0,0,0,1);
-fx-border-style: none;
}
.acordeonEmpresa .titled-pane:pressed{
-fx-text-fill: rgb(255,255,255);
-fx-body-color: rgba(0,0,0,0.3);
}
.titled-pane .content AnchorPane {
-fx-background-color: -colorFondo;
-fx-border-color: transparent;
-fx-border-style: none;
-fx-color: rgba(0,0,0,0);
}
.textos {
-fx-text-fill: rgb(255,255,255);
}
.botones {
-fx-background-color: rgba(0,0,0,0.5);
}
.slider .thumb {
-fx-padding: 9 19 9 19;
-fx-background-radius: 50;
-fx-background-insets: 0,1,1;
-fx-text-fill: black;
```



```
-fx-effect: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 3,
    0.0 , 0 , 1 );
}
.sliderIP .thumb {
    -fx-padding: 9 37 9 37;
    -fx-background-radius: 50;
    -fx-background-insets: 0,1,1;
    -fx-text-fill: black;
    -fx-effect: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 3,
        0.0 , 0 , 1 );
}
.sliderIP .track {
    -fx-background-color: null; /* Hide the track */
    -fx-background-insets: 1 0 -1 0, 0, 1;
    -fx-background-radius: 2.5, 2.5, 1.5;
    -fx-padding: 0.208333em; /* 2.5 */
}

.slider .track {
    -fx-background-color: null; /* Hide the track */
    -fx-background-insets: 1 0 -1 0, 0, 1;
    -fx-background-radius: 2.5, 2.5, 1.5;
    -fx-padding: 0.208333em; /* 2.5 */
}
.progress-bar .track{
    -fx-background-radius: 50;
    -fx-background-insets: 0,0,0;
    -fx-effect: dropshadow( three-pass-box , rgba(0,0,0,1) , 3, 0.0
        , 0 , 1 );
}
.progress-bar .bar {
    -fx-background-radius: 50;
    -fx-background-color: radial-gradient(center 50% -40%, radius
        200%, #00aa00 45%,#00cc00 50%);
    -fx-padding: 2 20 2 20;
    -fx-background-insets: 0, 1;
    -fx-effect: dropshadow( three-pass-box , rgba(0,0,0,0.4) , 5,
        0.0 , 0 , 1 );
    -fx-text-fill: #395306;
}
}
```

splash.fxml

```
<?import javafx.scene.effect.*?>
<?import javafx.scene.text.*?>
<?import javafx.scene.image.*?>
<?import java.lang.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<AnchorPane id="AnchorPane" prefHeight="229.0" prefWidth="429.0"
    stylesheets="@../Style/estilo1.css"
    xmlns="http://javafx.com/javafx/8"
    xmlns:fx="http://javafx.com/fxml/1">
```



Figura F.1: splash.fxml

```

fx:controller="Controller.SplashController">
<children>
  <AnchorPane layoutX="74.0" layoutY="14.0" prefHeight="295.0"
    prefWidth="493.0" style="-fx-background-color: #2b579a;"
    AnchorPane.bottomAnchor="0.0" AnchorPane.leftAnchor="0.0"
    AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
    <children>
      <ImageView fitHeight="16.0" fitWidth="297.0"
        layoutX="128.0" layoutY="156.0" pickOnBounds="true"
        preserveRatio="true">
        <image>
          <Image url="@../Images/loadingSplash.GIF" />
        </image>
      </ImageView>
      <Text fx:id="lblnombrePrograma" fill="WHITE"
        layoutX="40.0" layoutY="141.0" opacity="0.0"
        strokeType="OUTSIDE" strokeWidth="0.0" text="nombre"
        textAlignment="CENTER" wrappingWidth="278.2421875">
      <font>
        <Font name="Segoe UI" size="18.0" />
      </font>
    </Text>
    <Label fx:id="lblCerrarVentana" layoutX="406.0"
      prefHeight="27.0" prefWidth="26.0"
      styleClass="cerrarVentana1" text="X"
      textFill="WHITE">
    <font>
      <Font name="System Bold" size="11.0" />
    </font>
    <cursor>
      <Cursor fx:constant="HAND" />
    </cursor>
  </Label>
  <Text fx:id="lblBienvenido" fill="WHITE" layoutX="1.0"
    layoutY="108.0" opacity="0.0" strokeType="OUTSIDE"
    strokeWidth="0.0" text="Bienvenido"
    textAlignment="CENTER" wrappingWidth="432.2421875"
    AnchorPane.leftAnchor="1.0"
    AnchorPane.rightAnchor="2.0">

```

```

<font>
  <Font name="Segoe UI" size="48.0" />
</font>
</Text>
<Text fx:id="lblversionPrograma" fill="WHITE"
  layoutX="335.0" layoutY="142.0" opacity="0.0"
  strokeType="OUTSIDE" strokeWidth="0.0" text="1.0">
<font>
  <Font name="Segoe UI" size="20.0" />
</font>
</Text>
<Text fx:id="lblCargando" fill="WHITE" layoutY="221.0"
  opacity="0.0" strokeType="OUTSIDE" strokeWidth="0.0"
  text="Iniciando..." x="5.0">
<font>
  <Font name="Segoe UI" size="15.0" />
</font>
</Text>
</children>
</AnchorPane>
</children>
</AnchorPane>

```

principal.fxml

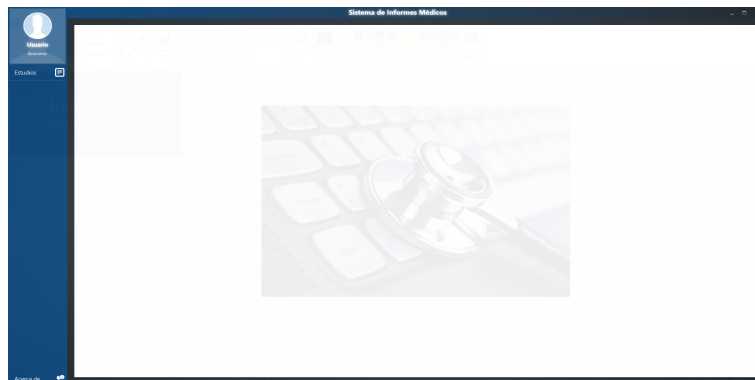


Figura F.2: principal.fxml

```

<?import javafx.geometry.*?>
<?import javafx.scene.shape.*?>
<?import javafx.scene.effect.*?>
<?import javafx.scene.image.*?>
<?import javafx.scene.text.*?>
<?import java.lang.*?>
<?import java.util.*?>
<?import java.net.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<BorderPane maxHeight="-Infinity" maxWidth="-Infinity"
  minHeight="-Infinity" minWidth="-Infinity" prefHeight="634.0"
  prefWidth="1280.0" styleClass="backgroundVPrincipal">

```




```
stylesheets="@../Style/estilo1.css"
xmlns="http://javafx.com/javafx/8"
xmlns:fx="http://javafx.com/fxml/1">
<bottom>
  <VBox fillWidth="false" prefHeight="0.0" prefWidth="638.0"
    BorderPane.alignment="CENTER">
    <children>
      <HBox alignment="BOTTOM_LEFT" maxWidth="3000.0"
        prefHeight="0.0" prefWidth="2000.0"
        VBox.vgrow="ALWAYS">
        <children>
          <Separator fx:id="separadorVertical"
            prefHeight="2.0" prefWidth="2000.0"
            styleClass="separador" HBox.hgrow="ALWAYS">
            <cursor>
              <Cursor fx:constant="V_RESIZE" />
            </cursor>
          </Separator>
          <Separator fx:id="separadorHorizVert"
            prefHeight="2.0" prefWidth="10.0"
            styleClass="separador">
            <cursor>
              <Cursor fx:constant="NW_RESIZE" />
            </cursor>
          </Separator>
        </children>
      </HBox>
    </children>
  </VBox>
</bottom>
<right>
  <HBox prefHeight="358.0" prefWidth="0.0"
    BorderPane.alignment="CENTER">
    <children>
      <VBox maxHeight="3000.0" prefHeight="2000.0"
        HBox.hgrow="ALWAYS">
        <children>
          <Separator fx:id="separadorHorizontal"
            orientation="VERTICAL" prefHeight="422.0"
            prefWidth="0.0" styleClass="separador2"
            HBox.hgrow="ALWAYS" VBox.vgrow="ALWAYS">
            <cursor>
              <Cursor fx:constant="H_RESIZE" />
            </cursor>
          </Separator>
          <Separator fx:id="separadorVertHoriz"
            orientation="VERTICAL" prefHeight="10.0"
            prefWidth="0.0" styleClass="separador">
            <cursor>
              <Cursor fx:constant="NW_RESIZE" />
            </cursor>
          </Separator>
        </children>
      </VBox>
    </children>
  </HBox>
</right>
```



```
</HBox>
</right>
<left>
  <HBox prefHeight="432.0" prefWidth="0.0"
    BorderPane.alignment="CENTER">
    <children>
      <Separator orientation="VERTICAL" prefHeight="200.0"
        styleClass="separador" />
      <VBox fx:id="vboxMenu" alignment="CENTER"
        prefHeight="475.0" prefWidth="187.0"
        styleClass="vbox" HBox.hgrow="ALWAYS">
        <children>
          <Separator fx:id="separadorVertical1"
            opacity="0.0" prefHeight="2.0"
            prefWidth="2000.0" styleClass="separador3">
            <cursor>
              <Cursor fx:constant="DEFAULT" />
            </cursor>
          </Separator>
          <StackPane prefHeight="2.0" prefWidth="142.0">
            <children>
              <ImageView fx:id="imagenFondoUsuario"
                fitHeight="150.0" fitWidth="142.0"
                opacity="0.3" pickOnBounds="true">
                <image>
                  <Image url="@../Images/user.png" />
                </image>
              </ImageView>
              <VBox alignment="CENTER" prefHeight="260.0"
                prefWidth="142.0" spacing="5.0">
                <children>
                  <StackPane minWidth="0.0"
                    prefHeight="72.0"
                    prefWidth="112.0">
                    <children>
                      <ImageView fx:id="imagen"
                        fitHeight="72.0"
                        fitWidth="72.0"
                        pickOnBounds="true"
                        styleClass="fotoAdmin">
                        <image>
                          <Image
                            url="@../Images/user.png"
                            />
                        </image>
                      </ImageView>
                      <Label fx:id="lblImg"
                        onMouseClicked="#MostrarMenuUsuario"
                        prefHeight="72.0"
                        prefWidth="72.0"
                        styleClass="fotoAdministrador">
                        <cursor>
                          <Cursor fx:constant="HAND"
                            />
                        </cursor>
                    </children>
                  </StackPane>
                </children>
              </VBox>
            </children>
          </StackPane>
        </children>
      </VBox>
    </children>
  </HBox>
</left>
</right>
</HBox>
```



