

**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS**  
**APLICADAS**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**



**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO**  
**EN SISTEMAS COMPUTACIONALES**

**TEMA:**

**IMPLEMENTACIÓN DEL SISTEMA AUTOMATIZADO DE REFERENCIA Y**  
**CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL**  
**MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.**

**AUTOR:** PAÚL BOLÍVAR VÁSQUEZ MÉNDEZ

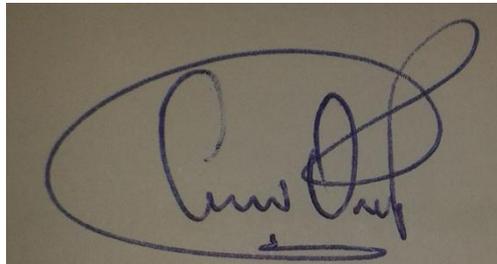
**DIRECTOR:** Ing. Msc. MIGUEL ORQUERA ANDRADE

**IBARRA – ECUADOR**

**2014**

## CERTIFICACIÓN

Certifico que el proyecto “IMPLEMENTACIÓN DEL SISTEMA AUTOMATIZADO DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE”, ha sido realizado en su totalidad por el señor Paúl Bolívar Vásquez Méndez, portador de la cédula de ciudadanía número: 1002545661

A handwritten signature in blue ink on a light-colored background. The signature is stylized and appears to read 'Miguel Orquera Andrade'.

**Ing. Miguel Orquera Andrade**

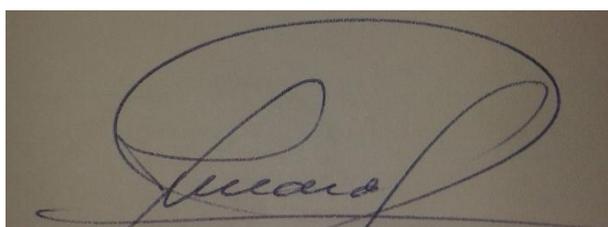
**DIRECTOR DE LA TESIS**

## CERTIFICADO FIN DE DESARROLLO

Que, el Sr. Vásquez Méndez Paúl Bolívar con N° de cédula de ciudadanía 1002545661, estudiante de la Facultad de Ingeniería en Ciencias Aplicadas, Carrera Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte, ha realizado su Trabajo de Grado, **IMPLEMENTACIÓN DEL SISTEMA AUTOMATIZADO DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE** cumpliendo con todos los Requisitos reglamentarios de aprobación de la Institución, con cualidades de responsabilidad y profesionalismo.

Para efecto, se extiende el presente **CERTIFICADO DE CULMINACIÓN DE TRABAJO DE GRADO**, en la ciudad de Ibarra a los 3 días del mes de Febrero del 2014.

Atentamente

A handwritten signature in blue ink on a light-colored background. The signature is cursive and appears to read 'Juan Carlos Armas Cárdenas'.

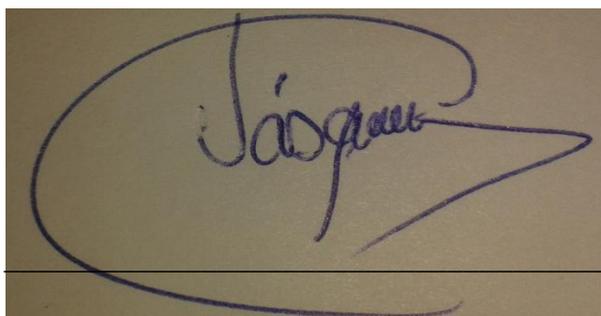
Ing. Juan Carlos Armas Cárdenas

Líder de Gestión TIC's

### **CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN**

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Paúl Bolívar Vásquez Méndez, con cédula de ciudadanía Nro. 100254566-1, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **"IMPLEMENTACIÓN DEL SISTEMA AUTOMATIZADO DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE"** que ha sido desarrollado para optar por el título de Ingeniería en Sistemas Computacionales en la Universidad Técnica del Norte, quedando en la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.



Firma

Nombre: Paúl Bolívar Vásquez Méndez

Cédula: 1002545661

Ibarra a los 3 días del mes de Febrero del 2014

## Universidad Técnica del Norte

### Biblioteca Universitaria

#### Autorización de uso y publicación de la Universidad Técnica del Norte

##### 1. Identificación de la Obra

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de texto completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto,

Para lo cual pongo a disposición la siguiente información:

#### DATOS DEL CONTACTO

<b>CÉDULA DE IDENTIDAD</b>	1002545661
<b>APELLIDOS Y NOMBRES</b>	VÁSQUEZ MÉNDEZ PAÚL BOLÍVAR
<b>DIRECCIÓN</b>	Cdla El Chofer: Panamá 1-201 y Bolivia
<b>EMAIL</b>	<a href="mailto:pvasquez@yachay.gob.ec">pvasquez@yachay.gob.ec</a>

### DATOS DE LA OBRA

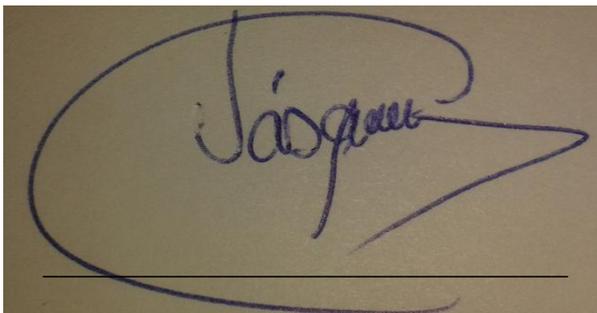
<b>TÍTULO</b>	IMPLEMENTACIÓN DEL SISTEMA AUTOMATIZADO DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE
<b>AUTOR</b>	PAÚL BOLÍVAR VÁSQUEZ MÉNDEZ
<b>FECHA</b>	03 – 02 – 2014
<b>PROGRAMA</b>	PREGRADO
<b>TITULO POR EL QUE OPTA</b>	INGENIERÍA EN SISTEMAS COMPUTACIONALES
<b>DIRECTOR</b>	ING. MIGUEL ORQUERA ANDRADE

### **AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD**

Yo, Paúl Bolívar Vásquez Méndez, con cédula de ciudadanía Nro. 1002545661, en calidad de autor y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la ley de Educación Superior Artículo 144.

### **CONSTANCIAS**

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.



Firma

Nombre: PAÚL BOLÍVAR VÁSQUEZ MÉNDEZ

Cédula: 1002545661

Ibarra a los 3 días del mes de Febrero

### DEDICATORIA

Con mucho cariño dedico este trabajo, fruto de sacrificio, a mi Madre, que con paciencia, dedicación y esfuerzo ha grabado en mí ser, el verdadero sentido de la responsabilidad. Quién con su ejemplo me llevó por el trabajo honrado y me inculcó hacia la superación ya que con su entusiasmo alegría y nobleza supo depositar todo su apoyo y confianza en mí para seguir siempre adelante. A mi tía María Méndez gratitud por su apoyo en los momentos que más necesitamos al extendernos una mano para salir adelante y a mi novia Jacque por ser un pilar fundamental en estos 4 años de constante paciencia y esmero.

Paúl Vásquez Méndez

### AGRADECIMIENTO

A Dios por la vida, la salud y la fortaleza que me brinda para que con responsabilidad y empeño concluya mis sueños y mis metas trazadas.

A la Universidad Técnica del Norte y la Facultad de Ciencias Aplicadas de cuyas aulas llevo los mejores recuerdos de mis épocas de estudiante universitario.

A mi tutor Ing. Miguel Orquera Andrade que con su ejemplo, enseñanza y su constancia ha sembrado en mí, la semilla del saber.

A mi querido Hospital San Vicente de Paúl que me abrió las puertas para ganar la experiencia necesaria en el desarrollo de este proyecto y a la Ciudad del Conocimiento Yachay E.P. que me permitió culminar con este anhelado trabajo de grado.

A las personas que colaboraron con la realización de este trabajo, ya que con su apoyo y empuje, han contribuido para que concluya una etapa de mi preparación importante en mi vida.

Paúl Vásquez Méndez

## RESUMEN

El Sistema de Referencia y Contrareferencia es el mecanismo a través del cual el Ministerio de Salud Pública, en el marco de sus procesos de descentralización de competencias y recursos, define estrategias que permitan garantizar a la población en general el acceso a los servicios de salud, con el concurso de los distintos actores involucrados entre los que se cuentan los Entes Territoriales, y los Prestadores de Servicios de Salud de carácter público.

El manual de referencia y contrareferencia, constituye una propuesta metodológica que plantea conceptos y desarrolla en forma sencilla el manejo del paciente para buscar una atención con calidad y calidez en el primer nivel de salud para luego ser remitido de ser el caso a un nivel superior o viceversa; para esta actividad es necesario una coordinación en los niveles involucrados, buscar una eficiente comunicación, reportar dicho evento por el médico de la Unidad operativa y registrar en el parte diario de atenciones.

Posteriormente, el médico especialista del hospital, luego de haber atendido al paciente, si lo amerita, deberá realizar una contrareferencia del paciente a la unidad de salud de donde vino, el mismo que deberá ir acompañado de su Historia Clínica en los cuales se adjuntará los datos de la enfermedad, sugerencia y tratamiento a seguir, o si es el caso, el mismo médico de especialidad deberá agendar una nueva cita para un determinado día.

## SUMMARY

The reference and counter system is the mechanism through which the Ministry of Health, as part of its decentralization of powers and resources, define strategies to ensure the general public access to health services, with the participation of different stakeholders among which include local authorities, and Health Service Providers of public character.

The reference manual and counter, is a methodology that develops and presents concepts in a simple patient management to find an efficient care in the primary care level and then be sent to be the case at a higher level, or vice versa, for this activity involves coordination at the levels involved, seek efficient communication, report the event by the physician of the operating unit and record the daily part of attentions.

Thereafter, the hospital doctor, after having attended the patient, if warranted, should conduct counter referral patient's health unit where he came from, the same shall be accompanied by all disease data, suggestion and course of treatment, or if applicable, the same medical team must schedule a new appointment for a particular day.

## ÍNDICE

CERTIFICACIÓN.....	i
CERTIFICADO FIN DE DESARROLLO.....	ii
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN.....	iii
AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD.....	vi
DEDICATORIA.....	vii
AGRADECIMIENTO.....	viii
RESUMEN.....	ix
SUMMARY.....	x

## INTRODUCCIÓN

1	INTRODUCCIÓN.....	2
1.1	ANTECEDENTES.....	2
1.1.1	Situación Actual.....	3
1.1.2	Prospectiva.....	3
1.2	OBJETIVOS.....	4
1.2.1	Objetivo General.....	4
1.2.2	Objetivos Específicos.....	4
1.3	JUSTIFICACIÓN.....	4
1.4	BENEFICIOS.....	6

## MARCO TEÓRICO

2	MARCO TEÓRICO.....	8
2.1	SOFTWARE LIBRE.....	8
2.2	HERRAMIENTAS DEVELOPER.....	9
2.3	PHP.....	10

2.4	POSTGRESQL .....	11
2.4.1	Generalidades .....	13
2.4.2	Características .....	14
2.5	SYMFONY .....	17
2.5.1	Historia .....	18
2.5.2	Características .....	19
2.5.3	Características para el desarrollo automatizado de proyectos web.....	20
2.5.4	Entorno de Desarrollo y Herramientas .....	21
2.5.5	La Arquitectura MVC.....	22
2.5.6	Ventajas.....	23
2.6	JQUERY .....	25
2.6.1	Características .....	25
2.6.2	Ventajas.....	26
2.7	DOCTRINE.....	27
2.7.1	Historia .....	28
2.7.2	Influencias .....	28
2.7.3	Demostración de uso .....	28
2.7.4	Características .....	29
2.7.5	Generación Automática de modelo .....	30
2.7.6	Lenguaje DQL .....	30
2.8	METODOLOGÍA .....	31
2.8.1	Xtremme Programming (XP) .....	31

## ANÁLISIS Y DISEÑO

3	ANÁLISIS Y DISEÑO.....	36
3.1	ESTÁNDAR IEEE 830 .....	36
3.1.1	Propósito.....	36
3.1.2	Alcance.....	36
3.1.3	Perspectiva del producto .....	37
3.1.4	Funcionalidad del producto .....	37
3.1.5	Restricciones .....	38
3.1.6	Suposiciones y dependencias .....	38
3.1.7	Evolución previsible del sistema .....	38
3.1.8	Comunes de las interfaces .....	38
3.2	Fase Exploratoria.....	40
3.2.1	Fase Exploratoria.....	40
3.2.1.1	Análisis de Requerimientos.....	40
3.2.1.2	Programación XP .....	42
3.2.1.3	Historias de Usuario .....	43
3.2.1.4	Historia de Usuario 1.....	44
3.2.1.5	Historia de Usuario 2.....	45
3.2.1.6	Historia de Usuario 3.....	46
3.2.1.7	Historia de Usuario 4.....	47
3.2.1.8	Historia de Usuario 5.....	48
3.2.1.9	Historia de Usuario 6.....	49
3.2.1.10	Historia de Usuario 7.....	50
3.2.1.11	Historia de Usuario 8.....	51
3.2.1.12	Historia de Usuario 9.....	52

3.2.1.13	Historia de Usuario 10.....	53
3.2.1.14	Historia de Usuario 11.....	54
3.2.1.15	Historia de Usuario 12.....	55
3.2.1.16	Historia de Usuario 13.....	56
3.2.1.17	Historia de Usuario 14.....	57
3.2.1.18	Historia de usuario 15.....	58
3.2.1.19	Historia de Usuario 16.....	59
3.2.1.20	Historia de Usuario 17.....	60
3.3	Planificación.....	61
3.3.1	Diagramas de casos de uso.....	61
3.3.2	Los Actores.....	61
3.3.3	Diagrama de Clases.....	65
3.3.4	Modelo de Datos Relacional.....	66
3.3.5	Diccionario de Datos.....	67
3.4	IEEE 1362.....	75
3.4.1.1	Alcance.....	75
3.4.1.2	Identificación.....	76
3.4.1.3	Visión general del documento.....	76
3.4.1.4	Visión general del sistema.....	76
3.4.1.5	Personal Involucrado.....	77
3.4.1.6	Descripción del sistema o situación actual.....	79
3.4.1.7	Mantenimiento/Soporte.....	79
3.4.1.8	Requisitos de la Instalación.....	79
3.5	Prototipo de la pantalla principal del sistema.....	80
3.5.1	Interfaces del sistema.....	80
3.6	Desarrollo.....	83

3.6.1	Documento del diseño final del sistema.....	83
3.6.2	Descripción detallada de la lógica de cada usuario .....	84
3.7	Plan de Contingencia .....	99
3.7.1	Activos e Interdependencias.....	99
3.7.2	Plan de Respaldo.....	101
3.7.3	Plan de Emergencia.....	101
3.7.4	Plan Recuperación.....	101

## CONCLUSIONES Y RECOMENDACIONES

4	Conclusiones y Recomendaciones .....	102
4.1	Conclusiones .....	102
4.2	Recomendaciones.....	103
	GLOSARIO DE TÉRMINOS.....	105

## BIBLIOGRAFÍA

5	Bibliografía .....	107
---	--------------------	-----

## LINKOGRAFÍA

6	Linkografía.....	109
---	------------------	-----

## ANEXOS

6.1	ANEXOS .....	111
-----	--------------	-----

## ÍNDICE DE FIGURAS

Figura 1. Flujo de trabajo actual dentro de la Zona de Salud 1 .....	5
Figura 2. Mapa conceptual de Software Libre.....	8
Figura 3. Componentes de PostgreSQL .....	12
Figura 4. La Arquitectura MVC.....	22
Figura 5. Diagrama Caso de uso general .....	62
Figura 6. Diagrama Caso de uso del perfil Administrador .....	63
Figura 7. Diagrama Caso de uso del perfil Invitado.....	63
Figura 8. Diagrama Caso de uso del perfil Médico.....	64
Figura 9. Diagrama Caso de uso del perfil Estadístico.....	64
Figura 10. Diagrama de Clases .....	65
Figura 11. Modelo de Datos Relacional .....	66
Figura 12. Prototipo de ingreso al sistema .....	80
Figura 13. Prototipo del rol administrador del Sistema .....	81
Figura 14. Prototipo del rol administrador del Sistema .....	81
Figura 15. Prototipo del estadístico del sistema .....	82
Figura 16. Prototipo del invitado del sistema.....	83
Figura 17. Ingreso al sistema.....	84
Figura 18.- Administrador del sistema.....	85
Figura 19.- Usuarios del sistema .....	86
Figura 20. Pacientes.....	87
Figura 21. Unidades Operativas .....	88
Figura 22. Patologías ingresadas .....	89
Figura 23.- Seguros .....	90
Figura 24. Servicios .....	91
Figura 25. Reportes generales.....	92
Figura 26. Pacientes.....	93
Figura 27. Referencias.....	94
Figura 28. Turnos .....	95
Figura 29. Pendientes.....	96
Figura 30. Referencias.....	97
Figura 31. Contrareferencia .....	97
Figura 32. Turnos enviados .....	98
Figura 33. Invitado .....	99

## ÍNDICE DE TABLAS

Tabla 1. Historia de usuario 1 .....	44
Tabla 2. Historia de Usuario 2.....	45
Tabla 3. Historia de Usuario 3.....	46
Tabla 4. Historia de Usuario 4.....	47
Tabla 5 Historia de Usuario 5.....	48
Tabla 6 Historia de Usuario 6.....	49
Tabla 7. Historia de Usuario 7.....	50
Tabla 8. Historia de Usuario 8.....	51
Tabla 9. Historia de Usuario 9.....	52
Tabla 10. Historia de Usuario 10.....	53
Tabla 11. Historia de Usuario 11.....	54
Tabla 12. Historia de Usuario 12.....	55
Tabla 13. Historia de Usuario 13.....	56
Tabla 14. Historia de Usuario 14.....	57
Tabla 15. Tabla de Usuario 15.....	58
Tabla 16. Tabla de Usuario 16.....	59
Tabla 17. Tabla de Usuario 17.....	60
Tabla 18. Tablas Bdds .....	68
Tabla 19. Tabla Instituciones .....	69
Tabla 20. Tabla Seguros.....	69
Tabla 21. Tabla Servicios .....	69
Tabla 22. Unidades Operativas.....	69
Tabla 23. Tabla Informaciones.....	70
Tabla 24. Tabla Localidades .....	70
Tabla 25. Tabla Diagnósticos.....	70
Tabla 26. Tabla Médicos.....	71
Tabla 27. Tabla establecimientos .....	71
Tabla 28. Tablas pacientes.....	72
Tabla 29. Tabla Referencias.....	73
Tabla 30. Tabla Contrareferencias.....	75
Tabla 31. Tabla desarrollador .....	77
Tabla 32. Tabla Administrador .....	78
Tabla 33. Tabla Estadístico .....	78



# CAPÍTULO 1

## Introducción



## 1 INTRODUCCIÓN

### 1.1 ANTECEDENTES

El Hospital General San Vicente de Paúl, líder en servicios de salud de la región 1 del norte del país se caracteriza por brindar atención mental, física y social a la comunidad, a través de acciones necesarias y oportunas, con atención de especialidades, tecnología de punta, dentro de un ambiente de calidad y calidez, dentro de las directrices que el Ministerio de Salud Pública implanta a nivel nacional en el país.

El manual de referencia y contrareferencia, constituye una propuesta metodológica que plantea conceptos y desarrolla en forma sencilla el manejo del paciente para buscar una eficiente atención en el primer nivel de atención y luego ser remitido de ser el caso a un nivel superior o viceversa; para esta actividad es necesario una coordinación en los niveles involucrados, buscar una eficiente comunicación, reportar dicho evento por el médico de la Unidad operativa y registrar en el parte diario de atenciones.

De esta forma los pacientes que son atendidos en sus unidades operativas sean estos el caso centros o subcentros de salud si fuese el caso serán remitidos a una unidad de mayor complejidad como es el Hospital San Vicente de Paúl donde existen médicos especialistas los cuales se encuentran listos para atender las necesidades de las unidades operativas de menor nivel.

### 1.1.1 Situación Actual

La congestión en los hospitales del sector público; ya que entre el 70 y 80 por ciento de las patologías que presentan los pacientes, pueden ser atendidas en las unidades de salud de primer nivel, de esta forma el ciudadano tiene que acudir a la unidad operativa a donde pertenece y si el médico considera que amerita ser atendido por un médico especialista, deberá ser transferido al hospital provincial. Para ello deberá llenar la hoja de referencia y llegar al hospital, sitio en el cual, a través de la trabajadora social coordina para la obtención del turno.

