



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:

INGENIERO EN SISTEMAS INFORMÁTICOS

TEMA:

DESARROLLO DE UN SISTEMA WEB DE GESTIÓN DE FICHAS MÉDICAS PARA LA ORGANIZACIÓN “*SALUD INTEGRAL PARA TODOS*”, BAJO LA METODOLOGÍA SCRUM.

AUTOR:

Luis Fernando Ordóñez Armijos

TUTOR:

Ing. Renato Mauricio Toasa Guachi, Msc.

QUITO, ECUADOR

2019

DECLARACIÓN DE AUTORÍA

El documento de tesis con título: “**DESARROLLO DE UN SISTEMA WEB DE GESTIÓN DE FICHAS MÉDICAS PARA LA ORGANIZACIÓN ‘SALUD INTEGRAL PARA TODOS’, BAJO LA METODOLOGÍA SCRUM.**”, ha sido desarrollado por el señor Luis Fernando Ordóñez Armijos con C.C. No. 1725351736 persona que posee los derechos de autoría y responsabilidad, restringiéndose la copia o utilización de la información de esta tesis sin previa autorización.

Luis Fernando Ordóñez Armijos

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Titulación certifico:

Que el trabajo de titulación “**DESARROLLO DE UN SISTEMA WEB DE GESTIÓN DE FICHAS MÉDICAS PARA LA ORGANIZACIÓN ‘SALUD INTEGRAL PARA TODOS’, BAJO LA METODOLOGÍA SCRUM.**”, presentado por Luis Fernando Ordóñez Armijos, estudiante de la Carrera Ingeniería en Sistemas Informáticos, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D. M., 16 de febrero de 2019

TUTOR

Ing. Renato Toasa, Msc

AGRADECIMIENTOS

A mis papis, Zoila y Luis, por ser los máximos referentes de vida, por su amor incondicional, por su noble dedicación y esfuerzo, si algo bueno soy, se lo debo todo a ustedes.

A mis hermanos Isabel y David, por no negarme su ayuda y cariño, nadie ni nada puede ocupar el lugar que ustedes tienen en mi alma.

A mi Andreita, por sacar siempre la mejor versión de mí y creer que merezco todo lo bueno que hay en el mundo; quédese conmigo y veamos florecer los girasoles que hemos sembrado juntos, sea mi amor con usted por siempre.

A mi pequeña Valentina, por pintar mi vida de colores cuando he visto amaneceres grises, tú tienes mi corazón entero.

A mi querida hermana Erikita, por conspirar junto a mí llevando a cabo nuestro plan maestro, que tu corazón sepa que te quiero.

A mi Limón, por estar conmigo cuando la madrugada era fría y no había nada más calentito que tu compañía.

A mi Tutor de Tesis, Renato Toasa, por poder contar con su buena voluntad en momentos cruciales de mi carrera.

Luis

DEDICATORIA

A padres y mis hermanos que merecen mucho más de lo que pueda ofrecer.

A mis amores, Andreita y Valentina, son mi más hermoso presente.

A Erikita, es bueno saber que cuento contigo.

Luis

TABLA DE CONTENIDOS

RESUMEN	ix
ABSTRACT	x
INTRODUCCIÓN	1
ANTECEDENTES DE LA SITUACIÓN OBJETO DE ESTUDIO	1
PLANTEAMIENTO DEL PROBLEMA	1
JUSTIFICACIÓN	2
OBJETIVOS	3
GENERAL.....	3
OBJETIVOS ESPECÍFICOS.....	3
DESCRIPCIÓN DE LOS CAPÍTULOS.....	3
1 CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.....	4
1.1 ESTADO DEL ARTE	4
1.2 LÓGICA DEL NEGOCIO.....	6
1.3 HERRAMIENTAS TÉCNICAS.....	7
1.3.1 Ubuntu Server:.....	8
1.3.2 PostgreSQL.....	8
1.3.3 WildFly.....	9
1.3.4 Java Enterprise Edition (JEE)	9
1.3.5 ICEfaces.....	15
1.3.6 METODOLOGÍA DE DESARROLLO	15
1.3.6.1 EQUIPO SCRUM	16
1.3.6.2 Herramientas de SCRUM.....	17
1.4 ALTERNATIVAS DE SOLUCIÓN	17
2 CAPÍTULO II. MARCO METODOLÓGICO.....	19
2.1 TIPO DE INVESTIGACIÓN	19
2.2 RECOPIACIÓN DE INFORMACIÓN.....	19
2.2.1 TÉCNICAS DE RECOPIACIÓN DE INFORMACIÓN	19
3 CAPÍTULO III. PROPUESTA	22
3.1 DIAGRAMA DE PROCESOS	22
3.2 ESPECIFICACIÓN DE REQUERIMIENTOS	23
3.2.1 ÁMBITO DEL SOFTWARE	23
3.2.2 FUNCIONES DEL PRODUCTO.....	25
3.2.3 CARACTERÍSTICAS DE LOS USUARIOS DEL SISTEMA.....	28
3.2.4 RESTRICCIONES	29
3.2.5 REQUISITOS	29
4 CAPÍTULO IV. IMPLEMENTACIÓN	32

4.1	DISEÑO GENERAL	32
4.2	ESQUEMA DE LA BASE DE DATOS.....	35
4.3	DIAGRAMA DE LA ARQUITECTURA DEL SISTEMA	37
4.4	DISEÑO DE INTERFACES	37
4.5	ESTÁNDARES DE PROGRAMACIÓN UTILIZADOS	41
4.6	PRUEBAS	44
4.6.1	PRUEBAS DE FUNCIONALIDAD	44
4.6.2	PRUEBAS DE ESTRÉS, CARGA Y RENDIMIENTO	46
4.7	IMPLEMENTACIÓN	53
4.7.1	PLAN DE IMPLEMENTACIÓN.....	53
4.7.2	REQUERIMIENTOS	54
4.7.3	MANUAL DE USUARIO.....	54
4.7.4	MANUAL TÉCNICO	54
4.7.5	PLAN DE CAPACITACIÓN	54
5	CONCLUSIONES	58
6	RECOMENDACIONES	59
7	REFERENCIAS BIBLIOGRÁFICAS	60
8	ANEXOS	63

LISTA DE FIGURAS

Figura 1.1: Ingreso al Sistema RDACAA	6
Figura 1.2: Registro de Fichas Médicas	7
Figura 1.3: Aplicaciones distribuidas de múltiples niveles	11
Figura 1.4: Servidor Java EE y sus Contenedores.....	12
Figura 1.5: Metodología Scrum.....	16
Figura 3.1: Diagrama del Proceso Actual	22
Figura 4.1: Esquema de la base de Datos saludemlogs	36
Figura 4.2: Esquema de la Base de Datos saludem	36
Figura 4.3: Diagrama de Arquitectura del Sistema	37
Figura 4.4: Interfaz de Ingreso al Sistema.....	37
Figura 4.5: Interfaz de Registro de Pacientes y Profesionales médicos	38
Figura 4.6: Interfaz de Agendamiento de Citas médicas.....	39
Figura 4.7: Interfaz de Registro de Atención médica.....	40
Figura 4.8: Interfaz de Reportes Gráficos	41
Figura 4.9: Añadir Grupo de Hilos.....	46
Figura 4.10: Personalizar grupos	47
Figura 4.11: Añadir petición HTTP	47
Figura 4.12: Configurar petición HTTP	47
Figura 4.13: Añadir Resumen de Resultados	48
Figura 4.14: Reporte de Resumen	48
Figura 4.15: Gráfica de Resultados	49
Figura 4.16: Árbol de Resultados.....	50
Figura 4.17: Resumen primer escenario.....	50
Figura 4.18: Resumen segundo escenario	50
Figura 4.19: Resumen tercer escenario	51
Figura 4.20: Resumen cuarto escenario	51
Figura 4.21: Gráfico de resultados cuarto escenario	51
Figura 4.22: Plan de Implementación.....	53

LISTA DE TABLAS

Tabla 1.1: Comparativa de Alternativas.....	18
Tabla 2.1: Modelo de Ficha de Observación.....	20
Tabla 3.1: Historia 01 - Configuración y parametrización.....	25
Tabla 3.2: Historia 02 – Usuarios y Perfiles	25
Tabla 3.3: Historia 03 –Información y parametrización médica.....	26
Tabla 3.4: Historia 04 – Citas médicas.....	27
Tabla 3.5: Historia 05 – Atención a pacientes.....	27
Tabla 3.6: Historia 06 – Órdenes de laboratorio	28
Tabla 3.7: Perfiles de grupos de usuarios.....	28
Tabla 3.8: Product Backlog	31
Tabla 4.1: Sprint Backlog – Sprint 0.....	32
Tabla 4.2: Sprint Backlog – Sprint 1	32
Tabla 4.3: Sprint Backlog – Sprint 2.....	33
Tabla 4.4: Sprint Backlog – Sprint 3.....	34
Tabla 4.5: Sprint Backlog – Sprint 4.....	34
Tabla 4.6: Sprint Backlog – Sprint 5.....	35
Tabla 4.7: Pruebas de Funcionalidad	44
Tabla 4.8: Plan de Implementación.....	53
Tabla 4.9: Plan de Capacitación	55
Tabla 4.10: Comparativa Proceso Actual Vs. Automatizado.....	56
Tabla 4.11: Resultados de la entrevista	56

RESUMEN

El presente proyecto tiene como objetivo implementar un sistema web de gestión de fichas médicas para la organización “*SALUD INTEGRAL PARA TODOS*”, bajo la metodología *Scrum* aplicada al desarrollo ágil de software por la importante característica de permitir su aplicación a grupos de trabajo con un número limitado de personas y la capacidad de ofrecer en poco tiempo entregables que satisfagan las necesidades del usuario final, mismo que forma parte del grupo y está en constante interacción con el desarrollador del Sistema.

Luego de la investigación llevada a cabo utilizando como herramienta fichas de observación se implementa un procedimiento automatizado para el agendamiento de citas médicas que van ligadas a un horario definido para cada profesional médico para posteriormente hacer el registro histórico de atención a pacientes.

Finalmente se concluye con el trabajo de implementación de la solución desarrollada para la organización.

Palabras clave: Desarrollo de software, Scrum, Metodología Ágil, observación, fichas médicas, historial médico.

ABSTRACT

The objective of this project is to implement a web management system for medical records for the organization " *SALUD INTEGRAL PARA TODOS* ", under the Scrum methodology applied to the agile development of software due to the important characteristic of allowing its application to working groups with a number limited of people and the ability to offer deliverables in a short time that satisfy the needs of the end user, who is part of the group and is in constant interaction with the System developer.

After the research carried out using observation tools as an observation tool, an automated procedure is implemented for scheduling medical appointments that are linked to a defined schedule for each medical professional to subsequently make the historical record of patient care.

Finally, it concludes with the work of implementation of the solution developed for the organization.

Keywords: Software development, Scrum, Agile Methodology, observation, medical records, medical history.

INTRODUCCIÓN

ANTECEDENTES DE LA SITUACIÓN OBJETO DE ESTUDIO

La tecnología crece a pasos agigantados y con ella van apareciendo nuevas necesidades en el mercado, la minimización de algunos costos han sido el eje fundamental para pensar en alternativas de rápida implementación, que ofrezcan niveles adecuados de seguridad, integridad y disponibilidad en cuanto a información médica de pacientes en las instituciones de salud.

En las principales ciudades del país existen varias empresas que ofrecen sistemas similares para la automatización y verificación de datos, principalmente realizados en plataformas de código cerrado, por lo cual los usuarios deben pagar licencias por el simple hecho de tener instalada la herramienta lo que conlleva, en la mayoría de los casos, desembolsos significativos para la obtención de soporte y mantenimiento.

“*Salud Integral para Todos*” es una organización sin fines de lucro que brinda servicios a la comunidad tales como ginecología, odontología, óptica y optometría, hasta ahora sigue gestionando la información de pacientes sin contar con una herramienta automatizada, que permita tratar y almacenar información médicamente relevante de un forma sistematizada y ágil.

Teniendo en cuenta lo anteriormente expuesto para el presente proyecto se hará uso de plataformas de código abierto dando como resultado una herramienta eficaz con la libertad de ser utilizada, estudiada, distribuida y mejorada sin ninguna restricción, claro que eso no significa cero inversión de capital, sin embargo, la independencia de las organizaciones para invertir de la forma más conveniente sus recursos económicos está garantizada.

PLANTEAMIENTO DEL PROBLEMA

Actualmente la organización para la gestión de la información asociada con el centro médico, los profesionales, los pacientes y su historial médico, realiza registros de forma manual usando hojas de cálculo Excel y fichas impresas en papel. Cabe destacar que tampoco cuentan con un procedimiento de agendamiento de citas médicas lo que no permite la debida organización de los horarios de atención de los profesionales médicos.

La mayor parte de los reclamos contra las instituciones de salud de modo general son quejas sobre mala atención, información equivocada, largo tiempo de espera al teléfono para agendar una cita o un examen, etc. Lo curioso es percibir que en plena era de la información, muchos hospitales, centros de diagnósticos y laboratorios insisten en realizar los procesos de forma manual. Esta situación dificulta contar con información pertinente en el momento adecuado obligando a los empleados de la organización a crear un registro nuevo del paciente debido a que la ficha inicial se extravió entre otros papeles.

JUSTIFICACIÓN

El presente proyecto tiene como objetivo implementar un sistema web de gestión de fichas médicas para la organización SALUD INTEGRAL PARA TODOS, bajo la metodología *Scrum* aplicada al desarrollo de software.

La gestión de fichas médicas mediante un sistema web generará impacto positivo en la confianza del paciente hacia el profesional de la salud, porque facilitará la lectura e interpretación de las prescripciones médicas, lo que por los malos hábitos caligráficos del médico ha sido difícil de conseguir a lo largo de los años.

Es importante considerar que los documentos físicos en papel son susceptibles a estropearse, romperse o extraviarse y su almacenamiento incide en costos, eso sin contar el espacio que ocupan. En su lugar, al almacenar de forma digital y sistematizada las fichas médicas se garantiza niveles adecuados de seguridad ofreciendo confidencialidad, integridad y disponibilidad de la información asociada a los pacientes de la organización que perdurará en el tiempo y su consulta se facilitará considerablemente.

El desarrollo de este sistema brindará también una solución de agendamiento de citas como instrumento que aportará significativamente a la mejora de la calidad de servicio ofrecida a sus pacientes agilizando la atención a cargo de los profesionales médicos. Además, el tiempo de consulta se optimiza, haciendo que la jornada laboral de la organización médica sea eficiente, beneficiando así a la organización que se mostrará confiable hacia el público que la frecuenta.

OBJETIVOS

GENERAL

Desarrollar un sistema web para la gestión de fichas médicas para la organización “*Salud Integral para Todos*”, usando la plataforma de programación JEE para aplicaciones web y la metodología de desarrollo ágil *Scrum*, con el fin de mejorar la calidad del servicio que ofrece a sus pacientes.

OBJETIVOS ESPECÍFICOS

- Revisar el proceso actual de citas médicas
- Determinar requerimientos del sistema
- Diseñar un modelo de base de datos que permita representar los datos del centro médico, los profesionales, los pacientes y su registro histórico
- Desarrollar un sistema para la gestión de fichas médicas
- Realizar de pruebas del sistema y calificación con usuarios
- Instalar y configurar los servicios necesarios previo despliegue del sistema.
- Implantar el sistema.

DESCRIPCIÓN DE LOS CAPÍTULOS

En el Capítulo I se contextualiza la problemática sobre la que la presente investigación pretende proponer solución, se hace referencia a investigaciones sobre la misma problemática o afines y se fundamenta teóricamente las herramientas a ser utilizadas.

En el Capítulo II se describe la metodología a ser utilizada durante todo el proyecto para solucionar la problemática antes descrita.

En el Capítulo III se hace referencia al proceso actual no automatizado utilizado por la Institución para llevar a cabo su giro de negocio y se presenta una propuesta automatizada que agilite sus tareas, generando valor agregado para sí misma y sus clientes.

En el Capítulo IV se describen las actividades para la realización de las pruebas del sistema seguido de un plan de implementación y finalizando con el plan de capacitación.

Finalmente, el presente trabajo culmina presentando las conclusiones a la que se llega y un grupo de puntos de recomendación relevantes.

1 CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

1.1 ESTADO DEL ARTE

Actualmente varios trabajos científicos trabajan con datos médicos, (Toasa, Maximiano, Reis, & Guevara, 2018), realizan un Dashboard basado en tecnologías web para mostrar información de encuestas de un Centro médico de Portugal. Evidenciando el impacto de datos médicos a nivel mundial. En 2016, la Organización Panamericana de Salud publicó un Análisis Sobre la Situación Actual y Recomendaciones para la Región, en donde fueron documentadas las opiniones vertidas por los participantes de un foro que por iniciativa de la Red Latinoamericana y del Caribe para el Fortalecimiento de los Sistemas de Información de Salud (RELAC SIS) fue llevado a cabo con la participación de 200 personas. El tema que se discutió en el foro trató sobre la situación de los **Registros Médicos Electrónicos (RME)** en América Latina y el Caribe (Organización Panamericana de la Salud, 2016).

Como resultado del análisis los participantes del foro identificaron beneficios y riesgos asociados a los **RME** listados a continuación:

Beneficios

- Con la ayuda de la tecnología adecuada se puede asegurar la identificación precisa de los pacientes;
- Se puede garantizar niveles adecuados la seguridad de los datos personales del paciente;
- Facilitan la integración con sistemas administrativos lo que puede agilizar el agendamiento y programación de consultas médicas;
- Si están correctamente implementados pueden ayudar a los profesionales a que dispongan menor tiempo dedicado a tareas administrativas;
- Permite compartir información clínica entre profesionales sanitarios tanto de la unidad como de diferentes hospitales;
- Mejoran la seguridad jurídica del profesional por el registro electrónico de la información del paciente;
- Incrementan la calidad asistencial al disponer de información del paciente inmediato y en tiempo real;

-
- Mejoran la eficiencia del proceso asistencial al evitar desperdicio de medicamentos y estudios de diagnóstico innecesarios;
 - Mejoran el proceso de toma de decisiones en todos los niveles;
 - Proporcionan indicadores de salud en tiempo real, para la toma de decisiones;
 - Disminuyen los errores por ilegibilidad en las recetas de los médicos;
 - Disminuyen del gasto en papel y ayudan a conservar el medio ambiente;
 - Facilitan la vigilancia epidemiológica.

Riesgos asociados

- Complejidad de implementación de este tipo de sistemas informatizados, ya que muchas veces son desestimados por las autoridades que gerencian los sistemas de salud;
- Riesgos relacionados a la confidencialidad, y si bien también fue indicado como un beneficio, durante la discusión de los riesgos la confidencialidad y la violación al derecho de privacidad de los pacientes y familiares fue claramente visualizado. En especial por cómo facilita el ingreso desde cualquier lugar a información sensible;
- Informatizar procesos ineficientes, o que no funcionan correctamente, y pensar que con un sistema se van a poder solucionar.

El estudio citado anteriormente da a entender que el tema tratado en el presente proyecto está vigente a nivel latinoamericano y queda mucho trabajo por hacer al respecto.

En Ecuador se puede evidenciar que al menos las entidades públicas de salud de las que el Ministerio de Salud ejerce rectoría ya cuentan con un sistema para el *Registro Diario Automatizado de Consultas y Atenciones Ambulatorias (RDACAA)* desde diciembre del 2012 cuando se emitió el Acuerdo Ministerial No. 00002687, que dispone el uso obligatorio de este a nivel nacional (Dirección Nacional de Estadística y Análisis de la Información de Salud, 2015).

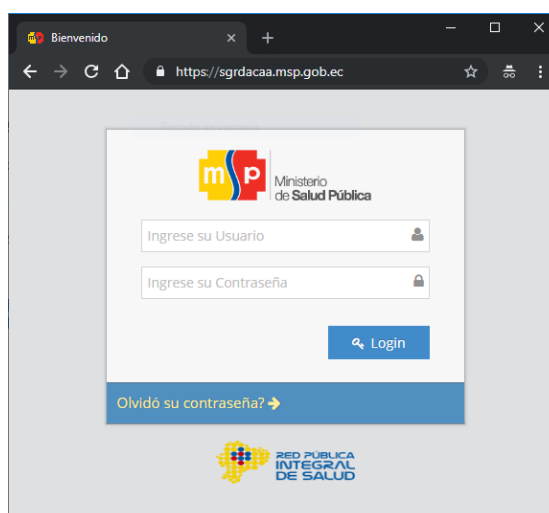


Figura 1.1: Ingreso al Sistema RDACAA

Fuente: <https://sgrdaca.msp.gob.ec/>

“Dicho instrumento es una fuente de consulta que servirá para contextualizar, orientar y facilitar el trabajo de los Profesionales Médicos/as, Obstetrices/Obstetras, Psicólogos/as, Odontólogos/as, responsables de la recopilación y almacenamiento de las consultas y atenciones; adicionalmente, permitirá determinar el tipo de consulta primera o subsecuente, el diagnóstico según la Clasificación Estadística Internacional de Enfermedades y Problemas Relacionados con la Salud CIE 10, y facilitará el cálculo de indicadores por medio de la Producción Ambulatoria” (Ministerio de Salud Pública del Ecuador, 2013).

1.2 LÓGICA DEL NEGOCIO

La organización *Salud Integral para Todos* centra su lógica de negocio en la atención a pacientes en las especialidades como ginecología, odontología, óptica y optometría; haciendo énfasis en la especialidad de óptica y optometría ofreciendo a sus clientes una alta gama de lentes y exámenes orientados a la visión humana, lo que también involucra la gestión de órdenes para ensamblaje de lentes en laboratorios especializados. El registro de los datos médicos de los pacientes se lo lleva en fichas impresas en papel y llevan un proceso no óptimo para su registro.

(HOSPITAL SANT JOAN DE DÉU BARCELONA, 2013) Define, haciendo referencia a la Organización Mundial de la salud, un **registro de pacientes** como *“un fichero de documentos que contiene información uniforme acerca de personas individuales, recogida de forma sistemática e integral, para que sirva a unos objetivos previamente establecidos científicos, clínicos o de política sanitaria”*.

Las historias clínicas se pueden utilizar como base para la investigación de fenómenos epidemiológicos por lo que es fundamental archivar y gestionar toda información relevante que pueda producir una atención sanitaria de pacientes que acuden a una institución médica, es importante mencionar que no se requiere un consentimiento informado explícito para guardar tal registro, sino que se basa en la confianza entre el profesional médico perteneciente a la institución y el paciente.

Para lograr el importante registro antes mencionado es necesario contar con un mecanismo automatizado que facilite su ejecución. Todo el proceso debe realizarse iniciando con un agendamiento de una cita previa a la atención médica y finalizar con el registro de cada atención armando un historial cronológico por paciente para posterior consulta o simplemente analizar la evolución y recuperación de cada uno.