```
        </Label>
    </children>
</StackPane>
<Button fx:id="btnUsuario"
    mnemonicParsing="false"
    onMouseClicked="#MostrarMenuUsuario"
    prefHeight="21.0"
    prefWidth="203.0"
    style="-fx-background-color:
transparent;" styleClass="botones"
    text="Administrador"
    VBox.vgrow="ALWAYS">
<cursor>
    <Cursor fx:constant="HAND" />
</cursor>
<effect>
    <Bloom threshold="0.38" />
</effect>
<font>
    <Font name="System Bold"
        size="15.0" />
</font>
</Button>
<Text fx:id="txtCargoEmpresa"
    fill="WHITE" layoutX="90.0"
    layoutY="92.0"
    strokeType="OUTSIDE"
    strokeWidth="0.0" text="C.E.O."
    textAlignment="CENTER"
    wrappingWidth="141.5625">
<font>
    <Font size="12.0" />
</font>
</Text>
</children>
<opaqueInsets>
    <Insets />
</opaqueInsets>
<padding>
    <Insets bottom="15.0" top="5.0" />
</padding>
<effect>
    <Glow level="0.17" />
</effect>
</VBox>
</children>
</StackPane>
<Separator opacity="0.11" prefWidth="200.0" />
<Accordion />
<Button fx:id="btnPedidos"
    alignment="BASELINE_LEFT"
    mnemonicParsing="false"
    onAction="#AbrirVentanaPedidos"
    prefHeight="42.0" prefWidth="213.0"
    styleClass="botonMatriculas" text="
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
        Estudios" />
    <Separator opacity="0.11" prefWidth="200.0" />
    <AnchorPane prefHeight="125.0" prefWidth="142.0"
        VBox.vgrow="ALWAYS" />
    <ImageView fitHeight="17.0" fitWidth="142.0"
        opacity="0.04" pickOnBounds="true"
        preserveRatio="true" />
    <Button fx:id="btnAcercaDe"
        alignment="BASELINE_LEFT"
        mnemonicParsing="false"
        onAction="#AbrirVentanaAcercaDe"
        prefHeight="28.0" prefWidth="329.0"
        styleClass="botonAcercaDe" text="Acerca de"
        />
    </children>
    <opaqueInsets>
        <Insets top="20.0" />
    </opaqueInsets>
    </VBox>
    <Separator orientation="VERTICAL" prefHeight="200.0"
        styleClass="separador" />
    </children>
    </HBox>
</left>
<center>
    <BorderPane fx:id="panelVentanas" maxHeight="2000.0"
        maxWidth="2000.0" minHeight="0.0" minWidth="0.0"
        prefHeight="200.0" prefWidth="200.0"
        BorderPane.alignment="CENTER">
    <top>
        <VBox prefHeight="0.0" prefWidth="490.0"
            BorderPane.alignment="CENTER">
    <children>
        <HBox fx:id="hboxStage" alignment="CENTER_RIGHT"
            prefHeight="40.0" prefWidth="490.0"
            styleClass="barraDeTitulo">
    <children>
        <AnchorPane prefHeight="200.0"
            prefWidth="200.0" HBox.hgrow="ALWAYS">
    <children>
        <Label fx:id="txtTitulo"
            alignment="CENTER" layoutX="6.0"
            layoutY="1.0" prefHeight="21.0"
            prefWidth="342.0" text="SAPHIENS"
            textFill="WHITE"
            AnchorPane.leftAnchor="6.0"
            AnchorPane.rightAnchor="5.0"
            AnchorPane.topAnchor="1.0">
    <font>
        <Font name="System Bold"
            size="18.0" />
    </font>
    <effect>
        <Bloom threshold="0.81" />
    </effect>
```

```

</Label>
<HBox layoutY="3.0" prefHeight="28.0"
  prefWidth="152.0" spacing="5.0">
  <padding>
    <Insets bottom="2.0" top="2.0" />
  </padding>
</HBox>
</children>
</AnchorPane>
<Button fx:id="btnMin"
  mnemonicParsing="false"
  prefHeight="34.0" prefWidth="30.0"
  styleClass="minimizarVentana" />
<Button fx:id="btnMax"
  mnemonicParsing="false"
  prefHeight="34.0" prefWidth="34.0"
  styleClass="maximizarVentana" />
<Button fx:id="btnCerrar"
  alignment="CENTER_RIGHT"
  mnemonicParsing="false"
  prefHeight="34.0" prefWidth="34.0"
  styleClass="cerrarVentana"
  textFill="WHITE" HBox.hgrow="ALWAYS" />
</children>
</HBox>
</children>
</VBox>
</top>
</BorderPane>
</center>
</BorderPane>

```

acercaDe.fxml

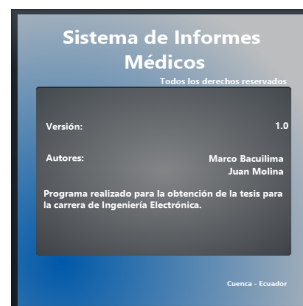


Figura F.3: acercaDe.fxml

```

<?import javafx.scene.image.*>
<?import javafx.scene.text.*>
<?import javafx.scene.effect.*>
<?import java.lang.*>
<?import java.net.*>
<?import java.util.*>
<?import javafx.scene.*>

```



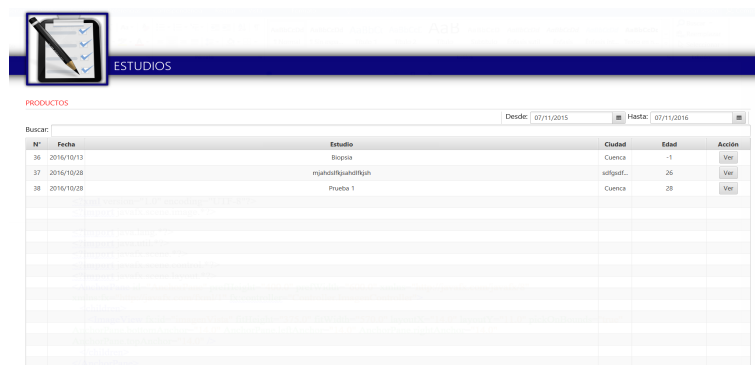
```
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<AnchorPane id="AnchorPane" onMouseClicked="#CerrarVentana"
  prefHeight="453.0" prefWidth="411.0" styleClass="background"
  stylesheets="@../Style/estilo1.css"
  xmlns="http://javafx.com/javafx/8"
  xmlns:fx="http://javafx.com/fxml/1"
  fx:controller="Controller.AcercaDeController">
<children>
  <Text fx:id="txtNombrePrograma" fill="WHITE" layoutX="35.0"
    layoutY="53.0" strokeType="OUTSIDE" strokeWidth="0.0"
    text="Sistema Integrado" textAlignment="CENTER"
    wrappingWidth="346.0">
    <font>
      <Font name="System Bold" size="28.0" />
    </font>
  </Text>
  <Pane layoutX="34.0" layoutY="117.0"
    onMouseClicked="#CerrarVentana" prefHeight="252.0"
    prefWidth="346.0" styleClass="panel">
    <children>
      <Text fill="WHITE" layoutX="10.0" layoutY="64.0"
        strokeType="OUTSIDE" strokeWidth="0.0" text="
          Versi n:" textAlignment="JUSTIFY"
          wrappingWidth="152.0">
        <font>
          <Font name="System Bold" size="13.0" />
        </font>
      </Text>
      <Text fill="WHITE" layoutX="10.0" layoutY="111.0"
        strokeType="OUTSIDE" strokeWidth="0.0" text="
          Autores: " textAlignment="JUSTIFY"
          wrappingWidth="179.0">
        <font>
          <Font name="System Bold" size="13.0" />
        </font>
      </Text>
      <Text fill="WHITE" layoutX="233.0" layoutY="112.0"
        strokeType="OUTSIDE" strokeWidth="0.0" text="Marco
          Bacuilima" textAlignment="RIGHT"
          wrappingWidth="103.0">
        <font>
          <Font name="System Bold" size="13.0" />
        </font>
      </Text>
      <Text fill="WHITE" layoutX="10.0" layoutY="164.0"
        strokeType="OUTSIDE" strokeWidth="0.0"
        text="Programa realizado para la obtenci n de la
          tesis para la carrera de Ingenier a Electr nica."
        textAlignment="JUSTIFY"
        wrappingWidth="325.826171875">
        <font>
          <Font name="System Bold" size="13.0" />
        </font>
      </Text>
    </children>
  </Pane>
</AnchorPane>
```

```

<Text fill="WHITE" layoutX="77.0" layoutY="63.0"
strokeType="OUTSIDE" strokeWidth="0.0" text="1.0"
textAlignment="RIGHT" wrappingWidth="265.0">
<font>
<Font name="System Bold" size="13.0" />
</font>
</Text>
<Text fill="WHITE" layoutX="209.0" layoutY="131.0"
strokeType="OUTSIDE" strokeWidth="0.0" text="Juan
Molina" textAlignment="RIGHT" wrappingWidth="127.0">
<font>
<Font name="System Bold" size="13.0" />
</font>
</Text>
</children>
</Pane>
<Text fill="WHITE" layoutX="163.0" layoutY="113.0"
strokeType="OUTSIDE" strokeWidth="0.0" text="Todos los
derechos reservados" textAlignment="RIGHT"
wrappingWidth="210.826171875">
<font>
<Font name="System Bold" size="12.0" />
</font>
</Text>
<Text fx:id="idCiudad" fill="WHITE" layoutX="261.0"
layoutY="414.0" strokeType="OUTSIDE" strokeWidth="0.0"
text="Cuenca – Ecuador" textAlignment="RIGHT"
wrappingWidth="113.0">
<font>
<Font name="System Bold" size="10.0" />
</font>
</Text>
</children>
</AnchorPane>

```

pedidos.fxml



N°	Fecha	Estudio	Ciudad	Edad	Acción
36	2016/10/13	Bicipia	Cuenca	-1	Ver
37	2016/10/28	mjanosfjuaatfjsh	adiga...	26	Ver
38	2016/10/28	Prueba 1	Cuenca	28	Ver

Figura F.4: pedidos.fxml