Posteriormente, el médico especialista del hospital, luego de haber atendido al paciente, si lo amerita, deberá realizar una contrareferencia del paciente a la unidad de salud de donde vino, el mismo que tendrá ir acompañado de todos los datos de la enfermedad, sugerencia y tratamiento a seguir, o si es el caso, el mismo médico de especialidad deberá agendar una nueva cita para un determinado día.

### 1.1.2 Prospectiva

Con una aplicación web nos permitirá automatizar, analizar, controlar, organizar de mejor manera la atención de primer nivel hacia el segundo nivel y viceversa sistematizando el formulario de referencia y contrareferencia para que este proceso sea lo más eficiente posible, evitando de esta manera las extensas filas para ser atendidos, debido a que es un procedimiento vital que diariamente realiza el Hospital. De esta forma obtendremos un acceso rápido y sencillo hacia el manejo de turnos en estadística especialmente en pacientes que acuden de las diferentes partes de la provincia e inclusive reflejar la demanda insatisfecha que existe en esta casa de salud.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo General

Implementar el sistema automatizado de referencia y contrareferencia del Hospital San Vicente de Paúl, utilizando Software Libre.

### 1.2.2 Objetivos Específicos

1. Realizar el levantamiento de requerimientos de software y de sistema en base del estándar IEEE 830 Y IEEE1362.
2. Realizar el diseño arquitectónico, específico y de datos de la aplicación.
3. Desarrollar el aplicativo web utilizando un lenguaje de programación, una herramienta de desarrollo y aplicando las mejores prácticas al respecto.
4. Realizar las pruebas unitarias, modulares y de sistema de la aplicación desarrollada.
5. Implantar el aplicativo en la infraestructura tecnológica de la institución.

## 1.3 JUSTIFICACIÓN

Mediante la reunión realizada por Gestión Técnica Hospitalaria con los diferentes Líderes de los servicios Gerente, Director Técnico Asistencial, Líder de Estadística con el fin de solucionar los inconvenientes que suceden en el servicio han tomado en cuenta que los cambios al principio generan molestias en las personas que venían de cierto u otro modo siendo atendidas por ello se solicita de manera urgente realizar un sistema el cual permita manejar la atención a estos pacientes ambulatorios.

El desarrollo de la aplicación se lo hará en PHP con el framework Symfony con el patrón MVC.

Con el patrón de diseño MVC nos permite trabajar por separado la lógica de control, la lógica de negocio y la lógica de presentación. Utilizando este tipo de patrón es posible conseguir más calidad, un mantenimiento más fácil y una de las cosas más importantes que permite normalizar y estandarizar el desarrollo de software.



**Figura 1.** Flujo de trabajo actual dentro de la Zona de Salud 1

### 1.4 BENEFICIOS

Los beneficiados directos de este proyecto son los funcionarios del proceso de Aseguramiento de a Calidad de Gestión del Hospital San Vicente de Paúl y los indirectos es la población de la Zonal 1 de Salud que es atendida en los centros y subcentros de salud de Carchi, Esmeraldas, Imbabura y Sucumbíos, la cual no tendrá que viajar únicamente para reserva el turno sino ir con seguridad para el día y la fecha reservado.



# CAPÍTULO 2

## Marco Teórico



Ministerio  
de Salud Pública

## 2 MARCO TEÓRICO

### 2.1 SOFTWARE LIBRE

Software libre explícitamente se refiere a la libertad que el usuario final tiene para ejecutar programas sea cual sea el índole o propósito, a continuación se detalla los 4 libertades fundamentales con los que el usuario trabaja en el software libre.

- ✚ Libertad para utilizar el programa con cualquier propósito de uso común.
- ✚ Libertad para distribución de copias sin restricción alguna
- ✚ Libertad para estudio y modificación. (Requisito: código fuente disponible)
- ✚ Libertad para publicar el programa mejorado. (Requisito: código fuente disponible)

Cuando se habla de software libre no se debe expresar como NO COMERCIAL, todos los diferentes programas estarán disponibles para uso comercialmente, desarrollo comercialmente, distribución comercial y programación comercial.

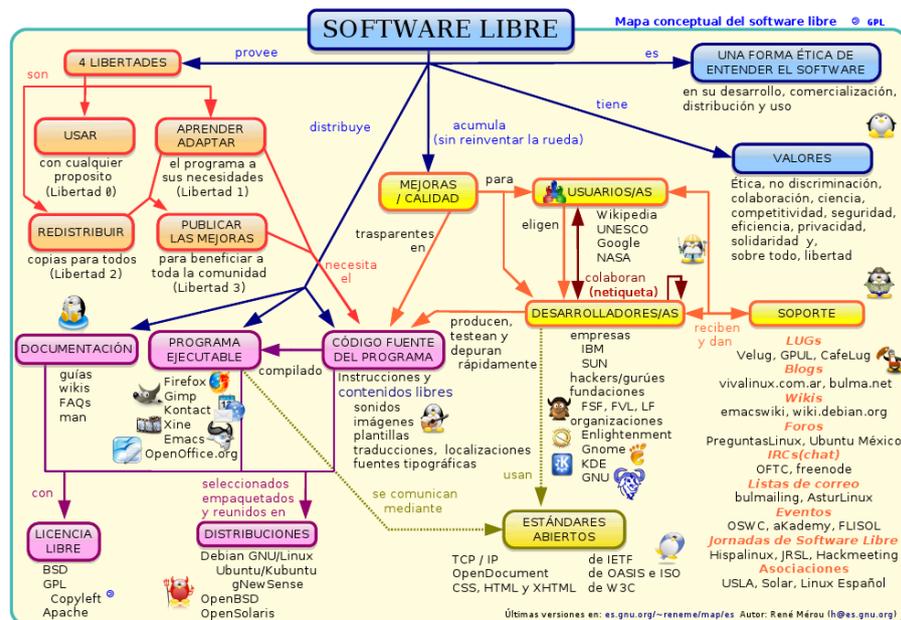


Figura 2. Mapa conceptual de Software Libre

Tomado (Soluciones, 2012)

### 2.2 HERRAMIENTAS DEVELOPER

Para el desarrollo de este proyecto se utilizó PostgreSQL 8.4 de la mano con framework Symfony 1.4 los cuales son de uso libre de acuerdo a las licencias y además siendo partícipes del decreto 1014 (Delgado, 2007) del Gobierno Ecuatoriano que sugiere el uso del software libre en Instituciones Públicas.

### 2.3 PHP



PHP es uno de los lenguajes del lado del servidor más extendidos en la WEB, Permite embeber tus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de forma fácil y eficaz sin tener que generar programas en sus lenguajes destino al HTML. (Maribel, 2012)

- **Tarea de PHP**
  - Funciones de correo electrónico
  - Gestión de base de datos
  - Gestión de archivos
  - Tratamiento de imágenes

El lenguaje de PHP solo ejecuta el código que se encuentra entre sus delimitantes. Los delimitantes más comunes son `<?php` para abrir una sección PHP y `?>` para cerrarla. El objetivo de estos delimitantes es separar el código PHP del resto de código, como por ejemplo el HTML. (Helma, 2009).

PHP conocido como un lenguaje de código abierto que resulta muy útil y sencillo para diseñar de forma rápida y eficaz aplicaciones Web con base de datos. (Christian, 2010)

PHP es un potente lenguaje de secuencia de comandos diseñado específicamente para permitir a los programadores crear aplicaciones en Web con distintas prestaciones de forma rápida. (Vikkam, 2012)

### 2.4 POSTGRESQL



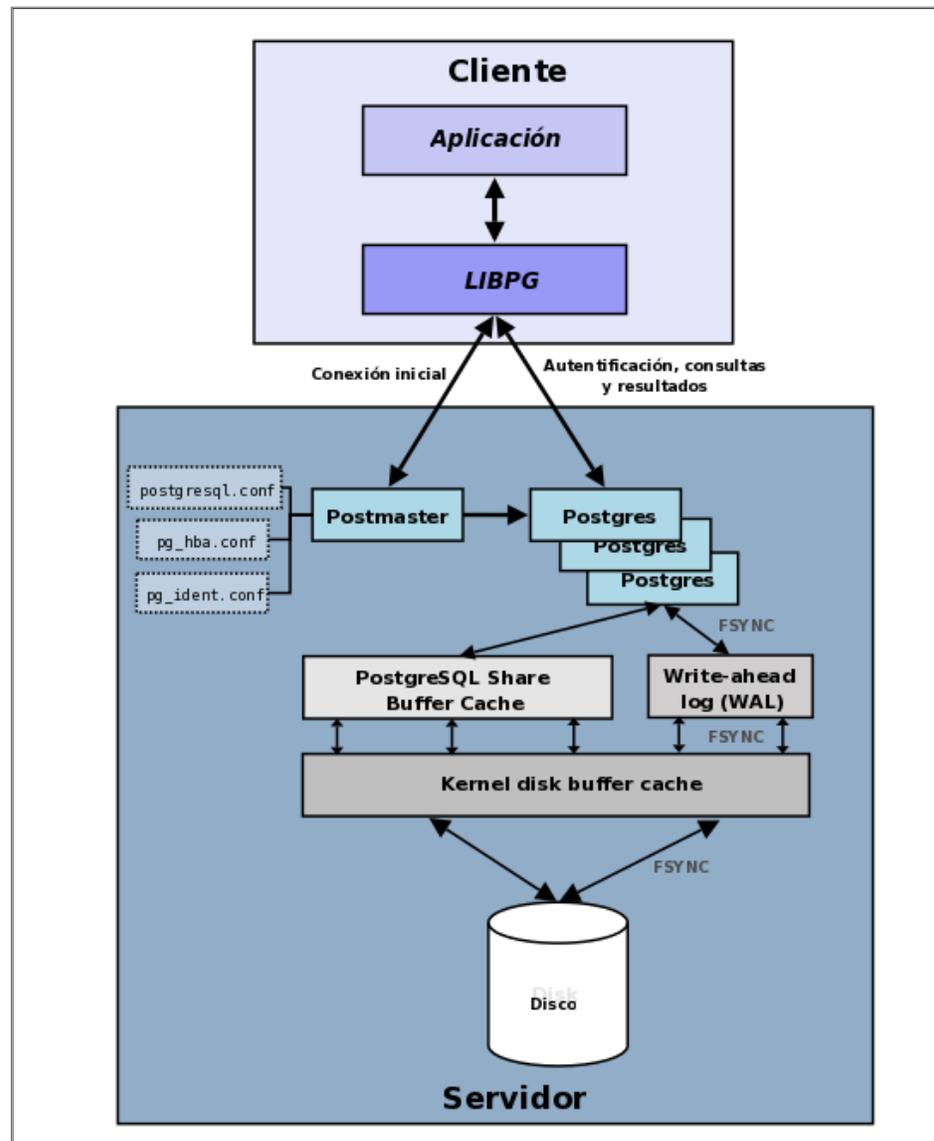
PostgreSQL es un sistema de gestión de base de datos relacional orientado a objetos y libre publicado bajo la licencia BSD con su código fuente disponible de forma libre.

Este sistema de gestión de base de datos en la actualidad el más potente del mercado tomando en cuenta que las últimas versiones no tiene nada q envidiar a las versiones de bases de datos comerciales. (Wikipedia, Wikipedia, 2013)

PostgreSQL una un modelo cliente/ servidor y usa multiprocesos en lugar de multihilos para de esta forma poder garantizar la estabilidad del sistema ya que un fallo en los procesos no afectará en el resto del sistema, de esta forma podrá seguir trabajando. (Spona, 2010)

PostgreSQL tiene características de la orientación a objetos, como puede ser tipos de datos, funciones, disparadores, herencias, restricciones, reglas e integridad transaccional, reuniendo todas estas característica podemos citar que PostgreSQL no es un base de datos netamente orientada a objetos. (Wikipedia, Wikipedia, 2013)

A continuación se muestra un gráfico que define de forma general los componentes más importantes en un sistema PostgreSQL.



**Figura 3.** Componentes de PostgreSQL

Tomado (RafaelMa, 2010)

### 2.4.1 Generalidades

- Aplicación cliente, esta es la aplicación cliente que utiliza PostgreSQL como administrador de base de datos. La conexión puede ocurrir vía TCP/IP o sockets locales.
- Demonio postmaster: Este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes (LibrosWeb, 2010)
- Ficheros de configuración: Los 3 ficheros principales de configuración utilizados por PostgreSQL, postgresql.conf, pg\_hba.conf y pg\_ident.conf
- Procesos hijos postgres: Procesos hijos que se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes. (RafaelMa, 2010)
- PostgreSQL share buffer cache: Memoria compartida usada por PostgreSQL para almacenar datos en caché. (LibrosWeb, 2010)
- Write-Ahead Log (WAL): Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO). (RafaelMa, 2010)
- Kernel disk buffer cache: Caché de disco del sistema operativo

- Disco: Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione. (LibrosWeb, 2010)

### 2.4.2 Características

Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo arranca hace más de 16 años, y durante este tiempo llega a consolidarse en:

- Estabilidad
- Potencia
- Robustez
- Facilidad de administración
- Implementación de estándares

Entre estas y muchas más han sido las principales características que han sobresalido durante su desarrollo. (Maribel, 2012)

Postgresql funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

A continuación vamos a nombrar algunas de las características más importantes y soportadas por PostgreSQL:

#### 2.4.2.1 Generales

- Es una base de datos 100% ACID
- Integridad referencial
- Tablespaces
- Nested transactions (savepoints)

- Replicación asincrónica/sincrónica / Streaming replication - Hot Standby
- Two-phase commit
- PITR - point in time recovery
- Copias de seguridad en caliente (Online/hot backups)
- Unicode
- Juegos de caracteres internacionales
- Regionalización por columna
- Multi-Version Concurrency Control (MVCC)
- Múltiples métodos de autenticación
- Acceso encriptado via SSL
- Actualización in-situ integrada (pg\_upgrade)
- SE-postgres
- Completa documentación
- Licencia BSD
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit. (Maribel, 2012)

### 2.4.2.2 Programación y desarrollo

- Funciones/procedimientos almacenados (stored procedures) en numerosos lenguajes de programación, entre otros PL/pgSQL (similar a PL/SQL en oracle), PL/Perl, PL/Python y PL/Tcl. (RafaelMa, 2010)
- Bloques anónimos de código en procedimientos (sentencias DO)
- Numerosos tipos de datos con la ventaja de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, tenemos disponibles, entre otros, tipos geométricos, de direcciones

- de red, de cadenas binarias, UUID, XML, matrices, etc. (RafaelMa, 2010)
- Soporta almacenamiento de objetos binarios grandes (gráficos, videos, sonido)
- APIs para programar en C/C++, Java, .Net, perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt entre otros. (José, 2012)

### 2.4.2.3 SQL

- SQL92,SQL99,SQL2003,SQL2008
- Llaves primarias (primary keys) y foráneas (foreign keys)
- Check, Unique y Not null constraints
- Restricciones de unicidad postergables (deferrable constraints)
- Columnas auto-incrementales
- Indices compuestos, únicos, parciales y funcionales en cualquiera de los metodos de almacenamiento disponibles, B-tree, R-tree, hash ó GiST
- Sub-selects
- Consultas recursivas
- Funciones 'Windows'
- Joins
- Vistas (views)
- Disparadores (triggers) comunes, por columna, condicionales.
- Reglas (Rules)
- Herencia de tablas (Inheritance)
- Eventos LISTEN/NOTIFY. (RafaelMa, 2010)

### 2.4.2.4 Límites

Algunos de los límites que presenta PostgreSQL se detallan a continuación:

LÍMITE	VALOR
Máximo tamaño base de datos	Ilimitado (Depende de su sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de tamaños por tabla	250 – 1600 (dependiendo del tipo)
Máximo número de índices por tabla	Ilimitado

Tabla 1. Límites de PostgreSQL

Tomado (LibrosWeb, 2010)

## 2.5 SYMFONY



Symfony es un framework diseñado para optimizar el desarrollo de las aplicaciones web gracias a sus diferentes características separando la lógica del negocio, la lógica del servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases que van dirigidas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. (Web, 2007)

Symfony se encuentra desarrollado completamente con PHP 5. Se ha realizado infinidad de pruebas con proyectos reales y se a utilizado en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayor parte de gestores de base de datos, como MySQL, PostgreSQL, Oracle y SQL server de Microsoft. Es multiplataforma ya que se puede ejecutar en diferentes tipos de sistemas operativos como Unix, Linux, Windows. (Wikipedia, Symfony, 2014)

### **2.5.1 Historia**

En el año 2003, Fabien Potencier, creador de Symfony y actual CEO de Sensio Labs, investigó acerca de las herramientas open source existentes para el desarrollo de aplicaciones web en PHP, pero ninguna de las existentes cumplió con sus expectativas. Cuando PHP 5 fue liberado, consideró que las herramientas que existían en ese momento habían madurado lo suficiente para ser integradas en un solo framework. Le llevó un año desarrollar el núcleo de Symfony. Basó su trabajo en el Modelo, Vista, Controlador, el ORM de Propel y el ayudante para realizar plantillas de Ruby on Rails. (Wikipedia, Symfony, 2014)

La primera versión de Symfony fue lanzada en octubre de 2005, por Fabien Potencier. Originalmente fue creado para el desarrollo de las aplicaciones de

Sensio. Luego, tras el éxito que tuvo en el desarrollo de una página web para el comercio electrónico y algunos otros proyectos, decidió liberarlo bajo una licencia open source. (Wikipedia, Symfony, 2014)

### 2.5.2 Características

Symfony fue diseñado para ajustarse a ciertos requisitos q a continuación los detallamos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y \*nix estándares).
- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.
- Sencillo de usar en la mayoría de casos, aunque es preferible para el desarrollo de grandes aplicaciones Web que para pequeños proyectos.
- Aunque utiliza MVC (Modelo Vista Controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.

- Una potente línea de comandos que facilitan generación de código, lo cual contribuye a ahorrar tiempo de trabajo. (Wikipedia, Symfony, 2014)

### **2.5.3 Características para el desarrollo automatizado de proyectos web**

Las características más comunes para el desarrollo de proyectos web están automatizadas en Symfony, tales como:

- La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.
- La capa de presentación utiliza plantillas que pueden ser construidos por desarrolladores de HTML que no tengan conocimientos del framework. Los helpers incluidos permiten minimizar el código utilizado en la presentación, ya que permiten la encapsulación de bloques extensos de código en llamadas simples a funciones.
- Los formularios tienen validación automatizada y lleno automático de datos lo cual asegura la calidad en el llenado de la base de datos y permite mejor comunicación con el usuario.
- El manejo de cache reduce el uso de la banda ancha y la carga hacia el servidor.
- La facilidad de soportar autenticación y credenciales facilita la creación de áreas restringidas y manejo de seguridad de los usuarios.
- El manejo del enrutamiento y las URL limpias permiten considerar direcciones de páginas amigables para los diferentes buscadores.
- Las listas son más fáciles ya que permiten la paginación, clasificación y filtraje automáticos.
- Los plugins, las factorías (patrón de diseño “Factory”) y los “mixin” proveen un alto nivel de extensibilidad a Symfony.
- La interacción con AJAX es mucho más sencilla mediante las ayudas que permiten encapsular los efectos de Java Script compatibles con todos los navegadores en una sola línea de código.

- Los datos incluyen mecanismos de escape que permiten una protección contra ataques producidos por datos corruptos.
- El soporte de e-mail incluido y la gestión de API's permiten a las aplicaciones web interactuar más allá de los navegadores.