A continuación, se presenta el proceso que debiera tener cualquier Sistema de Fichas médicas como el que este proyecto pretende automatizar.



Figura 1.2: Registro de Fichas Médicas

Fuente: Realización propia

1.3 HERRAMIENTAS TÉCNICAS

Para el desarrollo del sistema se han seleccionado herramientas con ventajas importantes tales como:

- Herramientas multiplataforma
- Documentación abundante del uso de las herramientas en foros y sitios oficiales
- La curva de aprendizaje es relativamente reducida para efectos de levantar los ambientes de desarrollo y producción.
- Las herramientas y los programas o aplicaciones que deriven de ellas tienen cuatro libertades esenciales: (0) ejecutar, (1) estudiar y modificar, (2) redistribuir copias exactas y (3) distribuir versiones modificadas de la aplicación o programa. (Free Software Foundation, 2018)

Partiendo de lo mencionado las herramientas seleccionadas son:

1.3.1 Ubuntu Server:

Debido a que es una de las distribuciones de Linux más utilizadas según (w3techs, 2013) en el mundo de los servidores se seleccionó a Ubuntu como la plataforma sobre la que se ejecuten los servicios necesarios para el presente proyecto.

Ubuntu es un sistema operativo de código abierto, una de las múltiples distribuciones Linux basado en Debian, con un enfoque en la calidad de lanzamiento, actualizaciones de seguridad empresarial y liderazgo en las capacidades clave de la plataforma para la integración, la seguridad y la usabilidad. Los lanzamientos de hitos de Ubuntu se realizan cada seis meses, los lanzamientos de soporte a largo plazo se realizan cada dos años. Canonical garantiza el mantenimiento y el soporte de la empresa durante cinco años con el Mantenimiento de seguridad extendido opcional disponible para los clientes de Canonical. Canonical también proporciona soporte comercial para implementaciones de Ubuntu en el escritorio, el servidor y la nube. (Canonical Ltd., 2019)

1.3.2 PostgreSQL

Se seleccionó este gestor de bases de datos debido a que es un potente gestor relacional y orientado a objetos de código abierto que toma el lenguaje SQL y lo amplía combinado con muchas de sus características que escalan y almacenan las cargas de trabajo de datos más complicadas de forma segura. Puede definir sus propios tipos de datos, construir funciones personalizadas haciendo uso de otros lenguajes de programación sin recompilar su base de datos, motivo por el cual se lo describe como altamente extensible. (The PostgreSQL Global Development Group, 2019)

PostgreSQL tiene poco más de 30 años de activo desarrollo remontando sus orígenes a 1986 como parte del proyecto POSTGRES en la Universidad de California en Berkeley.

PostgreSQL se ha ganado una sólida reputación por su arquitectura probada, confiabilidad, integridad de datos, conjunto de características sólidas, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para ofrecer constantemente soluciones innovadoras y de alto rendimiento. PostgreSQL se ejecuta en todos los sistemas operativos principales, ha sido compatible con ACID desde 2001 y tiene complementos tales como el ya popular extensor de base de datos geoespacial PostGIS.

PostgreSQL viene con muchas características destinadas a ayudar a los desarrolladores a crear aplicaciones, a los administradores a proteger la integridad de los

datos y a crear entornos tolerantes a fallas, y a administrar sus datos sin importar cuán grande o pequeño sea el conjunto de datos.

1.3.3 WildFly

Es un servidor de aplicaciones de código abierto compatible con Java EE. Una vez llamado EJB-OSS y luego JBoss, siempre fue uno de los más utilizados en el mundo. Redhat adquirió JBoss y toda la comunidad comprometida con él en 2005, con efecto de la versión 5. Gracias a una política interna de Redhat, JBoss se convirtió en WildFly en 2012; desde la versión 8, WildFly es la primera implementación de EE 7 de Redhat.

Este servidor de aplicaciones está construido con diferentes arquitecturas. Afortunadamente, WildFly mantiene la misma arquitectura central de JBoss 7, agregando la innovación de Java EE, que es un conjunto de productos de código abierto competitivos y un núcleo muy modular, que simplifica el intercambio de bibliotecas a través de las aplicaciones. Por lo tanto, WildFly puede considerarse como un conjunto de subproductos unidos por el mismo sistema.

Cada uno de estos productos tiene su configuración solo en un archivo XML, por lo que evitamos configuraciones redundantes. Además, se puede utilizar todos los instrumentos independientes proporcionados si se quiere una configuración detallada. El servidor de aplicaciones está diseñado para mantener las aplicaciones. Un servidor de aplicaciones Java debe ser capaz de administrar todas las clases desde las aplicaciones; por ejemplo, debe lidiar con las clases duplicadas, administrando un sistema de carga de clases más sofisticado.

WildFly sigue siendo la mejor solución como repositorio de aplicaciones web. Estas aplicaciones se vuelven más avanzadas año tras año gracias a una actualización continua de las especificaciones empresariales de Java. (Stancapiano, 2017)

1.3.4 Java Enterprise Edition (JEE)

La tecnología JEE (Java Enterprise Edition) constituye una solución propuesta por Sun Microsystems para el desarrollo de aplicaciones distribuidas. La base de la solución se sustenta el lenguaje Java, también creada por Sun. (Groussard, 2010)

El objetivo más importante de la plataforma Java EE 7 es simplificar el desarrollo al proporcionar una base común para los distintos tipos de componentes en la plataforma Java EE. Los desarrolladores se benefician de las mejoras de productividad con más anotaciones y menos configuración XML, más *Plain Old Java Objects (POJO)* y un empaquetado simplificado (Oracle, 2014).

1.3.4.1 Modelo de aplicación Java EE

El modelo de aplicación Java EE comienza con el lenguaje de programación Java y la máquina virtual Java. La probada portabilidad, seguridad y productividad del desarrollador que proporcionan forman la base del modelo de aplicación. El modelo de aplicación Java EE define una arquitectura para implementar servicios como aplicaciones de varios niveles que ofrecen la escalabilidad, accesibilidad y capacidad de administración que necesitan las aplicaciones de nivel empresarial. Este modelo divide el trabajo necesario para implementar un servicio de varios niveles en las siguientes partes (Oracle, 2014):

- La *lógica de negocio* y presentación a ser implementada por el *desarrollador*.
- Los *servicios estándar* del sistema proporcionados por la plataforma *Java EE*.

1.3.4.2 Aplicaciones distribuidas de múltiples niveles

La plataforma Java EE utiliza un modelo de aplicación distribuido de múltiples niveles para aplicaciones empresariales. La lógica de la aplicación se divide en componentes según la función, y los componentes de la aplicación que conforman una aplicación Java EE se instalan en varias máquinas en función del nivel en el entorno Java EE de varios niveles al que pertenece el componente de la aplicación (Oracle, 2014):

- Los componentes de nivel de cliente se ejecutan en la *máquina cliente*.
- Los componentes de nivel web se ejecutan en el *servidor Java EE*.
- Los componentes de nivel empresarial se ejecutan en el *servidor Java EE*.
- El software de *sistema de información empresarial (EIS)* se ejecuta en el *servidor EIS*.

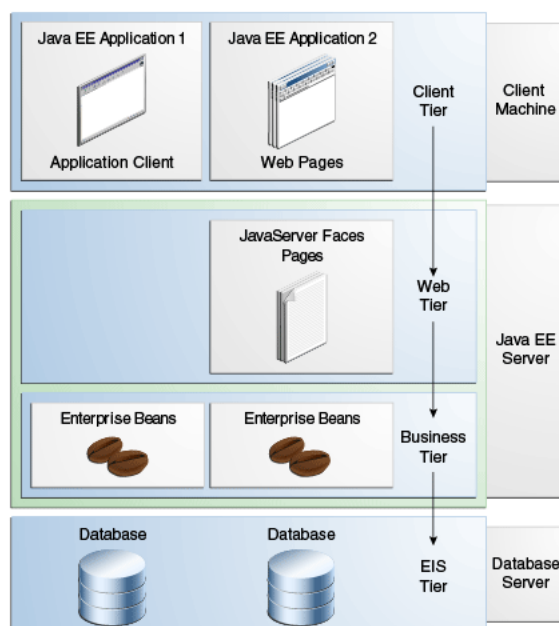


Figura 1.3: Aplicaciones distribuidas de múltiples niveles

Fuente: Oracle

1.3.4.3 Contenedores Java EE

Normalmente, las aplicaciones de múltiples niveles de cliente ligero son difíciles de escribir porque involucran muchas líneas de código complejo para manejar la gestión de transacciones y estados, multihilo, agrupación de recursos y otros detalles complejos de bajo nivel. La arquitectura Java EE basada en componentes e independiente de la plataforma hace que las aplicaciones sean fáciles de escribir porque la lógica empresarial está organizada en componentes reutilizables (Oracle, 2014).

Los contenedores son la interfaz entre un componente y la funcionalidad específica de plataforma de bajo nivel que admite el componente. Antes de que pueda ejecutarse, un componente web, *bean empresarial* o cliente de aplicación debe ensamblarse en un módulo Java EE y desplegarse en su contenedor.

El proceso de ensamblaje implica especificar la configuración del contenedor para cada componente en la aplicación Java EE y para la aplicación Java EE en sí. La configuración del contenedor personaliza el soporte subyacente proporcionado por el servidor Java EE, incluidos servicios como seguridad, gestión de transacciones, nombres de Java y búsquedas en la API de interfaz de directorio (JNDI), y conectividad remota.

El proceso de implementación instala los componentes de la aplicación Java EE en los contenedores de Java EE, como se ilustra a continuación:

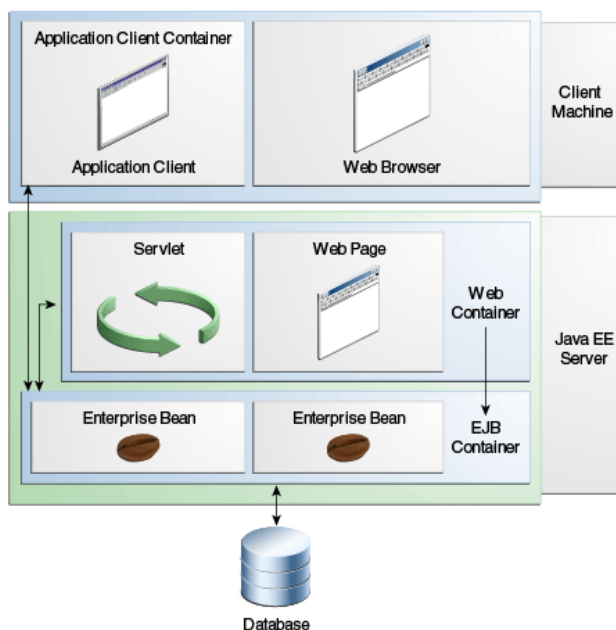


Figura 1.4: Servidor Java EE y sus Contenedores

Fuente: Oracle

Servidor Java EE: espacio de ejecución de un producto Java EE. Un servidor Java EE proporciona EJB y contenedores web.

Contenedor EJB: gestiona la ejecución de *beans empresariales* para aplicaciones Java EE. Los *Beans Empresariales* y su contenedor se ejecutan en el servidor Java EE.

Contenedor Web: administra la ejecución de páginas web, servlets y algunos componentes EJB para aplicaciones Java EE. Los componentes web y su contenedor se ejecutan en el servidor Java EE.

Contenedor del Cliente de Aplicación: gestiona la ejecución de los componentes del cliente de la aplicación. Los clientes de la aplicación y su contenedor se ejecutan en el cliente.

Contener Applet: gestiona la ejecución de applets. Consta de un navegador web y un complemento de Java que se ejecutan juntos en el cliente.

1.3.4.4 API de Java EE 7

Los siguientes puntos proporcionan un breve resumen de las tecnologías requeridas por la plataforma Java EE utilizadas para el presente proyecto:

1.3.4.4.1 Tecnología Enterprise JavaBeans

Un componente *Enterprise JavaBeans (EJB)*, o *enterprise bean*, es un cuerpo de código que tiene campos y métodos para implementar módulos de lógica empresarial (Oracle, 2014).

Los beans empresariales son o bien *beans de sesión* o *beans controlados por mensajes*. Un *bean de sesión* representa una conversación transitoria con un cliente. Cuando el cliente termina de ejecutarse, el *bean de sesión* y sus datos desaparecen. Un *bean controlado por mensajes* combina las características de un *bean de sesión* y un *detector de mensajes*, lo que permite que un componente empresarial reciba mensajes de forma *asíncrona*. Comúnmente, estos son mensajes de *Java Message Service (JMS)* (Oracle, 2014)).

1.3.4.4.2 Tecnología Java Servlet

La tecnología *Java Servlet* permite definir *clases servlet* específicas de HTTP. Una *clase servlet* amplía las capacidades de los servidores que alojan aplicaciones a través de un modelo de programación de *solicitud-respuesta*. Aunque los servlets pueden responder a cualquier tipo de solicitud, se usan comúnmente para extender las aplicaciones alojadas en los servidores web (Oracle, 2014).

1.3.4.4.3 Tecnología JavaServer Faces

La tecnología *JavaServer Faces* es un marco de interfaz de usuario para crear aplicaciones web. Los componentes principales de la tecnología *JavaServer Faces* son los siguientes (Oracle, 2014):

- Un marco de componentes GUI.
- Un modelo flexible para representar componentes en diferentes lenguajes y tecnologías de marcado.
- Un estándar para generar el marcado HTML.

1.3.4.4.4 API de persistencia de Java

La API de persistencia de Java (JPA) es una solución basada en estándares de Java para la persistencia y utiliza un enfoque de mapeo objeto / relacional para cerrar la brecha entre un modelo orientado a objetos y una base de datos relacional. La API de persistencia de Java también se puede utilizar en aplicaciones Java SE fuera del entorno Java EE y consta de las siguientes áreas (Oracle, 2014):

- La API de persistencia de Java
- El lenguaje de consulta
- Objetos/metadatos de mapeo relacional

1.3.4.4.5 API de transacción de Java

Java Transaction API (JTA) proporciona una interfaz estándar para demarcar las transacciones. La arquitectura de Java EE proporciona un compromiso automático predeterminado para gestionar *commits* y *rollbacks* de transacciones. Una confirmación automática significa que cualquier otra aplicación que esté viendo los datos visualizará los datos actualizados después de cada operación de lectura o escritura de la base de datos. Sin embargo, si una aplicación realiza dos operaciones de acceso a la base de datos separadas que dependen una de la otra, API de JTA determinará donde comienza la transacción completa, incluidas las operaciones de retroceso y las confirmaciones (Oracle, 2014).

1.3.4.4.6 Managed Beans

Objetos livianos administrados en contenedores (POJO) con requisitos mínimos, admiten un pequeño conjunto de servicios básicos, como la *inyección de recursos*, *devoluciones de llamada de ciclo de vida* e *interceptores*. Los Managed Beans se pueden utilizar en cualquier parte de una aplicación Java EE, no solo en módulos web (Oracle, 2014).

1.3.4.4.7 Contextos e inyección de dependencias (CDI)

Define un conjunto de servicios contextuales, proporcionados por contenedores de Java EE, que facilitan a los desarrolladores el uso de beans empresariales junto con la tecnología JavaServer Faces en aplicaciones web (Oracle, 2014).

1.3.4.4.8 Inyección de dependencia para Java

La inyección de dependencia para Java define un conjunto estándar de anotaciones (y una interfaz) para su uso en clases inyectables. En la plataforma Java EE, CDI proporciona soporte para la inyección de dependencia. Específicamente, puede usar puntos de inyección solo en una aplicación habilitada para CDI (Oracle, 2014).

1.3.4.4.9 Validación de Bean

La especificación de Validación de Bean define un modelo de metadatos y una API para validar datos en componentes de JavaBeans. En lugar de distribuir la validación de datos en varias capas, como el navegador y el lado del servidor, puede definir las restricciones de validación en un solo lugar y compartirlas en las diferentes capas (Oracle, 2014).

1.3.4.4.10 API de Java Message Service

La API del servicio de mensajes de Java (JMS) es un estándar de mensajería que permite a los componentes de la aplicación Java EE crear, enviar, recibir y leer mensajes. Permite la comunicación distribuida que está débilmente acoplada, es confiable y asíncrona (Oracle, 2014).

1.3.4.4.11 API de JavaMail

Las aplicaciones Java EE utilizan la API de JavaMail para enviar notificaciones por correo electrónico. La API de JavaMail tiene dos partes (Oracle, 2014):

- Una interfaz de nivel de aplicación utilizada por los componentes de la aplicación para enviar correo.
- Una interfaz de proveedor de servicios.

1.3.5 ICEfaces

ICEfaces es un marco de desarrollo de aplicaciones de Internet enriquecidas (RIA) de código abierto para Java EE. ICEfaces funciona en plataformas que van desde computadoras de escritorio a teléfonos inteligentes y desde Apple a Android. Mejora la eficiencia del desarrollador al tiempo que reduce el tiempo de comercialización y los costos operativos (ICESoft Technologies Inc., 2017).

1.3.6 METODOLOGÍA DE DESARROLLO

Scrum es un método ágil centrado en la gestión de proyectos. Por ello, resulta ser un método óptimo para planificar y hacer seguimiento de proyectos en general incluidos los referentes al desarrollo de software (Gutiérrez Plaza & Borillo Doménech, 2012).

Para el desarrollo del presente proyecto se aplicó la metodología de desarrollo ágil *Scrum* por la importante característica de formación de grupos de trabajo con un número limitado de personas y la capacidad de ofrecer en poco tiempo entregables que satisfagan las necesidades del usuario final, mismo que forma parte del grupo y está en constante interacción con el desarrollador del Sistema.

La metodología *Scrum* se resume claramente con los puntos mencionados en la siguiente lista:

- El propietario del producto genera una lista de actividades a realizarse ordenadas por un parámetro de prioridad.
- El equipo de desarrollo toma una parte de la lista de actividades priorizada y las organiza en un *Sprint*, decidiendo cómo se implementarán.

- El equipo de desarrollo planifica un tiempo (no más de 30 días) en el que se compromete a finalizar el conjunto de actividades definidas en el *Sprint*.
- En líder se encarga de que el equipo de desarrollo se mantenga enfocado en la meta.
- Al finalizar el tiempo planificado para un *Sprint* debe haber potencialmente un entregable listo para presentar.
- Antes de empezar con el siguiente *Sprint* se hace un análisis retrospectivo que tiene como objetivo la mejora continua.
- Hasta que la lista de prioridades se considere totalmente cubierta se realizarán las actividades descritas anteriormente.



Figura 1.5: Metodología Scrum

Fuente: www.scrumalliance.org

1.3.6.1 EQUIPO SCRUM

El equipo *SCRUM* se define como un grupo de personas que persiguen una misma meta aportando con sus mejores capacidades de manera responsable en cada iteración.

Los miembros del este equipo tienen roles específicos, los roles determinan el reparto de responsabilidades en un proyecto. Así vemos reflejado el equipo de trabajo (*Scrum team*) que lleva el peso del desarrollo del producto. El negocio, representado por el *Product owner*, que junto al *Scrum master* conforman el organismo de gobierno del proyecto. Y, por último, los usuarios clave (*Stakeholders*), conocedores del negocio y las necesidades de este (Galiano, 2016).

Para el presente proyecto se designó el siguiente equipo con su respectivo rol:

- Ing. Julio Villalba, gerente de la Institución como *Product Owner*

-
- Luis F. Ordóñez A, desarrollador del proyecto como *Scrum Master* y *Development Team*
 - Dr. Jorge Flores, médico oftalmólogo como *Stakeholder*

1.3.6.2 Herramientas de SCRUM

La metodología *SCRUM* requiere de herramientas que aportan a la planificación de las tareas a realizar y el registro de la realización de estas, en el presente proyecto se hará uso de:

- *Product Backlog* que se define como la lista organizada en orden de prioridad de las tareas a realizarse para el desarrollo del producto.
- *Sprint Backlog* que se define como la lista de tareas específicas (Sprint) tomadas del *Product Backlog*.
- *Incremento* que se refiere al conjunto de tareas que se realizan en cada Sprint y puede definirse como una parte del producto final desarrollado.

1.4 ALTERNATIVAS DE SOLUCIÓN

Dentro de las diversas opciones que encontramos dentro del mercado de desarrollo de aplicaciones web, existen varias opciones que los usuarios como centros médicos usan para sus gestiones.

Varias de ellas son online dentro de plataformas virtuales.

A continuación, se nombrará algunos ejemplos y como estas funcionan

1. SISTEMA DE AGENDAMIENTO DE CITAS DE METRORED.

- Las opciones que se tienen son: solicitar citas médicas, registro de citas programadas, y un historial de citas.
- Muestra las opciones a escoger para el horario del agendamiento de citas.
- Al agendar la cita se envía una notificación vía email.
- Se puede consultar el directorio médico.
- Los datos que el usuario puede personalizar son básicos y no permite poner una fotografía personalizada.
- Sitio WEB: <http://metroredapp.hmetro.med.ec/app/>

2. SISTEMA DE AGENDAMIENTO DE CITAS MÉDICAS DE LA USFQ (SIME)

- La página web ofrece pocas opciones.

- La información básica es accesible.
- Se obtiene información básica.
- SITIO WEB: <https://hclinica.simeusfq.com/citaweb/>

A continuación, un cuadro comparativo de las opciones que ofrece la solución propuesta en este trabajo frente a las anteriormente citadas.

Tabla 1.1: Comparativa de Alternativas

Módulo	Metrored	Sime	Salutem
Ingreso	Con credenciales	Con credenciales	Con credenciales
Agendamiento	Sí, de acuerdo con necesidad	Sí, de acuerdo con necesidad	Sí, de acuerdo con necesidad
Reagendamiento de citas	Si posee	No posee	Si posee
Personalización del usuario	Datos básicos sin fotografía	Datos básicos sin fotografía	Datos, descripción y fotografía
Histórico de citas	Si posee	No posee	Si posee
Histórico de atenciones	No posee	No posee	Si posee
Consulta de prescripciones	No posee	No posee	Si posee
Reportes estadísticos	No posee	No posee	Si posee
Avisos al email	Si posee	No posee	Si posee
Directorio de médicos	Si posee	Si posee	Si posee

2 CAPÍTULO II. MARCO METODOLÓGICO

2.1 TIPO DE INVESTIGACIÓN

Para desarrollo del presente proyecto se utilizó el Método Empírico de la investigación científica debido a que por medio de una serie de procedimientos prácticos y técnicas basadas principalmente en la observación se logrará una amplia y objetiva visión de la problemática permitiendo aportar con una solución que se ajuste a las necesidades de la Institución médica.

2.2 RECOPIACIÓN DE INFORMACIÓN

Para la recopilación de información se utilizó fichas de observación que aportaron al levantamiento de datos clave sobre los grupos de usuarios que intervendrán en el Sistema con su respectiva descripción de funciones.

