



INSTITUTO SUPERIOR TECNOLÓGICO

“SAN GABRIEL”

CARRERA:

INFORMATICA MENCION ANALISIS DE SISTEMAS

TRABAJO DE INVESTIGACION

**PREVIA A LA OBTENCION DEL TÍTULO DE TECNÓLOGO EN
INFORMÁTICA MENCIÓN ANALISIS DE SISTEMAS**

TITULO:

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL REGISTRO DE
HISTORIAS CLÍNICAS DE LOS PACIENTES DE LA CLÍNICA LA DOLOROSA,
DESARROLLADA EN JAVA Y GESTOR DE BASE DE DATOS MYSQL EN EL
PERIODO 2018”**

AUTOR:

PABLO RENATO GALARZA VALLEJO

RIOBAMBA – ECUADOR

2019

CERTIFICACIÓN

Certifico que el Sr. **PABLO RENATO GALARZA VALLEJO**, con el N° de Cédula 0604338269 ha elaborado bajo mi Asesoría el Proyecto de Investigación titulado:

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL REGISTRO DE HISTORIAS CLÍNICAS DE LOS PACIENTES DE LA CLÍNICA LA DOLOROSA, DESARROLLADA EN JAVA Y GESTOR DE BASE DE DATOS MYSQL EN EL PERIODO 2018”

Por tanto, autorizo la presentación para la calificación respectiva.

Ing. William Adriano

“El presente Proyecto de Investigación constituye un requisito previo para la obtención del Título de **Tecnólogo en Informática mención Análisis de Sistema**”



“Yo, **PABLO RENATO GALARZA VALLEJO** con N° de Cédula 0604338269 declaro que la investigación es absolutamente original, autentica, personal y los resultados y conclusiones a los que se han llegado es de mi absoluta responsabilidad.”

Pablo Renato Galarza Vallejo

INSTITUTO SUPERIOR TECNOLÓGICO

“SAN GABRIEL”

ESPECIALIDAD ANÁLISIS Y PROGRAMACIÓN EN SISTEMAS

TESIS DE GRADO

PREVIA A LA OBTENCION DEL TÍTULO DE: TECNÓLOGO EN INFORMÁTICA

TITULO:

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL REGISTRO DE HISTORIAS CLÍNICAS DE LOS PACIENTES DE LA CLÍNICA LA DOLOROSA, DESARROLLADA EN JAVA Y GESTOR DE BASE DE DATOS MYSQL EN EL PERIODO 2018”

APROBACIÓN DE LA TESIS:

ASESOR DE TESIS DE GRADO

PRESIDENTE DEL TRIBUNAL

MIEMBRO DEL TRIBUNAL

MIEMBRO DEL TRIBUNAL

FIRMAS DE RESPONSABILIDAD

NOMBRES	FECHA	FIRMAS
Ing. William Adriano Director de Tesis		
Miembro del Tribunal		
Miembro del Tribunal		
Miembro del Tribunal		

Nota de Tesis:

DEDICATORIA

A Dios.

Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

De igual forma dedico, mi tesis a mi madre Mercedes Vallejo por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

De igual forma dedico de manera especial a José María Vallejo que me impulsaba y aconsejaba guiándome hacia el sendero del bien.

Mi familia en general, porque me han brindado su apoyo incondicional y por compartir conmigo buenos y malos momentos.

Pablo Galarza

AGRADECIMIENTO

En primer lugar a Dios por haberme guiado por el camino de la felicidad hasta ahora; en segundo lugar a cada uno de los que son parte de mi familia

A mi madre, ya que sin el apoyo incondicional de ella no estaría donde estoy hoy en día, gracias a sus consejos su apoyo su perseverancia y sobre todo su amor.

A mis profesores a quienes les debo gran parte de mis conocimientos, gracias a su paciencia y enseñanza y finalmente un eterno agradecimiento a esta prestigiosa institución la cual abrió sus puertas, preparándonos para un futuro competitivo y formándonos como personas de bien.

Pablo Galarza

ABREVIATURAS

API: Application Program Interface

DAO: data access object

BEAN: Controlador de la vista, gestiona eventos emitidos por parte de la interfaz

HTML: Hypertext Markup Language

ASP: Active Server Pages

CSS: Cascading Styles Sheets

DLL: Dynamic Link Library

IDC: Internet Database Connector

ISAPI: Internet Server Application Program Interface

JSP: Java Server Pages

XAMPP: Apache Mariadb Php Perl

JRE: Java Runtime Environment

PDF: Portable Document Format

API : Application Programming Interface

CSV: Comma-Separated Values

XML: Xtensible Markup Language

GNU: General Public License

GLOSARIO DE TÉRMINOS

- **Plugin:** es un complemento que se relaciona con una aplicación o programa para agregarle una función nueva y generalmente muy específica. Este complemento adicional es ejecutado por la aplicación principal e interactúan por medio de la interfaz de programación de aplicaciones.
- **Xampp:** Es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl.
- **Java:** Es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.
- **NetBeans:** Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java.
- **MySQL:** Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual
- **PrimeFaces:** es una biblioteca de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web.
- **JasperReports:** es una biblioteca de creación de informes que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a ficheros PDF, HTML, XLS, CSV y XML.

Contenido

1	CAPÍTULO I	13
	MARCO REFERENCIAL	13
1.1	Formulación General Del Proyecto De Tesis	19
1.1.1	Antecedentes	19
1.1.2	Definición Del Problema	20
1.1.3	Justificación	20
1.1.4	Objetivo	21
2	CAPÍTULO II	21
	MARCO TEÓRICO	21
2.1.1	Conceptos de sistemas	23
2.1.2	Antecedentes del sistema	23
2.1.3	Importancia de los sistemas	24
2.2	Lenguajes De Programación	25
2.2.1	Java	25
2.2.2	Primeface	38
2.2.3	JasperReports	40
2.2.4	JavaBean	44
2.3	Entornos De Desarrollo Integrado IDE	48
2.3.1	Netbeans	48
2.4	Servidor de Base de Datos	51
2.4.1	MySql	51
3	CAPITULO III	23
	ANALISIS Y DISEÑO	23
3.1	Recopilación De Información	57
3.2	Descripción de la Clínica	57
3.2.1	Misión	57
3.2.2	Visión	57
3.3	Tipos De Investigación	58
3.3.1	Recursos Técnicos	58
3.3.2	Recursos Materiales	60
3.3.3	Factibilidad Operativa	60
3.4	Análisis De Los Requerimientos	60
3.4.1	Casos de uso	64
3.5	Diseño	69

3.5.1	Diseño conceptual	69
3.5.2	Modelo Relacional	71
3.5.3	Diccionario de Datos	71
4	CAPÍTULO IV	66
	IMPLEMENTACIÓN DE LA APLICACIÓN WEB	66
4.1	Configuración de las Herramientas de Desarrollo	82
4.2	Arquitectura del Sistema	83
4.3	Implementación de la Aplicación	83
4.3.1	Definición de Módulos de Implementación.....	84
4.3.2	Desarrollo de la Aplicación Implementación	84
4.3.3	Implantación de la Aplicación.....	85
4.4	Pruebas de la Aplicación	85
5	CAPÍTULO V	70
	CONCLUSIONES Y RECOMENDACIONES	70
5.1	Conclusiones.....	94
5.2	Recomendaciones.....	95

INDICE DE TABLAS

Tabla 1 Versiones del IDE Netbeans	49
Tabla 2 Frameworks, Tecnologías y Lenguajes incorporados en Netbeans	50
Tabla 3 HARDWARE	59
Tabla 4 Recurso Técnico Software	59
Tabla 5 Recurso Humano.....	59
Tabla 6 Recursos Materiales	60
Tabla 7 Control Usuario.....	62
Tabla 8 Control de Medico	62
Tabla 9 Control Usuario.....	63
Tabla 10 Descripción de los casos de uso Administrador, Medico	65
Tabla 11 Caso de Uso de Agregar Eliminar Modificar	66
Tabla 12 Caso de Uso de Historia Clínica	68
Tabla 13 Descripción de la Tabla Provincia.....	72
Tabla 14 Descripción de la Tabla Cantón	72
Tabla 15 Descripción de la Tabla Parroquia	72
Tabla 16 Descripción de la Tabla Antecedente	73
Tabla 17 Descripción de la Tabla Accidente.....	73
Tabla 18 Descripción de la Tabla Revisión.....	73
Tabla 19 Descripción de la Tabla Motivo	74
Tabla 20 Descripción de la Tabla Emergencia.....	74
Tabla 21 Descripción de la Tabla Registro	75
Tabla 22 Descripción de la Tabla APF (Antecedentes Personales Familiares)	76
Tabla 23 Descripción de la Tabla AVIEQ (Accidente, Violencia, Intoxicación, envenenamiento o Quemadura)	77
Tabla 24 Descripción de la Tabla EARS (Enfermedad Actual y Revisión de Sistemas).....	77
Tabla 25 Descripción de la Tabla Inicio de atención	77
Tabla 26 Descripción de la Tabla Consulta.....	78
Tabla 27 Descripción de la Tabla Emergencia.....	78
Tabla 28 Descripción de la Tabla Usuario	79
Tabla 29 Descripción de la Tabla Perfil	79
Tabla 30 Descripción de la Tabla Menú	80
Tabla 31 Características	86
Tabla 32 Tipos de Medida	87
Tabla 33 Identidad	87
Tabla 34 Contenido.....	88
Tabla 35 Navegación	88
Tabla 36 Utilidad	89
Tabla 37 Retroalimentación	89
Tabla 38 Test Usabilidad	90
Tabla 39 Indicador	91
Tabla 40 Indicadores de Evaluación	92

INDICE DE ILUSTRACIONES

Ilustración 1 Lenguaje de Programación java	25
Ilustración 2 Definición de una Clase.	28
Ilustración 3 Variables Auxiliares.....	29
Ilustración 4 Métodos	29
Ilustración 5 Constructores	30
Ilustración 6 Superclase	30
Ilustración 7 Paquetes	31
Ilustración 8 Variables locales	31
Ilustración 9 Método de la super-clase.....	32
Ilustración 10 Constructores	32
Ilustración 11 Clase PrimitivaGeometrica	33
Ilustración 12 Constructor sin parámetros.....	33
Ilustración 13 Manipular el objeto	34
Ilustración 14 Referencias.....	35
Ilustración 15 referencia Object	35
Ilustración 16 Claves	37
Ilustración 17 Primefaces.....	38
Ilustración 18 JasperReports	40
Ilustración 19 Secuencia	41
Ilustración 20 Estructura jasper.....	43
Ilustración 21 Java Bean	44
Ilustración 22 Propiedades.....	46
Ilustración 23 Propiedad Booleana	46
Ilustración 24 Propiedades Indexadas	47
Ilustración 25 Página de Inicio del Entorno de Desarrollo Integrado NETBEANS	48
4Ilustración 26 Sistema de Gestión de Base de Datos MySQL.....	51
Ilustración 27 Diagrama de caso de uso administrador	64
Ilustración 28 Diagrama de caso de uso del medico.....	65
Ilustración 29 Diagrama de caso de uso del usuario.....	67
Ilustración 30 Diseño conceptual 1	69
Ilustración 31 Diseño Conceptual 2	70
Ilustración 32 Modelo Relacional	71
Ilustración 33 Entorno de Desarrollo Netbeans.....	82
Ilustración 34 Arquitectura del Aplicativo	83
Ilustración 35 Pantalla Inicial de la Aplicación Web	84
Ilustración 36 Análisis	90
Ilustración 37 Registro de Historias Clínicas	91

Introducción

La Clínica “La Dolorosa” es un centro médico privado que se auto sustenta ofreciendo atención médica inmediata, esta casa de salud no posee un sistema automatizado de registro de historias clínicas que agilice el ingreso de información de los pacientes, lo que conlleva que se realice de forma manual, convirtiéndose este método de almacenamiento de información de los pacientes en un problema, pues genera pérdida de tiempo y acumulación de documentos físicos en los archivos, haciendo de esta actividad tediosa, y creando una desorganización total. De igual manera al momento de solicitar la historia clínica de los pacientes para diagnosticar el cuadro clínico que presentan, se convierte en un caos, ya que deben buscar en cada archivo la documentación necesaria, lo que perjudica la pronta atención a los pacientes.

Ante estos requerimientos se propone el diseño e implementación de un sistema informático con el que se automatiza el registro de las historias clínicas a través de una base de datos en donde además de ingresar la información de los pacientes atendidos en el año 2018 permite el registro de admisión y de emergencia de pacientes, lo que contribuye a una mejor organización de la información de la clínica, a la reducción de espacio físico, y a tener mayor seguridad en los datos existentes evitando pérdida de información.

Lo que motiva la realización de este trabajo de investigación fue una estrecha amistad mantenida con el personal de la casa de salud, a través de la cual se conoce las falencias que posee el área de recepción a la hora de registrar a los pacientes, y al poseer el conocimiento para solucionar dicho problema, se propone el diseño e implementación de un sistema informático que automatice el área de recepción de la clínica.

Por lo tanto, el objetivo de este trabajo de investigación propone un sistema para el registro de historias clínicas de los pacientes de la clínica La Dolorosa, desarrollada en java y gestor de base de datos MYSQL.

Cabe indicar que este trabajo de investigación consta de 5 capítulos los cuales se describen a continuación:

El capítulo I abarca el Marco Referencial, que constituye el sustento del trabajo de investigación porque contiene los antecedentes, definición del problema, justificación, el objetivo general y específico que son el eje medular para la tesis.

En el capítulo II se hace mención al Marco Teórico, en él se da a conocer el sustento teórico - práctico tanto del Lenguaje de Programación Java, el cual utiliza un entorno de desarrollo integrado llamado Netbeans que facilita la creación de sistemas web, como del sistema de gestor de base de datos MYSQL que es el encargado del almacenamiento de la información apropiado para sistemas web dinámicos.

El capítulo III, agrupa el análisis y diseño del sistema, en donde se presenta la recopilación de necesidades como el ingreso de admisión, referencias familiares, exámenes clínicos y generación de reportes de los mismos, con los que se diseña la base de datos con su respectiva interfaz de usuario, desarrollada en el Lenguaje de Programación Java.

El capítulo IV, hace referencia a la implementación del sistema informático en la clínica “La Dolorosa”, en donde se detalla la arquitectura del sistema, las herramientas utilizadas como son el servidor web apache, servidor de base de datos MYSQL, el Lenguaje de Programación JAVA y las pruebas necesarias de validación de número de cédula, códigos de diseño, redundancia de información, formatos de las historias clínicas y reportes generados.

El capítulo V, contiene las conclusiones y recomendaciones que se llegaron luego del desarrollo del trabajo de investigación, así como también el manual de usuario donde se especifica cada una de las funciones que el sistema realiza, junto al manual técnico, en donde se indica las herramientas que se utilizaron para la creación y funcionamiento del sistema.

Resumen

El trabajo de investigación titulado diseño e implementación de un sistema informático para el registro de historias clínicas, se implementó en la Clínica “La Dolorosa” de la ciudad de Riobamba provincia de Chimborazo, durante el período 2018. El diseño e implementación del sistema parte de la identificación de las necesidades de la clínica a través de la aplicación de entrevistas, una vez identificado el problema se propone automatizar el registro de historias clínicas, para lo cual se utilizó el framework jsf de JAVA, el mismo que sirvió para diseñar formularios, contenidos, página de acceso a usuarios, validación de los números de identificación e ingreso de exámenes clínicos. Para el almacenamiento de información se utilizó el sistema gestor de base de datos MYSQL que facilitó la elaboración de la base de datos conformada por tres módulos el general, el de reportes y el de emergencia. Con la ayuda de la herramienta Netbeans que es un entorno de desarrollo integrado se acopló el diseño realizado en framework jsf y la base de datos MYSQL, logrando obtener un sistema funcional y operativo. Los resultados obtenidos con la implementación del sistema informático fueron: reporte individual de la admisión de pacientes, y su reporte general por fecha, mes o año de ingreso; reporte de consultas médicas de pacientes generados por el número de cédula, y reporte general de las mismas realizadas por día, mes o año; así como también ofrece reportes de pacientes ingresados de emergencia de los cuales no se conoce su identificación, y reportes de historias clínicas con el diagnóstico de pacientes en forma personalizada.

Summary

The research titled “Design and implementation of a computer system for the registration of medical records”, was implemented in the clinic "La Dolorosa" of the city Riobamba, province of Chimborazo, during the period 2018. The design and implementation of the system starts with the identification of the needs of the clinic through the application of surveys. Once the problem was identified, it is proposed to automate the registration of medical records, for which the JAVA jfs framework was used. This served to design: forms, contents, user access page, validation of identification numbers and entry of clinical exams. For the storage of information, the MYSQL database management system was used, which facilitated the elaboration of the database consisting of three modules; general, reports and emergencies. With the help of the Netbeans tool, which is an integrated development environment, the design made in framework jsf and the MYSQL database was coupled, obtaining a functional and operational system. The results obtained with the implementation of the system were: individual report of the admission of patients and their general report by date, month or year of admission, also the report of medical consultations of patients generated by the ID number and general report of the same made by day, month or year; as well as it offers reports of emergency admitted patients whose identification and reports of medical records with a personalized diagnosis of patients are unknown.

CAPÍTULO I

MARCO REFERENCIAL

1.1 Formulación General Del Proyecto De Tesis

1.1.1 Antecedentes

La Clínica La Dolorosa se encuentra ubicado en la provincia de Chimborazo cantón Riobamba barrio “La Joya”. Tiene una trayectoria formal y bien definida en cuanto a atención hacia pacientes desde el año 2007, desde que brinda sus servicios el volumen de información de los pacientes siempre ha sido extenso y algo tedioso, este proceso es manejado por el personal del centro médico además se encuentra en constante crecimiento lo cual hace cada vez, más complejo el seguimiento, manipulación y administración de la información requerida.

La Clínica desde que abrió las puertas al público a trabajado con un sistema manual tomando información general y específica de cada paciente, asignando una numeración para cada paciente, y así no exista confusión o se pierda información en el momento que se necesite de la ficha, además su almacenamiento se encuentra por carpetas, la ubicación física está situada en un cuarto donde solo puede ingresar personal autorizado estas historias clínicas reposan dentro de archiveros completamente llenos.

El método de registro manual de información ha sido utilizado por diferentes centros de salud con algunas variaciones sin embargo en la actualidad este método cada vez está siendo remplazado adoptando nuevas formas de registro sabiendo esto La Clínica la Dolorosa está interesada en innovar y mejorar el tratamiento de la documentación pensando así en un sistema que remplace el actual que ha quedado obsoleto.

El sistema de registro propuesto será oportuna para las personas que laboran en la clínica, brindándoles una herramienta completa que les facilite el seguimiento de sus pacientes mediante MYSQL que es un sistema de gestión de bases de datos relacional desarrollado bajo licencia y está considerada como la base datos open source más popular del mundo y JAVA un lenguaje de programación y una plataforma informática.

1.1.2 Definición Del Problema

Como un sistema de registro de historias clínicas podría mejorar el tratamiento y organización de la información. La Clínica “La Dolorosa” requiere el desarrollo de un Sistema de registro a la medida que permitirá agilizar los procesos que son llevados a cabo en cada una de las áreas donde se identificaron los problemas, de tal forma que se elimina o minimiza la utilización de herramientas manuales como papel y esfero para el desarrollo de informes que son realizados por los empleados de la institución y en este sentido, hacer uso de solo un software que permita que dichos informes deseados por la institución puedan ser generados automáticamente, reduciendo así el tiempo y uso de herramientas no óptimas que actualmente son empleadas en la elaboración de registros en la institución; tomando en cuenta, que el sistema informático realizará.

Además la información estará segura y aislada de pérdida, daño, hurto y otras acciones que perjudiquen la seguridad de ésta, el sistema administrará a través de una base de datos, el registro, modificación y actualización de información. En este sentido, podrá aprovechar eficientemente el recurso informático con el que cuente, al redistribuirlo eficientemente y automatizar los procesos a través de la implementación del Sistema Informático.

1.1.3 Justificación

Un sistema de registro es una necesidad ya que suprime técnicas y procesos obsoletos que pueden llegar a retrasar determinadas actividades en una organización particular haciéndola competitivas frente a los distintos problemas y complicaciones que puede haber en el área de la salud.

Desde hace muchos años, las distintas organizaciones y empresas, en búsqueda de ser competitivas Desde hace muchos años, las distintas organizaciones y empresas, en búsqueda de ser competitivas y debido a la continua evolución de la informática y todo lo relacionado con la tecnología, han ido incursionando en este mundo. La manera como se maneja la información ha ido cambiando y casi ha obligado o los directivos de toda institución a autorizar los procesos y a optimizar la forma en que se manipula la información. Tal el caso del Ministerio de Salud Pública que hace pocos años Implemento un sistema informático

único de información (Sistema informático para la gestión integral de salud – SIGIS) facilitando el proceso.

La Clínica La Dolorosa estando consiente de la situación y sabiendo que depende del cliente ya que ellos son los mejores jueces al hablar de prestigio y calidad del servicio, una mala atención es un grave problema de incumplimiento del derecho humano a la salud y su forma de vida el sistema de registro pretende colaborar en estos procedimientos, de forma que se pueda hacer un debido uso optimizando los recursos con los que cuenta la clínica debido a que este sistema de registro significaría un gran ahorro de tiempo y optimización del personal implicado.

Con un sistema de registro para las historias clínicas no es necesario el cuarto que se utiliza actualmente lo únicamente necesario sería un computador donde se albergaría toda la información ahorrando espacio además no existiría el inconveniente de que se pierda documentos necesarios siendo así cada vez más eficientes y eficaces.

El proyecto tiene la finalidad de contribuir en el ingreso del paciente teniendo toda la información requerida obteniendo así un registro que se encargaría de administrar y ordenar los datos de cada paciente haciendo mucho más rápida la búsqueda de algún dato en específico llegando así a tener herramienta de trabajo muy práctica.

1.1.4 Objetivo

Objetivo General

- Desarrollar un sistema para el registro de historias clínicas de los pacientes de la clínica la dolorosa, desarrollada en java y gestor de base de datos mysql

Objetivo Especifico

- Investigar las características que el lenguaje de programación java y mysql a la hora de desarrollar un sitio web
- Investigar y diseñar los procedimientos y formatos usados en la Clínica La Dolorosa para el manejo de registro del paciente y creación de historias clínicas
- Desarrollo e implementación de un sistema para el registro de historias clínicas de los pacientes de la clínica “La Dolorosa”

CAPÍTULO II
MARCO TEÓRICO

SISTEMA

3.1.1 Conceptos de sistemas

El concepto de sistema es muy general. Podemos definir un sistema como un conjunto de elementos estructurados e interconectados de una manera definida. En otras palabras, es un conjunto de elementos asociados a un conjunto de relaciones entre elementos. Entre otros conceptos se encuentran los siguientes:

“El sistema es un conjunto de elementos organizados que se encuentran integrados, que buscan alguna meta o metas comunes, operando para ello sobre datos o información sobre energía o materia u organismos en una referencia temporal para producir como salida información”.

“Un sistema de control estudia la conducta del sistema con el fin de regularla de un modo conveniente para su supervivencia. Una de sus características es que sus elementos deben ser lo suficientemente sensitivos y rápidos como para satisfacer los requerimientos para cada función del control”.

“Un sistema es un conjunto organizado de cosas o partes interactivas e interdependientes, que se relacionan formando un todo unitario y complejo”.

3.1.2 Antecedentes del sistema

La Teoría General de Sistemas (TGS), surgió con los trabajos del biólogo alemán Ludwin Von Bertalanffy, publicados entre 1950 y 1968. Los Teoría General de Sistemas no busca solucionar problemas o intentar soluciones prácticas, pero sí producir teoría y formulaciones conceptuales que puedan crear condiciones de aplicación en la realidad empírica.¹

¹ TEORÍA GENERAL DE SISTEMAS. Von Gigch, Jhon p. Año de Edición 1993, Pág. 65.

3.1.3 Importancia de los sistemas

La importancia de los sistemas es debido al propósito de su implementación y radica en los resultados obtenidos entre los cuales se pueden mencionar

Sistemas naturales y artificiales: los naturales abundan en la naturaleza, la ecología de la vida y los artificiales aparecen en una infinita variedad alrededor de nosotros.

Sistema social, hombre-máquina y mecánico: los integrados pueden considerarse como sistemas sociales puros, distinguiéndose de otros sistemas, objetivos y procesos; los hombres-maquinas en la actualidad, casi todos los hombres emplean equipo de una u otra clase en sus trabajos organizados; y por último los mecánicos deben obtener sus propias entradas y mantenerlas.

- Sistemas abiertos y cerrados: el sistema abierto es aquel que interactúa con su ambiente; el sistema cerrado es aquel donde el ambiente que rodea no cambia.
- Sistemas permanentes y temporales: los permanentes son aquellos que duran mucho más que las operaciones que ellos realizan; y los temporales están destinados a durar cierto período y luego desaparecen.
- Sistemas estables y no estables: los estables son aquellas cuyas propiedades y operaciones no varían de manera importante. Los no estables son aquellos cuyas propiedades y operaciones cambian constantemente.
- Sistemas adaptivos y no adaptivos: el adaptivo es aquel que reacciona con un ambiente de tal manera que mejora su funcionamiento. Los no adaptivos son aquellos que no reaccionan oportunamente ante un cambio.
- Sistemas de procesos o sistemas productivos: es aquel que procesa las entradas para generar las salidas útiles a la sociedad.

- Sistemas de control de inventario: es aquel que controla grandes cantidades de dinero y numerosas transacciones diarias.

3.2 Lenguajes De Programación

3.2.1 Java



Ilustración 1 Lenguaje de Programación java

Fuente: <https://revistadigital.inesem.es/informatica-y-tics/java-el-lenguaje-de-programacion-universal/>

3.2.1.1 Descripción

Java es un lenguaje de programación creado por Sun Microsystems, (empresa que posteriormente fue comprada por Oracle) para poder funcionar en distintos tipos de procesadores. Su sintaxis es muy parecida a la de C o C++, e incorpora como propias algunas características que en otros lenguajes son extensiones: gestión de hilos, ejecución remota, etc. El código Java, una vez compilado, puede llevarse sin modificación alguna sobre cualquier máquina, y ejecutarlo. Esto se debe a que el código se ejecuta sobre una máquina hipotética o virtual, la Java Virtual Machine, que se encarga de interpretar el código (ficheros compilados .class) y convertirlo a código particular de la CPU que se esté utilizando (siempre que se soporte dicha máquina virtual)

3.2.1.2 Concepto de clase y objeto

Son el elemento fundamental a la hora de hablar de programación orientada a objetos es el concepto de objeto en sí, así como el concepto abstracto de clase. Un objeto es un conjunto

de variables junto con los métodos relacionados con éstas. Contiene la información (las variables) y la forma de manipular la información (los métodos). Una clase es el prototipo que define las variables y métodos que va a emplear un determinado tipo de objeto, es la definición abstracta de lo que luego supone un objeto en memoria. Poniendo un símil fuera del mundo de la informática, la clase podría ser el concepto de coche, donde nos vienen a la memoria los parámetros que definen un coche (dimensiones, cilindrada, maletero, etc), y las operaciones que podemos hacer con un coche (acelerar, frenar, adelantar, estacionar). La idea abstracta de coche que tenemos es lo que equivaldría a la clase, y la representación concreta de coches concretos (por ejemplo, Peugeot 307, Renault Megane, Volkswagen Polo...) serían los objetos de tipo coche

3.2.1.3 Concepto de campo, método y constructor

Toda clase u objeto se compone internamente de constructores, campos y/o métodos. Veamos qué representa cada uno de estos conceptos: un campo es un elemento que contiene información relativa a la clase, y un método es un elemento que permite manipular la información de los campos. Por otra parte, un constructor es un elemento que permite reservar memoria para almacenar los campos y métodos de la clase, a la hora de crear un objeto de la misma.

3.2.1.4 Concepto de herencia y polimorfismo

Con la herencia podemos definir una clase a partir de otra que ya exista, de forma que la nueva clase tendrá todas las variables y métodos de la clase a partir de la que se crea, más las variables y métodos nuevos que necesite. A la clase base a partir de la cual se crea la nueva clase se le llama superclase. Por ejemplo, podríamos tener una clase genérica Animal, y heredamos de ella para formar clases más específicas, como Pato, Elefante, o León. Estas clases tendrían todo lo de la clase padre Animal, y además cada una podría tener sus propios elementos adicionales. Una característica derivada de la herencia es que, por ejemplo, si tenemos un método dibuja (Animal a), que se encarga de hacer un dibujo del animal que se le pasa como parámetro, podremos pasarle a este método como parámetro tanto un Animal como un Pato, Elefante, o cualquier otro subtipo directo o indirecto de Animal. Esto se conoce como polimorfismo.

3.2.1.5 Modificadores de acceso

Tanto las clases como sus elementos (constructores, campos y métodos) pueden verse modificados por lo que se suelen llamar modificadores de acceso, que indican hasta dónde es accesible el elemento que modifican. Tenemos tres tipos de modificadores:

- Privado: el elemento es accesible únicamente dentro de la clase en la que se encuentra.
- Protegido: el elemento es accesible desde la clase en la que se encuentra, y además desde las subclases que hereden de dicha clase.
- Público: el elemento es accesible desde cualquier clase.

3.2.1.6 Clases abstractas e interfaces

Mediante las clases abstractas y los interfaces podemos definir el esqueleto de una familia de clases, de forma que los subtipos de la clase abstracta o la interfaz implementen ese esqueleto para dicho subtipo concreto. Por ejemplo, volviendo con el ejemplo anterior, podemos definir en la clase Animal el método dibuja () y el método imprime (), y que Animal sea una clase abstracta o un interfaz.

Vemos la diferencia entre clase, clase abstracta e interfaz con este supuesto:

- En una clase, al definir Animal tendríamos que implementar el código de los métodos dibuja () e imprime (). Las subclases que hereden de Animal no tendrían por qué implementar los métodos, a no ser que quieran redefinirlos para adaptarlos a sus propias necesidades
- En una clase abstracta podríamos implementar los métodos que nos interese, dejando sin implementar los demás (dejándolos como métodos abstractos). Dichos métodos tendrían que implementarse en las clases hijas.
- En un interfaz no podemos implementar ningún método en la clase padre, y cada clase hija tiene que hacer sus propias implementaciones de los métodos. Además, las clases hijas podrían implementar otros interfaces.