```

<?import javafx.scene.chart.*?>
<?import javafx.scene.effect.*?>

```



```
<?import javafx.geometry.*?>
<?import javafx.scene.paint.*?>
<?import javafx.scene.image.*?>
<?import javafx.scene.text.*?>
<?import javafx.scene.shape.*?>
<?import java.lang.*?>
<?import java.net.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<VBox fx:id="vBoxGeneral" prefHeight="871.0" prefWidth="817.0"
  stylesheets="@../Style/estilo1.css"
  xmlns="http://javafx.com/javafx/8"
  xmlns:fx="http://javafx.com/fxml/1"
  fx:controller="Controller.PedidosController">
<children>
  <AnchorPane id="AnchorPane" fx:id="panelExterno"
    maxHeight="2000.0" maxWidth="2000.0" minHeight="0.0"
    minWidth="0.0" prefHeight="853.0" prefWidth="817.0"
    stylesheets="@../Style/estilo1.css" VBox.vgrow="ALWAYS">
<children>
  <StackPane fx:id="stack" prefHeight="150.0"
    prefWidth="200.0" AnchorPane.bottomAnchor="0.0"
    AnchorPane.leftAnchor="0.0"
    AnchorPane.rightAnchor="0.0"
    AnchorPane.topAnchor="0.0" />
  <AnchorPane fx:id="panelInterno" minWidth="791.0"
    prefHeight="785.0" prefWidth="791.0"
    styleClass="panelInterno"
    AnchorPane.bottomAnchor="18.0"
    AnchorPane.leftAnchor="18.0"
    AnchorPane.rightAnchor="18.0"
    AnchorPane.topAnchor="18.0">
<children>
  <AnchorPane layoutY="122.0" prefHeight="47.0"
    prefWidth="41.0" style="-fx-background-color:
    #060078;">
    <effect>
      <DropShadow blurType="GAUSSIAN"
        height="147.0" radius="36.25"
        spread="0.07" width="0.0" />
    </effect>
  </AnchorPane>
  <AnchorPane layoutX="225.0" layoutY="122.0"
    prefHeight="47.0" prefWidth="556.0"
    style="-fx-background-color: #060078;"
    AnchorPane.leftAnchor="225.0"
    AnchorPane.rightAnchor="0.0">
<children>
  <Text fill="WHITE" layoutX="14.0"
    layoutY="35.0" strokeType="OUTSIDE"
    strokeWidth="0.0" text="ESTUDIOS"
    wrappingWidth="232.1923828125">
<font>
```




ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
<Font name="Segoe UI Semilight"
      size="30.0" />
</font>
</Text>
<Button layoutX="497.0" layoutY="-2.0"
        mnemonicParsing="false"
        onAction="#Actualizar" prefHeight="49.0"
        prefWidth="59.0"
        styleClass="botonActualizar"
        AnchorPane.rightAnchor="0.0"
        AnchorPane.topAnchor="-2.0" />
</children>
<effect>
  <DropShadow blurType="GAUSSIAN"
              height="147.0" radius="36.25"
              spread="0.07" width="0.0" />
</effect>
</AnchorPane>
<VBox fx:id="vBoxContenedor" layoutX="38.0"
      layoutY="217.0" prefHeight="514.0"
      prefWidth="708.0" spacing="5.0"
      AnchorPane.bottomAnchor="20.0"
      AnchorPane.leftAnchor="38.0"
      AnchorPane.rightAnchor="35.0"
      AnchorPane.topAnchor="217.0">
  <children>
    <ProgressBar fx:id="progresoCarga"
                 maxWidth="2000.0" prefHeight="0.0"
                 prefWidth="718.0" progress="0.2"
                 visible="false" VBox.vgrow="NEVER" />
    <VBox fx:id="vBoxProductos"
          alignment="TOP_CENTER" minWidth="558.0"
          prefHeight="508.0" prefWidth="718.0"
          spacing="2.0" VBox.vgrow="SOMETIMES">
      <children>
        <HBox prefHeight="13.0"
              prefWidth="718.0" spacing="5.0">
          <children>
            <Text fill="RED" layoutX="14.0"
                  layoutY="80.0"
                  strokeType="OUTSIDE"
                  strokeWidth="0.0"
                  text="PRODUCTOS">
              <font>
                <Font name="Tw Cen MT
                      Condensed Bold"
                      size="17.0" />
              </font>
            </Text>
            <Pane prefHeight="200.0"
                  prefWidth="200.0"
                  HBox.hgrow="ALWAYS" />
          </children>
        </HBox>
      </children>
    </VBox>
  </children>
</VBox>
```



```
<Separator layoutX="14.0"
  layoutY="82.0" prefHeight="7.0"
  prefWidth="973.0"
  AnchorPane.leftAnchor="14.0"
  AnchorPane.rightAnchor="14.0"
  AnchorPane.topAnchor="82.0" />
<HBox alignment="TOP_CENTER"
  prefHeight="23.0"
  prefWidth="515.0" spacing="5.0">
<children>
  <Pane prefHeight="25.0"
    prefWidth="34.0"
    HBox.hgrow="ALWAYS" />
  <Separator
    orientation="VERTICAL"
    prefHeight="200.0" />
  <Text strokeType="OUTSIDE"
    strokeWidth="0.0"
    text="Desde: ">
  <font>
    <Font name="Tw Cen MT
      Condensed Bold"
      size="17.0" />
  </font>
</Text>
  <DatePicker fx:id="fechaInicio"
    onAction="#CambioFecha"
    HBox.hgrow="ALWAYS" />
  <Text strokeType="OUTSIDE"
    strokeWidth="0.0"
    text="Hasta: ">
  <font>
    <Font name="Tw Cen MT
      Condensed Bold"
      size="17.0" />
  </font>
</Text>
  <DatePicker fx:id="fechaFinal"
    onAction="#CambioFecha"
    HBox.hgrow="ALWAYS" />
  <Separator
    orientation="VERTICAL"
    prefHeight="200.0" />
</children>
</HBox>
<HBox alignment="TOP_CENTER"
  prefHeight="23.0"
  prefWidth="515.0" spacing="5.0"
  VBox.vgrow="NEVER">
<children>
  <Text strokeType="OUTSIDE"
    strokeWidth="0.0"
    text="Buscar: ">
  <font>
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
<Font name="Tw Cen MT
Condensed Bold"
size="17.0" />
</font>
</Text>
<TextField fx:id="txtBuscar"
onKeyReleased="#Buscar"
prefHeight="25.0"
prefWidth="575.0"
HBox.hgrow="ALWAYS" />
</children>
</HBox>
<HBox alignment="TOP_CENTER"
prefHeight="431.0"
prefWidth="718.0"
VBox.vgrow="ALWAYS">
<children>
<TableView
fx:id="tablaProductos"
onKeyReleased="#Buscar"
onMouseClicked="#TablaProductosMouseClicked"
prefHeight="327.0"
prefWidth="636.0"
HBox.hgrow="ALWAYS">
<contextMenu>
<ContextMenu>
<items>
<MenuItem
mnemonicParsing="false"
onAction="#VerPedido"
text="Ver
Pedido" />
<MenuItem
mnemonicParsing="false"
onAction="#DespacharPedido"
text="Cambiar
Estado"
visible="false"
/>
<SeparatorMenuItem
mnemonicParsing="false"
/>
<MenuItem
mnemonicParsing="false"
onAction="#EliminarPedido"
text="Eliminar
Pedido" />
</items>
</ContextMenu>
</contextMenu>
</TableView>
</children>
</HBox>
</children>
<padding>
```

```

        <Insets bottom="20.0" top="2.0" />
    </padding>
</VBox>
</children>
</VBox>
<ImageView fitHeight="180.0" fitWidth="185.0"
    layoutX="39.0" layoutY="14.0"
    pickOnBounds="true">
    <image>
        <Image url="@../Images/pedidos.png" />
    </image>
</ImageView>
<Rectangle arcHeight="5.0" arcWidth="5.0"
    fill="#2197ff00" height="184.0" layoutX="39.0"
    layoutY="12.0" stroke="#060078"
    strokeLineCap="ROUND" strokeLineJoin="ROUND"
    strokeType="INSIDE" strokeWidth="5.0"
    width="188.0">
    <effect>
        <DropShadow />
    </effect>
</Rectangle>
<Label fx:id="lblCargando" layoutX="666.0"
    layoutY="177.0" opacity="0.0"
    text="Cargando... &#10;Por favor espere."
    textFill="RED" AnchorPane.rightAnchor="35.0"
    AnchorPane.topAnchor="177.0" />
</children>
</AnchorPane>
</children>
</AnchorPane>
</children>
</VBox>

```

pedidoHecho.fxml

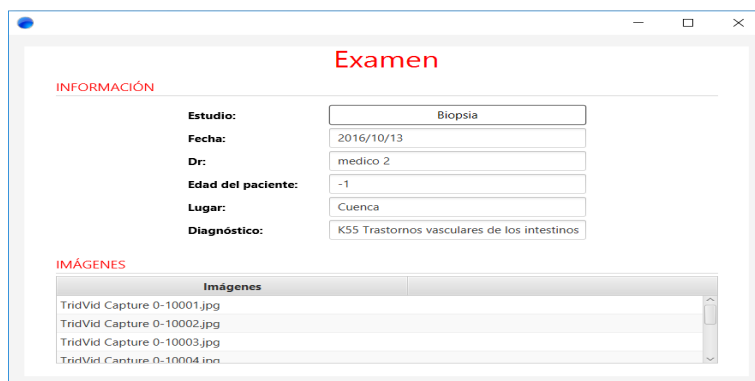


Figura F.5: pedidoHecho.fxml

```

<?import javafx.scene.chart.*?>
<?import javafx.scene.effect.*?>
<?import javafx.geometry.*?>