“Una ficha de observación es una forma de organización de información documental. Es usada para realizar el acopio de los contenidos que se encuentre en las fuentes” (Quiroz, 2011).

2.2.1 TÉCNICAS DE RECOPIACIÓN DE INFORMACIÓN

La ficha de observación propuesta para el estudio está conformada por cuatro apartados importantes:

1. Registro de información inicial;
2. Registro transaccional;
3. Extracción de reportes;
4. Observaciones.

Los primeros tres apartados tienen como objetivo identificar la frecuencia con que una actividad o función se realizará por un grupo de usuarios específicos del Sistema y el campo de observaciones está para poder generar anotaciones más concretas que describan con mejor detalle las funciones del grupo de usuarios.

Tabla 2.1: Modelo de Ficha de Observación

Ficha de Observación			
Número.	Observador.		
Fecha. Alta	Actividad.		
Grupo de Usuarios.			
Registro de información inicial		Frecuencia	
	Siempre	Casi siempre	Casi nunca
Registro transaccional		Frecuencia	
	Siempre	Casi siempre	Casi nunca
Extracción de reportes		Frecuencia	
	Siempre	Casi siempre	Casi nunca
Observaciones			

Los grupos de usuarios que se determinó para cada actividad son los siguientes:

- Super administradores
- Administrador Institucional
- Recepcionistas
- Médicos
- Pacientes

A continuación, se mostrará un resumen de las observaciones utilizando fichas cuyo detalle se encuentra en el Anexo 1:

1. La administración de parámetros del Sistema debe ser gestionada por personal debidamente autorizado y con conocimiento del alcance que podría tener una modificación a este nivel porque, aunque la información que se puede manipular no está directamente relacionada al giro de negocio de la institución, claramente tendrá efectos en el buen funcionamiento de todo el sistema.

-
2. La gestión de la Información Institucional debe ser asignada a usuario que tengan profundo conocimiento del giro de negocio de la institución ya que las opciones de configuración que se ofrecen para este grupo de usuarios afectan directamente al funcionamiento específico de esta.
 3. Los recepcionistas se encargarán del ingreso de nuevos pacientes, la gestión de órdenes de laboratorio y el agendamiento de citas médicas dependiendo del horario definido para cada profesional
 4. Los médicos tendrán acceso a la personalización de fichas médicas según su especialidad, registrar las atenciones a pacientes con previa cita o en casos emergentes sin ella y tendrán acceso a reportes de horarios, agendamiento de citas y atenciones realizadas.
 5. El paciente tendrá la opción de agendar citas dependiendo de la agenda disponible para cada médico y tendrá la capacidad de ver las atenciones médicas registradas a su nombre. Es necesario que el paciente esté previamente registrado por el recepcionista.

3 CAPÍTULO III. PROPUESTA

3.1 DIAGRAMA DE PROCESOS

La siguiente Figura muestra el proceso actual no automatizado con el que trabaja la Organización Salud Integral para Todos.

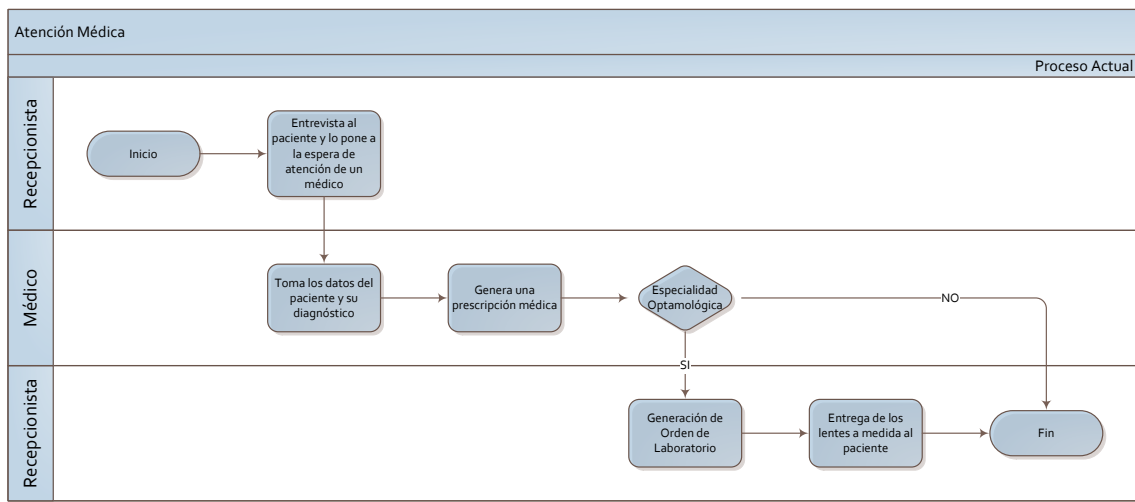


Figura 3.1: Diagrama del Proceso Actual
Fuente: Realización propia

Con el análisis previo se propone un sistema que cumpla con el proceso ilustrado a continuación:

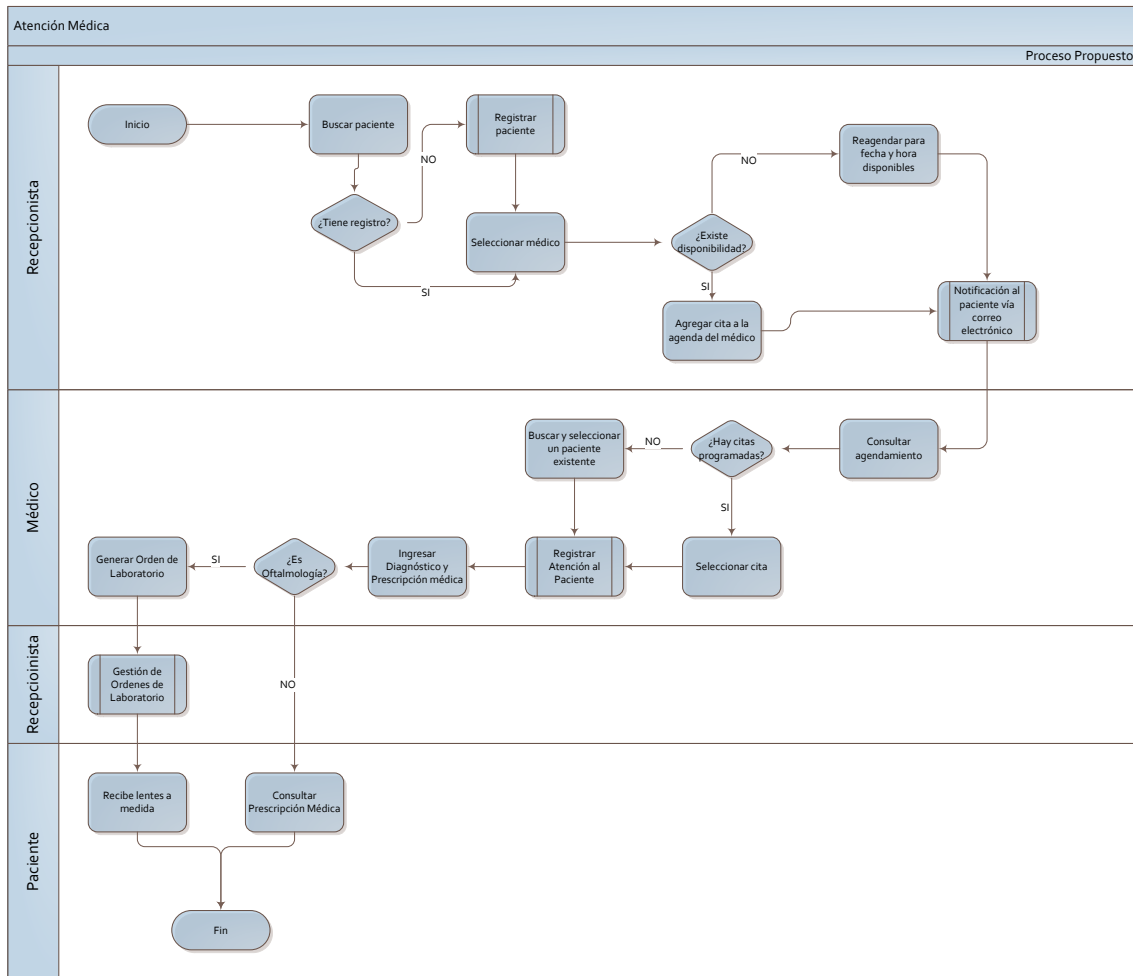


Figura 3-1: Diagrama de Procesos Propuesto

Fuente: Realización propia

3.2 ESPECIFICACIÓN DE REQUERIMIENTOS

3.2.1 ÁMBITO DEL SOFTWARE

El sistema por desarrollar será nombrado con la palabra *salud* en latín: *salutem*. Será un sistema independiente y su base de datos será alimentada con la información de los pacientes que frecuentan la organización *Salud Integral para Todos*. El objetivo principal del sistema será automatizar la gestión de citas médicas y registrar el histórico de pacientes, aportando ventajas, tales como:

- Agilizar el agendamiento y programación de consultas médicas;
- Asegurar la identificación precisa de los pacientes;
- Garantizar niveles adecuados de seguridad, brindado confidencialidad, integridad y disponibilidad de la información del paciente;

-
- Ayudar a los profesionales a optimizar el tiempo de atención a pacientes;
 - Contribuir a la disminución de errores por ilegibilidad en las prescripciones médicas;
 - Aportar en la conservación del medio ambiente debido a que no se usará papel para registrar la ficha médica del paciente.

El Sistema contará con un **módulo de administración** que contempla la gestión de:

- **Códigos** de parametrización, que permitirá agrupar y organizar las variables generales del sistema.
- **Usuarios**, que permitirá el registro, modificación y activación de usuarios del sistema.
- **Grupos y Perfiles** de Usuario, que permitirá agrupar al usuario otorgando los permisos necesarios para el uso de las opciones del sistema mediante la configuración de perfiles.
- **Historial de cambios**, que permitirá el monitoreo cronológico del historial de cambios de cada registro del sistema.

También tendrá un módulo de **Transacciones y Reportes** que gestionará:

- El **registro de profesionales médicos** y su respectivo horario de atención, información que será utilizada para agendar citas médicas.
- El **registro de pacientes y su historial médico** con la finalidad de guardar un histórico de las atenciones de estos, que luego podrá ser consultado usando claves de búsqueda, tales como, cédula de identidad, apellidos, número de ficha médica, etc.
- La **generación de citas médicas** según la disponibilidad del personal médico con el fin de llevar control óptimo del tiempo disponible para la atención médica.
- **Órdenes de Laboratorio**, con la finalidad de hacer un seguimiento al proceso de envío y recepción de lentes con medida a los laboratorios ópticos colaboradores para su posterior entrega a los clientes de la institución.

El sistema no se podrá utilizar para llevar la contabilidad, ni ningún tipo de facturación de la Institución médica, tampoco contempla la gestión de inventario ni la interacción con otros sistemas de información existente.

El sistema validará únicamente el formato de correo electrónico mas no su existencia, debido que no se trata de un sistema de *Email marketing*, en cuyo caso es esencial esa clase de validación. Otro de los factores que impiden que se haga dicha validación es el costo de las soluciones existentes en el mercado, tales como *Verify Email* (<https://verify-email.org/>), costo que la Organización no está en condiciones de asumir.

3.2.2 FUNCIONES DEL PRODUCTO

Las historias de usuario descritas a continuación dan una visión general de las funciones que tendrá el sistema.

Tabla 3.1: Historia 01 - Configuración y parametrización

Historia de Usuario	
Número. 01	Nombre. Configuración y parametrización
Riesgo en desarrollo. Alta	Prioridad en el negocio. Alta
Usuario: Superadministrador	Desarrollador. Luis F. Ordóñez A.
Descripción. El sistema deberá proporcionar las facilidades para crear y modificar:	
<ul style="list-style-type: none"> - Parámetros asociados en grupos maestros, tales grupos permitirán identificar y referenciar: <i>Módulos del Sistema, Grupos de Usuarios, Parámetros Generales</i> y otras listas de referencia de las que el propio sistema hará uso. - Opciones de menú disponibles por cada módulo que éste contemple. - Instituciones dentro de las cuales estará presente la propia Institución médica a la que el presente documento hace referencia. 	
Validaciones. Se verificará que el sistema:	
<ul style="list-style-type: none"> - Valide que cada parámetro tenga un código único que permita su identificación. - Facilite la agrupación de las opciones de menú por módulo. - Permita identificar que una Institución es un Laboratorio de montaje de lentes a medida. 	

Tabla 3.2: Historia 02 – Usuarios y Perfiles

Historia de Usuario	
Número. 02	Nombre. Usuarios y Perfiles
Riesgo en desarrollo. Alta	Prioridad en el negocio. Alta
Usuario: Superadministrador	Desarrollador. Luis F. Ordóñez A.

Descripción. Todo usuario del Sistema debe ser registrado en su base de datos, para ello se solicitará información obligatoria de cada nuevo usuario como:

- Cédula de Identidad o Pasaporte (identificación),
- Apellidos y nombres,
- Correo electrónico, y
- Nombre de usuario

Todo usuario estará adscrito a un grupo previamente registrado al que se asignará un determinado conjunto de opciones de menú a los cuales podrá acceder para crear, actualizar, eliminar o simplemente consultar registros propios de cada opción.

Validaciones. Se verificará que el sistema:

- Impida el registro de identificaciones y nombres de usuario duplicados,
 - Asigne temporalmente como contraseña inicial la identificación encriptada en *SHA-256*.
 - Garantice que cada usuario cambie en el primer ingreso la contraseña inicial
 - Permita que sólo un usuario autorizado sea el único que pueda restablecer contraseñas de otros usuarios.
 - Permita la configuración de un conjunto de opciones de menú específicas a las que cada grupo de usuarios tendrá acceso.
 - Limite las operaciones sobre cada registro que una opción de menú permite ejecutar, las operaciones pueden ser: creación, actualización, eliminación o simple consulta.
-

Tabla 3.3: Historia 03 –Información y parametrización médica

Historia de Usuario

Número. 03	Nombre. Información y parametrización médica
Riesgo en desarrollo. Alta	Prioridad en el negocio. Alta
Usuarios: Administrador	Desarrollador. Luis F. Ordóñez A.
Institucional, Médico, Recepcionista	

Descripción. El Sistema facilitará el registro de:

- Horarios de atención institucional. (Opción para el Administrador Institucional)
- Profesionales médicos y su especialidad. (Opción para el Administrador Institucional)

-
- Horario de atención para cada profesional. (Opción para el Administrador Institucional)
 - Pacientes que frecuenten la institución. (Opción para el Administrador Institucional, Recepcionista y Médico)
 - Formato de fichas médicas por especialidad. (Opción para el Administrador Institucional)

Validaciones. Se verificará que tanto un profesional como un paciente tenga un registro único dentro de la Institución médica y que cada especialidad tenga una ficha configurable que se utilizará para el registro de atención al paciente.

Tabla 3.4: Historia 04 – Citas médicas

Historia de Usuario

Número. 04

Nombre. Citas médicas

Riesgo en desarrollo. Alta

Prioridad en el negocio. Alta

Usuarios: Administrador Institucional, Médico, Paciente, Recepcionista

Desarrollador. Luis F. Ordóñez A.

Descripción. El sistema permitirá agendar citas médicas dependiendo del horario definido para cada profesional y la disponibilidad en función de citas previas en la agenda de este, permitiendo incluso cancelación y reagendamiento.

Validaciones. Se verificará que:

- Una cita no podrá ocupar el mismo espacio de tiempo que otra.
 - No se pueda agendar en horas que la agenda de cada médico no contemple.
 - El paciente no pueda cancelar o reagendar citas que no sean las suyas.
-

Tabla 3.5: Historia 05 – Atención a pacientes

Historia de Usuario

Número. 05

Nombre. Atención a pacientes

Riesgo en desarrollo. Alta

Prioridad en el negocio. Alta

Usuario: Médico

Desarrollador. Luis F. Ordóñez A.

Descripción. Cada médico podrá registrar una atención a pacientes regido por una agenda de citas, sin embargo, en casos emergentes también está facultado para realizar atenciones sin cita previa.

Validaciones. Se verificará que haya no más de una atención por cada cita agendada o emergente de un paciente en un sólo día. Así mismo que una atención no pueda ser modificado pasado el día de su registro.

Tabla 3.6: Historia 06 – Órdenes de laboratorio

Historia de Usuario	
Número. 05	Nombre. Órdenes de laboratorio
Riesgo en desarrollo. Alta	Prioridad en el negocio. Alta
Usuario: Administrador Institucional, Recepcionista	Desarrollador. Luis F. Ordóñez A.
Descripción. El sistema permitirá la gestión de las órdenes de laboratorio generadas si hubo atenciones médicas a pacientes en la especialidad de Optometría, dando opciones para registrar fechas de envío a laboratorios de ensamblaje, recepción y posterior entrega de lentes a sus pacientes.	
Validaciones. Se verificará que cada atención perteneciente a la especialidad de Optometría genere una orden de Laboratorio.	

3.2.3 CARACTERÍSTICAS DE LOS USUARIOS DEL SISTEMA

En la siguiente tabla se muestran las características de los usuarios categorizados por perfiles.

Tabla 3.7: Perfiles de grupos de usuarios

Grupo de Usuarios	Área Funcional	Actividad
Super administradores (root)	Configuración y parametrización del sistema	Gestión de usuarios, grupos de usuarios y perfiles. Administración de parámetros generales: directorio de archivos, formatos de fecha y hora, configuración de la apariencia del sistema. Obtención de reportería: historial de cambios y acceso de usuarios.
Administradores Institucionales (admin)	Gestión de información exclusiva de la Institución.	Gestión de profesionales médicos y pacientes.

		Gestión de horario de atención institucional y de profesionales médicos. Configuración de formularios para fichas médicas. Obtención de reportería: Agendamiento de citas, atenciones a pacientes, ordenes de laboratorio.
Recepcionistas	Atención al cliente.	Registro de nuevos pacientes. Agendamiento de citas médicas Gestión de ordenes de laboratorio.
Médicos	Gestión del historial médico de pacientes.	Registro de atenciones a pacientes. Obtención de reportes: Citas agendadas, Atenciones realizadas. Gestión de información personal básica.
Pacientes	Gestión de datos personales	Agendamiento de citas médicas. Consulta de atenciones, prescripción médica e historial médico en general.

3.2.4 RESTRICCIONES

- Tanto en Servidor de Aplicaciones como el Sistema Gestor de Base de Datos se instalarán en el mismo equipo debido a temas de presupuesto de la organización.
- Al ser un sistema diseñado a la medida de los requerimientos de la Institución su uso se limita únicamente a sus administradores y usuarios registrados no permitiendo así a usuarios anónimos ni sistemas externos.
- Debido al precario sistema de almacenamiento de información física de pacientes, la base de datos se irá alimentando conforme los pacientes requieran atención médica en la Institución.

3.2.5 REQUISITOS

Los requerimientos realizados por la Institución para el desarrollo del sistema están plasmados en las listas de requerimientos funcionales y no funcionales mostrados a continuación.

3.2.5.1 FUNCIONALES

RF01: El sistema contemplará tres módulos:

- Módulo de *Configuración*, que gestionará opciones afines a la seguridad y parametrización del sistema.
- Módulo *Transaccional*, que gestionará opciones de propias del la Institución, tales como registro de médicos, pacientes, horarios de atención, agendamiento de citas, atenciones al paciente y ordenes de laboratorio.
- Módulo del *Paciente*, en donde cada paciente podrá gestionar su información personal y consultar su historial médico.

RF02: Se restringirá el uso del sistema a usuarios debidamente registrados con un nombre de usuario único y su contraseña.

RF03: Cada usuario estará adscrito a un grupo y este a su vez tendrá acceso a un conjunto de opciones específicas dentro de cada módulo del sistema, pudiendo estar en varios grupos a la vez.

RF04: Los usuarios podrán cambiar su contraseña o recuperarla haciendo uso de un asistente virtual o a su vez solicitar que el superadministrador la restablezca.

RF05: Se permitirá subir archivos de imagen para la mejor identificación de profesionales médicos, pacientes e incluso la misma Institución.

RF06: La ficha médica podrá ser configurable por cada especialidad.

RF07: Las citas médicas se podrán agendar tomando en cuenta el horario de atención de cada profesional médico.

RF08: Al agendar, cancelar o reagendar una cita se envíe un correo electrónico al paciente.

RF09: El paciente podrá acceder al Sistema para agendar una cita médica o consultar sus atenciones registradas.

RF10: El sistema guardará el historial de cambios de cada registro, para posterior consulta, facilitando la identificación de cambios no autorizados sobre éstos.

3.2.5.2 NO FUNCIONALES

RNF01: Haciendo referencia a la seguridad las contraseñas de los usuarios serán almacenadas usando el algoritmo *SHA-256* para encriptarlas.

RNF02: El sistema debe ser capaz de ejecutarse en cualquier plataforma ya sea Windows, Linux o MacOS en los principales navegadores existentes: Firefox, Chrome, Safari y Edge.

RNF03: El sistema de garantizar alta disponibilidad y proveer de respaldos para proveer puntos de restauración en caso de fallos.

RNF03: El sistema debe proveer mecanismos de parametrización para evitar al máximo modificaciones en el código fuente.

Tomando en cuenta los requerimientos funcionales y no funcionales antes descritos las tareas que se registrarán en el *Product Backlogs* que es simplemente una lista de las actividades por realizarse dentro del proyecto, misma que resultaría como se muestra en la *Tabla 3-8*. A cada tarea se le asigna una un tiempo en horas y se agrupan dentro de un *Sprint* que es un periodo de no más de 30 días dentro de los cuales el equipo se compromete en terminar un conjunto de ellas.

Tabla 3.8: Product Backlog

Tarea	Prioridad	Complejidad	Sprint	Horas de trabajo
Levantamiento de Procesos	5	5	0	60
Diseño de Base de Datos	5	5	1	20
Diseño de Interfaces de usuario	5	4	1	20
Instalación y configuración de servidor de aplicaciones y de bases de datos	5	4	1	10
Desarrollo del módulo de seguridad	5	5	2	65
Desarrollo de registro de profesionales médicos y horario de atención	5	5	3	65
Desarrollo de registro de pacientes, agendamiento de citas y atención médica	5	5	4	90
Desarrollo de la gestión de órdenes de laboratorio y reportería en general	5	3	5	50
Despliegue del Sistema y pruebas	5	3	6	20
TOTAL				400

4 CAPÍTULO IV. IMPLEMENTACIÓN

4.1 DISEÑO GENERAL

En este trabajo se utilizó la metodología **Scrum** aplicada a proyectos de software, por lo que a continuación, se describen las tareas y entregables en cada *Sprint*.