3.2.1.7 Componentes de un programa Java

Clases

Para definir una clase se utiliza la palabra reservada `class`, seguida del nombre de la clase:

```
class MiClase
{
    ...
}
```

Ilustración 2 Definición de una Clase.

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

Es recomendable que los nombres de las clases sean sustantivos (ya que suelen representar entidades), pudiendo estar formados por varias palabras. La primera letra de cada palabra estará en mayúscula y el resto de letras en minúscula. Por ejemplo, `DatosUsuario`, `Cliente`, `GestorMensajes`. Cuando se trate de una clase encargada únicamente de agrupar un conjunto de recursos o de constantes, su nombre se escribirá en plural. Por ejemplo, `Recursos`, `MensajesError`

Campos y variables

Dentro de una clase, o de un método, podemos definir campos o variables, respectivamente, que pueden ser de tipos simples, o clases complejas, bien de la API de Java, bien que hayamos definido nosotros mismos, o bien que hayamos copiado de otro lugar.

Al igual que los nombres de las clases, suele ser conveniente utilizar sustantivos que describan el significado del campo, pudiendo estar formados también por varias palabras.

En este caso, la primera palabra comenzará por minúscula, y el resto por mayúscula. Por ejemplo, `apellidos`, `fechaNacimiento`, `numIteraciones`.

De forma excepcional, cuando se trate de variables auxiliares de corto alcance se puede poner como nombre las iniciales del tipo de datos correspondiente:

```
int i;
Vector v;
MiOtraClase moc;
```

Ilustración 3 Variables Auxiliares

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

Por otro lado, las constantes se declaran como final static, y sus nombres se escribirán totalmente en mayúsculas, separando las distintas palabras que los formen por caracteres de subrayado ('_'). Por ejemplo, ANCHO_VENTANA, MSG_ERROR_FICHERO.

Métodos

Los métodos o funciones se definen de forma similar a como se hacen en C: indicando el tipo de datos que devuelven, el nombre del método, y luego los argumentos entre paréntesis:

```
void imprimir(String mensaje)
{
    ... // Código del método
}

double sumar(double... numeros){
    //Número variable de argumentos
    //Se accede a ellos como a un vector:
    //numeros[0], numeros[1], ...
}

Vector insertarVector(Object elemento, int posicion)
{
    ... // Código del método
}
```

Ilustración 4 Métodos

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

Al igual que los campos, se escriben con la primera palabra en minúsculas y el resto comenzando por mayúsculas. En este caso normalmente utilizaremos verbos.

Constructores

Podemos interpretar los constructores como métodos que se llaman igual que la clase, y que se ejecutan con el operador new para reservar memoria para los objetos que se creen de dicha clase:

```

MiClase()
{
    ... // Código del constructor
}

MiClase(int valorA, Vector valorV)
{
    ... // Código de otro constructor
}

```

Ilustración 5 Constructores

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

No tenemos que preocuparnos de liberar la memoria del objeto al dejar de utilizarlo. Esto lo hace automáticamente el garbage collector. Aún así, podemos usar el método `finalize()` para liberar manualmente. Si estamos utilizando una clase que hereda de otra, y dentro del constructor de la subclase queremos llamar a un determinado constructor de la superclase, utilizaremos `super`. Si no se hace la llamada a `super`, por defecto la superclase se construirá con su constructor vacío. Si esta superclase no tuviese definido ningún constructor vacío, o bien quisiésemos utilizar otro constructor, podremos llamar a `super` proporcionando los parámetros correspondientes al constructor al que queramos llamar. Por ejemplo, si heredamos de `MiClase` y desde la subclase queremos utilizar el segundo constructor de la superclase, al comienzo del constructor haremos la siguiente llamada a `super`:

```

SubMiClase()
{
    super(0, new Vector());
    ... // Código de constructor subclase
}

```

Ilustración 6 Superclase

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

Paquetes

Las clases en Java se organizan (o pueden organizarse) en paquetes, de forma que cada paquete contenga un conjunto de clases. También puede haber subpaquetes especializados dentro de un paquete o subpaquete, formando así una jerarquía de paquetes, que después se plasma en el disco duro en una estructura de directorios y subdirectorios igual a la de paquetes y subpaquetes (cada clase irá en el directorio/subdirectorio correspondiente a su paquete/subpaquete). Cuando queremos indicar que una clase pertenece a un determinado paquete o subpaquete, se coloca al principio del fichero la palabra reservada `package` seguida por los paquetes/subpaquetes, separados por `'.'`:

```
package paq1.subpaq1;
...
class MiClase {
...

```

Ilustración 7 Paquetes

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

Si queremos desde otra clase utilizar una clase de un paquete o subpaquete determinado (diferente al de la clase en la que estamos), incluimos una sentencia import antes de la clase (y después de la línea package que pueda tener la clase, si la tiene), indicando qué paquete o subpaquete queremos importar:

Herencia

Cuando queremos que una clase herede de otra, se utiliza al declararla la palabra extends tras el nombre de la clase, para decir de qué clase se hereda. Para hacer que Pato herede de Animal:

Class Pato extends Animal

Con esto automáticamente Pato tomaría todo lo que tuviese Animal (aparte, Pato puede añadir sus características propias). Si Animal fuese una clase abstracta, Pato debería implementar los métodos abstractos que tuviese.

Punteros this y super

El puntero this apunta al objeto en el que nos encontramos. Se utiliza normalmente cuando hay variables locales con el mismo nombre que variables de instancia de nuestro objeto:

```
public class MiClase
{
    int i;
    public MiClase(int i)
    {
        this.i = i;          // i de la clase = parametro i
    }
}
```

Ilustración 8 Variables locales

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

También se suele utilizar para remarcar que se está accediendo a variables de instancia.

El puntero super se usa para acceder a un elemento en la clase padre. Si la clase Usuario tiene un método getPermisos, y una subclase UsuarioAdministrador sobrescribe dicho método, podríamos llamar al método de la super-clase con:

```
public class UsuarioAdministrador extends Usuario {
    public List<String> getPermisos() {
        List<String> permisos = super.getPermisos();
        permisos.add(PERMISO_ADMINISTRADOR);
        return permisos;
    }
}
```

Ilustración 9 Método de la super-clase

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

También podemos utilizar this y super como primera instrucción dentro de un constructor para invocar a otros constructores. Dado que toda clase en Java hereda de otra clase, siempre será necesario llamar a alguno de los constructores de la super-clase para que se construya la parte relativa a ella. Por lo tanto, si al comienzo del constructor no se especifica ninguna llamada a this o super, se considera que hay una llamada implícita al constructor sin parámetros de la super-clase (super()). Es decir, los dos constructores siguientes son equivalentes:

```
public Punto2D(int x, int y, String etiq) {
    // Existe una llamada implícita a super()

    this.x = x;
    this.y = y;
    this.etiq = etiq;
}

public Punto2D(int x, int y, String etiq) {
    super();

    this.x = x;
    this.y = y;
    this.etiq = etiq;
}
```

Ilustración 10 Constructores

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

Pero es posible que la super-clase no disponga de un constructor sin parámetros. En ese caso, si no hacemos una llamada explícita a super nos dará un error de compilación, ya que estará intentando llamar a un constructor inexistente de forma implícita. Es posible también, que aunque el constructor sin parámetros exista, nos interese llamar a otro constructor a la hora de construir la parte relativa a la super-clase. Imaginemos por ejemplo que la clase Punto2D anterior deriva de una clase PrimitivaGeometrica que almacena, como información común de

todas las primitivas, una etiqueta de texto, y ofrece un constructor que toma como parámetro dicha etiqueta. Podríamos utilizar dicho constructor desde la subclase de la siguiente forma:

```
public Punto2D(int x, int y, String etiq) {
    super(etiq);

    this.x = x;
    this.y = y;
}
```

Ilustración 11 Clase PrimitivaGeometrica

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

También puede ocurrir que en lugar de querer llamar directamente al constructor de la superclase nos interese basar nuestro constructor en otro de los constructores de nuestra misma clase. En tal caso llamaremos a `this` al comienzo de nuestro constructor, pasándole los parámetros correspondientes al constructor en el que queremos basarnos. Por ejemplo, podríamos definir un constructor sin parámetros de nuestra clase punto, que se base en el constructor anterior (más específico) para crear un punto con una serie de datos por defecto:

```
public Punto2D() {
    this(DEFAULT_X, DEFAULT_Y, DEFAULT_ETIQ);
}
```

Ilustración 12 Constructor sin parámetros

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

Es importante recalcar que las llamadas a `this` o `super` deben ser siempre la primera instrucción del constructor.

Clases útiles

Cuando se programa con Java, se dispone de antemano de un conjunto de clases ya implementadas. Estas clases (aparte de las que pueda hacer el usuario) forman parte del propio lenguaje (lo que se conoce como API (Application Programming Interface) de Java). En esta sección vamos a ver una serie de clases que conviene conocer ya que nos serán de gran utilidad para realizar nuestros programas:

Object

Esta es la clase base de todas las clases en Java, toda clase hereda en última instancia de la clase Object, por lo que los métodos que ofrece estarán disponibles en cualquier objeto Java, sea de la clase que sea. En Java es importante distinguir claramente entre lo que es una variable, y lo que es un objeto. Las variables simplemente son referencias a objetos, mientras que los objetos son las entidades instanciadas en memoria que podrán ser manipulados mediante las referencias que tenemos a ellos (mediante variable que apunten a ellos) dentro de nuestro programa. Cuando hacemos lo siguiente:

```
new MiClase()
```

Se está instanciando en memoria un nuevo objeto de clase MiClase y nos devuelve una referencia a dicho objeto. Nosotros deberemos guardarnos dicha referencia en alguna variable con el fin de poder acceder al objeto creado desde nuestro programa:

```
MiClase mc = new MiClase();
```

Es importante declarar la referencia del tipo adecuado (en este caso tipo MiClase) para manipular el objeto, ya que el tipo de la referencia será el que indicará al compilador las operaciones que podremos realizar con dicho objeto. El tipo de esta referencia podrá ser tanto el mismo tipo del objeto al que vayamos a apuntar, o bien el de cualquier clase de la que herede o interfaz que implemente nuestro objeto. Por ejemplo, si MiClase se define de la siguiente forma:

```
public class MiClase extends Thread implements List {  
    ...  
}
```

Ilustración 13 Manipular el objeto

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

Podremos hacer referencia a ella de diferentes formas:

```
MiClase mc = new MiClase();  
Thread t = new MiClase();  
List l = new MiClase();  
Object o = new MiClase();
```

Ilustración 14 Referencias

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

Esto es así ya que al heredar tanto de Thread como de Object, sabemos que el objeto tendrá todo lo que tienen estas clases más lo que añade MiClase, por lo que podrá comportarse como cualquiera de las clases anteriores. Lo mismo ocurre al implementar una interfaz, al forzar a que se implementen sus métodos podremos hacer referencia al objeto mediante la interfaz ya que sabemos que va a contener todos esos métodos. Siempre vamos a poder hacer esta asignación 'ascendente' a clases o interfaces de las que deriva nuestro objeto. Si hacemos referencia a un objeto MiClase mediante una referencia Object por ejemplo, sólo podremos acceder a los métodos de Object, aunque el objeto contenga métodos adicionales definidos en MiClase. Si conocemos que nuestro objeto es de tipo MiClase, y queremos poder utilizarlo como tal, podremos hacer una asignación 'descendente' aplicando una conversión cast al tipo concreto de objeto:

```
Object o = new MiClase();  
...  
MiClase mc = (MiClase) o;
```

Ilustración 15 referencia Object

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

Si resultase que nuestro objeto no es de la clase a la que hacemos cast, ni hereda de ella ni la implementa, esta llamada resultará en un `ClassCastException` indicando que no podemos hacer referencia a dicho objeto mediante esa interfaz debido a que el objeto no la cumple, y por lo tanto podrán no estar disponibles los métodos que se definen en ella. Una vez hemos visto la diferencia entre las variables (referencias) y objetos (entidades) vamos a ver como se hará la asignación y comparación de objetos.

Puesto que hemos dicho que las variables simplemente son referencias a objetos, la asignación estará copiando una referencia, no el objeto. Es decir, tanto la variable `mc1` como `mc2` apuntarán a un mismo objeto.

El método clone es un método de la clase Object que estará disponible para cualquier objeto Java, y nos devuelve un Object genérico, ya que al ser un método que puede servir para cualquier objeto nos debe devolver la copia de este tipo. De él tendremos que hacer una conversión cast a la clase de la que se trate como hemos visto en el ejemplo. Al hacer una copia con clone se copiarán los valores de todas las variables de instancia, pero si estas variables son referencias a objetos sólo se copiará la referencia, no el objeto. Es decir, no se hará una copia en profundidad. Si queremos hacer una copia en profundidad deberemos sobrescribir el método clone para hacer una copia de cada uno de estos objetos. Para copiar objetos también podríamos definir un constructor de copia, al que se le pase como parámetro el objeto original a copiar.

System

Esta clase nos ofrece una serie de métodos y campos útiles del sistema. Esta clase no se debe instanciar, todos estos métodos y campos son estáticos. Podemos encontrar los objetos que encapsulan la entrada, salida y salida de error estándar, así como métodos para redireccionarlas, que veremos con más detalle en el tema de entrada/salida. También nos permite acceder al gestor de seguridad instalado, como veremos en el tema sobre seguridad.

Clave	Contenido
file.separator	Separador entre directorios en la ruta de los ficheros. Por ejemplo "/" en UNIX.
java.class.path	Classpath de Java
java.class.version	Versión de las clases de Java
java.home	Directorio donde está instalado Java
java.vendor	Empresa desarrolladora de la implementación de la plataforma Java instalada
java.vendor.url	URL de la empresa
java.version	Versión de Java
line.separator	Separador de fin de líneas utilizado
os.arch	Arquitectura del sistema operativo
os.name	Nombre del sistema operativo
os.version	Versión del sistema operativo
path.separator	Separador entre los distintos elementos de una variable de entorno tipo PATH. Por ejemplo ":"
user.dir	Directorio actual
user.home	Directorio de inicio del usuario actual
user.name	Nombre de la cuenta del usuario actual

Ilustración 16 Claves

Fuente: <http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

Patrón DAO

Es bastante normal hacer aplicaciones que almacenan y recogen datos de una base de datos. Suele ser habitual, también, querer hacer nuestra aplicación lo más independiente posible de una base de datos concreta, de cómo se accede a los datos o incluso de si hay o no base de datos detrás. Nuestra aplicación debe conseguir los datos o ser capaz de guardarlos en algún sitio, pero no tiene por qué saber de dónde los está sacando o dónde se guardan.

Hay una forma de hacer esto que ha resultado bastante eficiente en el mundo JEE y de aplicaciones web, pero que es aplicable a cualquier tipo de aplicación que deba recoger datos de algún sitio y almacenarlos. Es lo que se conoce como patrón DAO (Data Access Object).

La idea de este patrón es sencilla. En primer lugar, debemos hacernos las clases que representan nuestros datos. Por ejemplo, podemos hacer una clase Persona con los datos de la persona y los métodos set() y get() correspondientes.

Luego hacemos una interface. Esta interface tiene que tener los métodos necesarios para obtener y almacenar Personas. Esta interface no debe tener nada que la relaciones con una base de datos ni cualquier otra cosa específica del medio de almacenamiento que vayamos a usar, es decir, ningún parámetro debería ser una Connection, ni un nombre de fichero, etc. Por ejemplo, puede ser algo así

¿Qué es Java Web Start y cómo se ejecuta?

El software de Java Web Start permite descargar y ejecutar aplicaciones Java desde la Web.

El software de Java Web Start:

- Permite activar las aplicaciones con un simple clic
- Garantiza que se está ejecutando la última versión de la aplicación
- Elimina complejos procedimientos de instalación o actualización

Obtención del software de Java Web Start

Java Web Start está incluido en Java Runtime Environment (JRE) desde la versión Java 5.0. Esto significa que al instalar Java, Java Web Start se instalará automáticamente. El software de Java Web Start se inicia automáticamente, cuando una aplicación Java que utiliza tecnología Java Web Start se descarga por primera vez. El software de Java Web Start almacena en caché toda la aplicación de forma local en la computadora. Por lo tanto, los siguientes inicios son casi instantáneos, ya que todos los recursos necesarios ya están disponibles localmente. Cada vez que inicie la aplicación, el componente de software de Java Web Start comprueba el sitio web de la aplicación para comprobar si existe una nueva versión y, si es así, la descarga e inicia automáticamente.

3.2.2 Primeface



Ilustración 17 Primefaces

Fuente: <https://www.albertcoronado.com/2013/11/26/portlets-liferay-con-jsf-framework-primefaces/>

Es una librería de componentes visuales de código abierto para el conjunto Java Server Faces 2.0 desarrollada y mantenida por Prime Technology. Su objetivo principal es ofrecer un conjunto de componentes para facilitar la creación y diseño de aplicaciones web. Los componentes de PrimeFaces cuentan con soporte nativo de Ajax, pero no se encuentra implícito, de tal manera que se tiene que especificar que componentes se deben actualizar al realizar una petición proporcionando así mayor control sobre los eventos. Cuenta también con un modulo adicional TouchFaces para el desarrollo de aplicaciones web para dispositivos móviles con navegadores basados en WebKit. Las principales características de PrimeFaces son:

- Soporte nativo de Ajax, incluyendo Push/Coment.
- Kit para crear aplicaciones web móviles.
- Es compatible con otras librerías de componentes como Jboss RichFaces.
- Uso de JavaScript no intrusivo.
- Es un proyecto open source, activo y estable.

Características

- Posee un gran set de componentes UI (HtmlEditor, Dialog, AutoComplete, Charts, DatePicker, DataTables, TreeComponent, etc).
- Tecnología Ajax basada an la API Ajax JSF 2.0
- Muy liviana
- Solo un archivo jar
- No se requiere configuración
- No tiene dependencias (salvo por supuesto JSF runtime 2.0 o 2.1)
- Soporta Ajax Push via websockets
- Mobile UI kit que permite la creación de aplicaciones web para dispositivos mobiles
- Soporta Skinning y posee más de 35 temas preconstruídos. Soporta el editor visial de temas.
- Documentación mantenida por la comunidad muy actualizada.

3.2.3 JasperReports



Ilustración 18 JasperReports

Fuente: <https://www.openwayclub.com/way4-news-for-clients/2018/4/20/jasperreport-server>

Es una potente herramienta en código abierto de generación de informes que permite generar información detallada en formato PDF, HTML, XLS, CSV o XML. Ha sido desarrollada completamente en Java, por lo que puede ser usada en una amplia variedad de aplicaciones Java para generar contenido dinámico.

JasperReports trabaja de forma muy similar a un compilador y un intérprete. Se siguen los siguientes pasos:

EL desarrollador diseña el reporte codificándolo en XML de acuerdo a las etiquetas y atributos definidos en `jasperreports.dtd`. En el XML, el desarrollador puede definir todo el formato del informe describiendo la posición del texto, gráficos, dibujos, realizar cálculos, etc,...

El archivo XML es compilado mediante el método `compileReport()`, localizado en la clase `net.sf.jasperreports.engine.JasperCompileManager`

Una vez obtenido el archivo `.jasper`, se necesitan datos dinámicos que cumplimenten el reporte. Por ello se acude al `JRDataSource`. Para rellenar el informe, se pueden usar los métodos `*fillReportXXX()*` de la clase `net.sf.jasperreports.engine.JasperFillManager`. Estos métodos reciben como parámetro el objeto del diseño del informe, o un fichero que representa tal objeto, en un formulario serializado, así como una conexión JDBC a la base de datos en donde se almacenan los datos que rellenarán el informe o un Bean que contenga estos datos.

Se obtiene un archivo `.print` exportable en varios formatos PDF, HTML, XML, XSL, CVS.

A continuación presentamos un modelo gráfico con la secuencia.

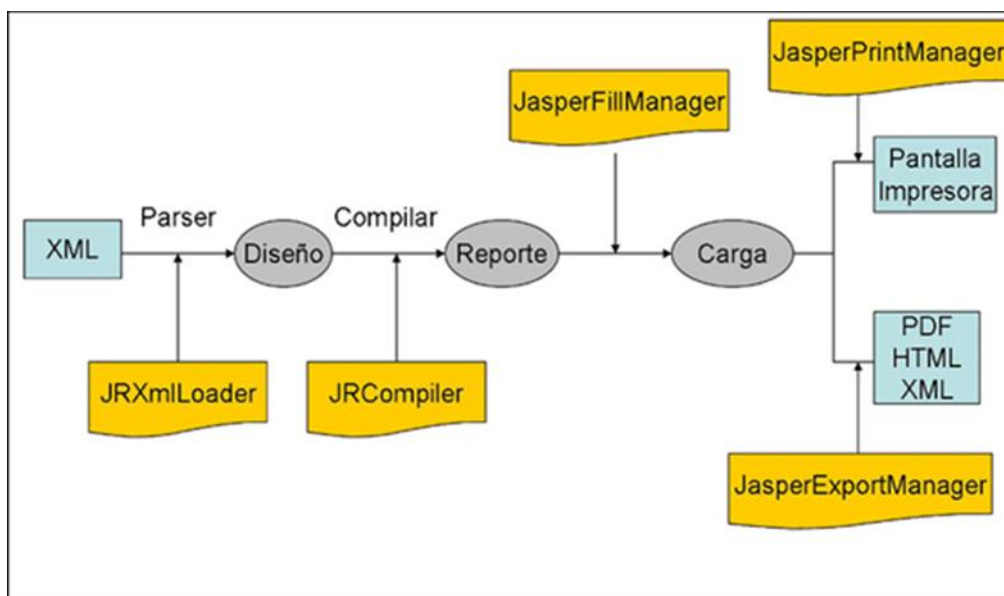


Ilustración 19 Secuencia

Fuente: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/237>

La creación del XML que da soporte a la creación del archivo suele ser compleja, especialmente para los desarrolladores con poca experiencia. Por ello, se utiliza con frecuencia algunas librerías de apoyo para la edición y exportación de contenidos.

Ejemplos de uso

Para desarrollar este ejemplo, lo primero que ha de hacerse es el diseño del informe, utilizando para ello la herramienta gráfica iReport. Se trata de un diseño muy sencillo en el que va a haber sólo dos campos de texto: uno que mostrará el título del informe y otro que simplemente contendrá un nombre.

El funcionamiento es bien sencillo. El usuario ha de introducir dos valores: la tarea a ejecutar y la ubicación del fichero a manipular. Si se sigue el proceso completo, es decir, si se parte del diseño de iReport sin compilar, cuya extensión es .jrxml, se debe ejecutar la aplicación con los siguientes argumentos:

Usar el jrxml para generar el diseño compilado jasper, mediante el método `compile`

Report ToFile:

Usar el jasper para generar el fichero jrprint, mediante el método `fillReportToFile`. En este caso, se utilizan tres argumentos: la ruta del fichero jasper, un map con los parámetros con los

que se rellenará el informe (el título y un nombre) y un datasource vacío (JREmptyDataSource), diseñado para que JasperReports prescinda de utilizar el datasource que todo informe de iReport obliga a tener. En este caso, prueba3.jrxml disponía de una conexión Hsql.

Usar el jrprint para generar el fichero final, en este caso PDF, mediante la función exportReportToPdfFile. Además, se envían dos parámetros: el título del informe (en la variable que hemos denominado ReportTitle) y un nombre (en la variable que hemos llamado Name).

Opcionalmente, si se desea enviar el informe a la impresora, también emplea el jrprint. Usa el método printReport y, como es obvio, no se genera ningún otro fichero.

```

print E:/eclipse/workspace/JasperConfluence/src/prueba3

package com.cassiopea.ejemplo;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import java.io.File;
import java.util.HashMap;
import java.util.Map;
import net.sf.jasperreports.engine.JREmptyDataSource;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperCompileManager;
import net.sf.jasperreports.engine.JasperExportManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperPrintManager;
import net.sf.jasperreports.engine.export.JExcelApiExporter;
import net.sf.jasperreports.engine.export.JRCsvExporter;
import net.sf.jasperreports.engine.export.JRXlsExporter;
import net.sf.jasperreports.engine.export.JRXlsExporterParameter;
import net.sf.jasperreports.engine.util.JRLoader;

/**
 * Clase en la que se ejecuta cada uno de los pasos de la generación de informes
 *
 * por separado pasándole como primer argumento la acción a realizar:
 *
 * compile | fill | print | pdf | xml | xmlembed | html | xls | csv
 *
 * y como segundo argumento el path absoluto del fichero del informe
 */

```

Ilustración 20 Estructura jasper

Fuente: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/237>

3.2.4 JavaBean



Ilustración 21 Java Bean

Fuente: <https://www.javatpoint.com/ejb-full-form>

Descripción

Un JavaBean o bean es un componente hecho en software que se puede reutilizar y que puede ser manipulado visualmente por una herramienta de programación en lenguaje Java.

Para ello, se define un interfaz para el momento del diseño (design time) que permite a la herramienta de programación o IDE, interrogar (query) al componente y conocer las propiedades (properties) que define y los tipos de sucesos (events) que puede generar en respuesta a diversas acciones.

Aunque los beans individuales pueden variar ampliamente en funcionalidad desde los más simples a los más complejos, todos ellos comparten las siguientes características:

Introspection: Permite analizar a la herramienta de programación o IDE como trabaja el bean

Customization: El programador puede alterar la apariencia y la conducta del bean.

Events: Informa al IDE de los sucesos que puede generar en respuesta a las acciones del usuario o del sistema, y también los sucesos que puede manejar.

Properties: Permite cambiar los valores de las propiedades del bean para personalizarlo (customization).

Persistence: Se puede guardar el estado de los beans que han sido personalizados por el programador, cambiando los valores de sus propiedades.

En general, un bean es una clase que obedece ciertas reglas:

Un bean tiene que tener un constructor por defecto (sin argumentos)

Un bean tiene que tener persistencia, es decir, implementar el interface Serializable.

Un bean tiene que tener introspección (introspection). Los IDE reconocen ciertas pautas de diseño, nombres de las funciones miembros o métodos y definiciones de las clases, que permiten a la herramienta de programación mirar dentro del bean y conocer sus propiedades y su conducta.

Propiedades

Una propiedad es un atributo del JavaBean que afecta a su apariencia o a su conducta. Por ejemplo, un botón puede tener las siguientes propiedades: el tamaño, la posición, el título, el color de fondo, el color del texto, si está o no habilitado, etc.

Las propiedades de un bean pueden examinarse y modificarse mediante métodos o función es miembro, que acceden a dicha propiedad, y pueden ser de dos tipos:

getter method: lee el valor de la propiedad

setter method: cambia el valor de la propiedad.

Un IDE que cumpla con las especificaciones de los JavaBeans sabe como analizar un bean y conocer sus propiedades. Además, crea una representación visual para cada uno de los tipos de propiedades, denominada editor de propiedades, para que el programador pueda modificarlas fácilmente en el momento del diseño.

Cuando un programador, coge un bean de la paleta de componentes y lo deposita en un panel, el IDE muestra el bean sobre el panel. Cuando seleccionamos el bean aparece una hoja de propiedades, que es una lista de las propiedades del bean, con sus editores asociados para cada una de ellas.

El IDE llama a los métodos o funciones miembro que empiezan por get, para mostrar en los editores los valores de las propiedades. Si el programador cambia el valor de una propiedad se llama a un método cuyo nombre empieza por set, para actualizar el valor de dicha propiedad y que puede o no afectar al aspecto visual del bean en el momento del diseño.

Las especificaciones JavaBeans definen un conjunto de convenciones (design patterns) que el IDE usa para inferir qué métodos corresponden a propiedades.

```
public void setNombrePropiedad(TipoPropiedad valor)
```

```
public TipoPropiedad getNombrePropiedad( )
```

Cuando el IDE carga un bean, usa el mecanismo denominado reflection para examinar todos los métodos, fijándose en aquellos que empiezan por set y get. El IDE añade las propiedades que encuentra a la hoja de propiedades para que el programador personalice el bean.

Propiedades simples

Una propiedad simple representa un único valor.

Ejemplo

```
//miembro de la clase que se usa para guardar el valor de la propiedad
private String nombre;
//métodos set y get de la propiedad denominada Nombre
public void setNombre(String nuevoNombre){
    nombre=nuevoNombre;
}
public String getNombre(){
    return nombre;
}
```

Ilustración 22 Propiedades

Fuente: <https://www.javatpoint.com/ejb-full-form>

En el caso de que dicha propiedad sea booleana se escribe

```
//miembro de la clase que se usa para guardar el valor de la propiedad
private boolean conectado=false;
//métodos set y get de la propiedad denominada Conectado
public void setConectado(boolean nuevoValor){
    conectado=nuevoValor;
}
public boolean isConectado(){
    return conectado;
}
```

Ilustración 23 Propiedad Booleana

Fuente: <https://www.javatpoint.com/ejb-full-form>

Propiedades indexadas

Una propiedad indexada representa un array de valores.

```
//miembro de la clase que se usa para guardar el valor de la propiedad
private int[] numeros={1,2,3,4};

//métodos set y get de la propiedad denominada Numeros, para el array completo
public void setNumeros(int[] nuevoValor){
    numeros=nuevoValor;
}
public int[] getNumeros(){
    return numeros;
}

//métodos get y set para un elemento de array
public void setNumeros(int indice, int nuevoValor){
    numeros[indice]=nuevoValor;
}
public int getNumeros(int indice){
    return numeros[indice];
}
```

Ilustración 24 Propiedades Indexadas

Fuente: <https://www.javatpoint.com/ejb-full-form>

Propiedades ligadas (bound)

Los objetos de una clase que tiene una propiedad ligada notifican a otros objetos (listeners) interesados, cuando el valor de dicha propiedad cambia, permitiendo a estos objetos realizar alguna acción. Cuando la propiedad cambia, se crea un objeto (event) que contiene información acerca de la propiedad (su nombre, el valor previo y el nuevo valor), y lo pasa a los otros objetos (listeners) interesados en el cambio.

Propiedades restringidas (constrained)

Una propiedad restringida es similar a una propiedad ligada salvo que los objetos (listeners) a los que se les notifica el cambio del valor de la propiedad tienen la opción de vetar (veto) cualquier cambio en el valor de dicha propiedad. (Este tipo de propiedad no se estudiará en este capítulo).