```



```
<?import javafx.scene.paint.*?>
<?import javafx.scene.image.*?>
<?import javafx.scene.text.*?>
<?import javafx.scene.shape.*?>
<?import java.lang.*?>
<?import java.net.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<VBox fx:id="vBoxGeneral" prefHeight="654.0" prefWidth="817.0"
  xmlns="http://javafx.com/javafx/8"
  xmlns:fx="http://javafx.com/fxml/1"
  fx:controller="Controller.PedidoHechoController">
  <children>
    <AnchorPane id="AnchorPane" fx:id="panelExterno"
      maxHeight="2000.0" maxWidth="2000.0" minHeight="0.0"
      minWidth="0.0" prefHeight="1000.0" prefWidth="817.0"
      stylesheets="@../Style/estilo1.css" VBox.vgrow="ALWAYS">
      <children>
        <StackPane fx:id="stack" prefHeight="943.0"
          prefWidth="817.0" AnchorPane.bottomAnchor="96.0"
          AnchorPane.leftAnchor="0.0"
          AnchorPane.rightAnchor="0.0"
          AnchorPane.topAnchor="0.0" />
        <AnchorPane fx:id="panelInterno" minWidth="791.0"
          prefHeight="700.0" prefWidth="791.0"
          styleClass="panelInterno"
          AnchorPane.bottomAnchor="18.0"
          AnchorPane.leftAnchor="18.0"
          AnchorPane.rightAnchor="18.0"
          AnchorPane.topAnchor="18.0">
          <children>
            <Label fx:id="lblEstadoPedido" alignment="CENTER"
              prefHeight="23.0" prefWidth="781.0"
              text="Examen" textFill="RED"
              AnchorPane.leftAnchor="0.0"
              AnchorPane.rightAnchor="0.0">
              <font>
                <Font name="Segoe UI Emoji" size="36.0" />
              </font>
            </Label>
            <VBox fx:id="vBoxContenedor" layoutX="37.0"
              layoutY="41.0" prefHeight="577.0"
              prefWidth="708.0" spacing="5.0"
              AnchorPane.bottomAnchor="0.0"
              AnchorPane.leftAnchor="37.0"
              AnchorPane.rightAnchor="36.0"
              AnchorPane.topAnchor="41.0">
              <children>
                <ProgressBar fx:id="progresoCarga"
                  maxWidth="2000.0" prefHeight="0.0"
                  prefWidth="718.0" progress="0.2"
                  visible="false" VBox.vgrow="NEVER" />
              </children>
            </VBox>
          </children>
        </AnchorPane>
      </children>
    </AnchorPane>
  </children>
</VBox>
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
<VBox fx:id="vBoxLocal" minWidth="558.0"
  prefHeight="269.0" prefWidth="718.0"
  spacing="2.0" VBox.vgrow="ALWAYS">
  <children>
    <HBox fx:id="hboxDatosPedido"
      spacing="5.0">
      <children>
        <Text fill="RED" layoutX="14.0"
          layoutY="80.0"
          strokeType="OUTSIDE"
          strokeWidth="0.0"
          text="INFORMACIN">
          <font>
            <Font name="Tw Cen MT
              Condensed Bold"
              size="17.0" />
          </font>
        </Text>
        <Pane prefHeight="0.0"
          prefWidth="393.0"
          HBox.hgrow="ALWAYS" />
      </children>
    </HBox>
    <Separator layoutX="14.0"
      layoutY="82.0" prefHeight="7.0"
      prefWidth="973.0"
      AnchorPane.leftAnchor="14.0"
      AnchorPane.rightAnchor="14.0"
      AnchorPane.topAnchor="82.0" />
    <HBox alignment="CENTER"
      layoutX="14.0" layoutY="89.0"
      minWidth="558.0"
      prefHeight="408.0"
      prefWidth="718.0" spacing="30.0"
      AnchorPane.leftAnchor="14.0"
      AnchorPane.rightAnchor="14.0"
      AnchorPane.topAnchor="89.0">
      <children>
        <VBox prefHeight="314.0"
          prefWidth="465.0"
          spacing="5.0">
          <children>
            <HBox
              alignment="CENTER_LEFT"
              />
            <HBox alignment="CENTER"
              spacing="10.0" />
            <HBox
              alignment="CENTER_LEFT">
              <children>
                <Text
                  fx:id="lblLocal"
                  strokeType="OUTSIDE"
                  strokeWidth="0.0"
                  text="Estudio:"
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
        wrappingWidth="165.0">
    <font >
        <Font
            name="System
            Bold "
            size="15.0 "
            />
    </font >
</Text >
<TextField
    fx:id="txtNumeroDespacho"
    alignment="CENTER"
    editable="false "
    prefHeight="25.0 "
    prefWidth="352.0 "
    />
</children >
</HBox >
<HBox
    alignment="CENTER_LEFT">
    <children >
        <Text
            strokeType="OUTSIDE"
            strokeWidth="0.0 "
            text="Fecha: "
            wrappingWidth="165.0">
        <font >
            <Font
                name="System
                Bold "
                size="15.0 "
                />
        </font >
    </Text >
    <TextField
        fx:id="txtFechaPedido"
        editable="false "
        prefHeight="31.0 "
        prefWidth="248.0 "
        promptText="Fecha "
        HBox.hgrow="ALWAYS"
        />
    </children >
</HBox >
<HBox
    alignment="CENTER_LEFT">
    <children >
        <Text
            strokeType="OUTSIDE"
            strokeWidth="0.0 "
            text="Dr: "
            wrappingWidth="165.0">
        <font >
            <Font
                name="System
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
        Bold "
        size="15.0 "
    />
</font>
</Text>
<TextField
    fx:id="txtUsuarioPedido "
    editable="false "
    prefHeight="31.0 "
    prefWidth="248.0 "
    promptText="Quien
    hizo el examen "
    HBox.hgrow="ALWAYS"
/>
</children>
</HBox>
<HBox
    alignment="CENTER_LEFT">
<children>
<Text
    strokeType="OUTSIDE"
    strokeWidth="0.0 "
    text="Edad del
    paciente:"
    wrappingWidth="165.0">
<font>
    <Font
        name="System
        Bold "
        size="15.0 "
    />
    </font>
</Text>
<TextField
    fx:id="txtDespachaPedido"
    editable="false "
    prefHeight="31.0 "
    prefWidth="248.0 "
    promptText="Edad
    del paciente"
    HBox.hgrow="ALWAYS"
/>
</children>
</HBox>
<HBox
    alignment="CENTER_LEFT">
<children>
<Text
    strokeType="OUTSIDE"
    strokeWidth="0.0 "
    text="Lugar:"
    wrappingWidth="165.0">
<font>
    <Font
        name="System
```




ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```

                Bold"
                size="15.0"
            />
        </font>
    </Text>
    <TextField
        fx:id="txtLocal"
        editable="false"
        prefHeight="31.0"
        prefWidth="248.0"
        promptText="Lugar
donde se hizo el
examen"
        HBox.hgrow="ALWAYS"
    />
</children>
</HBox>
<HBox
    alignment="CENTER_LEFT">
    <children>
        <Text
            strokeType="OUTSIDE"
            strokeWidth="0.0"
            text="Diagnóstico:"
            wrappingWidth="165.0">
            <font>
                <Font
                    name="System
                    Bold"
                    size="15.0"
                />
            </font>
        </Text>
        <TextField
            fx:id="txtDireccion"
            editable="false"
            prefHeight="31.0"
            prefWidth="248.0"
            promptText="Diagnóstico"
            HBox.hgrow="ALWAYS"
        />
    </children>
</HBox>
</children>
</VBox>
</children>
</HBox>
</padding>
    <Insets bottom="20.0" top="2.0" />
</padding>
</VBox>
<VBox fx:id="vBoxProductos"
    alignment="TOP_CENTER" minWidth="558.0"
    prefHeight="508.0" prefWidth="718.0"

```



```
        spacing="2.0" VBox.vgrow="ALWAYS">
    <children>
        <HBox prefHeight="13.0"
            prefWidth="718.0" spacing="5.0">
            <children>
                <Text fill="RED" layoutX="14.0"
                    layoutY="80.0"
                    strokeType="OUTSIDE"
                    strokeWidth="0.0"
                    text="IM GENES">
                <font>
                    <Font name="Tw Cen MT
                        Condensed Bold"
                        size="17.0" />
                </font>
            </Text>
            <Pane prefHeight="200.0"
                prefWidth="200.0"
                HBox.hgrow="ALWAYS" />
            </children>
        </HBox>
        <Separator layoutX="14.0"
            layoutY="82.0" prefHeight="7.0"
            prefWidth="973.0"
            AnchorPane.leftAnchor="14.0"
            AnchorPane.rightAnchor="14.0"
            AnchorPane.topAnchor="82.0" />
        <TableView fx:id="tablaProductos"
            onMouseClicked="#TablaProductosMouseClicked"
            prefHeight="274.0"
            prefWidth="553.0" />
    </children>
    <padding>
        <Insets bottom="20.0" top="2.0" />
    </padding>
</VBox>
</children>
</VBox>
<Label fx:id="lblCargando" layoutX="666.0"
    layoutY="177.0" opacity="0.0"
    text="Cargando... &#10;Por favor espere."
    textFill="RED" AnchorPane.rightAnchor="35.0"
    AnchorPane.topAnchor="177.0" />
</children>
</AnchorPane>
</children>
</AnchorPane>
</children>
</VBox>
```

imagen.fxml

```
<?import javafx.scene.image.*?>
<?import java.lang.*?>
<?import java.util.*?>
```

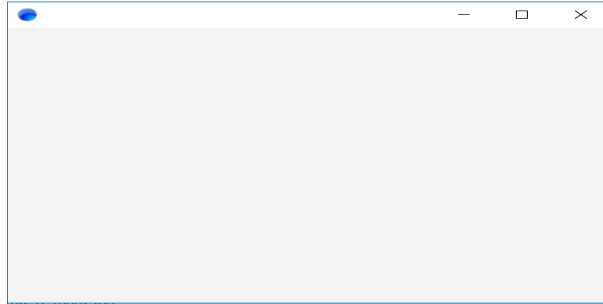


Figura F.6: imagen.fxml

```
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<AnchorPane id="AnchorPane" prefHeight="400.0" prefWidth="600.0"
  xmlns="http://javafx.com/javafx/8"
  xmlns:fx="http://javafx.com/fxml/1"
  fx:controller="Controller.ImagenController">
  <children>
    <ImageView fx:id="imagenVista" fitHeight="375.0"
      fitWidth="570.0" layoutX="14.0" layoutY="11.0"
      pickOnBounds="true" AnchorPane.bottomAnchor="14.0"
      AnchorPane.leftAnchor="14.0" AnchorPane.rightAnchor="14.0"
      AnchorPane.topAnchor="14.0" />
  </children>
</AnchorPane>
```

SplashController.java

```
import Model.Archivo;
import Model.CRUD;
import Model.Classes.Gestor;
import static Model.Prototipo1.direccion;
import static Model.Prototipo1.nombreArchivo;
import Model.Ventana;
import Model.WindowAnimation.Aparecer;
import Model.WindowAnimation.AparecerDesdeDerecha;
import Model.WindowAnimation.AparecerDesdeIzquierda;
import java.net.URL;
import java.util.Arrays;
import java.util.List;
import java.util.ResourceBundle;
import javafx.application.Platform;
import javafx.collections.ObservableList;
import javafx.concurrent.Service;
import javafx.concurrent.Task;
import javafx.concurrent.WorkerStateEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Label;
import javafx.scene.control.Tooltip;
import javafx.scene.image.ImageView;
```



```
import javafx.scene.input.MouseEvent;
import javafx.scene.text.Text;
import javafx.stage.Stage;
import javafx.stage.StageStyle;
import javax.swing.JOptionPane;
import org.controlsfx.dialog.CommandLinksDialog;
public class SplashController implements Initializable {
    @FXML
    private Text lblnombrePrograma;
    @FXML
    private Text lblBienvenido;
    @FXML
    private Text lblversionPrograma;
    @FXML
    private Text lblCargando;
    @FXML
    private Label lblCerrarVentana;
    @FXML
    private Text lblNombreEmpresaMia;
    @FXML
    private ImageView imgLogoEmpresaMia;
    @FXML
    private Text lblCiaLtda;
    public static CRUD crud;
    public static Gestor gestor = new Gestor(); // Gestiona la
        informaci n del programa.
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        Platform.runLater(() -> {
            configuracionVentana(rb);
            iniciarPrograma();
        });
    }
    public void configuracionVentana(ResourceBundle rb) {
        rb = ResourceBundle.getBundle("Model/Languages/language_" +
            new Archivo().leerArchivoProperties(
                direccion, nombreArchivo, "idioma"));
        lblBienvenido.setText(rb.getString("Bienvenido"));
        lblCargando.setText(rb.getString("Iniciando"));
        lblCerrarVentana.setToolTipText(new
            Tooltip(rb.getString("Cerrar")));
        lblversionPrograma.setText(new
            Archivo().leerArchivoProperties(direccion,
                nombreArchivo, "versionPrograma"));
        lblnombrePrograma.setText(new
            Archivo().leerArchivoProperties(direccion,
                nombreArchivo, "nombrePrograma"));
        lblCerrarVentana.setOnMouseClicked(
            (MouseEvent event) -> {
                Platform.exit();
                System.exit(0);
            }
        );
    }
    private void iniciarPrograma() {
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
Service servicio = new Service<ObservableList<String>>() {
    @Override
    protected Task createTask() {
        return new Task<ObservableList<String>>() {
            @Override
            protected ObservableList<String> call() throws
                InterruptedException {
                crud = new CRUD();
                if (crud.obtenerLista("Exámenes").isEmpty())
                {
                    gestor.actualizarListaExámenes();
                }
                return null;
            }
        };
    }
};
servicio.start();
servicio.setOnRunning(
    new EventHandler<WorkerStateEvent>() {
        @Override
        public void handle(WorkerStateEvent t
        ) {
            new AparecerDesdeIzquierda(lblBienvenido).play();
            new AparecerDesdeDerecha(lblnombrePrograma).play();
            new AparecerDesdeDerecha(lblversionPrograma).play();
            new Aparecer(lblCargando).play();
        }
    }
);
servicio.setOnSucceeded(
    new EventHandler<WorkerStateEvent>() {
        @Override
        public void handle(WorkerStateEvent t
        ) {
            new
                Archivo().escribirArchivoProperties(direccion ,
                nombreArchivo, "inicio", "si");
            Stage st = new Stage();
            new Ventana().abrirVentana(st, "principal",
            StageStyle.UNDECORATED, new
            PrincipalController(st));
            new
                Ventana().cerrarVentana(lblCerrarVentana);
        }
    }
);
servicio.setOnFailed(
    new EventHandler<WorkerStateEvent>() {
        @Override
        public void handle(WorkerStateEvent event
        ) {
            mensajeSeleccion();
            lblCargando.setText("Reinicie el programa para una
            nueva conexión");
        }
    }
);
```