Tabla 4.1: Sprint Backlog – Sprint 0

Sprint 0		
Entregables: Fichas de observación, historias de usuario, requerimientos		
Tarea	Responsable	Horas de Trabajo
Levantamiento de procesos	Desarrollador	30
Generación de las fichas de observación	Desarrollador	10
Generación de historias de usuario	Desarrollador	10
Generación de requerimientos funcionales y no funcionales	Desarrollador	10

Tabla 4.2: Sprint Backlog – Sprint 1

Sprint 1		
Entregables: Esquema de Base de datos, diseño de interfaces de usuario, instalación y configuración de servidor de aplicaciones y bases de datos.		
Tareas	Responsable	Horas de Trabajo
Definición y creación de tablas que involucran seguridad, parametrización y registro de cambios del sistema:	Desarrollador	10
- Usuarios		
- Perfiles		
- Menús		
- Maestros		
- Parámetros		
- Historial		

Definición y creación de tablas que involucran la transaccionalidad del sistema: Instituciones Profesionales Personas Horarios Citas Atenciones Órdenes	Desarrollador	10
Diseño de las interfaces correspondientes a la configuración y parametrización del sistema	Desarrollador	5
Diseño de las interfaces correspondientes parte transaccional del sistema	Desarrollador	5
Diseño de las interfaces correspondientes a la configuración y parametrización del sistema	Desarrollador	5
Diseño de las interfaces correspondientes parte transaccional del sistema	Desarrollador	5
Instalación del servidor de aplicaciones	Desarrollador	5
Instalación del servidor de bases de datos	Desarrollador	5

Tabla 4.3: Sprint Backlog – Sprint 2

Sprint 2		
Entregables: Módulo de seguridad y parametrización		
Tareas	Responsable	Horas de Trabajo
Desarrollo de las funciones CRUD para las tablas de: Maestros Parámetros Usuarios	Desarrollador	25
Desarrollo de las funciones CRUD para las tablas de:	Desarrollador	20

Menús		
Submenús		
Perfiles		
Desarrollo de las funciones CRUD para la tabla de registro de cambios del Sistema:	Desarrollador	20
Historial		

Tabla 4.4: Sprint Backlog – Sprint 3**Sprint 3****Entregables:** Registro de Profesionales y horarios de atención

Tareas	Responsable	Horas de Trabajo
Desarrollo de la opción para registrar el horario de atención institucional	Desarrollador	25
Desarrollo de la opción para registrar Profesionales médicos por institución	Desarrollador	20
Desarrollo de la opción para registrar horarios de atención por profesional médico.	Desarrollador	20

Tabla 4.5: Sprint Backlog – Sprint 4**Sprint 4****Entregables:** Registro de Pacientes, agendamiento de Citas y Atención Médica

Tareas	Responsable	Horas de Trabajo
Desarrollo de la opción para el registro de nuevos Pacientes y el mantenimiento de los existentes	Desarrollador	20
Desarrollo de la opción para realizar el agendamiento médico	Desarrollador	25
Desarrollo de la opción para personalizar fichas médicas por especialidad	Desarrollador	20
Desarrollo de la opción para registrar las Atenciones a pacientes	Desarrollador	25

Tabla 4.6: Sprint Backlog – Sprint 5**Sprint 5****Entregables:** Gestión de órdenes de laboratorio

Tareas	Responsable	Horas de Trabajo
Desarrollo de la opción para el envío de órdenes a Instituciones catalogadas como laboratorios	Desarrollador	10
Desarrollo de la opción para la recepción de órdenes por parte de los laboratorios	Desarrollador	10
Desarrollo de la opción para la entrega de lentes a medida a los pacientes y finalización de la orden de laboratorio.	Desarrollador	10
Creación de Gráficos de resumen de citas y atenciones médicas	Desarrollador	10
Creación de Gráficos de resumen ingresos al Sistema y Usuario por Grupos	Desarrollador	10

4.2 ESQUEMA DE LA BASE DE DATOS

Para el desarrollo del presente proyecto se utilizarán dos esquemas bases de datos distintas: una que contemple los componentes de *seguridad y transacciones* del sistema; y otra base que almacenará todo el *registro de cambios* de cada campo en la base anterior.

Es importante mencionar que la tabla *historial* que está en la base de datos que se le denominará *salutemlogs* (**Figura 4.1**) tendrá un crecimiento considerable debido a que ahí se guardará cada operación de creación, modificación y borrado por cada tabla y campo de la base de datos transaccional que se denominará *salutem* (**Figura 4.2**), adicionalmente los ingresos o intentos de ingresos al sistema también serán registrados en la misma estructura que ofrece la tabla *historial*.

El diseño antes descrito tiene como objetivo agilizar las tareas de respaldo y restauración de la base de datos transaccional en donde realmente se guarda la información relevante, esto permitirá que el tiempo de recuperación en caso de fallos sea el mínimo necesario y no afecte la continuidad del negocio.

4.3 DIAGRAMA DE LA ARQUITECTURA DEL SISTEMA

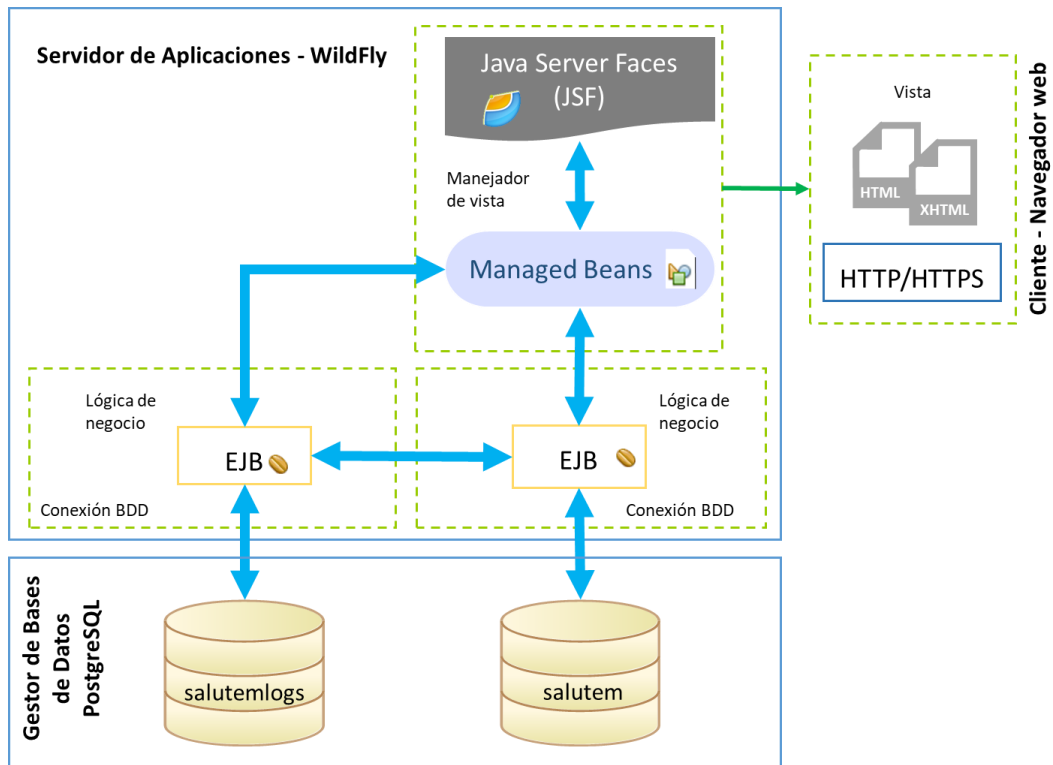


Figura 4.3: Diagrama de Arquitectura del Sistema
Fuente: Realización propia

4.4 DISEÑO DE INTERFACES

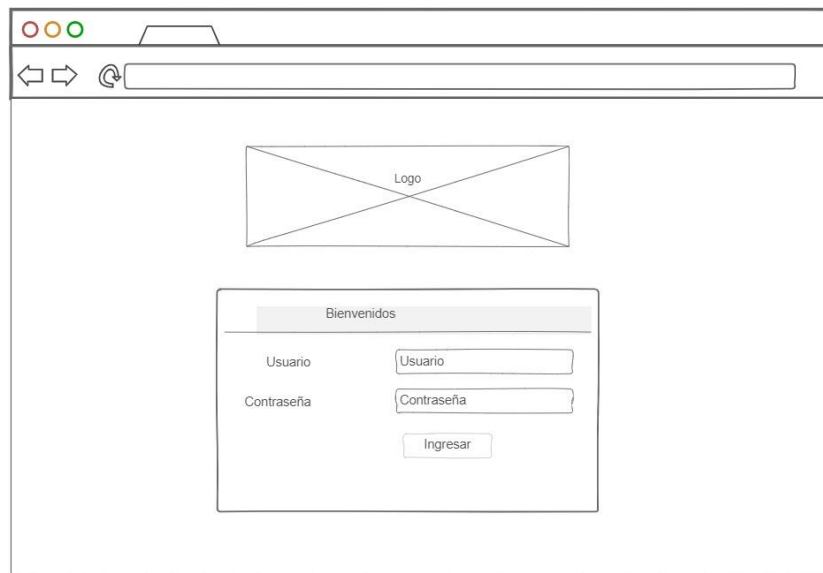


Figura 4.4: Interfaz de Ingreso al Sistema
Fuente: Realización propia

La Interfaz de Ingreso tendrá a la vista un logotipo sobre el formulario donde se solicitarán las credenciales previo a la autenticación de los usuarios.

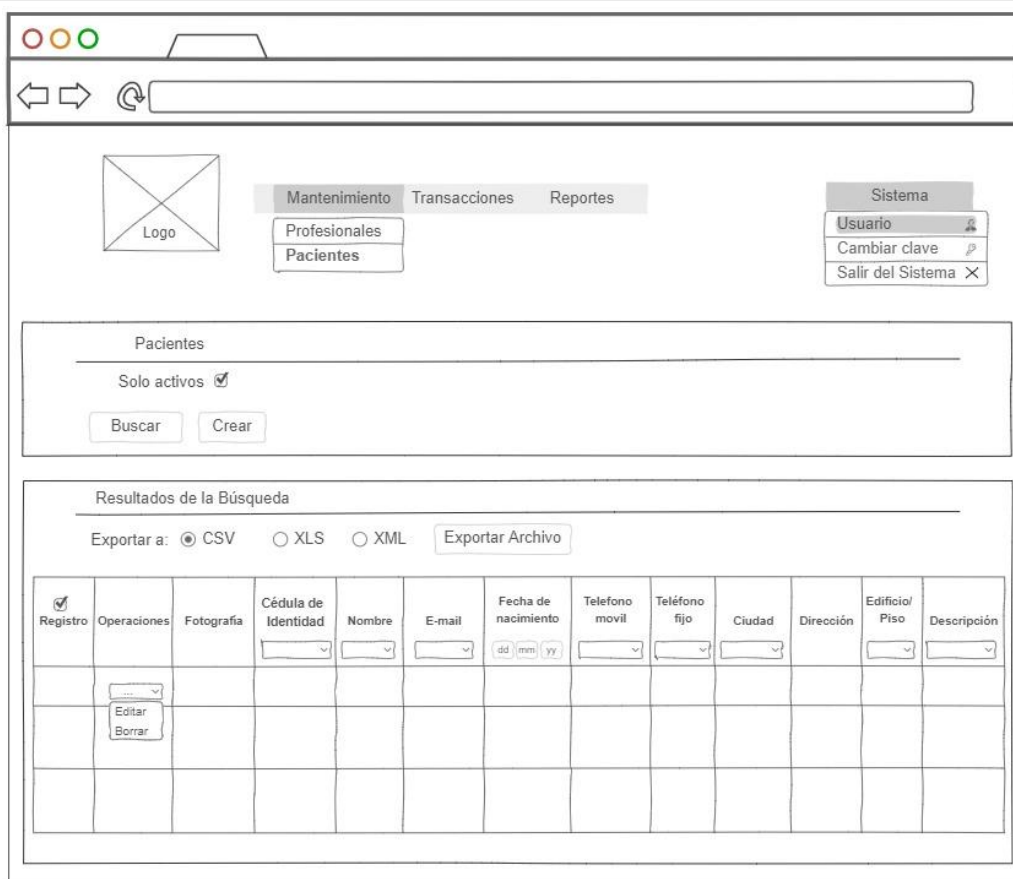


Figura 4.5: Interfaz de registro de pacientes y profesionales médicos
Fuente: Realización propia

La Interfaz de Registro de Pacientes y Profesionales médicos serán similares en apariencia y estarán dentro del marco definido de una plantilla maestra que regirá todo el sistema donde siempre serán visibles: el logotipo del sistema o la institución a la que pertenece el usuario que haya ingresado, una barra de opciones de menú; y un botón de acciones donde se podrá ver el nombre del usuario y las opciones de cambiar clave y salir del sistema.

The screenshot shows a web browser window with a navigation menu at the top. The menu includes 'Mantenimiento', 'Transacciones', 'Reportes', and 'Sistema'. Under 'Sistema', there are options for 'Usuario', 'Cambiar clave', and 'Salir del Sistema'. The 'Citas' option is selected under 'Transacciones'. Below the menu is a search form titled 'Citas' with a checkbox for 'Solo activos' and fields for 'Paciente', 'Fecha', 'Especialidad', 'Profesional', and 'Hora'. There are 'Grabar' and 'Buscar' buttons. Below the search form is a section titled 'Resultados de la Búsqueda' with radio buttons for 'Exportar a: CSV', 'XLS', and 'XML', and an 'Exportar Archivo' button. Below this is a table with columns: 'Registro', 'Operaciones', 'Fecha', 'Paciente', 'Profesional', and 'Observaciones'. The table has a dropdown menu for 'Registro' and 'Operaciones' with 'Editar' and 'Borrar' options.

Figura 4.6: Interfaz de Agendamiento de Citas médicas

Fuente: Realización propia

La Interfaz de Agendamiento de Citas médicas tendrá un buscador de pacientes y opciones de búsqueda de horarios disponibles dentro de la agenda de cada médico para poder generar nuevas citas.

The image shows a web browser window with a navigation bar at the top containing back, forward, and refresh icons. The main content area is a form titled "Atenciones". The form is divided into two main sections: "Ficha médica" and "Datos".

Ficha médica

- Peso:
- Talla:
- Temperatura:
- Presión:

Datos

- Nombres:
- Apellidos:
- CI:
- Dirección:
- Teléfono:
- Ocupación:
- Edad:
- Género:

Below these sections, there are two more input fields:

- Motivo:
- Indicaciones:

At the bottom of the form, there are two buttons: "Guardar" and "Cancelar".

Figura 4.7: Interfaz de Registro de Atención médica

Fuente: Realización propia

La Interfaz de Registro de Atención médica tendrá una sección para datos informativos de cada paciente y otra para el registro de datos que podrán ser diferentes por cada atención médica sujeta a una especialidad que contemple el sistema, además de observaciones generales que el médico podrá registrar.

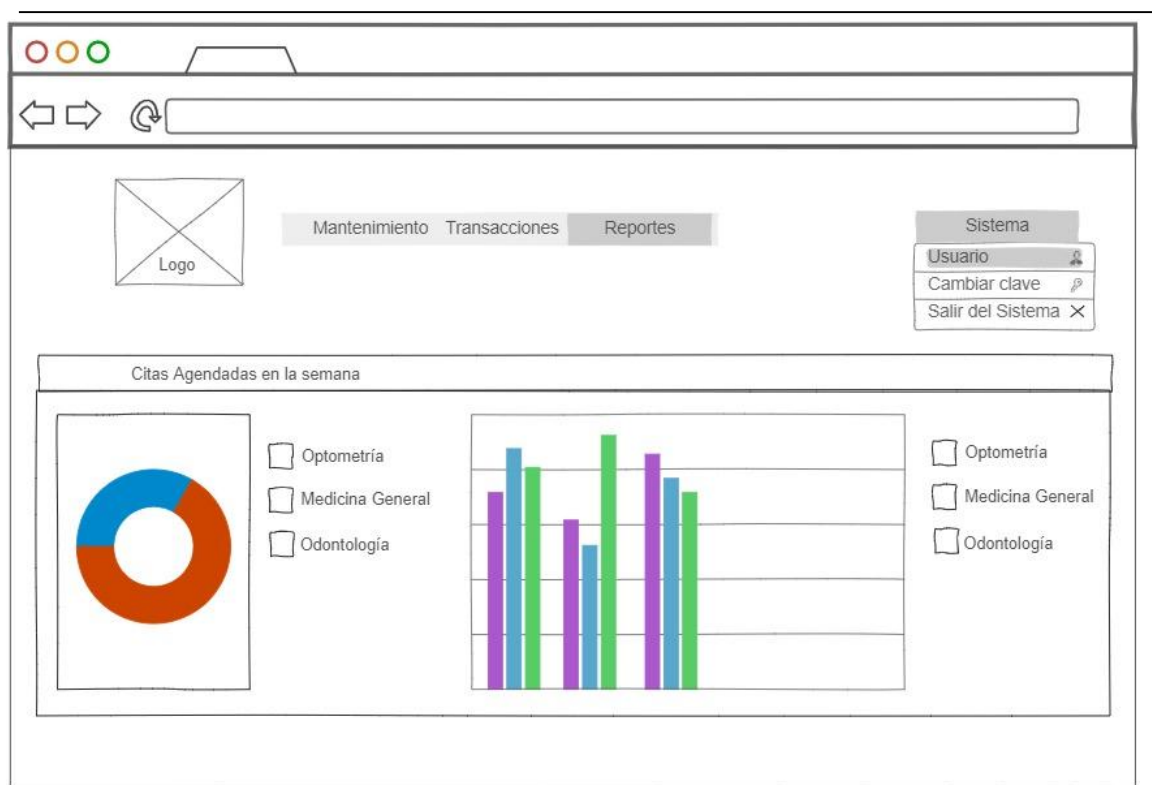


Figura 4.8: Interfaz de Reportes Gráficos

Fuente: Realización propia

La interfaz de Reportes gráficos se mostrará como primera pantalla luego del ingreso al sistema en esta interfaz se mostrarán datos estadísticos a nivel general dependiendo del módulo accedido, por ejemplo, si se entra al módulo de Seguridad, los datos mostrados harán referencia a la frecuencia de ingreso de los usuarios al Sistema, por otro lado, si se entra al módulo transaccional se mostrarán datos referentes a las citas y atención generada a pacientes.

4.5 ESTÁNDARES DE PROGRAMACIÓN UTILIZADOS

El lenguaje de programación utilizado es JAVA, por lo tanto, los estándares de codificación estarán de acuerdo con Convenciones de Código para el lenguaje de programación JAVA (Sun Microsystems Inc., 1999).

Una de las prácticas recomendadas y utilizadas como estándar para la escritura de identificadores en código JAVA es *CamelCase*, se llama así porque cada palabra en el identificador se junta sin espacios, pero con la primera letra de Cada palabra aparece en mayúsculas, con aspecto de joroba de camello (Sanauilla, 2008).

Existen dos variedades en *CamelCase*: *UpperCamelCase*, que Inicia la primera letra con mayúscula y *lowerCamelCase*, que Inicia la primera letra con minúscula; estas dos variedades serán usadas en ciertos casos descritos a continuación.

-
- Los **paquetes** se nombrarán siempre con letras minúsculas y sin caracteres especiales, asegurándose que el prefijo del paquete corresponda a un nombre de dominio de primer nivel, tal como: ec, com, org, etc.

Ejemplos:

```
org.salutem.entidades
org.salutem.controladores
org.salutem.excepciones
```

- Las **clases** se nombrarán siempre con un sustantivo, iniciando con una letra mayúscula, en el caso de palabras compuesta cada palabra que componga el nombre deberá comenzar en mayúscula (*lowerCamelCase*). Para nombrar un **interfaz** se utilizará el mismo criterio que para las **clases**, pero por convenio se utilizará el prefijo “I”.

Ejemplos:

```
class Profesionales
class Pacientes
class IMantenimientos
```

- Los **métodos** se nombrarán con verbos y usando minúsculas a menos que sean palabras compuestas en tal caso cada palabra componente empezado por la segunda iniciará en mayúscula (*lowerCamelCase*).

Ejemplos:

```
public void iniciar();
public String grabar();
public String insertarProfesional();
```

- Las **variables** se escribirán con minúsculas a menos que sean palabras compuestas en tal caso cada palabra componente empezado por la segunda iniciará en mayúscula (*lowerCamelCase*).

Ejemplos:

```
Pacientes paciente;
String cedulaPaciente;
Boolean activo;
```

- Las **constantes** se escribirán con mayúsculas y si son nombres compuestos estos se separarán usando un guión bajo:

Ejemplos:

```
int LONGITUD_MINIMA;
int LONGITUD_MINIMA;
```

En cuanto a la nomenclatura a usarse para nombrar relaciones y campos en la base de datos se tomarán en cuentas las siguientes recomendaciones:

- Para nombrar una tabla se utilizarán minúsculas sin ningún carácter especial y denotarán pluralidad.

Ejemplos:

```
personas
atenciones
citas
```

- Los campos de cada tabla también serán escritos en minúsculas y sin caracteres especiales.
- Toda tabla tener un identificador único con el nombre “id”, campo de tipo entero autoincremental y será usado como clave primaria (primary key).
- Las restricciones que hacen referencia a llaves primarias se nombrarán usando el nombre de la tabla, nombre del campo y el sufijo “pkey”, separados por un guión bajo.

Ejemplos:

```
pacientes_id_pkey
atenciones_id_pkey
```

- Si un campo hace referencia a otra tabla (foreign key) este utilizará el nombre de la tabla referenciada en singular.
- Las restricciones que hacen referencia a llaves foráneas se nombrarán usando el nombre de la tabla, nombre del campo y el sufijo “fkey”, separados por un guión bajo.

Ejemplos:

```
citas_paciente_fkey
atenciones_paciente_fkey
```

A continuación, se presenta un script con la nomenclatura estándar para la creación de una tabla en postgresql:

```
CREATE TABLE pacientes
(
  id serial,
  persona integer,
  institucion integer,
  activo boolean,
  creado timestamp without time zone,
  creadopor text,
```

```

actualizado timestamp without time zone,
actualizado por text,
CONSTRAINT pacientes_pkey PRIMARY KEY (id),
CONSTRAINT pacientes_institucion_fkey FOREIGN KEY (institucion)
REFERENCES instituciones (id),
CONSTRAINT pacientes_persona_fkey FOREIGN KEY (persona)
REFERENCES personas (id)
);

```

4.6 PRUEBAS

Para la realización de las pruebas de funcionalidad se tomaron como base todas las historias de usuario descritas anteriormente debido a que ahí se encuentran especificadas las validaciones del sistema a realizarse.

Para realizar las pruebas de rendimiento del sistema se utilizará la herramienta JMeter debido a que está orientado a este tipo de pruebas.