### 2.5.4 Entorno de Desarrollo y Herramientas

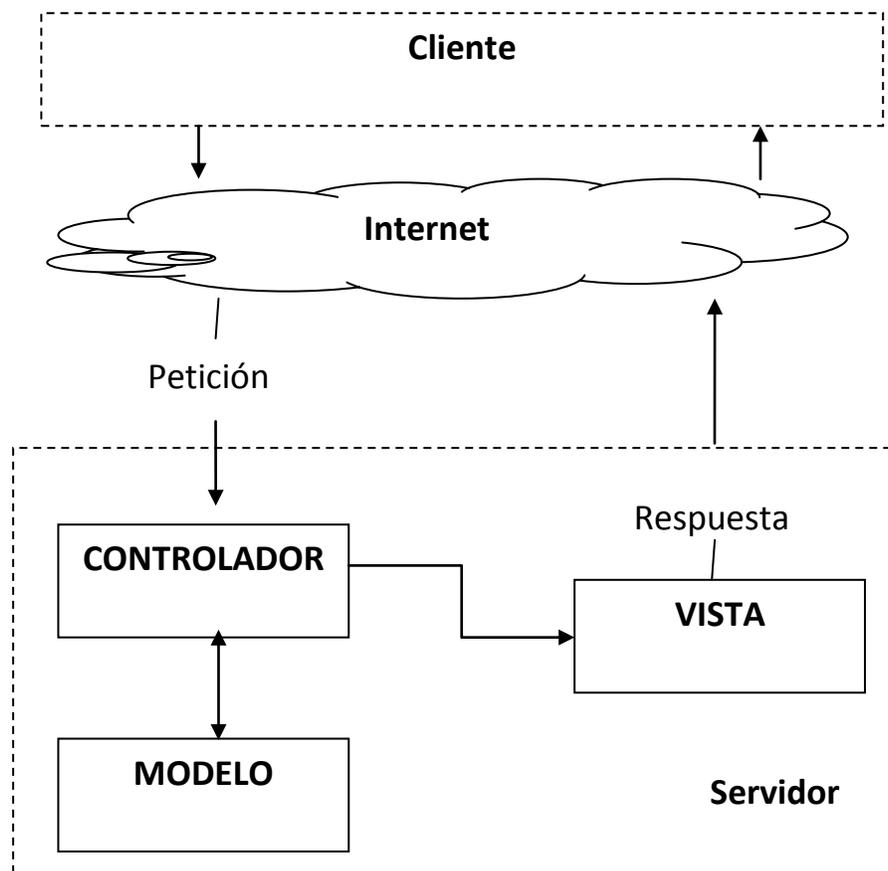
Symfony puede ser completamente personalizado para cumplir con los requisitos de las empresas que disponen de sus propias políticas y reglas para la gestión de proyectos y la programación de aplicaciones. Por defecto incorpora varios entornos de desarrollo diferentes e incluye varias herramientas que permiten automatizar las tareas más comunes de la ingeniería del software:

- Las herramientas que generan automáticamente código han sido diseñadas para hacer prototipos de aplicaciones y para crear fácilmente la parte de gestión de las aplicaciones.
- El framework de desarrollo de pruebas unitarias y funcionales proporciona las herramientas ideales para el desarrollo basado en pruebas (*"test-driven development"*).
- La barra de depuración web simplifica la depuración de las aplicaciones, ya que muestra toda la información que los programadores necesitan sobre la página en la que están trabajando.
- La interfaz de línea de comandos automatiza la instalación de las aplicaciones entre servidores.
- Es posible realizar cambios *"en caliente"* de la configuración (sin necesidad de reiniciar el servidor).

El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación.

### 2.5.5 La Arquitectura MVC

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC y está formado por tres niveles.



**Figura 4.** La Arquitectura MVC

**Tomado** (Wikipedia, MVC, 2014)

- El MODELO en si presenta la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La VISTA, el modelado de una página web la cual permite la comunicación con el usuario.

- El CONTROLADOR procesa las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (Doyle, 2010)

### 2.5.6 Ventajas

Entre las múltiples ventajas de Symfony podemos citar las siguientes:

- Todo el código está orientado a objetos y completamente en PHP5.
- Implementación del patrón de diseño Modelo-Vista-Controlador (MVC) para una estructura clara y flexible.
- Abstracción de base de datos vía Mapeo Relacional de Objetos (ORM) las tablas de la base de datos están disponibles como objetos en el código. La capa ORM está basada en Propel o Doctrine.
- Generación automática y configurable de “selecciones de administración”.
- Integración de las librerías javascript más populares (jQuery, Prototype, YUI, entre otras), las cuales incluyen de serie funciones AJAX listas para usar en nuestras aplicaciones.

- Avanzado sistema de cache que puede integrarse con otros sistemas de cache existentes como caché de archivos, APC, memcache y otros.
- Un parseador YML (YAML) propio, de forma que los ficheros de configuración y la descripción del modelo de datos pueden ser descritos de forma sencilla y rápida (a diferencia de los ficheros XML, con un sinfín de tags de apertura y cierre).
- Documentación de gran cantidad, así como una amplia (y activa) comunidad de desarrolladores.
- Symfony genera código orientado a objetos para las funcionalidades más comunes del manejo de base de datos.
- Genera interfaces CRUD (Crear Leer Actualizar Eliminar) para las tablas de base de datos.
- Permite trabajar en distintos ambientes de desarrollo (en el que se activa una barra de herramientas para depuración), test, pero también es posible crear uno propio.
- Contiene 8.500 test utilitarios y funcionales totalmente automáticos, dando como resultado uno de los frameworks más estables y robustos.
- Muy adecuado para metodologías ágiles de desarrollo como XP (Extreme Programming) o Scrum.

### 2.6 JQUERY



jQuery es considerado un framework de JavaScript es decir, un conjunto de funciones que ya fueron desarrolladas y probadas, están listas para utilizarlas de una manera muy simplificada. En pocas palabras podemos lograr los mismos resultados, en menos tiempo sin necesidad de programar una funcionalidad completamente.

Interactúa con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones (FLV) y agregar interacción técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el Bar Camp NYC. jQuery es la biblioteca de JavaScript más utilizada. (Julio, 2011)

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. (Wikipedia, jQuery, 2013)

#### 2.6.1 Características

Entre las características principales tenemos las siguientes:

- Selección de elementos DOM.

- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.5. (Wikipedia, jQuery, 2013)

### 2.6.2 Ventajas

- Ahorra líneas de código
- Transparencia en el soporte aplicaciones para los navegadores principales.
- Provee un mecanismo para la captura de eventos.
- Provee un conjunto de funciones para animar el contenido de la página de forma muy sencilla.
- Integra funcionalidades para trabajar con Ajax. (Capacity, 2013)

### 2.7 DOCTRINE



Doctrine es una librería para PHP que nos permite trabajar con un esquema de base de datos como si fuese un conjunto de objetos, y no de tablas y registros.

Es un mapeador de objetos-relacional (ORM) escrito en PHP que proporciona una capa de persistencia para objetos PHP. Es una capa de abstracción que se sitúa justo encima de un SGBD.

Doctrine está inspirado en Hibernate, que es uno de los ORM más populares y grandes que existen y nos brindan una capa de abstracción de la base de datos muy compleja. La característica más importante es que le da la posibilidad de escribir consultas de base de datos en un lenguaje propio llamado Doctrine Query Language.

es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional, utilizando un motor de persistencia sistema de gestión de base de datos

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma java Object – relational Mapping

### 2.7.1 Historia

El proyecto Doctrine empezó con Konsta Vesterinen, también conocido como zYne, el 13 de abril de 2006 se hizo el primer envío al repositorio svn.

### 2.7.2 Influencias

Doctrine tiene influencias de docenas de proyectos de personas muy diferentes. Las mayores influencias son de Hibernate (el ORM de Java) y de ActiveRecord (de **Ruby on Rails**). Ambos tienen una implementación completa tanto en Java como en Ruby. El propósito de Doctrine es construir una solución igual de potente para PHP.

### 2.7.3 Demostración de uso

Doctrine 1.x se basa en el active record pattern con datos, en los que una clase se corresponde con una tabla de base de datos.

Si un programador quiere crear un nuevo objeto "Usuario" en la base de datos, no tendrá que escribir ninguna sentencia SQL, simplemente lo siguiente:

```
$user = new User();
$user->name = 'Juan';
$user->password = '123';
$user->save();
echo "El usuario con id $user->id se ha guardado.";
```

### 2.7.4 Características

Una característica de Doctrine es el bajo nivel de configuración que necesita para empezar un proyecto. Doctrine puede generar clases a partir de una base de datos existente y después el programador puede especificar relaciones y añadir funcionalidad extra a las clases autogeneradas. No es necesario generar o mantener complejos esquemas XML de base de datos como en otros frameworks.

Otra característica importante de Doctrine es la posibilidad de escribir consultas de base de datos utilizando un dialecto de SQL denominado **DQL** (Doctrine Query Language) que está inspirado en Hibernate (Java).

Otras características notables de Doctrine son:

- Soporte para datos jerárquicos;
- Soporte para hooks (métodos que pueden validar o modificar las escrituras y lecturas de la base de datos) y eventos para manejar la lógica de negocio relacionada;
- Herencia;
- Un framework de caché que utiliza diversos motores como memcached, SQLite o APC;
- Transacciones ACID;
- Diversos comportamientos del modelo (conjuntos anidados, internacionalización, log, índice de búsqueda);
- Una función "compilar" que combina varios archivos PHP del framework en uno solo para evitar el descenso de rendimiento que provoca incluir varios archivos PHP.

### 2.7.5 Generación Automática de modelo

Cuando se trabaja con ORM, necesitas crear el conjunto de clases que representa el modelo de la aplicación, luego estas clases serán vinculadas al esquema de la base de datos de forma automática con un motor ORM.

Aunque son cosas diferentes, cuando diseñas un modelo relacional y un modelo de clases, suelen ser muy parecidos. Doctrine se aprovecha de esta similitud y nos permite generar de forma automática el modelo de clases basándose en el modelo relacional de tablas.

Es decir, si tenemos una tabla llamada usuarios, se autogenerará una clase llamada Usuarios cuyas propiedades son las columnas de dicha tabla.

ACID a un conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.

### 2.7.6 Lenguaje DQL

DQL es un lenguaje creado para ayudar al programador a extraer objetos de la base de datos. Entre las ventajas de usar este lenguaje se encuentran:

- Está diseñado para extraer objetos, no filas, que es lo que nos interesa.
- Entiende las relaciones, por lo que no es necesario escribir los joins a mano.
- Portable con diferentes bases de datos.

Es importante considerar el uso de DQL para obtener la información a cargar en lugar de usar la “forma automática” de Doctrine para mejorar el rendimiento. En el ejemplo anterior, cuando se accede a \$comment->User, hemos dicho que se está cargando un nuevo objeto de forma dinámica, pues bien, esto no es óptimo porque realiza consultas SQL de más.

### 2.8 METODOLOGÍA

Las metodologías y estándares que se utilizan en el desarrollo de software proporcionan las guías para poder conocer todo el camino a recorrer desde antes de empezar la implementación, con lo cual nos aseguramos la calidad de nuestro sistema final, así como también el cumplimiento en la entrega del mismo en un tiempo estipulado.

Para la elaboración del presente proyecto de tesis, selecciona la metodología XP ya que simplifica el diseño acelerando el desarrollo y facilitando el mantenimiento, poniendo más énfasis en la adaptabilidad que en la previsibilidad adoptando las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto y aplicarlo de manera dinámica durante el ciclo de vida del software.

#### 2.8.1 Xtremme Programming (XP)

Mejor conocida por su nombre en inglés Extreme Programming (XP), es una de las llamadas Metodologías Ágiles de desarrollo de software más exitosas de los tiempos recientes, nace como nueva disciplina de desarrollo de software hace aproximadamente unos seis años, y ha causado un gran revuelo entre el colectivo de programadores del mundo. Kent Beck, su autor, es un programador que ha trabajado en múltiples empresas y que actualmente lo hace como programador en la conocida empresa automovilística DaimlerChrysler.

Con sus teorías ha conseguido el respaldo de gran parte de la industria del software y el rechazo de otra parte.

La programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica. Los Valores originales de la programación extrema son: simplicidad, comunicación, retroalimentación (feedback) y coraje. Un quinto

valor, respeto, fue añadido en la segunda edición de Extreme Programming Explained. Los cinco valores se detallan a continuación:

- **Simplicidad:** La simplicidad es la base de la programación extrema. Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Un diseño complejo del código junto a sucesivas modificaciones por parte de diferentes desarrolladores hace que la complejidad aumente exponencialmente.

Para mantener la simplicidad es necesaria la refactorización del código, ésta es la manera de mantener el código simple a medida que crece. También se aplica la simplicidad en la documentación, de esta manera el código debe comentarse en su justa medida, intentando eso sí que el código esté auto-documentado.

Para ello se deben elegir adecuadamente los nombres de las variables, métodos y clases. Los nombres largos no decrementan la eficiencia del código ni el tiempo de desarrollo gracias a las herramientas de autocompletado y refactorización que existen actualmente.

Aplicando la simplicidad junto con la autoría colectiva del código y la programación por parejas se asegura que cuanto más grande se haga el proyecto, todo el equipo conocerá más y mejor el sistema completo.

- **Comunicación** La comunicación se realiza de diferentes formas. Para los programadores el código comunica mejor cuanto más simple sea. Si el código es complejo hay que esforzarse para hacerlo inteligible. El código auto-documentado es más fiable que los comentarios ya que éstos últimos pronto quedan desfasados con el código a medida que es modificado. Debe comentarse sólo aquello que no va a variar, por ejemplo el objetivo de una clase o la funcionalidad de un método.

Las pruebas unitarias son otra forma de comunicación ya que describen el diseño de las clases y los métodos al mostrar ejemplos concretos de cómo utilizar su funcionalidad. Los programadores se comunican constantemente gracias a la programación por parejas. La comunicación con el cliente es fluida ya que el cliente forma parte del equipo de desarrollo. El cliente decide qué

características tienen prioridad y siempre debe estar disponible para solucionar dudas.

- **Retroalimentación** feedback Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real. Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en lo que es más importante. Considérense los problemas que derivan de tener ciclos muy largos. Meses de trabajo pueden tirarse por la borda debido a cambios en los criterios del cliente o malentendidos por parte del equipo de desarrollo. El código también es una fuente de retroalimentación gracias a las herramientas de desarrollo. (Villafuerte, 2009)

Por ejemplo, las pruebas unitarias informan sobre el estado de salud del código. Ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debidos a cambios recientes en el código.

- **Coraje o valentía** Los puntos anteriores parecen tener sentido común, entonces, ¿por qué coraje? Para los gerentes la programación en parejas puede ser difícil de aceptar, porque les parece como si la productividad se fuese a reducir a la mitad ya que solo la mitad de los programadores está escribiendo código. (Villafuerte, 2009)

Hay que ser valiente para confiar en que la programación por parejas beneficia la calidad del código sin repercutir negativamente en la productividad. La simplicidad es uno de los principios más difíciles de adoptar. Se requiere coraje para implementar las características que el cliente quiere ahora sin caer en la tentación de optar por un enfoque más flexible que permita futuras modificaciones. No se debe emprender el desarrollo de grandes marcos de trabajo (frameworks) mientras el cliente espera.

En ese tiempo el cliente no recibe noticias sobre los avances del proyecto y el equipo de desarrollo no recibe retroalimentación para saber si va en la dirección

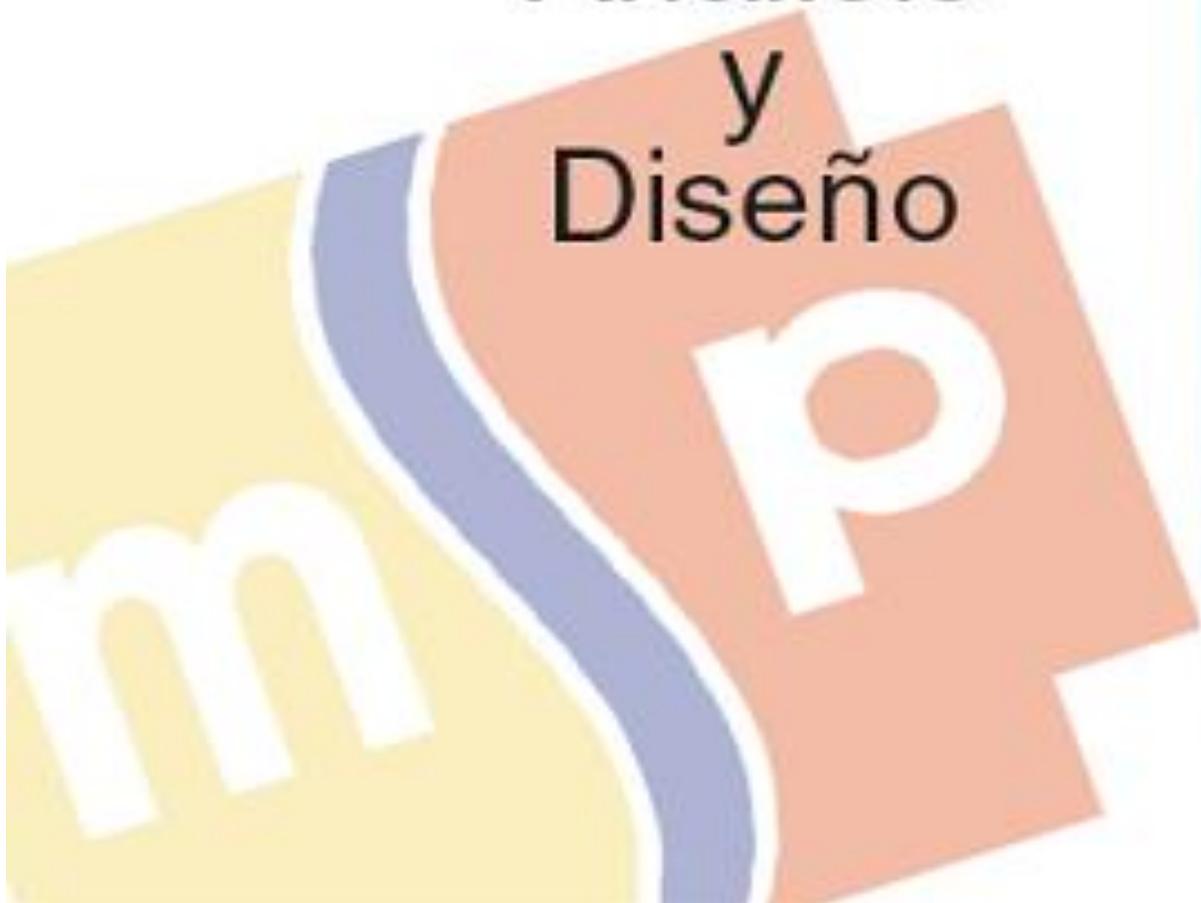
correcta. La forma de construir marcos de trabajo es mediante la refactorización del código en sucesivas aproximaciones. (Villafuerte, 2009)

- **Respeto** El respeto se manifiesta de varias formas. Los miembros del equipo se respetan los unos a otros, porque los programadores no pueden realizar cambios que hacen que las pruebas existentes fallen o que demore el trabajo de sus compañeros. Los miembros respetan su trabajo porque siempre están luchando por la alta calidad en el producto y buscando el diseño óptimo o más eficiente para la solución a través de la refactorización del código. Los miembros del equipo respetan el trabajo del resto no haciendo menos a otros, si no orientándolos a realizarlo mejor, obteniendo como resultado una mejor autoestima en el equipo y elevando el ritmo de producción en el equipo. (EcuRed, 2012)



# CAPÍTULO 3

Análisis  
y  
Diseño



Ministerio  
de Salud Pública

### 3 ANALISIS Y DISEÑO

#### 3.1 ESTÁNDAR IEEE 830

La IEEE (the institute of electrical and electronics engineers), es un instituto internacional dedicado a promover la innovación y la excelencia tecnológica en beneficio de la humanidad. La IEEE dice q para todo trabajo de software es necesario entregar a los clientes la especificación de requerimientos, cuales necesitan, dividirlos y documentarlos, todo debe estar correctamente documentado. Existe un estándar llamado IEEE 830 SRS para una adecuada especificación de requerimientos para el desarrollo de Software.

##### 3.1.1 Propósito

El propósito de esta especificación de requisitos es definir los requerimientos de los módulos de la aplicación IMPLEMENTACIÓN DEL SISTEMA AUTOMATIZADO DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE desarrollado por Paúl Vásquez Méndez como proyecto final previo a la obtención del título de Ingeniero en Sistemas.

Esta especificación está destinada a ser leída tanto por el asesor del presente proyecto, desarrolladores actuales y futuros, así como a cualquier usuario interesado en esta aplicación.

##### 3.1.2 Alcance

Con una aplicación web nos permitirá automatizar, analizar, controlar, organizar de mejor manera la atención de primer nivel hacia el segundo nivel y viceversa sistematizando el formulario de referencia y contrareferencia para que este proceso sea lo más eficiente posible, evitando de esta manera las extensas filas para ser atendidos, debido a que es un procedimiento vital que diariamente realiza el Hospital. De esta forma obtendremos un

acceso rápido y sencillo hacia el manejo de turnos en estadística especialmente en pacientes que acuden de las diferentes partes de la provincia e inclusive reflejar la demanda insatisfecha que existe en esta casa de salud.

### 3.1.3 Perspectiva del producto

El producto es una aplicación web asado en web por lo tanto requiere de configuración para acceder a la red. También requiere software de base de datos. En cuanto a la disponibilidad de memoria no es de mayor trascendencia y tampoco necesita ser instalado, puesto que esta aplicación se ejecutará en el browser instalado en la PC.