```
    }
  }
);
}
public void mensajeSeleccion() {
    List<CommandLinksDialog.CommandLinksButtonType> links =
        Arrays
            .asList(new
                CommandLinksDialog.CommandLinksButtonType(
                    "Iniciar de todos modos",
                    "Quiero iniciar el programa aunque no est
                        conectado a la base de datos.",
                    false),
                    new
                        CommandLinksDialog.CommandLinksButtonType(
                            "Salir",
                            "Cerrar el programa para reiniciar
                                con una nueva conexi n.",
                            false));
    CommandLinksDialog dlg = new CommandLinksDialog(links);
    dlg.setTitle("Informaci n");
    dlg.getDialogPane().setHeaderText("No existe comunicaci n
        con la Base de Datos.");
    dlg.getDialogPane().setContentText("Escoga una de las
        siguientes opciones:");
    dlg.initStyle(StageStyle.DECORATED);
    dlg.showAndWait().ifPresent(result ->
        System.out.println("Result is " + result));
    switch (dlg.getResult().getText()) {
        case "Iniciar de todos modos":
            new Ventana().abrirVentana("login", 2);
            new Ventana().cerrarVentana(lblCerrarVentana);
            break;
        case "Salir":
            System.exit(1);
            break;
        default:
            JOptionPane.showMessageDialog(null, "\n"
                + "Existe un problema con la conexi n a la
                    base de datos.\n\n"
                + "- Aseg rese que el DBMS est activo.\n"
                + "- Si est activo, comun quese con
                    Servicio T cnico. \n");
            break;
    }
}
}
```

PrincipalController.java

```
import static Controller.SplashController.gestor;
import Model.AparienciaVentana;
import Model.Archivo;
import static Model.Prototipo1.direccion;
import static Model.Prototipo1.nombreArchivo;
```



```
import Model.WindowAnimation.Aparecer;
import Model.WindowAnimation.AparecerDesdeAbajo;
import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.application.Platform;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.Separator;
import javafx.scene.control.Tooltip;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.text.Text;
import javafx.stage.Modality;
import javafx.stage.Stage;
import javafx.stage.StageStyle;
public class PrincipalController implements Initializable {
    @FXML
    private Separator separadorVertical1;
    @FXML
    private ImageView imagenFondoUsuario;
    @FXML
    private ImageView imagen;
    @FXML
    private Label lblImg;
    @FXML
    private Button btnUsuario;
    @FXML
    private Button btnAcercaDe;
    @FXML
    private Separator separadorVertical;
    @FXML
    private Separator separadorHorizVert;
    @FXML
    private BorderPane panelVentanas;
    @FXML
    private HBox hboxStage;
    @FXML
    private Label txtTitulo;
    @FXML
    private Button btnMin;
    @FXML
    private Button btnMax;
```



```
@FXML
private Button btnCerrar;
@FXML
private Separator separadorHorizontal;
@FXML
private Separator separadorVertHoriz;
@FXML
private Text txtCargoEmpresa;
@FXML
private Button btnPedidos;
public Stage stage;
public Parent root;
public String ventana = "";
@FXML
private VBox vboxMenu;
@Override
public void initialize(URL url, ResourceBundle rb) {
    configuracionVentana(rb);
}
public PrincipalController(Stage st) {
    this.stage = st;
}
public void configuracionVentana(ResourceBundle rb) {
    gestor.mantenerConexion();
    txtTitulo.setText(" " + new
        Archivo().leerArchivoProperties(direccion,
            nombreArchivo, "nombrePrograma"));
    AparienciaVentana configVentana = new
        AparienciaVentana(stage, btnMin, btnMax, btnCerrar,
            hboxStage, separadorVertical,
                separadorHorizontal, separadorHorizVert,
                separadorVertHoriz);
    String maximizar = new
        Archivo().leerArchivoProperties(direccion,
            nombreArchivo, "ventanaMaximizada");
    if (maximizar.equals("si")) {
        configVentana.setMaximized(true);
    } else {
        configVentana.setMaximized(false);
    }
    rb = ResourceBundle.getBundle("Model/Languages/language_" +
        new Archivo().leerArchivoProperties(direccion,
            nombreArchivo, "idioma"));
    btnMin.setTooltip(new Tooltip(rb.getString("Minimizar")));
    btnMax.setTooltip(new Tooltip(rb.getString("Maximizar")));
    btnCerrar.setTooltip(new Tooltip(rb.getString("Cerrar")));
    btnAcercaDe.setTooltip(new
        Tooltip(rb.getString("AcercaDe")));
    btnUsuario.setText("Usuario");
    txtCargoEmpresa.setText("Anonimo");
    String foto = "/Images/user.png";
    imagen.setImage(new
        Image(getClass().getResourceAsStream(foto)));
    imagenFondoUsuario.setImage(new
        Image(getClass().getResourceAsStream(foto)));
}
```




ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
Circle circ = new Circle((imagen.getFitWidth() / 2),
    (imagen.getFitHeight() / 2), (imagen.getFitHeight() /
    2));
circ.setStrokeWidth(2);
imagen.setClip(circ);
txtTitulo.setOpacity(0);
btnUsuario.setOpacity(0);
btnAcercaDe.setOpacity(0);
Platform.runLater(() -> {
    new Aparecer(txtTitulo).play();
    new Aparecer(btnUsuario).play();
    new Aparecer(btnAcercaDe).play();
    new Aparecer(txtCargoEmpresa).play();
});
try {
    if (!ventana.equals("inicial")) {
        root =
            FXMLLoader.load(getClass().getResource("/View/inicial.fxml"));
        panelVentanas.setCenter(root);
        root.setOpacity(0);
        new AparecerDesdeAbajo(root).play();
        ventana = "inicial";
    }
} catch (IOException ex) {
}
}
@FXML
private void MostrarMenuUsuario(MouseEvent event) {
}
@FXML
private void AbrirVentanaAcercaDe(ActionEvent event) {
    try {
        Stage st = new Stage();
        FXMLLoader loader = new
            FXMLLoader(getClass().getResource("/View/acercaDe.fxml"));
        Parent rootAbout
            = loader.load();
        Scene scene = new Scene(rootAbout);
        scene.setFill(Color.TRANSPARENT);
        st.setScene(scene);
        st.initStyle(StageStyle.TRANSPARENT);
        st.initModality(Modality.APPLICATION_MODAL);
        st.show();
    } catch (IOException ex) {
    }
}
@FXML
private void AbrirVentanaPedidos(ActionEvent event) {
    try {
        root =
            FXMLLoader.load(getClass().getResource("/View/pedidos.fxml"));
        panelVentanas.setCenter(root);
        root.setOpacity(0);
        new AparecerDesdeAbajo(root).play();
    } catch (IOException ex) {
    }
}
```



```
        System.err.println("Error en el metodo initialize de la
                           clase PedidosController, Hay un problema al abrir la
                           ventana pedidos.");
    }
}
}
```

AcercaDeController.java

```
import Model.Archivo;
import static Model.Prototipo1.direccion;
import static Model.Prototipo1.nombreArchivo;
import java.awt.Desktop;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.URL;
import java.util.ResourceBundle;
import java.util.logging.Level;
import java.util.logging.Logger;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.input.MouseEvent;
import javafx.scene.text.Text;
import javafx.stage.Stage;
public class AcercaDeController implements Initializable {
    @FXML
    private Text txtNombrePrograma;
    @FXML
    private Text idCiudad;
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        txtNombrePrograma.setText(new
            Archivo().leerArchivoProperties(direccion,
            nombreArchivo, "nombrePrograma"));
        txtNombrePrograma.requestFocus();
    }
    @FXML
    private void CerrarVentana(MouseEvent event) {
        Stage stage = (Stage)
            txtNombrePrograma.getScene().getWindow();
        stage.close();
    }
    private void AbrirPaginaWeb(MouseEvent event) {
        try {
            Desktop.getDesktop().browse(new URI(""));
        } catch (URISyntaxException | IOException ex) {
            Logger.getLogger(AcercaDeController.class.getName()).log(Level.SEVERE,
                null, ex);
        }
    }
}
}
```