4.6.1 PRUEBAS DE FUNCIONALIDAD

El objetivo de las pruebas de funcionalidad es validar que los requerimientos realizados por el usuario se cumplan, a continuación, se muestra una lista de verificación de todas las funcionalidades que contempla el sistema junto a las observaciones realizadas. Todas las pruebas de funcionalidad y validación de requerimiento se encuentran avaladas por las firmas del representante de la Institución, Ing. Julio Villalba, y su colaborador, Dr. Jorge Flores. (Ver Anexo 2)

Tabla 4.7: Pruebas de Funcionalidad

Actividad	Validación
Creación de parámetros asociados en grupos maestros y opciones de menú del sistema.	<p>Se verifica que:</p> <ul style="list-style-type: none"> - cada parámetro tiene un código único y no permite duplicados. - las opciones de menú se agrupan por módulos. - se pueden crear Instituciones e identificarlas como Laboratorios.
Creación de usuarios y perfiles.	<p>Se verifica que el sistema:</p> <ul style="list-style-type: none"> - impide el registro de identificaciones y nombres de usuario duplicados. - asigna temporalmente como contraseña inicial la identificación encriptada en <i>SHA-256</i>. - solicita el cambio de contraseña al primer inicio de

<p>Creación de horarios de atención Institucional, profesionales médicos, horarios de atención por profesional, pacientes y formato de fichas médicas.</p>	<p>sesión del nuevo usuario.</p> <ul style="list-style-type: none"> - permite el ingreso de usuarios debidamente registrados. - permite la configuración de un conjunto de opciones de menú específicas a las que un grupo de usuarios tendrá acceso. - limita las operaciones de creación, actualización, eliminación o simple consulta sobre cada registro que una opción de menú permite. <p>Se verifica que tanto un profesional como un paciente tenga un registro único dentro de la Institución médica y que cada especialidad tiene una ficha configurable que se utiliza para el registro de atención al paciente.</p>
<p>Agendamiento, reagendamiento y cancelación de citas médicas regidos por el horario de cada profesional.</p>	<p>Se verifica que:</p> <ul style="list-style-type: none"> - una cita no puede ocupar el mismo espacio de tiempo que otra. - no se puede agendar en horas que la agenda de cada médico no contemple. - el paciente no puede cancelar o reagendar citas que no sean las suyas. - se notifica mediante correo electrónico cuando se agenda o reagenda una cita.
<p>Registro de Atención a Pacientes.</p>	<p>Se verificará que no se puede registrar más de una atención por cada cita agendada y una atención no pueda ser modificada pasado el día de su registro.</p>
<p>Gestión de Órdenes de Laboratorio.</p>	<p>Se verifica que cada atención perteneciente a la especialidad de Optometría genera una orden de Laboratorio, misma que pasa por un proceso de envío a un Laboratorio registrado, recepción del paquete enviado y finalmente entrega al paciente.</p>

4.6.2 PRUEBAS DE ESTRÉS, CARGA Y RENDIMIENTO

4.6.2.1 Objetivo

Realizar pruebas HTTP REQUEST para determinar el rendimiento del sistema *salutem* utilizando la aplicación JMeter.

4.6.2.2 Objetivos Específicos

- Presentar reportes en gráficos.
- Presentar el resumen del plan de pruebas.

4.6.2.3 Procedimiento

Se crearán 4 escenario de pruebas con acceso simultáneo al sistema de 50, 100, 1000 y 2000 usuarios respectivamente, cada escenario será descrito de forma secuencial a continuación.

- Creación de Plan de pruebas

Un plan de prueba describe una serie de pasos que JMeter ejecutará tomando en cuenta uno o más grupos de subprocesos, controladores lógicos, generadores de muestras, temporizadores y elementos de configuración (The Apache Software Foundation, 2018).

- Grupo de Hilos

El grupo de hilos es el punto inicial de cualquier plan de pruebas y debe ir bajo el Plan de Pruebas. Para crear un nuevo grupo de Hilos se hace clic derecho sobre el plan de pruebas como se muestra a continuación:

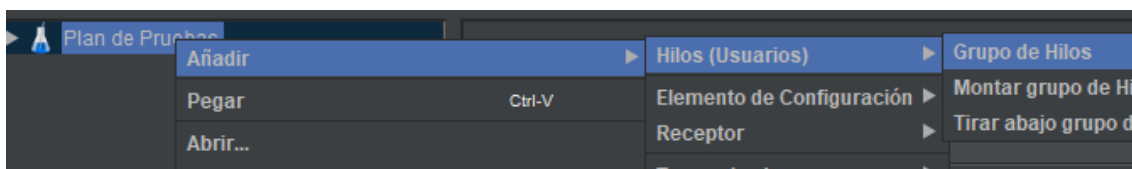


Figura 4.9: Añadir Grupo de Hilos

Fuente: Aplicación JMeter

- Hilos (Usuarios o conexiones simultaneas)

Agregar un elemento de grupo de subproceso, el grupo de subprocesos le indica a JMeter el número de usuarios que se desea simular.

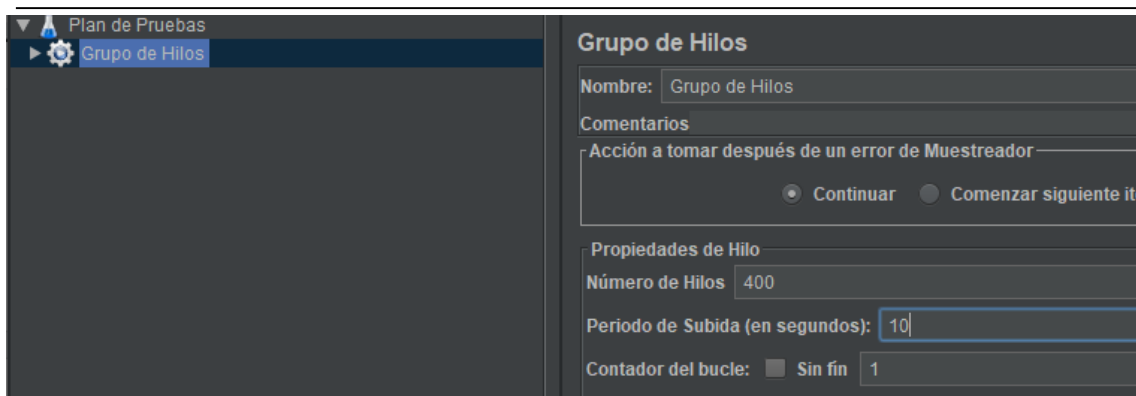


Figura 4.10: Personalizar grupos
Fuente: Aplicación JMeter

Aquí se modifican las propiedades predeterminadas:

Número de hilos: Indica la cantidad de usuarios que acceden al sistema dentro del plan de pruebas.

Periodos de subida: Indica cada cuanto tiempo tardará cada usuario en acceder al sistema dentro del plan de pruebas.

- **Agregar la propiedad de Petición HTTP**

Este elemento permite establecer valores predeterminados que se usan para las solicitudes HTTP

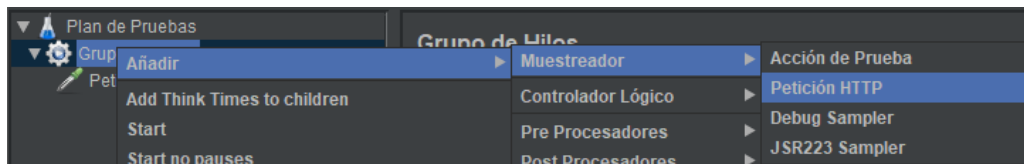


Figura 4.11: Añadir petición HTTP
Fuente: Aplicación JMeter

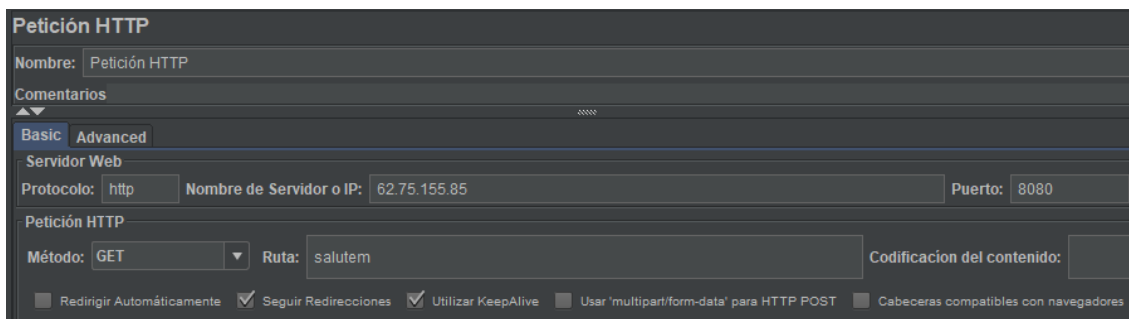


Figura 4.12: Configurar petición HTTP
Fuente: Aplicación JMeter

Aquí se digita el protocolo, la IP, el puerto del servidor y la ruta de inicio del sistema objeto de las pruebas.

- **Agregar un receptor de resultados**

El elemento final dentro del plan de pruebas es un Receptor, este elemento es el responsable, de almacenar todos los resultados de las peticiones HTTP en un archivo para luego mostrarlos en pantalla.

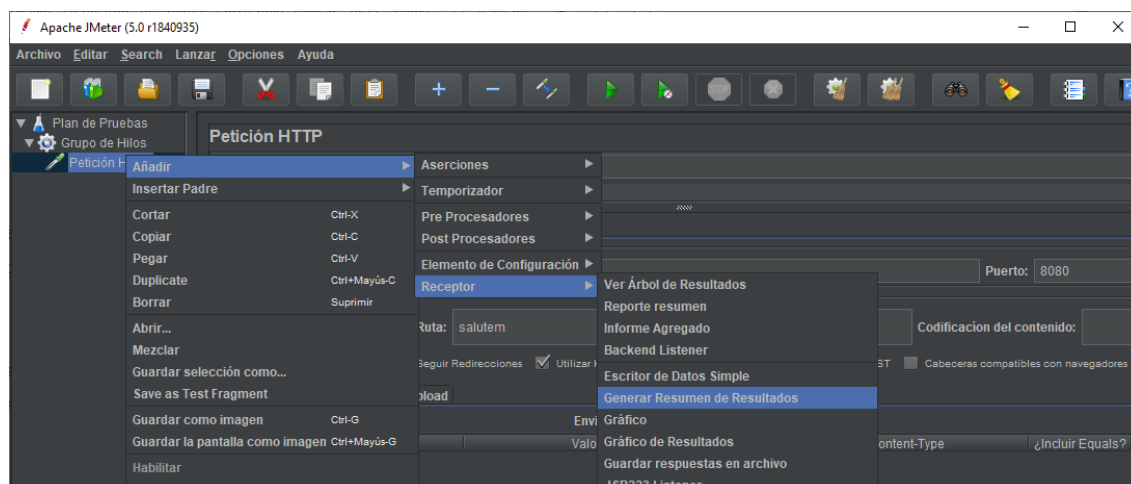


Figura 4.13: Añadir Resumen de Resultados
Fuente: Aplicación JMeter

- **Resumen de Resultados (Receptor)**

El Resumen de Resultados crea un registro para cada solicitud. JMeter tiene en cuenta el tiempo total durante el cual se han generado las solicitudes HTTP.

Reporte resumen											
Nombre: Reporte resumen											
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo		Navegar...		Log/Mostrar sólo:		Escribir en Log		Sólo Errores		Éxitos	
Configurar											
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Está...	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de B...	
Petición HTTP	150	795	555	3949	386,17	0,00%	36,9/min	5,68	0,15	9459,3	
Total	150	795	555	3949	386,17	0,00%	36,9/min	5,68	0,15	9459,3	

Figura 4.14: Reporte de Resumen
Fuente: Aplicación JMeter

El resumen muestra los siguientes resultados:

- **Muestras:** Número de solicitudes enviadas.
- **Media:** Una media aritmética de los resultados.
- **Mín.:** Tiempo de respuesta mínimo en milisegundos.
- **Máx.:** Tiempo máximo de respuesta en milisegundos.
- **Desv. Estándar:** Desviación Estándar.

- **% Error:** Porcentaje de pruebas fallidas.
 - **Rendimiento:** Cuantas peticiones por minuto ha gestionado el servidor.
 - **Kb/sec:** Velocidad de descarga.
 - **Sent Kb/sec:** Velocidad de carga.
 - **Media de bytes:** Tamaño de respuesta promedio.
- **Gráfico de Resultados (Receptor)**

El Gráfico de Resultados genera un gráfico simple que representa todos los tiempos de muestreo. En la parte inferior del gráfico se detalla la información de las muestras y su tiempo de respuesta en milisegundos.

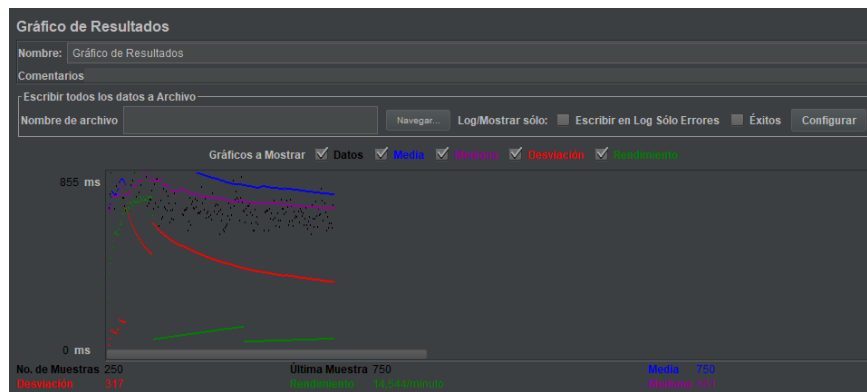


Figura 4.15: Gráfica de Resultados
Fuente: Aplicación JMeter

- **Árbol de Resultados (Receptor)**

El Árbol de Resultados despliega en forma de árbol de todas las respuestas de la muestra, aquí permite ver en detalle datos de interés como el tiempo que tomó obtener una respuesta y si esta fue favorable o no.

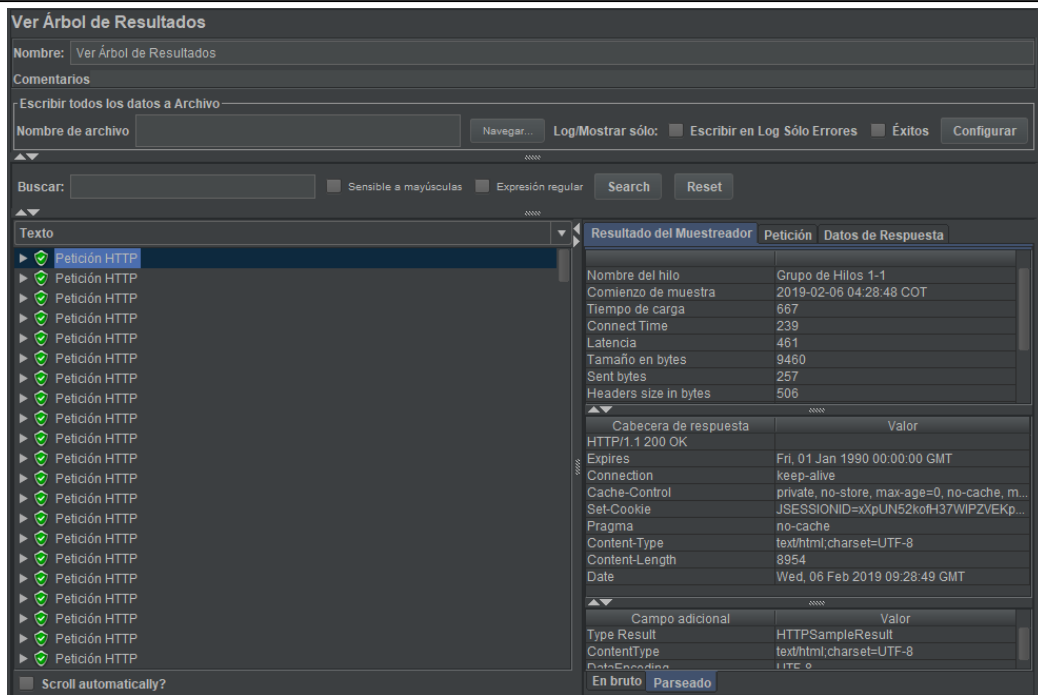


Figura 4.16: Árbol de Resultados
Fuente: Aplicación JMeter

4.6.2.4 Resultado de las pruebas

- 50 usuarios concurrentes

Reporte resumen

Nombre: 50 Usuarios Concurrentes

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos Configurar

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
Petición HTTP	50	654	590	1181	85,10	0,00%	4,8/sec	44,24	1,20	9459,5
Total	50	654	590	1181	85,10	0,00%	4,8/sec	44,24	1,20	9459,5

Figura 4.17: Resumen primer escenario
Fuente: Aplicación JMeter

- 100 usuarios concurrentes

Reporte resumen

Nombre: 100 Usuarios Concurrentes

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos Configurar

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
Petición HTTP	100	618	535	803	49,74	0,00%	9,5/sec	87,99	2,39	9459,0
Total	100	618	535	803	49,74	0,00%	9,5/sec	87,99	2,39	9459,0

Figura 4.18: Resumen segundo escenario
Fuente: Aplicación JMeter

- **1000 usuarios concurrentes**

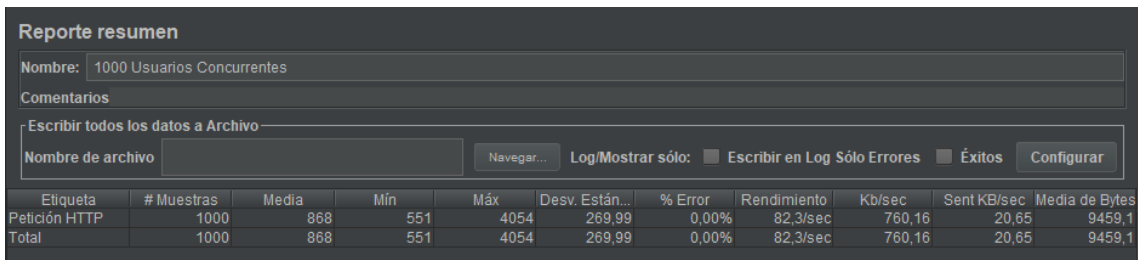


Figura 4.19: Resumen tercer escenario
Fuente: Aplicación JMeter

- **2000 usuarios concurrentes**

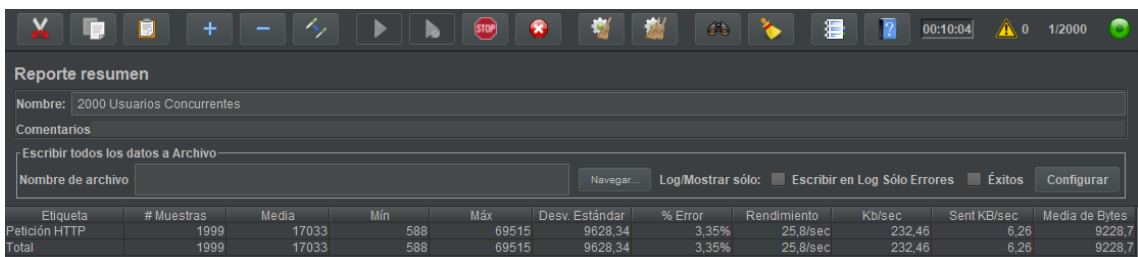


Figura 4.20: Resumen cuarto escenario
Fuente: Aplicación JMeter

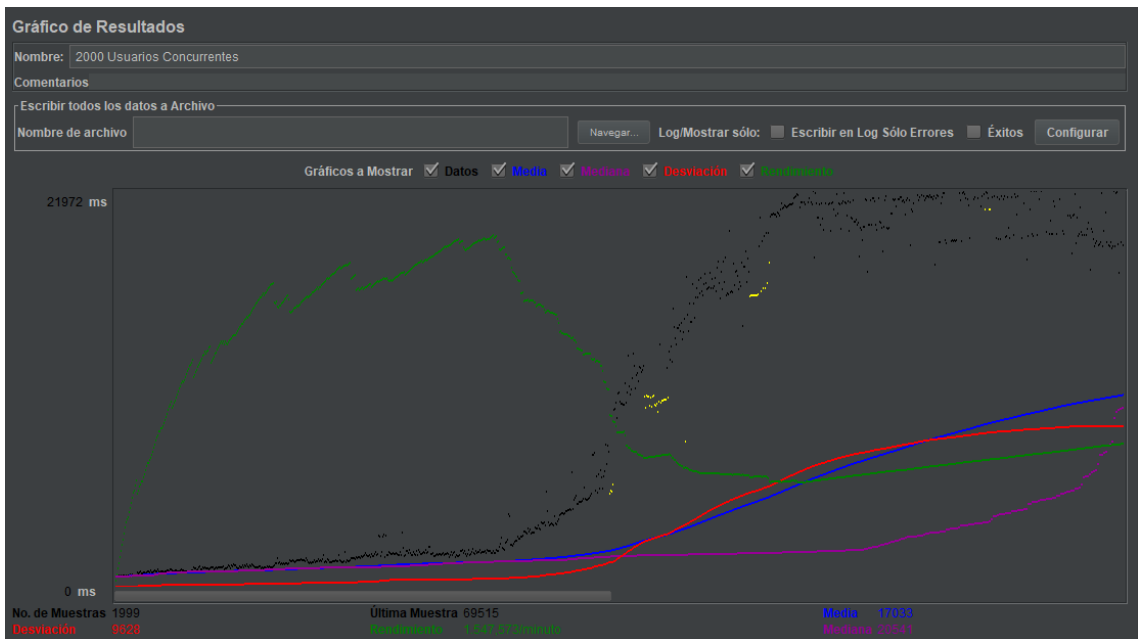


Figura 4.21: Gráfico de resultados cuarto escenario
Fuente: Aplicación JMeter

Conclusiones de las pruebas realizadas:

1. Los primeros escenarios de prueba de 50 y 100 usuarios concurrentes entregan resultados aceptables en cuanto a rendimiento logrando gestionar de 4 a 9 peticiones por segundo obteniendo el 100% de éxito.

-
2. En el escenario de pruebas que contempla 1000 usuarios concurrentes, se evidencia que el rendimiento sube considerablemente debido a que logra gestionar 82 peticiones por segundo obteniendo el 100% de éxito.
 3. En el último escenario donde 2000 usuarios concurrentes hacen una petición al servidor se puede evidenciar que no todas las peticiones se logran gestionar habiendo un porcentaje de error de aproximadamente 3% y el rendimiento sufre una baja considerable en relación con los demás escenarios siendo este de 25 peticiones despachadas por segundo.
 4. Las **pruebas de carga** tienen como objetivo validar la respuesta de la aplicación sometida a un número de usuarios o transacciones esperados en el ambiente de producción, este tipo de pruebas está evidenciado en los dos primeros escenarios con 50 y 100 peticiones concurrentes.
 5. Las **pruebas de rendimiento** son realizadas para medir la respuesta de la aplicación ante un número de usuarios o transacciones al que se prevé podría llegar a tener el sistema en casos no ordinarios, los resultados de esta prueba se evidencian con el tercer escenario de pruebas con 1000 peticiones concurrentes.
 6. Las **pruebas de estrés** nos permiten establecer un número de usuarios o transacciones simultáneas con el que la aplicación deja de responder o empieza a fallar, esto podemos evidenciar con el cuarto escenario de pruebas con 2000 peticiones concurrentes.

