

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

Proyecto Previo a la Obtención del Título de

INGENIERO EN SISTEMAS COMPUTACIONALES

PROYECTO:

“Sistema de Automatización del Tratamiento de Información de Historias Clínicas y Medicamentos en el Subcentro de Salud de San Antonio de Ibarra”

Autor: *Jorge Aníbal Ipiales Ipiales*

Director: *Ing. Mauricio Rea*

Ibarra – Ecuador

2010-2011

CERTIFICO:

Que la tesis previa a la obtención del título de Ingeniero en Sistemas Computacionales con el desarrollo del proyecto “**SISTEMA DE AUTOMATIZACIÓN DEL TRATAMIENTO DE INFORMACIÓN DE HISTORIAS CLÍNICAS Y MEDICAMENTOS EN EL SUBCENTRO DE SALUD DE SAN ANTONIO DE IBARRA**” ha sido desarrollado y terminado en su totalidad por el **Sr. Jorge Aníbal Ipiales Ipiales** con **C.C. 100194086-3** bajo mi supervisión para lo cual firmo en constancia.

Atentamente,

Ing. Mauricio Rea
DIRECTOR DE TESIS

DEDICATORIA

Este trabajo, me gustaría dedicar:

A DIOS, por darme vida y salud y con esto lograr terminar una de mis grandes metas en mi vida.

A la MEMORIA DE MI PADRE, Marceliano Ipiales y a MI MADRE, Rosario Ipiales, por darme la vida, su sacrificio e inculcarme buenos principios, los cuales me sirvieron para llegar a esta instancia de mi vida.

Pero principalmente, a MI ESPOSA, Tatiana Yalamá, por el amor y paciencia brindados durante todo el desarrollo de este trabajo, a mis hijos Alan y Mateo, porque son mi inspiración de vida y mi fuerza para culminar con éxito todo proyecto emprendido.

AGRADECIMIENTO

Tengo que expresar mi agradecimiento a muchas personas, que de una u otra manera influyeron para que este trabajo pueda realizarse:

Agradezco a la Universidad, por haberme dado la oportunidad de estudiar tan prestigiosa carrera.

De manera especial, agradezco al Ing. Mauricio Rea, mi Director de Tesis, el mismo que aportó con su tiempo, experiencia y dedicación para dar sus opiniones y correcciones tanto al documento como al desarrollo del Proyecto.

Al Subcentro de Salud "San Antonio", especialmente a la Lcda. Maritza Álvarez, Directora de esta casa de salud, por darme la apertura y facilidades para terminar este proyecto en esta Institución.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1	2
INTRODUCCIÓN-SUBCENTRO DE SALUD SAN ANTONIO	2
1.1. ANTECEDENTES DEL SUBCENTRO DE SALUD DE SAN ANTONIO DE IBARRA.	3
1.2. LOCALIZACIÓN DEL SUBCENTRO DE SALUD SAN ANTONIO.....	4
1.3. PERSONAL DE LA INSTITUCIÓN.....	5
1.4. CARACTERÍSTICAS DEL ESTABLECIMIENTO.....	6
1.5. SERVICIOS QUE BRINDAN EN LA INSTITUCIÓN.	7
1.6. MISIÓN.....	8
1.7. VISIÓN	8
1.8. VALORES Y SÍMBOLOS DE LA INSTITUCIÓN	8
CAPÍTULO 2	10
MARCO TEÓRICO	10
2.1. PLATAFORMA WEB.	11
2.1.1. Servicios del Portal Web.	11
2.1.2. Modalidades de Portales.....	13
2.2. ANÁLISIS DE HERRAMIENTAS LIBRES PARA EL DISEÑO WEB.	13
2.2.1. Elección de Herramientas	13
2.2.2. Herramientas de Software Libre.....	17
2.2.3. Software Propietario y Software Libre.	18
2.2.4. Criterios en la selección de Herramientas de Software Libre.	19
2.2.5. Servidor Web.	21
2.2.6. Base de Datos.....	24
2.2.7. Lenguaje de Programación	25
2.2.7. Framework.	27
2.2.8. Entornos de Desarrollo (IDE) para PHP.....	30
2.3. EL SOFTWARE LIBRE Y DESARROLLADORES.	30
2.3.1. Definición Aceptada de Software Libre	30
2.3.2. Los Desarrolladores	31
2.3.3. Licencias.	32
2.4. DIFERENCIAS ENTRE SOFTWARE LIBRE, CÓDIGO ABIERTO Y FREWARE	35
2.4.1. Software Libre (Free Software)	35
2.4.2. Código Abierto (Open Source)	36
2.4.3. Formas de Distribución, Freeware y Shareware	37
2.5. EL SOFTWARE LIBRE EN ECUADOR	39
2.5.1. Antecedentes	39
2.5.2. El Decreto 1014.....	40

2.5.3. Gratas Consecuencias al Usar Software Libre.....	41
2.6. SOFTWARE MÉDICO EN EL MERCADO	44
2.6.1. Introducción.....	44
2.6.2. Empresas Desarrolladoras.....	46
2.6.3. Industria de Software en Ecuador	46
2.6.4. La Importancia del sector del software en Ecuador.	47
2.6.5. Perfil del Cliente.....	48
2.7. ANÁLISIS DE METODOLOGÍAS PARA EL DESARROLLO DE SOFTWARE.....	49
CAPÍTULO 3.....	54
DISEÑO DE LA SOLUCIÓN	54
3.1. INICIO	55
3.1.1. Modelado Actual de Actividades.	55
3.1.2. Análisis de Requerimientos.....	63
3.2. ELABORACIÓN DE LA SOLUCIÓN.....	67
3.2.1.- Flujo de Trabajo de Requerimientos.	67
3.2.2.- Análisis del Modelo óptimo de Trabajo.	68
3.2.3. Diseño	71
3.3. ALCANCE DEL SISTEMA.	99
CAPÍTULO 4.....	102
CONSTRUCCIÓN DEL SISTEMA.....	102
4.1. ANÁLISIS Y RECOLECCIÓN DE INFORMACIÓN.....	103
4.1.1. Análisis de Usuarios del Sistema	103
4.1.2. Construcción del Producto (Casos de Uso).	105
4.1.3. Redefinición de Análisis y Diseño.	128
4.1.4. Ficha de Historias Clínicas.....	131
4.1.5. Ficha de Información de medicamentos.....	134
4.1.6. Ficha Datos Personales de Médicos	137
4.1.7. Ficha Datos Básicos de Turnos	138
4.2. DISEÑO	139
4.2.1. Delimitación del Sistema	139
4.2.2. Diseño de Tablas de la Base de Datos	141
4.3. ARQUITECTURA Y CÓDIFICACIÓN DEL SISTEMA.....	150
4.3.1. Análisis de Arquitectura del sistema.....	150
4.3.2. El Layout.	152
4.3.3. Las hojas de estilo, imágenes y archivos JavaScript	154
4.3.4. Las Acciones	158
4.3.5. Las Plantillas	160
4.3.6. Los Plugins de Symfony	162
4.3.6. Arquitectura por defecto vs. Arquitectura de los Plugins	163

4.4. IMPLEMENTACIÓN	164
4.4.1. Análisis del Producto Terminado.....	164
4.4.2. Correcciones y Mejoras.....	166
4.5. INSTALACIÓN Y CAPACITACIÓN.....	168
4.5.1. Instalación	168
4.5.2. Capacitación	169
CAPÍTULO 5.....	171
CONCLUSIONES Y RECOMENDACIONES	171
5.1. CONCLUSIONES	172
5.2. RECOMENDACIONES.....	173
BIBLIOGRAFÍA	175

INTRODUCCIÓN

San Antonio de Ibarra es una Parroquia rural, que forma parte de la provincia de Imbabura, la misma que en los últimos años ha crecido poblacionalmente y por ende sus requerimientos de salud.

Razones suficientes para que el Subcentro de Salud de San Antonio, incremente y diversifique servicios y exigiendo además a los profesionales de la salud mayor preparación y actualización en las diferentes ramas, pues la salud y el bienestar humanos son la base y la razón de ser de todos los esfuerzos del desarrollo.

En la actualidad es primordial que se optimicen los recursos en todas las entidades sean públicas o privadas incrementando como herramienta a un sistema informático, pues la información es la pieza clave para que una institución desempeñe su trabajo en una forma eficiente.

A pesar del avance en el área de la Informática, donde la mayoría de Instituciones en nuestro país tienen sistemas de almacenamiento de información, se ha observado que en esta institución no poseen un historial magnético, por no utilizar un programa informático que les ayude a llevar los datos organizados y seguros.

Cabe destacar que en la actualidad la Informática en general es una de las herramientas más trascendentales para el desarrollo institucional de las organizaciones, por este motivo que se necesita explotar la tecnología para tener una mejor perspectiva empresarial y mejorar sus recursos: financieros, profesionales y materiales.

Por no contar con una herramienta informática que agilite el quehacer diario de la institución, el Subcentro ha estado brindando un servicio que no cumple con las perspectivas del paciente, por lo que se hace necesario implementar esta herramienta.

En el presente documento expondré una solución informática para tratar de resolver la problemática de atención al paciente.

CAPÍTULO 1

INTRODUCCIÓN-SUBCENTRO DE SALUD SAN ANTONIO



- 1.1. Antecedentes del Subcentro de Salud de San Antonio de Ibarra.
- 1.2. Localización del Subcentro de Salud.
- 1.3. Personal de la Institución.
- 1.4. Características del Establecimiento.
- 1.5. Servicios que Brindan en la Institución.
- 1.6. Misión.
- 1.7. Visión.
- 1.8. Valores y Símbolos de la Institución.

1.1. ANTECEDENTES DEL SUBCENTRO DE SALUD DE SAN ANTONIO DE IBARRA.

San Antonio de Ibarra, es una parroquia férreamente unida y organizada que garantiza la equidad en el marco de su desarrollo integral y competitivo, con una población capacitada y potenciada en sus actividades artísticas artesanales, agrícolas y empresariales a través del comercio asociativo ligado al turismo nacional e internacional y respaldos en el manejo sustentables de los recursos naturales que proyecta el bienestar colectivo a los presentes y futuras generaciones, desarrollando circuitos eco-turísticos que potencialicen nuestro entorno urbano.