PedidosController.java



```
import static Controller.SplashController.gestor;
import Model.Archivo;
import Model.Classes.Examenes;
import static Model.Prototipo1.direccion;
import static Model.Prototipo1.examenHecho;
import static Model.Prototipo1.nombreArchivo;
import Model.WindowAnimation.AparecerDesdeAbajo;
import java.net.URL;
import java.sql.Date;
import java.time.LocalDate;
import java.util.ResourceBundle;
import javafx.application.Platform;
import javafx.beans.property.SimpleBooleanProperty;
import javafx.beans.property.SimpleStringProperty;
import javafx.beans.value.ObservableValue;
import javafx.collections.transformation.FilteredList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.DatePicker;
import javafx.scene.control.Label;
import javafx.scene.control.ProgressBar;
import javafx.scene.control.TableCell;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.image.Image;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.StackPane;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import javafx.stage.StageStyle;
import javafx.util.Callback;
public class PedidosController implements Initializable {
    @FXML
    private AnchorPane panelInterno;
    @FXML
    private AnchorPane panelExterno;
    @FXML
    private VBox vBoxGeneral;
    @FXML
    private ProgressBar progresoCarga;
    @FXML
    private Label lblCargando;
    @FXML
    private VBox vBoxContenedor;
    @FXML
    private VBox vBoxProductos;
    @FXML
```



```
private TableView<Exámenes> tablaProductos;
@FXML
private TextField txtBuscar;
FilteredList<Exámenes> filteredData;
FilteredList<Exámenes> listaFiltrada;
@FXML
private StackPane stack;
@FXML
private DatePicker fechaInicio;
@FXML
private DatePicker fechaFinal;
@Override
public void initialize(URL url, ResourceBundle rb) {
    Platform.runLater(() -> {
        configuracionVentana(rb);
    });
}
public void configuracionVentana(ResourceBundle rb) {
    progresoCarga.setProgress(0);
    progresoCarga.setVisible(false);
    tablaProductos.setOpacity(0);
    lblCargando.visibleProperty().bind(progresoCarga.visibleProperty());
    fechaInicio.setValue(LocalDate.now().minusYears(1));
    fechaFinal.setValue(LocalDate.now());
    gestor.actualizarListaExámenes();
    configurarTablaInventario();
    tablaProductos.setItems(gestor.getListaExámenes());
}
public void configurarTablaInventario() {
    tablaProductos.setOpacity(0); // No muestra la tabla.
    tablaProductos.setEditable(false); // No es editable
    la tabla
    tablaProductos.setColumnResizePolicy(TableView.CONSTRAINED_RESIZE_POLICY);
    tablaProductos.getSelectionModel().setCellSelectionEnabled(false);
    TableColumn<Exámenes, String> idExamen = new
        TableColumn("N");
    idExamen.setStyle("-fx-alignment: CENTER;");
    idExamen.setMinWidth(50);
    idExamen.setMaxWidth(50);
    idExamen.setPrefWidth(50);
    idExamen.setCellValueFactory(cellData -> new
        SimpleStringProperty(cellData.getValue().getIdExamen() +
            ""));
    TableColumn<Exámenes, String> fecha = new
        TableColumn("Fecha");
    fecha.setStyle("-fx-alignment: CENTER;");
    fecha.setMinWidth(80);
    fecha.setMaxWidth(100);
    fecha.setPrefWidth(80);
    fecha.setCellValueFactory(cellData -> new
        SimpleStringProperty(cellData.getValue().getFecha() +
            ""));
    TableColumn<Exámenes, String> estudio = new
        TableColumn("Estudio");
    estudio.setStyle("-fx-alignment: CENTER;");
}
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
estudio.setCellValueFactory(cellData -> new
    SimpleStringProperty(cellData.getValue().getEstudio() +
        ""));
TableColumn<Examenes, String> lugar = new
    TableColumn("Ciudad");
lugar.setStyle("-fx-alignment: CENTER;");
lugar.setMinWidth(70);
lugar.setMaxWidth(100);
lugar.setPrefWidth(70);
lugar.setCellValueFactory(cellData -> new
    SimpleStringProperty(cellData.getValue().getLugarPaciente()));
TableColumn<Examenes, String> edad = new TableColumn("Edad");
edad.setStyle("-fx-alignment: CENTER;");
edad.setMinWidth(40);
edad.setMaxWidth(600);
edad.setPrefWidth(40);
edad.setCellValueFactory(cellData -> new
    SimpleStringProperty(cellData.getValue().getEdadPaciente()));
TableColumn colBotonAccion = new TableColumn("<Acci n");
colBotonAccion.setMinWidth(100);
colBotonAccion.setMaxWidth(100);
colBotonAccion.setPrefWidth(100);
colBotonAccion.setStyle("-fx-alignment: CENTER;");
colBotonAccion.setSortable(false);
colBotonAccion.setCellValueFactory(
    new Callback<TableColumn.CellDataFeatures<Examenes,
        Boolean>, ObservableValue<Boolean>>() {
        @Override
        public ObservableValue<Boolean>
            call(TableColumn.CellDataFeatures<Examenes, Boolean>
                param) {
            return new SimpleBooleanProperty(param.getValue() !=
                null);
        }
    });
colBotonAccion.setCellFactory(
    new Callback<TableColumn<Examenes, Boolean>,
        TableCell<Examenes, Boolean>>() {
        @Override
        public TableCell<Examenes, Boolean>
            call(TableColumn<Examenes, Boolean> param) {
            return new ButtonCell(tablaProductos);
        }
    });
tablaProductos.getColumns().addAll(idExamen, fecha, estudio,
    lugar, edad, colBotonAccion);
new AparecerDesdeAbajo(tablaProductos).play();
}
@FXML
private void VerPedido(ActionEvent event) {
    if
        (tablaProductos.getSelectionModel().getSelectedItem().size()
            > 0) {
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
Examenes p =
    tablaProductos.getSelectionModel().getSelectedItem();
examenHecho.setDiagnostico(p.getDiagnostico());
examenHecho.setDiagnosticoRemision(p.getDiagnosticoRemision());
examenHecho.setDoctorEncargado(p.getDoctorEncargado());
examenHecho.setEdadPaciente(p.getEdadPaciente());
examenHecho.setEstudio(p.getEstudio());
examenHecho.setFecha(p.getFecha());
examenHecho.setIdExamen(p.getIdExamen());
examenHecho.setImagenes(p.getImagenes());
examenHecho.setInforme(p.getInforme());
examenHecho.setLugarPaciente(p.getLugarPaciente());
examenHecho.setPremedicacion(p.getPremedicacion());
try {
    Parent root =
        FXMLLoader.load(getClass().getResource("/View/pedidoHecho.fxml"));
    Scene scene = new Scene(root);
    Stage stage = new Stage();
    stage.setScene(scene);
    scene.getStylesheets().add(getClass().getResource("/Style/estilo"
        + new
        Archivo().leerArchivoProperties(direccion,
        nombreArchivo, "estilo") + ".css")
        .toExternalForm());
    stage.initStyle(StageStyle.DECORATED);
    stage.getIcons().add(new Image("Images/" + new
        Archivo().leerArchivoProperties(direccion,
        nombreArchivo, "icono")));
    stage.setWidth(1200);
    stage.setHeight(800);
    stage.show();
} catch (Exception ex) {
    System.out.println("Existi un error al abrir la
        ventana pedidoHecho.fxml");
}
}
}
@FXML
private void Actualizar(ActionEvent event) {
    gestor.actualizarListaExamenes(stack);
    tablaProductos.refresh();
}
private void FiltrarPorLocal(ActionEvent event) {
    filtrarTabla();
}
@FXML
private void CambioFecha(ActionEvent event) {
    filtrarTabla();
}
@FXML
private void DespacharPedido(ActionEvent event) {
}
@FXML
private void EliminarPedido(ActionEvent event) {
}
```



```
@FXML
private void TablaProductosMouseClicked(MouseEvent event) {
}
private class ButtonCell extends TableCell<Examenes, Boolean> {
    final Button cellButton = new Button("Ver");
    ButtonCell(final TableView tblView) {
        cellButton.setOnAction((ActionEvent t) -> {
            tablaProductos.getSelectionModel().select((Examenes)
                getTableRow().getItem());
            Examenes p =
                tablaProductos.getSelectionModel().getSelectedItem();
            examenHecho.setDiagnostico(p.getDiagnostico());
            examenHecho.setDiagnosticoRemision(p.getDiagnosticoRemision());
            examenHecho.setDoctorEncargado(p.getDoctorEncargado());
            examenHecho.setEdadPaciente(p.getEdadPaciente());
            examenHecho.setEstudio(p.getEstudio());
            examenHecho.setFecha(p.getFecha());
            examenHecho.setIdExamen(p.getIdExamen());
            examenHecho.setImagenes(p.getImagenes());
            examenHecho.setInforme(p.getInforme());
            examenHecho.setLugarPaciente(p.getLugarPaciente());
            examenHecho.setPremedicacion(p.getPremedicacion());
            try {
                Parent root =
                    FXMLLoader.load(getClass().getResource("/View/pedidoHecho.fxml"));
                Scene scene = new Scene(root);
                Stage stage = new Stage();
                stage.setScene(scene);
                scene.getStylesheets().add(getClass().getResource("/Style/
                    + new
                    Archivo().leerArchivoProperties(direccion,
                        nombreArchivo, "estilo") + ".css")
                        .toExternalForm());
                stage.initStyle(StageStyle.DECORATED);
                stage.getIcons().add(new Image("Images/" + new
                    Archivo().leerArchivoProperties(direccion,
                        nombreArchivo, "icono")));
                stage.setWidth(900);
                stage.setHeight(600);
                stage.show();
            } catch (Exception ex) {
                System.out.println("Existi un error al abrir
                    la ventana pedidoHecho.fxml");
            }
        });
    }
}
@Override
protected void updateItem(Boolean t, boolean empty) {
    super.updateItem(t, empty);
    if (!empty) {
        setGraphic(cellButton);
    } else {
        setGraphic(null);
    }
}
```



```
}
@FXML
private void Buscar(KeyEvent event) {
    filtrarTabla();
}
public void filtrarTabla() {
    try {
        Date fechaIni = Date.valueOf(fechaInicio.getValue());
        Date fechaFin = Date.valueOf(fechaFinal.getValue());
        listaFiltrada = new
            FilteredList<>(gestor.getListasExamenes());
        listaFiltrada.setPredicate(examen -> {
            if ((fechaIni.before(new
                java.util.Date(examen.getFecha())) ||
                fechaIni.equals(new
                java.util.Date(examen.getFecha())))
                && (fechaFin.after(new
                java.util.Date(examen.getFecha())) ||
                fechaFin.equals(new
                java.util.Date(examen.getFecha())))) {
                if
                    (examen.getEstudio().toLowerCase().contains(txtBuscar.getText().to
                    {
                        return true;
                    } else if
                    (examen.getEdadPaciente().toLowerCase().contains(txtBuscar.getText
                    {
                        return true;
                    } else if
                    (examen.getLugarPaciente().toLowerCase().contains(txtBuscar.getTex
                    {
                        return true;
                    } else if
                    (examen.getDoctorEncargado().toLowerCase().contains(txtBuscar.getT
                    {
                        return true;
                    }
                }
                return false;
            }
        }
        return false;
    });
    tablaProductos.setItems(listaFiltrada);
} catch (Exception e) {
}
}
}
```

Pedidohecho.java

```
import Model.Archivo;
import static Model.Prototipo1.direccion;
import static Model.Prototipo1.examenHecho;
import static Model.Prototipo1.imagen;
import static Model.Prototipo1.nombreArchivo;
import Model.WindowAnimation.AparecerDesdeAbajo;
```




```
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.application.Platform;
import javafx.beans.property.SimpleStringProperty;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.ProgressBar;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.input.KeyEvent;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.StackPane;
import javafx.scene.layout.VBox;
import javafx.scene.text.Text;
import javafx.stage.Stage;
import javafx.stage.StageStyle;
import javafx.util.Duration;
import org.apache.commons.net.ftp.FTPClient;
import org.apache.commons.net.ftp.FTPReply;
import org.controlsfx.control.Notifications;
import org.controlsfx.control.PopOver;
public class PedidoHechoController implements Initializable {
    @FXML
    private AnchorPane panelInterno;
    @FXML
    private AnchorPane panelExterno;
    @FXML
    private VBox vBoxGeneral;
    @FXML
    private ProgressBar progresoCarga;
    @FXML
    private Label lblCargando;
    @FXML
    private VBox vBoxContenedor;
    @FXML
    private VBox vBoxProductos;
```



```
@FXML
private TableView<String> tablaProductos;
@FXML
private TextField txtFechaPedido;
@FXML
private TextField txtNumeroDespacho;
@FXML
private StackPane stack;
@FXML
private Label lblEstadoPedido;
@FXML
private VBox vboxLocal;
@FXML
private Text lblLocal;
@FXML
private TextField txtDireccion;
@FXML
private TextField txtUsuarioPedido;
@FXML
private TextField txtDespachaPedido;
@FXML
private TextField txtLocal;
@FXML
private HBox hboxDatosPedido;
@Override
public void initialize(URL url, ResourceBundle rb) {
    Platform.runLater(() -> {
        configuracionVentana(rb);
    });
}
public void configuracionVentana(ResourceBundle rb) {
    progresoCarga.setProgress(0);
    progresoCarga.setVisible(false);
    tablaProductos.setOpacity(0);
    lblCargando.visibleProperty().bind(progresoCarga.visibleProperty());
    mostrarImagenes();
    configurarTablaInventario();
    txtNumeroDespacho.setText(examenHecho.getEstudio());
    txtFechaPedido.setText(examenHecho.getFecha());
    txtUsuarioPedido.setText(examenHecho.getDoctorEncargado());
    txtDespachaPedido.setText(examenHecho.getEdadPaciente());
    txtLocal.setText(examenHecho.getLugarPaciente());
    txtDireccion.setText(examenHecho.getDiagnostico());
}
public void mostrarImagenes() {
    String server = "ftp.cidems.com.ec";
    String username = "cidemscom";
    String password = "Cid%.95E5";
    FTPClient ftp = new FTPClient();
    int respuesta, i;
    String[] lista;
    try {
        System.out.println("CONECTANDO AL SERVIDOR FTP");
        ftp.connect(server);
        ftp.login(username, password);
    }
```