4.7 IMPLEMENTACIÓN

Aquí se explica en detalle los procesos realizados para implementar el software en la institución, lo que se requiere y consideraciones que sean relevantes.

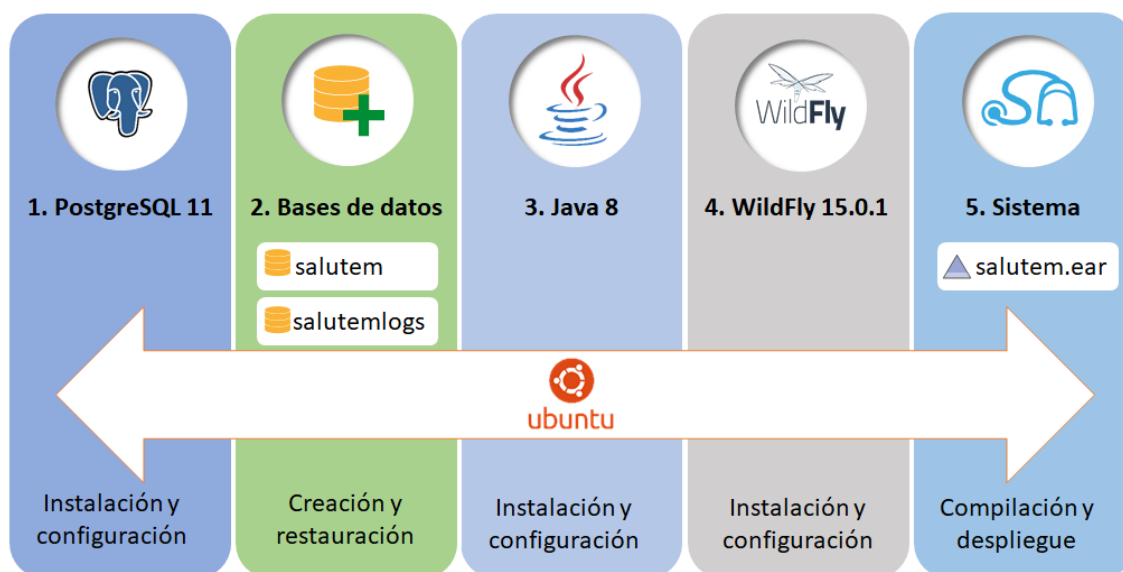


Figura 4.22: Plan de Implementación

Fuente: Realización Propia

4.7.1 PLAN DE IMPLEMENTACIÓN

Como se puede apreciar en la **Figura 4.22**, para la implementación del sistema se instalarán y configurarán los servicios necesarios utilizando como plataforma Ubuntu Server. El detalle de cada proceso se describe en el Manual Técnico que será incluido como Anexo al presente proyecto.

Tabla 4.8: Plan de Implementación

Tarea	Planificación
Instalación y configuración de PostgreSQL	05/11/2018
Creación y restauración de las bases de datos	05/11/2018
Instalación y configuración de Java 8	05/11/2018
Instalación y configuración de WildFly	05/11/2018
Compilación final y despliegue del Sistema	05/11/2018

4.7.2 REQUERIMIENTOS

A continuación, se detalla el hardware y software recomendado para la implementación del sistema tanto del lado del servidor como del cliente.

SOFTWARE PARA SERVIDOR

- Ubuntu Server 18.0.4 LTS
- Oracle JDK 1.8
- WildFly 15.0.1 Final
- PostgreSQL 11.1

SOFTWARE PARA USUARIO

- Navegador web, preferiblemente Google Chrome o Mozilla Firefox
- Lector de PDFs, para visualizar reportes
- Libre Office Calc o Microsoft Excel, para visualizar reportes

HARDWARE PARA SERVIDOR

- RAM al menos de 8GB
- Procesador Core i5 o Core I7
- Tarjeta de Red 1Gbps
- Disco duro o unidad en estado sólido de al menos 128 GB, preferiblemente 500 GB

HARDWARE PARA USUARIO

- RAM al menos de 2GB
- Procesador Celeron, Pentium o Core i3 de última generación
- Disco duro o unidad en estado sólido de al menos 128 GB

4.7.3 MANUAL TÉCNICO

Ver Anexo 4

4.7.4 MANUAL DE USUARIO

Ver Anexo 5

4.7.5 PLAN DE CAPACITACIÓN

La capacitación al usuario se hizo siguiendo un cronograma para su realización en la semana del 03 al 08 de diciembre de 2018 tomando en cada funcionalidad presente en el sistema y haciendo partícipes a Julio Villalba, gerente general de la Institución, Jorge Flores, médico y Luis Ordóñez como desarrollador del presente proyecto.

Tabla 4.9: Plan de Capacitación

Fecha	Actividad	Participantes
3/12/2018	Creación de parámetros asociados en grupos maestros y opciones de menú del sistema	Luis Ordóñez Julio Villalba
4/12/2018	Creación de usuarios y perfiles	Luis Ordóñez Julio Villalba
5/12/2018	Creación de horarios de atención Institucional, profesionales médicos, horarios de atención por profesional, pacientes y formato de fichas médicas.	Luis Ordóñez Julio Villalba
6/12/2018	Agendamiento, reagendamiento y cancelación de citas médicas regidos por el horario de cada profesional	Luis Ordóñez Julio Villalba Jorge Flores
7/12/2018	Registro de Atención a Pacientes	Luis Ordóñez Julio Villalba Jorge Flores Luis Ordóñez
8/12/2018	Gestión de Órdenes de Laboratorio	Julio Villalba Jorge Flores

4.7.6 COMPARATIVA PROCESO ACTUAL VS PROCESO AUTOMATIZADO

Tabla 4.10: Comparativa Proceso Actual Vs. Automatizado

Grupo de	Actividad	Manual	Autom.	Dif.	%
Usuarios					
Recepcionistas	Registro de nuevos pacientes.	00:05	00:02	00:03	60.0
	Agendamiento de citas médicas	00:07	00:02	00:05	28.6
	Gestión de órdenes de laboratorio.	00:10	00:04	00:06	40.0
				Promedio	42.9
Médicos	Registro de atenciones a pacientes.	00:10	00:05	00:05	50.0
Pacientes	Agendamiento de citas médicas.	00:05	00:02	00:03	40.0

En la tabla comparativa anterior se puede apreciar el tiempo que disponían para realizar una tarea de forma manual con relación al tiempo que ocupan para realizar la misma tarea con el uso del sistema; con lo que se puede concluir que el sistema aportó a la optimización de tiempo para el recepcionista en un 42.9% para el médico en un 50% y para el Paciente en un 40%.

Cabe destacar que el tiempo que se tomó en cuenta para el registro de la atención al paciente por parte de un profesional médico, es sólo lo que se toma para digitar los resultados y no el tiempo que demora la consulta y evaluación general de una persona.

Se entrevistó también a un paciente, para averiguar el nivel de satisfacción sobre el uso del sistema al realizar una cita médica. La entrevista consta de diez preguntas que se pueden responder seleccionando un número en la escala de 1 a 5, siendo 5 el puntaje máximo de satisfacción; los resultados se muestran a continuación:

Tabla 4.11: Resultados de la entrevista

Nro.	Pregunta	Resultado
1	El sistema se encuentra siempre disponible.	5
2	La información que brinda el sistema es confiable.	5
3	Las consultas y reportes que brinda el sistema son exactos y no presenta inconsistencias.	5
4	El sistema facilita la actualización de los datos básicos de paciente.	5

5	La apariencia del sistema es amigable, facilitando la visualización de datos.	4
6	Para operar el sistema se requiere una capacitación extensa.	4
7	El sistema presenta los mensajes de advertencia adecuados.	4
8	El sistema ofrece las opciones para realizar agendar una cita médica sin mayor esfuerzo.	5
9	Se considera que el tiempo de atención médica se ha optimizado utilizando el sistema.	5
10	En general, el sistema satisface las necesidades del usuario.	5
	TOTAL	47

Del resultado obtenido en la entrevista se puede determinar que el nivel de satisfacción para la persona entrevistada con el uso del sistema es del 94%.

5 CONCLUSIONES

- La Organización *Salud Integral para Todos* ha estado llevando, desde sus inicios el registro de sus pacientes y su historial médico en formatos impresos, tal y como muestra la revisión del proceso de citas médicas anterior a la automatización.
- Luego de la determinación de requerimientos se diseñó un modelo de base de datos que permita desarrollar un sistema automatizado para gestionar fichas médicas, reemplazando así los archivadores físicos por almacenamiento digital, lo que optimiza recursos y ofrece un aporte significativo a la conservación del medio ambiente al reducir el uso de papel.
- Con las pruebas de funcionalidad del sistema los usuarios validaron su operatividad y con base en las pruebas de carga, rendimiento y estrés se puede determinar que el sistema tiene un comportamiento adecuado para una concurrencia de 50 a 100 usuarios, sin que ello afecte el tiempo de respuesta a cada petición.
- Una vez desarrollado el sistema se instalaron todos los servicios necesarios para la implantación final de este, poniendo a disposición de los usuarios una herramienta que les permite optimizar su tiempo de tal forma que el recepcionista no tiene que registrar los datos básicos de una persona más de una vez, el paciente puede realizar una cita con sólo acceder al sistema y el profesional puede observar la evolución de un paciente sin esfuerzo gracias al histórico médico registrado en el sistema.
- La metodología *Scrum* ayudó a la rápida obtención de resultados adecuados a las necesidades de los miembros de la institución, quienes formaron parte del equipo, y su aportación constante fue determinante para la finalización exitosa del presente proyecto.

6 RECOMENDACIONES

- Se recomienda resguardar los respaldos de la base de datos transaccional en un equipo diferente al servidor de producción para que sirva como instrumento de recuperación ante fallos.
- En siguientes fases de desarrollo se debería considerar hacer referencia a la Clasificación Internacional de Enfermedades (CIE) para llevar un registro completo y sistematizado con estándares internacionales los diagnósticos de cada paciente.
- Así mismo, para futuras implementaciones del sistema es recomendable que exista una base de medicamentos alineados con el Cuadro Nacional de Medicamentos Básicos que ayude en la prescripción médica que entrega el profesional al paciente.
- En caso de que la organización tenga previsto usar el sistema para hacer publicidad de sus servicios o productos mediante correo electrónico (*email marketing*), es recomendable hacer una comprobación de la validez de estos, optimizando así el alcance al público interesado. Cabe destacar que la libre elección del consumidor, en especial su derecho a la privacidad, pueden estar comprometidos por el uso de malas prácticas de *email marketing*, aunque eso depende de la legislación sobre protección de datos de cada país, tema que puede ser analizado en otro proyecto investigativo.

7 REFERENCIAS BIBLIOGRÁFICAS

- Canonical Ltd. (12 de Enero de 2019). *Ubuntu*. Recuperado el 12 de Enero de 2019, de <https://www.ubuntu.com/community/debian>
- Dirección Nacional de Estadística y Análisis de la Información de Salud. (2015). *Producción Estadística 2006 - 2014*. Dirección Nacional de Estadística y Análisis de la Información de Salud, Producción Estadística. Recuperado el 13 de Enero de 2019, de <https://www.salud.gob.ec/wp-content/uploads/2015/01/PRODUCCION-ESTADISTICA-2006-2014.pdf>
- Free Software Foundation. (15 de Diciembre de 2018). *El sistema operativo GNU*. (F. S. Foundation, Editor) Recuperado el 08 de Enero de 2019, de <https://www.gnu.org/philosophy/philosophy.html>
- Galiano, J. L. (2016). *Implantar Scrum con éxito*. Barcelona: Oberta UOC Publishing, SL. Recuperado el 18 de Enero de 2019
- Groussard, T. (2010). *Java Enterprise Edition*. Barcelona, España: ENI. Retrieved Enero 12, 2019
- Gutiérrez Plaza, J., & Borillo Doménech, R. (2012). *2a. Conferencia Agile-Spain: CAS2011*. Universitat Jaume I. Servei de Comunicació i Publicacions. Recuperado el 15 de Enero de 2019
- HOSPITAL SANT JOAN DE DÉU BARCELONA. (7 de Mayo de 2013). *Los registros de pacientes son una herramienta importante para la investigación*. Recuperado el 05 de Enero de 2018, de <https://metabolicas.sjdhospitalbarcelona.org/noticia/registros-pacientes-herramienta-importante-investigacion>
- ICESoft Technologies Inc. (2017). *ICEfaces Overview*. Recuperado el 16 de Enero de 2019, de <http://www.icesoft.org/java/projects/ICEfaces/overview.jsf>
- Ministerio de Salud Pública del Ecuador. (2013). *Instructivo para el llenado del Registro Diario Automatizado de Consultas y Atenciones Ambulatorias (RDACAA)*. Recuperado el 13 de Enero de 2019, de https://aplicaciones.msp.gob.ec/salud/archivosdigitales/documentosDirecciones/dnn/archivos/instructivo-rdaca__final_04_09_2013.pdf

-
- Oracle. (2014). *Java Platform, Enterprise Edition The Java EE Tutorial, Release 7*. Recuperado el 14 de Enero de 2019, de <https://docs.oracle.com/javaee/7/tutorial/overview001.htm>
- Oracle. (2014). *Java Platform, Enterprise Edition The Java EE Tutorial, Release 7*. Recuperado el 14 de Enero de 2019, de <https://docs.oracle.com/javaee/7/tutorial/overview002.htm>
- Oracle. (2014). *Java Platform, Enterprise Edition The Java EE Tutorial, Release 7*. Recuperado el 15 de Enero de 2019, de <https://docs.oracle.com/javaee/7/tutorial/overview003.htm>
- Oracle. (2014). *Java Platform, Enterprise Edition The Java EE Tutorial, Release 7*. Recuperado el 16 de Enero de 2019, de <https://docs.oracle.com/javaee/7/tutorial/overview004.htm>
- Oracle. (2014). *Java Platform, Enterprise Edition The Java EE Tutorial, Release 7*. Recuperado el 14 de Enero de 2019, de <https://docs.oracle.com/javaee/7/tutorial/overview007.htm>
- Organización Panamericana de la Salud. (2016). *Electronic Medical Records in Latin America and the Caribbean: An Analysis of the current situation*. Organización Panamericana de la Salud. Recuperado el 13 de Enero de 2019, de <http://iris.paho.org/xmlui/handle/123456789/28210>
- Quiroz, A. (27 de Septiembre de 2011). *Procesos de Investigación Jurídica*. Recuperado el 05 de Enero de 2019, de <https://web.archive.org/web/20121001112822/http://alfredoquiroz.wordpress.com/2011/09/27/aviso-importante-8/>
- Rodríguez, M. (2017). *Scrum desde cero*. Madrid: Mc. Graw-Hill.
- Sanaualla, M. (25 de Junio de 2008). *CamelCase Notation- Naming Convention for Programming Languages*. Recuperado el 28 de Febrero de 2019, de <https://sanaualla.info/2008/06/25/camelcase-notation-naming-convention-for-programming-languages/>
- Stancapiano, L. (2017). *Mastering Java EE Development with WildFly*. Packt Publishing Ltd. Retrieved Enero 12, 2019
- Sun Microsystems Inc. (1999). *Convenciones de Código para el lenguaje de programación JAVA*. Recuperado el 13c de Enero de 2019, de Convenciones de Código para el lenguaje de programación

-
- The Apache Software Foundation. (2018). *Apache JMeter*. Recuperado el 31 de Enero de 2019, de <http://jmeter.apache.org/usermanual/get-started.html>
- The PostgreSQL Global Development Group. (2019, Enero 12). *PostgreSQL*. Retrieved Enero 12, 2019, from <https://www.postgresql.org/about/>
- Toasa, R., Maximiano, M., Reis, C., & Guevara, D. (2018). Data visualization techniques for real-time information—A custom and dynamic dashboard for analyzing surveys' results. *13th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-7.
- w3techs. (21 de Octubre de 2013). *Debian/Ubuntu extend the dominance in the Linux web server market at the expense of Red Hat/CentOS*. Recuperado el 15 de 10 de 2018, de https://w3techs.com/blog/entry/debian_ubuntu_extend_the_dominance_in_the_linux_web_server_market_at_the_expense_of_red_hat_centos

8 ANEXOS

Anexo 1: Fichas de Observación

Anexo 2: Firmas de Aceptación

Anexo 3: Encuesta de Satisfacción

Anexo 4: Manual Técnico

Anexo 5: Manual de Usuario

ANEXO 1

Fichas de Observación

Tabla 8.1: Ficha de Observación 01 – Parametrización del Sistema

Ficha de Observación

Número. 01 **Observador.** Luis F. Ordóñez A.
 Fecha. 03/09/2018 **Actividad.** Parametrización del Sistema
 Grupo de Usuarios. Super administrador

	Registro de información inicial		
	Siempre	Casi siempre	Casi nunca
Creación de nuevos usuarios		x	
Creación de perfiles			x
Creación de nuevas opciones de menú			x

	Registro transaccional		
	Siempre	Casi siempre	Casi nunca
Restauración de contraseñas		x	
Asignación de grupos a cada usuario		x	
Modificación de perfiles de usuario			x

	Extracción de reportes		
	Siempre	Casi siempre	Casi nunca
Listado de usuarios		x	
Reporte de accesos al sistema			x
Listar historial de cambios			x

Observaciones

La administración de parámetros del Sistema debe ser gestionada por personal debidamente autorizado y con conocimiento del alcance que podría tener una modificación a este nivel porque, aunque la información que se puede manipular no está directamente relacionada al giro de negocio de la institución, claramente tendrá efectos en el buen funcionamiento de todo el sistema.

Tabla 8.2: Ficha de Observación 02 – Gestión de Información Institucional

Ficha de Observación			
Número. 02	Observador. Luis F. Ordóñez A.		
Fecha. 03/09/2018	Actividad. Gestión de Información Institucional		
Grupo de Usuarios. Administrador Institucional			
Registro de información inicial	Frecuencia		
	Siempre	Casi siempre	Casi nunca
Creación de profesionales médicos	x		
Creación de horarios	x		
Personalización de fichas médicas por especialidad	x		
Registro transaccional	Frecuencia		
	Siempre	Casi siempre	Casi nunca
Modificación de información de médicos		x	
Modificación de Datos Institucionales		x	
Modificación de fichas médicas			x
Extracción de reportes	Frecuencia		
	Siempre	Casi siempre	Casi nunca
Listado de pacientes y profesionales	x		
Reporte de agendamiento de citas	x		
Reporte de atenciones médicas	x		

Observaciones

La gestión de la Información Institucional debe ser asignada a usuario que tengan profundo conocimiento del giro de negocio de la institución ya que las opciones de configuración que se ofrecen para este grupo de usuarios afectan directamente al funcionamiento específico de esta.

Tabla 8.3: Ficha de Observación 03 – Gestión de Pacientes**Ficha de Observación**

Número. 03

Observador. Luis F. Ordóñez A.

Fecha. 04/09/2018

Actividad. Gestión de pacientes

Grupo de Usuarios. Recepcionistas

	Registro de información inicial		
	Siempre	Casi siempre	Casi nunca
Creación de nuevos pacientes			x
	Registro transaccional		
	Siempre	Casi siempre	Casi nunca
Modificación de datos de pacientes			x
Agendamiento de citas		x	
Gestión de órdenes de laboratorio	x		
	Extracción de reportes		
	Siempre	Casi siempre	Casi nunca
Listado de pacientes	x		
Consulta de agenda médica	x		
Reporte de órdenes de laboratorio	x		

Observaciones

Los recepcionistas se encargarán del ingreso de nuevos pacientes, la gestión de órdenes de laboratorio y el agendamiento de citas médicas dependiendo del horario definido para cada profesional

Tabla 8.4: Ficha de observación 04 – Atención a pacientes**Ficha de Observación**

Número. 04

Observador. Luis F. Ordóñez A.

Fecha. 04/09/2018

Actividad. Atención a pacientes

Grupo de Usuarios. Médicos

Registro de información inicial**Frecuencia****Siempre****Casi siempre****Casi nunca**

Personalización de fichas médicas por especialidad x

Registro transaccional**Frecuencia****Siempre****Casi siempre****Casi nunca**

Modificación de fichas médicas x

Registro de atención médica x

Extracción de reportes**Frecuencia****Siempre****Casi siempre****Casi nunca**

Consulta de horario de atención x

Consulta de agenda médica x

Reporte de atenciones x

Observaciones

Los médicos tendrán acceso a la personalización de fichas médicas según su especialidad, registrar las atenciones a pacientes con previa cita o en casos emergentes sin ella y tendrán acceso a reportes de horarios, agendamiento de citas y atenciones realizadas.

Tabla 8.5: Ficha de Observación 05 – Agendamiento de Citas**Ficha de Observación**

Número. 05

Observador. Luis F. Ordóñez A.

Fecha. 04/09/2018

Actividad. Agendamiento de citas

Grupo de Usuarios. Pacientes

	Registro de información inicial		
	Siempre	Casi siempre	Casi nunca
Información personal			x
	Registro transaccional		
	Siempre	Casi siempre	Casi nunca
Agendamiento de citas		x	
	Extracción de reportes		
	Siempre	Casi siempre	Casi nunca
Consulta de horarios de atención	x		
Consulta de agenda médica	x		
Reporte de atenciones	x		

Observaciones

El paciente tendrá la opción de agendar citas dependiendo de la agenda disponible para cada médico y tendrá la capacidad de ver las atenciones médicas registradas a su nombre. Es necesario que el paciente esté previamente registrado por el recepcionista.

ANEXO 2

Firmas de aceptación



Quito, 15 de enero de 2019

Por medio del presente documento, la organización **“Salud Integral para Todos”** de la ciudad de Quito, acepta y da por concluido el trabajo de desarrollo del Sistema Web de Gestión de Fichas Médicas, realizado por el Sr. Luis Fernando Ordóñez Armijos, después de haberse hecho las pruebas necesarias y suficientes con las que se ha demostrado que cumple satisfactoriamente con los requerimientos sostenidos de acuerdo al análisis realizado.