“La parroquia de San Antonio toma la categoría de parroquia rural del cantón Ibarra el 29 de mayo 1861, en pleno, la junta parroquial resuelve institucionalizar, como fecha cívica parroquial”¹.

Las artesanías y la escultura forma parte de la cultura y tradición de San Antonio de Ibarra, la actividad artesanal de San Antonio comenzó en 1880 con la creación del Liceo Artístico, una especie de escuela- taller que impartía conocimientos y práctica en pintura, escultura, tallado, y carpintería. En la década de 1950 aparecen figuras costumbristas diversificándose la producción para consumo de los visitantes.

“De acuerdo al sexto censo de Población de Noviembre del 2.001, la parroquia de San Antonio cuenta con: 13.987 Habitantes”².

La propuesta de servicios de la salud del Subcentro de San Antonio pretende asegurar el mejoramiento continuo de la gestión de autoridades, equipo de profesionales, usuarios y comunidad en general para satisfacer la demanda y expectativas de los pacientes que demandan de servicios de salud en el mismo, mediante el diseño de gestión administrativa.

Tomando en consideración que el principal prestador de servicios de salud es el MSP , El Subcentro de Salud San Antonio fue creado en el año de 1976, en sus inicios funcionaba en el CENTRO CÍVICO de san Antonio de Ibarra en un lugar muy limitado, después de

¹ MONOGRAFIA DE SAN ANTONIO DE IBARRA, 2 ed. Quito, Junio del 2007, pag. 39.

² MONOGRAFIA DE SAN ANTONIO DE IBARRA, 2 ed. Quito, Junio del 2007, pag. 39., por la Sra. Susana Dávila de Padilla.

funcionar 5 años aproximadamente en este lugar donde era poco funcional con una mala distribución espacial de las áreas de trabajo, sin embargo en la actualidad cuenta con su propio edificio, de infraestructura mixta en buen estado de conservación y además , el edificio cuenta con un espacio físico suficiente para realizar ampliaciones.

1.2. LOCALIZACIÓN DEL SUBCENTRO DE SALUD SAN ANTONIO.

San Antonio de Ibarra parroquia rural perteneciente al cantón Ibarra de la provincia de Imbabura, ubicada a 5.50 Km. De la capital provincial y a 174 Km. de la capital ecuatoriana.

“Cuenta con una superficie de 29,07 Km² (Veinte y nueve con siete kilómetros cuadrados), de acuerdo a su topografía, se encuentra entre los 2,040 msnm y 4,620 msnm; por su situación Geográfica, San Antonio se privilegia con un clima templado en forma general”³.

Actualmente el edificio en el que funciona el Subcentro se encuentra en las calles Bolívar y Francisco Terán, área de gran circulación vehicular y peatonal, la localización en forma gráfica lo mostramos en la **Gráfico 1.1**.

³ MONOGRAFIA DE SAN ANTONIO DE IBARRA, 2 ed. Quito, Junio del 2007, pag. 40.

al centro de salud Nro. 1, aunque el trabajo con esta unidad se la basa principalmente en la utilización de servicios complementarios como servicio de laboratorio.

El Subcentro de salud San Antonio es una unidad operativa de segundo nivel con una capacidad resolutive media, se encuentra interrelacionado con varios servicios y entes de salud tanto de mayor como menor complejidad.

“Dentro del equipo de trabajo del Subcentro de Salud se cuenta con:

- 4 Médicos Generales
- 1 Medico Rural
- 1 Obstetra
- 1 Odontólogo
- 2 Internos de Medicina
- 2 Licenciadas en Enfermería
- 1 Licenciadas Enfermeras rurales
- 2 Internas de enfermería
- 1 Auxiliar de Odontología
- 1 Auxiliar de enfermería
- 1 Inspector sanitario”⁴

1.4. CARACTERÍSTICAS DEL ESTABLECIMIENTO.

El Subcentro de Salud San Antonio, actualmente tiene local propio con una infraestructura en buen estado y continúa ampliándose para poder brindar un mejor servicio a la comunidad.

El edificio en el que funciona el Subcentro se encuentra en las calles Bolívar y Francisco Terán, área de gran circulación vehicular y peatonal, el espacio físico se distribuyen las siguientes áreas para la atención de los servicios de salud.

- 4 consultorios

⁴ DIAGNÓSTICO SITUACIONAL DE LA PARROQUIA DE SAN ANTONIO, Investigación realizada por: Dra. Mercy Vásquez, Lic. María Fernanda Valle, Lic. Anita Pulla, Lic Lorena Chamorro, período 2009-2010, pag. 8.

- 1 farmacia y post consulta
- Estadística
- Preparación
- Vacunas
- Curaciones
- Odontología
- Sala de espera
- Espacio para sanidad y laboratorio

Las áreas de atención el Subcentro no tiene muy bien delimitadas, no hay total privacidad para la consulta y demás dependencias, el Subcentro ha visto la necesidad de ampliar sus áreas mediante la construcción de una sección, pero la asignación de presupuestos del Gobierno Central por medio del Ministerio de Salud no ha dispuesto un presupuesto con esta finalidad.

1.5. SERVICIOS QUE BRINDAN EN LA INSTITUCIÓN.

La Unidad Operativa cumple su misión a través de los siguientes servicios.

- Atención Medicina General
- Atención Odontológica
- Atención Gineco Obstetricia
- Curaciones e inyecciones
- Promoción de la salud y prevención de enfermedades con la aplicación de todos los programas vigentes en el MSP.

El área de influencia del Subcentro tiene una cobertura geográfica circundante de 2Km incluyéndose en ella 26 comunidades correspondientes a la parroquia.

La unidad operativa se encuentra vinculada con la mayoría de los entes en salud, cuenta con la colaboración de voluntarios de salud, distribuidos en las diferentes comunidades de la parroquia, especialmente en aquella que geográficamente se encuentra más distante.

Se determina que existe una tasa de utilización de un 13% de la unidad operativa según

los registros estadísticos del Subcentro, es decir que existe una población muy significativa que demanda los servicios de esta institución; sin embargo no se cubren las necesidades; los niños, embarazadas y ancianos son los grupos que en forma mayoritaria requieren atención en este centro de atención, por lo que diariamente las consultas otorgan turnos según la capacidad disponible por lo que muchos usuarios no son atendidos, quienes a su vez deben buscar otro tipo de soluciones de acuerdo a la emergencia o requerimiento del caso de salud que adolecen.

1.6. MISIÓN

“El Subcentro de Salud de San Antonio asume la misión de ofertar y brindar servicios en forma, eficiente, eficaz, efectiva y participativa orientadas al mejoramiento de la calidad de vida fundamentando su acción para:

- Brindar servicios en salud humanitarios, de alta calidad y accesibles.
- Comprometer la participación ciudadana en el mejoramiento de la calidad de vida en salud.
- Promover la organización interinstitucional en la comunidad para mejorar la calidad de vida”⁵.

1.7. VISIÓN

“La proyección es llegar a ser una Institución de Excelencia, altamente competitiva, con procesos productivos de calidad, con una organización y gestión administrativa eficiente, flexible y transparente, para dar respuesta de forma oportuna y efectiva a las demandas y/o requerimientos de los usuarios y usuarias de los servicios de salud”⁶.

1.8. VALORES Y SÍMBOLOS DE LA INSTITUCIÓN

El Subcentro de Salud de San Antonio está comprometido con brindar servicios de salud

⁵ ALVAREZ, Maritza & INGA, Fredisulinda: Participación Comunitaria para el Mejoramiento de la Calidad de Atención en el Subcentro de Salud de San Antonio. Ibarra, 2006, Trabajo de Grado (Maestría en Gerencia Integral de Salud para el Desarrollo Local), Universidad Técnica Particular de Loja, Escuela de Medicina, Pags. 42-43.

⁶ ALVAREZ, Maritza & INGA, Fredisulinda: Participación Comunitaria para el Mejoramiento de la Calidad de Atención en el Subcentro de Salud de San Antonio. Ibarra, 2006, Trabajo de Grado (Maestría en Gerencia Integral de Salud para el Desarrollo Local), Universidad Técnica Particular de Loja, Escuela de Medicina, Pag. 43.

accesibles y de alta calidad a la comunidad que sirve. Por sobre todo se valora:

- Dignidad - Respetar el valor y la dignidad inherente de cada persona.
- Colaboración - Trabajar en conjunto con las personas que poseen la misma visión y valores para lograr la metas compartidas.
- Justicia - Abogar por un cambio social y actuar de manera que se pueda promover el respeto para todas las personas.
- Excelencia - Exceder las expectativas a través del trabajo en equipo y el perfeccionamiento de los servicios.

La Institución no cuenta con símbolos propios, pues solo cuenta con la insignia del Ministerio de Salud Pública, aumentado el nombre "Subcentro de Salud San Antonio" y su dirección, como se presenta en el **Gráfico 1.2.**



Gráfico 1.2. Símbolo del Subcentro de Salud San Antonio.

Fuente: Subcentro de Salud San Antonio

CAPÍTULO 2

MARCO TEÓRICO



- 2.1. **Plataforma Web**
- 2.2. **Análisis de Herramientas Libres para diseño Web**
- 2.3. **El Software Libre y Desarrolladores.**
- 2.4. **Diferencias entre Software Libre, Código Abierto Y Freeware**
- 2.5. **El Software Libre en Ecuador.**
- 2.6. **Software Médicos en el Mercado.**
- 2.7. **Análisis de Metodologías para el Desarrollo de Software.**

2.1. PLATAFORMA WEB.

Un portal de Internet es un sitio web cuyo objetivo es ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros, documentos, aplicaciones, compra electrónica, etc. Principalmente están dirigidos a resolver necesidades específicas de un grupo de personas o de acceso a la información y servicios de una institución pública o privada [WWW.002].

En Base a este concepto, se deduce que un Portal web puede personalizarse para ayudar a solucionar una serie de necesidades, entre ellas la de gestionar información de alguna institución médica. El término **portal** tiene como significado puerta grande, es considerado como un intermediario de información y precisamente su nombre hace referencia a su función u objetivo, por lo que además se puede decir que es el punto de partida de un usuario que desea buscar y generar información.