ANEXO F. CÓDIGO FUENTE DEL SOFTWARE SIM 1.1

```
ftp.enterLocalPassiveMode();
respuesta = ftp.getReplyCode();
System.out.println("RESPUESTA " + respuesta);
ftp.changeWorkingDirectory("/cidems.com.ec/marcoftp/" + examenHecho);
if (FTPReply.isPositiveCompletion(respuesta) == true) {
    System.out.println("LISTANDO ARCHIVOS");
    lista = ftp.listNames();
    ObservableList<String> listaCompleta =
        FXCollections.observableArrayList();
    for (i = 2; i < lista.length; i++) {
        listaCompleta.add(lista[i]);
    }
    tablaProductos.setItems(listaCompleta);
} else {
    System.out.println("ERROR DE CONEXION");
}
ftp.logout();
ftp.disconnect();
} catch (IOException e) {
    System.out.println("Error de conexion");
}
}
public void configurarTablaInventario() {
    tablaProductos.setOpacity(0);
    tablaProductos.setEditable(false);
    tablaProductos.setColumnResizePolicy(TableView.CONSTRAINED_RESIZE_POLICY);
    tablaProductos.getSelectionModel().setCellSelectionEnabled(false);
    TableColumn<String, String> categoria = new
        TableColumn("Im genes");
    categoria.setMinWidth(390);
    categoria.setMaxWidth(410);
    categoria.setPrefWidth(390);
    categoria.setCellValueFactory(cellData -> new
        SimpleStringProperty(cellData.getValue()));
    tablaProductos.getColumns().addAll(categoria);
    new AparecerDesdeAbajo(tablaProductos).play();
}
@FXML
private void TablaProductosMouseClicked(MouseEvent event) {
    if
        (tablaProductos.getSelectionModel().getSelectedItem().size()
        > 0) {
        if (event.getClickCount() == 2) {
            String p =
                tablaProductos.getSelectionModel().getSelectedItem();
            imagen = p;
            descargaImagen(examenHecho.getImagenes(), p);
            try {
                Parent root =
                    FXMLLoader.load(getClass().getResource("/View/imagen.fxml"));
                Scene scene = new Scene(root);
                Stage stage = new Stage();
                stage.setScene(scene);
                scene.getStylesheets().add(getClass().getResource("/Style/
                    + new
```




```
public void mostrarMensaje(String mensaje, String titulo, String
imagen, int tiempo, Pos p) {
    Notifications notificationBuilder = Notifications.create()
        .title(titulo) // Muestra el t tulo del Mensaje.
        .text(mensaje) // Muestra el mensaje.
        .graphic(new ImageView(new
            Image(getClass().getResource("/Images/" +
            imagen).toExternalForm())) // Muestra imagen.
        .hideAfter(Duration.seconds(tiempo)) // Duraci n
            que se muestra el mensaje.
        .position(p)
        .onAction((ActionEvent arg0) -> {
            });
    notificationBuilder.show();
}

public void hacerDevoluci n() {
    PopOver popOver = new PopOver();
    popOver.setDetachable(false);
    popOver.setTitle("Devoluci n");
    popOver.setDetached(true);
    popOver.setArrowSize(10);
    popOver.setArrowIndent(10);
    popOver.setCornerRadius(10);
    popOver.setAutoHide(true);
    popOver.setHideOnEscape(true);
    popOver.setOpacity(0.99);
    popOver.arrowLocationProperty().set(PopOver.ArrowLocation.TOP_CENTER);
    VBox contenedor = new VBox();
    contenedor.setPadding(new Insets(10, 10, 10, 10));
    contenedor.setSpacing(10);
    contenedor.setPrefWidth(220);
    contenedor.setPrefHeight(100);
    contenedor.setAlignment(Pos.CENTER);
    Label lblMensaje = new Label("");
    lblMensaje.setStyle("-fx-text-fill: red");
    VBox unidadesProducto = new VBox();
    unidadesProducto.setStyle("-fx-alignment: CENTER;\n"
        + "-fx-font-size: 20pt;");
    Label lblUnidadesProducto = new Label("Devuelven:");
    TextField txtCantidadProducto = new TextField();
    txtCantidadProducto.setStyle("-fx-alignment: CENTER;");
    txtCantidadProducto.setOnKeyTyped((KeyEvent event) -> {
        if ("1234567890".contains(event.getCharacter())) {
        } else {
            event.consume();
        }
    });
    txtCantidadProducto.setOnAction(new
        EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                try {
                } catch (Exception ex) {
                    popOver.hide();
                }
            }
        });
}
```



```
        }
    }
});
unidadesProducto.getChildren().addAll(lblUnidadesProducto,
    txtCantidadProducto);
Button btnGuardarProducto = new Button("Guardar");
btnGuardarProducto.setOnAction((ActionEvent e) -> {
    try {
        } catch (Exception ex) {
            popOver.hide();
        }
    });
contenedor.getChildren().addAll(unidadesProducto,
    lblMensaje, btnGuardarProducto);
popOver.setContentNode(contenedor);
popOver.show(panelInterno, (panelInterno.getWidth() / 2) +
    150, 150);
}
}
```

ImagenController.java

```
import java.io.File;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
public class ImagenController implements Initializable {
    @FXML
    private ImageView imagenVista;
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        File file = new File("E:\\imgs\\dato.jpg");
        Image image = new Image(file.toURI().toString());
        imagenVista.setImage(image);
    }
}
```

Anexo G

STSIVA XXI, 2016



UNIVERSIDAD DE CUENCA
desde 1867



Bucaramanga, July 22, 2016

Dear author,

We are pleased to inform you that your submission to the 2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA) has been accepted. This year, the STSIVA will be hosted at Universidad Pontificia Bolivariana seccional Bucaramanga, from August 31 to September 2, 2016.

Submission information:

Authors: Juan Molina, Marco Bacuilima, Carlos Villie Morocho, Ruben Medina, Alexandra La Cruz
Title: Video and Imaging Gastroenterological Medical Equipment Oriented to Telemedicine
ID: 46

Decision: Accepted (Poster Presentation)

Camera ready submission deadline: August 12, 2016

STSIVA makes no distinction in terms of quality between oral and poster presentations. Therefore, all accepted papers written in English (categories C2, C3 and C4) will be published in the conference proceedings and will be available in IEEE Xplore®.

Please visit <http://stsiva.org> for more information.

We look forward to seeing you in Bucaramanga!

Sincerely,

Handwritten signature of Miguel Altuve in black ink.

Miguel Altuve, PhD
STSIVA 2016 General Chair

Handwritten signature of Luis Ángel Silva in black ink.

Luis Ángel Silva, PhD
STSIVA 2016 Program Chair

Handwritten signature of Hernán Benítez in black ink.

Hernán Benítez, PhD
IEEE SPS Colombian Chapter

Anexo H

Manual de Usuario

SIM 1.1 El uso del software de Sistema de Informes Médicos 1.1 (SIM 1.1) es muy sencillo e intuitivo, al iniciar el software se observa la figura H.1, se observa que en la parte izquierda se tiene la sección donde se encuentra una imagen que representa al usuario conectado, debajo de ella el botón **Estudios**, y al final otro botón que es para conocer acerca de la información del software y sus autores, este tiene de texto **Acerca de**. Al presionar el botón "Estudios", muestra

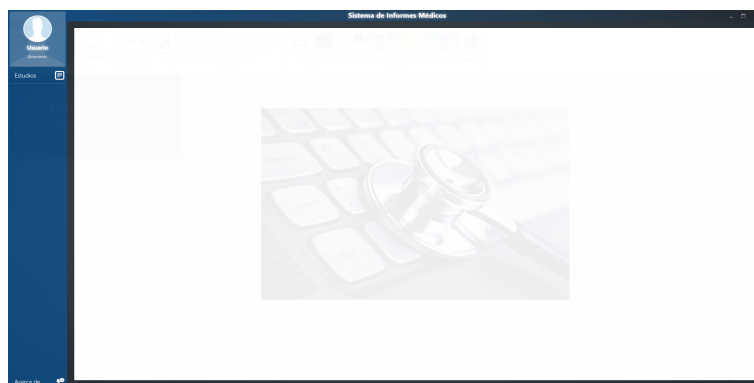
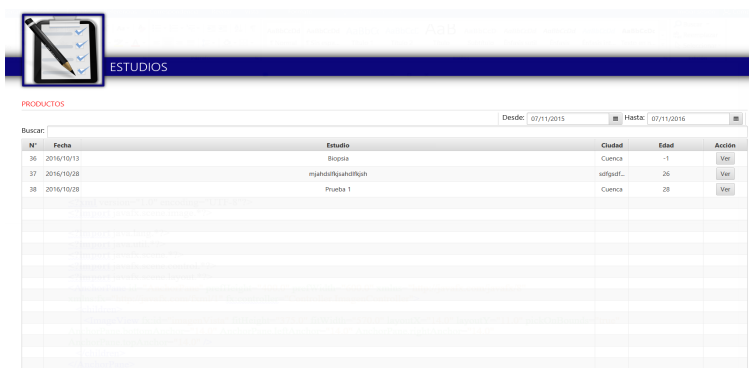


Figura H.1: Ventana principal de SIM 1.1

en la parte derecha del programa una vista donde se puede ver los estudios que se encuentran en el servidor web en ese momento. En la figura H.2 se puede observar también que tiene un buscador, aquí se puede buscar por estudio,