### 3.1.4 Funcionalidad del producto

El software de Referencia y Contrareferencia debe realizar básicamente las siguientes funciones:

- I. Seguridad
  - a. Gestión de usuarios
  - b. Gestión de roles
- II. Creación
  - a. Apertura de Historia Clínica
- III. Formulario
  - a. Diagnóstico médico
  - b. Envío de formulario
  - c. Envío de asignación de turnos
- IV. Reportes
  - a. Pacientes por unidad operativa
  - b. Recepción de turno
  - c. Pacientes asignados
  - d. Referencia justificada
- V. Auditoria
  - a. Control de accesos
  - b. Control de ingresos

### 3.1.5 Restricciones

La restricción principal del software es la modificación de datos ingresados por el médico ya que por políticas en la ley de auditoría médica los formularios deben ser sin correcciones, se podrían realizar modificaciones pero directamente desde el motor de base de datos PostgreSQL.

### 3.1.6 Suposiciones y dependencias

Sistema de Gestión de Estadística del HSVP para la asignación de turnos y atención médica.

### 3.1.7 Evolución previsible del sistema

Explícitas no existen, pero pueden existir en el futuro.

### 3.1.8 Comunes de las interfaces

#### 3.1.8.1. Interfaces de usuario

- Multisesión, varios usuarios pueden acceder al sistema al mismo tiempo, pero un usuario no puede iniciar varias sesiones simultáneamente.
- Basado en menús.

#### 3.1.8.2. Interfaces de Hardware

Cada uno de los clientes deberá tener acceso a Internet por medio de la cual tendrán acceso al sistema, la aplicación estará instalada en una cuchilla del servidor blade del HSVP.

### 3.1.8.3. Interfaces de software

La aplicación contará con formularios, es decir una ventana que recogerá los datos suministrados por el médico los registra en la aplicación y los devuelve para que realice el proceso correspondiente.

### 3.1.8.4. Requisitos de rendimiento

- **Seguridad**

El sistema será el único medio de administrar y gestionar Referencias y Contrareferencias enviadas al HVSP.

- **Fiabilidad**

La aplicación será fiable con un 95% de soporte a fallos, con esto se da a entender que el sistema tiene bajo porcentaje de fallos y por lo tanto se garantiza una buena fiabilidad.

- **Disponibilidad**

El sistema deberá estar diseñado para trabajar 24/7 ya que en ciertos subcentros se labora los 7 días de la semana.

- **Mantenibilidad**

El desarrollo deberá permitir que el sistema quede abierto a cambios en la codificación posteriores, en caso de ser necesario.

- **Portabilidad**

La aplicación será altamente eficaz, ya que podrá ser ejecutada en cualquier navegador web independientemente del Sistema Operativo.

### 3.2 Fase Exploratoria

#### 3.2.1 Fase Exploratoria

##### 3.2.1.1 Análisis de Requerimientos

Obtenemos una idea clara del uso del sistema de Referencia y contrareferencia recopilando información del sector salud de la Zona 1 (Esmeraldas, Carchi, Imbabura y Sucumbíos) sobre el manejo y el uso por parte de los profesionales del sector salud, luego de analizar los diferentes actores internos (Administrador, Médico, Estadístico) y actores externos (Paciente).

En primera instancia se establece la distribución de las agendas en las diferentes unidades operativas, las cuales se ubican en diferentes sectores de cada una de las provincias de las provincias que integran la zona 1 tomando en cuenta el déficit de un sistema el cual permita automatizar la forma manual con que realizan en la actualidad el manejo de las agendas para las atenciones médicas en las unidades operativas que pertenecen en MSP, de esta forma se permite evitar la pérdida de formularios de las diferentes atenciones realizadas.

Para optimizar esta forma de atención se ha tomado en cuenta los siguientes perfiles de usuario:

- Administrador
- Médico
- Estadístico
- Paciente

## Referencia y Contrareferencia

---

- **Administrador** permite ingresar al sistema, crear usuarios tanto médicos como estadísticos, permite crear unidades operativas las cuales son los centros y subcentros de salud de la zona 1, modificar datos informativos del pacientes como nombres, apellidos, fecha de nacimiento, dirección, exceptuando la cédula de ciudadanía la cual es clave principal, crea nuevos registros **CIE 10** en el caso de aparecer nuevas enfermedades tal es el caso AH1N1, genera reportes de las personas las cuales accedieron al sistema puede ser por fechas o por usuarios.
- **Médico** permite crear y guardar pacientes los cuales van hacer atendidos en las unidades operativas y guardarlos, ingresar y guardar los datos al formulario de atenciones, permite revisar atenciones subsecuentes realizadas en la unidad operativa.
- **Estadístico** permite revisar atenciones subsecuentes realizadas en las diferentes unidades operativas, ingresa turnos asignados por el estadístico, revisa atenciones futuras, permite realizar reportes de procedencia de donde los pacientes son atendidos, permite un reporte de referencias justificadas las cuales son válidas por el médico del Hospital, permite reportes de efectivas y recibidas en este caso las personas que han sido atendidas y enviadas las contrareferencia a las unidades operativas, reporte de accesos de los estadísticos de las unidades, permite imprimir las referencias y reportes esta es opcional para evitar gasto de material ya que el sistema es para evitar el consumo de papel.
- **Invitado** permite ingresar al sistema de forma online ya sea mediante apellidos, cédula o fecha de nacimiento, revisar el día y la hora de atención.
- Los perfiles tanto como Administrador, Médico y Estadístico tiene un acceso directo al sistema para realizar procesos.

- El perfil paciente únicamente va a permitir realizar búsquedas, no tiene un acceso directo a los procesos del sistema.
  - Los usuarios Médicos, Estadísticos deberán permanecer activos, caso contrario no les permitirá ingresar a sus respectivos módulos.
2. Los usuarios de cada uno de los perfiles del sistema tendrán que loggarse para ingresar al mismo, el sistema verificará el usuario, además comprobará si se encuentra activo, una vez establecido esto le dará ingreso al módulo que le corresponda a este perfil de usuario, caso contrario no le permitirá ingresar a dicho módulo y regresará a la página de loggeo.

### 3.2.1.2 Programación XP

Se decidió utilizar metodología XP por su comunicación, su realimentación o reutilización del código desarrollado. Al utilizar sus 4 fases previas de desarrollo:

- **Planificación**, en esta parte el programador se le permitirá realizar una recopilación de información a los actores directos o en este caso a los propios clientes, mediante Historias de Usuarios, analizando cada uno de los roles, la prioridad del negocio y su riesgo en el desarrollo, juntos de la mano con reuniones de seguimiento, permitiendo corregir el proceso cuando esto falle, se trabajará al igual que los casos de uso.
- **Diseño**, en esta parte nos permitirá mantener la coherencia de nombres de todo aquello que se va a implementar, tomando en cuenta las posibles soluciones al problema, representar los objetos, la clase a donde pertenece representada en tarjetas de igual forma analizando los objetivos que debe cumplir cada objeto.
- **Desarrollo**, un análisis a fondo utilizando estándares de implementación utilizando directamente el código desarrollado

- dentro del software, con disposición del cliente, siempre dejando las optimizaciones para el final.
- **Pruebas**, el pilar básico de nuestro desarrollo, protecciones contra fallos, soluciones, test y la parte principal pruebas de aceptación, evaluación del cliente,

### 3.2.1.3 Historias de Usuario

- Yo administrador registro nuevo usuario para acceder
- Yo administrador creo unidades operativas
- Yo administrador modifíco datos del paciente
- Yo administrador creo enfermedades CIE 10
- Yo administrador genero reporte de ingresos.
- Yo médico/estadístico ingreso al sistema
- Yo médico creo y guardar datos de pacientes
- Yo médico ingreso y guardo datos al formulario 053
- Yo médico/estadístico reviso atenciones subsecuentes
- Yo médico/estadístico reviso atenciones de referencias.
- Yo estadístico asigno e imprimo atención futura.
- Yo estadístico genero reporte de procedencias.
- Yo estadístico genero reporte de referencias justificadas.
- Yo estadístico imprimo referencia.
- Yo invitado ingreso al sistema
- Yo invitado reviso día hora y número de turno.
- Yo invitado imprimo asignación.

### 3.2.1.4 Historia de Usuario 1

<b>Historias de usuario</b>	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	1
<b>Título:</b>	Registro nuevo usuario para acceder
<b>Fecha:</b>	26/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px;"> <p>El administrador tiene el rol principal para poder crear nuevos usuarios los cuales podrán ingresar al sistema. En la creación se solicitará los datos informativos básicos del usuario, el mismo que tendrá acceso directo con el sistema ya sea en este caso Médico o Estadístico. El sistema trabajará con la cédula de ciudadanía como user y el password por defecto hasta su primer acceso será el mismo que la cédula, si el usuario olvidase la clave tendrá que comunicar al administrador para su respectivo reseteo. En el caso que el médico dejase la unidad operativa este usuario sería únicamente desactivado más no eliminado.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Líder de TICs: Ing. Juan Carlos Armas</p> <p>ROL ADMINISTRADOR PRIORIDAD DEL NEGOCIO 1 (ALTA)</p> </div>

**Tabla 2.** Historia de usuario 1

### 3.2.1.5 Historia de Usuario 2

Historias de usuario	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	2
<b>Título:</b>	Crear unidades Operativas
<b>Fecha:</b>	26/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px;"> <p>El Administrador está en la capacidad de crear nuevas unidades operativas en las diferentes parroquias o provincias si fuese el caso con su respectivo código de área, el cual es asignado por la Coordinación de Salud N° 1 para el respectivo enlace con el médico y así realizar las atenciones dentro del sistema.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Líder de ADMISIONES Econ. José Hidrobo G.</p> <p>ROL ADMINISTRADOR PRIORIDAD DEL NEGOCIO 1 (ALTA)</p> </div>

**Tabla 3.** Historia de Usuario 2

### 3.2.1.6 Historia de Usuario 3

Historias de usuario	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	3
<b>Título:</b>	Modifico datos del paciente
<b>Fecha:</b>	26/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px; min-height: 150px;"> <p>El Administrador está en la capacidad de modificar datos del paciente ya sean estos fecha de nacimiento, nombres, apellidos, estado civil, género, empresa donde trabaja. En si únicamente datos informativos únicamente.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Líder de ADMISIONES Econ. José Hidrobo G.</p> <p>ROL ADMINISTRADOR PRIORIDAD DEL NEGOCIO 8 (BAJA)</p> </div>

**Tabla 4.** Historia de Usuario 3

### 3.2.1.7 Historia de Usuario 4

Historias de usuario	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	4
<b>Título:</b>	Creo enfermedades CIE 10
<b>Fecha:</b>	26/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px;"> <p>El Administrador está en la capacidad de aumentar enfermedades CIE 10 para el ingreso y reporte de mortalidad y morbilidad en los pacientes que acudan con patologías a las unidades operativas clasificando y codificando de las enfermedades y una amplia variedad de signos, síntomas, hallazgos anormales, denuncias, circunstancias sociales y causas externas de daños y/o enfermedad autorizados por el MSP.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Asistente de ADMISIONES Sra. Elsa Vega</p> <p>ROL ADMINISTRADOR PRIORIDAD DEL NEGOCIO 5 (MEDIA)</p> </div>

**Tabla 5.** Historia de Usuario 4

### 3.2.1.8 Historia de Usuario 5

Historias de usuario	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	5
<b>Título:</b>	Genero reporte de ingresos
<b>Fecha:</b>	26/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px; min-height: 150px;"> <p>El Administrador está en la capacidad de generar reportes de los ingresos al sistema realizados. Los cuales detallan el día la hora el nombre del médico, la unidad operativa a la fue asignado.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Líder de ADMISIONES Econ. José Hidrobo G.</p> <p>ROL ADMINISTRADOR PRIORIDAD DEL NEGOCIO 4 (MEDIA)</p> </div>

**Tabla 6** Historia de Usuario 5

## Referencia y Contrareferencia

### 3.2.1.9 Historia de Usuario 6

Historias de usuario	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
Número de historia:	6
Título:	Ingreso al sistema
Fecha:	26/11/13
Descripción de la historia:	<p>Tanto el médico como el estadístico previamente tuvieron que haber registrado sus datos al administrador del sistema Para poder ingresar al sistema se necesitan la cédula y la clave del usuario, a continuación pedirá ser modificada, si el usuario olvida su clave, la solicitud se la realizaría directamente al administrador del mismo. Ya que cada usuario maneja diferente interfaz gráfica.</p>
Anotaciones:	<p>La información de esta HISTORIA DE USUARIO se la obtuvo de Líder de TICs: Ing. Juan Carlos Armas</p> <p>ROL MÉDICO / ROL ESTADÍSTICO PRIORIDAD DEL NEGOCIO 1 (ALTA) RIESGO EN DESARROLLO 3 (ALTA)</p>

**Tabla 7** Historia de Usuario 6

### 3.2.1.10 Historia de Usuario 7

Historias de usuario	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	7
<b>Título:</b>	Creo datos y guardo datos de pacientes
<b>Fecha:</b>	26/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px;"> <p>El médico está en la posibilidad de crear pacientes por una única vez, los parámetros para su ingreso son su cédula de identidad, apellidos tanto paterno como materno, sus nombres, su fecha de nacimiento, su género, su estado civil, su instrucción, la empresa donde trabaja, si tiene seguro de salud.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Líder de TICs Ing, Juan Carlos Armas</p> <p>ROL MÉDICO PRIORIDAD DEL NEGOCIO 1 (ALTA)</p> </div>

**Tabla 8.** Historia de Usuario 7

### 3.2.1.11 Historia de Usuario 8

<b>Historias de usuario</b>	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	8
<b>Título:</b>	Ingreso y guardo datos al formulario 053
<b>Fecha:</b>	26/11/13
<b>Descripción de la historia:</b>	<p>El usuario médico está en la posibilidad de ingresar datos al formulario 053 en nuestro caso nuestra hoja de referencia, los parámetros que incluyen dentro del sistema son obligatorios, para permitir avanzar y guardar los mismos, tales como motivo de referencia, resumen del cuadro clínico, hallazgos relevantes de exámenes y procedimientos, diagnósticos CIE10, si la enfermedad del CIE10 es presuntiva o definitiva y el plan de tratamiento realizado.</p>
<b>Anotaciones:</b>	<p>La información de esta HISTORIA DE USUARIO se la obtuvo de Director Médico Asistencia Dr. Edisson Ayala Arroyo</p> <p>ROL MÉDICO PRIORIDAD DEL NEGOCIO 1 (ALTA)</p>

**Tabla 9.** Historia de Usuario 8

### 3.2.1.12 Historia de Usuario 9

<b>Historias de usuario</b>	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	9
<b>Título:</b>	Revisar atenciones subsecuentes
<b>Fecha:</b>	29/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px; min-height: 150px;"> <p>Tanto el médico como el estadístico están en la posibilidad de revisar las atenciones que se realizaron en la unidad operativa o en el hospital base o viceversa a manera de una historia clínica del paciente para verificar la evolución para ser atendido.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Gerencia Médica Dra. Yolanda Checa B.</p> <p>ROL MÉDICO/ESTADÍSTICO PRIORIDAD DEL NEGOCIO 5 (MEDIA)</p> </div>

**Tabla 10.** Historia de Usuario 9

### 3.2.1.13 Historia de Usuario 10

<b>Historias de usuario</b>	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	10
<b>Título:</b>	Reviso atenciones de referencias.
<b>Fecha:</b>	29/11/13
<b>Descripción de la historia:</b>	<p>Tanto el Médico como el Estadístico pueden realizar una revisión de las referencias de cada uno de los pacientes, en este caso el sistema desplegará el listado de las anteriores atenciones con sus fechas respectivas de atención, ya sean estas en la unidad operativas o en el hospital base o a su vez en el hospital de especialidades.</p>
<b>Anotaciones:</b>	<p>La información de esta HISTORIA DE USUARIO se la obtuvo de Líder de ADMISIONES Econ. José Hidrobo G.</p> <p>ROL MÉDICO/ESTADÍSTICO PRIORIDAD DEL NEGOCIO 5 (MEDIA)</p>

**Tabla 11.** Historia de Usuario 10

### 3.2.1.14 Historia de Usuario 11

Historias de usuario	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	11
<b>Título:</b>	Asigno e imprimo atención futura.
<b>Fecha:</b>	29/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px;"> <p>El estadístico tiene la posibilidad de asignar el turno para el médico especialista el cual es solicitado por la unidad operativa, este dependerá del sistema automatizado de estadística que tiene el Hospital San Vicente de Paúl, de acuerdo al número de pacientes que recibe cada médico y el día de atención, se asignará el turno e imprimirá para entregar a la unidad operativa para ser informado al paciente.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Asistente de ADMISIONES Ing. José Ruiz</p> <p>ESTADÍSTICO PRIORIDAD DEL NEGOCIO 5 (MEDIA)</p> </div>

**Tabla 12.** Historia de Usuario 11

### 3.2.1.15 Historia de Usuario 12

<b>Historias de usuario</b>	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	12
<b>Título:</b>	Genero reporte de procedencias
<b>Fecha:</b>	29/11/13
<b>Descripción de la historia:</b>	El estadístico tiene la posibilidad de generar reportes sobre las procedencias de los pacientes para así obtener un informe de las enfermedades por lugar de procedencia, con esto se puede dar las morbilidades más comunes en un lugar o detallar enfermedades catastróficas en una zona.
<b>Anotaciones:</b>	La información de esta HISTORIA DE USUARIO se la obtuvo de Asistente de ADMISIONES Ing. José Ruiz  ESTADÍSTICO PRIORIDAD DEL NEGOCIO 8 (BAJA)

**Tabla 13.** Historia de Usuario 12

### 3.2.1.16 Historia de Usuario 13

<b>Historias de usuario</b>	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	13
<b>Título:</b>	Genero reporte de referencias justificadas.
<b>Fecha:</b>	29/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px;"> <p>El estadístico está en la capacidad de generar reportes e imprimirlos de las referencias justificadas en este caso el médico del centro de salud u hospital base envía la hoja de referencia y el médico tratante del Hospital General o de Especialidades, realiza una evaluación si el paciente enviado verdaderamente debía recibir atención en dicha casa de salud. Caso contrario se emite al médico de Salud Pública un reporte indicando que dicha patología se podía atender en el mismo centro de salud.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Asistente de ADMISIONES Ing. José Ruiz</p> <p>ESTADÍSTICO PRIORIDAD DEL NEGOCIO 8 (BAJA)</p> </div>

**Tabla 14.** Historia de Usuario 13

### 3.2.1.17 Historia de Usuario 14

Historias de usuario	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	14
<b>Título:</b>	Imprimo referencia
<b>Fecha:</b>	29/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px; min-height: 150px;"> <p>El estadístico tiene la posibilidad de realizar la impresión del documento para ser anexado a la Historia Clínica de lo contrario permanecerá activo dentro de la base de datos hasta futuras revisiones o atenciones.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Asistente de ADMISIONES Ing. José Ruiz</p> <p>ESTADÍSTICO PRIORIDAD DEL NEGOCIO 7 (MEDIA)</p> </div>

**Tabla 15.** Historia de Usuario 14

### 3.2.1.18 Historia de usuario 15

Historias de usuario	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	15
<b>Título:</b>	Ingreso al sistema
<b>Fecha:</b>	29/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px; min-height: 150px;"> <p>El invitado es aquella persona que puede acceder al sistema únicamente mediante un número de cédula. Generalmente es aquella persona que vía web podrá tener un acceso a sus atenciones.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Asistente de ADMISIONES Ing. José Ruiz</p> <p>INVITADO PRIORIDAD DEL NEGOCIO 4 (MEDIA)</p> </div>

**Tabla 16.** Tabla de Usuario 15

### 3.2.1.19 Historia de Usuario 16

Historias de usuario	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	16
<b>Título:</b>	Reviso día hora y número de turno.
<b>Fecha:</b>	29/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px; min-height: 150px;"> <p>El invitado podrá realizar una búsqueda de su propio historial de atenciones médicas en las cuales únicamente podrá observar sus atenciones subsecuentes en las cuales detallará la unidad de donde fue enviado hacia que médico especialista fue referido, la hora de atención y el consultorio médico para ser atendido.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Asistente de ADMISIONES Ing. José Ruiz</p> <p>INVITADO PRIORIDAD DEL NEGOCIO 8 (BAJA)</p> </div>

**Tabla 17.** Tabla de Usuario 16

### 3.2.1.20 Historia de Usuario 17

Historias de usuario	
<b>SISTEMA DE REFERENCIA Y CONTRAREFERENCIA PARA EL HOSPITAL SAN VICENTE DE PAÚL MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.</b>	
<b>Número de historia:</b>	17
<b>Título:</b>	Imprimo asignación
<b>Fecha:</b>	29/11/13
<b>Descripción de la historia:</b>	<div style="border: 1px solid black; padding: 10px; min-height: 150px;"> <p>El invitado podrá realizar una impresión del médico que a sido asignado en el Hospital General o en el Hospital de Especialidades para recordar el día, la hora y el consultorio a ser atendido.</p> </div>
<b>Anotaciones:</b>	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <p>La información de esta HISTORIA DE USUARIO se la obtuvo de Asistente de ADMISIONES Ing. José Ruiz</p> <p>INVITADO PRIORIDAD DEL NEGOCIO 8 (BAJA)</p> </div>

**Tabla 18.** Tabla de Usuario 17

### 3.3 Planificación

#### 3.3.1 Diagramas de casos de uso

En lenguaje de modelado unificado, un diagrama de casos de uso es una forma de diagrama de comportamiento UML pero de forma mejorada. De esta forma nos permite realizar una notación gráfica para representar estos casos de uso. Estas notaciones definen la naturaleza del caso de uso, detallándolos de mejor forma.