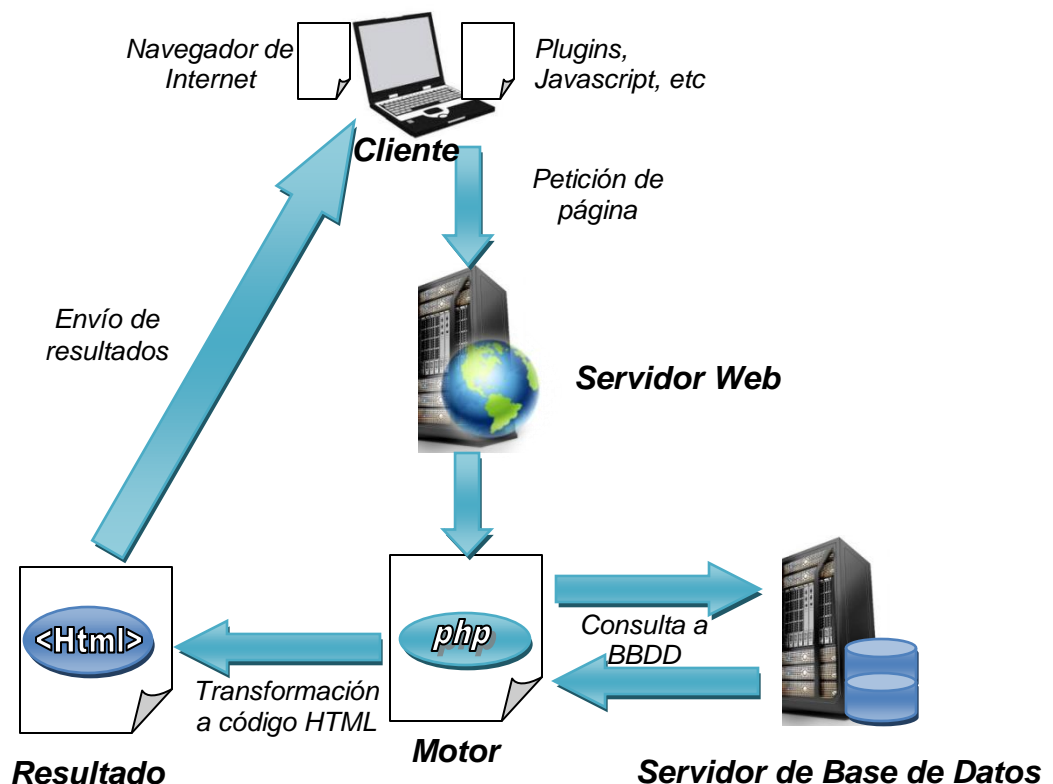


Gráfico 2.1: Arquitectura Web

Fuente: Diseño Propio

2.1.1. Servicios del Portal Web.

Un portal web ofrece además, otros servicios adicionales, aquí podemos citar algunos, los más comunes que podemos encontrar:

- **Los Servicios de búsqueda**, que incluye mecanismos de búsqueda, directorios y páginas amarillas para localizar negocios o servicios en la web.
- **Contenidos**, es decir, información de varios tópicos como noticias, deportes, pronósticos de clima, listas de eventos locales, mapas, opciones de entretenimiento, juegos, enlaces a estaciones de radio y a otros sitios con contenido especial en ciertas áreas de interés como coches, viajes y salud entre otros.
- **Facilidades para construir comunidades virtuales** tales como chat, servicios de mensajería, aplicaciones para intercambio de fotografías, anuncios clasificados, posibilidad de construir páginas de web personales.
- **Facilidades de comercialización**: anuncios clasificados para trabajos, carros y casas; subastas, pequeños agregados de vendedores y ligas a otros sitios que también se dedican a la venta.
- **Aplicaciones de productividad personal** tales como e-mail, calendarios, directorios personales, almacenamiento de archivos y servicio de pagos de cuentas.

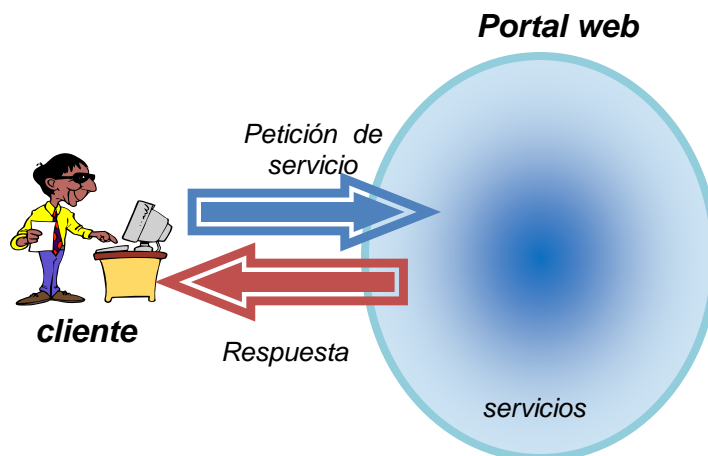


Gráfico 2.2: Petición de servicios Portal Web.

Fuente: Diseño Propio

2.1.2. Modalidades de Portales.

Existen dos modalidades de portales:

1. **Portales horizontales**, estos portales son llamados masivos o de propósito general, pues va dirigido a una gran audiencia, y trata de llegar a toda la gente con muchas cosas, muchos tópicos, con una gran gama de entretenimiento y curiosidades. Como ejemplo de portales de esta categoría están AOL, AltaVista, Lycos, Yahoo, MSN, entre otros.
2. **Portales verticales**, se dirigen a usuarios para ofrecer contenido y comercio dentro de un tema específico como puede ser un portal de música, un portal de finanzas personales o de deportes, es decir solo se dedica a un tema específico, en este tipo de portales puede estar inmerso un portal de comercio electrónico, pues solo se dedica a realizar comercio, aunque los tipos de productos que ofrecen pueden ser varios.

Resumiendo, con una WEB, Internet se convierte en una fuente de clientes e ingresos esencial para todas aquellas agencias que desean ofrecer su saber hacer, su profesionalidad y su experiencia a todos los clientes.

Un aspecto importante que hay que tener en cuenta es que los portales normalmente tienen programación que requiere muchos recursos computacionales y por su alto tráfico generalmente se necesita hospedar en servidores de internet.

2.2. ANÁLISIS DE HERRAMIENTAS LIBRES PARA EL DISEÑO WEB.

2.2.1. Elección de Herramientas

Al momento de plantearse el montar un portal web de Gestión de Información, uno de los factores fundamentales a tener en cuenta son los recursos financieros disponibles para hacer frente al proyecto. En el caso de que dichos recursos sean escasos (o nulos), la opción respecto al software está clara: hay que utilizar software libre.

Pero en el campo de software libre también tenemos una infinidad de herramientas con las que podemos dedicarnos a realizar nuestro proyecto web, en los siguientes puntos veremos las ventajas o desventajas si es que lo tuviesen al elegir tal o cual herramienta.

Apache, por ser un servidor web más conocido y confiable en el mundo, **PHP** uno de los lenguajes de programaciones web más flexibles y poderosos, y **MySQL** está entre las bases de datos más populares. Los tres son opensource y gratuitos. Forman, al día de hoy, una combinación clásica para desarrolladores web y servidores.

Si se busca esta fórmula para trabajar en proyectos web (Apache, Mysql y PHP), existe forma de configurarlo en nuestro sistema operativo, aunque el proceso puede ser un poco artesanal.

Cuando se busca una forma más rápida de arrancar con nuestro servidor, existen aplicaciones que ya incluyen estos paquetes y con un sencillo instalador nos dejan todo listo para trabajar en un par de minutos, estos nos ayudan en gran parte el trabajo de instalar y configurar, en la **Tabla 2.1.**, se realiza un resumen de una muestra de estas aplicaciones:





APLICACIÓN	CARACTERÍSTICAS
	<p>WAMP es la abreviación de Windows, Apache, Mysql y PHP y es un proyecto desarrollado por franceses. Este paquete instala la versión 1.3 de Apache, PHP5, MySQL, PhpMyAdmin y SQLitemanager. Una de las ventajas de este paquete es la ofrecer la versión de PHP5 en el instalador y sus constantes actualizaciones.</p>
	<p>Otro paquete para Windows que tiene la ventaja de contar con un sitio en nuestro idioma es EasyPHP. Nos ofrece Apache, Mysql, PHP y varias herramientas para desarrollar como es el caso de PhpMyAdmin.</p>
	<p>MAMP es un paquete fácil de instalar que configurará todas estas aplicaciones en nuestra computadora. La abreviación de MAMP proviene de: Macintosh, Apache, Mysql y PHP. Es un paquete gratuito publicado bajo licencia GNU. Su objetivo es el de proveer un entorno de desarrollo bajo el Mac OS y sus creadores no recomiendan usarlo como servidor web en Internet.</p>
	<p>Este proyecto pensó inicialmente en el sistema operativo Linux. Hoy en día cuenta con distribuciones para diferentes sistemas operativos. XAMPP ha sido probado exitosamente en las distribuciones de Ubuntu, SuSE, RedHat, Mandrake and Debian. También tienen una versión que trabaja con Windows, otra para MacOSX y Solaris.</p>

Tabla 2.1.: Resumen de aplicaciones libres para servidores web.

Actualmente se cuenta con frameworks que facilitan aún más el trabajo de programar en un proyecto web. Existen múltiples Frameworks en JavaScript y PHP, muchos de ellos muy buenos y de código abierto otros no tan libres pero de igual forma muy buenos y algunos cuantos de licencia libre, pero promovidos por grandes corporaciones como google o yahoo.

El aspecto más interesante que incluyen estos frameworks es que incluyen código reutilizable, pues son un conjunto de librerías a las cuales los podemos llamar en cualquier punto del proyecto, de esta manera el programador ahorra mucho tiempo los cuales puede utilizarlo en la presentación del mismo es decir en la capa de la Vista.

Y con esto muchas personas resultaron beneficiadas de estos códigos reutilizables. Hoy en día cada vez son más personas las cuales se unen a estas formas de reutilizar a tal grado que las compañías grandes han liberado una serie de elementos, aquí se muestra (**Tabla 2.2.**) algunos de ellos y sus características.