3.3 Entornos De Desarrollo Integrado IDE

3.3.1 Netbeans



Ilustración 25 Página de Inicio del Entorno de Desarrollo Integrado NETBEANS

Fuente: http://www.consultorjava.com/wp/wpcontent/uploads/2015/09/herramienta_desarrollo_netbeans.pdf

Descripción

Es un entorno de desarrollo integrado, que básicamente está compuesto por un editor de texto, un editor de diseño, un compilador y un depurador, es utilizado para el desarrollo de aplicaciones Web, Móviles y de Escritorio con distintos lenguajes de programación como son Java, C++, Ruby y PHP entre otros.

Es una herramienta de código abierto, es multiplataforma porque puede ser instalado en cualquier sistema operativo y hardware sin problemas, es multilenguaje porque en la instalación se puede escoger el idioma, además contiene servidores web y es fácil de instalarlo y utilizarlo.

IDE Netbeans y su influencia en el desarrollo de software

El Entorno de Desarrollo Integrado Netbeans influye positivamente en el proceso de Desarrollo de Software ya que permite organizar la información por etapas, comenzando por qué se puede ejecutar en cualquier Sistema Operativo debido a su característica multiplataforma, además posee un asistente para la conexión a distintas bases de datos, módulos de integración, librerías, plugins para la generación de reportes y estadísticas.

Netbeans es considerada una herramienta sin costo por ser de código abierto y gratuito, además porque no genera gastos de licencia.

Versiones

A continuación, se detallan las versiones del IDE Netbeans que estuvieron estables en el mercado:

VERSIÓN	FECHA DE LANZAMIENTO
Netbeans 8.0.1	Octubre 2014
Netbeans 7.4	Octubre 2013
Netbeans 7.0	Abril 2011
Netbeans 6.9.1	Agosto 2010
Netbeans 6.8	Diciembre 2009
Netbeans 6.7	Junio 2009
Netbeans 6.5	Noviembre 2008
Netbeans 5.5	Octubre 2006
Netbeans 4.1	Mayo 2005
Netbeans 3.6	Abril 2004
Netbeans 3.1	Diciembre 2000

Tabla 1 Versiones del IDE Netbeans

Fuente: http://www.consultorjava.com/wp/wp%20content/uploads/2015/09/herramienta_desarrollo_netbeans.pdf

Frameworks, tecnologías y lenguajes incorporados en el IDE Netbeans

Lenguaje de Programación	Tecnología	Framework
PHP	PHP 5.4	Zend Framework Symfony1 y 2 Framework, Nette Framework 2
JAVA	JAVA SE	
	JAVA ME 2	
		JSF, Struts 1.3,EJB, Spring3.2, JPA, Hibernate3.6,Web RESTful(JAX-RS), MAVEN
	JAVA FX 2.2	
	JAVA CARD 3 CONNECT	
CC++	C/C++	
HTML 5	HTML 5	

Tabla 2 Frameworks, Tecnologías y Lenguajes incorporados en Netbeans

Fuente: http://www.consultorjava.com/wp/wpcontent/uploads/2015/09/herramienta_desarrollo_netbeans.pdf

A más de los elementos mencionados anteriormente, también incluye Servidores de aplicaciones como son:

- GlassFish Server Open Source 4.x
- Apache Tomcat 7.x
- JBoss AS 7.x
- WebLogic 11g

También incluye Plugines como son:

- Inreport
- JasperReport
- Sublime Theme
- Ruby on Rails
- JavaScript

Además como se mencionó el IDE Netbeans cuenta con conexiones a diferentes Bases Datos, y un editor SQL que permite el acceso a:

- Mysql
- Oracle
- Java DB
- PostgreSQL

3.4 Servidor de Base de Datos

3.4.1 MySql



4Ilustración 26 Sistema de Gestión de Base de Datos MySQL

Fuente: <http://velozityweb.com/blog/php/como-crear-un-formulario-de-registro-de-usuarios-con-php-y-mysql/#sthash.3wJvrX2.dpbs>

Descripción

MySQL, el sistema de gestión de base de datos Open Source SQL más popular, es desarrollado, distribuido y soportado por Oracle Corporation.

MySQL es un sistema de gestión de bases de datos, una base de datos es una colección estructurada de datos. Puede ser cualquier cosa, desde una simple lista de compras hasta una galería de imágenes o la gran cantidad de información en una red corporativa. Para agregar, acceder y procesar datos almacenados en una base de datos de computadora, necesita un sistema de administración de bases de datos como MySQL Server. Dado que los ordenadores son muy buenos para manejar grandes cantidades de datos, los sistemas de gestión de bases de datos desempeñan un papel central en la informática, como utilidades independientes o como partes de otras aplicaciones.

Las bases de datos MySQL son relacionales, una base de datos relacional almacena los datos en tablas separadas en lugar de poner todos los datos en un gran almacén.

Las estructuras de la base de datos se organizan en archivos físicos optimizados para la velocidad. El modelo lógico, con objetos como bases de datos, tablas, vistas, filas y columnas, ofrece un entorno de programación flexible. Se configura reglas que rigen las relaciones entre los diferentes campos de datos, como uno-a-uno, uno-a- muchos, único, obligatorio u opcional, y “punteros” entre las diferentes tablas. La base de datos hace cumplir estas reglas, por lo que, con una base de datos bien diseñada, su aplicación nunca ve datos inconsistentes, duplicados, huérfanos, obsoletos o que faltan.

La parte SQL de “MySQL” significa “Structured Query Language “. SQL es el lenguaje estandarizado más común utilizado para acceder a las bases de datos. Dependiendo de su entorno de programación, puede ingresar SQL directamente, incrustar instrucciones SQL en código escrito en otro idioma o utilizar una API específica de idioma que oculte la sintaxis SQL.

SQL está definido por el estándar ANSI / ISO SQL. El estándar SQL ha estado evolucionando desde 1986 y existen varias versiones. En este manual, “SQL-92”, se refiere al estándar publicado en 1992, “SQL: 1999”, se refiere al estándar publicado en 1999, y “SQL: 2003”, se refiere a la versión actual del estándar. Se utiliza la frase “el estándar SQL”, para referirse a la versión actual del SQL Standard en cualquier momento.

MySQL Server fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha sido utilizado con éxito en entornos de producción altamente exigentes durante varios años. Aunque en constante desarrollo, MySQL

Server ofrece hoy un amplio y útil conjunto de funciones. Su conectividad, velocidad y seguridad hacen que MySQL Server sea muy adecuado para acceder a bases de datos en Internet.

El software MySQL ofrece un servidor de base de datos SQL (Structured Query Language) muy rápido, multihilo, multiusuario y robusto. MySQL Server está diseñado para sistemas de producción de carga crítica y de misión crítica, así como para incrustar en software de implementación masiva.

El software de MySQL es Dual Licensed, es decir los usuarios pueden optar por utilizar el software MySQL como un producto de código abierto bajo los términos de la GNU General Public License o pueden adquirir una licencia comercial estándar de Oracle.

Características de MySQL

Internos y portabilidad

Escrito en C y C ++

Funciona en varias plataformas diferentes

Proporciona motores de almacenamiento transaccionales y no transaccionales

Diseñado para que sea relativamente fácil agregar otros motores de almacenamiento.

Esto es útil si desea proporcionar una interfaz SQL para una base de datos interna

Tipos de datos

Muchos tipos de datos: firmados / enteros sin signo de 1, 2, 3, 4, y 8 bytes de largo, FLOAT, DOUBLE, CHAR, VARCHAR, BINARY, VARBINARY, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM, y OpenGIS tipos espaciales.

Longitud fija y longitud variable.

Declaraciones y funciones

Completo operador y función de apoyo en la SELECT lista y la WHERE cláusula de las consultas

Soporte completo para SQL GROUP BY y ORDER BY cláusulas. El apoyo a las funciones de grupo (COUNT(), AVG(), STD(), SUM(), MAX(), MIN(), y GROUP_CONCAT()).

Soporte para DELETE, INSERT, REPLACE, y UPDATE para devolver el número de filas que se han cambiado (afectados), o para devolver el número de filas coincidentes en lugar mediante el establecimiento de una bandera cuando se conecta al servidor.

Puede consultar tablas de diferentes bases de datos en la misma sentencia.

Seguridad

Un sistema de privilegios y contraseñas que es muy flexible y seguro, que permite la verificación basada en host.

Seguridad de contraseña por cifrado de todo el tráfico de contraseñas cuando se conecta a un servidor.

Escalabilidad y límites

Soporte para bases de datos grandes. Utilizamos MySQL Server con bases de datos que contienen 50 millones de registros. También sabemos de usuarios que usan MySQL Server con 200.000 tablas y alrededor de 5.000.000.000 de filas.

Compatibilidad con hasta 64 índices por tabla. Cada índice puede consistir de 1 a 16 columnas o partes de columnas. El ancho máximo del índice para las InnoDBtablas es de 767 bytes o 3072 bytes. El ancho máximo del índice para las MyISAMtablas es 1000 bytes. Un índice puede usar un prefijo de una columna para CHAR, VARCHAR, BLOB, o TEXT tipos de columna.

Conectividad

Los clientes pueden conectarse mediante sockets TCP / IP en cualquier plataforma.

En sistemas Windows, los clientes pueden conectarse utilizando canalizaciones con nombre si el servidor se inicia con la—enable-named-pipe opción. Los servidores Windows también admiten conexiones de memoria compartida si se inicia con la— shared-memory opción. Los clientes pueden conectarse a través de la memoria compartida mediante la— protocol=memory opción.

En sistemas Unix, los clientes pueden conectarse utilizando archivos de socket de dominio Unix.

Los programas cliente de MySQL se pueden escribir en muchos idiomas. Una biblioteca de cliente escrita en C está disponible para clientes escritos en C o C ++ o para cualquier lenguaje que proporciona enlaces C.

APIs para C, C ++, Eiffel, Java, Perl, PHP, Python, Ruby y Tcl están disponibles, permitiendo que los clientes de MySQL sean escritos en muchos idiomas.

La interfaz Connector / ODBC (MyODBC) proporciona soporte de MySQL para programas cliente que utilizan conexiones ODBC (Open Database Connectivity). Por ejemplo, puede utilizar MS Access para conectarse a su servidor MySQL. Los clientes se pueden ejecutar en Windows o Unix. Conector / fuente ODBC está disponible. Todas las funciones de ODBC 2.5 son compatibles, al igual que muchos otros.

MySQL Connector / Net permite a los desarrolladores crear fácilmente aplicaciones .NET que requieren conectividad de datos segura y de alto rendimiento con MySQL. Implementa las interfaces ADO.NET requeridas y se integra en las herramientas ADO.NET conscientes. Los desarrolladores pueden crear aplicaciones utilizando sus idiomas .NET. MySQL Connector / Net es un controlador ADO.NET totalmente administrado escrito en C # 100% puro.

Localización

El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas

Soporte completo para distintos conjuntos de caracteres, incluyendo latin1(CP1252), german, big5, ujis, varios conjuntos de caracteres Unicode, y mucho más. Por ejemplo, los caracteres escandinavos " å" , " ä" y " ö" están permitidos en nombres de tabla y columna.

Todos los datos se guardan en el juego de caracteres elegido.

La zona horaria del servidor se puede cambiar dinámicamente y los individuales pueden especificar su propia zona horaria.

Cientes y Herramientas

MySQL incluye varios programas de cliente y utilidad. Estos incluyen programas de línea de comandos como mysqldump y mysqladmin, y programas gráficos como MySQL Workbench

MySQL Server tiene soporte integrado para sentencias SQL para comprobar, optimizar y reparar tablas. Estas instrucciones están disponibles desde la línea de comandos a través del cliente mysqlcheck, MySQL también incluye myisamchk, una utilidad de línea de comandos muy rápida para realizar estas operaciones en MyISAM tablas.

CAPITULO III
ANALISIS Y DISEÑO

4.1 Recopilación De Información

En la recopilación de información que se efectuó, nos dirigimos hacia el barrio la joya en donde se encuentra la clínica, está situada en un lugar estratégico a pocos metros del hospital general docente de la ciudad de Riobamba lo que habitualmente existe un intercambio de pacientes haciendo de la Clínica La Dolorosa un lugar concurrido, su infraestructura se encuentra en buenas condiciones y cuentan con los equipos necesarios para una buena atención, el personal que labora está completamente calificado para reaccionar en cualquier emergencia, los mismos que fueron entrevistados y en específico sobre el registro de información de los pacientes, declarando que existen formularios impresos que son llenados de forma manual e inmediatamente ingresados al archivo donde reposan en carpetas afirmando que al necesitar una historia clínica en específico deben buscarla haciendo esta actividad algo tediosa no solo para el personal sino también para el cliente

El personal cuenta con toda la predisposición de mejorar el servicio, una de las formas es la actualización de los archivos con la ayuda de un sistema de registro de historias clínicas que les ayudara a la búsqueda de información oportuna, ya su vez contando con una base de datos donde reposan los elementos necesarios de cada paciente, este método cuenta con muchas facilidades ya su vez más seguro.

4.2 Descripción de la Clínica

4.2.1 Misión

Contribuir a mejorar la calidad de vida de los habitantes de la región, facilitando el acceso al cuidado de su salud, otorgando servicios integrales, oportunos, de alta calidad técnica y profesional, financieramente sustentable, en un ambiente de calidez y seguridad.

Contar con personal competente y comprometido, integrando una red de salud privada con presencia nacional, que fomenta la innovación y mejora continua de los procesos operacionales y de servicio.

4.2.2 Visión

Ser la mejor institución de salud privada de la región, ser reconocidos por la alta calidad y seguridad de sus servicios, percibidos como un referente en la industria de la salud se puede

realizar de forma técnica y profesional a través de una herramienta tecnológica como el sistema propuesto, lo que mejorara el desempeño de la clínica.

4.3 Tipos De Investigación

- La investigación histórica: trata de la experiencia pasada, describe lo que era y representa una búsqueda crítica de la verdad que sustenta los acontecimientos pasados. El investigador depende de fuentes primarias y secundarias las cuales proveen la información y a las cuáles el investigador deberá examinar cuidadosamente con el fin de determinar su confiabilidad por medio de una crítica interna y externa.
- Investigación de Campo: Se trata de la aplicación de toda la información que ha sido investigada, comprender y resolver la problemática que actualmente se maneja en la institución donde se plantea el desarrollo del sistema web. El investigador trabaja en el ambiente natural en que conviven las personas y las fuentes consultadas, de donde se obtendrán los datos más relevantes a ser analizados y procesados.
- La investigación descriptiva: según se mencionó, trabaja sobre realidades de hecho y su característica fundamental es la de presentar una interpretación correcta.

4.3.1 Recursos Técnicos

HARDWARE:

DESCRIPCIÓN	
Procesador:	Intel(R) Core (TM) i5
CPU:	260 GHz
Memoria RAM:	4;00 GB
Sistema Operativo:	Windows 10 de 64 bits, procesador x64

Monitor:	14 pulgadas
Mouse:	Genius
Teclado:	

Tabla 3 HARDWARE

Fuente: Pablo Galarza

Software

DESARROLLO	
Sistema operativo:	WINDOWS 8.1 Pro.
Lenguaje :	JAVA
Gestor de base de datos:	MYSQL
Documentación:	OFFICE 2013 PROFESIONAL

Tabla 4 Recurso Técnico Software

Fuente: Pablo Galarza

RECURSOS HUMANOS

FUNCIONES	RESPONSABLE
Investigador	Pablo Renato Galarza Vallejo
Diseñador	
Programador	
Asesor de Tesis	

Tabla 5 Recurso Humano

Fuente: Pablo Galarza

4.3.2 Recursos Materiales

N ^o	DESCRIPCION
1	Impresora Epson L355 multifunción
2	PC portatil HP CORE-I5
3	Resmas de papel
4	Suministros de oficina: esferos, lápiz, borrador
5	Anillados y empastados
6	Movilización

Tabla 6 Recursos Materiales

Fuente: Pablo Galarza

4.3.3 Factibilidad Operativa

El sistema es operativamente viable ya que protege, almacena y reduce el tiempo a la hora de ingresar las historias clínicas de los pacientes, sin el sistema los mismos procesos, al realizarlos de una forma manual consumirían tiempo incensario y la clínica podrían estar expuestas a pérdida de información, mientras que con el sistema optimizamos los procesos mediante un control eficiente y eficaz. Además permite a la Clínica La Dolorosa conocer la posibilidad de poner en marcha la nueva tecnología para el registro de las historias clínicas, aprovechando los beneficios que ofrece, por otra parte, el correcto funcionamiento y uso del equipo, estará supeditado a la capacidad de los miembros encargados del sistema razón por la cual se debe garantizar el pleno entendimiento y manejo del equipo en cuestión.

4.4 Análisis De Los Requerimientos

En las tablas que se presentan a continuación se detalla los códigos y descripciones de los requerimientos del sistema.

Como resultado del análisis de las reuniones y entrevistas se da el planteamiento de requerimientos, entre los que se detalla requerimientos técnicos necesarios para el desarrollo del sistema web, así requerimientos funcionales.

Control de Administrador	
Código	Característica
HU01	Se debe ingresar, modificar y eliminar las diferentes provincias registradas
HU02	Permite ingresar, modificar y eliminar los diferentes cantones que existen en una ciudad
HU03	Es necesario ingresar, modificar y eliminar las diferentes parroquias que existen en un cantón
HU04	Necesito ingresar, modificar y eliminar los diferentes perfiles del sistema
HU05	Como administrador necesito ingresar, modificar y eliminar las diferentes revisiones del sistema
HU06	Se puede ingresar, modificar y eliminar los diferentes motivos del sistema
HU07	Como administrador necesito ingresar, modificar y eliminar los diferentes antecedentes del sistema
HU08	Como administrador necesito ingresar, modificar y eliminar los diferentes accidentes del sistema
HU09	Como administrador necesito ingresar, modificar y eliminar los diferentes registros del sistema
HU10	Como administrador necesito ingresar, modificar y eliminar los diferentes apf del sistema
HU11	Como administrador necesito ingresar, modificar y eliminar los diferentes avieq del sistema
HU12	Como administrador necesito reporte de historia clínica de los pacientes registrados
HU13	Como administrador necesito reporte de consultas de los pacientes registrados
HU14	Como administrador necesito registrar, modificar y eliminar los usuarios en el sistema
HU15	Como administrador necesito listas los usuarios registrados en el sistema
HU16	Como administrador necesito registrar, modificar y eliminar los

	perfiles en el sistema
HU17	Como administrador necesito listas los perfiles registrados en el sistema

Tabla 7 Control Usuario

Fuente: Pablo Galarza

Control de Administrador	
Código	Característica
HU01	Como médico necesito ingresar los diferentes registros del sistema
HU02	Como médico necesito modificar los diferentes registros registrados
HU03	Como médico necesito eliminar los diferentes registros registrados
HU04	Como médico necesito ingresar los diferentes apf del sistema
HU05	Como médico necesito modificar los diferentes apf registrados
HU06	Como médico necesito eliminar los diferentes apf registrados
HU07	Como médico necesito ingresar los diferentes avieq del sistema
HU08	Como médico necesito modificar los diferentes avieq registrados
HU09	Como médico necesito eliminar los diferentes avieq registrados
HU10	Como médico necesito ingresar los diferentes ears del sistema
HU11	Como médico necesito modificar los diferentes ears registrados
HU12	Como médico necesito eliminar los diferentes ears registrados
HU13	Como médico necesito ingresar los diferentes ears del sistema
HU14	Como médico necesito modificar los diferentes ears registrados
HU15	Como médico necesito eliminar los diferentes ears registrados
HU16	Como médico necesito reporte de historia clínica de los pacientes registrados
HU17	Como médico necesito reporte de consultas de los pacientes registrados

Tabla 8 Control de Medico

Fuente: Pablo Galarza

Control Usuario	
Código	Característica
HU01	Como usuario necesito ingresar el tipo de seguro de un paciente.
HU02	Como usuario necesito ingresar paciente de tipo emergencia.
HU03	Como usuario necesito validar que el número de cédula sea verídico.
HU04	Como usuario necesito ingresar el tipo de parentesco del contacto con el paciente.

Tabla 9 Control Usuario

Fuente: Pablo Galarza

4.4.1 Casos de uso

Diagrama de Casos de Uso

La Ilustración 27, muestra el caso de uso del Administrador en la cual se representa las acciones que puede realizar en el sistema, como ejemplo tenemos el login, ingresos, modificaciones y eliminación de entidades de la clínica, así como la creación de usuarios.

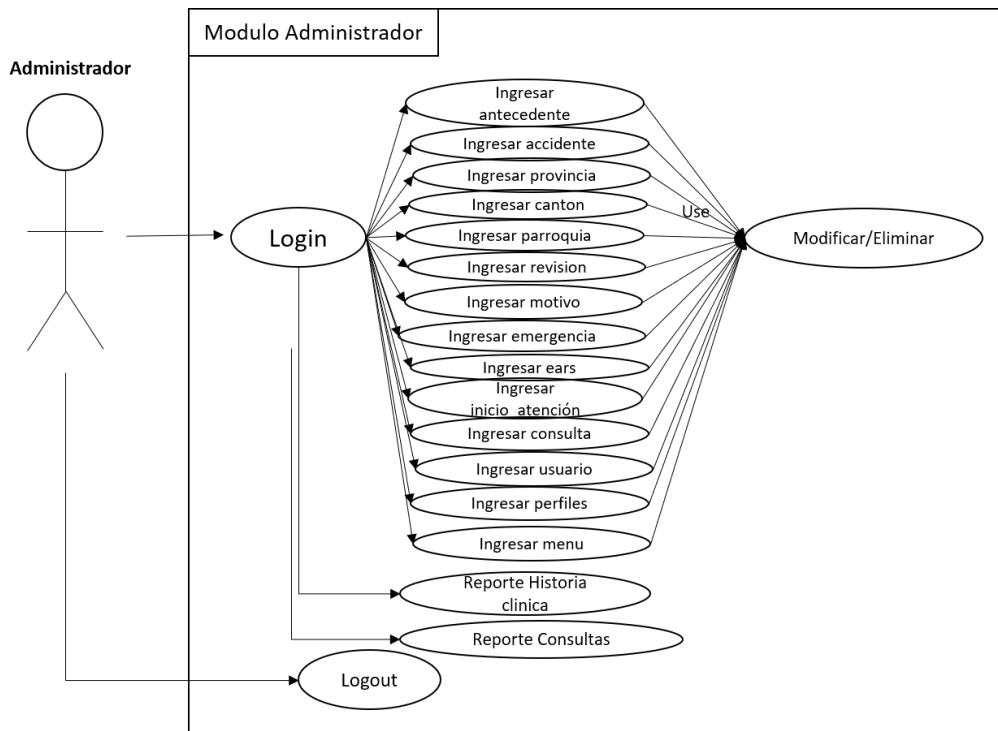


Ilustración 27 Diagrama de caso de uso administrador

Fuente: Pablo Galarza

La ilustración 28, muestra el caso de uso del médico en la cual se representa las acciones que puede realizar en el sistema, como ejemplo tenemos el login, ingreso modificación y eliminación de las diferentes entidades, así como la gestión de los pacientes y los reportes.

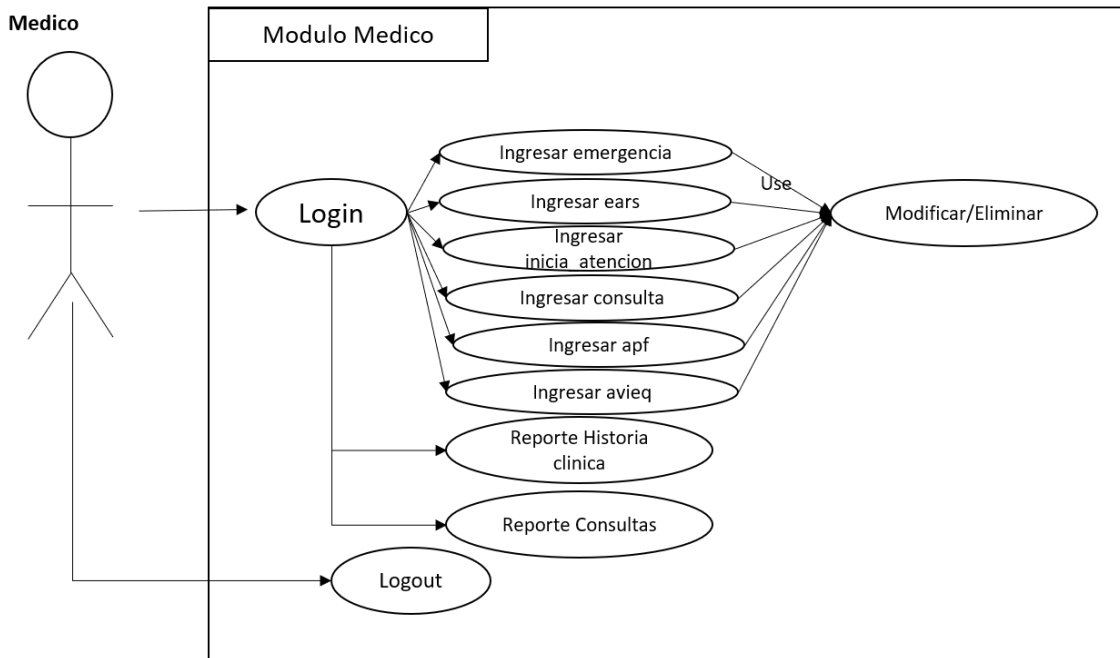


Ilustración 28 Diagrama de caso de uso del medico

Fuente: Pablo Galarza

Descripción de los casos de uso

Casos de Uso	Login
Actores	Administrador, Medico
Tipo	Básico
Propósito	Permitir que el usuario ingrese al sistema mediante su respectivo usuario y contraseña
Resumen	El administrador podrá acceder al sistema, gestionando toda la información de los diferentes usuarios registrados
Precondiciones	Es necesario ingresar con un usuario y contraseña correctos
Flujo Principal	Se ingresará al sistema para manipulación del mismo
Excepciones	Se podrá recuperar la contraseña

Tabla 10 Descripción de los casos de uso Administrador, Medico

Fuente: Pablo Galarza

Casos de Uso	Agregar, eliminar, modificar
Actores	Administrador, Medico
Tipo	Básico
Propósito	Permitir que el usuario administrador y medico puedan realizar los registros pertinentes
Resumen	El administrador podrá acceder al sistema, gestionando toda la información de los diferentes usuarios registrados, así como las configuraciones necesarias para que el usuario medico haga uso del sistema
Precondiciones	Es necesario ingresar con un usuario y contraseña correctos
Flujo Principal	Se ingresará los datos de las diferentes entidades según su perfil dentro del sistema
Excepciones	Se podrá recuperar la contraseña

Tabla 11 Caso de Uso de Agregar Eliminar Modificar

Fuente: Pablo Galarza

La ilustración 29, muestra el caso de uso del usuario en la cual se representa las acciones que puede realizar en el sistema, como ejemplo tenemos la validación de la cedula, ingresar el parentesco del contacto con el paciente, ingreso de tipo de seguro e ingreso de paciente del tipo emergencia.

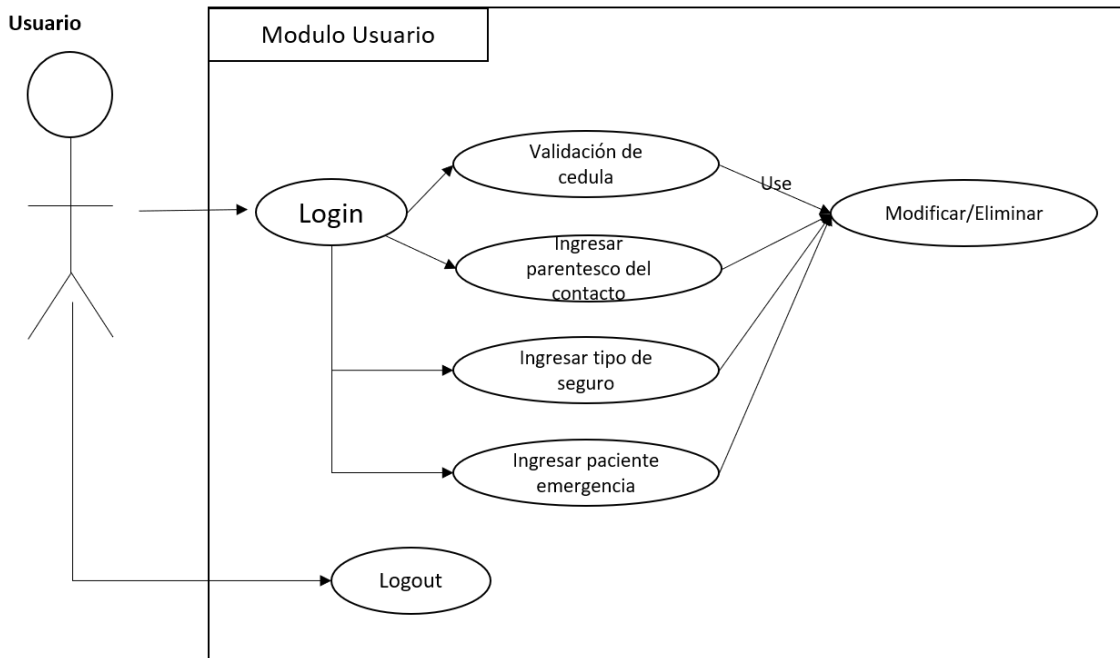


Ilustración 29 Diagrama de caso de uso del usuario

Fuente: Pablo Galarza

Casos de Uso	Reportes consulta he Historia Clínica
Actores	Administrador, Medico
Tipo	Básico
Propósito	Permitir que el medico lleve a cabo los reportes pertinentes de los pacientes registrados
Resumen	El medico podrá acceder al sistema, gestionando toda la información de los diferentes pacientes registrados, así como las generaciones de los reportes pertinentes.