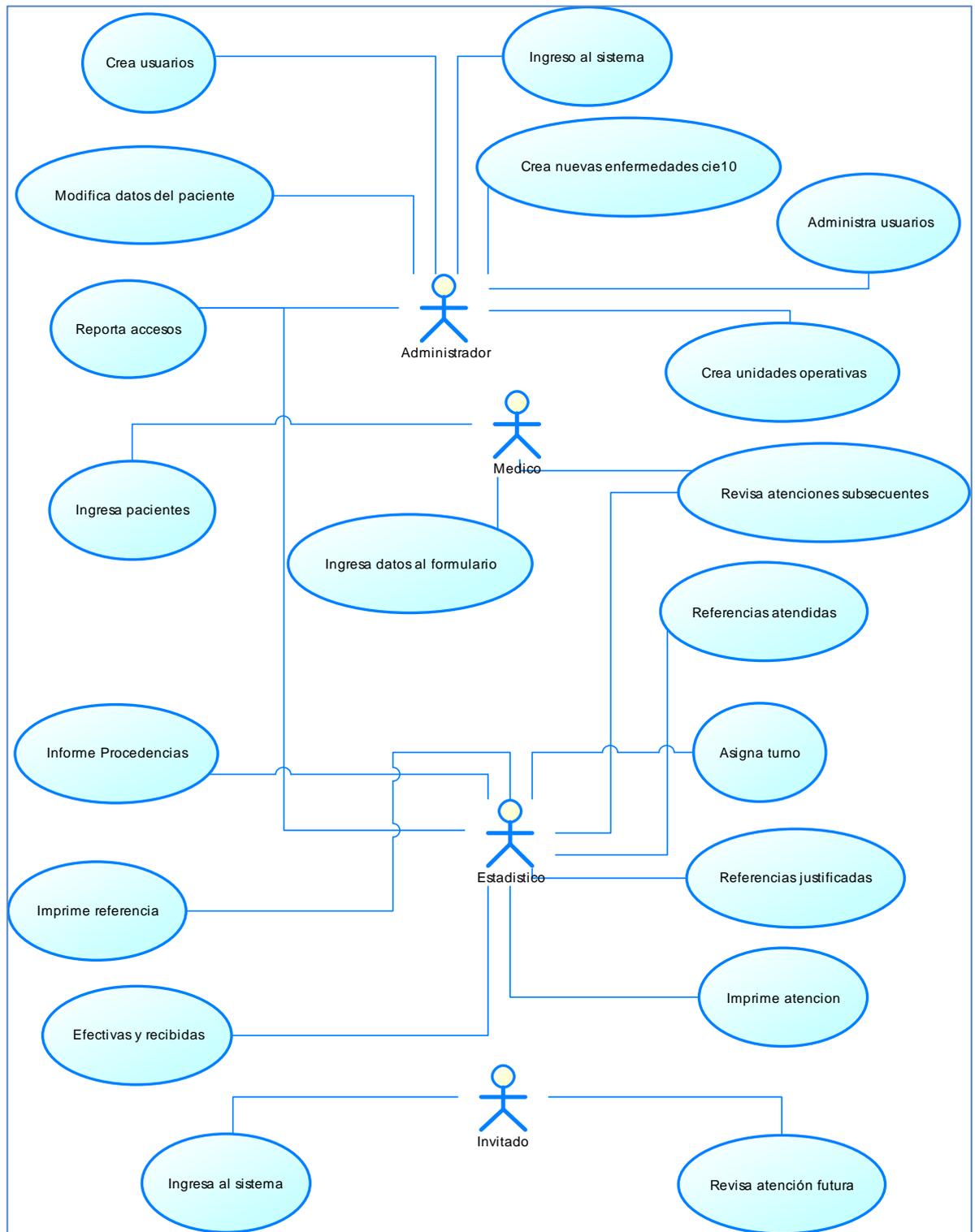
Su ventaja principal es la facilidad para interpretarlos, lo que sean especialmente útiles en la comunicación con el cliente.

Tiene 3 elementos básicos en los cuales vamos a señalar los siguientes:

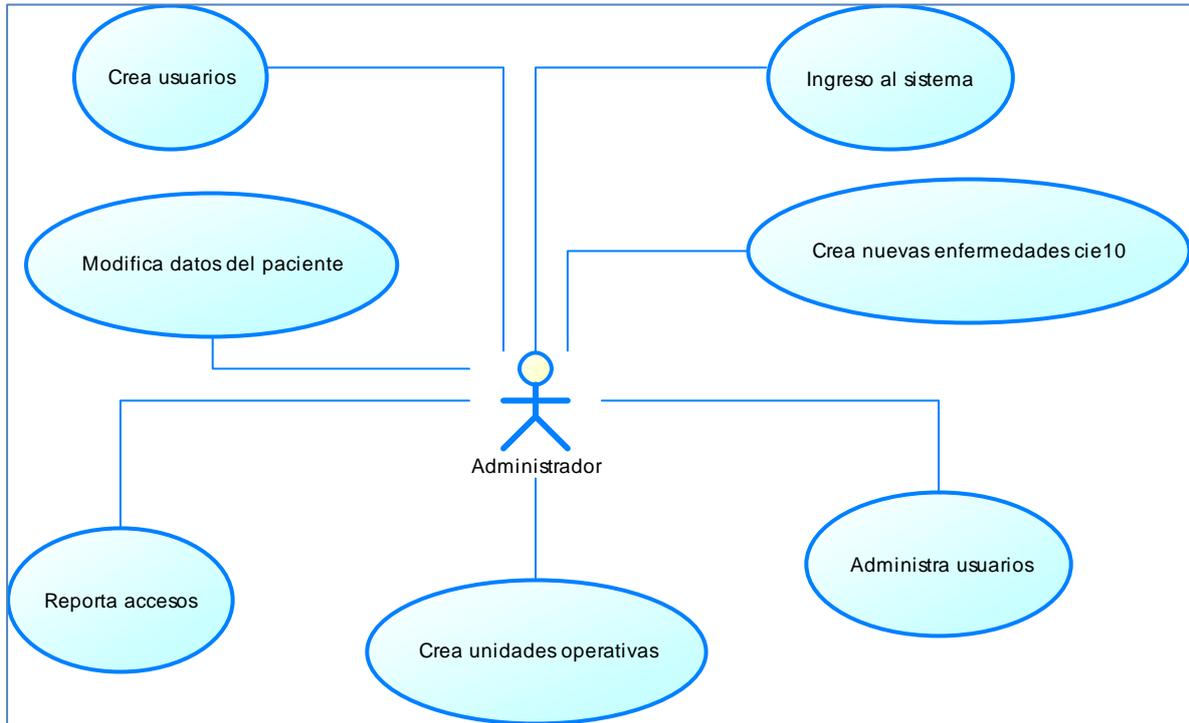
- **Actores**, estos representan un tipo de usuario en el sistema, no es necesario que sea un ente humano, a la vez puede ser un sistema existente dentro de una empresa.
- **Caso de uso**, es la tarea que va a desarrollarse a cabo con el apoyo del sistema el cual estamos desarrollando, generalmente se la representa con un óvulo.
- **Asociaciones**, hay una asociación entre un actor y un caso de uso si el actor interactúa con el sistema para llevar a cabo el caso de uso.
- **Escenario**, es la interacción entre el sistema y los actores, que pueden ser descritos mediante una secuencia de mensajes.

#### 3.3.2 Los Actores

- Administrador
- Médico
- Estadístico
- Invitado



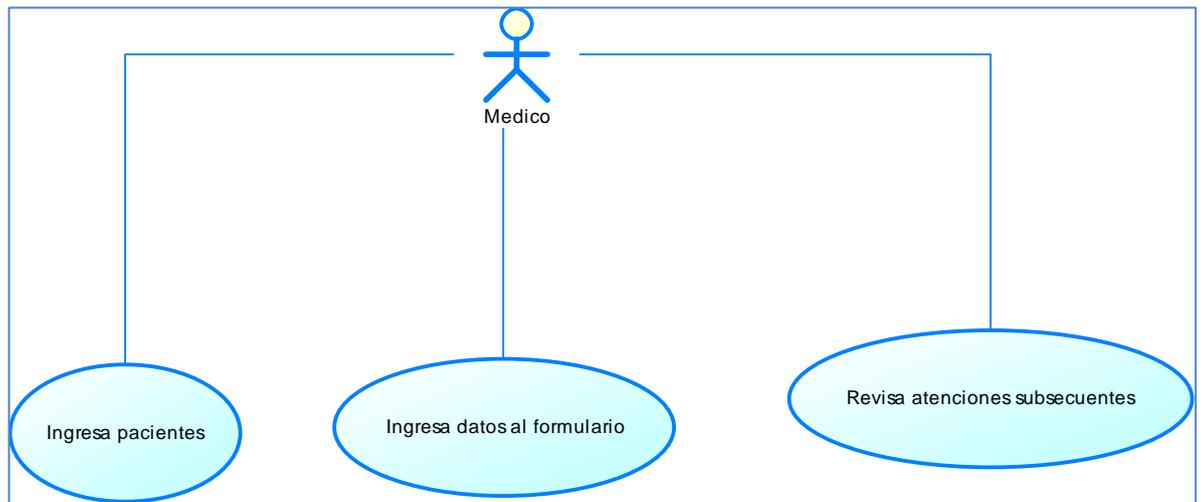
**Figura 5.** Diagrama Caso de uso general



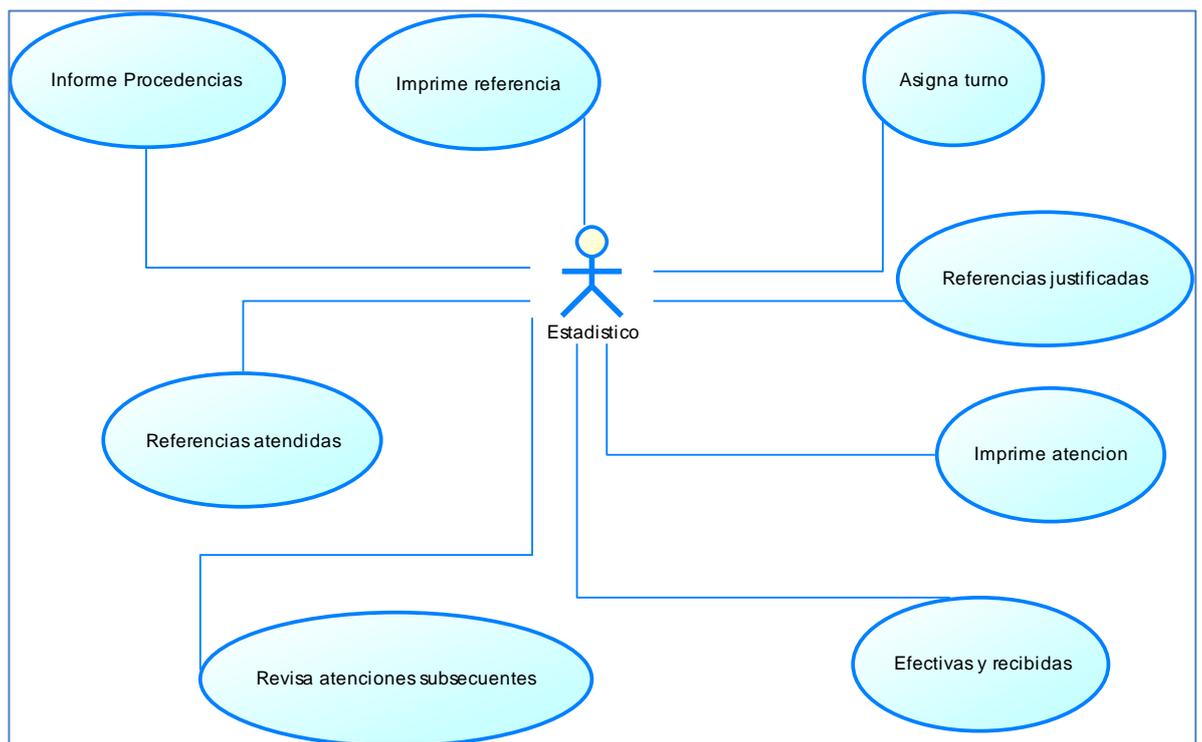
**Figura 6.** Diagrama Caso de uso del perfil Administrador



**Figura 7.** Diagrama Caso de uso del perfil Invitado



**Figura 8.** Diagrama Caso de uso del perfil Médico



**Figura 9.** Diagrama Caso de uso del perfil Estadístico

## 3.3.3 Diagrama de Clases

Es esta parte se mostrará de forma gráfica un diagrama estático que describe el sistema en el cual en si nuestro diseño de la base de datos cada tabla es un objeto dentro de nuestro sistema con sus respectivas clases las cuales son comunes entre sí.

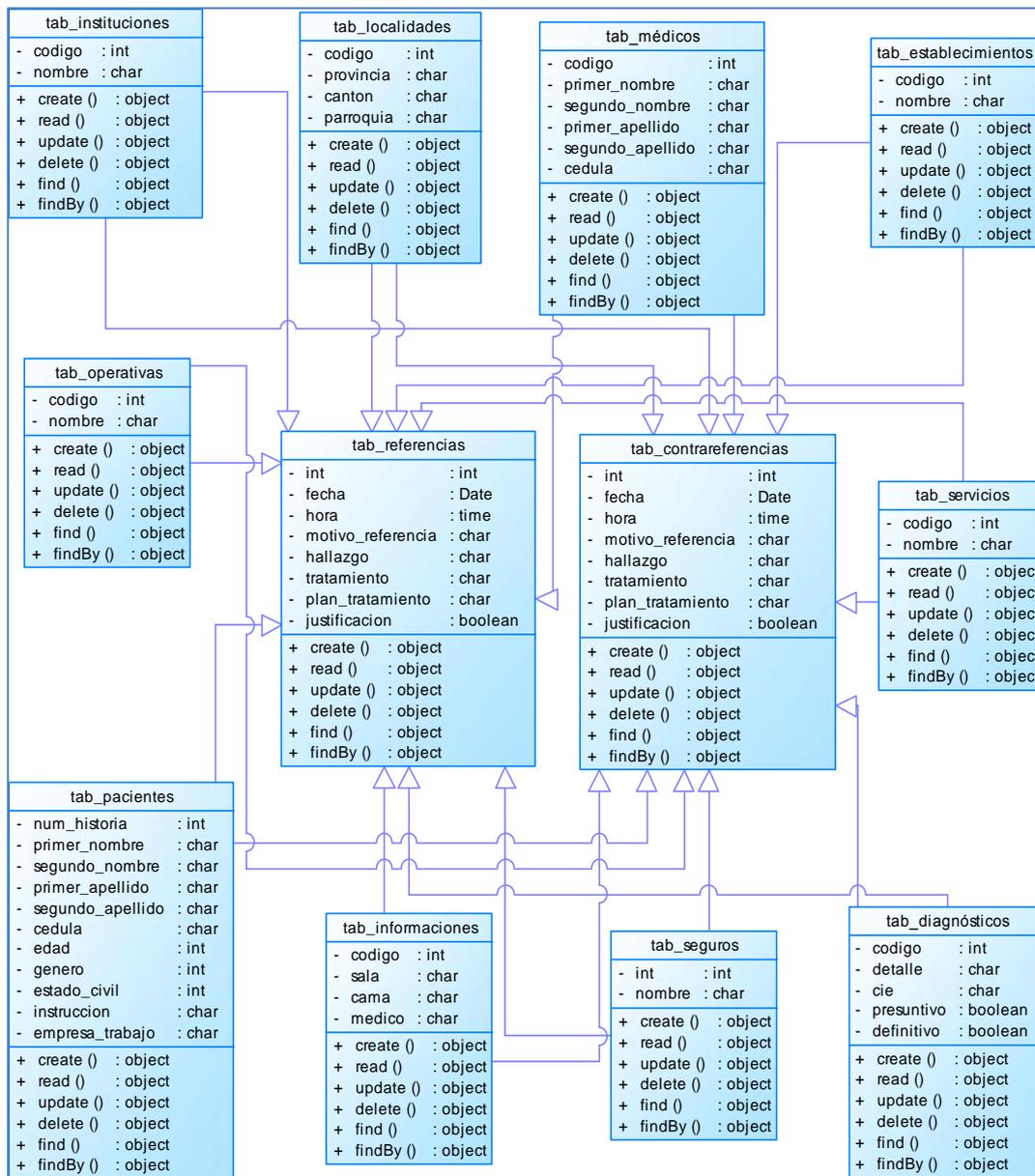


Figura 10. Diagrama de Clases

## 3.3.4 Modelo de Datos Relacional

En este modelo de datos relacional, es el que se ha implantado la base de datos del Sistema de Referencia y contrareferencia se estableció interconexiones (relaciones) entre los datos (que están en las tablas) y a través de dichas conexiones relacionar una o más tablas.

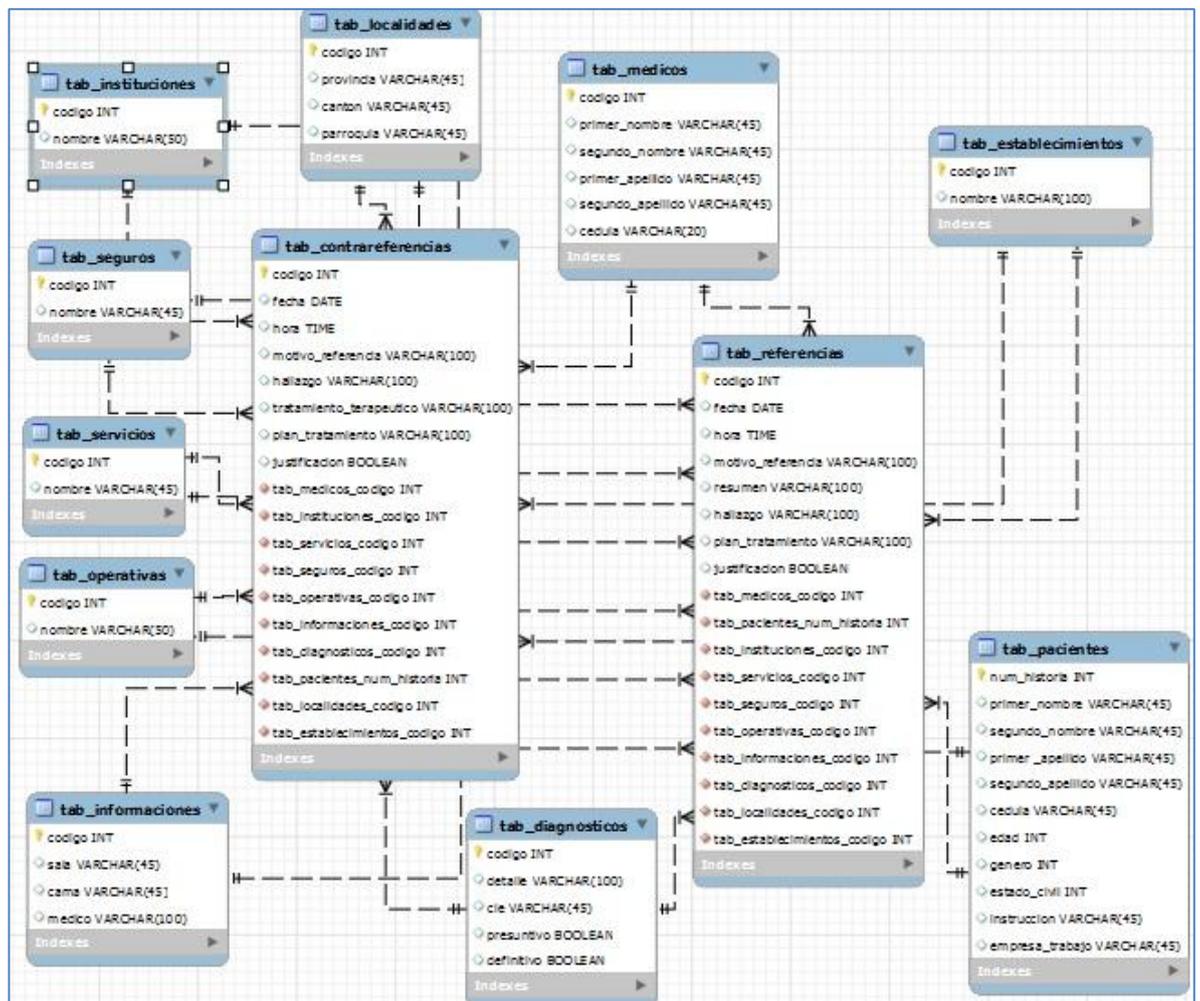


Figura 11. Modelo de Datos Relacional

### 3.3.5 Diccionario de Datos

**Nombre del Proyecto:** Implementación del sistema de Referencia y Contrareferencia para el Hospital San Vicente de Paúl mediante la utilización de software libre.