FRAMEWORK	CARACTERÍSTICAS
Zend Framework	Es muy desacoplado, por lo que mucha gente lo considera una librería de componentes más que un framework. Por otra parte, gracias a esto, podemos utilizarlo en conjunto con otros frameworks. Tiene detrás a Zend, una de las empresas más importantes de la comunidad PHP, y gran contribuidora a su código desde sus primeras versiones. Recomendado para proyectos grandes.
Symfony	El framework que suele utilizar Yahoo, por ejemplo. Bastante popular hoy en día. Más complejo y elegante que CakePHP y CodeIgniter, pero menos que Zend. Muy completo. El "azúcar sintáctico" y la generación de código ayuda a recortar el tiempo de desarrollo. Recomendado para cualquier tipo de proyecto, en ocasiones combinado con algún componente de Zend.
CakePHP	Toma muchas de sus ideas de Ruby on Rails, por lo que gustará especialmente a los fans de la autoconfiguración y de lo "mágico". Debido a esto, no es lo más adecuado para aquellos que quieran tener un mayor control. Recomendado para proyectos no demasiado complejos, en ocasiones combinado con algún componente de Zend.
CodeIgniter	Probablemente, el menos popular de los cuatro. Muy flexible y sencillo. A menudo demasiado, obligando al programador a desarrollar muchas funcionalidades que deberían incluirse por defecto. Muy fácil de aprender, principalmente porque no hay mucho que aprender, al ser simple hasta el extremo. También es el que menos recursos utiliza, al ser el más ligero.

Tabla 2.2.: Resumen de los Frameworks más populares que utilizan código PHP.

[WWW.003]

Y para facilitar aún más el trabajo de programación web, se cuenta con un entorno de desarrollo como es NETBEANS, hecho principalmente para el lenguaje de programación Java, pero actualmente podemos descargarnos el paquete de Netbeans para trabajar con PHP5 y superiores. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans permite crear aplicaciones Web con PHP 5, un potente debugger integrado y además viene con soporte para Symfony un gran framework MVC escrito en php. Al tener también soporte para AJAX, cada vez más desarrolladores de aplicaciones LAMP o WAMP, están utilizando NetBeans como IDE. Al día de hoy hay disponibles dos productos: el NetBeans IDE y NetBeans Platform:

NetBeans IDE	NetBeans Platform
Es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.	Una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

Tabla 2.3.: Resumen de 2 productos Netbeans

Ambos productos son de código abierto y gratuito para uso tanto comercial como no comercial. El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL) v1.0 and the GNU General Public License (GPL) v2. **[WWW.004]**

2.2.2. Herramientas de Software Libre

Tal vez uno de los mayores desafíos que presenta el Software Libre es la variedad de aplicaciones licenciadas para cada una de las ramas de la actividad humana.

Sería muy pretencioso recomendar Software Libre específico que cubra toda la variedad de realidades que pueden requerir soluciones particulares. De hecho, se recomienda, en el caso de estar a cargo de una organización que plantea usar software licenciado como libre en forma exclusiva, se vincule con una comunidad de usuarios que pueda brindarle contención y asesoramiento.

Actualmente hay excelentes herramientas de Software libre, que tienen iguales o mejores prestaciones a sus equivalentes comerciales. En el siguiente cuadro se resume algunas de ellas.

HERRAMIENTA	DESCRIPCIÓN
Netbeans	NetBeans es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.
Symfony	Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características.
MySQL Workbench	Es una herramienta que permite modelar diagramas de entidad-relación para bases de datos MySQL, podemos utilizarla para diseñar el esquema de una base de datos nueva, la misma que se lo hace en representación visual.
The Gimp (Edición y retoque de fotografías)	Equivalente a Photoshop. Su última versión ha mejorado muchísimo muchas carencias que su versión anterior tenía. Ahora es más amigable y más potente.
Joomla (Sistema de Administración de Contenidos en Línea o CMS).	Gracias a este programa se puede tener aplicaciones para Podcast, traducción en línea, comentarios e incluso foro y comercio electrónico. Una buena opción y con muchos recursos en línea que permiten un aprendizaje sencillo.
OsCommerce (Comercio electrónico).	Esta herramienta permite tener una tienda en línea y controlar desde los precios, hasta calcular impuestos, tipos de cambio y costos de entrega.
OpenOffice. (Aplicaciones para la productividad de tu negocio).	Open Office es el equivalente a Microsoft Office (Word, Excel, Powerpoint), tiene gran compatibilidad con Microsoft Office, y permite tener hojas de cálculo y procesador de texto sin preocuparse de pagar por licencia.

Tabla 2.4. Resumen de Herramientas libres para diferentes propósitos.

En la mayoría de los casos, este tipo de Software sustituye a sus equivalentes comerciales, tanto en facilidad de uso como en poder y eficacia.

2.2.3. Software Propietario y Software Libre.

Tradicionalmente se ha asociado software libre con software de baja calidad, pensando que no ofrece ninguna garantía porque no ha pasado por controles de calidad, no ofrece soporte, etc., pero esta forma de pensar está cambiando cada vez más, debido a la aparición de software libre de alta calidad y al soporte de las comunidades de software libre en la red.

Cada vez más las personas utilizan la web como fuente de información, especialmente para acceder a noticias de última hora y redes sociales; este hecho está situando a la web a un nivel similar al de medios de masas tan poderosos como la televisión o la prensa escrita. A pesar de ello, una característica que diferencia a la web de otros medios de comunicación es la sencillez de las infraestructuras necesarias para la construcción de un sitio.

En vista del crecimiento de la tecnología y la necesidad de ampliar la visión del tratamiento de información de las entidades públicas y privadas se ve la necesidad de crear portales web en donde se puede administrar la información.

Para la implantación de un sitio web deben considerarse los distintos elementos software que permitirán configurar el sitio con todas las funcionalidades requeridas (presentación estática de contenidos, transacciones, contenidos dinámicos, seguridad, etc.).

La Elección puede estar entre herramientas comerciales y herramientas libres, la respuesta a esta interrogante depende de los recursos con que se cuenta antes de empezar el proyecto, estos recursos pueden ser humanos y económicos.

Para elegir los recursos humanos apropiados, es decir el equipo de trabajo apropiado, se debe contar con personas con experiencia y responsables, con un buen conocimiento en la materia y sobre todo con ética profesional.

Para solventar los gastos se debe contar con los recursos económicos necesarios, que van a ser menores en comparación con sus equivalentes comerciales.

Las herramientas libres son una buena alternativa para proyectos investigativos, y que por supuesto al ser libres no significa que son despreciables, pues la mayoría de este tipo de herramientas son de gran calidad iguales o superiores a las otras herramientas.

En una empresa de software privativo es necesario reinventar la rueda desde cero, beneficiándose por la venta de licencias del código. Por otro lado, en una empresa de software libre se reaprovecha la gran cantidad de código libre de alta calidad existente en el mercado, acelerando el desarrollo, en este caso se puede aprovechar de otras formas, pero entre ellas no estará la venta de licencias.

En una empresa de software privativo, existe más flexibilidad a la hora de contratar programadores brillantes, o programadores no tan brillantes. Podemos montar una empresa de software privativo contratando un ejército de programadores expertos en un solo lenguaje de programación, encerrándolos en una sala y haciéndoles trabajar fuerte hasta que el programa esté listo. Este sistema podría funcionar con otro tipo de empresa como puede ser las empresas cárnicas, y les va muy bien. Sin embargo, este modelo de negocio simplemente no funciona cuando tratamos el software libre: no podemos mandar a este ejército de expertos en un solo lenguaje de programación a dar la consultoría y los servicios que serán la principal fuente de renta de una empresa de software libre. Tanto en las empresas de software libre, como en las de software privativo, hay otra cultura de empresa posible. Esta cultura de empresa es posible en software privativo, e indispensable si queremos aprovechar las ventajas del software libre [WWW.005].

2.2.4. Criterios en la selección de Herramientas de Software Libre.

En la actualidad, al momento de proponerse diseñar un software o un portal con algún propósito específico, se debe tomar en cuenta muchos aspectos, los cuales pueden ser relevantes en el trabajo que se va a realizar, los mismos que aceleran o detienen el trabajo, perjudicando el objetivo propuesto.

Es así que en el caso de software libre, estas herramientas ofrecen ventajas de índole económica, social y operativa.

Una vez terminado el software, como mercadería, por lo general no está a la venta, en realidad lo que el usuario adquiere, a través de una transacción monetaria o sin ella, es una licencia respecto de los usos que puede dar a los programas en cuestión.

Aquí se debe tomar en cuenta que a diferencia de, por ejemplo, un libro o un disco, mercaderías en las que el cliente adquiere título real sobre algo que puede prestar, regalar, revender, citar, alquilar, resumir, etc., en cambio al comprar un programa (software), el usuario por regla general no adquiere derecho de propiedad alguno, en muchos casos se ha visto que ni siquiera pasa a ser propietario del medio magnético u óptico (disco compacto, memorias flash, etc.) en el que el software es entregado, el mismo que continúa siendo propiedad del autor original.

Las licencias de uso de un programa en particular regula las maneras en las que el usuario puede utilizarlo, las mismas que se las puede clasificar en dos grandes categorías: por un lado están las licencias conocidas como "libres", y por otro las "propietarias". La gran diferencia entre estos tipos de licencias consiste en que un software licenciado de modo propietario por lo general otorga al usuario solamente el derecho de ejecutar el programa con errores incluidos en determinada computadora, prohibiendo expresamente todo otro uso, mientras que el software gobernado por una licencia libre permite al usuario no solo ejecutar el programa en tantas computadoras como desee, sino también copiarlo, inspeccionarlo, modificarlo, mejorarlo, corregir errores y distribuirlo, o contratar a alguien para que lo haga por él.