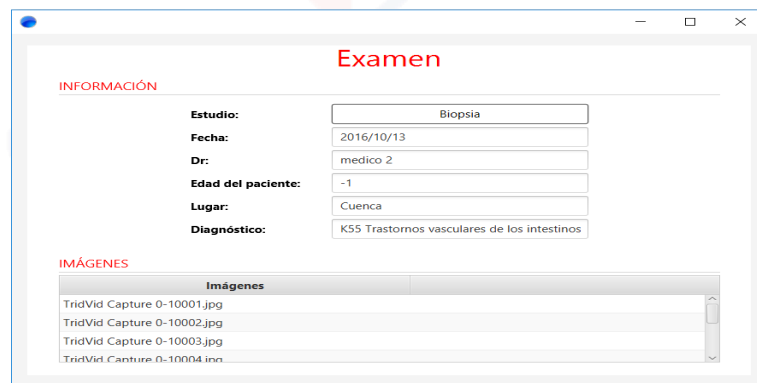
ciudad o edad filtrándonos toda la lista de estudios que se tenga, también consta para un filtrado mediante fechas y así obtener los estudios más actuales. Para utilizar el buscador solo es necesario escribir lo que deseamos encontrar y automáticamente irán apareciendo los estudios con las características que se busca. En la tabla donde se encuentran los estudios con su respectiva información. En la parte dere-



ID	Fecha	Estudio	Ciudad	Edad	Acción
36	2016/10/13	Bispos	Cuenca	1	Ver
37	2016/10/28	mpahidiljyaidllqah	Alfajal...	26	Ver
38	2016/10/28	Prueba 1	Cuenca	28	Ver

Figura H.2: Ventana donde muestra los estudios.

cha se observa que cada estudio consta de un botón donde al presionarlo nos abrirá una nueva ventana. La siguiente figura [H.3](#) muestra la nueva ventana donde está mayor información acerca del estudio seleccionado, ya se puede ver el estudio realizado, su diagnóstico, el lugar, la fecha cuando se realizó, la edad del paciente y el Dr que hizo el estudio. Además, consta con una tabla donde se encuentra la lista de imágenes del estudio. Para ver las imágenes, solo es necesario de hacer doble click en el nombre de cualquier imagen y automáticamente se abrirá una nueva ventana con la imagen del exámen. Para ver otra imagen solo hay que cerrar la última ventana y seleccionar otra imagen. De la misma forma, si se desea ver otro estudio, solo hay que cerrar la ventana del estudio que se escogió y buscar uno nuevo.



The screenshot shows a software window titled "Examen" with a light blue border. The window is divided into two main sections: "INFORMACIÓN" and "IMÁGENES".

INFORMACIÓN

Estudio:	Biopsia
Fecha:	2016/10/13
Dr:	medico 2
Edad del paciente:	-1
Lugar:	Cuenca
Diagnóstico:	K55 Trastornos vasculares de los intestinos

IMÁGENES

Imágenes
TridVid Capture 0-10001.jpg
TridVid Capture 0-10002.jpg
TridVid Capture 0-10003.jpg
TridVid Capture 0-10004.jpg

Figura H.3: Información de un estudio realizado.

Bibliografía

- [1] *Para qué sirve y cómo funciona una endoscopia*, 2016 (Visita: Septiembre 15, 2016). <http://mejorconsalud.com/para-que-sirve-y-como-funciona-una-endoscopia>.
- [2] *Sistema de endoscopia Evis Exera III de Olympus*, 2016 (Visita: Octubre 2, 2016). <http://www.elhospital.com/temas/Sistema-de-endoscopia-Evis-Exera-III-de-Olympus+8094145>.
- [3] J. Pickering. Analogue to digital and digital to analogue converters (adcs and dacs): A review update. *Metron Designs Ltd, Norwich, UK*, 2014.
- [4] H. Mick. *The following discussion of NTSC video standards is from the first edition of Voice Vision*. Focal Press, 2012.
- [5] *Hello, Arduino: Write Code That Interacts with the Real World*, 2016 (Visita: Octubre 8, 2016). <http://www.codemag.com/article/1305081>.
- [6] *Arduino Micro*, 2016 (Visita: Octubre 8, 2016). <https://www.pololu.com/product/2188>.
- [7] *Boss FS-5U Nonlatching Footswitch*, 2016 (Visita: Agosto 1, 2016). <http://www.pololu.com/product/2188>.



[//www.musiciansfriend.com/accessories/boss-fs-5u-nonlatching-footswitch](http://www.musiciansfriend.com/accessories/boss-fs-5u-nonlatching-footswitch).

- [8] *Arduino UNO*, 2016 (Visita: Agosto 5, 2016). <https://www.arduino.cc/en/Main/ArduinoBoardUno>.
- [9] *Telemedicina-FMH*, 2011 (visita: Septiembre 18, 2016). <http://www.encore-usa.com/es/product/ENMVG>.
- [10] J. Magaz. *Tratado elemental de Fisiología Humana*. Imprenta y Librería de Nicolás Moya, 1985.
- [11] E. Marieb. *Anatomía y Fisiología Humana*. Pearson. Addison Wesley, 2008.
- [12] A. Guyton and J. Hall. *Tratado de fisiología médica*. Elsevier, 2011.
- [13] C. Boticario and M. Cascales Angosto. *Digestión y metabolismo energético de los nutrientes*. UNED, Centro de Plasencia, 2012.
- [14] Abreu García, L. Martínez, A. Peinado, C. Panero, JI. Garrido, and A. Mendoza. *Gastroenterología: endoscopia diagnóstica y terapéutica*. Médica Panamericana, 2007.
- [15] National Institutes of Health et al. “el aparato digestivo y su funcionamiento”. *Digestive Diseases–Ulcerative Colitis, NDDIC*, 2008.
- [16] R Llanio Navarro. Gastroenterología: Principios básicos y pruebas diagnósticas. *La Habana: Editorial Pueblo y Educación*, 53:349–55, 1991.
- [17] M. Rometti. La imagen del intestino delgado. *Revista de Gastroenterología de México 2010*, 75(Supl 2), 2010.

- [18] Novick AC Kavoussi LR. Duffey b, monga m. principles of endoscopy. *Campbell-Walsh Urology, Philadelphia*, 2012.
- [19] GM Lentz, R. Lobo, D. Gershenson, and VL. Katz. Lentz gm. endoscopy: hysteroscopy and laparoscopy. *Comprehensive Gynecology, Philadelphia*, 2012.
- [20] *Telemedicina-FMH*, 2012 (visita: Octubre 11, 2016). <http://www.scribd.com/doc/7125380/TelemedicinaFMH>.
- [21] MB Cabral, P Galván, and V Cane. Telemedicine: fine and applicability. *Memorias del Instituto de Investigaciones en Ciencias de la Salud*, 6(1):40–44, 2008.
- [22] Olga Ferrer Roca. *Telemedicina*. Ed. Médica Panamericana, 2001.
- [23] A. dos Santos, A. Fernández, H. Alves, C. de Souza, M. de Melo, and L. Messina. Desarrollo de la telesalud en américa latina. 2013.
- [24] *Scielo-Cuba*. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1729-519X2010000100017.
- [25] C. Ruiz, A. Zuluaga, and A. Trujillo. Telemedicina: Introducción, aplicación y principios de desarrollo. 2007.
- [26] E. Armanie, J. Bohórquez, M. Chiurillo, E. Valderrama, J. Martínez, N. Granda, L. Bohórquez, E. Dugarte, J. Vegas, and N. Contreras. Hallazgos clínicos, endoscópicos e histológicos asociados a la infección por helicobacter pylori considerando los genotipos *CAG A* y



- VAC A en pacientes con dispepsia: Servicio de gastroenterología. *Gen*, 64(2):76–81, 2010.
- [27] C. Nico and H. Gerard. *Y/C* separation of composite color video signals using samples with non-opposite sub-carrier phases. *Video Processing and Visual Perception Group*, 2015.
- [28] B. Patel and H. Bhavesh. Analog to digital converter (adc) review. *International Journal of Emerging Trends in Electrical and Electronics (IJETEE – ISSN: 2320-9569)*, 2013.
- [29] D. Cooper and D. Helfrick. Instrumentación electrónica moderna y técnicas de medición. *Editorial Prentice-Hall Hispanoamericana, SA México*, 1991.
- [30] H. Lewis. A 10-b 20-msample/s analog-to-digital converter. *Journal of Solid-State Circuits*, 1992.
- [31] B. Trevor. Cq-datv. *Production Team / Ian Pawson*, 2014.
- [32] J. Duffield and D. Altmanshofer. Television receiver having the capability to associate any hdtv and any ntsc channel, October 24 1995. US Patent 5,461,427.
- [33] A. Leavy. *Developing a Sensor Payload Module for the Pioneer Robot*. PhD thesis, Murdoch University, 2012.
- [34] Tecktronics. *A Guide to Standard and High-Definition Digital Video Measurements*. Primer, 2009.
- [35] Monolithic. Ams1117 datasheet. *Advanced Monolithic Systems, Inc*, 2016.
- [36] STMicroelectronics. M24c16, m24c08, m24c04, m24c02, m24c01. datasheet. *STMicroelectronics*, 2016.



- [37] D. Ausilio. *Arduino: A low-cost multipurpose lab equipment. Behavior research methods*. Issue 2, 2012.
- [38] SolidWorks Corporation. *Guía del estudiante para el aprendizaje del software SolidWorks*. DS SolidWorks, 2010.
- [39] Universidad de Cantabria. *CAD 3D*. Digteq, 2010.
- [40] *Wampserver: a Windows web development environment*, 2015 (Visita: Septiembre 20, 2016). <http://www.wampserver.com/en/>.
- [41] J. Weaver. *Javafx script: Dynamic java scripting for rich internet/client-side applications*. Apress, 2007.
- [42] A. Watt and N. Eng. *Database Design - 2nd Edition*. BCCampus, 2012.
- [43] A. Silberschatz. *FUNDAMENTOS DE BASES DE DATOS*. Bell Laboratories, 2007.
- [44] A. Holzner. *JAVA 2*. Coriolis, 2009.
- [45] G. Wielenga. *Beginning netbeans ide: For java developers*. Apress, 2015.
- [46] *Introducción a CSS*, 2008 (Visita: Julio 18, 2016). <http://librosweb.es/libro/css/>.
- [47] J. Pavón. *Estructura de las Aplicaciones Orientadas a Objetos El patrón Modelo-Vista-Controlador (MVC)*. Programación Orientada a Objetos Facultad de Informática, 2010.
- [48] GitHub. Webcam-capture. *GitHub*, 2015.
- [49] GitHub. Library to generate ms word documents from java code. *GitHub*, 2016.



- [50] *Arduino*, 2016 (Visita: Octubre 14, 2016). <https://www.arduino.cc/>.
- [51] *Encore*, 2016 (Visita: Octubre 14, 2016). <http://www.encore-usa.com/es/>.
- [52] *Netbeans*, 2016 (Visita: Octubre 14, 2016). <https://netbeans.org/>.
- [53] *Oracle*, 2016 (Visita: Octubre 14, 2016). <https://www.oracle.com/index.html>.
- [54] *Webcam Capture API*, 2016 (Visita: Octubre 14, 2016). <http://webcam-capture.sarxos.pl/>.
- [55] *ControlsFX*, 2016 (Visita: Octubre 14, 2016). <http://fxexperience.com/controlsfx/>.

UNIVERSIDAD DE CUENCA
desde 1867