- **Nombre del programa:** SRC
- **Lenguaje de Programación:** Php
- **Motor de Base de datos:** PostgreSQL
- **Nombre de la base de datos:** hsvp

#### 2.2.1.2. Tablas

N°	Nombre	Descripción	Clave primaria
1	tab_instituciones	Se utiliza para almacenar los nombres y los códigos del hospital general.	codigo
2	tab_seguros	Servirá para ingresar las diferentes aseguradoras que existen tanto públicas como privadas.	codigo
3	tab_servicios	Se ingresarán los diferentes servicios los cuales cuenta el hospital general.	codigo
4	tab_operativas	Permitirá el ingreso de las diferentes unidades operativas existentes.	codigo
5	tab_informaciones	Nos facilita el ingreso en el caso que el paciente sea referido desde la misma unidad operativa hacia hospitalización.	codigo
6	tab_localidades	Guarda todas las localidades en donde se realicen las referencias para los hospitales generales.	codigo

## Referencia y Contrareferencia

---

7	tab_diagnosticos	Nos permite almacenar los diferentes diagnósticos que el médico ingrese incluyendo su codificación cie 10 y su diagnóstico presuntivo o definitivo.	codigo
8	tab_medicos	En esta tabla nos permitirá guardar los datos principales de los médicos incluyendo su código de médico para el acceso al sistema.	codigo
9	tab_establecimientos	Servirá para almacenar si el establecimiento es una unidad operativa o es un hospital base.	codigo
10	tab_pacientes	Permitirá el ingreso de los datos personales de los pacientes para ser guardados en la base de datos.	num_historia
11	tab_referencias	Se utiliza para almacenar la información de la hoja principal del sistema en el caso de referencias.	codigo
12	tab_contrareferencias	Se utiliza para almacenar la información de la hoja principal del sistema en el caso de contrareferencias.	codigo

**Tabla 19.** Tablas Bdds

## Referencia y Contrareferencia

---

### tab\_instituciones

Campos	Descripción	Tipo de Dato
codigo	clave primaria	serial
nombre	nombre de la unidad operativa	character varying (50)

**Tabla 20.** Tabla Instituciones

### tab\_seguros

Campos	Descripción	Tipo de Dato
codigo	clave primaria	serial
nombre	nombre de la aseguradora	character varying (45)

**Tabla 21.** Tabla Seguros

### tab\_servicios

Campos	Descripción	Tipo de Dato
codigo	clave primaria	serial
nombre	servicio al que será referido	character varying (45)

**Tabla 22.** Tabla Servicios

### tab\_operativas

Campos	Descripción	Tipo de Dato
codigo	clave primaria	serial
nombre	nombre de la unidad operativa	character varying (50)

**Tabla 23.** Unidades Operativas

## Referencia y Contrareferencia

---

### tab\_informaciones

Campos	Descripción	Tipo de Dato
codigo	clave primaria	Serial
sala	sala a la que es referido el paciente	character varying (45)
Cama	cama a la que será asignado el paciente	character varying (45)
Medico	médico el que realizó la atención	character varying (100)

**Tabla 24.** Tabla Informaciones

### tab\_localidades

Campos	Descripción	Tipo de Dato
Código	clave primaria	Serial
Provincia	provincias de las unidades operativas	character varying (45)
Canton	cantones de las unidades operativas	character varying (45)
Parroquia	parroquias de las unidades operativas	character varying (45)

**Tabla 25.** Tabla Localidades

### tab\_diagnosticos

Campos	Descripción	Tipo de Dato
Código	clave primaria	Serial
Detalle	detalle de la enfermedad	character varying (100)
Cie	codificación de la enfermedad	character varying (45)
Presuntivo	registra una enfermedad presuntiva o definitiva	boolean

**Tabla 26.** Tabla Diagnósticos

## Referencia y Contrareferencia

---

### tab\_medicos

<b>Campos</b>	<b>Descripción</b>	<b>Tipo de Dato</b>
Código	clave primaria	Int
primer nombre	primer nombre del médico	character varying (45)
segundo nombre	segundo nombre del médico	character varying (45)
primer apellido	primer apellido del médico	character varying (45)
segundo apellido	segundo apellido del médico	character varying (45)
Cedula	cédula del médico	character varying (20)

**Tabla 27.** Tabla Médicos

### tab\_establecimientos

<b>Campos</b>	<b>Descripción</b>	<b>Tipo de Dato</b>
Código	clave primaria	Serial
Nombre	nombre del establecimiento	character varying (100)

**Tabla 28.** Tabla establecimientos

### tab\_pacientes

Campos	Descripción	Tipo de Dato
num_historia	clave primaria	Int
primero_nombre	primer nombre del paciente	character varying (45)
segundo_nombre	segundo nombre del paciente	character varying (45)
primer_apellido	primer apellido del paciente	character varying (45)
segundo_apellido	segundo apellido del paciente	character varying (45)
Cedula	cedula del paciente	character varying (20)
Edad	edad del paciente	Int
Genero	genero del paciente	Int
estado_civil	estado civil del paciente	Int
Instrucción	instrucción académica del paciente	character varying (45)
Empresa	empresa donde labora actualmente el paciente	character varying (45)

**Tabla 29.** Tablas pacientes

### tab\_referencias

Campos	Descripción	Tipo de Dato
Código	clave primaria	Serial
Fecha	fecha del sistema	Date
Hora	hora del sistema	Time
motivo_referencia	el motivo que el paciente visita al médico	character varying (100)
Resume	detalle del médico en base a chequeo físico	character varying (100)

## Referencia y Contrareferencia

---

Hallazgo	registro del médico de acuerdo a análisis de laboratorio clínico	character varying (100)
plan_tratamiento	el tratamiento más adecuado emitido por el médico	character varying (100)
Justificación	la justificación si es válida por el médico del hospital general	Boolean
Código	clave foránea de la tabla tab_médicos	Serial
num_historia	clave foránea de la tabla tab_pacientes	Int
Código	clave foránea de tab_instituciones	Serial
Código	clave foránea de tab_servicios	Serial
Código	clave foránea de la tabla tab_seguros	Serial
Código	clave foránea de la tabla tab_operativas	Serial
Código	clave foránea de la tabla tab_informaciones	Serial
Código	clave foránea de la tabla tab_diagnosticos	Serial
Código	clave foránea de la tabla tab_localidades	Serial
Código	clave foránea de la tabla tab_establecimientos	Serial

**Tabla 30.** Tabla Referencias

## Referencia y Contrareferencia

---

### tab\_contrareferencias

<b>Campos</b>	<b>Descripción</b>	<b>Tipo de Dato</b>
Código	clave primaria	serial
Fecha	fecha del sistema	date
Hora	hora del sistema	time
motivo_referencia	el motivo que el paciente visita al médico	character varying (100)
Hallazgo	registro del médico de acuerdo a análisis de laboratorio clínico	character varying (100)
tratamiento_terapeutico	Tratamiento que se realizó en el tiempo que estuvo en el hospital general	character varying (100)
plan_tratamiento	el tratamiento más adecuado emitido por el médico	character varying (100)
Justificación	la justificación si es válida por el médico del hospital general	boolean
Código	clave foránea de la tabla tab_médicos	serial
num_historia	clave foránea de la tabla tab_pacientes	int
Código	clave foránea de tab_instituciones	serial
Código	clave foránea de tab_servicios	serial
Código	clave foránea de la tabla seguros	serial
Código	clave foránea de la tabla operativas	serial

Código	clave foránea de la tab_informaciones	serial
Código	clave foránea de la tab_diagnosticos	serial
Código	clave foránea de la tab_localidades	serial
Código	clave foránea de la tab_establecimientos	serial

**Tabla 31.** Tabla Contrareferencias

### 3.4 IEEE 1362

La IEEE (the institute of electrical and electronics engineers), es un instituto internacional dedicado a promover la innovación y la excelencia tecnológica en beneficio de la humanidad. La IEEE dice q para todo trabajo de software es necesario entregar a los clientes la especificación de requerimientos, cuales necesitan, dividirlos y documentarlos, todo debe estar correctamente documentado. Existe un estándar llamado IEEE 1362 para una adecuada especificación de requerimientos para el desarrollo del sistema aquí se puede definir tanto software más hardware más personas más reglamentos más procedimientos

#### 3.4.1.1 Alcance

El presente documento tiene la finalidad de definir los requisitos de sistema que serán necesarios y la base fundamental para estructurar la plataforma de servicios sobre la cual se ejecutara la aplicación web que se desea desarrollar.

Este documento antecede a la Especificación de Requisitos de Software (ERSoftware) mismo que representa la descripción general de la estructura que soportara dicha aplicación web.

La información aquí detallada servirá a los técnicos de desarrollo para estructurar o configurar la plataforma requerida para el perfecto funcionamiento de este proyecto de software.

### **3.4.1.2 Identificación**

Implementación del sistema automatizado de referencia y contrareferencia para el hospital San Vicente de Paúl mediante la utilización de software libre.

### **3.4.1.3 Visión general del documento**

- El alcance de este documento está marcado por:
- Describir los requisitos de tecnología, humanos, materiales, procedimentales y de software para el perfecto funcionamiento de la aplicación web que se desarrollará, requisitos que deben ser provistos o cumplidos por el HSVP.
- Está dirigido especialmente a los técnicos de sistemas del Dpto. de Informática del HSVP adquirente del Sistema.
- Se considera que el contenido total de este documento es CONFIDENCIAL

### **3.4.1.4 Visión general del sistema**

La aplicación web a desarrollar permitirá la automatización del sistema de referencia y contrareferencia de tal forma que el profesional médico pueda realizar su atención y la misma sea recibida en el Hospital San Vicente de Paúl, de esta forma mejorando la eficiencia en la atención con calidad y calidez.

## Referencia y Contrareferencia

---

Se agilizará el tiempo de atención, el tiempo de revisión de la misma, se evitara repetición de recetas médicas e interrupción de la consulta en pacientes nuevos, entre sus prestaciones destacan las siguientes:

- Módulo seguridad
- Módulo creación
- Módulo formulario
- Módulo reportes
- Módulo Auditoría

### 3.4.1.5 Personal Involucrado

Nombre	Paúl Bolívar Vásquez Méndez
Rol	Desarrollador
Categoría profesional	Estudiante
Responsabilidades	<ul style="list-style-type: none"><li>• Codificación de software</li><li>• Documentación interna de software</li><li>• Documentación en bitácoras</li><li>• Coordinar la integración de módulos</li><li>• Diseñar Interfaces del usuario</li><li>• Diseñar la arquitectura del proyecto</li><li>• Documentar el diseño</li><li>• Diseñar base de datos</li></ul>
Información de contacto	0998810274 <a href="mailto:pvasquez@yachay.gob.ec">pvasquez@yachay.gob.ec</a> <a href="mailto:pbvm1783@gmail.com">pbvm1783@gmail.com</a>
Aprobación	OK

**Tabla 32.** Tabla desarrollador

## Referencia y Contrareferencia

---

Nombre	Juan Carlos Armas
Rol	Administrador
Categoría profesional	Líder de TIC's
Responsabilidades	<ul style="list-style-type: none"><li>• Administración</li><li>• Capacitación</li></ul>
Información de contacto	0984688733 <a href="mailto:jcarmas@pucesi.edu.ec">jcarmas@pucesi.edu.ec</a> <a href="mailto:juan.arms@dpsi.gob.ec">juan.arms@dpsi.gob.ec</a>
Aprobación	Personal Calificado

**Tabla 33.** Tabla Administrador

Nombre	José Hidrobo Guzman
Rol	Líder de Admisiones
Categoría profesional	Economista
Responsabilidades	<ul style="list-style-type: none"><li>• Operador del sistema</li></ul>
Información de contacto	0984688737 <a href="mailto:jose.hidrobo@dpsi.gob.ec">jose.hidrobo@dpsi.gob.ec</a>
Aprobación	Personal Calificado

**Tabla 34.** Tabla Estadístico

**Fuente.** Propia

### 3.4.1.6 Descripción del sistema o situación actual

Se realiza de forma manual el registro y atención de pacientes desde las unidades operativas hacia el hospital general en nuestro caso el HSVP. De esta forma existe pérdida de información en el recorrido que realiza este formulario 053.

### 3.4.1.7 Mantenimiento/Soporte

Las políticas de mantenimiento de la Bdds dependerán del departamento de TIC's de la empresa

### 3.4.1.8 Requisitos de la Instalación

- **Hardware**

Servidor como requerimientos mínimos Core i3 2.x  
GHz, 4Gb RAM, 100Gb libres de disco, red  
10/100/1000 Mbps

- **Software de base**

Multiplataforma por ser una aplicación web funciona en todos los sistemas operativos con un navegador base.

- **Software de desarrollo**

PgSQL 8.4  
Ubuntu 12.04  
Apache  
Php5  
NetBeans 7.x

### 3.5 Prototipo de la pantalla principal del sistema

En primera instancia podemos observar la pantalla de ingreso al sistema en la cual solicita usuario y password para el ingreso.



Figura 12. Prototipo de ingreso al sistema

#### 3.5.1 Interfaces del sistema

- **Administrador**

En esta ventana podremos visualizar al usuario Administrador el cual tendrá ciertos privilegios que permitirá interactuar con pacientes, unidades operativas, patologías, seguros, servicios y reportes.



Figura 13. Prototipo del rol administrador del Sistema

- **Médico**

En esta ventana podremos visualizar el usuario Médico el cual tendrá los privilegios que permitirá interactuar directamente con los pacientes, las referencias, los turnos recibidos y las atenciones pendientes.



Figura 14. Prototipo del rol administrador del Sistema

- **Estadístico**

En esta ventana podremos visualizar el usuario Estadístico el cual tendrá los privilegios que permitirá interactuar directamente con el médico y sus atenciones, tanto revisar referencias directa e inversa tanto como la asignación de turnos del Hospital Base..

ecuator  
ama la vida

Ministerio de Salud Pública  
**Hospital San Vicente de Paúl**  
Coordinación Zonal de Salud # 1

m p

USUARIO : | | | SALIR

INICIO  
REFERENCIAS  
CONTRAREFERENCIAS  
ASIGNADOS

Referencias »  
Búsqueda

Cédula:    
 Apellidos y nombres:

**Figura 15.** Prototipo del estadístico del sistema

- **Invitado**

En esta ventana podremos visualizar usuario Invitado el cual tendrá exclusivamente el acceso hacia el historial de turnos asignados a este paciente.



The image shows a web application interface for Hospital San Vicente de Paúl. At the top, there is a header with the Ecuador logo on the left, the text 'ecuador ama la vida' in the center, and the 'mSP' logo on the right. Below the header, the text 'Ministerio de Salud Pública' and 'Hospital San Vicente de Paúl' is displayed, along with 'Coordinación Zonal de Salud # 1'. The main content area features a navigation menu with 'INICIO' and 'Consultas » Turnos'. Below the menu, there is a form with a label 'Ingrese No. cédula:', an input field, and a 'Consultar' button. A 'SALIR' button is located in the top right corner of the interface.

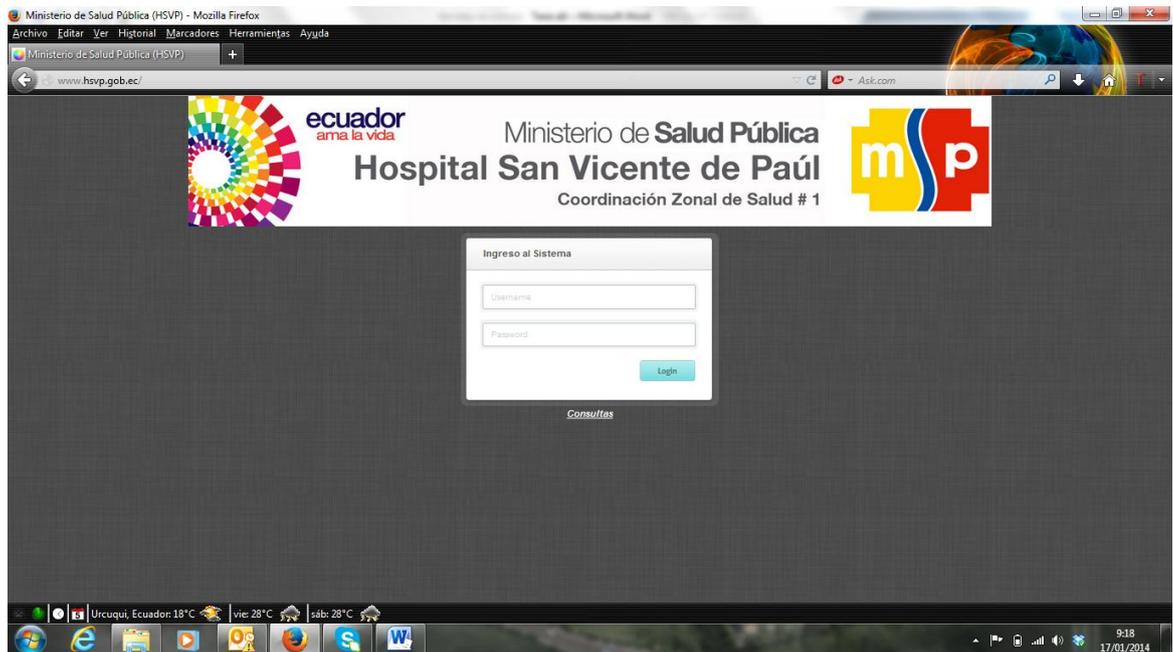
**Figura 16.** Prototipo del invitado del sistema

### 3.6 Desarrollo

#### 3.6.1 Documento del diseño final del sistema

A continuación se realizará la captura de pantallas del sistema ya desarrollado en el cual permite el ingreso solicita usuario y password y para los invitados únicamente número de cédula, estos podrán ver un

historial de los turnos asignados por el personal de estadística del Hospital San Vicente de Paúl.



**Figura 17.** Ingreso al sistema

### 3.6.2 Descripción detallada de la lógica de cada usuario

A continuación detallaremos cada una de las pantallas de los usuarios que interactúan con el sistema.

#### 2.5.2.1. Lógica del rol Administrador

El Administrador previo registro en el sistema tiene el acceso a un menú estrictamente diseñado para crear:

- Usuarios
- Pacientes
- Unidades operativas
- Patologías
- Seguros
- Reportes

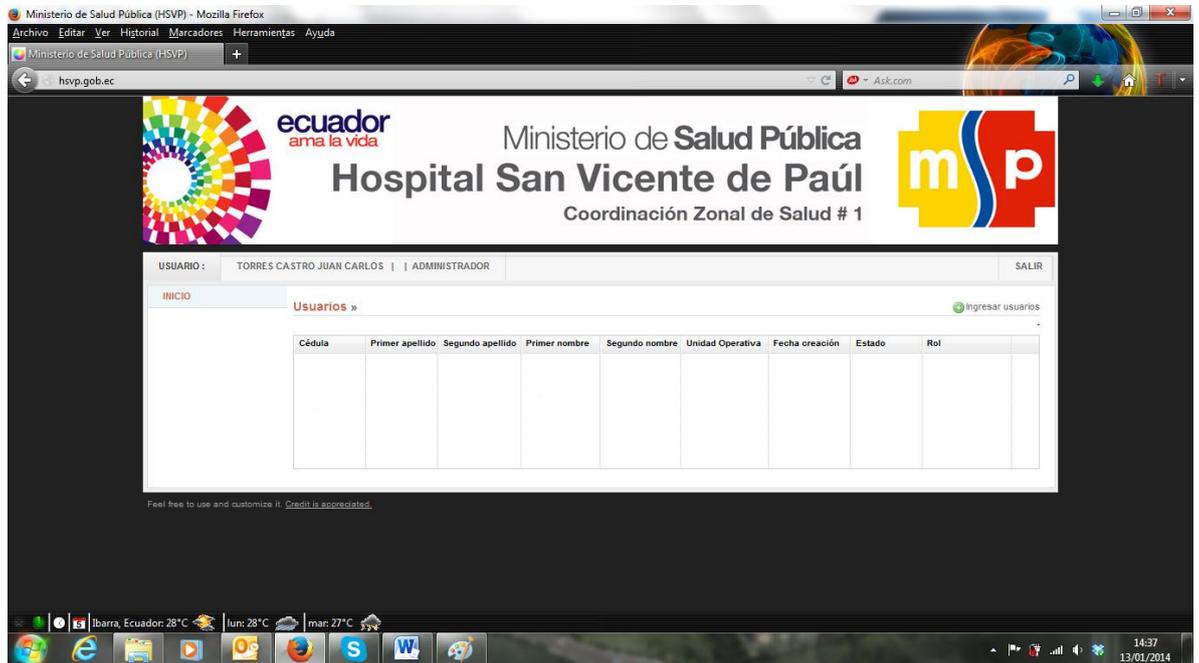


**Figura 18.-** Administrador del sistema

A continuación se detalla cada uno de los menús los cuales despliegan los registros de los usuarios.