PARÁMETROS	SOFTWARE PROPIETARIO	SOFTWARE LIBRE
Estaciones de Trabajo	La estación de trabajo requiere altos recursos en cuanto a hardware.	Puede ser equipos reciclados de segunda mano, baratos, con poco recurso de hardware.
Pago de Licencias	Es necesario pagar licencias aún en el caso que sean donados son limitados en la forma de trabajar.	Existen una buena cantidad de programas que pueden ser usados legalmente sin paga alguna.
Condiciones de Licencias	La aceptación de las condiciones de licencia diseñadas con el expreso propósito de evitar la difusión del conocimiento encapsulado en los programas	Las condiciones de licenciamiento son claras, sencillas de cumplir, están diseñadas para fomentar la difusión de las ideas y mecanismos corporizados en los programas.
Idioma del Software	Sólo están disponibles en los idiomas que el propietario decidió soportar, normalmente el castellano no está, por considerarlo minoritario.	Aunque los programas que no estén disponibles en el lenguaje deseado pueden ser traducidos, sin entrar en conflicto con la licencia.
Difusión del Software	No se puede apropiarse directa ni indirectamente de estas herramientas, cuyas condiciones de licenciamiento le prohíben expresamente aprender cómo	Estas herramientas se puede hacer tan propias como desee, incluyendo no sólo la capacidad de usarlas, sino también de desmantelarlas,

funcionan, asignándole así el rol de consumidor.	recombinarlas, crearlas y mejorarlas.
Se enseña al estudiante que el software que usa no debe ser compartido con nadie, pues hacerlo es delito, y que es normal y aceptable renunciar al derecho a la solidaridad a cambio de un poco de comodidad personal.	Se enseña al estudiante que la solidaridad y el compartir son actitudes socialmente positivas también en el mundo del software, y que existen modalidades para copiarlo y compartirlo libremente sin violar la ley

Tabla 2.5. Tabla comparativa entre software privativo y libre.

2.2.5. Servidor Web.

Un servidor web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP). Un servidor web básico tiene un esquema de funcionamiento muy sencillo, ejecutando de forma infinita el bucle siguiente:

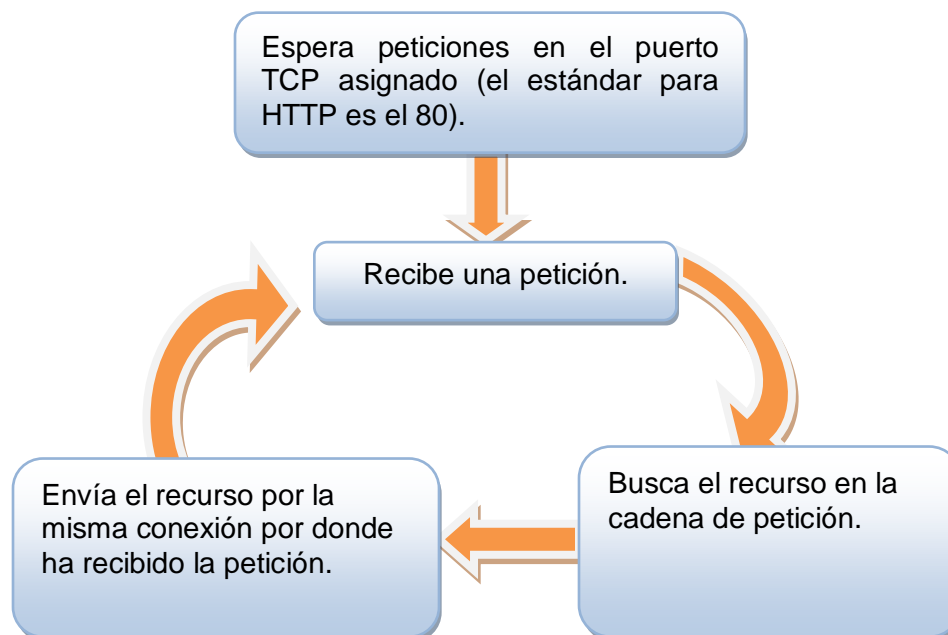


Gráfico 2.3.: Esquema de funcionamiento de un Servidor Web

Fuente: Diseño Propio

Un servidor web que siguiese el esquema anterior cumpliría los requisitos básicos de los servidores HTTP, aunque, eso sí, sólo podría servir ficheros estáticos.

A partir del esquema anterior se han diseñado y construido todos los programas servidores de HTTP que existen, variando sólo el tipo de peticiones (páginas estáticas, CGI, Servlets, etc.) que pueden atender, en función de que sean o no multi-proceso, multi-hilados, etc.

Uno de los servidores más populares y eficientes en el diseño de portales es **Apache**, pues es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

La historia de Apache se remonta a febrero de 1995, donde empieza el proyecto del grupo Apache, el cual está basado en el servidor Apache http de la aplicación original de NCSA. El desarrollo de esta aplicación original se estancó por algún tiempo tras la marcha de Rob McCool por lo que varios webmaster siguieron creando sus parches para sus servidores web hasta que se contactaron vía email para seguir en conjunto el mantenimiento del servidor web, fue ahí cuando formaron el grupo Apache. Fueron Brian Behlendorf y Cliff Skolnick quienes a través de una lista de correo coordinaron el trabajo y lograron establecer un espacio compartido de libre acceso para los desarrolladores. Fue así como fue creciendo el grupo Apache, hasta lo que es hoy, aquella primera versión y sus sucesivas evoluciones y mejoras alcanzaron una gran implantación como software de servidor inicialmente solo para sistemas operativos UNIX y fruto de esa evolución es la versión para Windows **[WWW.006]**.

Apache es una muestra, al igual que otros software libre de que el esfuerzo grupal dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar.

En cuanto a la licencia de Apache es una descendiente de la licencias BSD, no es GPL. Esta licencia permite hacer lo que quiera con el código fuente **[WWW.006]**.

En el siguiente gráfico se resume las razones del porque Apache ha llegado a ganar mucha popularidad en muchos ámbitos empresariales y tecnológicos:

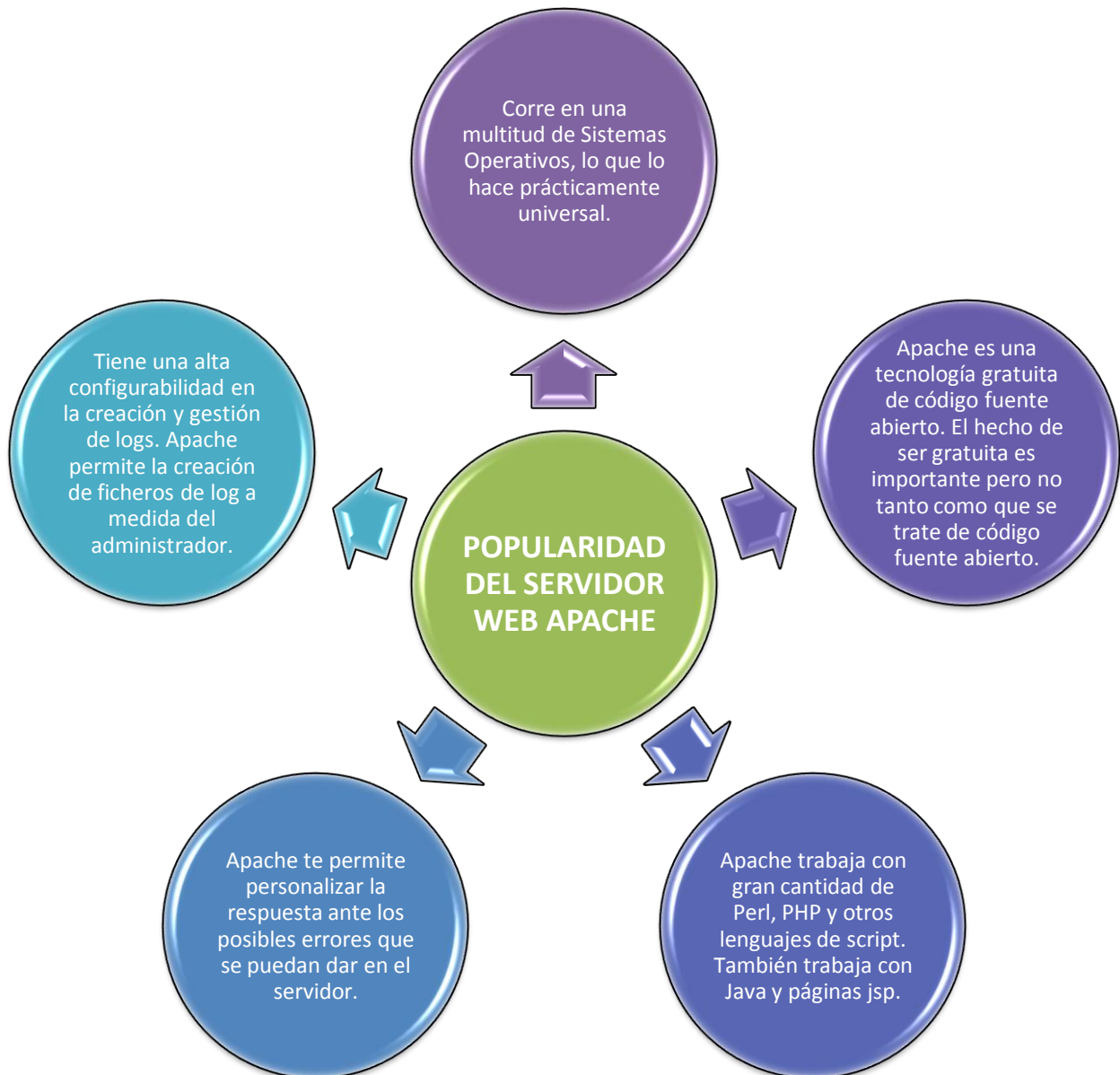


Gráfico 2.4. Razones de la Popularidad de servidor Web Apache.

Fuente: Diseño Propio

Se pueden extender las características de Apache hasta donde nuestra imaginación y conocimientos lleguen, hay instaladores para distintos sistemas operativos, estas las podemos encontrar en la web www.apache.org.

Todas estas razones suficientes para elegir el servidor web Apache, no solo por ser el servidor más utilizado sino porque ese porcentaje de utilización es un indicador de su estabilidad y calidad.