Precondiciones	Es necesario ingresar con un usuario y contraseña correctos
Flujo Principal	Se llevará a cabo la generación de los reportes

Tabla 12 Caso de Uso de Historia Clínica

Fuente: Pablo Galarza

4.5 Diseño

4.5.1 Diseño conceptual

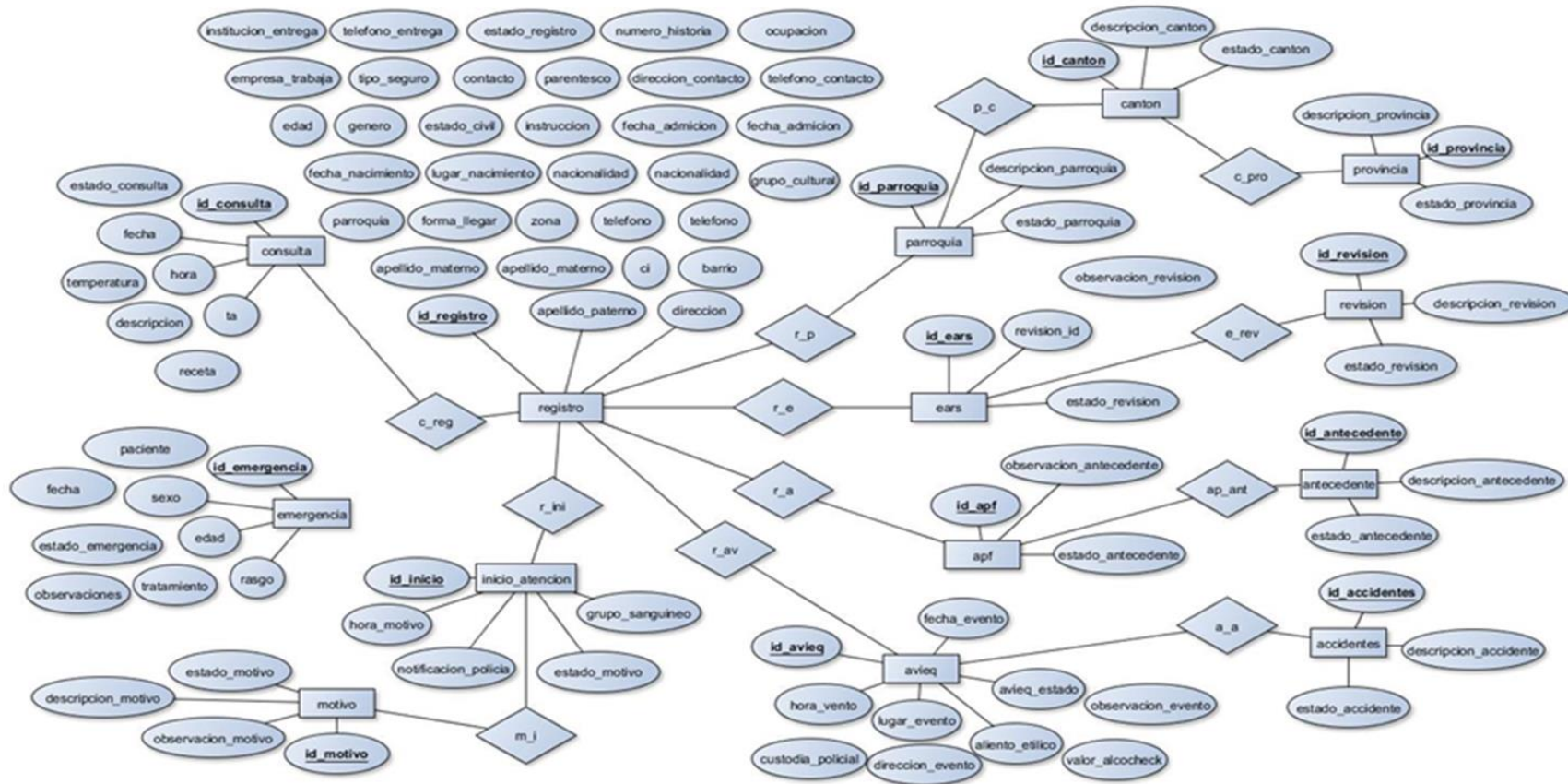


Ilustración 30 Diseño conceptual 1

Fuente: Pablo Galarza

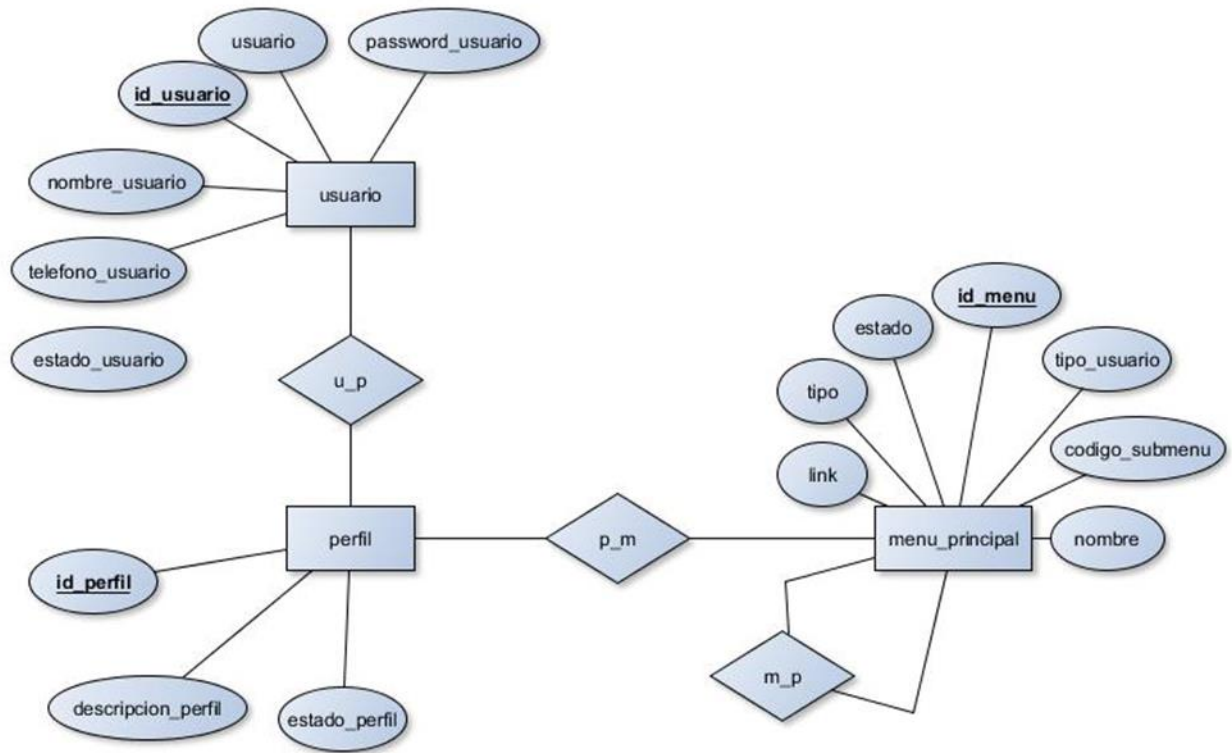


Ilustración 31 Diseño Conceptual 2

Fuente: Pablo Galarza

4.5.2 Modelo Relacional

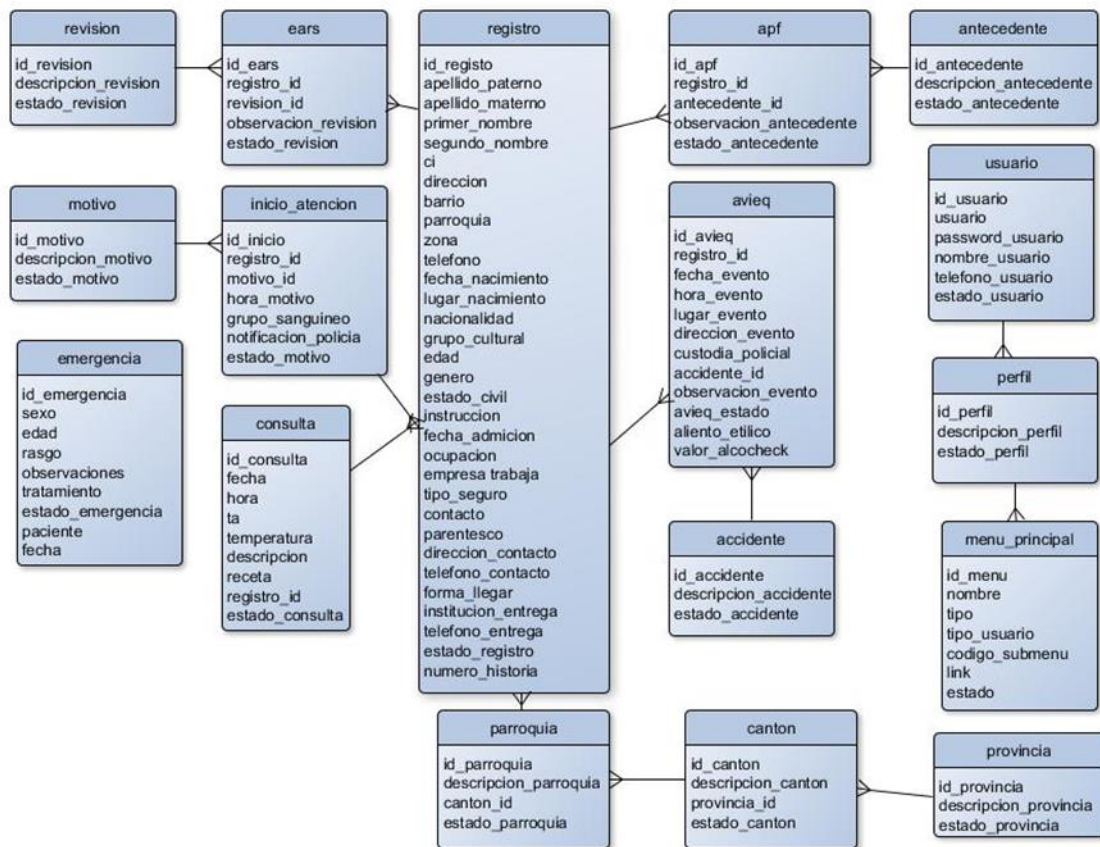


Ilustración 32 Modelo Relacional

Fuente: Pablo Galarza

4.5.3 Diccionario de Datos

Tabla1. Provincia

Esta tabla nos permite guardar las provincias disponibles que se podrán utilizar cuando se cree un nuevo registro de paciente en la clínica.

Columna	Tipo	Descripción
id_provincia	Int	Identificador de la tabla provincia
descripcion_provincia	String	Descripción de provincia

estado_provincia	Int	Estado lógico del registro
------------------	-----	----------------------------

Tabla 13 Descripción de la Tabla Provincia

Fuente: Pablo Galarza

Tabla2. Cantón

Esta tabla nos permite guardar los cantones disponibles que se podrán utilizar cuando se cree un nuevo registro de paciente en la clínica.

Columna	Tipo	Descripción
id_canton	Int	Identificador de la tabla cantón
descripción_canton	String	Descripción de cantón
provincia_id	Int	Foránea de la tabla provincia
estado_canton	Int	Estado lógico del registro

Tabla 14 Descripción de la Tabla Cantón

Fuente: Pablo Galarza

Tabla3. Parroquia

Esta tabla nos permite guardar las parroquias disponibles que se podrán utilizar cuando se cree un nuevo registro de paciente en la clínica.

Columna	Tipo	Descripción
id_parroquia	Int	Identificador de la tabla parroquia
descripción_parroquia	String	Descripcion de parroquia
canton_id	Int	Foranea de la tabla canton
estado_parroquia	Int	Estado lógico del registro

Tabla 15 Descripción de la Tabla Parroquia

Fuente: Pablo Galarza

Tabla4. Antecedente

Esta tabla nos permite guardar los antecedentes disponibles que se podrán utilizar cuando se cree un nuevo registro de paciente en la clínica.

Columna	Tipo	Descripción
id_antecedente	Int	Identificador de la tabla antecedente
descripción_antecedente	String	Descripción de antecedente
estado_antecedente	Int	Estado lógico del antecedente

Tabla 16 Descripción de la Tabla Antecedente

Fuente: Pablo Galarza

Tabla5. Accidente

Esta tabla nos permite guardar los accidentes disponibles que se podrán utilizar cuando se cree un nuevo registro de paciente en la clínica.

Columna	Tipo	Descripción
id_accidente	Int	Identificador de la tabla accidente
descripción_accidente	String	Descripción de accidente
estado_accidente	Int	Estado lógico del accidente

Tabla 17 Descripción de la Tabla Accidente

Fuente: Pablo Galarza

Tabla6. Revisión

Esta tabla nos permite guardar las revisiones disponibles que se podrán utilizar cuando se cree un nuevo registro de paciente en la clínica.

Columna	Tipo	Descripción
id_revisión	Int	Identificador de la tabla revisión
descripción_revisión	String	Descripción de revisión
estado_revisión	Int	Estado lógico del revisión

Tabla 18 Descripción de la Tabla Revisión

Fuente: Pablo Galarza

Tabla7. Motivo

Esta tabla nos permite guardar los motivos disponibles que se podrán utilizar cuando se cree un nuevo registro de paciente en la clínica.

Columna	Tipo	Descripción
id_motivo	Int	Identificador de la tabla motivo
descripción_motivo	String	Descripción de motivo
estado_motivo	Int	Estado lógico del motivo

Tabla 19 Descripción de la Tabla Motivo

Fuente: Pablo Galarza

Tabla8. Emergencia

Esta tabla nos permite guardar las emergencias disponibles que se podrán utilizar cuando se cree un nuevo registro de paciente en la clínica.

Columna	Tipo	Descripción
id_emergencia	Int	Identificador de la tabla emergencia
sexo	String	Sexo de emergencia
edad	Int	Edad de la emergencia
rasgo	String	Rasgos del paciente
observaciones	String	Observaciones de la emergencia
tratamiento	String	Tratamiento del paciente
estado_emergencia	Int	Estado lógico de la emergencia
paciente	String	Nombre del paciente
fecha	Date	Fecha de ingreso del paciente

Tabla 20 Descripción de la Tabla Emergencia

Fuente: Pablo Galarza

Tabla9. Registro (Registro de Admisión)

Esta tabla nos permite guardar el registro de los pacientes al momento de ingresar a la clínica.

COLUMNA	TIPO	DESCRIPCION
id_registro	Int	Clave primaria de la tabla registro
apellido_paterno	String	Apellido del padre
apellido_materno	String	Apellido de la madre
primer_nombre	String	Primer nombre
segundo_nombre	String	Segundo nombre

ci	String	Cedula del paciente
direccion	String	Direccion del paciente
barrio	String	Barrio del paciente
parroquia	String	Parroquia del paciente
zona	String	Zona del paciente
telefono	String	Telefono del paciente
fecha_nacimiento	Date	Fecha de nacimiento del paciente
lugar_nacimiento	String	Lugar de nacimiento del paciente
nacionalidad	String	Nacinalidad del paciente
grupo_cultural	String	Grupo cultural del paciente
edad	Int	Edad del paciente
genero	String	Genero del paciente
estado_civil	String	Estado civil de paciente
instruccion	String	Instrucción del paciente
fecha_admicion	Date	Fecha de admicion del paciente
ocupacion	String	Ocuapcion del paciente
empresa_trabaja	String	Empresa en la que trabaja
tipo_seguro	String	Tipo de seguro social
contacto	String	Nombre del contacto
parentesco	String	Parentesco del contacto con el paciente
dirección_contacto	String	Direccion del contacto
teléfono_contacto	String	Telefono del contacto
forma_llegar	String	Forma en la que llego el paciente a la clinica
institucion_entrega	String	Institucion que entrega la paciente
telefono_entrega	String	Telefono del que entrega al paciente
estado_registro	Int	Estado lógico del registro
numero_historia	Int	Número de la historia clínica

Tabla 21 Descripción de la Tabla Registro

Fuente: Pablo Galarza

Tabla10. APF (Antecedentes Personales Familiares)

Esta tabla nos permite guardar los antecedentes personales familiares disponibles que presenta el paciente al momento de ingreso a la clínica.

Columna	Tipo	Descripción
id_apf	int	Identificador de la tabla apf
registro_id	int	Foránea de la tabla registro
antecedente_id	int	Foránea de la tabla antecedente
observacion_antecedente	String	Observación de la tabla antecedente
estado_antecedente	Int	Estado lógico del antecedente

Tabla 22 Descripción de la Tabla APF (Antecedentes Personales Familiares)

Fuente: Pablo Galarza

Tabla11. AVIEQ (Accidente, Violencia, Intoxicación, envenenamiento o Quemadura)

Esta tabla nos permite guardar los registros de los AVIEQ que presenta el paciente al ingresar a la clínica.

Columna	Tipo	Descripción
id_avieq	int	Identificador de la tabla avieq
registro_id	int	Foránea de la tabla registro
fecha_emergencia	int	Fecha de la emergencia
hora_emergencia	time	Hora de la emergencia
lugar_emergencia	string	Lugar donde sucedió la emergencia
dirección_emergencia	string	Dirección de la emergencia
custodia_policial	Boolean	Indicador de custodia policial
accidente_id	int	Foranea de la tabla accidente
observación_evento	String	Observación de la emergencia
avieq_estado	int	Estado lógico del registro
aliento_etilico	boolean	Indicador de aliento etilico

valor_alcocheck	float	Valor de alcocheck
-----------------	-------	--------------------

Tabla 23 Descripción de la Tabla AVIEQ (Accidente, Violencia, Intoxicación, envenenamiento o Quemadura)

Fuente: Pablo Galarza

Tabla12. EARS (Enfermedad Actual y Revisión de Sistemas)

Esta tabla nos permite guardar los registros de los EARS que presenta el paciente al ingresar a la clínica.

Columna	Tipo	Descripción
id_ears	int	Identificador de la tabla ears
registro_id	int	Foránea de la tabla registro
revisión_id	int	Foranea de la tabla revisión
observación_revisión	string	Observacion de revision
estado_revisión	Int	Estado lógico de revision

Tabla 24 Descripción de la Tabla EARS (Enfermedad Actual y Revisión de Sistemas)

Fuente: Pablo Galarza

Tabla13. Inicio (Inicio de atención)

Esta tabla nos permite guardar los registros de los inicios de atención de los pacientes al ingresar a la clínica.

Columna	Tipo	Descripción
id_inicio	int	Identificador de la tabla inicio
registro_id	int	Foránea de la tabla registro
motivo_id	int	Foranea de la tabla motivo
hora_motivo	string	Hora de inicio de atencion
grupo_sanguineo	string	Grupo sanguíneo del paciente
notificación_policia	boolean	Notificacion a la policia
estado_motivo	int	Estado lógico del registro

Tabla 25 Descripción de la Tabla Inicio de atención

Fuente: Pablo Galarza

Tabla14. Consulta

Esta tabla nos permite guardar las consultas de los pacientes registrados en la clínica.

Columna	Tipo	Descripción
id_consulta	int	Identificador de la tabla consulta
fecha	date	Fecha de consulta
hora	time	Hora de consulta
ta	string	Ta de consulta
temperatura	float	Temperatura del paciente
descripcion	string	Descripción de la consulta
receta	string	Receta de la consulta
registro_id	int	Foranea de la tabla registro
estado_consulta	int	Estado lógico del registro

Tabla 26 Descripción de la Tabla Consulta

Fuente: Pablo Galarza

Tabla15. Emergencia

Esta tabla nos permite guardar los pacientes que no presentan datos al momento de ingresar a la clínica.

Columna	Tipo	Descripción
id_emergencia	int	Identificador de la tabla emergencia
sexo	string	Edad del paciente
edad	int	Edad del paciente
rasgo	string	Rasgos del paciente
observaciones	string	Observaciones del paciente
tratamiento	string	Tratamiento del paciente
estado_emergencia	string	Estado lógico del registro
paciente	int	Paciente ingresado
fecha	int	Fecha de ingreso

Tabla 27 Descripción de la Tabla Emergencia

Fuente: Pablo Galarza

Tabla16. Usuario

Esta tabla nos permite guardar los usuarios que serán autorizados para ingresar al sistema de la clínica.

Columna	Tipo	Descripción
id_usuario	int	Identificador de la tabla usuario
usuario	string	Usuario del sistema
password_usuario	string	Contraseña de acceso del usuario
nombre_usuario	string	Nombre de la persona a registrarse
telefono_usuario	string	Telefono de la persona
estado_usuario	int	Estado lógico del registro

Tabla 28 Descripción de la Tabla Usuario

Fuente: Pablo Galarza

Tabla17. Perfil

Esta tabla nos permite guardar los perfiles de usuario que serán autorizados para ingresar al sistema de la clínica.

Columna	Tipo	Descripción
id_perfil	int	Identificador de la tabla perfil
descripción_perfil	string	Descripcion del perfil
estado_perfil	int	Estado lógico del perfil

Tabla 29 Descripción de la Tabla Perfil

Fuente: Pablo Galarza

Tabla18. Menú Principal

Esta tabla nos permite guardar los menús de los usuarios que serán autorizados para ingresar al sistema de la clínica.

Columna	Tipo	Descripción
id_menu	int	Identificador de la tabla menu
nombre	string	Nombre del menu
tipo_usuario	int	Foranea de la tabla perfil
tipo	String	Tipo de menú M(menu) S(submenu)

codigo_submenu	int	Foránea de la tabla menú
estado	int	Estado lógico del menú
link	string	Link del menu

Tabla 30 Descripción de la Tabla Menú

Fuente: Pablo Galarza

CAPÍTULO IV
IMPLEMENTACIÓN DE
LA PLICACIÓN WEB

5.1 Configuración de las Herramientas de Desarrollo

El entorno de desarrollo que se utilizó es Netbeans el cual se procedió a instalar la versión de prueba del programa en nuestro equipo, una vez instalado se puede visualizar el IDE de la siguiente manera.

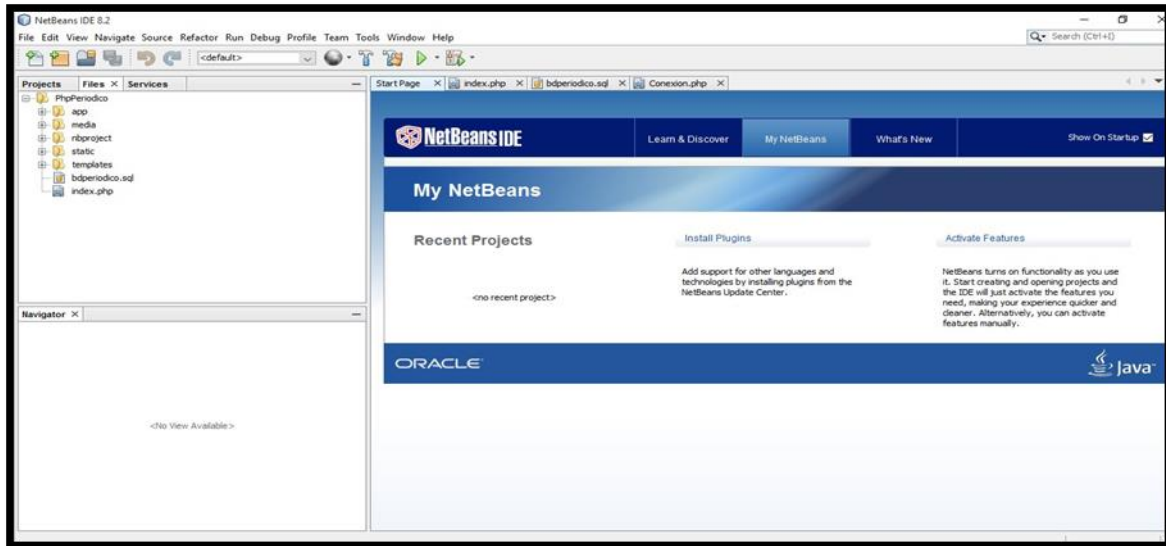


Ilustración 33 Entorno de Desarrollo Netbeans

Fuente: Pablo Galarza

Base de Datos que se utilizó es MySQL que viene dentro del paquete de software XAMPP 7.2.9 a continuación se detalla la instalación:

- Descargo XAMPP desde el la página oficial de Apache Friends(<https://www.apachefriends.org/es/index.html>)
- Al hacer clic nos aparecerá un wizard de instalación, en el que solo le aceptamos y damos siguiente.
- Abrimos las aplicaciones y realizamos las configuraciones necesarias como declaración de puertos etc.
- Para mandar a correr al servidor de base de datos bata con dar clic en star en la pestaña mySql y el servidor de base de datos estará listo para su uso.

Este programa permite realizar desarrollo en varios lenguajes de programación utilizando una interfaz amigable e intuitiva, posee un conjunto de herramientas que facilitan el diseño e implementación de sitios web de una manera rápida y óptima, presentando entornos de Diseñador y Desarrollador.

5.2 Arquitectura del Sistema

Dentro de la arquitectura del sistema de Historias Clínicas, se considera la utilización de recursos tecnológicos mismo que ayudaran al óptimo funcionamiento del sistema. En la siguiente figura se detallan la estructura necesaria de la implementación del sistema.

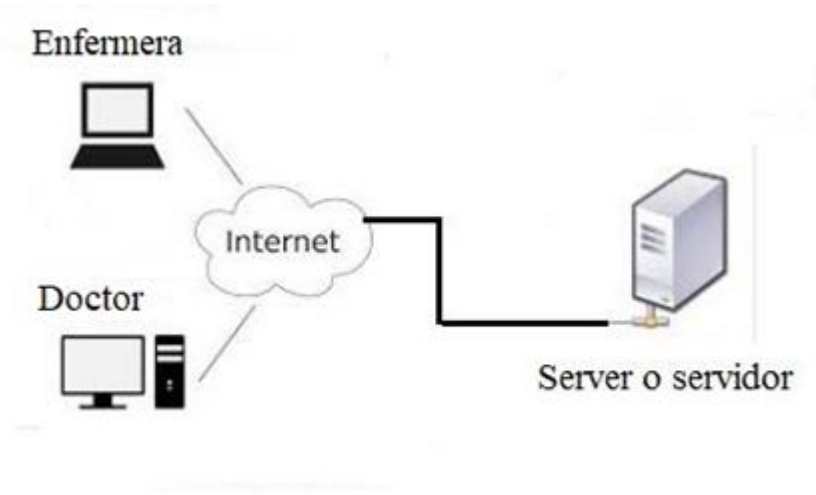


Ilustración 34 Arquitectura del Aplicativo

Fuente: Pablo Galarza

En el diagrama se puede apreciar la interacción de los recursos de los que hace uso el sistema para su funcionamiento. Para ingresar al sistema el cliente es necesario hacerlo a través de una computadora con acceso a internet, tan solo escribiendo la url del sistema, el servidor recibirá la petición y servirá la aplicación dependiendo el perfil del usuario logueado, presentando de esta manera como respuesta del servidor al cliente la página web desplegada dentro del navegador, haciendo que de esta manera el cliente solo haga peticiones y que el proceso duro con mayor robustez lo haga el servidor, brindando una mejor experiencia al usuario.

5.3 Implementación de la Aplicación

Para iniciar la implementación es necesario tener el resultado del análisis y diseño que se realizó, para poder planificar el proceso de desarrollo que será mediante fases evolutivas especificando con módulos cada uno de los requerimientos principales, de este modo resultara más practico el control del desarrollo y permitirá ir corrigiendo posibles inconvenientes a medida que se vayan presentando.

5.3.1 Definición de Módulos de Implementación

Para la definición de los módulos de implementación se hizo un estudio de los principales requerimientos que necesitaba la clínica, así como cuales serían los actores que participarían dentro del sistema dando como resultado un produc backlog con un total de 57 requerimientos de usuario. Dentro de los módulos principales a realizarse encontramos: Modulo de Historias Clínicas, Modulo de Emergencia, Modulo de consulta, Modulo de usuarios, Modulo de Reportes, Modulo de Configuración.

5.3.2 Desarrollo de la Aplicación Implementación

Una vez definido todos y cada uno de los pasos anteriores (análisis y diseño) se procede a la implementación el sistema de acuerdo con los requerimientos especificados y tratando en lo posible de satisfacer todos los pormenores en cuestión de usabilidad facilitando así el uso del sistema y optimizando las tareas. Al iniciar la aplicación, ésta presenta su pantalla principal que es una recopilación de toda la información principal y los eventos que la institución realiza.

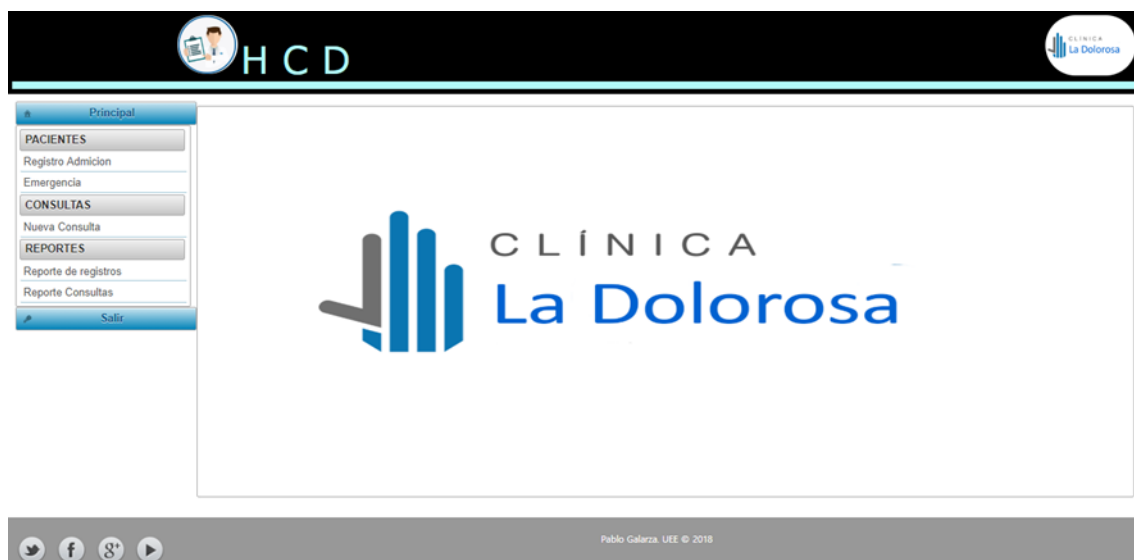


Ilustración 35 Pantalla Inicial de la Aplicación Web

Fuente: Pablo Galarza

5.3.3 Implantación de la Aplicación

La implantación del aplicativo se la realizó en un Hosting Web para la institución. El dominio con el cual está vinculada es el sitio del Sistema de Historia Clínica tendrá un enlace que se llamará www.clinicaladolorosa.com El proceso de publicación del aplicativo en el hosting se detalla en el Manual Técnico, que consta como (ANEXO 1).

5.4 Pruebas de la Aplicación

Para la medición de calidad de la aplicación me base en la norma ISO/IEC 25000, norma que mesura la calidad de los productos de esta naturaleza.

Dicho estándar fue creado con la finalidad de ayudar en los procesos de recolección de características de calidad y evaluación a través de un modelo de proceso en el estudio y análisis de la usabilidad de productividad de software.