- **Usuarios**

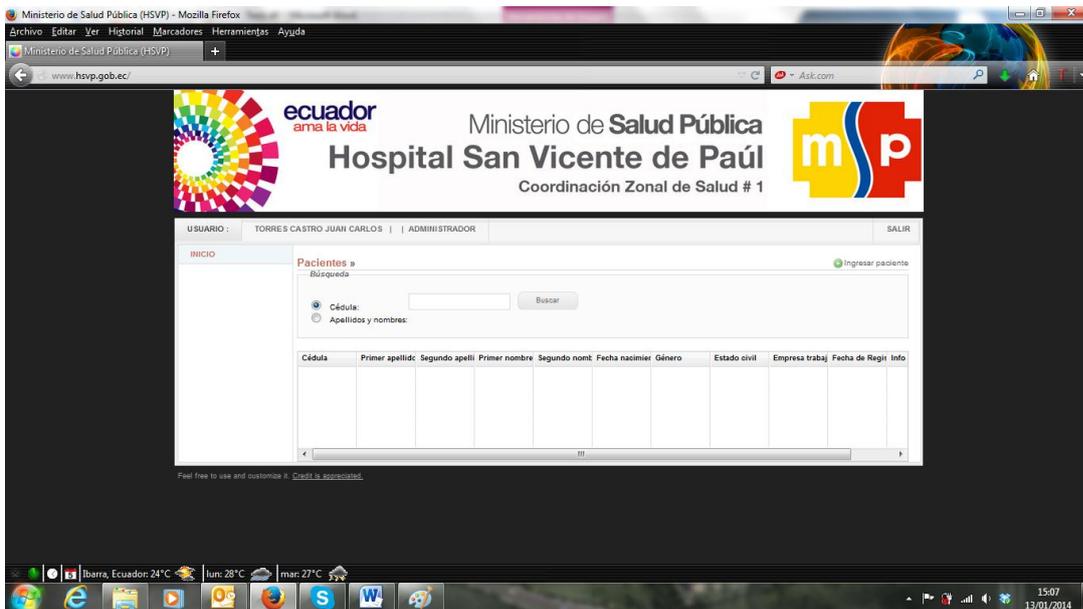
En la siguiente captura podemos observar la lista de usuarios registrados los cuales tendrán acceso al sistema, los campos necesarios para el respectivo ingreso.



**Figura 19.-** Usuarios del sistema

- **Pacientes**

En la siguiente captura podemos observar los pacientes registrados, los cuales se ingresarán desde su respectiva unidad operativa para a continuación ser atendidos en el hospital San Vicente de Paúl.



**Figura 20.** Pacientes

- **Unidades operativas**

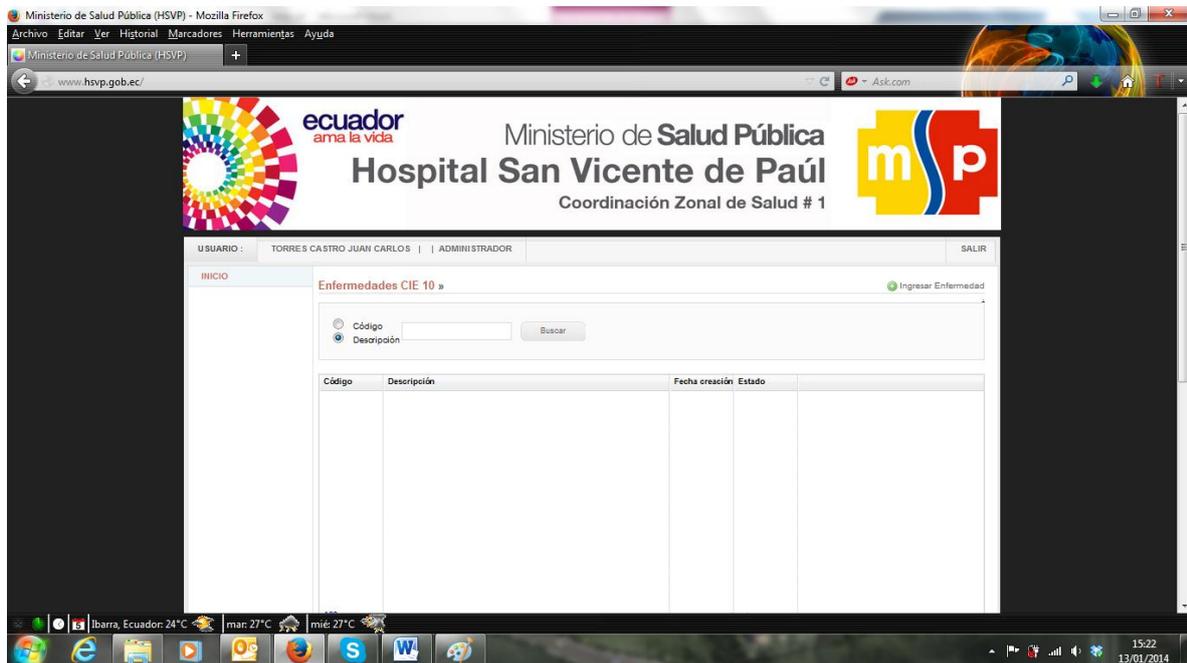
En la siguiente captura podemos observar las unidades operativas de la Zona 1 en los cuales el Hospital San Vicente de Paúl tiene influencia para así realizar un registro de los médicos.



**Figura 21.** Unidades Operativas

- **Patologías**

En la siguiente captura de pantalla podemos observar las diferentes patologías las cuales según el CIE10 tienen un registro dentro de la OMS.



**Figura 22.** Patologías ingresadas

- **Seguros**

En la siguiente captura de pantalla podemos observar las diferentes aseguradoras tanto públicas como privadas que trabajan con los pacientes ingresados.



**Figura 23.- Seguros**

- **Servicios**

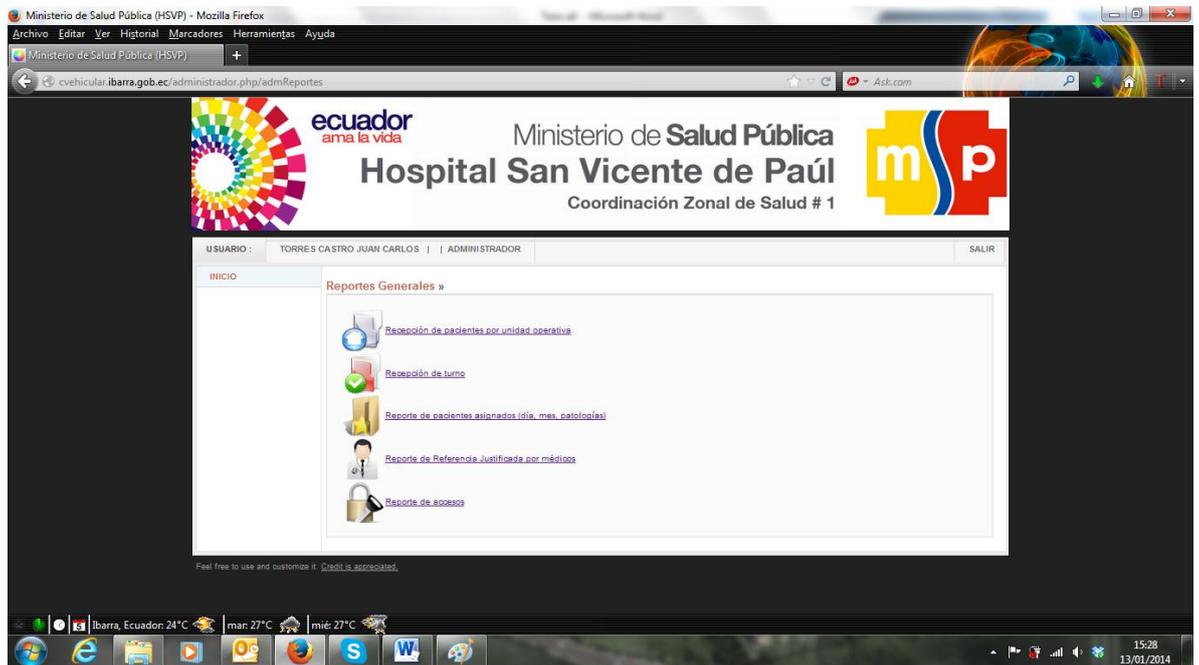
En la siguiente captura de pantalla podemos observar los diferentes servicios con los cuales el Hospital sirve a los pacientes.



**Figura 24.** Servicios

- **Reportes Generales**

En la siguiente captura de pantalla podemos observar los reportes que realiza el sistema para entrega a las autoridades.



**Figura 25.** Reportes generales

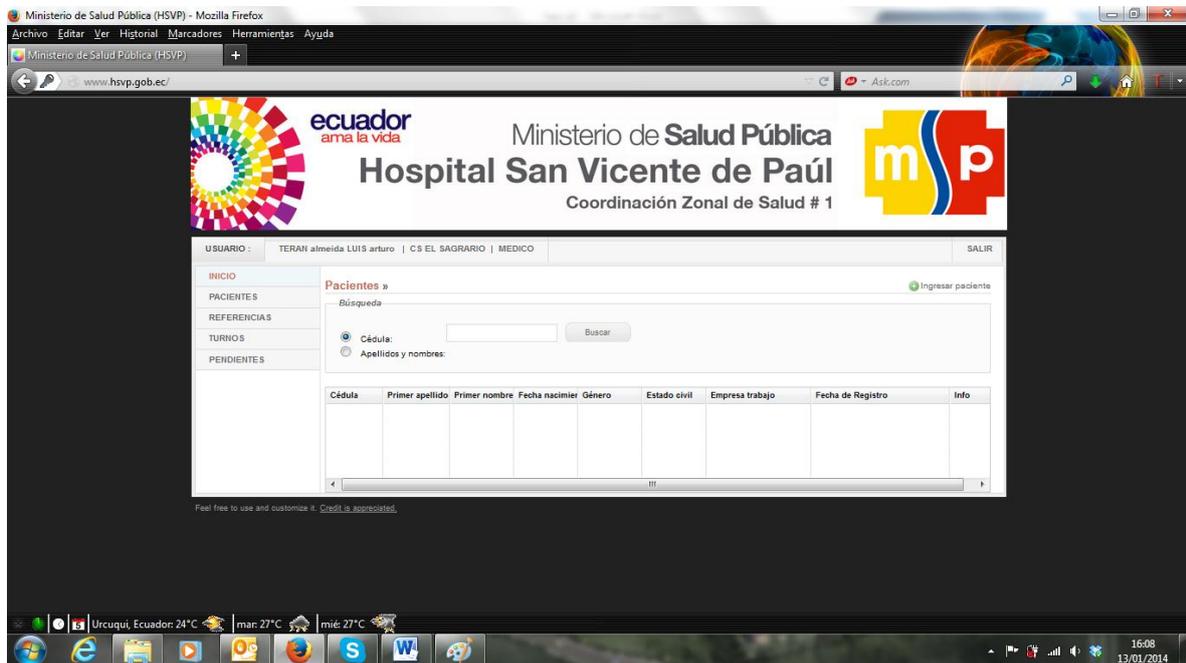
## 2.5.2.2. Lógica del rol Médico

El Médico previo registro en el sistema tiene el acceso a un menú estrictamente diseñado para leer los datos:

- **Pacientes**

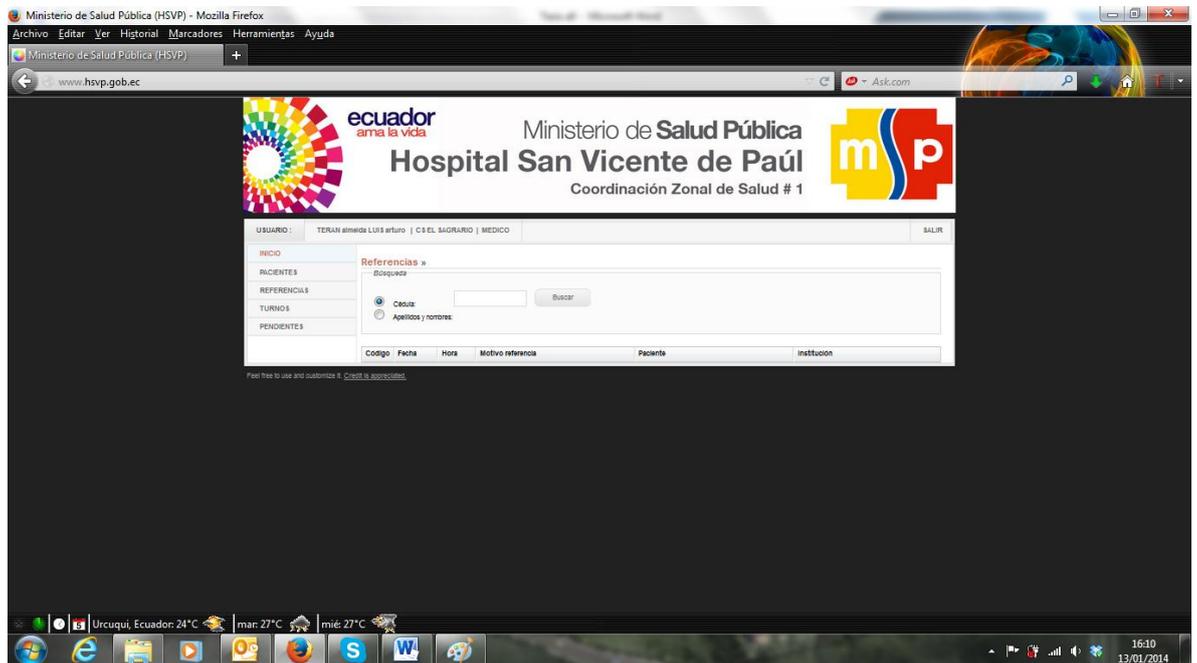
Los cuales se encuentran ingresados en el sistema, los mismo son ingresadas desde las unidades operativas.

# Referencia y Contrareferencia



**Figura 26.** Pacientes

- **Rerencias**  
Referencias las mismas que han sido atendidas por el médico.



**Figura 27.** Referencias

- **Turnos**

Los turnos serán asignados por el Estadístico mediante las referencias desde las unidades operativas hacia el médico del Hospital San Vicente.

# Referencia y Contrareferencia



**Figura 28.** Turnos

- **Pendientes**

Los pacientes pendientes son aquellos que deben seguir un tratamiento continuo dentro del Hospital San Vicente, estos pacientes no regresan a su centro de salud hasta que el médico especialista los de un alta ambulatoria.

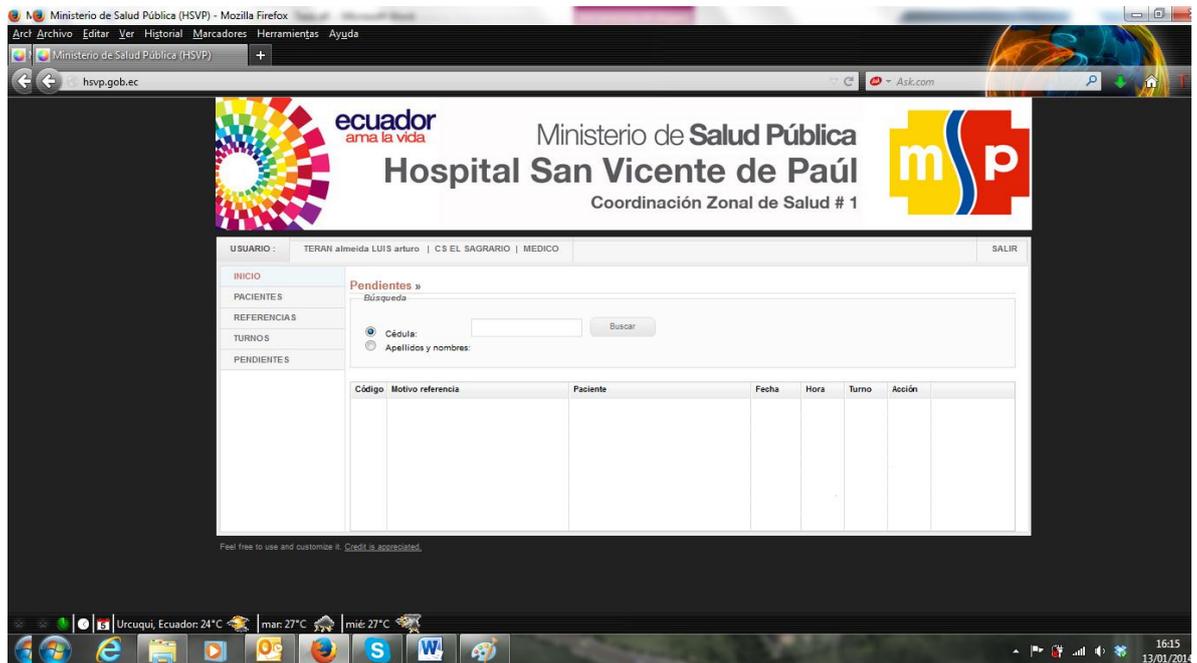


Figura 29. Pendientes

### 2.5.2.3. Lógica del rol Estadístico

El Estadístico previo registro en el sistema tiene el acceso a un menú estrictamente diseñado para obtener datos y asignar turnos.

- **Referencias**

Puede observar todas las referencias que el médico a enviado al Hospital San Vicente de Paúl.

# Referencia y Contrareferencia

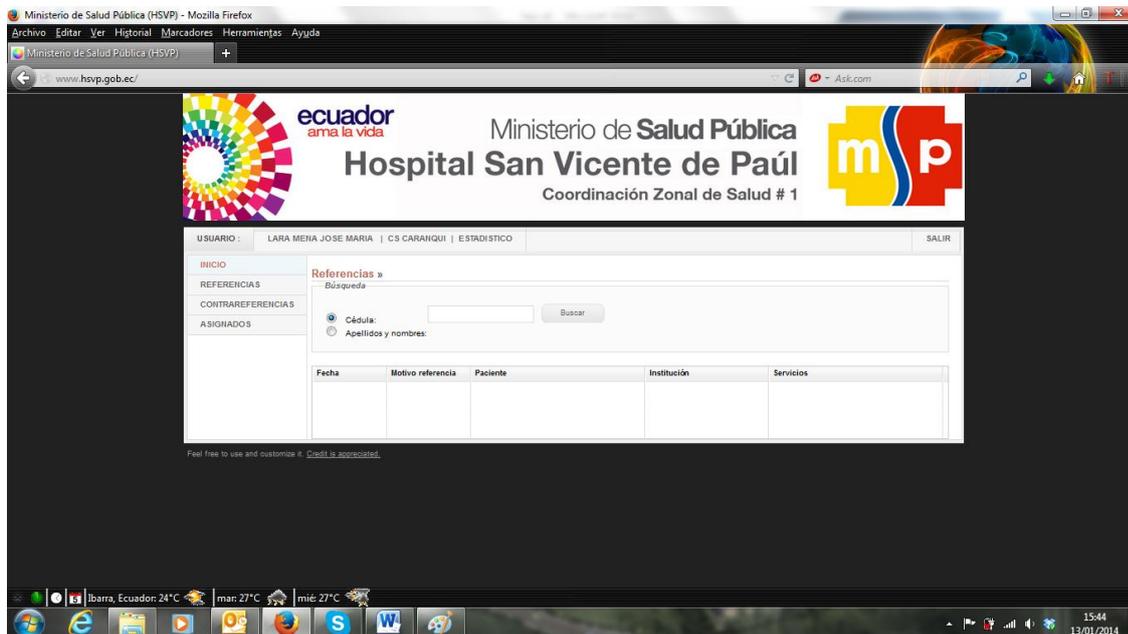


Figura 30. Referencias

- **Contrareferencias**

Puede observar todas las contrareferencias que el médico a enviado a las unidades operativas.