2.2.6. Base de Datos

En principio y tras ver tres sistemas (MySQL, PostgreSQL e InterBase) que eran capaces de soportar, en mayor o menor grado, todos los requerimientos de funcionamiento, se descartó InterBase, por su pobre rendimiento en ambientes con muchos usuarios concurrentes, pues es una herramienta básica que se puede utilizar para experimentar y en la educación, mas no para implementar un portal web, y por la limitación encontrada en el tamaño máximo que puede tener una fila de una tabla, que en este caso era un tema importante; por lo tanto la elección se centraba entre MySQL y PostgreSQL, ambos SGBD (Servidor de Gestión de Base de Datos) cuentan con una base de usuarios bastante extendida, con multitud de interfaces, APIs y documentación y además son portables a una gran cantidad de plataformas.

Postgresql es complejo y trata de estar a la altura de grandes SCBD como es Oracle, es por eso que Mysql es una buena opción en el caso de que se requiera agilidad y seguridad en las transacciones, este es utilizado en la mayoría de portales que no tienen muchas transacciones con pocos usuarios concurrentes, al contrario de PostgreSQL que es más lento pero que es óptimo para muchos usuarios concurrentes.

MySQL es un gestor de base de datos sencillo de usar y rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

PARÁMETROS	POSTGRESQL	MYSQL
Velocidad	Es de 2 a 3 veces más lenta que MySQL.	Mayor velocidad tanto al conectar con el servidor como al servir selects y demás
Recursos	Consume bastante más recursos y carga más el sistema	Consume muy pocos recursos, tanto de CPU como de memoria.
Roll-backs y Subselects	Soporta transacciones roll-backs y subselects sin ningún problema.	No soporta transacciones, roll-backs ni subselects.
Concurrencia	Soporta una gran cantidad de transacciones concurrentes.	No soporta mucha cantidad de transacciones concurrentes.
Consistencia de la BD	Para sistemas más serios en las BD con información realmente importante, bancos, etc., PostgreSQL es una mejor opción pese a su mayor lentitud.	En este caso, le toma mayor importancia la velocidad o agilidad en las transacciones, dejando a un lado la consistencia de la BD.

Tabla 2.5. Tabla comparativa entre postgresql y mysql.

A continuación vamos a resumir en un gráfico las características principales de MySQL:

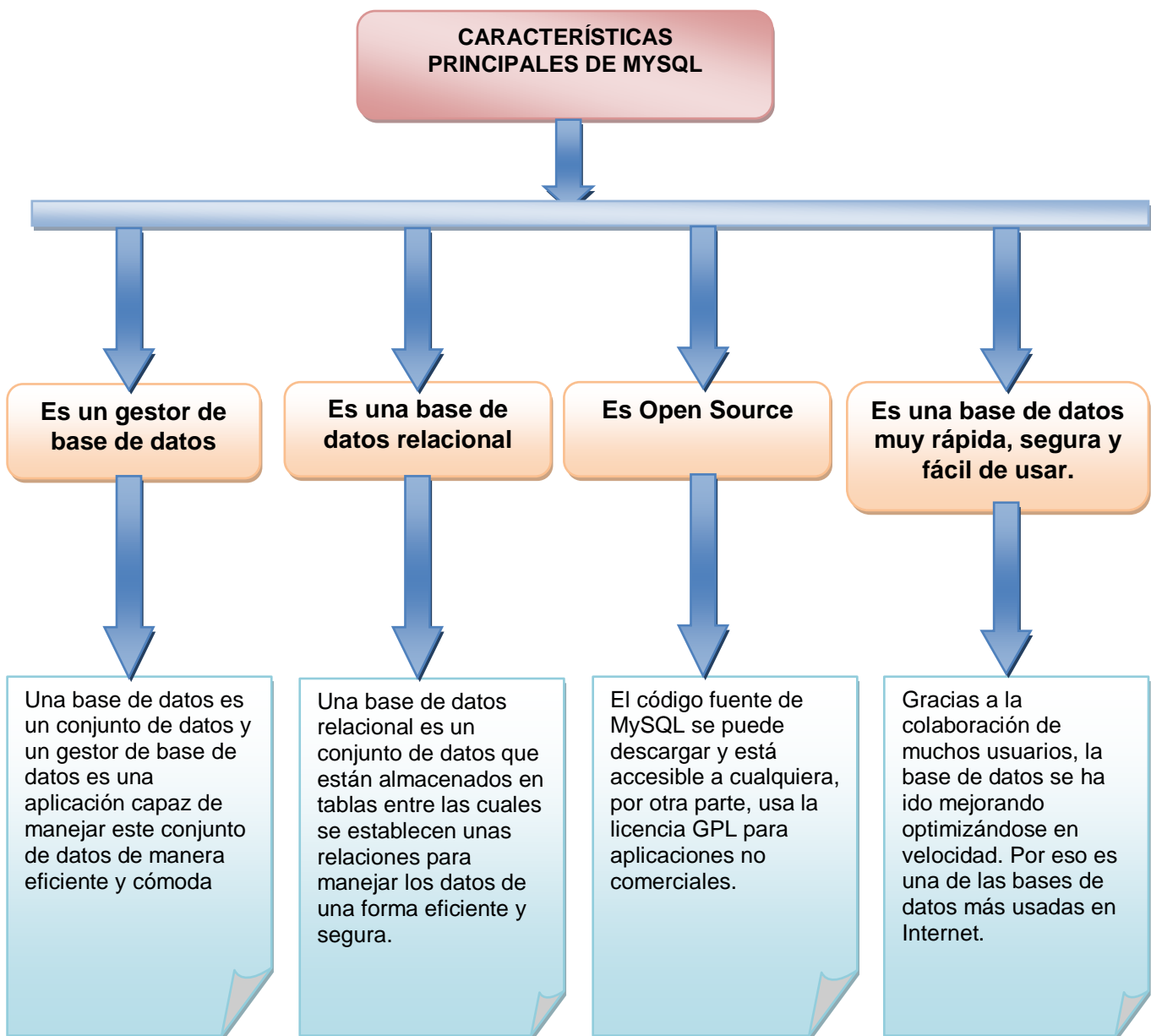


Gráfico 2.5.: Características principales de MySql.

Fuente: Diseño Propio

2.2.7. Lenguaje de Programación

Con lo que respecta al lenguaje de programación se optó entre dos posibilidades, Java y PHP, cualquiera de estos dos podrían servir perfectamente para las necesidades del

proyecto a desarrollar.

Es así que la elección del presente proyecto está en **PHP**, pues la sintaxis de PHP se asemeja a la del lenguaje C, resulta sencillo de comprender y de mantener; por otra parte, la gran calidad de la documentación disponible y el enorme número de funciones que proporciona hacen realmente sencillo su empleo.

PHP Es un lenguaje del lado del servidor, esto significa que el código PHP va a ser interpretado en el servidor. Así, lo que el usuario verá en pantalla será el contenido generado por la interpretación del archivo PHP que el usuario está solicitando.

Una de las grandes ventajas que tiene PHP es su gran versatilidad a la hora de trabajar con bases de datos, permitiendo realizar páginas con información almacenada en forma persistente en bases de datos y mostrarla según la petición de los usuarios.

Dentro de un archivo PHP se puede alternar entre código html y código PHP. Esto nos permite realizar páginas dinámicas de forma fácil: realizando toda la estructura con html y luego agregando el contenido dinámico.

Es así que, una página html común, pero en lugar de ser index.html es index.php. Cuando se quiera agregar código PHP habrá que hacerlo dentro de las etiquetas `<?php y ?>` o `<? y ?>`, por ejemplo:

```
<html>
<head>
<title>hola mundo</title>
</head>
<body>
<h1>incluir código PHP en html</h1>
<p><? acá va el código PHP ?></p>
</body>
</html>
```

PHP es un lenguaje de scripting que permite la generación dinámica de contenidos en un servidor web. El significado de sus siglas es HyperText Preprocessor. Entre sus principales características cabe destacar su potencia, su alto rendimiento, su facilidad de aprendizaje y su escasez de consumo de recursos **[WWW.007]**.

El código **PHP** puede incluirse dentro del código html de la página. Para delimitar la

sección de código PHP podemos hacerlo de varias formas:

- Usando las etiquetas `<?php y <?`
- Usando las etiquetas `<? y ?>`
- Mediante `<script lenguaje="php"> </script>`

El funcionamiento de las páginas en **PHP** alojadas en un servidor se esquematiza en la siguiente figura:

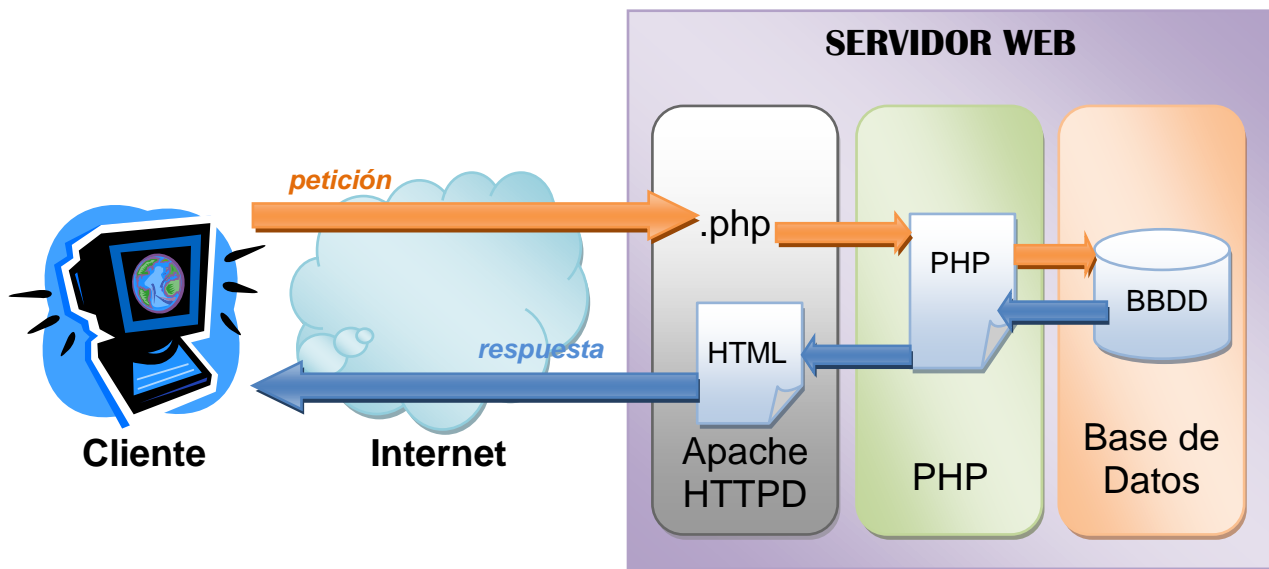


Gráfico 2.6. Funcionamiento de Páginas PHP

Fuente: Diseño Propio

Es decir la forma de trabajar es:

1. El navegador del cliente solicita el documento **PHP**.
2. Llega la solicitud del servidor y el servidor localiza el documento, lanza el intérprete de PHP y ejecuta todo su código.
3. Una vez ejecutado el código se genera el resultado en HTML y lo devuelve al servidor para que lo transfiera al cliente.
4. El servidor transfiere el resultado en HTML y es mostrado en el navegador del cliente.