Con la finalidad de valorar la usabilidad como facilidad de uso, comprensión y satisfacción en su automatismo, así también la productividad; que son tiempos estimados de respuesta automatizado y no automatizado para completar una tarea dentro del proceso característico del trabajo de titulación se detalla a continuación en la Tabla **Características y métricas de evaluación** se precisa las métricas para la evaluación de la productividad del sistema web.

Características y métricas de evaluación

Característica	Sub-característica	Acotación de métrica	Propósito	Fuente
Productividad	Comportamiento en el tiempo	Tiempo de respuesta automatizado y no automatizado	Tiempo óptimo para completar una tarea.	Procesos importantes. Test.
	Utilización de recursos	Esfuerzo empleado en relación al recurso	Eficiencia de los recursos utilizados.	Diseño Test

Usabilidad	Identidad	Pertinencia con la unidad educativa	Que tan coherente es la identidad del sistema web con la institución	Test
	Contenido	Distribución de información	Qué nivel de correspondencia precisa la información presentada en el sistema	Test
	Navegación	Distinción de sus funcionalidades	Que tan evidente son las funcionalidades en el sistema	Test
	Utilidad	Objetivo del sistema	El sistema web en qué nivel corresponde resolviendo y ayudando un problema.	Test

Tabla 31 Características

Fuente: Pablo Galarza

Se considera como punto de partida para el análisis de los datos, tipos de medidas que se definen en la Tabla **Tipos de Medida**. Cada tipo será utilizado para clasificar los atributos como:

- Tiempo (T): Mide intervalos de tiempo.
- Ratio (R): Expresa un porcentaje específico de cumplimiento de la norma.

Tipos de Medida

Medida	Tipo	Dominio	Unidad	Símbolo
Tiempo	Entero		s/m/h/d	T
Ratio	Entero	0-100	%	R

Tabla 32 Tipos de Medida

Fuente: Pablo Galarza

Validación de Usabilidad

La capacitación y uso del sistema web se realizó al 100% del personal del departamento de la clínica “La Dolorosa”, donde se puso a consideración el test de usabilidad el cual tiene como finalidad la evaluación de identidad, contenido, navegación, utilidad y retroalimentación por su alta interactividad con el sistema web, siendo 15 el total de personas involucradas.

Identidad

Identidad	SI	NO	%
¿Con la información que se ofrece en la página principal, es posible saber a qué institución o departamento corresponde el sitio?	15	0	100
¿Relaciona los colores predominantes en el sitio web con la clínica “La Dolorosa”?	13	2	86.67
¿De los elementos que muestra esta pantalla, hay algo que usted crea que está fuera de contexto?	1	14	93.33
¿Distingue alguna imagen que represente (logotipo) a la clínica “La Dolorosa” y que aparece en un lugar importante dentro de la página?	15	0	100
TOTAL :			95

Tabla 33 Identidad

Fuente: Pablo Galarza

Análisis: De acuerdo a los resultados obtenidos de las preguntas cerradas de SI y NO de la sección de Identidad que forma parte del test de usabilidad que se realizó a una muestra de 15 personas pertenecientes a la clínica “La Dolorosa”, se obtuvo un total del 95% de aceptación representada en la Tabla identidad.

Contenido

Contenido	SI	NI	%
¿Considera la información visualizada organizada y de acuerdo a su perfil de usuario?	15	0	100
¿Es fácil distinguir los datos que deben ser ingresados o enviados en la aplicación web?	13	2	86,67
¿La información generada es suficiente para tener una descripción clara?	15	0	100
¿Encontró información redundante en la aplicación web?	0	15	100
TOTAL :			96,68

Tabla 34 Contenido

Fuente: Pablo Galarza

Análisis: De acuerdo a los resultados obtenidos de las preguntas cerradas de SI y NO de la sección de Contenido que forma parte del test de usabilidad que se realizó a una muestra de 15 personas pertenecientes a la clínica “La Dolorosa”, se obtuvo un total del 96,68% de aceptación verificando que el contenido es de manera clara y fácil de entender de acuerdo a su funcionalidad, representada en la Tabla Contenido

Navegación

Navegación	SI	NI	%
¿La forma en que se navega por la aplicación web, y sus diferentes opciones, es clara? ¿Se distingue fácilmente?	14	1	93,99
¿Logra distinguir la funcionalidad de los diferentes módulos que posee el sistema?	14	1	93,99
TOTAL :			93,99

Tabla 35 Navegación

Fuente: Pablo Galarza

Análisis: De acuerdo a los resultados obtenidos de las preguntas cerradas de SI y NO de la sección de Navegación que forma parte del test de usabilidad que se realizó a una muestra de 15 personas pertenecientes a la clínica “La Dolorosa” de la institución, se obtuvo un total del

93,33% de aceptación verificando la fácil distinción de los módulos que posee el sistema, representada en la Tabla Navegación.

Utilidad

Utilidad	SI	NI	%
¿Le queda claro cuál es el objetivo de la aplicación web? ¿Qué contenido y servicio ofrece?	15	0	100
¿Cree que los contenidos y servicios que se ofrecen en esta aplicación son de utilidad para su caso personal?	15	0	100
TOTAL :			100

Tabla 36 Utilidad

Fuente: Pablo Galarza

Análisis: De acuerdo a los resultados obtenidos de las preguntas cerradas de SI y NO de la sección de Utilidad que forma parte del test de usabilidad que se realizó a una muestra de 15 personas pertenecientes a la clínica “La Dolorosa”, se obtuvo un total del 100% de aceptación lo cual demuestra que el sistema será de gran utilidad para cada uno de los procesos que realice, representada en la Tabla Utilidad.

Retroalimentación

Retroalimentación	SI	NI	%
¿Le llamó la atención positivamente a la utilidad que ofrece la aplicación web?	15	0	100
TOTAL :			100

Tabla 37 Retroalimentación

Fuente: Pablo Galarza

Análisis: De acuerdo a los resultados obtenidos de las preguntas cerradas de SI y NO de la sección de retroalimentación que forma parte del test de usabilidad que se realizó a una muestra de 15 personas pertenecientes a la clínica “La Dolorosa”, se obtuvo un total del 100% de aceptación lo cual nos muestra que el sistema es atractivo para cada uno de los miembros de dicho departamento, representada en la Tabla Retroalimentación.

Resultados del Test de Usabilidad

Secciones	Positivo(%)	Negativo(%)
Identidad	95	5
Contenido	96,68	3,32
Navegación	93,33	6,67
Utilidad	100	0
Retroalimentación	100	0
Total	97%	3%

Tabla 38 Test Usabilidad

Fuente: Pablo Galarza

La Tabla del Test de Usabilidad, nos muestra los resultados obtenidos de acuerdo a cada uno de las secciones que conforman el test de usabilidad. Dichos resultados demuestran la gran aceptación que el sistema ha tenido el sistema dentro de la clínica “La Dolorosa”.

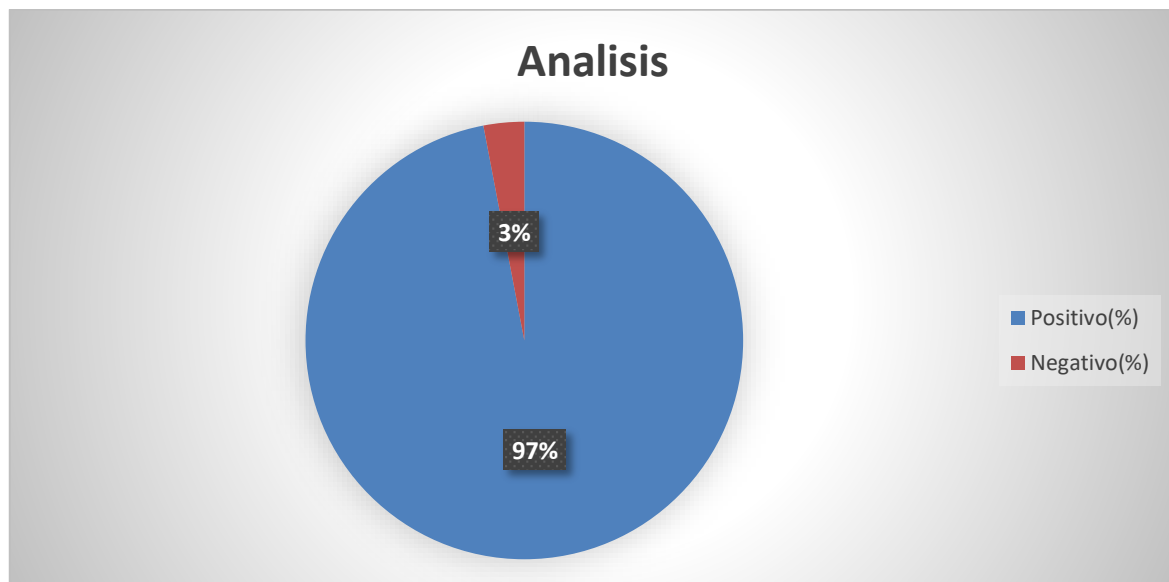


Ilustración 36 Análisis

Fuente: Pablo Galarza

Análisis: Con el test de usabilidad rescatando la identidad, contenido, utilidad, navegación y retroalimentación realizada por las personas que conforman la clínica “La Dolorosa”, se logró evaluar a la usabilidad del sistema como aceptada obteniendo un resultado del 97%,

evidenciando facilidad de uso, correspondencia de identidad institucional y una curva de aprendizaje mínima.

Validación productividad

Se considera las actividades o tareas relevantes que debe cumplir el sistema web y la métrica de evaluación en horas; que optimiza el proceso manualmente llevado para los miembros que conforman la clínica “La Dolorosa”.

Indicadores de Actividad

Indicador E1: Registro Historia Clínica.

Métrica	No automatizado	Automatizado
Tiempo(min)	10,47	3,71

Tabla 39 Indicador

Fuente: Pablo Galarza

La tabla presenta la métrica en minutos para el proceso de Registro Historia Clínica que están dirigidas al registro del paciente de la clínica “La Dolorosa”, estos, por parte del personal, mediante la comparación de un proceso no automatizado a uno automatizado, para comprobar su productividad, valores obtenidos tras un promedio resultante de una ficha de observación aplicado a un total de 20 lecturas.

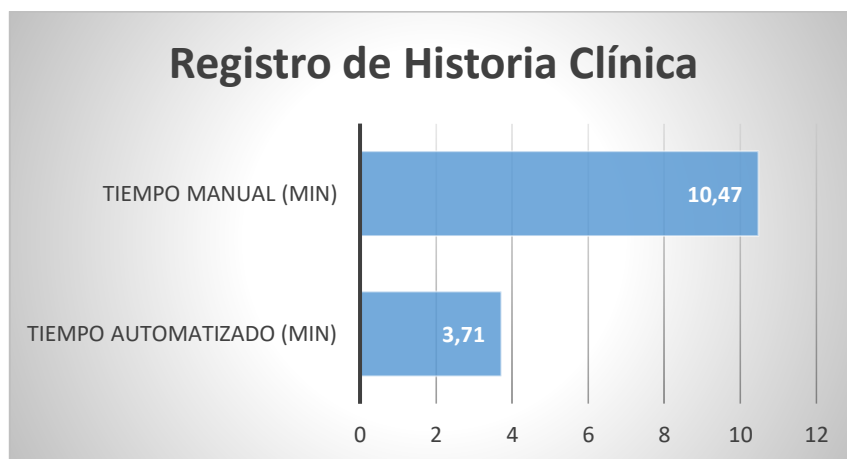


Ilustración 37 Registro de Historias Clínicas

Fuente: Pablo Galarza

En la figura, se evidencia que el tiempo empleado para esta tarea es 10,47 minutos una cantidad mayor al tiempo automatizado que es de 3,71 minutos

En donde:

10,47 minutos es igual al 100% del tiempo utilizado para el proceso de Registro Historia Clínica

3,71 minutos es el tiempo utilizado después de la implantación del sistema web, es decir 6,76 minutos es la diferencia entre el tiempo no automatizado y el automatizado que aumenta la productividad del proceso.

$$X = \frac{3.71 \cdot 100}{10.47} = 35,43\%$$

Resumen de los indicadores de Evaluación

Indicadores	Tiempo Manual	Tiempo Automatizado
E1	10,47	3,71
Total, minutos	10,47	3,71

Tabla 40 Indicadores de Evaluación

Fuente: Pablo Galarza

Análisis: El resultado de la regla de 3 evidencia un empleo de tiempo de **35,43 %** logrado en el proceso automatizado en relación a los procesos de forma manual, obteniendo una reducción del **64,57%** en los tiempos aumentando la productividad para el desarrollo de las diferentes actividades a través del automatizado de los procesos que se lleva a cabo dentro de la clínica bajo la regla de proporción matemática para averiguar la cantidad optimizada en ratio o valor porcentual comúnmente conocida como regla de tres.

CAPÍTULO V
CONCLUSIONES Y
RECOMENDACIONES

6.1 Conclusiones

- Todas las ventajas y características que tiene el lenguaje java y mysql fueron analizadas para posteriormente estudiarlas, encontrando que permite desarrollar sistemas que beneficien y aporten al funcionamiento de cualquier lugar que lo necesite mejorando el control y su seguridad.
- Con las técnicas de recolección de datos como encuesta y entrevistas podremos tener una idea clara del manejo o del funcionamiento de los procesos de historias clínicas con el fin de plasmarlos en un sistema informático
- El sistema desarrollado facilita el uso de registro de historias clínicas para las enfermeras optimizando tiempo, además de la posible pérdida de información de los pacientes, y ahorrando espacio físico de almacenamiento de la documentación

6.2 Recomendaciones

- Investigar los distintos framework que nos permita un manejo fácil y rápido en la utilización de plantillas predeterminadas con el fin de no programar desde cero esto será de mucha ayuda ya que la persona encargada optimizara tiempo y recursos.
- Realizar un análisis exhaustivo utilizando entrevistas periódicas que ayudara a detectar las diferentes necesidades que se puedan presentar en la Clínica de este modo poder corregir las problemáticas
- Todos los procesos manuales que existen se los pueden automatizar a través de sistemas informáticos sean de escritorio o web con el objetivo de mejorar los servicios que se presta además de utilizar las nuevas tecnologías que tenemos a nuestro alcance.
- Capacitar a los empleados en el buen uso del sistema informático mediante el estudio del manual, de este modo se podrán utilizar de una forma correcta y sin contratiempos, esto ayudara a que no existan errores o fallas futuras en la aplicación

BIBLIOGRAFÍA

WEB BIBLIOGRÁFICA

[1] Introducción al lenguaje Java, Desde:

<http://www.jtech.ua.es/dadm/restringido/java/sesion01-apuntes.pdf>

[2] Javier Garcia de Jalon, Jose Ignacio Rodriguez, Iñigo Mingo, Aitor Imaz, Alfonso Brazalez. APRENDA JAVA Editorial TECNUN pág: 10-30. Desde:

<http://www4.tecnun.es/asignaturas/Informat1/AyudaInf/aprendainf/Java/Java2.pdf>

[3] Jorge Martinez Ladron de Guevara, FUNDAMENTOS DE PROGRAMACION EN JAVA, Editorial EME, pág: 90 – 102. Desde:

<https://docs.google.com/file/d/0Byy7aUI9u4fBdnZjVnZOampmTjA/edit>

[4] BOUDREAU Tim, GLICK Jesse, GREENE Simeon, SPURLIN Vaughn, J. WOHR Jack; NETBEANS: THE DEFINITIVE GUIDE: DEVELOPING, DEBUGGING, AND DEPLOYING JAVA CODE, Editorial O'REILLY, pág: 15 – 22. Desde:

https://books.google.es/books?hl=es&lr=&id=JCMBWozfcJUC&oi=fnd&pg=PR5&dq=netbeans&%20ots=Wz_T479jd3&sig=UscP0ev-Tix8IIXQOoxv5C-nWrA#v=onepage&q=netbeans&f=false

[5] ÁNGELES ÁNGELES Francisco, DREAMWEAVER, año 2013. Desde:

<http://repository.uaeh.edu.mx/bitstream/bitstream/handle/123456789/14825/Dreamweaver.pdf?sequence=1>

[6] JAVA SERVER PAGES TECHNOLOGY. Desde:

<http://www.oracle.com/technetwork/java/javaee/jsp/index.html>

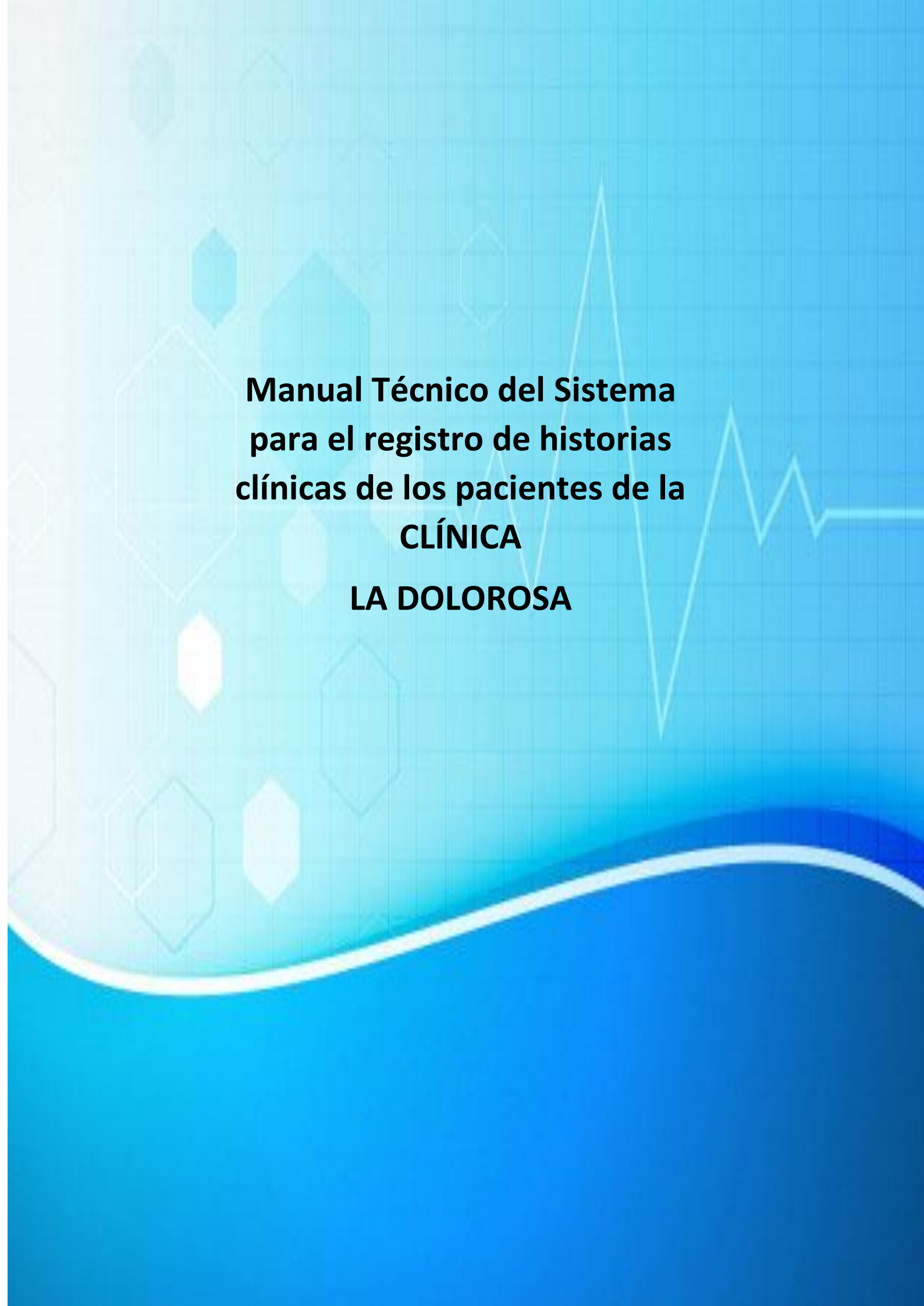
[7] OSMOSIS Latina, MENÚ CURSO JAVA WEB (JSP, SERVLETS), año 2000 - 2011. Desde: <https://javaweb.osmosislatina.com/curso/jsp.htm>

[8] Francisco Manuel Pérez Montes; EJERCICIOS DE PROGRAMACIÓN EN JAVA, Editorial Edita Asoc. Por la Innovación Educativa Eduinnova. Desde:

<http://www.eduinnova.es/monografias2011/ene2011/java.pdf>

- [9] LUJÁN MORA Sergio, ASP. Desde:
https://gplsi.dlsi.ua.es/~slujan/asp/Introduccion_a_ASP.htm
- [10] MENDOZA Geovanny, HERRAMIENTA DE DESARROLLO NETBEANS, año 2015. Desde:
http://www.consultorjava.com/wp/wp-content/uploads/2015/09/herramienta_desarrollo_netbeans.pdf
- [11] COBO Ángel, DISEÑO Y PROGRAMACIÓN DE BASE DE DATOS, Editorial Vision Libros. Desde:
https://books.google.es/books?hl=es&lr=lang_es&id=anCDr9N-kGsC&oi=fnd&pg=PA7&dq=Gestor+de+Base+de+Datos+&ots=UYBBu6qrU&sig=A110qlzNT8_k8PP9EHRZA-Jo4M1I#v=onepage&q&f=false
- [12] THE MAIN FEATURES OF MySQL. Desde:
<https://dev.mysql.com/doc/refman/5.7/en/features.html>
- [13] MySQL DOCUMENTATION. Desde: <https://dev.mysql.com/doc/>
- [14] PROGRAMACIÓN JAVA. Desde:
<http://puntocomnoesunlenguaje.blogspot.com/2013/12/java-jdbc-conectar-base-de-datos.html>

ANEXOS



**Manual Técnico del Sistema
para el registro de historias
clínicas de los pacientes de la
CLÍNICA
LA DOLOROSA**

Introducción

Este sistema web se ha diseñado para el registro de pacientes de la Clínica “La Dolorosa”. El mismo que permitirá agilizar el registro de pacientes para que puedan hacer uso de una mejor administración y uso de los datos de los pacientes, además que se desarrollan en la institución de manera online.

A continuación, detallaremos algunas especificaciones técnicas que se deben tomar en cuenta para el correcto funcionamiento del sistema.

Requerimientos del sistema.

Para el correcto funcionamiento del sistema es necesario cumplir con una serie de requisitos, tanto hardware como software.

Los requisitos previos de software instalado para poder ejecutar el sistema serían:

1. Gestor de Base de datos: MYSQL Versión 8.0.12
2. Lenguaje de programación: Java
3. Herramientas de diseño Web: HTML5, CSS3, JavaScript, JQuery, Java JDK8, ID Netbeans
4. Navegador Web: Se puede utilizar cualquier explorador web, preferiblemente Google Chrome
5. IDE: NetBeans Versión 8.2.10
6. Administrador de Base de Datos: phpMy Admin
7. Servidor de aplicación: Servidor XAMPP Versión 3.2.2

Los requisitos mínimos de hardware serían los siguientes:

1. Microsoft Windows 10, Vista, 7, 8.1.
2. PC con procesador Pentium, a velocidad de 200MHz o superior
3. 1 GB de espacio de disco

MODELO DE LA BASE DE DATOS DEL SISTEMA

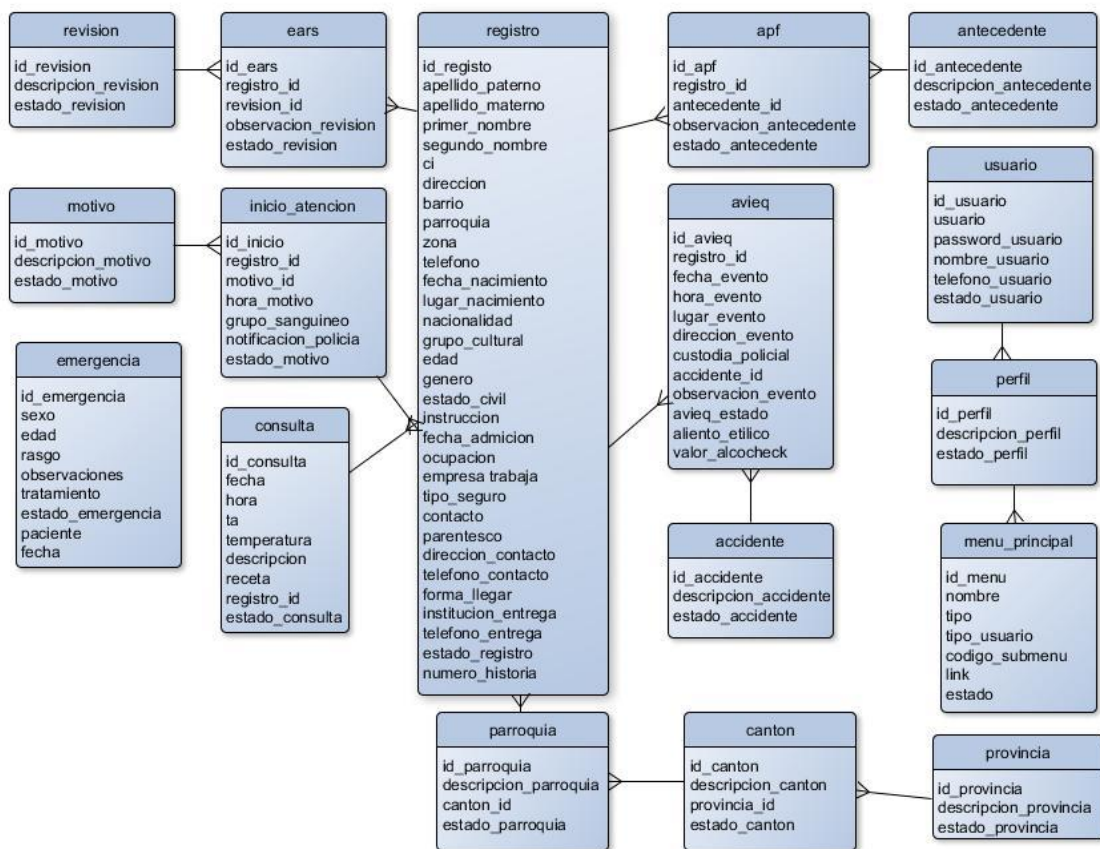


Gráfico de la Base de Datos

TABLAS, PROCEDIMIENTOS ALMACENADO Y VISTAS DEL SISTEMA EN EL ADMINISTRADOR DE LA BASE DE DATOS PhpMyAdmin

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id_registro	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	apellido_paterno	varchar(250)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
3	apellido_materno	varchar(250)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
4	primer_nombre	varchar(250)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
5	segundo_nombre	varchar(250)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
6	ci	varchar(10)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
7	direccion	varchar(250)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
8	barrio	varchar(250)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
9	parroquia	int(11)			No	Ninguna			Cambiar Eliminar Más
10	zona	varchar(250)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
11	telefono	varchar(10)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
12	fecha_nacimiento	date			No	Ninguna			Cambiar Eliminar Más
13	lugar_nacimiento	varchar(250)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
14	nacionalidad	varchar(250)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
15	grupo_cultural	varchar(30)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
16	edad	int(11)			No	Ninguna			Cambiar Eliminar Más
17	genero	varchar(20)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
18	estado_civil	varchar(30)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más

Administrador de la base de datos phpMyAdmin

ARQUITECTURA DEL DISEÑO DEL DIRECTORIO SISTEMA

Para la programación del sistema de Historias Clínicas, hemos utilizado el patrón de diseño DAO así como una programación en capas que nos permitirá encapsular de una manera correcta nuestro código para posibles ataques informáticos, además que permitirá que el sistema sea estable y este propenso a realizar cualquier tiempo de cambio en el futuro.

ARCHIVOS DEL DIRECTORIO DEL SISTEMA	<p>Nuestro sistema se divide en varios paquetes, cada una de ella contiene archivos que cumplen una función específica, a continuación detallaremos las que se utilizaron en nuestro proyecto.</p> <p>DIRECTORIO Web Pages</p> <p>En esta carpeta tendremos todas las vistas del sistema si queremos cambiar alguna parte del diseño tendremos que hacerlo mediante este directorio</p> <p>1.WEB-INF.- Contiene la plantilla del sistema.</p>
--	--

	<p>2. css. - Contiene los css del sistema, todos los diseños del sistema.</p> <p>3. img. – Contiene contiene todas las imágenes del s utilizaremos para cada mantenimiento.</p> <p>4. reportes. - Permite almacenar los reportes.</p> <p>5. Web Pages.- En esta carpeta tendremos todas nuestras páginas web del sistema, por cada mantenimiento tendremos un sin número de páginas, login, funciones, etc.</p> <p>PAQUETE BEAN</p> <p>Dentro de este paquete se mantendrán los archivos que harán una gestión de las vistas, llamando a los diferentes métodos del dao, y control de eventos de las vistas.</p> <p>PAQUETE DAO</p> <p>Dentro de este paquete se mantendrán toda la lógica de cana entidad como las acciones del CRUD y conexión a la base de datos.</p> <p>PAQUETE MODEL</p> <p>Dentro de este paquete se mantendrán todas las propiedades de las clases que forman parte del sistema.</p>
--	--

ARQUITECTURA DEL FUNCIONAMIENTO DEL SISTEMA

Para explicar cómo funciona nuestro sistema en la parte de la arquitectura, se lo ha realizado mediante la realización de capas; para poder encapsular el código y al mismo tiempo tenerlo listo para futuros cambios.

A continuación, detallaremos el funcionamiento de 1 mantenimiento (Cargo provincia) para explicar cómo se lo realizó.

Conexión

Conexión docente del instituto

La clase conexión tendrá todos los parámetros de nuestro servidor y base de datos; de esta manera cada vez que queremos realizar una acción (insertar, editar, listar o eliminar) deberemos realizar una instancia a la misma. Para la conexión utilizaremos la forma JDBC .

JDBC es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede. Es decir, no existen métodos propios de cada base de datos, si no que JDBC define una interface estándar que las bases de datos tienen que soportar

SCRIPT DE LA CONECCION

```
public void Conectar() {
    try {
        String url = "jdbc:mysql://localhost:3306/bdhc?zeroDateTi
        String usuario = "root";
        String contraseña = "";
        Class.forName("com.mysql.jdbc.Driver");
        cn = DriverManager.getConnection("jdbc:mysql://localhost:
    } catch (Exception e) {
        //
        System.out.printf("error");
        System.out.printf(e.getMessage());
    }
}
```

Mediante este script se realiza la conexión a la base de datos, para los cuales se hace uso de un usuario (root), contraseña, el driver de conexión, y el nombre de la base de datos, para lo cual hacemos uso del Driver manager que mediante estos parámetros nos permite mantenernos conectados a la base de datos.