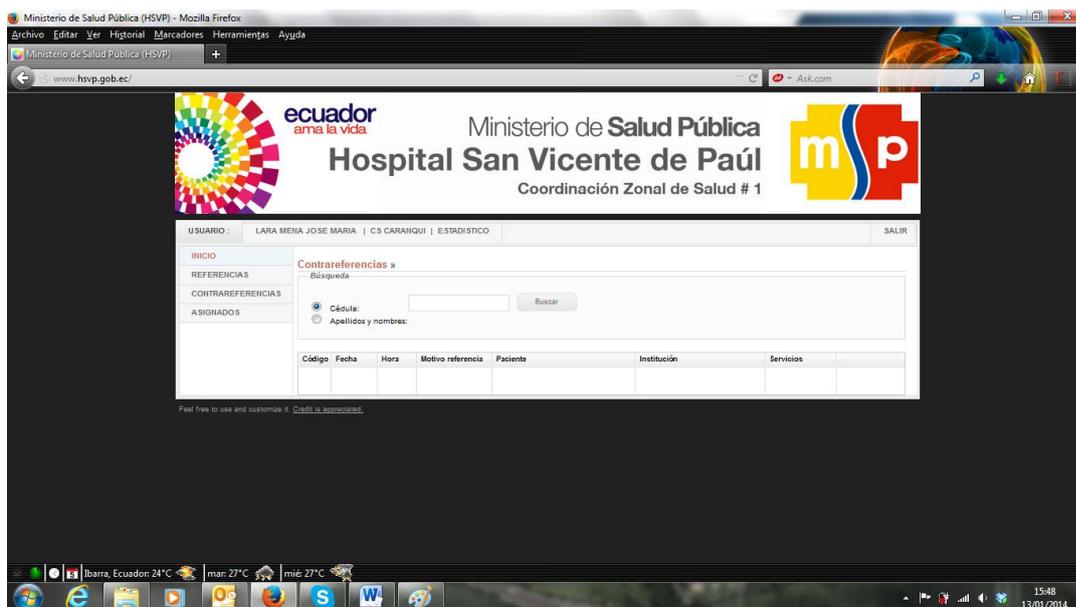
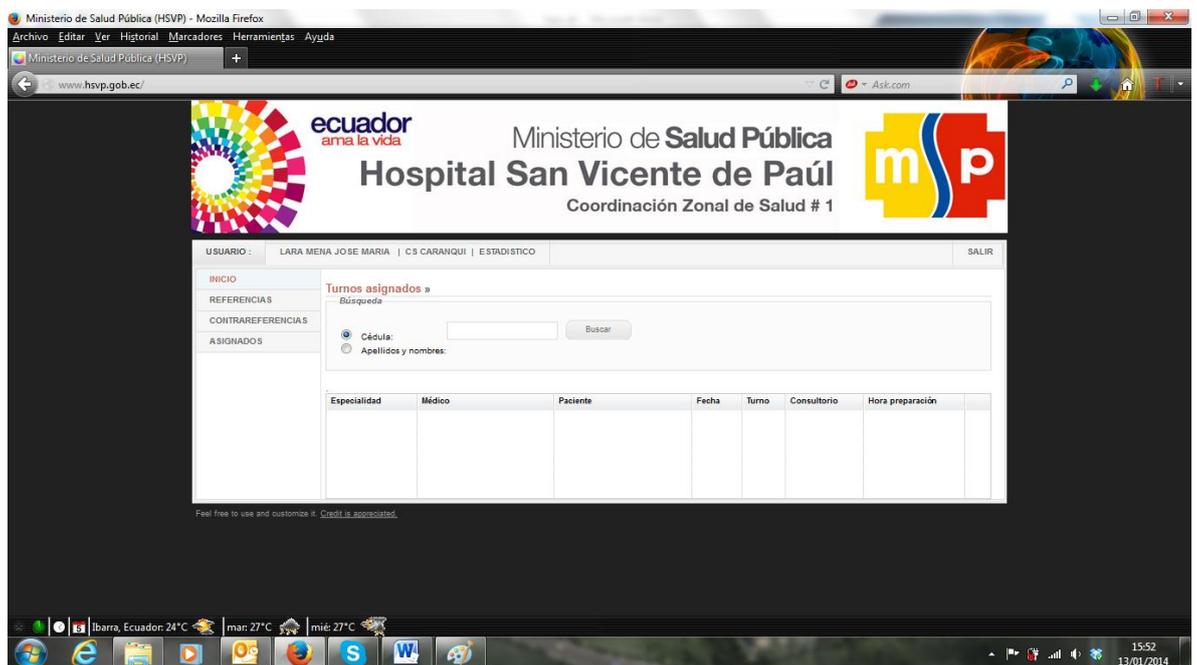


Figura 31. Contrareferencia

- **Turnos Asignados**

Puede observar los turnos que el estadístico a enviado al Hospital San Vicente.



**Figura 32.** Turnos enviados

#### 2.5.2.4. Lógica del rol Invitado

- **Turnos Asignados**

Puede observar los turnos que el paciente tiene a futuro o a su vez el historial de atenciones.

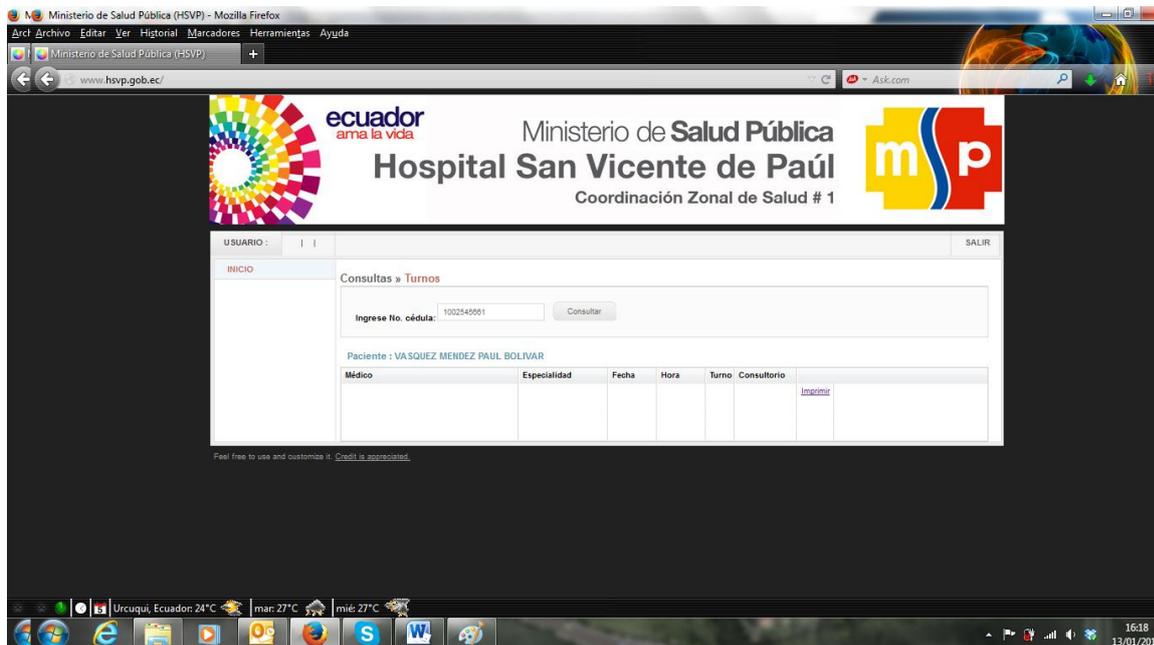


Figura 33. Invitado

## 3.7 Plan de Contingencia

El plan de contingencia o emergencia, constituye un instrumento fundamental para brindar una respuesta oportuna, adecuada y coordinada a una situación de emergencia causada ya sea por fenómenos destructivos de origen natural o humano.

### 3.7.1 Activos e Interdependencias

Este análisis demuestra que una amenaza materializada puede llegar afectar tanto al Hospital San Vicente de Paúl como a las unidades operativas de la zona 1, esto no impide a la casa de salud seguir laborando y atendiendo sus pacientes, supondría una interrupción temporal de su atención. Además afectaría negativamente a la imagen provocando malestar en los clientes.

Así se evaluaría la siguiente amenaza y su impacto:

- **Amenaza:**
  - Equipos de computación
  - Red
  - Internet
- **Impacto**
  - Pérdida de % de clientes
  - Imposibilidad de realizar ingresos
  - Imposibilidad de obtener reportes

El plan de contingencias contendría las siguientes contramedidas:

- **Medidas técnicas**
  - Contratar un proveedor alternativo
  - Respaldo de archivos y base de datos
  - Equipos de cómputo de respaldo.
- **Medidas organizativas**
  - Hosting alternativo
  - Copias de respaldos diarios.
  - Actualización de datos
- **Medidas humanas**
  - Formación al personal para actuar en caso de caídas del sistema.
  - Definir un responsable para comunicar en caso de percance.

### 3.7.2 Plan de Respaldo

- Realizar copias de respaldo
- Revisar copias de respaldo
- Pruebas de simulacro
- Custodia de backups

### 3.7.3 Plan de Emergencia

- Activar hosting alternativo
- Restaurar copias de respaldo
- Revisión de copias de respaldo
- Reanudación de la actividad.

### 3.7.4 Plan Recuperación

- Evaluación de daños
- Traslado de datos
- Reanudación de la actividad



# CAPÍTULO 4

## Conclusiones y Recomendaciones



## 4 Conclusiones y Recomendaciones

### 4.1 Conclusiones

- La Tecnología de la Información y Comunicaciones involucra a recursos los cuales permiten realizar actividades de forma confiables, garantizando una eficiencia y rapidez en su procesamiento de datos o información.
- La implementación del sistema de referencia y contrareferencia para el Hospital San Vicente de Paúl permitirá obtener la información de datos, procesamientos y almacenamiento con mayor eficiencia ya que este reducirá tiempos de espera de ejecución y una mejor atención con calidad y calidez al usuario final.
- Las aplicaciones para el desarrollo del sistema se decidieron por normas informáticas establecidas en decreto ministerial 1014 y por el Hospital San Vicente de Paúl, donde se utilizó PHP como lenguaje de programación y como motor de base de datos a PostgreSQL.
- Este sistema tendrá una ayuda importante tanto para el Hospital San Vicente de Paúl como para la Coordinación de Salud N°1 los cuales van de la mano para el atención ambulatoria de pacientes en la región 1.
- Con este proyecto los estudiantes desarrolladores y docentes de la Universidad Técnica de la Carrera de Ingeniería en Sistemas Computacionales obtendrán una fuente de consultas y guía para el desarrollo de aplicaciones con bases de datos.
- La utilización de software libre hoy en día agrupa nuevas tecnologías, facilitando el desarrollo de aplicaciones utilizando menos recursos.

- Symfony realiza una estructura estandarizando la separación de capas permitiendo crecer de forma rápida y eficiente.
- El fácil acceso a Internet en la actualidad, datos emitidos por Distrito 1 determinan que el 99% de unidades operativas tienen cobertura para el ingreso al sistema desarrollado.
- La base de datos PostgreSQL es una de las más potentes gestores de base de datos y la mejor parte herramienta libre la cual tiene compatibilidad con gran variedad de tipos de datos. A su vez soporta diferentes funcionalidades que los motores de bases de datos comerciales tienen como constraints, triggers, procedimientos almacenados.
- El uso del framework Symfony en la actualidad está en aumento pero a su vez las herramientas de consulta son mínimas.

### 4.2 Recomendaciones

- Con respecto al mantenimiento del sistema sobre el crecimiento de datos, es estrictamente necesario fomentar el crecimiento al equipos de trabajo de TIC's del Hospital San Vicente de Paúl asigne a una persona responsable de la misma.
- El poco apoyo recibido por parte del MSP, a pesar de estar desarrollado en software libre según decreto ministerial, ha sido totalmente negativo especialmente para el área tecnológica la atención es bajo en si es una institución de salud y por ende más atención se pone a los funcionarios que tienen que ver con esta área de influencia.

- Este sistema se convierte en una herramienta completamente útil y fundamental tanto para médicos como para estadísticos los cuales interactúan de forma directa con el paciente, se recomienda su utilización diaria, así como una capacitación adecuada al personal designado para entender su funcionamiento adecuado.
- Se recomienda al Líder de TIC's del Hospital San Vicente de Paúl instalar un servidor de producción para acceder directamente mediante una ip pública y un servidor de dns.
- Se recomienda para el uso del sistema el uso del navegador Mozilla Firefox cualesquier versión para de esta forma evitar incompatibilidad de templates, css y javacripts.
- El modelo MVC es completamente recomendado para el desarrollo de aplicaciones web ya que permite que el código sea organizado y de fácil mantenimiento.
- Comunicar mediante correo institucional ZIMBRA a los usuarios que van a trabajar con este sistema para su respectiva socialización en los diferentes roles.

### GLOSARIO DE TERMINOS

**ACID.-** Atomicidad, Consistencia, Aislamiento y Durabilidad.

**AJAX.-** JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas.

**BSD.-** La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution).

**CRUD.-** Create, Read, Update, Delete.

**DOM.-** Document object model.

**DQL.-** Doctrine Query Language.

**FRAMEWORK.-** Términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

**GNU.-** Ñu en inglés, es un acrónimo recursivo de “GNU No es Unix”.

**HIBERNATE.-** Es software libre, distribuido bajo los términos de la licencia GNU.

**HSVP.-** Hospital San Vicente de Paúl.

**HTML.-** Hypertext Markup Language.

**IEEE 830.-** Requisitos de software.

**IEEE1362.-** Requisitos de sistema.

**JAVASCRIPT.-** Es un lenguaje de programación interpretado, dialecto del estándar. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

**MVC.-** Modelo – Vista – Controlador.

**MVCC.-** Multiversion, concurrency, control.

**ORM.-** Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional, utilizando un motor de persistencia.

**PHP.-** es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

**SGBD.-** Es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos

**SQL.-** Structured query language.

**SYMFONY.-** Es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador.

**XML.-** siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible').

**XP.-** Xtremme Programming

### 5 Bibliografía

- Capacity. (16 de 03 de 2013). *Blog corporativo*. Obtenido de <http://blog.capacityacademy.com/2013/03/16/jquery-que-es-origenes-ventajas-desventajas/>
- Christian, C. (2010). Php Programación web avanzada. En C. Christian, *programación Web* (pág. 28). Caracas: AdventureTeam.
- Delgado, R. C. (2007). Decreto 1014. *Presidente Constitucional de la República*, (págs. 1, 2). Chile.
- Doyle, M. (2010). Framework Symfony. En M. Doyle, *Arquitectura Práctica* (pág. 261 ). La Paz: Zuñiga.
- EcuRed. (25 de 02 de 2012). *ProgramaciónX*. Obtenido de [http://www.ecured.cu/index.php/Programaci%C3%B3n\\_Extrema\\_%28XP%29](http://www.ecured.cu/index.php/Programaci%C3%B3n_Extrema_%28XP%29)
- Helma, S. (2009). *Programación de Base de Datos con MySql y PHP*. Buenos Aires: Marcombo.
- José, R. E. (2012). *Base de Datos PostgreSQL*. Mexico: TexMa.
- Julio, G. L. (2011). *Diseño y creación de Portales Web*. Bogotá: MerTec.
- LibrosWeb. (2 de 10 de 2010). *PostgreSQL-es*. Obtenido de Sobre PostgreSQL: [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)
- Maribel, S. M. (2012). Php con PostreSQL. En *Php con PostgreSQL* (págs. 17 - 20). Buenos Aires - Argentina: Pyme.
- RafaelMa. (2 de 10 de 2010). *Sobre PostgreSQL*. Obtenido de Org: [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)
- Soluciones, I. (13 de 12 de 2012). *Soluciones informáticas*. Obtenido de <http://www.solucionesim.net/blog/wp-content/uploads/2012/05/mapa-conceptual-software-libre.png>
- Spona, H. (2010). *Programación de BDDs y PHP*. Caracas: Pronet.
- Vikkam, V. (2012). fundamentos PHP. En Vaswani, *Php* (pág. 15). Asunción.

Villafuerte, V. (18 de 12 de 2009). *Extreme Programming* . Obtenido de Explained:  
<http://extremeprogramming.host56.com/PRINCIPIOS.php>

Web, L. (14 de 10 de 2007). *Symfony en pocas palabras*. Obtenido de  
[http://librosweb.es/symfony\\_1\\_1/capitulo\\_1/symfony\\_en\\_pocas\\_palabras.html](http://librosweb.es/symfony_1_1/capitulo_1/symfony_en_pocas_palabras.html)

Wikipedia. (29 de 11 de 2013). *jQuery*. Obtenido de <http://es.wikipedia.org/wiki/JQuery>

Wikipedia. (7 de 12 de 2013). *Wikipedia*. Obtenido de PostgreSQL:  
<http://es.wikipedia.org/wiki/PostgreSQL>

Wikipedia. (21 de 01 de 2014). *MVC*. Obtenido de  
[http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador)

Wikipedia. (07 de 01 de 2014). *Symfony*. Obtenido de <http://es.wikipedia.org/wiki/Symfony>

### 6 Linkografía

- Capacity. (16 de 03 de 2013). *Blog corporativo*. Obtenido de <http://blog.capacityacademy.com/2013/03/16/jquery-que-es-origenes-ventajas-desventajas/>
- Christian, C. (2010). Php Programación web avanzada. En C. Christian, *programación Web* (pág. 28). Caracas: AdventureTeam.
- Delgado, R. C. (2007). Decreto 1014. *Presidente Constitucional de la República*, (págs. 1, 2). Chile.
- Doyle, M. (2010). Framework Symfony. En M. Doyle, *Arquitectura Práctica* (pág. 261 ). La Paz: Zuñiga.
- EcuRed. (25 de 02 de 2012). *ProgramaciónX*. Obtenido de [http://www.ecured.cu/index.php/Programaci%C3%B3n\\_Extrema\\_%28XP%29](http://www.ecured.cu/index.php/Programaci%C3%B3n_Extrema_%28XP%29)
- Helma, S. (2009). *Programación de Base de Datos con MySql y PHP*. Buenos Aires: Marcombo.
- José, R. E. (2012). *Base de Datos PostgreSQL*. Mexico: TexMa.
- Julio, G. L. (2011). *Diseño y creación de Portales Web*. Bogotá: MerTec.
- LibrosWeb. (2 de 10 de 2010). *PostgreSQL-es*. Obtenido de Sobre PostgreSQL: [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)
- Maribel, S. M. (2012). Php con PostreSQL. En *Php con PostgreSQL* (págs. 17 - 20). Buenos Aires - Argentina: Pyme.
- RafaelMa. (2 de 10 de 2010). *Sobre PostgreSQL*. Obtenido de Org: [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)
- Soluciones, I. (13 de 12 de 2012). *Soluciones informáticas*. Obtenido de <http://www.solucionesim.net/blog/wp-content/uploads/2012/05/mapa-conceptual-software-libre.png>
- Spona, H. (2010). *Programación de BDDs y PHP*. Caracas: Pronet.
- Vikkam, V. (2012). fundamentos PHP. En Vaswani, *Php* (pág. 15). Asunción.

Villafuerte, V. (18 de 12 de 2009). *Extreme Programming* . Obtenido de Explained:  
<http://extremeprogramming.host56.com/PRINCIPIOS.php>

Web, L. (14 de 10 de 2007). *Symfony en pocas palabras*. Obtenido de  
[http://librosweb.es/symfony\\_1\\_1/capitulo\\_1/symfony\\_en\\_pocas\\_palabras.html](http://librosweb.es/symfony_1_1/capitulo_1/symfony_en_pocas_palabras.html)

Wikipedia. (29 de 11 de 2013). *jQuery*. Obtenido de <http://es.wikipedia.org/wiki/JQuery>

Wikipedia. (7 de 12 de 2013). *Wikipedia*. Obtenido de PostgreSQL:  
<http://es.wikipedia.org/wiki/PostgreSQL>

Wikipedia. (21 de 01 de 2014). *MVC*. Obtenido de  
[http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador)

Wikipedia. (07 de 01 de 2014). *Symfony*. Obtenido de <http://es.wikipedia.org/wiki/Symfony>

### 6.1 ANEXOS (ver cd)

- Sistema Integrado de salud (Exposición realizada por la Ministra de Salud Msc. Carina Vance en enlace sabatino #258 Modelo de gestión de salud)
- Manual Operativo red Integrada de Servicios de Salud, Febrero 2013)
- Decreto 1014 Presidencia Constitucional de la República.
- Manual de técnico.
- Manual de usuario.