(SIP) Salud Integral
para todos

M.S.P. 2017-209-001-1100

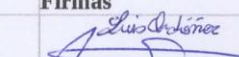


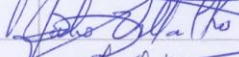
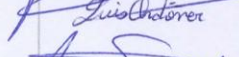

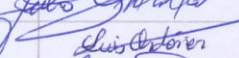

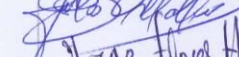
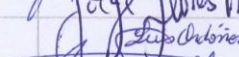

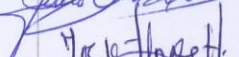



Ing. Julio Villalba Guashamin

1703325934

REPRESENTANTE

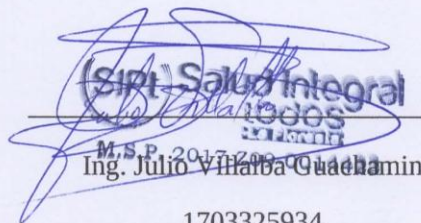
SALUD INTEGRAL PARA TODOS

Pruebas de Funcionalidad del Sistema de Gestión de Fichas Médicas

Fecha	Actividad	Participantes	Firmas
3/12/2018	Creación de parámetros asociados en grupos maestros y opciones de menú del sistema	Luis Ordóñez	
		Julio Villalba	
4/12/2018	Creación de usuarios y perfiles	Luis Ordóñez	
		Julio Villalba	
5/12/2018	Creación de horarios de atención Institucional, profesionales médicos, horarios de atención por profesional, pacientes y formato de fichas médicas.	Luis Ordóñez	
		Julio Villalba	
6/12/2018	Agendamiento, reagendamiento y cancelación de citas médicas regidos por el horario de cada profesional	Luis Ordóñez	
		Julio Villalba	
7/12/2018	Registro de Atención a Pacientes	Jorge Flores	
		Luis Ordóñez	
8/12/2018	Gestión de Órdenes de Laboratorio	Julio Villalba	
		Jorge Flores	
		Luis Ordóñez	
		Julio Villalba	
		Jorge Flores	

Validaciones realizadas en las pruebas de funcionalidad del Sistema

Actividad	Validación
Creación de parámetros asociados en grupos maestros y opciones de menú del sistema	Se verifica que: - cada parámetro tiene un código único y no permite duplicados. - las opciones de menú se agrupan por módulos. - se pueden crear Instituciones e identificarlas como Laboratorios.
Creación de usuarios y perfiles	Se verifica que el Sistema: - impide el registro de identificaciones y nombres de usuario duplicados. - asigna temporalmente como contraseña inicial la identificación encriptada en MD5 - solicita el cambio de contraseña al primer inicio de sesión del nuevo usuario - permite el ingreso de usuarios debidamente registrados - permite la configuración de un conjunto de opciones de menú específicas a las que un grupo de usuarios tendrá acceso - limita las operaciones de creación, actualización, eliminación o simple consulta sobre cada registro que una opción de menú permita.
Creación de horarios de atención Institucional, profesionales médicos, horarios de atención por profesional, pacientes y formato de fichas médicas.	Se verifica que tanto un profesional como un paciente tenga un registro único dentro de la Institución médica y que cada especialidad tiene una ficha configurable que se utiliza para el registro de atención al paciente
Agendamiento, reagendamiento y cancelación de citas médicas regidos por el horario de cada profesional	Se verifica que: - una cita no puede ocupar el mismo espacio de tiempo que otra - no se puede agendar en horas que la agenda de cada médico no contemple - el paciente no puede cancelar o reagendar citas que no sean las suyas. - se notifica mediante correo electrónico cuando se agenda o reagenda una cita
Registro de Atención a Pacientes	Se verificará que no se puede registrar más de una atención por cada cita agendada y una atención no pueda ser modificada pasado el día de su registro.
Gestión de Órdenes de Laboratorio	Se verifica que cada atención perteneciente a la especialidad de Optometría genera una orden de Laboratorio, misma que pasa por un proceso de envío a un Laboratorio registrado, recepción del paquete enviado y finalmente entrega al paciente.


 M.S.P. 2017-700-00000000
 Ing. Julio Villalba Guadalupe

1703325934

REPRESENTANTE

SALUD INTEGRAL PARA TODOS

ANEXO 3

Encuesta de Satisfacción

Encuesta de satisfacción de usuarios del Sistema de Fichas Médicas *Salutem*

Para responder marque la casilla con el valor que usted considere correcto.

Nro.	Pregunta	Escala				
		1	2	3	4	5
1	El sistema se encuentra siempre disponible.					
2	La información que brinda el sistema es confiable.					
3	Las consultas y reportes que brinda el sistema son exactos y no presenta inconsistencias.					
4	El sistema facilita la actualización de los datos básicos de paciente.					
5	La apariencia del Sistema es amigable, facilitando la visualización de datos.					
6	Para operar el sistema se requiere una capacitación extensa.					
7	El sistema presenta los mensajes de advertencia adecuados.					
8	El sistema ofrece las opciones para realizar agendar una cita médica sin mayor esfuerzo.					
9	Se considera que el tiempo de atención médica se ha optimizado utilizando el sistema.					
10	En general, el sistema satisface las necesidades del usuario.					

ANEXO 4

Manual Técnico

1. INSTALACIÓN Y CONFIGURACIÓN DE POSTGRESQL 11



1. Instalar las herramientas wget y vim en el caso de no tenerlas

```
sudo apt install -y wget vim
```

2. Importar la clave de la firma del repositorio

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo  
apt-key add -
```

3. Agregar repositorio

```
RELEASE=$(lsb_release -cs)  
  
echo "deb http://apt.postgresql.org/pub/repos/apt/ ${RELEASE}"-pgdg main |  
sudo tee /etc/apt/sources.list.d/pgdg.list
```

4. Comprobar existencia del repositorio

```
cat /etc/apt/sources.list.d/pgdg.list
```

5. Instalar postgresql 11

```
sudo apt update  
sudo apt -y install postgresql-11
```

6. Configurar postgresql

Teniendo en cuenta que el servidor cuenta con 4GB de RAM, tomar 2GB para uso exclusivo para el servidor de Base de Datos:

```
Gestor de Base de Datos: Postgresql 11  
Sistema Operativo: Ubuntu Server 18.04  
Tipo de aplicación: web  
Memoria designada: 2 GB  
Número de CPUS (núcleos): 2  
Número de conexiones: 400  
Tipo de almacenamiento: SSD
```

6.1. Editar el archivo de configuración *postgresql.conf*

```
sudo vim /etc/postgresql/11/main/postgresql.conf
```

Tomar como referencia el nombre de la propiedad y el número de línea descrita a continuación (utilizar el comando *:set nu* de vim para mostrar las líneas):

```
59 listen_addresses = '*'
```

```

64 max_connections = 400
117 shared_buffers = 512MB
126 work_mem = 1310kB
127 maintenance_work_mem = 128MB
166 effective_io_concurrency = 200
167 max_worker_processes = 2
169 max_parallel_workers_per_gather = 1
171 max_parallel_workers = 2
213 max_wal_size = 2GB
214 min_wal_size = 1GB
215 checkpoint_completion_target = 0.7
202 wal_buffers = 16MB
316 random_page_cost = 1.1
334 effective_cache_size = 1536MB
348 default_statistics_target = 100

```

Tomar en cuenta la página web:

https://pgtune.leopard.in.ua/?utm_source=postgresweekly&utm_medium=email#/
 donde se recomienda los valores de configuración dependiendo los recursos disponibles, tales como: memoria, cpu, almacenamiento, etc.

6.2. Generar contraseña para el usuario postgres

```

sudo su - postgres
psql
postgres=# \password
Ingrese la nueva contraseña: clave
Ingrésela nuevamente: clave
postgres=# \q

```

6.3. Editar el archivo de configuración *pg_hba.conf*

```
vim /etc/postgresql/11/main/pg_hba.conf
```

En la sección **# IPv4 local connections**, agregar:

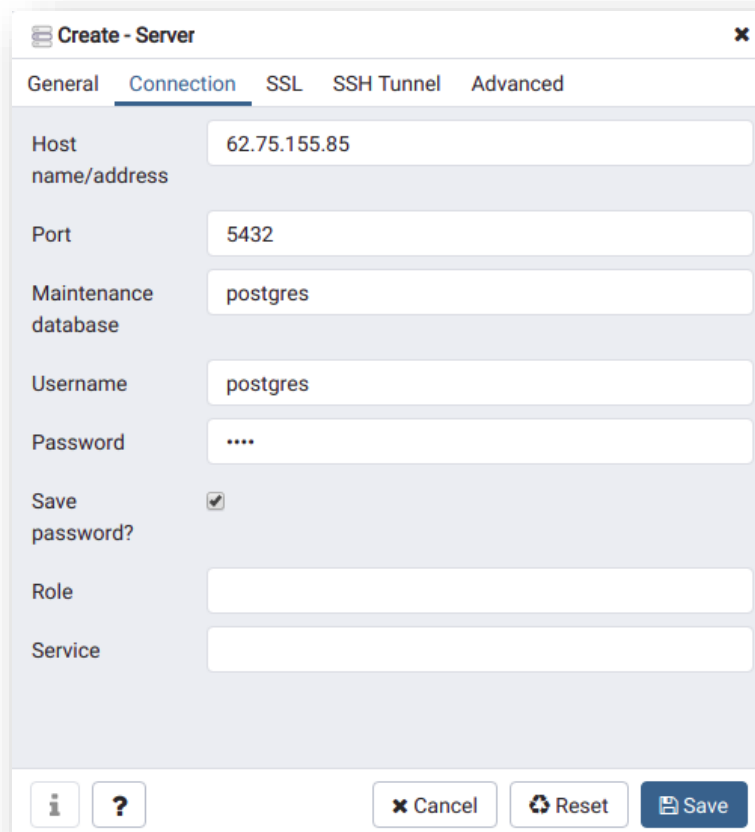
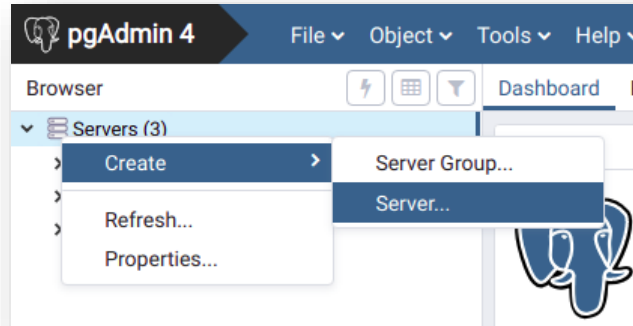
```
host          all          all          0.0.0.0/0          md5
```

Para que los cambios surtan efecto es necesario reiniciar el servidor postgresql con el comando:

```
sudo systemctl restart postgresql
```

7. Crear la conexión al Servidor PostgreSQL

Acceder al servidor mediante pgAdmin 4 y crear una nueva conexión al servidor de base de datos.

A screenshot of the 'Create - Server' dialog box in pgAdmin 4. The dialog has a title bar with a close button and a tabbed interface with 'General', 'Connection', 'SSL', 'SSH Tunnel', and 'Advanced' tabs. The 'Connection' tab is active. The form contains the following fields:

- Host name/address: 62.75.155.85
- Port: 5432
- Maintenance database: postgres
- Username: postgres
- Password: masked with dots
- Save password?: checked
- Role: empty
- Service: empty

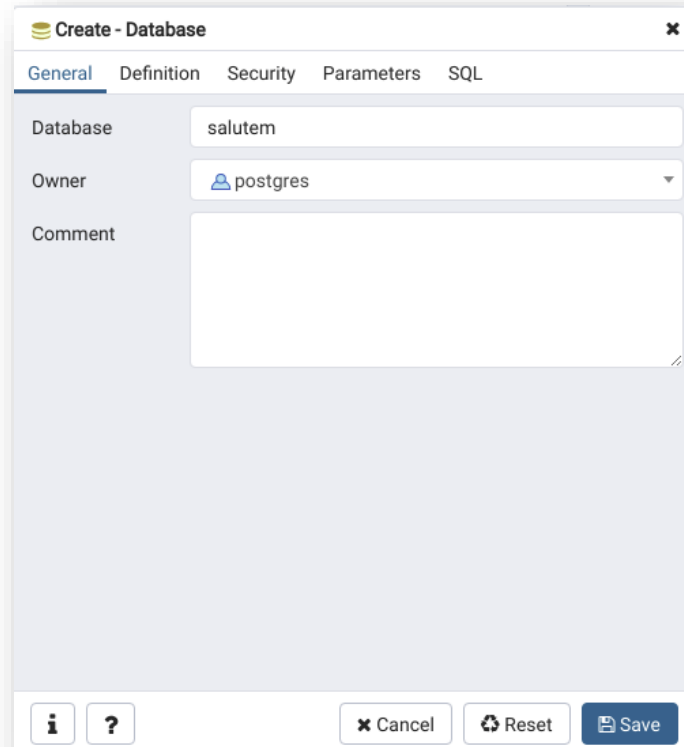
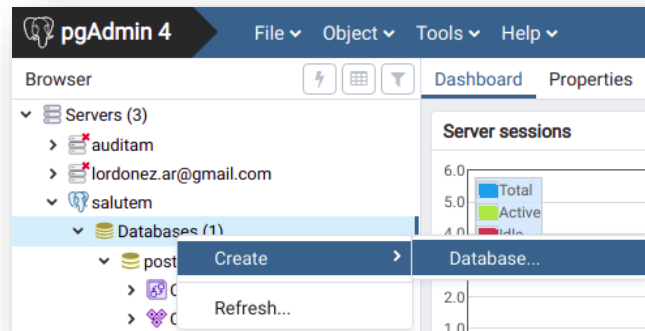
At the bottom, there are buttons for 'Cancel', 'Reset', and 'Save', along with information and help icons.

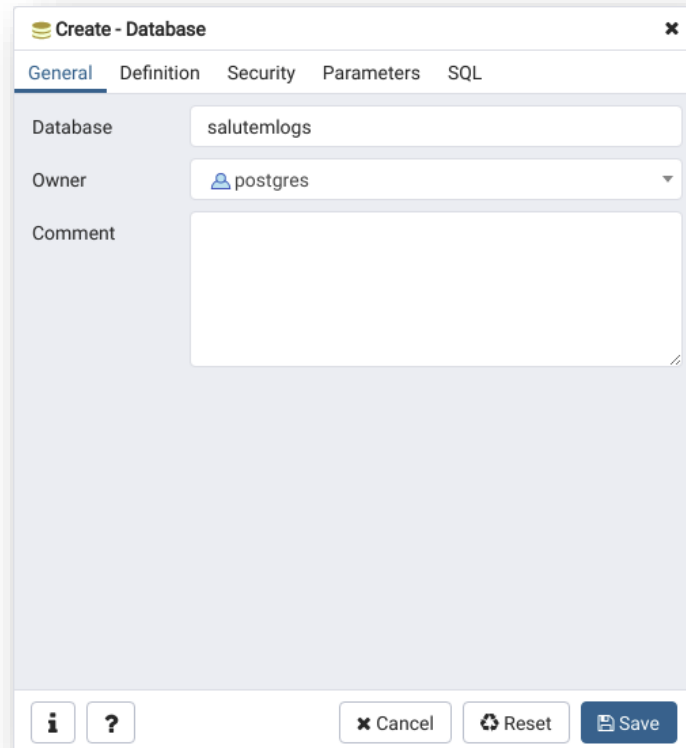
3. CREACIÓN DE LAS BASES DE DATOS Y RESTAURACIÓN



1. Crear las bases de datos:

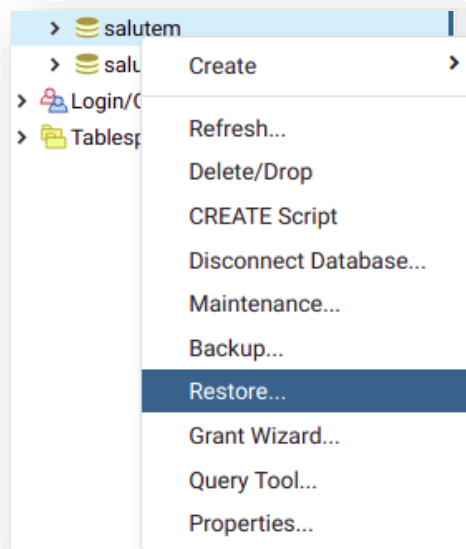
- *salutem*; y
- *salutemlogs*






2. Restaurar las bases de datos


Luego restauraremos las bases de datos con los registros necesarios para iniciar el sistema:



Restore (Database: salitem) 



General | Restore options

Format

Filename 

Number of jobs

Role name

3. INSTALACIÓN Y CONFIGURACIÓN DE ORACLE JAVA 8.



1. Instalar el gestor de repositorios con el comando:

```
sudo apt install software-properties-common -y
```

2. Agregar el repositorio de java de oracle:

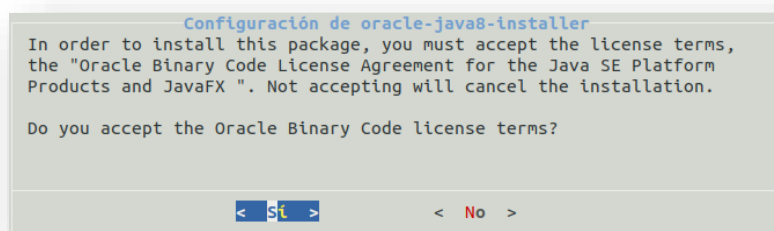
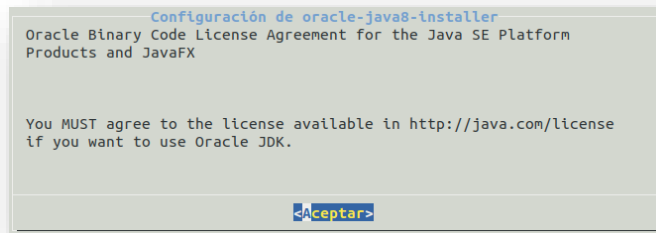
```
sudo add-apt-repository ppa:webupd8team/java
```

3. Instalar Oracle Java 8:

```
sudo apt update
```

```
sudo apt -y install oracle-java8-installer
```

4. Aceptar los términos de licencia:



5. Configurar JDK 8 como predeterminado

```
sudo apt install oracle-java8-set-default
```

6. Comprobar la versión de java:

```
java -version
```

5. INSTALACIÓN Y CONFIGURACIÓN DE WILDFLY 15.0.1



1. Crear usuario

Crear un usuario (preferiblemente un usuario sin acceso a opciones de superusuario, por motivos de seguridad) con el nombre *wildfly*.

```
sudo adduser wildfly
Enter new UNIX password: clave
Retype new UNIX password: clave
```

2. Instalar WildFly 15.0.1

Luego de acceder con el usuario wildfly ejecutar:

```
wget https://download.jboss.org/wildfly/15.0.1.Final/wildfly-15.0.1.Final.zip
```

Descomprimir el paquete recién descargado:

```
unzip wildfly-15.0.1.Final.zip
```

3. Configurar el módulo de PostgreSQL en Wildfly

Acceder al directorio del servidor para la creación de módulos:

```
cd /home/wildfly/wildfly-15.0.1.Final/modules/system/layers/base/org
```

Crear la estructura postgresql/main y acceder a ella:

```
mkdir -p postgresql/main
cd postgresql/main
```

Descargar *el driver jdbc* de postgresql

```
wget https://jdbc.postgresql.org/download/postgresql-42.2.5.jar
```

Crear el archivo *module.xml*:

```
vim module.xml
```

Agregar al archivo el siguiente texto:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-42.2.5.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

4. Crear *datasources* en WildFly

Ahora hay que indicarle al servidor que se conectará a las bases de datos usando el módulo de PostgreSQL, para lo cual se editará el archivo *standalone.xml*

```
vim /home/wildfly/wildfly-15.0.1.Final/standalone/configuration/standalone.xml
```

Ubicar la etiqueta `<datasources>` y agregar los dos recursos que conectarán a las bases de datos: *salutem* y *salutemlogs*

```
<datasource jta="true" jndi-name="java:/Salutem" pool-name="Salutem"
enabled="true" use-java-context="true" use-ccm="true">
  <connection-url>jdbc:postgresql://localhost:5432/salutem</connection-url>
  <driver>org.postgresql</driver>
  <pool>
    <min-pool-size>8</min-pool-size>
    <max-pool-size>32</max-pool-size>
  </pool>
  <security>
    <user-name>postgres</user-name>
    <password>clave</password>
  </security>
</datasource>

<datasource jta="true" jndi-name="java:/Salutemlogs" pool-name="Salutemlogs"
enabled="true" use-java-context="true" use-ccm="true">
  <connection-url>jdbc:postgresql://localhost:5432/salutemlogs</connection-
url>
  <driver>org.postgresql</driver>
  <pool>
    <min-pool-size>8</min-pool-size>
    <max-pool-size>32</max-pool-size>
  </pool>
  <security>
    <user-name>postgres</user-name>
    <password>clave</password>
  </security>
</datasource>
```

Luego ubicar la etiqueta `<drivers>` y agregar:

```
<driver name="org.postgresql" module="org.postgresql">
  <xa-datasource-class>org.postgresql.xa.PGXADatasource</xa-datasource-
class>
</driver>
```


5. Modificar de las interfaces de WildFly

Para poder acceder a la interfaz de administración y a las aplicaciones instaladas en el servidor es necesario modificar el archivo `standalone.xml` ubicando la etiqueta `<interfaces>` y reemplazar el contenido por:

```
<interface name="management">
  <any-address/>
</interface>
<interface name="public">
  <any-address/>
</interface>
```

6. Habilitar la subida de archivos mayores a 10 MB

Para habilitar la subida de archivos mayores a 10MB es necesario modificar el archivo `standalone.xml` ubicando las etiquetas: `<http-listener>`; `<https-listener>` y reemplazar por:

```
<http-listener name="default" socket-binding="http" max-post-size="107374182400" redirect-socket="https" enable-http2="true"/>
<https-listener name="https" socket-binding="https" max-post-size="107374182400" security-realm="ApplicationRealm" enable-http2="true"/>
```

En donde la propiedad: `max-post-size="107374182400"` definirá el tamaño máximo de los archivos que se puedan subir al servidor, en este caso 100MB

7. Crear usuario para configuración remota

Ejecutamos el comando `add-user.sh`

```
/home/wildfly/wildfly-15.0.1.Final/bin/add-user.sh
```

Seguir las indicaciones del asistente para crear un usuario administrador (Management User), para el caso crear el usuario con nombre `wildflyadmin`.

8. Modificar opciones JVM

Para modificar las opciones de memoria de la máquina virtual de java editar el archivo: `standalone.conf`

```
vim /home/wildfly/wildfly-15.0.1.Final/bin/standalone.conf
```

Ubicar los valores dentro de `JAVA_OPTS`:

```
-Xms64m
-Xmx512m
-XX:MetaspaceSize=96M
-XX:MaxMetaspaceSize=256m
```

Reemplazar los valores tal que la opción *Xmx* sea igual a la memoria RAM asignada para el servidor de aplicaciones, para el caso caso: **2GB=2048MB** (se puede multiplicar por 4 cada valor):