MODELOS

MODELO CANTON

```
public class mCanton {  
  
    private int id_canton;  
    private String descripcion_canton;  
    private mProvincia provincia;  
  
    public mCanton() {  
        this.provincia= new mProvincia();  
    }  
  
    public int getId_canton() {  
        return id_canton;  
    }  
  
    public void setId_canton(int id_canton) {  
        this.id_canton = id_canton;  
    }  
  
    public String getDescripcion_canton() {  
        return descripcion_canton;  
    }  
  
    public void setDescripcion_canton(String descripcion_canton) {  
        this.descripcion_canton = descripcion_canton;  
    }  
}
```

Cada tabla de la base de datos debe estar representada en forma de clase en el directorio model, cada una clase debe tener los mismos atributos que se consideren que se utilizaran para realizar acciones, así mismos constructores y sus respectivos getter y setter para poner y recoger valores cuando se realicen instancias a estas clases.

DAO(Clase que obtiene la lógica de los objetos, comportamientos y acciones)

Aquí tendremos la lógica de nuestro proyecto, para poder utilizar debemos previamente implementar su debido modelo para poder programar en cada método implementando. A continuación, detallaremos cada parte del código para saber cómo funciona.

PRODEDIMIENDO ALAMCENADO

```
DELIMITER $$  
CREATE PROCEDURE listaConsultas()  
BEGIN  
SELECT * FROM consulta as c INNER join registro as r  
on c.registro_id = r.id_registro  
END $$  
DELIMITER;  
  
call listaConsultas()
```

Cada mantenimiento necesita interactuar con su respectiva tabla en la base de datos, por eso para las funciones de insertar, editar, y eliminar se utilizaron funciones, el cual con tal solo enviarle los parámetros solicitados realizara una acción, y en el caso que ocurra un error podremos capturar el error y evitaremos que el sistema deje de compilar en ese preciso momento. En cambio, para poder presentar los listados de las consultas llamaremos a las funciones.

FUNCION DAO INSERTAR

```
public boolean registrar(mCanton can) throws Exception {  
    try {  
        this.Conectar();  
        System.out.println("Esto sale en pantalla\n");  
        PreparedStatement st = this.getCn().prepareStatement("INSERT INTO canton(descripcion_canton,  
st.setString(1, can.getDescripcion_canton());  
st.setInt(2, can.getId_canton());  
st.setInt(3, 1);  
st.executeUpdate();  
    } catch (Exception e) {  
        System.out.println("Error...\n");  
        System.out.println(e.getMessage());  
        throw e;  
    } finally {  
        this.Cerrar();  
    }  
    return true;  
}
```

Esta función en DAO, lo primero que hace es hacer una instancia o llamada a la base de datos, para luego preparar el procedimiento almacenado que vamos a utilizar, en este cargo SP_CARGO, luego enviamos los parámetros necesarios que nos pide el procedimiento y procedemos a ejecutar; para terminar con el retorno de un mensaje, cabe recalcar que el caso que exista un error retornaremos en este caso el mensaje del error porque estamos trabajando con try catch para las excepciones.

FUNCION DAO EDITAR

```
public void modificar(mCanton can) throws Exception {
    try {
        this.Conectar();
        System.out.println("Esto sale en pantalla\n");
        PreparedStatement st = this.getCn().prepareStatement("UPDATE canton SET descripcion_canton=?,
        st.setString(1, can.getDescripcion_canton());
        st.setInt(2, can.getProvincia().getId_provincia());
        st.setInt(3, can.getId_canton());
        st.executeUpdate();
    } catch (Exception e) {
        System.out.println("Esto sale en pantalla\n");
        System.out.println(e.getMessage());
        throw e;
    } finally {
        this.Cerrar();
    }
}
```

Esta función en DAO, lo que hace es crea una llama a la función que se creara en la base de datos, luego del objeto recibido va sacando los parámetros, para luego ejecutar la función y en caso de éxito o error retornar una respuesta

FUNCION DAO ELIMINAR

```
public void eliminar(mCanton can) throws Exception {
    try {
        this.Conectar();
        System.out.println("Esto sale en pantalla\n");
        PreparedStatement st = this.getCn().prepareStatement("UPDATE canton SET estado_canton=0
        st.setInt(1, can.getId_canton());
        st.executeUpdate();
    } catch (Exception e) {
        System.out.println("Esto sale en pantalla\n");
        System.out.println(e.getMessage());
        throw e;
    } finally {
        this.Cerrar();
    }
}
```

Esta función lo que hace es recibir un objeto con los parámetros del objeto a eliminar, seguidamente hace una llamada a la función de la base de datos, para tomar el id de la entidad a eliminar, tras hacer esto ejecuta la función para después recibir una respuesta de éxito o error.

BEAN(Controlador de la vista, gestiona eventos emitidos por parte de la interfaz)

Esta capa es una capa que será la responsable de contralar la vista, todos los eventos y variables que aparecen en la vista, así como la llamada de las funciones según sea el caso. A esta capa se la conoce como la capa bean, a continuación, como ejemplo mostraremos el funcionamiento de la capa vean de cantón.

FUNCION BEAN INSERTAR, MODIFICAR, ELIMINAR, LISTAR

```
public void registrar() throws Exception {
    CantonDAO dao;
    try {
        dao = new CantonDAO();
        if (dao.registrar(canton)) {
            this.listar();
            FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(FacesMessage.SEVERITY_
        ) else {
            FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(FacesMessage.SEVERITY_
        )
    } catch (Exception e) {
        System.out.println("Error..\n");
        throw e;
    }
}
```

En esta función lo que se hace es llamar a las funciones que se encuentran en la capa del dao, para ello se reciben los parámetros en forma de objeto, se hace una instancia a la capa dao, tras esta llamada se hace un llamado a la función enviado como parámetros el objeto de la entidad a la que pertenezca, luego se ejecuta la función para recibir una respuesta de éxito o error según sea el caso.

VISTA

En esta capa es la cara de la aplicación web, dentro de esta se gestiona de una mejor forma todos los recursos como las imágenes y el diseño de cada uno de los formularios, para ello como muestra para la explicación del funcionamiento de las vistas tenemos la vista de cantón que nos servirá como ejemplo.

FORMULARIO VISTA XHTML

```
<ui:define name="content">
  <p:fieldset legend="Formulario Canton">
    <h:form id="form1">
      <p:messages id="messages" showDetail="true" autoUpdate="true" closable="true" />
      <p:commandButton icon="ui-icon-plusthick" value="Nuevo" oncomplete="PF('wdlgDatos').show()" update="dlgD">
      <p:commandButton icon="ui-icon-note" value="Listar" action="#{cantonBean.listar()}" update="data"/>

      <p:separator/></p:separator>
      <p:dataTable id="data" var="canton" value="#{cantonBean.lstCanton}"
        rows="8"
        paginator="true"
        paginatorTemplate="{CurrentPageReport} {FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}"
        currentPageReportTemplate="Pagina: {currentPage} de {totalPages}">
        <p:column headerText="Descripción">
          <p:outputLabel value="#{canton.descripcion_canton}"/>
        </p:column>
        <p:column headerText="Provincia">
          <p:outputLabel value="#{canton.provincia.descripcion_provincia}"/>
        </p:column>
        <p:column headerText="Acciones">
          <p:commandButton icon="ui-icon-pencil" actionListener="#{cantonBean.leerId(canton)}" oncomplete="PF('wdlgDatos').show()" update="data">
          <p:commandButton icon="ui-icon-trash" actionListener="#{cantonBean.eliminar(canton)}" update="data">
          <p:confirm header="Confirmation" message="Esta seguro de eliminar el registro?" icon="ui-icon-alert">
          </p:confirm>
          <p:confirmDialog global="true" showEffect="fade" hideEffect="fade">
            <p:commandButton value="Yes" type="button" styleClass="ui-confirmdialog-yes" icon="ui-icon-check">
            <p:commandButton value="No" type="button" styleClass="ui-confirmdialog-no" icon="ui-icon-close">
          </p:confirmDialog>
        </p:column>
      </p:dataTable>
    </h:form>
  </p:fieldset>
```

La estructura de la capa de esta vista está construida de tal manera que haga uso de todas las funciones de crud. Se puede ver que para crear un nuevo objeto se muestra un botón para llamar a la función de creación. Para mostrar los datos se muestra una tabla que hace una llamada a la función justo en el post constructor de la capa vean. Dentro de la tabla también se muestran los botones de modificación y eliminación.

Reportes

En esta capa se puede encontrar la capa de los reportes, así como también la gestión de los recursos como las imágenes y el diseño de cada uno de los reportes, para ello como muestra para la explicación del funcionamiento de la creación de los reportes para una mejor explicación tenemos la vista del reporte historia clínica que nos servirá como ejemplo.

FORMULARIO VISTA LLAMADA AL REPORTE

```
<p:column headerText="Acciones">
  <p:commandButton icon="ui-icon-print" ajax="false"
    actionListener="#{registroBean.verReporteHistoria(registro.id_registro)}" />
```

Se puede ver dentro del código que para llamar a un nuevo reporte se muestra un botón

para llamar a la función de creación (). Para mostrar los datos se muestra una tabla que hace una llamada a la función justo en el post constructor de la capa view. Dentro de la tabla también se muestran los botones de modificación y eliminación.

BEAN DEL REPORTE

```
public void verReporteHistoria(int id_historia)
    throws ClassNotFoundException, InstantiationException, IllegalAccessException, SQLException {

    System.out.print(id_historia);

    //Instancia hacia la clase reporteClientes
    cReporte rCliente = new cReporte();
    FacesContext facesContext = FacesContext.getCurrentInstance();
    ServletContext servletContext = (ServletContext) facesContext.getExternalContext().getContext();
    String ruta = servletContext.getRealPath("/reportes/reporteHistoriaClinica.jasper");

    rCliente.getReporteConsulta(ruta, id_historia);
    FacesContext.getCurrentInstance().responseComplete();
}
```

Dentro de este archivo se encuentra la llamada al reporte que se encuentra hecho con iReport, una herramienta de NetBeans para la creación de los mismos.

La función `verReporteHistoria` recibe como parámetro el identificador de la historia clínica (`id_historia`), luego se hace una instancia de la clase `cReporte`, ya que esta es la encargada de la creación de reportes. A continuación, lo que se hace es enviar por parámetros la ruta donde se encuentra el reporte junto con el identificador de la historia clínica.

CLASE REPORTE (CREPORTE)

```
public void getReporteConsulta(String ruta, Integer idC) throws ClassNotFoundException, InstantiationException,
Connection conexion;
Class.forName("com.mysql.jdbc.Driver").newInstance();
conexion = DriverManager.getConnection("jdbc:mysql://localhost:3306/bdhc", "root", "");
//Se definen los parametros si es que el reporte necesita
Map parameter = new HashMap();

parameter.put("id_con", idC);

try {
    File file = new File(ruta);

    HttpServletResponse httpResponse = (HttpServletResponse) FacesContext.getCurrentInstance().getEx

    httpResponse.setContentType("application/pdf");
    httpResponse.setHeader("Content-Type", "application/pdf");

    net.sf.jasperreports.engine.JasperReport jasperReport = (net.sf.jasperreports.engine.JasperReport) JRLo

    JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, parameter, conexion);

    JRExporter jrExporter = null;
    jrExporter = new JRPdfExporter();
    jrExporter.setParameter(JRExporterParameter.JASPER_PRINT, jasperPrint);
    jrExporter.setParameter(JRExporterParameter.OUTPUT_STREAM, httpResponse.getOutputStream());

    jrExporter.setParameter(JRExporterParameter.OUTPUT_STREAM, httpResponse.getOutputStream());

    if (jrExporter != null) {
        try {
            jrExporter.exportReport();
        } catch (JRException e) {
            e.printStackTrace();
        }
    }

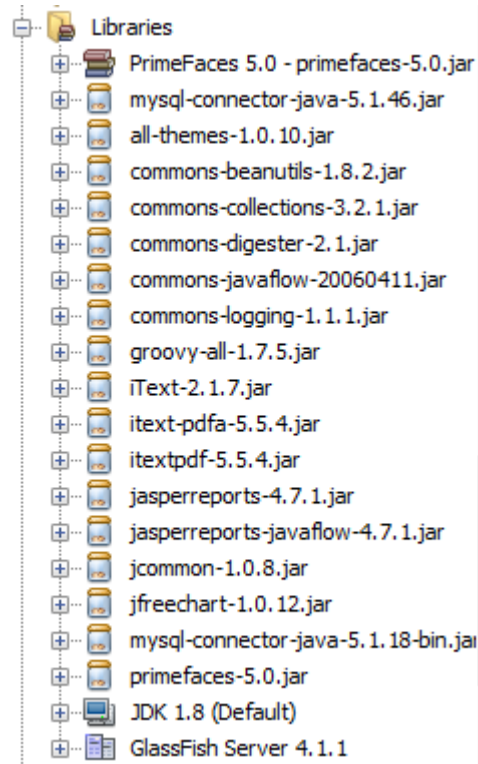
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if (conexion != null) {
        try {
            conexion.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
}
```



Dentro de este archivo se encuentra la clase llamada cReporte, encargada de la creación de reportes. Dentro de esta clase se encuentra la función getReporteConsulta que recibe como parámetros la ruta donde se encuentra el reporte creado, así como el identificador de los datos de la consulta.

Dentro del cuerpo de la función, se encuentra instanciado el driver de conexión para conexión con a base de datos, continuamos con el mapeo de los parámetros como el identificador de la consulta que lleva el alias id_con. Se crea una instancia de la clase File para la creación de un nuevo archivo, posteriormente agregamos las cabeceras señalando que es un archivo pdf el que se va a crear. A continuación, hacemos una llamada la ubicación del reporte.

Creación de Ireport.

Es necesario bajarse las librerías de irepor para neatbeans que se muestran a continuación.



Para la creación es necesario primero crear una conexión a la base de datos, para lo cual tenemos que hacer clic en el siguiente icono de  HC .

A continuación, se nos desplegará un formulario en el cual nos pedirá, el nombre de la conexión, el driver de conexión, la url de la base de datos, así como las credenciales necesarias para la conexión a la base de datos.

Database JDBC connection

Name

JDBC Driver

JDBC URL

Credentials

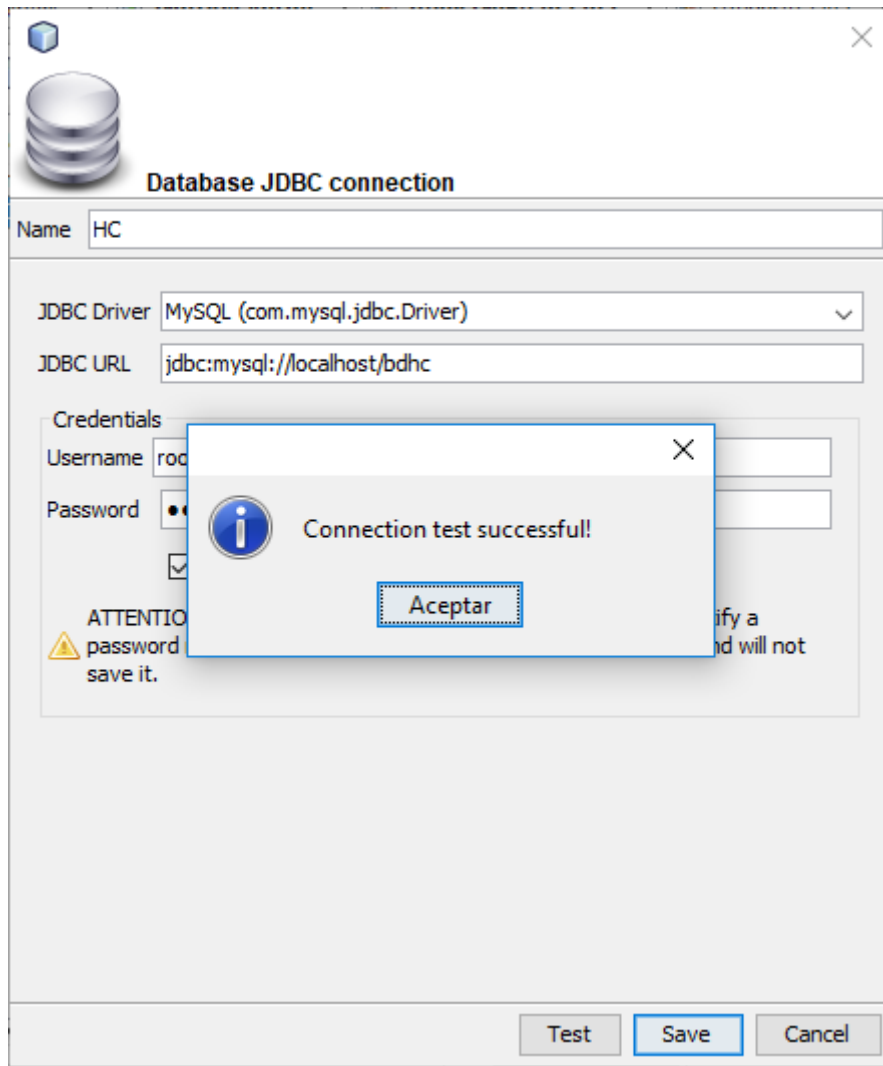
Username


Password

Save password

ATTENTION! Passwords are stored in clear text. If you dont specify a password now, iReport will ask you for one only when required and will not save it.

A continuación, realizamos un test de conexión hacia la base de datos.



Luego procedemos con la creación de la consulta que nos traerá los datos para este reporte, para ello hacemos clic en este icono , a continuación creamos la consulta en el siguiente espacio.

Report query

Report query | JavaBean Datasource | DataSource Provider | CSV Datasource | Excel Datasource

Query language: SQL

Load query | Save query

```

SELECT * FROM `registro` as r
INNER JOIN inicio_atencion as i on i.registro_id =
r.id_registro
INNER JOIN avieq as a on a.registro_id = r.id_registro
INNER JOIN apf as ap on ap.registro_id = r.id_registro
INNER JOIN ears as e on e.registro_id = r.id_registro
INNER JOIN parroquia as p on p.id_parroquia =
r.parroquia
INNER JOIN canton as c on c.id_canton = p.canton_id
INNER JOIN provincia as pr on pr.id_provincia =
c.provincia_id
INNER JOIN motivo as m on m.id_motivo =i.motivo_id
INNER JOIN accidentes as acc on acc.id_accidente =
a.accidente_id

```

Fields provider for sql queries ready.

Automatically Retrieve Fields | Read Fields | Qu... | Send...

Drag a parameter into the query editor to add a parameter. Hold CTL to add multiple parameters as query chunks.

Available parameters

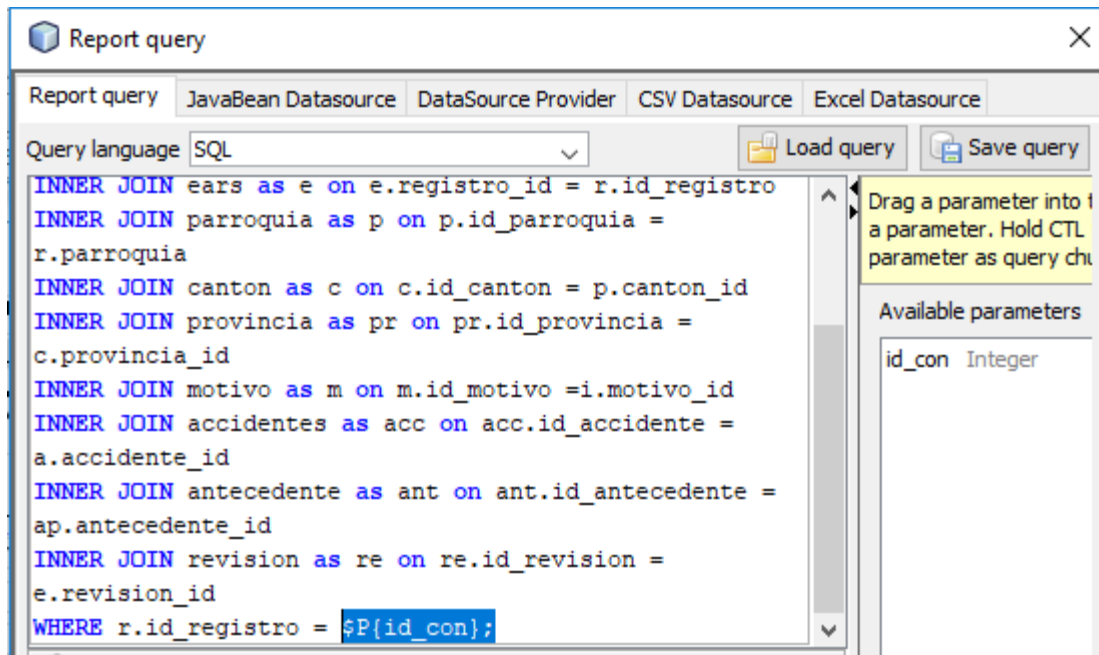
id_con Integer

New parameter

Field name	Field type	Description
id_registro	java.lang.Integer	
apellido_paterno	java.lang.String	
apellido_materno	java.lang.String	
primer_nombre	java.lang.String	
segundo_nombre	java.lang.String	
...	java.lang.String	

Filter expression... | Sort options... | Preview data ▼ | OK | Cancel

Posteriormente agregamos los parámetros dentro de la consulta junto con un \$.



Guardamos esta consulta y procedemos con el diseño del reporte, con la agregación de imágenes así como con la creación de etiquetas del reporte, agregamos las variables que nos trae la consulta.

Historia Clínica

Dr. Darío Javier Vargas Amaguaya

Historia Clínica : \${numero_historia}

Apellido Paterno :	\${apellido_paterno}	Apellido Materno :	\$F	Primer Nombre :	\${primer_nombre}
Segundo Nombre :	\$F	# Cedula :	\${ci}	Dirección :	\${direccion}
Barrio :	\${barrio}	Parroquia :	\$F	Cantón :	\$F
Provincia :	\$F	Zona :	\${zona}	# Teléfono :	\${telefono}
Fecha de N :	\$F	Lugar de N :	\${lugar_nacimiento}	Nacionalidad :	\${nacionalidad}
Grupo Cultural :	\${grupo_cultural}	Edad :	\${edad}	Genero :	\${genero}
Estado Civil :	\${estado_civil}	Instrucción :	\${instruccion}	Fecha Admision :	\${fecha_admision}
Ocupación :	\${ocupacion}	Empresa Trabaja :	\$F	Tipo Seg. Sa :	\${tipo_seguro}
		En caso nesasario llamar a :	\${contacto}	Parentesco :	\${parentesco}
Dirección :	\$F	# Telefono :	\$F	Forma de llegar :	\${forma_llegar}
Institu. form. entre :	\$F	# Telefono :	\${telefono_entrega}		

2.- INICIO DE ATENCIÓN Y MOTIVO

Hora :	\$F	Trauma :	\$F	Grupo Sanguineo :	\$F	Notificacion Policia :	\$F
3.- ACCIDENTE, VIOLENCIA, INTOXICACION, ENVENENAMIENTO O QUEMADURA							
Fecha y Hora del evento :	\$F	Lugar del evento :	\$F	Dirección del evento :	\$F		
Accidente :	\${descripcion_accidente}				Custodia policial :	\${cutodia_policial}	

Para ver si todo ha salido bien necesitamos realizar una pre visualización, para ello es necesario hacemos clic en esta icono [Preview](#) . El cual nos da una pre visualización de como quedara el reporte al cual llamaremos posteriormente.

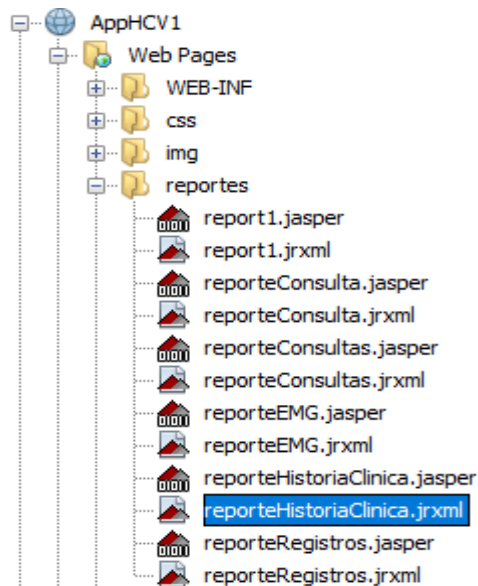
Historia Clínica

Dr. Darío Javier Vargas Amaguaya

Historia Clínica : 101

Apellido Paterno :	<u>Gallegos</u>	Apellido Materno :	<u>Carrillo</u>	Primer Nombre :	<u>Katy</u>
Segundo Noombre :	<u>Maribel</u>	# Cedula :	<u>0604585236</u>	Dirección :	<u>Av. Guayaquil y</u>
Barrio :	<u>La Dolorosa</u>	Parroquia :	<u>Maldonado</u>	Cantón :	<u>RIOBAMBA</u>
Provincia :	<u>Chimborazo</u>	Zona :	<u>1</u>	# Teléfono :	<u>032655988</u>
Fecha de N :	<u>1/08/08 0:00</u>	Lugar de N :	<u>Riobamba</u>	Nacionalidad :	<u>Ecuatoriana</u>
Grupo Cultural :	<u>Mestizos</u>	Edad :	<u>20</u>	Genero :	<u>masculino</u>
Estado Civil :	<u>Soltero</u>	Instruccion :	<u>2000</u>	Fecha Admicion :	<u>14/08/18 0:00</u>
Ocupación :	<u>0</u>	Empresa Trabaja :	<u>ESPOCH</u>	Tipo Seg. Sa :	<u>SOCIAL</u>
		En caso nesesarrio llamar a :	<u>William Gallegos</u>	Parentesco :	<u>Papa</u>

Para finalizar es necesario, que tanto el compilado como el archivo .jasper lo coloquemos en el siguiente directorio,



Donde a través de código java, lo llamaremos mediante un botón que se encontrar en la vista de cualquier formulario deseado.



Manual de Usuario del Sistema

Introducción

El propósito de este Manual es facilitar al usuario la operación de las diferentes etapas del sistema se puede consultar la información y administra el Sistema Informático, en la primera parte encontramos los módulos como el de administrador y usuario mostrando las diferentes opciones con las que se puede trabajar.

En el módulo de Administrador encontramos el login que es el inicio del sistema, el cual ingresando un usuario y contraseña se despliega un menú principal con diferentes opciones que ayudan en todo el proceso de registro de un paciente y la edición de las diferentes opciones.

En el módulo de Usuario ingresamos de la misma forma como en la del administrador con la diferencia que el modulo estrictamente se utiliza para el registro de los pacientes siguiendo los pasos que el sistema genera.

Antes de comenzar a hacer uso concreto del software, le recomendamos leer atentamente las instrucciones de este Manual, ya que las mismas le servirán para realizar las operaciones con mayor facilidad y seguridad. Esperamos que este desarrollo le sea de gran utilidad y signifique un valor agregado.

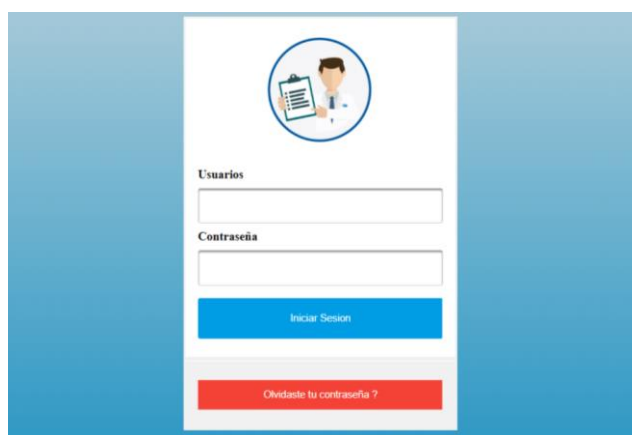
MÓDULO	SIGNIFICADO
ADMINISTRADOR	<p data-bbox="571 259 1321 342">Este módulo permite al administrador trabajar con los siguientes mantenimientos.</p> <ol data-bbox="635 365 965 1126" style="list-style-type: none"> <li data-bbox="635 365 965 398">1. Registro de Admisión <li data-bbox="635 421 834 454">2. Emergencia <li data-bbox="635 477 882 510">3. Nueva consulta <li data-bbox="635 533 930 566">4. Reporte de registro <li data-bbox="635 589 946 622">5. Reporte de Consulta <li data-bbox="635 645 866 678">6. Configuración <ul data-bbox="730 701 906 1126" style="list-style-type: none"> <li data-bbox="730 701 906 734">• Provincia <li data-bbox="730 757 866 790">• Cantón <li data-bbox="730 813 906 846">• Parroquia <li data-bbox="730 869 890 902">• Usuarios <li data-bbox="730 925 850 958">• Perfil <li data-bbox="730 981 890 1014">• Revisión <li data-bbox="730 1037 906 1070">• Accidente <li data-bbox="730 1093 938 1126">• Antecedente <p data-bbox="571 1216 1362 1299">Por cada mantenimiento tendremos disponibles las acciones de guardar, editar, eliminar, listar, etc.</p>

USUARIO	<p>Este módulo permite al usuario trabajar con los siguientes aspectos.</p> <ol style="list-style-type: none">1. Registro de Admisión2. Emergencia3. Nueva consulta4. Reporte de registro5. Reporte de Consulta <p>Tenderemos disponibles las acciones de guardar, editar, eliminar, listar.</p>
----------------	--

MÓDULO DEL ADMINISTRADOR

El módulo del administrador da la facilidad de trabajar con todas las tablas del sistema, entre las cuales se pueden realizar diferentes acciones, para poder mostrar el funcionamiento del sistema y su manejo sea correcto y eficiente.

LOGIN



The image shows a login interface with a white background and a blue gradient border. At the top center is a circular icon of a doctor with a clipboard. Below the icon are two input fields: the first is labeled 'Usuarios' and the second is labeled 'Contraseña'. Under the 'Contraseña' field is a blue button with the text 'Iniciar Sesión'. At the bottom of the form is a red button with the text 'Olvidaste tu contraseña?'.

Para usar cualquier funcionalidad del sistema, el usuario que desea ejecutarla debe estar autenticado. Para loguearse, deberá proveer sus credenciales (código de usuario y contraseña).