2.2.7. Framework.

En el apartado **2.2.1** se dijo que un framework en si se refiere a un conjunto de bibliotecas con código que puede ser reutilizado en cualquier punto de su proyecto, los mismos que

ayudan y facilitan en gran parte el desarrollo de una aplicación.

Esta palabra "framework" se refiere a 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.

En el transcurso del desarrollo de una aplicación un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, estas mismas características permite que otro proyecto pueda ser organizado y construido con mucha facilidad, ahorrando tiempo y dinero, lo cual puede ser clave para los intereses de una empresa o institución.

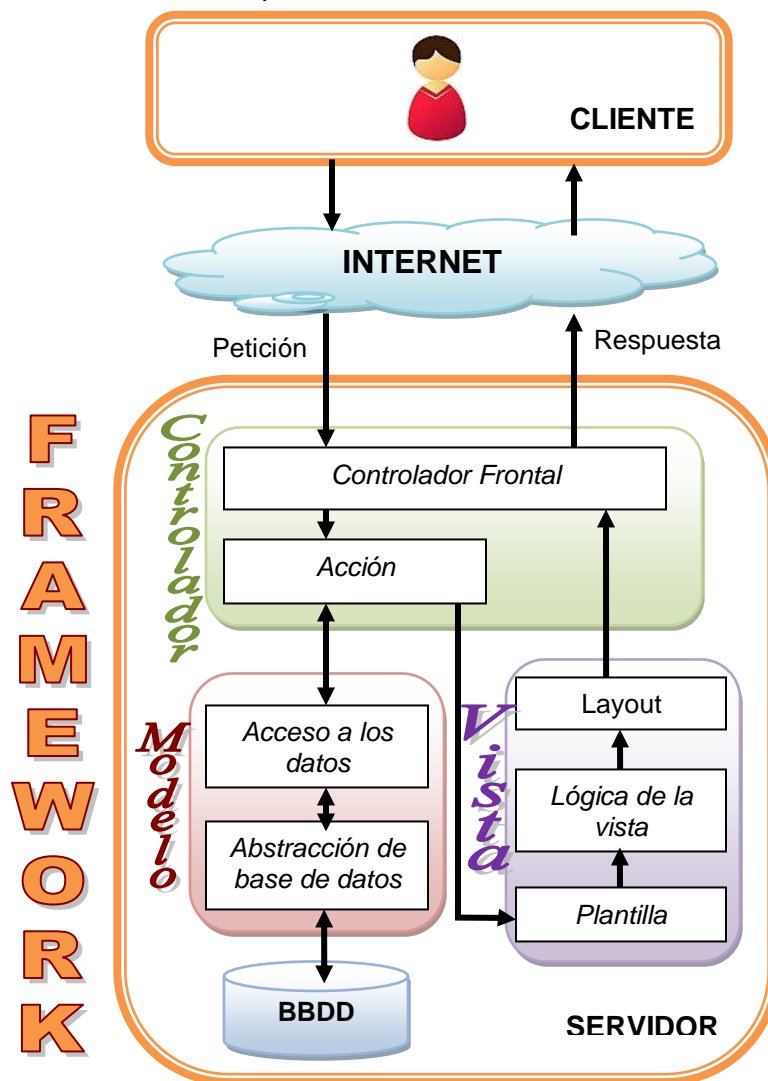


Gráfico 2.7. Estructura de un Framework con patrón de diseño MVC.

Fuente: Diseño Propio

Como ya se dijo, este tipo de programación modular permite trabajar con módulos, los mismos que pueden ser unidos según los requerimientos, este sistema puede ir incrementando en tamaño y prestaciones que ofrece.

En cuanto a la arquitectura para cada problema siempre hay buenas soluciones y para la programación web, la solución más utilizada actualmente para organizar el código es el patrón de diseño **MVC**. En pocas palabras, el patrón de diseño **MVC** organiza el código en base a su función. De hecho, este patrón separa el código en tres capas:

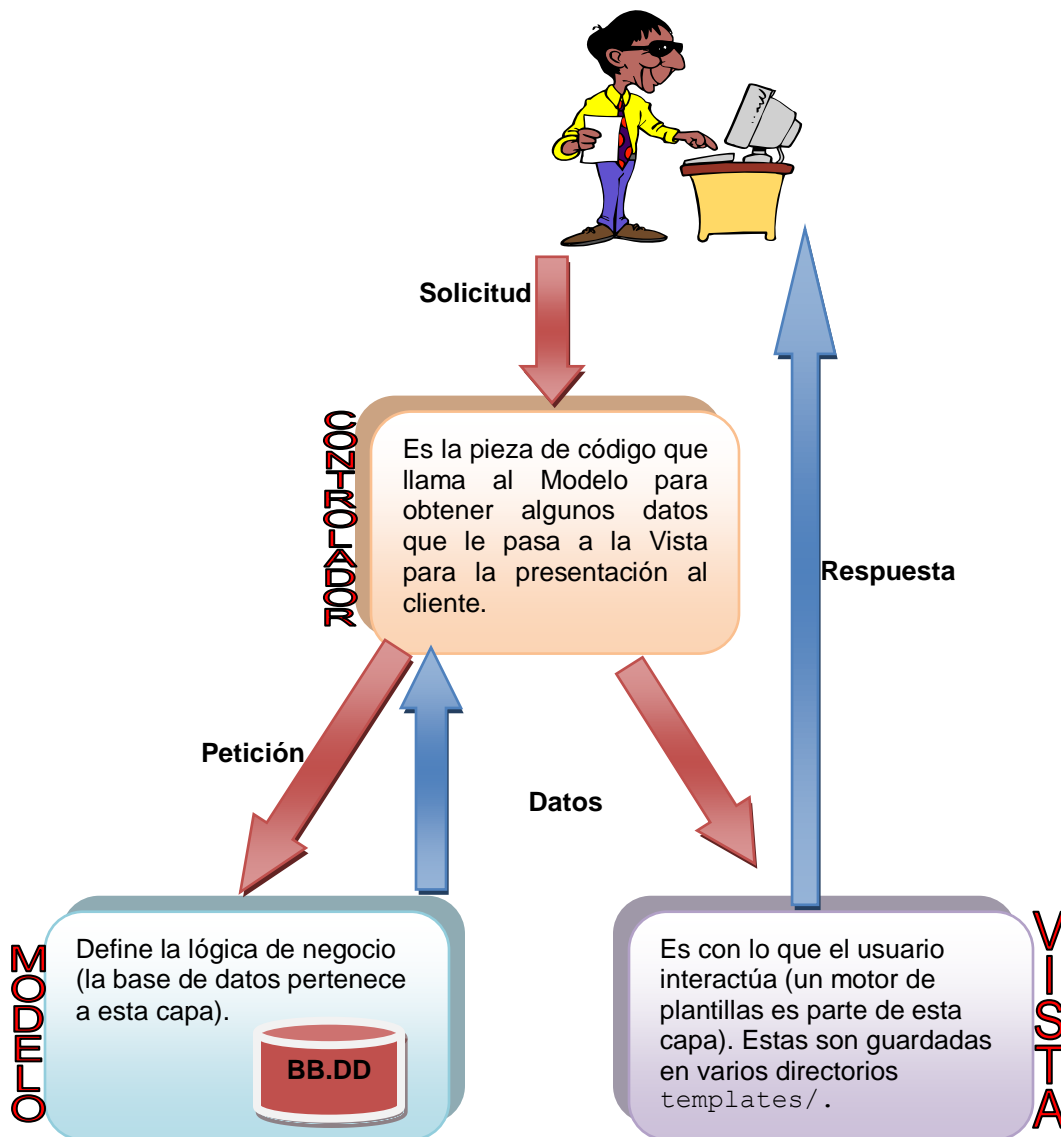


Gráfico 2.8.: Esquema de la Arquitectura MVC.

Fuente: Diseño Propio

2.2.8. Entornos de Desarrollo (IDE) para PHP

Es necesario un entorno de desarrollo integrado para programar, pues la programación visual nos facilitará aún más el desarrollo de una aplicación, a la hora de elegir una de estas herramientas se debe tomar en cuenta en que lenguaje de programación se va a trabajar, pues en este caso se necesita buscar un entorno de desarrollo que soporte PHP.

Un entorno de desarrollo integrado (en inglés integrated development environment) es un programa informático compuesto por un conjunto de herramientas de programación, puede ser el caso de que esta herramienta se dedique en exclusiva a un sólo lenguaje de programación o bien, puede utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic, por ejemplo, puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para este paquete de aplicaciones.

Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación, este es el caso de Netbeans, al que mediante plugins se le puede añadir soporte de lenguajes adicionales, los mismos que pueden ser Java o PHP. **[WWW.008]**

2.3. EL SOFTWARE LIBRE Y DESARROLLADORES.

2.3.1. Definición Aceptada de Software Libre

Para poder dar una definición, se va a tomar en cuenta las diferencias ético-filosóficas, los desarrolladores están interesados del software como herramienta de trabajo, así pues, se hace necesario dar una definición del software libre más abierta que las propuestas por FSF y la iniciativa Open Source pero lo suficientemente restrictiva como para dejar fuera soluciones propietarias y cerradas; así, una posible definición de software libre podría ser la siguiente:

“Es SOFTWARE LIBRE aquel que puede utilizarse y distribuirse libremente, mediante el pago de unas tasas o de forma gratuita; cuyo código fuente es accesible o, en caso

contrario, está construido de acuerdo con unas especificaciones públicas y cuyo desarrollo es llevado a cabo por la comunidad de usuarios o cuenta con las aportaciones de los mismos.”