```
-Xms256m  
-Xmx2048m  
-XX:MetaspaceSize=384M  
-XX:MaxMetaspaceSize=1024m
```

9. Iniciar del servidor

Para iniciar el servidor ejecutar:

```
/home/wildfly/wildfly-15.0.1.Final/bin/standalone.sh&
```

10. Parar el servidor

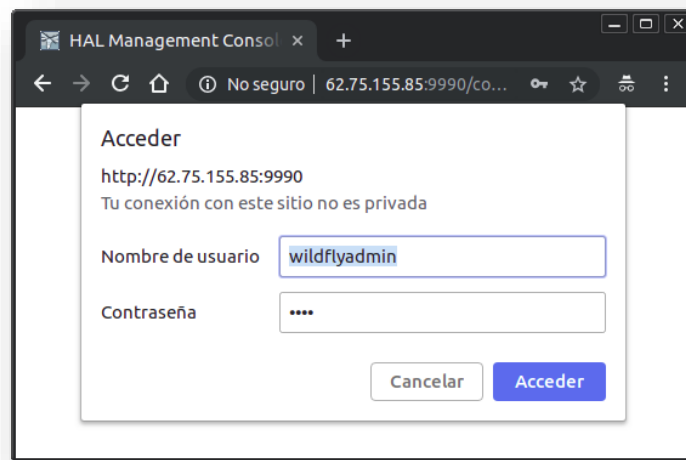
Para dar de baja el servidor wildfly ejecutar:


```
/home/wildfly/wildfly-15.0.1.Final/bin/jboss-cli.sh --connect  
command=:shutdown
```

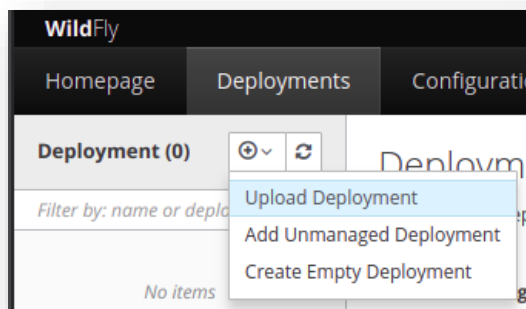
6. DESPLIEGUE DE LA APLICACIÓN

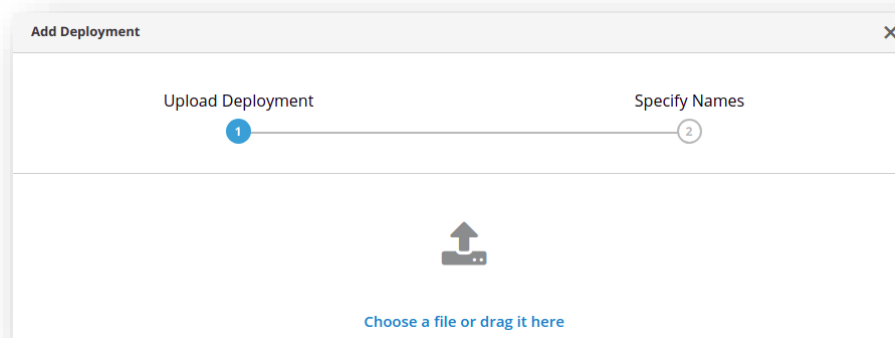


Para desplegar el sistema luego de ser compilado y haber generado el *archivo.ear*, es necesario ingresar a la consola de administración vía web, digitando en el navegador la ip del servidor junto al puerto de administración: 9990 usando las credenciales generadas anteriormente para el efecto:

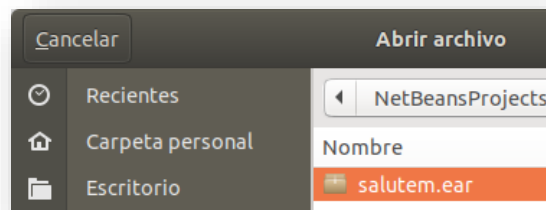


Luego acceder a la Sección *Deployments* hacer clic en el icono  y seleccionar *Upload Deployment*:

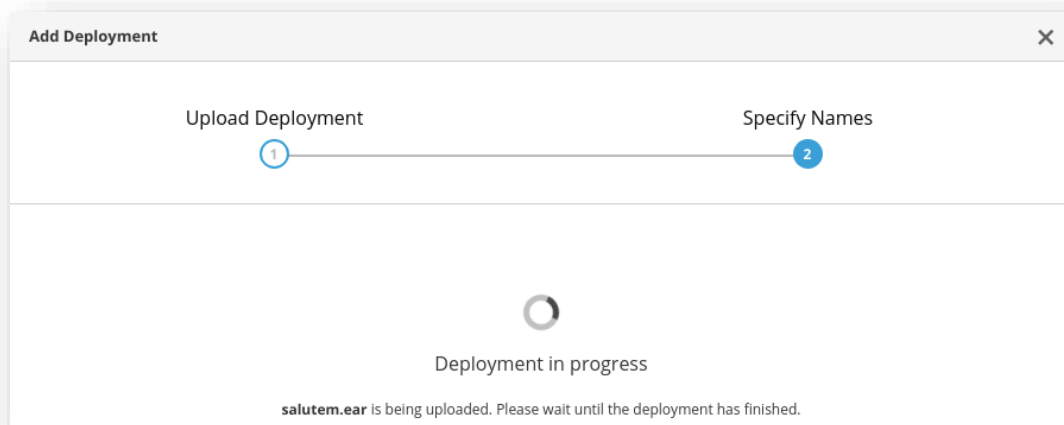




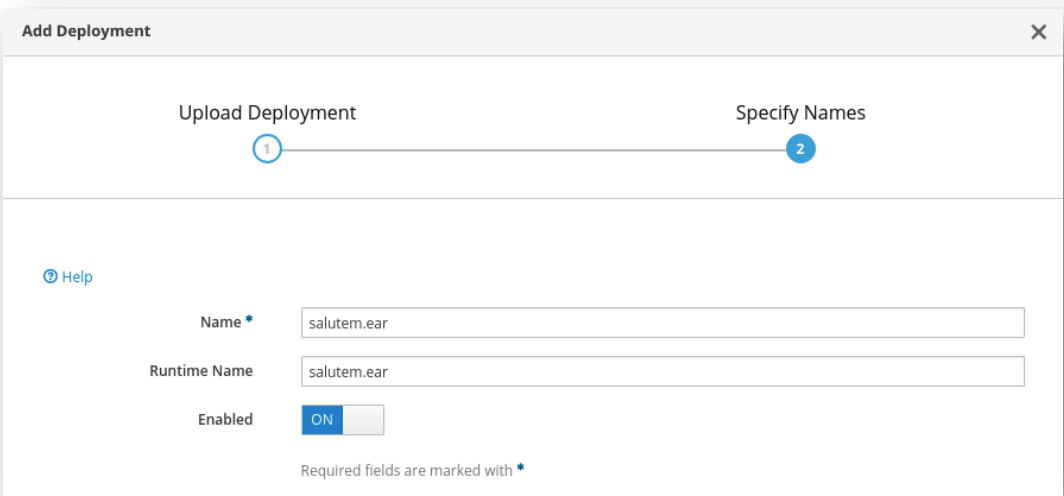
Dar clic en *Choose a file or drag it here* y subir el archivo *salutem.ear*



Hacer clic en *NEXT*

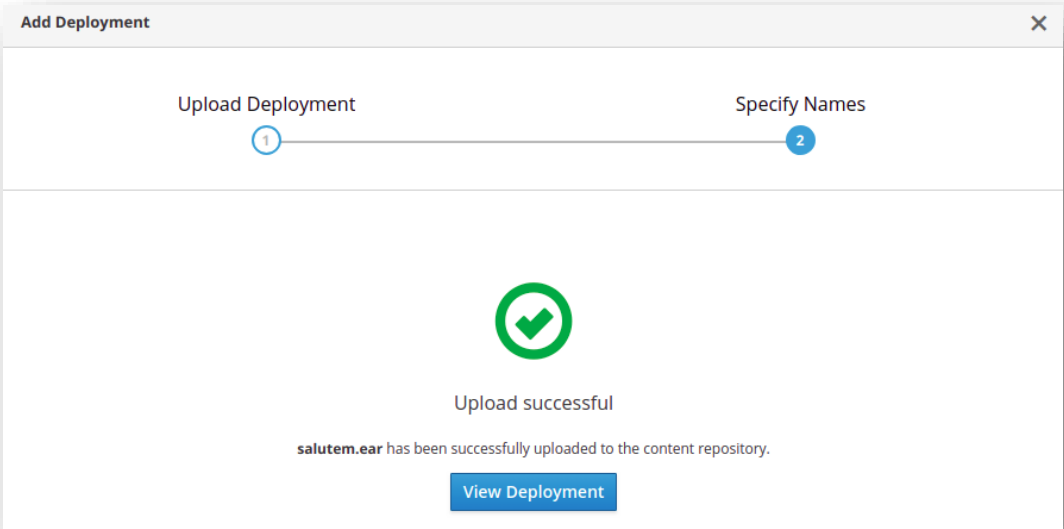


Poner en *ON* la opción *Enabled* y dar clic en *Finish*



The screenshot shows a dialog box titled "Add Deployment" with a close button (X) in the top right corner. At the top, there is a progress indicator with two steps: "Upload Deployment" (marked with a circled 1) and "Specify Names" (marked with a circled 2). Below the progress indicator, there is a "Help" icon and text. The main area contains three input fields: "Name *" with the value "salutem.ear", "Runtime Name" with the value "salutem.ear", and "Enabled" with a toggle switch set to "ON". A note at the bottom states "Required fields are marked with *".

Esperar un momento, se sabrá que todo salió bien si aparece el siguiente cuadro de resumen:



The screenshot shows the same "Add Deployment" dialog box, but now it displays a success message. At the top, the progress indicator shows "Upload Deployment" (marked with a circled 1) and "Specify Names" (marked with a circled 2). In the center, there is a large green checkmark icon. Below the icon, the text reads "Upload successful" and "salutem.ear has been successfully uploaded to the content repository." At the bottom, there is a blue button labeled "View Deployment".

Cerrar el asistente. Todo está listo para acceder al sistema vía web digitando en la barra del direcciones del navegador la dirección que apunta al sistema con el puerto 8080 <http://ipserver:8080/salutem/>



ANEXO 5

Manual de Usuario

MANUAL DE USUARIO

SISTEMA DE FICHAS MÉDICAS



MANUAL DE USUARIO SISTEMA DE FICHAS MÉDICAS SALUTEM

INTRODUCCIÓN

El presente manual tiene por objetivo guiar al usuario en el manejo adecuado de la aplicación web para la gestión del Sistema de Fichas Médicas realizando una descripción de las principales pantallas y su funcionalidad.

INGRESO AL SISTEMA

Para ingresar a la aplicación desde Internet se lo hace a través de un navegador web. En la barra de direcciones se debe escribir:

<http://Direccion_IP_Servidor:Puerto/Nombre_Aplicación_Web/>

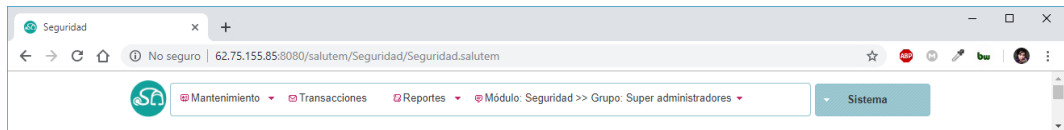
Como se muestra en la imagen, o la dirección web designada por el área que administra el sistema, y presionar <ENTER>.



En la ventana de Ingreso del sistema, se debe digitar las credenciales de acceso (Usuario y contraseña) y presionar el botón *Ingresar* o bien *¡Olvidé mi contraseña!* en el caso de necesitar el restablecimiento de sus credenciales



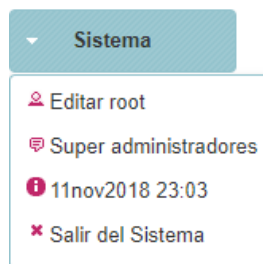
VISTA PRINCIPAL DEL SISTEMA



Compuesta principalmente por menús de navegación mediante las opciones principales:

- Mantenimiento
- Transacciones
- Reportes
- Módulo:
- Y el botón Sistema

Al dar clic en el botón *Sistema* se puede ver: *el nombre del usuario, el grupo al que está asignado, la versión del sistema y la opción para Salir del Sistema.*



En donde *Editar* se puede modificar la información de la institución, cambiar la clave o revisar el historial de cambios.

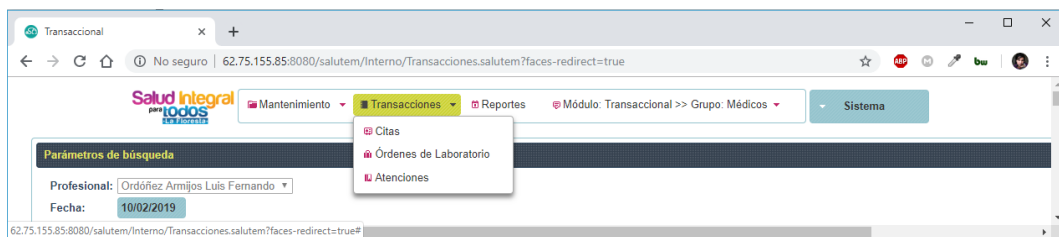


MENÚ DEL SISTEMA

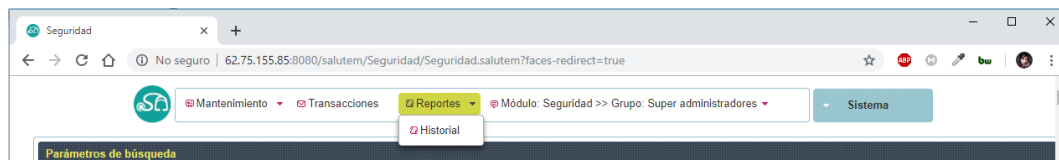
Mantenimiento: Este menú contiene las opciones para la creación de los datos para la gestión del proceso de fichas médicas.



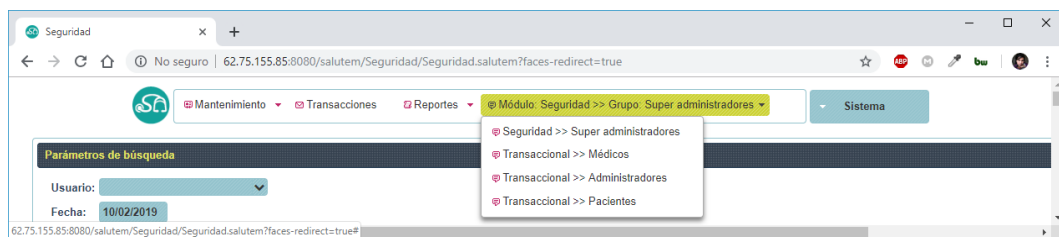
Transacciones: Este menú contiene las opciones de agendamiento de citas médicas, gestiones de órdenes de laboratorio y atenciones.



Reportes: Este menú contiene el historial de acciones.



Módulos: Este menú contiene los módulos



Seguridad – Super administradores: Es en donde el sistema proporciona las facilidades de crear y modificar parámetros asociados en grupos maestros.

Transaccional-Médicos: Permite agendar citas, emitir órdenes de laboratorio y atenciones.

Transaccional-Administradores: Permite visualizar las citas y atención de pacientes.

Transaccional-Pacientes: Permite agendar citas médicas.

FUNCIONALIDAD DE ÁREAS DE TRABAJO

A continuación, se explica la funcionalidad estándar de todos los componentes de una de las pantallas de trabajo que se utilizan en todas las opciones del sistema. El área de trabajo se divide en dos partes principales como se muestra en la imagen:

1. En la primera parte tenemos: Criterios de Búsqueda y los botones *Buscar* y *Nuevo* para agregar un nuevo registro.



2. En la segunda parte tenemos: *Resultado Búsqueda*, donde se cargan los registros obtenidos a través del botón *Buscar* y filtrados por los Criterios de Búsqueda.


The screenshot shows a table titled 'Resultado Búsqueda'. At the top, there are export options: 'Exportar a: CSV XLS XML' and an 'Exportar Archivo' button. The table has 12 records and is on page 1 of 2. The table columns are 'Registro', 'Operaciones', 'Código', 'Nombre', and 'Descripción'. The data rows are as follows:

Registro	Operaciones	Código	Nombre	Descripción
1	- ...	DS	Días de la Semana	Días de la Semana
2	- ...	ESP	Especialidades	
3	- ...	GD	Grupos de Datos	Grupos de Datos
4	- ...	GHU	Género Humano	
5	- ...	GRPUSR	Grupos de Usuarios	
6	- ...	M001	Maestro	Maestro de pruebas
7	- ...	MDS	Módulos del Sistema	
8	- ...	PG	Parámetros Generales	
9	- ...	TD	Tipos de Datos	Tipos de Datos
10	- ...	TF	Tipo de Foco	

El botón *Buscar* permite filtrar, por los Criterios de Búsqueda ingresados.

This screenshot is identical to the one above, showing the 'Resultado Búsqueda' table with 12 records and search filters.

Los controles de navegación permiten desplazarse entre número de páginas dependiendo de la cantidad de registros encontrados y se pueden cargar en páginas de 5, 10, 15, 50 y todos los registros encontrados.


Y el botón  el cual permite seleccionar campos adicionales a los que se muestran por defecto principalmente aquellos que tienen relación con el historial de cambios en la sección de *Resultado de Búsqueda*.



Order	Name	Visible	Sort
↓	Registro	<input checked="" type="checkbox"/>	
↓	Operaciones	<input checked="" type="checkbox"/>	
↓	Código	<input checked="" type="checkbox"/>	↑ ↓
↓	Nombre	<input checked="" type="checkbox"/>	↑ ↓
↓	Descripción	<input checked="" type="checkbox"/>	↑ ↓
↓	Creación	<input checked="" type="checkbox"/>	↑ ↓
↓	Creador por	<input checked="" type="checkbox"/>	↑ ↓
↓	Modificado	<input checked="" type="checkbox"/>	↑ ↓
↓	Modificado por	<input checked="" type="checkbox"/>	↑ ↓

Al dar clic en el botón *Exportar a*: se puede exportar los registros desplegados hacia un archivo seleccionable como: CSV, XLS o XML.

Exportar a: CSV XLS XML Exportar Archivo

Además, el botón *Buscar* permite filtrar, por parámetros de búsqueda ingresados. Los registros existentes que coinciden con los parámetros mencionados están agrupados por páginas. Una vez desplegada la información en los Resultados Búsqueda, en la columna Operaciones puede presionar el botón  donde:

La opción *Editar* permite modificar la información del registro a través del botón *Grabar* si la información es correcta y *Salir* para cancelar la acción.

03. Menús

Módulo:

Código:

Nombre:

Descripción:

Activo:

Icono:

El botón *Borrar* permite eliminar el registro a través del botón *Eliminar* y *Cancelar* para cancelar la acción.

03. Menús

Módulo:

Código:

Nombre:

Descripción:

Activo:

Icono:

El botón *Historial* despliega una ventana donde se visualiza los cambios realizados en cada registro los cuales también pueden ser exportados como: CSV, XLS o XML. Y el botón *Cancelar* para cerrar esta ventana.

Tabla = Personas; ID = 32

Exportar a: CSV XLS XLSX XML

2 Registro(s). Página 1 de 1

Registro	Fecha	Usuario	IP	Operación	Tabla	ID	Valor Anterior	Valor Nuevo
1	10/02/2019 18:49	root	190.12.11.161	[U] Actualizar	Personas	32	'cedula' = 'PG001'; 'descripcion' = null	'cedula' = 'PG 001'; 'descripcion' = ''
2	03/11/2018 23:15	root	192.168.2.24	[C] Crear	Personas	32		'id' = 32, 'rol' = null, 'email' = 'limon.ordonez@gmail.com', 'fecha' = '18/11/2014 00:00', 'activo' = 'S', 'cedula' = 'PG001', 'genero' = null, 'userid' = 'PG001', 'nombres' = 'Limon', 'apellidos' = 'Ordóñez', 'direccion' = 'OE13G S/N y S38B', 'ocupacion' = 'Domini', 'descripcion' = null

2 Registro(s). Página 1 de 1

Dependiendo de cada *Perfil* se han designado operaciones: Consultar (R), Actualizar (U), Eliminar (D) y Ver Historial (H) sobre los registros del sistema, a continuación una tabla resumen de las opciones de menú por cada grupo de usuario.

Grupo	Módulo	Menú	Submenú	R	U	D	R	H	
Super administradores	Seguridad	Mantenimiento	Maestros	SI	SI	SI	SI	SI	
			Parámetros	SI	SI	SI	SI	SI	
			Menús	SI	SI	SI	SI	SI	
			Submenús	SI	SI	SI	SI	SI	
			Personas	SI	SI	SI	SI	SI	
			Usuarios	SI	SI	SI	SI	SI	
			Perfiles	SI	SI	SI	SI	SI	
			Instituciones	SI	SI	SI	SI	SI	
			Materiales	SI	SI	SI	SI	SI	
	Reportes	Historial	SI	SI	SI	SI	SI		
	Transaccional	Mantenimiento	Profesionales	SI	SI	SI	SI	SI	
			Pacientes	SI	SI	SI	SI	SI	
			Transacciones	Atenciones	SI	SI	SI	SI	SI
				Órdenes de Laboratorio	SI	SI	SI	SI	SI
Médicos	Transaccional	Mantenimiento	Pacientes	SI	SI	SI	SI	SI	
			Horas	SI	SI	SI	SI	SI	
			Horarios	SI	SI	SI	SI	SI	
			Campos	SI	SI	SI	SI	SI	
			Profesionales	SI	SI	SI	SI	SI	
		Transacciones	Órdenes de Laboratorio	SI	SI	SI	SI	SI	
			Citas	SI	SI	SI	SI	SI	
Administradores	Transaccional	Mantenimiento	Profesionales	SI	SI	SI	SI	SI	
			Horas	SI	SI	SI	SI	NO	
			Horarios	SI	SI	SI	SI	SI	
		Transacciones	Citas	SI	SI	SI	SI	SI	
			Atenciones	SI	SI	SI	SI	SI	
			Órdenes de Laboratorio	SI	SI	SI	SI	SI	

Pacientes	transaccional	Transacciones	Citas	SI	SI	NO	SI	SI
			Atenciones	NO	NO	NO	SI	SI
Recepcionistas	Transaccional	Mantenimiento	Horas	NO	NO	NO	SI	SI
			Horarios	NO	NO	NO	SI	SI
			Pacientes	SI	SI	NO	SI	SI
			Profesionales	NO	NO	NO	SI	NO
		Transacciones	Citas	SI	SI	NO	SI	SI
			Órdenes de Laboratorio	NO	SI	NO	SI	SI
			Atenciones	NO	NO	NO	SI	SI