MENU PRINCIPAL

Dentro del menú principal encontraremos las diferentes opciones con las que podremos trabajar.





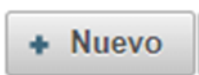



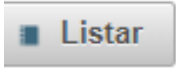

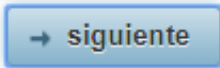

En la parte izquierda de nuestro menú principal tendremos el menú con las opciones de cada mantenimiento como lo son:

1. Registro de Admisión
2. Emergencia
3. Nueva consulta
4. Reporte de registro
5. Reporte de Consulta
6. Configuración
 - Provincia
 - Cantón

- Parroquia
- Usuarios
- Perfil
- Revisión
- Accidente
- Antecedente

BOTONES GENERALES

BOTÓN	SIGNIFICADO
	Permite verificar si el usuario y contraseña son las correctas para poder acceder al menú principal.
	Permite cancelar una opción, y retroceder a la pantalla principal donde estamos anteriormente.
	de Permite eliminar algún registro
	Permite guardar un registro
	Permite abrir paso para crear un nuevo registro.
	Permite realizar una búsqueda de los registros que tenemos en dicha tabla.

	Verifica los registros existentes
	Permite editar los registros
	Permite continuar con los procesos de registro
	Permite visualizar en pdf

EXPLICACION DE UN MANTENIMIENTO

MANTENIMIENTO DE CARGO																																																																			
ACCION	<div data-bbox="922 1070 1118 1144" style="text-align: center;">  </div> <p style="text-align: center;">NUEVO REGISTRO</p>																																																																		
<div data-bbox="252 1301 1337 1989" style="border: 1px solid gray; padding: 5px;"> <p>Datos</p> <p>1 2 3 4 5</p> <p>Registro de Admisión</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Historia # : *</td> <td style="width: 15%;"><input type="text" value="# historia"/></td> <td style="width: 15%;">Apellido Paterno *</td> <td style="width: 15%;"><input type="text" value="Apellido Paterno"/></td> <td style="width: 15%;">Apellido Materno: *</td> <td style="width: 15%;"><input type="text" value="Apellido Materno"/></td> </tr> <tr> <td>Primer Nombre: *</td> <td><input type="text" value="Primer Nombre"/></td> <td>Segundo Nombre: *</td> <td><input type="text" value="Segundo Nombre"/></td> <td>Cédula: *</td> <td><input type="text" value="Cédula"/></td> </tr> <tr> <td>Dirección: *</td> <td><input type="text" value="Dirección"/></td> <td>Barrio *</td> <td><input type="text" value="Barrio"/></td> <td>Parroquia</td> <td><input type="text" value="Seleccione"/></td> </tr> <tr> <td>Zona: *</td> <td><input type="text" value="Zona"/></td> <td>Telefono: *</td> <td><input type="text" value="032XXXXXX"/></td> <td>Fecha Nacimiento: *</td> <td><input type="text"/></td> </tr> <tr> <td>Lugar de Nacimiento: *</td> <td><input type="text" value="Lugar de Nacimiento"/></td> <td>Nacionalidad: *</td> <td><input type="text" value="Nacionalidad"/></td> <td>Grupo Cultural:</td> <td><input type="text"/></td> </tr> <tr> <td>Edad:</td> <td><input type="text" value="Edad"/></td> <td>Genero:</td> <td><input type="text"/></td> <td>Estado Civil:</td> <td><input type="text" value="Soltero"/></td> </tr> <tr> <td>Instrucción Ultimo año: *</td> <td><input type="text" value="Instrucción"/></td> <td>Fecha Admisión: *</td> <td><input type="text"/></td> <td>Ocupacion:</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Empresa Trabaja:</td> <td><input type="text" value="N/D"/></td> <td>Tipo Seguro Salud:</td> <td><input type="text" value="N/D"/></td> <td></td> <td></td> </tr> </table> <p>Datos de contacto</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">En caso necesario llamar a: *</td> <td style="width: 15%;"><input type="text" value="Contacto"/></td> <td style="width: 15%;">Parentesco: *</td> <td style="width: 15%;"><input type="text" value="Parentesco"/></td> <td style="width: 15%;">Direccion Contacto: *</td> <td style="width: 15%;"><input type="text" value="Dirección"/></td> </tr> <tr> <td>Telefono Contacto: *</td> <td><input type="text" value="09XXXXXXXX"/></td> <td>Formas de llegar:</td> <td><input type="text"/></td> <td>Intitucion Entrega: *</td> <td><input type="text" value="Institución"/></td> </tr> <tr> <td>Telefono Institucion: *</td> <td><input type="text" value="Telefono"/></td> <td></td> <td></td> <td></td> <td></td> </tr> </table> <p style="text-align: right;"></p> </div>		Historia # : *	<input type="text" value="# historia"/>	Apellido Paterno *	<input type="text" value="Apellido Paterno"/>	Apellido Materno: *	<input type="text" value="Apellido Materno"/>	Primer Nombre: *	<input type="text" value="Primer Nombre"/>	Segundo Nombre: *	<input type="text" value="Segundo Nombre"/>	Cédula: *	<input type="text" value="Cédula"/>	Dirección: *	<input type="text" value="Dirección"/>	Barrio *	<input type="text" value="Barrio"/>	Parroquia	<input type="text" value="Seleccione"/>	Zona: *	<input type="text" value="Zona"/>	Telefono: *	<input type="text" value="032XXXXXX"/>	Fecha Nacimiento: *	<input type="text"/>	Lugar de Nacimiento: *	<input type="text" value="Lugar de Nacimiento"/>	Nacionalidad: *	<input type="text" value="Nacionalidad"/>	Grupo Cultural:	<input type="text"/>	Edad:	<input type="text" value="Edad"/>	Genero:	<input type="text"/>	Estado Civil:	<input type="text" value="Soltero"/>	Instrucción Ultimo año: *	<input type="text" value="Instrucción"/>	Fecha Admisión: *	<input type="text"/>	Ocupacion:	<input type="checkbox"/>	Empresa Trabaja:	<input type="text" value="N/D"/>	Tipo Seguro Salud:	<input type="text" value="N/D"/>			En caso necesario llamar a: *	<input type="text" value="Contacto"/>	Parentesco: *	<input type="text" value="Parentesco"/>	Direccion Contacto: *	<input type="text" value="Dirección"/>	Telefono Contacto: *	<input type="text" value="09XXXXXXXX"/>	Formas de llegar:	<input type="text"/>	Intitucion Entrega: *	<input type="text" value="Institución"/>	Telefono Institucion: *	<input type="text" value="Telefono"/>				
Historia # : *	<input type="text" value="# historia"/>	Apellido Paterno *	<input type="text" value="Apellido Paterno"/>	Apellido Materno: *	<input type="text" value="Apellido Materno"/>																																																														
Primer Nombre: *	<input type="text" value="Primer Nombre"/>	Segundo Nombre: *	<input type="text" value="Segundo Nombre"/>	Cédula: *	<input type="text" value="Cédula"/>																																																														
Dirección: *	<input type="text" value="Dirección"/>	Barrio *	<input type="text" value="Barrio"/>	Parroquia	<input type="text" value="Seleccione"/>																																																														
Zona: *	<input type="text" value="Zona"/>	Telefono: *	<input type="text" value="032XXXXXX"/>	Fecha Nacimiento: *	<input type="text"/>																																																														
Lugar de Nacimiento: *	<input type="text" value="Lugar de Nacimiento"/>	Nacionalidad: *	<input type="text" value="Nacionalidad"/>	Grupo Cultural:	<input type="text"/>																																																														
Edad:	<input type="text" value="Edad"/>	Genero:	<input type="text"/>	Estado Civil:	<input type="text" value="Soltero"/>																																																														
Instrucción Ultimo año: *	<input type="text" value="Instrucción"/>	Fecha Admisión: *	<input type="text"/>	Ocupacion:	<input type="checkbox"/>																																																														
Empresa Trabaja:	<input type="text" value="N/D"/>	Tipo Seguro Salud:	<input type="text" value="N/D"/>																																																																
En caso necesario llamar a: *	<input type="text" value="Contacto"/>	Parentesco: *	<input type="text" value="Parentesco"/>	Direccion Contacto: *	<input type="text" value="Dirección"/>																																																														
Telefono Contacto: *	<input type="text" value="09XXXXXXXX"/>	Formas de llegar:	<input type="text"/>	Intitucion Entrega: *	<input type="text" value="Institución"/>																																																														
Telefono Institucion: *	<input type="text" value="Telefono"/>																																																																		

Al hacer clic en la opción de nuevo se despliega una ventana donde se ingresa la información necesaria.

ACCION



EDICIÓN DE UN REGISTRO

The screenshot shows a web application interface for editing a patient record. The interface is divided into a navigation menu on the left and a main form area. The navigation menu includes options like 'Principal', 'PACIENTES', 'CONSULTAS', 'REPORTES', and 'CONFIGURACION'. The main form area is titled 'DatosEdicion' and contains a 'Edición de Admisión' tab. The form fields are as follows:

Edición de Admisión					
Historia # : *	101	Apellido Paterno *	Gallegos	Apellido Materno : *	Carrillo
Primer Nombre : *	Katy	Segundo Nombre : *	Maribel	Cédula : *	0604585236
Dirección : *	Av. Guayaquil y Puruha	Barrio *	La Dolorosa	Parroquia	Veloz
Zona : *	1	Telefono : *	032655988	Fecha Nacimiento : *	2008-08-01
Lugar de Nacimiento : *	Riobamba	Nacionalidad : *	Ecuadoriana	Grupo Cultural :	Mestizos
Edad :	20	Genero :	Masculino	Estado Civil :	Soltero
Instrucción Ultimo año : *	2000	Fecha Admisión : *	2018-08-14	Ocupacion : *	true
Empresa Trabaja : *	ESPOCH	Tipo Seguro Salud : *	SOCIAL	En caso necesario llamar a : *	William Gallegos
Parentesco : *	Papa	Direccion Contacto : *	Av. Guayaquil y Puruha	Telefono Contacto : *	0985632569
Formas de llegar:	Ambulatorio	Intitucion Entrega : *	Bomberos	Telefono Institucion : *	032566988

Buttons: Cancelar, siguiente

2.-INICIO DE ATENCIÓN Y MOTIVO

Horas : 10:09

Antecedente : TRAUMA

Grupo Sanguíneo : ORH+

Observaciones :

Cancelar

siguiente

Apellido	Apellido	Dirección	Acciones
Gallegos Carrillo	Katy Maribel	Av. Guayaquil y Puruha	[icon] [icon] [icon]
Roma Salas	Fredy Waldon	Av. Primera Constituyente	[icon] [icon] [icon]
perez villa	will mario	riobamba	[icon] [icon] [icon]
valle viñan	william geovanny	riobamba	[icon] [icon] [icon]
uyuy fghfgh	uuuu yuyuyu	yyyytjuytuy	[icon] [icon] [icon]

3.-ACCIDENTE, VIOLENCIA, INTOXICACION, ENVENENAMIENTO O QUEMADURA

Fecha emergencia : 03/09/2018

Hora emergencia : 07:15

Lugar emergencia : * Ouyoyu

Dirección emergencia : * jklhklgh

Custodia Policial :

Accidente : CAIDA

Observaciones : *

Aliento Etílico :

Valor Alcocheck : 0.0

Cancelar

siguiente

Apellido	Dirección	Acciones
Katy Maribel	Av. Guayaquil y Puruha	[icon] [icon] [icon]
Fredy Waldon	Av. Primera Constituyente	[icon] [icon] [icon]
will mario	riobamba	[icon] [icon] [icon]
william geovanny	riobamba	[icon] [icon] [icon]
uuuu yuyuyu	yyyytjuytuy	[icon] [icon] [icon]

4.-ANTECEDENTES PERSONALES Y FAMILIARES

Antecedente : ALERGICO

Observacion antecedente :

Cancelar

siguiente

Apellido	Apellido	Dirección	Acciones
Gallegos Carrillo	Katy Maribel	Av. Guayaquil y Puruha	[icon] [icon] [icon]
Roma Salas	Fredy Waldon	Av. Primera Constituyente	[icon] [icon] [icon]
perez villa	will mario	riobamba	[icon] [icon] [icon]
valle viñan	william geovanny	riobamba	[icon] [icon] [icon]
uyuy fghfgh	uuuu yuyuyu	yyyytjuytuy	[icon] [icon] [icon]

Al guardar la información si es necesario editarla existe una opción que permite desplegar la información y cambiarla.

ACCION



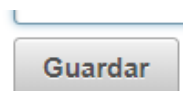
ELIMINACION DE UN REGISTRO

The screenshot shows a web application interface for a hospital (HCD). The main content area displays a table of patient records under the heading 'Formulario de Admisión'. A confirmation dialog box is overlaid on the table, asking '¿Esta seguro de eliminar el registro?' (Are you sure you want to delete the record?). The dialog has 'Yes' and 'No' buttons. The table has columns for '# Historia', 'CI', 'Apellido', 'Apellido', 'Direccion', and 'Acciones'. The data rows are as follows:

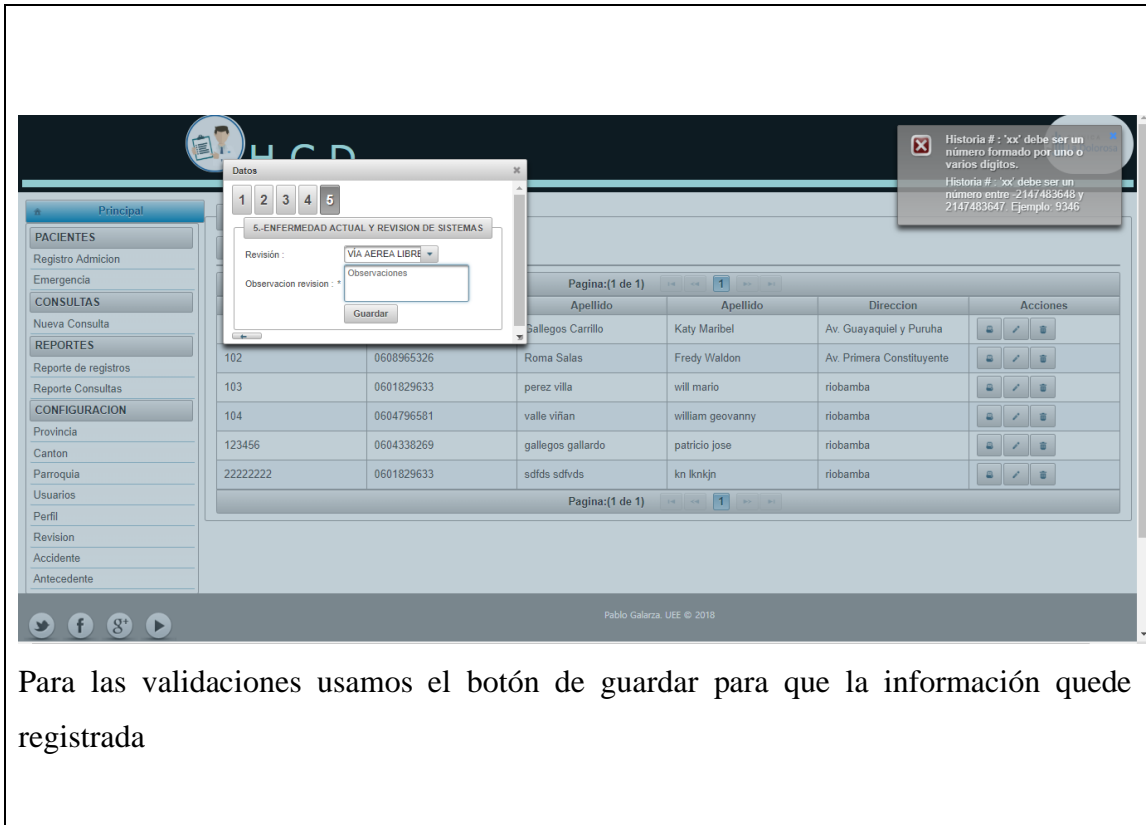
# Historia	CI	Apellido	Apellido	Direccion	Acciones
101	0604585236	...	aty Maribel	Av. Guayaquil y Puruha	[Icons]
102	0608965326	...	edy Waldon	Av. Primera Constituyente	[Icons]
103	0601829633	...	li mario	riobamba	[Icons]
104	0604796581	...	illiam geovanny	riobamba	[Icons]
123456	0604338269	galegos gallardo	patricio jose	riobamba	[Icons]
22222222	0601829633	sdfds sdfds	kn lknkn	riobamba	[Icons]

Una vez guardada la información y si deseamos eliminarla pulsamos la opción de borrar y se desplegará un mensaje el cual nos preguntara la confirmación si deseamos eliminar dicho registro de la tabla cargo. Dependiendo de la respuesta sí o no podremos ejecutar dicha acción.

ACCION



VALIDACIONES



Para las validaciones usamos el botón de guardar para que la información quede registrada



Debemos insertar la en el filtro de búsqueda para poder encontrar los datos correspondientes, se deberá ingresar el número de cedula y poder verificar los datos del paciente.

MANTENIMIENTOS GENERALES

La configuración nos ayuda a cambiar según las necesidades diferentes opciones que nos permita un manejo óptimo del sistema que ayude a un mejor desarrollo y máxima optimización de tiempo y recursos.

Provincia

Formulario Registro Admicion

+ Nuevo | Listar

Descripción	Acciones
Chimborazo	[edit] [delete]
Tungurahua	[edit] [delete]
Chimborazos	[edit] [delete]
Chimborazos2	[edit] [delete]
Chimborazos3	[edit] [delete]
Chimborazos4	[edit] [delete]
Chimborazos5	[edit] [delete]

Pagina: (1 de 1)

En la opción de provincia podemos editar o añadir o eliminar más opciones según sea el caso

Cantón

Formulario Canton

+ Nuevo | Listar

Descripción	Provincia	Acciones
RIOBAMBA	Chimborazo	[edit] [delete]
Ambato	Tungurahua	[edit] [delete]

Pagina: (1 de 1)

En la parte izquierda la opción de cantón permite aumentar o eliminar listar cantones según se requiera

Parroquia

The screenshot shows a web application interface for 'Formulario Canton'. The header includes the 'HCD' logo and 'CLINICA La Dolorosa'. The left sidebar contains navigation menus for 'PACIENTES', 'CONSULTAS', 'REPORTES', and 'CONFIGURACION'. The main content area displays a table with the following data:

Descripción	Canton	Provincia	Acciones
Maldonado	RIOBAMBA	Chimborazo	[Edit] [Delete]
Lizarsaburu	RIOBAMBA	Chimborazo	[Edit] [Delete]
Veloz	RIOBAMBA	Chimborazo	[Edit] [Delete]

Navigation controls include 'Nuevo' and 'Listar' buttons, and pagination information: 'Pagina: (1 de 1)'.

Permite ingresar las distintas parroquias urbanas que existen en la ciudad listarlas y editarlas

PROYECTO DE TESIS
INSTITUTO TECNOLÒGICO SUPERIOR PARTICULAR
“SAN GABRIEL”



TÍTULO DEL PROYECTO

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL REGISTRO DE HISTORIAS CLÍNICAS DE LOS PACIENTES DE LA CLÍNICA LA DOLOROSA, DESARROLLADA EN JAVA Y GESTOR DE BASE DE DATOS MYSQL EN EL PERIODO 2017

PREVIA A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN INFORMÁTICA

MENCIÓN: ANÁLISIS EN SISTEMAS

PRESENTADO POR: PABLO GALARZA

ANALIZADO Y APROBADO POR LA COMISIÓN TÉCNICA:

Dra. Mirella Vera R. Mg. Sc.
VICERRECTORA ACADÈMICA

FECHA DE APROBACIÓN.....

INDICE

1. ASPECTOS GENERALES	137
1.1 TÍTULO DEL PROYECTO	137
1.2 PROPONENTE	137
1.3 ASESOR DOCENTE.....	137
1.4 LUGAR DE REALIZACIÓN.....	137
1.5 TIEMPO DE DURACIÓN.....	137
1.6. FECHA ESTIMADA DE INICIACIÓN.....	137
2. FORMULACIÓN GENERAL DEL PROYECTO DE TESIS	138
2.1. ANTECEDENTES:.....	138
2.2. JUSTIFICACIÓN:.....	139
2.3 OBJETIVOS	140
2.3.1 OBJETIVO GENERAL:	140
2.3.2 OBJETIVOS ESPECIFICOS:	140
2.4 MARCO TEÒRICO:.....	140
2.4.1 SISTEMA.....	140
2.4.3 JAVA	143
2.4.2 MYSQL	146
2.5 BIBLIOGRAFIA:.....	148
3. EJECUCIÓN DEL PROYECTO DE INVESTIGACIÓN	149
3.1 PLAN GENERAL DEL TRABAJO	149
3.1.1ANALISIS PRELIMINAR DEL PROBLEMA:.....	149
3.1.2 DESARROLLO DEL PROYECTO.....	152
DESEMPEÑO	153
DISPONIBILIDAD	153
USABILIDAD.....	153
ESCALABILIDAD.....	153
SEGURIDAD.....	153
3.2 HIPÒTESIS.....	155
3.3 VARIABLES	155
3.3.1 VARIABLE INDEPENDIENTE:	155
Tabla 4 Variable Independiente.....	156
3.3.2 VARIABLE DEPENDIENTE:.....	157

3.4	CRONOGRAMA TENTATIVO.....	158
3.5.	RECURSOS	159
3.5.1.	RECURSOS TÈCNICOS	159
3.5.2	RECURSOS HUMANOS:	160
3.5.3	RECURSOS MATERIALES:	59
3.6	MÈTODOS Y TÈCNICAS	160
3.6.1	MÈTODOS:.....	160
3.6.2	TÈCNICAS:.....	161
3.7	PRESUPUESTO:.....	161
3.8	FUENTE DE FINANCIAMIENTO:.....	162

TABLA DE CONTENIDOS

Tabla 1 Estudio de factibilidad Técnico	150
Tabla 2 Estudio de Factibilidad Operativo	151
Tabla 3 Estudio de Factibilidad Operativo	151
Tabla 4 Variable Independiente	156
Tabla 5 Variable Dependiente	157
Tabla 6 Recurso Técnico Hardware.....	159
Tabla 7 Recurso Técnico Software	159
Tabla 8 Recurso Humano.....	160
Tabla 9 Recurso Material.....	160
Tabla 10 Presupuesto	161

TABLA DE IMÁGENES

Ilustración 1 Interfaz grafica.....	154
Ilustración 2 Diseño lógico del sistema	155
Ilustración 3 Cronograma Tentativo	158

1. ASPECTOS GENERALES

1.1 TÍTULO DEL PROYECTO

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL REGISTRO DE HISTORIAS CLÍNICAS DE LOS PACIENTES DE LA CLÍNICA LA DOLOROSA, DESARROLLADA EN JAVA Y GESTOR DE BASE DE DATOS MYSQL EN EL PERIODO 2017

1.2 PROPONENTE

NOMBRE: PABLO RENATO GALARZA VALLEJO

ESPECIALIDAD: ANÁLISIS DE SISTEMAS

ÀREA: INFORMÁTICA

1.3 ASESOR DOCENTE

Ing. Ricardo Jaramillo

1.4 LUGAR DE REALIZACIÓN

Instituto Tecnológico Superior “San Gabriel”

1.5 TIEMPO DE DURACIÓN

6 meses

1.6 FECHA ESTIMADA DE INICIACIÓN

Enero 2018

1. FORMULACIÓN GENERAL DEL PROYECTO DE TESIS

2.1. ANTECEDENTES:

La Clínica La Dolorosa se encuentra ubicada en la provincia de Chimborazo cantón Riobamba barrio “La Joya”. Tiene una trayectoria formal y bien definida en cuanto a atención hacia pacientes desde el año 2007, desde que brinda sus servicios el volumen de información siempre ha sido extenso y algo tedioso este proceso es manejado por el centro médico además se encuentra en constante crecimiento lo cual hace cada vez, más complejo el seguimiento, manipulación y administración de la información requerida.

La Clínica desde que abrió las puertas al público a trabajado con un sistema manual tomando información general y específica de cada paciente teniendo a su vez una numeración para cada persona, y así que no exista confusión o se pierda información de la misma en el momento que se necesite del documento además su almacenamiento se encuentra por carpetas y su ubicación es en un cuarto donde solo puede ingresar personal autorizado estas historias clínicas reposan dentro de archiveros completamente llenos.

El método de registro manual de información ha sido utilizado por diferentes centros de salud con algunas variaciones sin embargo en la actualidad este método cada vez está siendo remplazado adoptando nuevas formas de registro sabiendo esto La Clínica la Dolorosa está interesada en innovar y mejorar el tratamiento de la documentación pensando así en un sistema que remplace el actual que ha quedado obsoleto.

El sistema de registro propuesto será oportuna para las personas que laboran en la clínica, brindándoles una herramienta completa que les facilite el seguimiento de sus pacientes mediante MYSQL que es un sistema de gestión de bases de datos relacional desarrollado bajo licencia y está considerada como la base datos open source más popular del mundo y JAVA un lenguaje de programación y una plataforma informática.

2.2. JUSTIFICACIÓN:

Un sistema de registro es una necesidad ya que suprime técnicas y procesos obsoletos que pueden llegar a retrasar determinadas actividades en una organización particular haciéndola competitiva frente a los distintos problemas y complicaciones que puede haber en el área de la salud.

Desde hace muchos años, las distintas organizaciones y empresas, en búsqueda de ser competitivas. Desde hace muchos años, las distintas organizaciones y empresas, en búsqueda de ser competitivas y debido a la continua evolución de la informática y todo lo relacionado con la tecnología, han ido incursionando en este mundo. La manera como se maneja la información ha ido cambiando y casi ha obligado a los directivos de toda institución a autorizar los procesos y a optimizar la forma en que se manipula la información. Tal el caso del Ministerio de Salud Pública que hace pocos años Implemento un sistema informático único de información (Sistema informático para la gestión integral de salud – SIGIS) ² facilitando el proceso.

La Clínica La Dolorosa estando consiente de la situación y sabiendo que depende del cliente ya que ellos son los mejores jueces al hablar de prestigio y calidad del servicio, una mala atención es un grave problema de incumplimiento del derecho humano a la salud y su forma de vida el sistema de registro pretende colaborar en estos procedimientos, de forma que se pueda hacer un debido uso optimizando los recursos con los que cuenta la clínica debido a que este sistema de registro significaría un gran ahorro de tiempo y optimización del personal implicado.

Con un sistema de registro para las historias clínicas no es necesario el cuarto que se utiliza actualmente lo únicamente necesario sería un computador donde se albergaría toda la información ahorrando espacio además no existiría el inconveniente de que se pierda documentos necesarios siendo así cada vez más eficientes y eficaces.

El proyecto tiene la finalidad de contribuir en el ingreso del paciente teniendo toda la información requerida obteniendo así un registro que se encargaría de administrar y ordenar los datos de cada paciente haciendo mucho más rápida la búsqueda de algún dato en específico llegando así a tener herramienta de trabajo muy práctica.

² Raymond Ministerio de Salud Pública. <http://www.salud.gob.ec/fortalecimiento-de-la-red-de-servicios-de-salud-y-mejoramiento-de-la-calidad/>

2.3 OBJETIVOS

2.3.1 OBJETIVO GENERAL:

- Desarrollar un sistema para el registro de historias clínicas de los pacientes de la clínica La Dolorosa, desarrollada en java y gestor de base de datos mysql.

2.3.2 OBJETIVOS ESPECIFICOS:

- Investigar los procedimientos y formatos usados en la Clínica La Dolorosa para el manejo de registro del paciente
- Análisis y diseño entidad relación
- Análisis y diseño de la interfaz en java
- Implementación del sistema en los equipos computacionales de la clínica
- Realizas las pruebas de funcionamiento
- Realizar los cambios requeridos en función a los resultados obtenidos en las pruebas de funcionamiento.
- Capacitar al personal que va a manejar el sistema.

2.4 MARCO TEÒRICO:

2.4.1 SISTEMA

2.4.1.1 CONCEPTOS DE SISTEMAS

El concepto de sistema es muy general. Podemos definir un sistema como un conjunto de elementos estructurados e interconectados de una manera definida. En otras palabras, es un conjunto de elementos asociados a un conjunto de relaciones entre elementos. Entre otros conceptos se encuentran los siguientes:

- “El sistema es un conjunto de elementos organizados que se encuentran integrados, que buscan alguna meta o metas comunes, operando para ello sobre datos o información sobre energía o materia u organismos en una referencia temporal para producir como salida información”.

- “Un sistema de control estudia la conducta del sistema con el fin de regularla de un modo conveniente para su supervivencia. Una de sus características es que sus elementos deben ser lo suficientemente sensitivos y rápidos como para satisfacer los requerimientos para cada función del control”.
- “Un sistema es un conjunto organizado de cosas o partes interactivas e interdependientes, que se relacionan formando un todo unitario y complejo”.³

2.4.1.2 ANTECEDENTES DE SISTEMAS

La Teoría General de Sistemas (TGS), surgió con los trabajos del biólogo alemán Ludwin Von Bertalanffy, publicados entre 1950 y 1968. La Teoría General de Sistemas no busca solucionar problemas o intentar soluciones prácticas, pero sí producir teoría y formulaciones conceptuales que puedan crear condiciones de aplicación en la realidad empírica.⁴

2.4.1.3 IMPORTANCIA DE LOS SISTEMAS

La importancia de los sistemas es debido al propósito de su implementación y radica en los resultados obtenidos entre los cuales se pueden mencionar:

- Evolución de la tecnología y la globalización de los mercados a nivel mundial.
- Hacen que las empresas para ser competitivas adopten las tecnologías a los diferentes procesos.
- Facilita el control interno.
- Sirve a la gerencia para tomar decisiones que permiten obtener la máxima rentabilidad de las utilidades.

2.4.1.4 CARACTERÍSTICAS DE LOS SISTEMAS

Los sistemas se caracterizan por ofrecer una diversidad de opciones, útiles para las personas o entes que las usan:

- Ofrece la información de acuerdo con las necesidades del usuario, distribuyéndola de la forma más conveniente.

³ SISTEMA DE INFORMACIÓN ADMINISTRATIVA. Robert G. Murdick con Jhon C. Munson. Segunda Edición. Prentice Hall.

⁴ TEORÍA GENERAL DE SISTEMAS. Von Gigch, Jhon p. Año de Edición 1993, Pág. 65.

- Proporciona los bienes adecuados en momentos oportunos.
- Genera y difunde información.
- Proporciona asesoría a clientes.
- Adaptados a las necesidades de cada cliente.
- Da seguridad garantizada en sus programas y bases de datos (Archivos).
- Da operación monousuarios o red.
- Entrega de manuales de operaciones y procedimientos administrativos.

2.4.1.5 CLASIFICACIÓN DE LOS SISTEMAS

Existe una gama de sistemas que pueden ir desde lo más simple a lo más complejo, lo cual depende de la estructura organizativa de la empresa y de los datos que deseamos obtener para el propósito establecido, entre los que se pueden mencionar los siguientes:

- Sistemas de nivel operacional o transaccional.
- Sistemas de automatización de oficina.
- Sistemas de nivel de talento o conocimiento.
- Sistemas de nivel gerencial (apoyo de decisiones y apoyo a los ejecutivos).
- Sistemas de nivel estratégico.
- Sistemas inteligentes.
- Sistemas naturales y artificiales: los naturales abundan en la naturaleza, la ecología de la vida y los artificiales aparecen en una infinita variedad alrededor de nosotros.
- Sistema social, hombre-máquina y mecánico: los integrados pueden considerarse como sistemas sociales puros, distinguiéndose de otros sistemas, objetivos y procesos; los hombres-maquinas en la actualidad, casi todos los hombres emplean equipo de una u otra clase en sus trabajos organizados; y por último los mecánicos deben obtener sus propias entradas y mantenerlas.
- Sistemas abiertos y cerrados: el sistema abierto es aquel que interactúa con su ambiente; el sistema cerrado es aquel donde el ambiente que rodea no cambia.
- Sistemas permanentes y temporales: los permanentes son aquellos que duran mucho más que las operaciones que ellos realizan; y los temporales están destinados a durar cierto período y luego desaparecen.⁵

⁵ TEORÍA GENERAL DE SISTEMAS. Von Gigch, Jhon p. Año de Edición 1993, Pág. 70.

- Sistemas estables y no estables: los estables son aquellas cuyas propiedades y operaciones no varían de manera importante. Los no estables son aquellos cuyas propiedades y operaciones cambian constantemente.
- Sistemas adaptivos y no adaptivos: el adaptivo es aquel que reacciona con un ambiente de tal manera que mejora su funcionamiento. Los no adaptivos son aquellos que no reaccionan oportunamente ante un cambio.
- Sistemas de procesos o sistemas productivos: es aquel que procesa las entradas para generar las salidas útiles a la sociedad.
- Sistemas de control de inventario: es aquel que controla grandes cantidades de dinero y numerosas transacciones diarias.

2.4.3 JAVA

CONCEPTO DE JAVA

Java es un lenguaje de programación originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y como un componente fundamental de la plataforma Java de Sun Microsystems. El lenguaje deriva mucho de su sintaxis de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos.

JAVA es un lenguaje de propósito general, por medio del cual se pueden hacer diferentes aplicaciones gráficas que corran sobre un escritorio (aplicaciones standalone), aplicaciones para dispositivos móviles, applets para navegadores, aplicaciones del lado de un servidor de páginas webs, etc.

Al día de hoy, la potencia de JAVA queda demostrada en los servidores de aplicaciones empresariales, cuya plataforma de desarrollo se denomina y especifica como J2EE, y en el desarrollo de programas que se ejecutan en dispositivos móviles como los PDAs o los celulares, principalmente juegos, y cuya plataforma de desarrollo se ha denominado J2ME.⁶

CARACTERÍSTICAS DE JAVA

Lenguaje Simple.- Se lo conoce como lenguaje simple porque viene de la misma estructura de c y c++; ya que c++ fue un referente para la creación de java por eso utiliza determinadas características de c++ y se han eliminado otras.

⁶ Ceballos, 2013

Orientado a Objeto.- Toda la programación en java en su mayoría está orientada a objeto, ya que al estar agrupados en estructuras encapsuladas es más fácil su manipulación.

Distribuido.- Permite abrir sockets, establecer y aceptar conexiones con los servidores o clientes remotos; facilita la creación de aplicaciones distribuidas ya que proporciona una colección de clases para aplicaciones en red.

Robusto.- Es altamente fiable en comparación con c, se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.

Seguro.- La seguridad es una característica muy importante en java ya que se han implementado barreras de seguridad en el lenguaje y en el sistema de ejecución de tiempo real.

Indiferente a la arquitectura.- Java es compatible con los más variados entornos de red, cualesquiera sean estos desde Windows 95, Unix a Windows Nt y Mac, para poder trabajar con diferentes sistemas operativos. Java es muy versátil ya que utiliza byte-codes que es un formato intermedio que sirve para transportar el código eficientemente o de diferentes plataformas (Hardware - Software).

Portable.- Por ser indiferente a la arquitectura sobre la cual está trabajando, esto hace que su portabilidad sea muy eficiente, sus programas son iguales en cualquiera de las plataformas, ya que java especifica tamaños básicos, esto se conoce como la máquina virtual de java.

Interpretado y compilado a la vez.- Java puede ser compilado e interpretado en tiempo real, ya que cuando se construye el código fuente este se transforma en una especie de código de máquina.

Multihebra o Multihilos.- Java tiene una facilidad de cumplir varias funciones al mismo tiempo, gracias a su función de multihilos ya que por cada hilo que 31 el programa tenga se ejecutaran en tiempo real muchas funciones al mismo tiempo.

Dinámico.- El lenguaje java es muy dinámico en la fase de enlazado, sus clases solamente actuaran en medida en que sean requeridas o necesitadas con esto permitirá que los enlaces se puedan incluir incluso desde fuentes muy variadas o desde la red.⁷

Produce Applets.- En java se pueden crear aplicaciones independientes y applets, las independientes se comportan como cualquier programa escrito.

Por otra parte los applets son considerados pequeños programas, tienen la capacidad de ejecutar funciones muy complejas.

Alto rendimiento.- Java es considerado de alto rendimiento por ser tan veloz en el momento de correr los programas y por ahorrarse muchas líneas de código.⁸

2.4.3.1 RAZONES PARA ESCOGER JAVA POR SOBRE OTROS LENGUAJES

Es orientado a objetos: si bien existen detractores de esta modalidad, la programación orientada a objetos resulta muy conveniente para la mayoría de las aplicaciones, y es esencial para los videojuegos. Entre las ventajas más evidentes que ofrece se encuentra un gran control sobre el código y una mejor organización, dado que basta con escribir una vez los métodos y las propiedades de un objeto, independientemente de la cantidad de veces que se utilicen.

- Es muy flexible: Java es un lenguaje especialmente preparado para la reutilización del código; permite a sus usuarios tomar un programa que hayan desarrollado tiempo atrás y actualizarlo con mucha facilidad, sea que necesiten agregar funciones o adaptarlo a un nuevo entorno.
- Funciona en cualquier plataforma: a diferencia de los programas que requieren de versiones específicas para cada sistema operativo (tales como Windows o Mac), las aplicaciones desarrolladas en Java funcionan en cualquier entorno, dado que no es el sistema quien las ejecuta, sino la máquina virtual (conocida como Java Virtual Machine o JVM).

⁷ (Holzner, 2012)

⁸ Ceballos, 2013

- Su uso no acarrea inversiones económicas: programar en Java es absolutamente gratis; no es necesario adquirir ninguna licencia, sino simplemente descargar el kit de desarrollo (Java Development Kit o JDK) y dar riendas sueltas a la imaginación.
- Es de fuente abierta: Java ofrece el código de casi todas sus librerías nativas para que los desarrolladores puedan conocerlas y estudiarlas en profundidad, o bien ampliar su funcionalidad, beneficiándose a ellos mismos y a los demás.
- Es un lenguaje expandible: continuando con el punto anterior, cada programador tiene la libertad de revisar y mejorar el código nativo de Java, y su trabajo puede convertirse en la solución a los problemas de muchas personas en todo el mundo. Infinidad de desarrolladores han aprovechado esta virtud del lenguaje y continúan haciéndolo.⁹

2.4.2 MYSQL

2.4.2.1 ENTORNO DE MYSQL

MySQL es un sistema de gestor de Base de Datos Relacionales, licenciado bajo la GPL (General Public License) de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada para la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, la posibilidad de integrar este gestor a un software propietario.

Este gestor de Base de Datos Relacionales, es, probablemente el más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debido, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.¹⁰

⁹ Julián Pérez Porto y Ana Gardley. Publicado: 2010. Actualizado: 2013.

¹⁰ ARAMAYO, Cristian; MySQL; http://www.salnet.com.ar/inv_mysql/mysql.htm

2.4.2.2 CARACTERÍSTICAS GENERALES

Las principales características de este gestor de Base de Datos Relacionales son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.)
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- En el caso particular de la programación orientada a la web, es famosa la dupla PHP + MySQL para desarrollar sitios web dinámicos.
- MySQL puede ser utilizado para múltiples aplicaciones informáticas como: ejecutar algunas consultas, y ver los resultados.
- MySQL puede ser usado también en modo batch: es decir, se pueden colocar toda una serie de consultas en un archivo, y posteriormente decirle a MySQL que ejecute dichas consultas.^{11 12}

2.4.2.3 Características de MySQL versión 4.0 en adelante

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan.

Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.

¹¹ ANONIMO; Qué es MySQL?; http://tutorial.unlugar.com/tut_mysql/pop_adm/1.htm

¹² ARAMAYO, Cristian; MySQL; http://www.salnet.com.ar/inv_mysql/mysql.htm

- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.
- Tiene soporte SSL (protocolo que maneja un algoritmo de encriptación).¹³

Sin embargo a partir de la versión 5 ya se ha implementado el uso de triggers.

2.4.2.4 SISTEMAS OPERATIVOS DE SOPORTE PARA MYSQL

El Gestor de Base de Datos Relacional MySql es muy conocido en la plataforma Linux, aunque trabaja en otras plataformas como:

- Windows,
- Sco,
- Sun, - IBM Aix, - HP-Ux.

Cabe decir que mientras se trabaje aplicaciones sencillas que no sean muy críticas, será muy eficaz.¹⁴

2.5 BIBLIOGRAFÍA:

- Fundamentos de programación en Java, Jorge Martínez editorial EME
- El lenguaje de programación Java es independiente de Sun Microsystem
- Database Concepts System Avi Silberschatz, Henry F. Korth, S. Surdasha. McGraw-Hill, 2010

WEBGRAFIA

<http://ri.ufg.edu.sv/jspui/bitstream/11592/6742/3/615.1-A323p-Capitulo%20II.pdf>

¹³ Enciclopedia, Características de MySQL; <http://es.wikipedia.org/wiki/MySQL>

¹⁴ ANONIMO; Qué es Mysq!?: http://tutorial.unlugar.com/tut_mysql/pop_adm/1.htm

<http://bibdigital.epn.edu.ec/bitstream/15000/451/1/CD-0419.pdf>

<http://definicion.de/java/>

3. EJECUCIÒN DEL PROYECTO DE INVESTIGACIÒN

3.1 PLAN GENERAL DEL TRABAJO

3.1.1 ANALISIS PRELIMINAR DEL PROBLEMA:

Cientos de pacientes pasan cada año por la clínica, consultan al médico de cabecera. Y casi todos dejan un rastro de papeles y pruebas analíticas. El enorme crecimiento de la actividad asistencial plantea un problema de almacenar esa ingente cantidad de papeles y datos en historias clínicas encontrándose desbordados.

Es importante tomar en cuenta que a veces existe una inconformidad de los pacientes y sus familiares por la lentitud de los procesos que realiza la clínica la búsqueda de la información necesaria para cada paciente por la cantidad de historias clínicas guardadas hace que colapse la atención oportuna

Los profesionales que trabajan en la clínica han manifestado su inconformidad ya que la cantidad de papeleo es muy grande haciendo que desarrollen su trabajo de forma muy limitada.

3.1.1.1 DEFINICIÒN DEL PROBLEMA

Como un sistema de registro de historias clínicas podría mejorar el tratamiento y organización de la información

La Clínica “La Dolorosa” requiere el desarrollo de un Sistema de registro a la medida que permitirá agilizar los procesos que son llevados a cabo en cada una de las áreas donde se identificaron los problemas, de tal forma que se elimina o minimiza la utilización de herramientas manuales como papel y esfero para el desarrollo de informes que son realizados por los empleados de la institución y en este sentido, hacer uso de solo un software que permita que dichos informes deseados por la institución puedan ser generados automáticamente, reduciendo así el tiempo y uso de herramientas no óptimas que actualmente son empleadas en la elaboración de registros en la institución; tomando en cuenta, que el sistema informático realizará.

Además la información estará segura y aislada de pérdida, daño, hurto y otras acciones que perjudiquen la seguridad de ésta, el sistema administrará a través de una base de datos, el registro, modificación y actualización de información. En este sentido, podrá aprovechar eficientemente el recurso informático con el que cuente, al redistribuirlo eficientemente y automatizar los procesos a través de la implementación del Sistema Informático.

3.1.1.2 ESTUDIO DE FACTIBILIDAD

TÉCNICA:

El desarrollo del sistema informático utilizando java y el gestor de datos Msq, es factible ya que cuenta con los recursos de Software y Hardware necesarios para su creación. Se dispone con el Hardware para la implementación del sistema y el Software, será suministrado con las siguientes características:

HARDWARE	
Computadora portátil; HP	
Windows 10	Sistema Operativo
Java	Lenguaje de Programación
MySQL	Motor de base de datos

Tabla 41 Estudio de factibilidad Técnico

OPERATIVA

La presente investigación es de fácil comprensión y manipulación, para su enseñanza se utilizara un máximo de 2 horas para su capacitación.

ACTIVIDAD	TIEMPO
Ingreso al Front End	5 segundos
Ingreso al Back End	5 segundos
Ingreso de la información a la base de datos	5 minutos
Modificaciones en la base de datos	5 minutos
TOTAL	10 minutos, 10 segundos

Tabla 42 Estudio de Factibilidad Operativo por cada búsqueda de paciente en el sistema

ACTIVIDAD	TIEMPO
Impresión y registro de documentacion necesaria	35 minutos
Archivar la documentación	15 minutos
Detalle informativo de la historia	10 minutos
TOTAL	1 horas

Tabla 43 Estudio de Factibilidad Operativo por cada búsqueda de paciente de forma manual

ECONOMICA

Para la elaboración del presente proyecto el estudiante del Instituto Tecnológico Superior “San Gabriel” cuenta con los recursos necesarios para financiar un 100% del costo, además existen los elementos necesarios para su ejecución.

LEGAL

Todos los componentes que participan en el sistema de registro responden a la necesidad de mejorar el servicio de la Clínica La Dolorosa y cumple con todos los requisitos enmarcados en la constitución nacional vigente. Además, se ha utilizado software libre que sigue los lineamientos del reglamento. Por los antecedentes mencionados legalmente es factible la realización de la investigación a implementarse en la casa de salud.

3.1.2 DESARROLLO DEL PROYECTO

REQUERIMIENTOS FUNCIONALES

Expresan la naturaleza del funcionamiento del sistema (cómo interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento).

Ingreso

- El ingreso del paciente al sistema será realizada de forma interactiva. Se le preguntará los datos personales (pueden ser varios).
- Se podrá generar una copia impresa de la historia clínica (sin valor oficial) en el ordenador desde donde se realice el proceso.

Gestión

- La enfermera de la recepción será la encargado de llenar los parámetros
- Los Médicos tendrán acceso a las listas de los pacientes que estén ingresados y la aplicación les debe permitir rellenar las historias clínicas.

REQUERIMIENTOS NO FUNCIONALES

Desempeño

El sistema tiene la habilidad de procesar las operaciones de un usuario individual dentro de unos tiempos de respuesta deseados.

Disponibilidad

El período en el cual el sistema está en operación para ser utilizado por el usuario final, será indefinido ya que será manejado a la hora que lo necesiten

Usabilidad

Es fácil de manejar es practica haciendo más eficiente la atención

Escalabilidad

La escalabilidad del sistema es estable ya que cuenta con el diseño como la infraestructura tecnológica de hardware y de software que lo soporta.

Seguridad

Cuenta con usuario y contraseña controlando el acceso a los servicios e información; así como la capacidad de detectar, aislar la información.

3.1.2.1 PLANIFICACIÓN DEL SISTEMA

Se recopilara toda la información respectiva a este proyecto, así como también se investigara referente al lenguaje de programación java y mysql para la base de datos con la intención de reducir la programación manejando un ambiente de desarrollo completamente grafico que proporcionara la interacción del usuario con el sistema para lo cual se analizara en internet.

Este trabajo se realizara en el instituto tecnológico superior san Gabriel en donde tiene su funcionamiento, utilizando los equipos y los datos que proporcionaran las autoridades u pacientes

Se instalara en los equipos que disponga los laboratorios de cómputo del instituto para de esa manera efectuar las pruebas necesarias con la finalidad de controlar los resultados adecuados y dar prontamente soluciones en caso de existir novedades al momento de la implantación de este sistema.

Luego se capacitara a los usuarios que utilizan en la clínica para la buena gestión y utilización del sistema

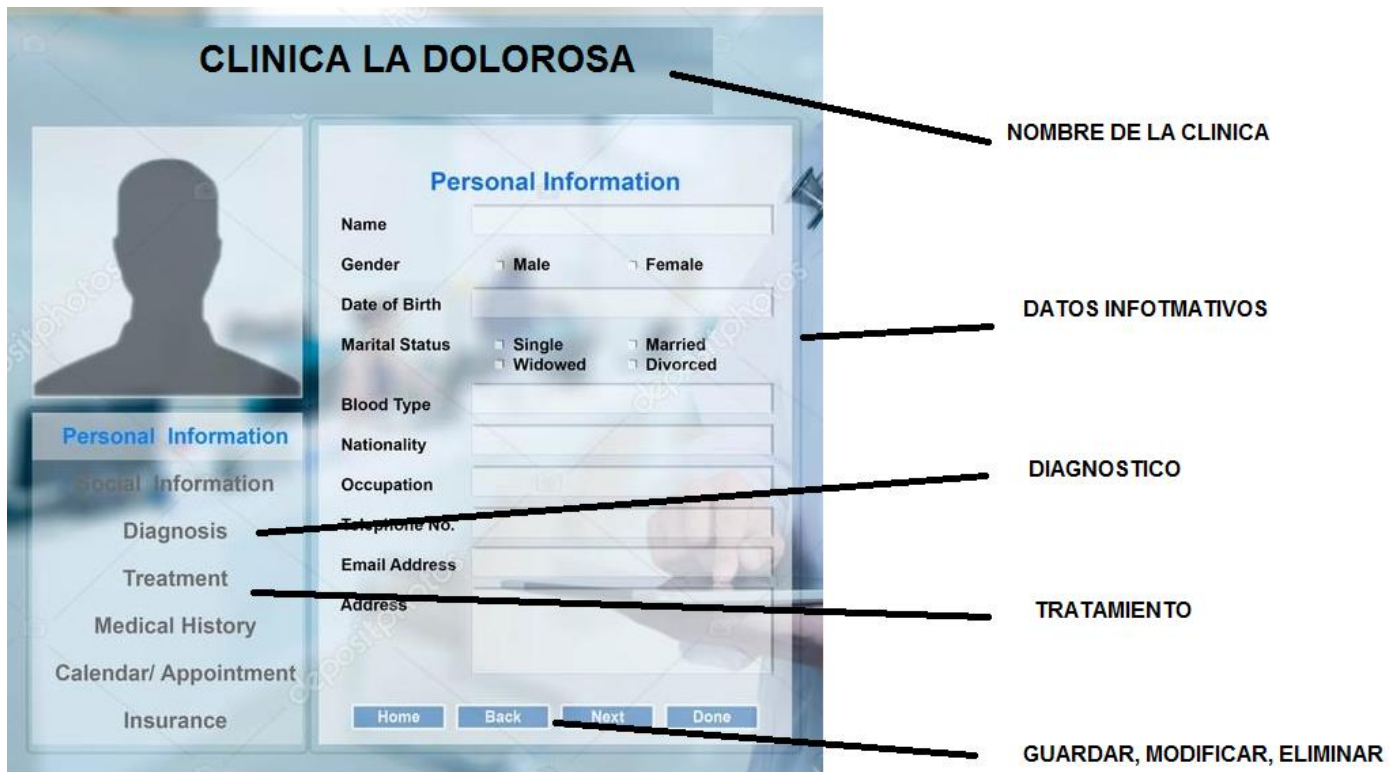


Ilustración 38 Interfaz grafica

3.1.2.2 DISEÑO CONCEPTUAL Y LÓGICO DEL SISTEMA

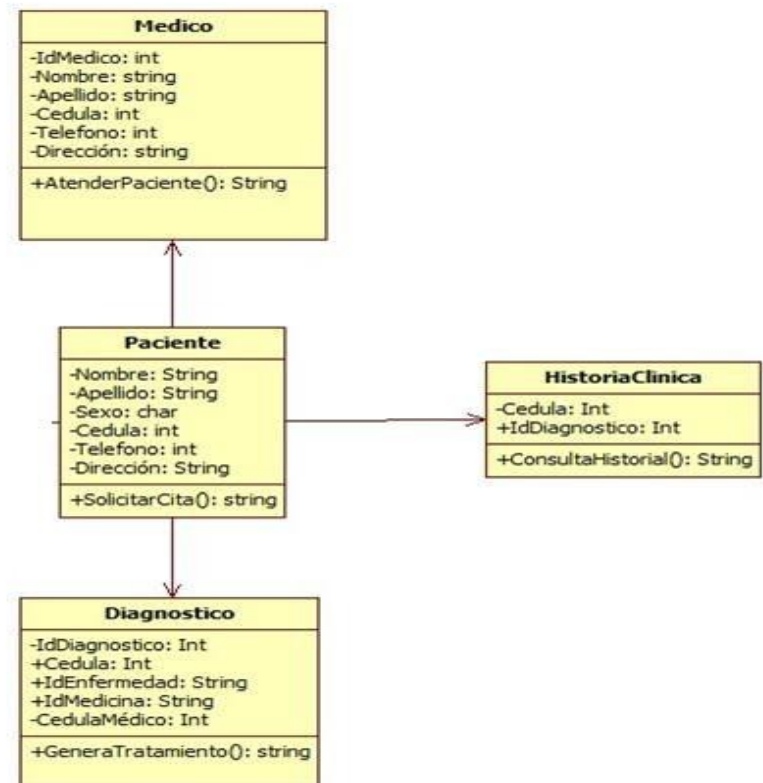


Ilustración 39 Diseño lógico del sistema

3.2 HIPÓTESIS

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA QUE REGISTRARA LAS HISTORIAS CLÍNICAS DE LOS PACIENTES, DESARROLLADA EN JAVA Y GESTOR DE BASE DE DATOS MYSQL.

3.3 VARIABLES

3.3.1 VARIABLE INDEPENDIENTE:

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL REGISTRO DE HISTORIAS CLINICAS.

Variable independiente	Definición	Factores	Indicadores	Técnicas
Diseño e implementación de un Sistema de registro	Tiene como finalidad registrar y guardar las Historias clínicas.	Componentes	Aplicaciones	-Observación Directa
		Módulos	Caja virtual	-Análisis del contenido de programación
			Plantillas	

Tabla 44 Variable Independiente

3.3.2 VARIABLE DEPENDIENTE

DISEÑO E IMPLEMENTACIÓN EN JAVA Y GESTOR DE BASE DE DATOS MYSQL

Variable dependiente	Definición	Factores	Indicadores	Técnicas
Java	Lenguaje de programación	códigos	constantes, clases, métodos, paquetes	-Entrevista
Mysql	Gestor de base de datos	almacenamiento	Componentes Indicadores	-Fichaje de los datos recopilados necesarios para elaborar el esquema necesario.

Tabla 45 Variable Dependiente

3.4 CRONOGRAMA TENTATIVO

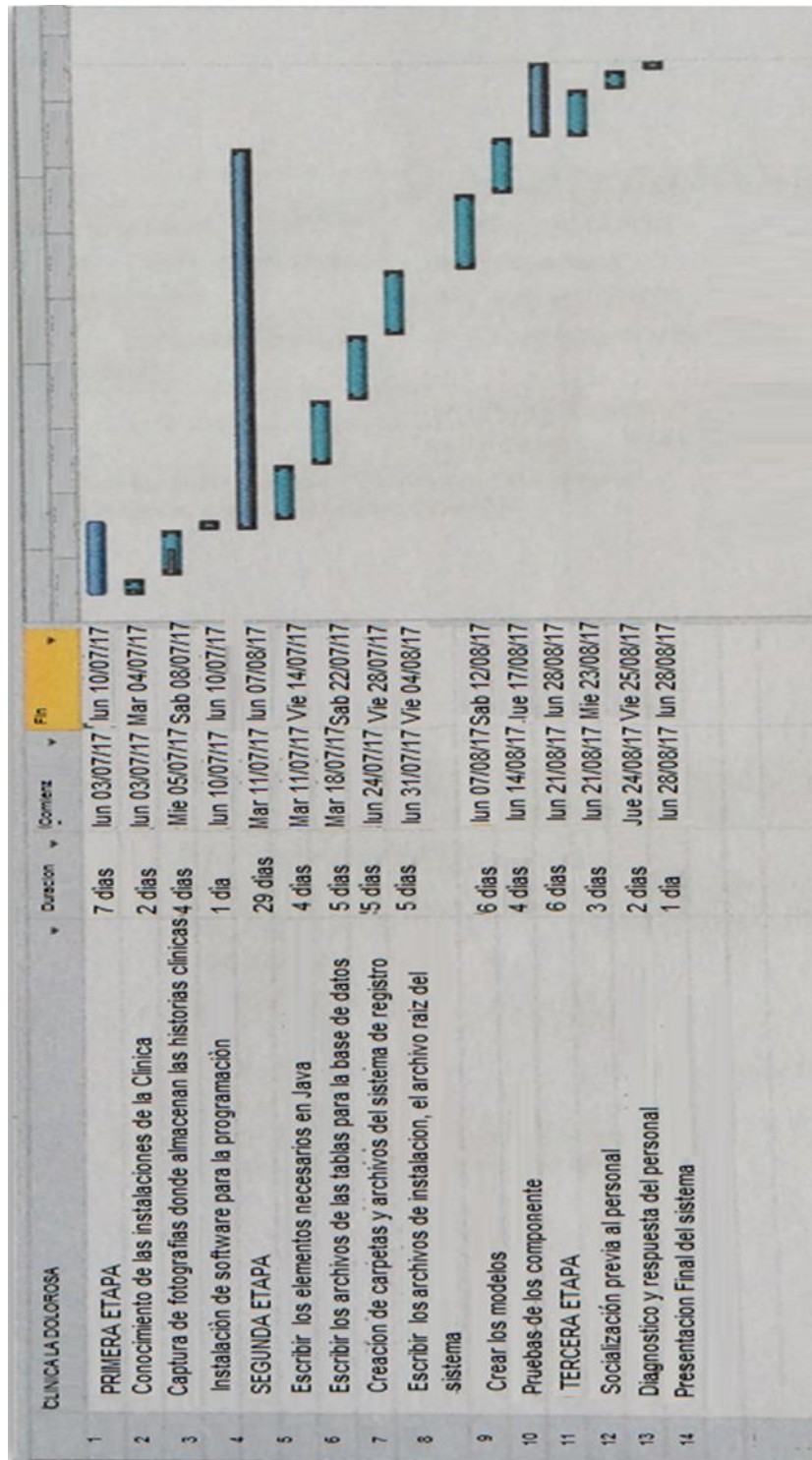


Ilustración 40 Cronograma Tentativo

3.5 RECURSOS

3.5.1 RECURSOS TÈCNICOS

HARDWARE:

DESCRIPCION	
Procesador:	Intel(R) Core (TM) i5
CPU:	260 GHz
Memoria RAM:	4;00 GB
Sistema Operativo:	Windows 10 de 64 bits, procesador x64
Monitor:	14 pulgadas
Mouse:	Genius
Teclado:	

Tabla 46 Recurso Técnico Hardware

SOFTWARE:

DESARROLLO	
Sistema operativo:	WINDOWS 8.1 Pro.
Lenguaje :	JAVA
Gestor de base de datos:	MYSQL
Documentación:	OFFICE 2013 PROFESIONAL

Tabla 47 Recurso Técnico Software

3.5.2 RECURSOS HUMANOS:

FUNCIONES	RESPONSABLE
Investigador	Pablo Renato Galarza Vallejo
Diseñador	
Programador	
Asesor de Tesis	

Tabla 48 Recurso Humano

3.5.3 RECURSOS MATERIALES:

N^a	DESCRIPCION
1	Impresora Epson L355 multifunción
2	PC portátil HP CORE-I5
3	Resmas de papel
4	Suministros de oficina: esferos, lápiz, borrador
5	Anillados y empastados
6	Movilización

Tabla 49 Recurso Material

3.6 MÉTODOS Y TÉCNICAS

3.6.1 MÉTODOS:

Método Analítico: Me permitirá determinar porque se ha venido realizando el ingreso de forma poco eficiente, utilizando técnicas de registro manuales que ocasionan demora en los tiempos de respuesta de resultados.

Método sintético: Me permitirá deducir como el registro que se lo hace manual, se puede realizar de forma técnica y profesional a través de una herramienta tecnológica como el sistema propuesto, lo que mejorara el desempeño de la clínica.

Método inductivo: Me permitirá el análisis de porque la falta de un sistema que permita el registro de los pacientes la falta del mismo causa efectos negativos en la organización de la información de los pacientes, así como la pérdida de tiempo y recursos dentro de la Clínica.

3.6.2 TÉCNICAS:

LA ENTREVISTA

La entrevista es una técnica de recopilación de información mediante una conversación profesional, con la que además de adquirir información acerca de la Clínica “La Dolorosa”, tiene importancia desde el punto de vista educativo.

3.7 PRESUPUESTO:

Nº	DESCRIPCIÓN	COSTOS
1	Impresora Epson lx3300 multifunción	\$ 200,00
3	2 Resmas de Papel	\$ 6,00
4	Lápices, esferos y borrador	\$ 4,00
5	Anillados y empastados	\$ 40,00
6	Movilización-Transporte	\$ 15,00
	TOTAL	\$ 265,00

Tabla 50 Presupuesto

3.8 FUENTE DE FINANCIAMIENTO:

El actual proyecto será capitalizado en su totalidad por el desarrollador Pablo Renato Galarza Vallejo

FECHA DE PRESENTACIÓN:.....

PROPONENTE