En esta definición de software libre tiene cabida tanto el Software Libre o el Código Abierto como iniciativas comerciales como Java o PHP. Entonces, se adopta esta definición para referirse al software libre.

2.3.2. Los Desarrolladores

Para los desarrolladores es necesario conocer que en la industria del software en donde se incluye software de código abierto, las ganancias se irán del desarrollo de software a la implementación y servicios de consultoría.

En el modelo de software comercial en la actualidad las ganancias son creadas cuando un producto de software es desarrollado, pero en un ambiente de código abierto o en la creación de este tipo de software el negocio está en el servicio para implementar y mantener el software.

La tecnología de código abierto promete software más económico para los usuarios pero esto no significa un software de menos calidad ya que puede ser de la misma o superior al de tipo comercial.

Los desarrolladores de software, especialmente compañías locales que apenas empiezan a cosechar algo de negocios, serán las más afectadas cuando la industria del software se mueva de un modelo de software comercial a uno que incluya software de código abierto, para tales compañías, el primer paso es aceptar que el software libre es real y entonces transformar sus negocios para sobrevivir.

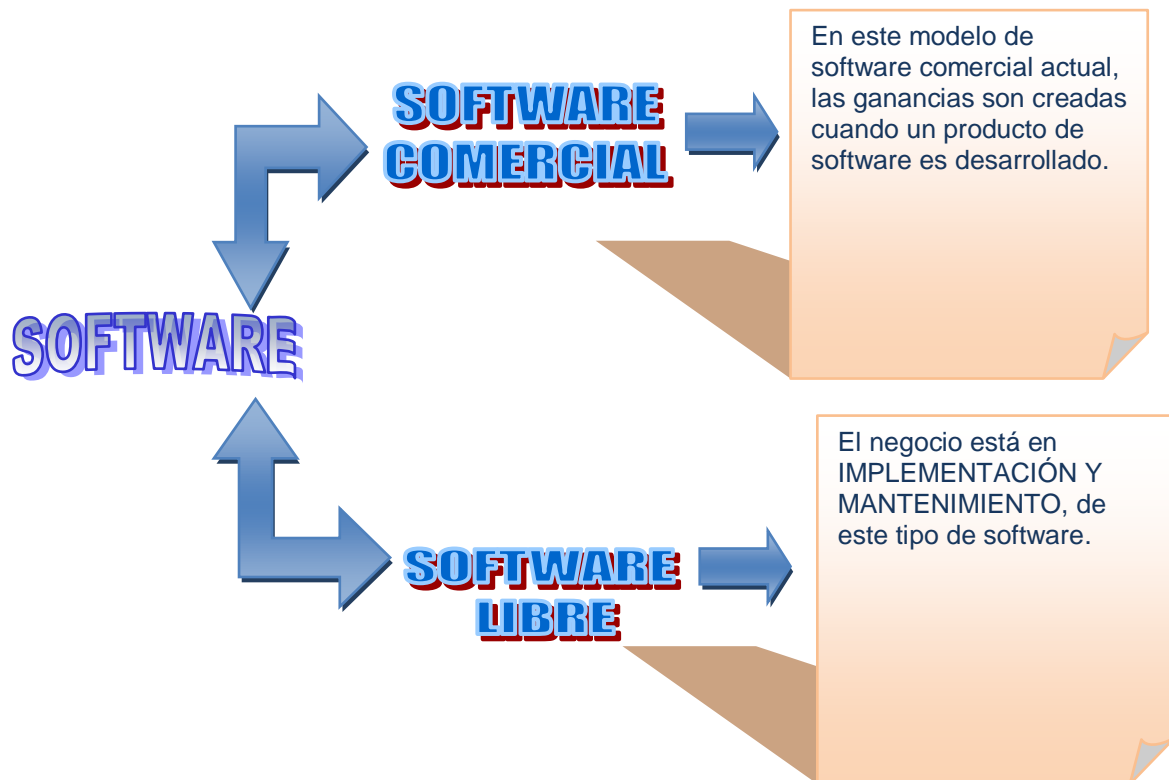


Gráfico 2.9. Utilidad generada en el desarrollo de Software Comercial y Software Libre.

Fuente: Diseño Propio

2.3.3. Licencias.

La LICENCIA es aquella autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer "actos de explotación legales". Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciatarario.

Existen distintas variantes del concepto o grupos de licencias desde el punto de vista de software libre que a continuación los resumiremos:

2.3.3.1 Licencias GPL

Esta licencia es una de las más utilizadas a la cual le conocemos como Licencia Pública General de GNU (GNU GPL). Aquí el autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que

todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que sea imposible crear un producto con partes no licenciadas GPL: el conjunto tiene que ser GPL.

En la práctica esto hace que las licencias de software libre se dividan en dos grandes grupos, aquellas que pueden ser mezcladas con código licenciado bajo GNU GPL (y que inevitablemente desaparecerán en el proceso, al ser el código resultante licenciado bajo GNU GPL) y las que no lo permiten al incluir mayores u otros requisitos que no contemplan ni admiten la GNU GPL y que por lo tanto no pueden ser enlazadas ni mezcladas con código gobernado por la licencia GNU GPL.

Podemos encontrar en el sitio web oficial de GNU (<http://www.gnu.org/>) una lista de licencias que cumplen las condiciones impuestas por la GNU GPL y otras que no.

Aproximadamente el 60% del software licenciado como software libre emplea una licencia GPL.

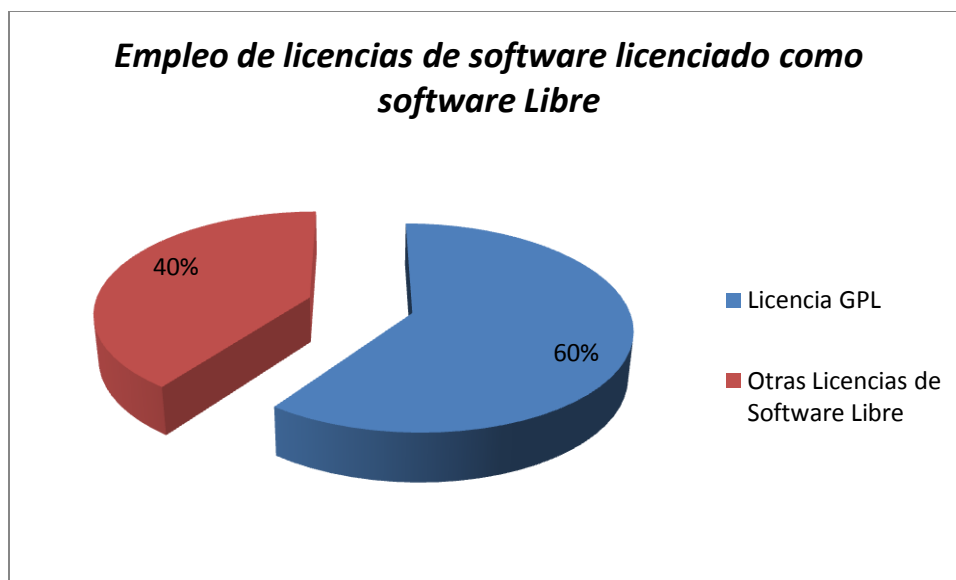


Gráfico 2.10. Porcentajes de uso de software licenciado como software Libre.

Fuente: Diseño Propio

2.3.3.2. Licencias estilo BSD

Se llaman así porque se utilizan en gran cantidad de software distribuido junto a los

sistemas operativos BSD (BSD son las siglas de "Berkeley Software Distribution", así se llamó a las distribuciones de código fuente que se hicieron en la Universidad de Berkeley en California y que en origen eran extensiones del sistema operativo UNIX de AT&T Research.). El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son compatibles. Puede argumentarse que esta licencia asegura "verdadero" software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre. Otras opiniones están orientadas a destacar que este tipo de licencia no contribuye al desarrollo de más software libre.

2.3.3.3. Licencias estilo MPL y Derivadas

Esta licencia de software libre tiene un gran valor porque ya que fue el instrumento que empleó Netscape Communications Corp. para liberar su Netscape Communicator 4.0 y la cual le sirvió para empezar ese proyecto tan importante para el mundo del Software Libre: **Mozilla**. Se utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos. La MPL es Software Libre y promueve eficazmente la colaboración evitando el efecto "viral" de la GPL (si usas código licenciado GPL, tu desarrollo final tiene que estar licenciado GPL).

Desde un punto de vista del desarrollador la GPL presenta un inconveniente en este punto, y lamentablemente mucha gente se cierra en banda ante el uso de dicho código. No obstante la MPL no es tan excesivamente permisiva como las licencias tipo BSD. Estas licencias son denominadas de copyleft débil. La NPL (luego la MPL) fue la primera licencia nueva después de muchos años, que se encargaba de algunos puntos que no fueron tenidos en cuenta por las licencias BSD y GNU. En el espectro de las licencias de software libre se la puede considerar adyacente a la licencia estilo BSD, pero perfeccionada.

2.3.3.4. Copyleft

Hay que hacer constar que el titular de los derechos de autor (copyright) de un software

bajo licencia copyleft puede también realizar una versión modificada bajo su copyright original, y venderla bajo cualquier licencia que desee, además de distribuir la versión original como software libre. Esta técnica ha sido usada como un modelo de negocio por una serie de empresas que realizan software libre (por ejemplo MySQL); esta práctica no restringe ninguno de los derechos otorgados a los usuarios de la versión copyleft. También podría retirar todas las licencias de software libre anteriormente otorgadas, pero esto obligaría a una indemnización a los titulares de las licencias en uso. En España, toda obra derivada está tan protegida como una original, siempre que la obra derivada parta de una autorización contractual con el autor. En el caso genérico de que el autor retire las licencias "copyleft", no afectaría de ningún modo a los productos derivados anteriores a esa retirada, ya que no tiene efecto retroactivo. En términos legales, el autor no tiene derecho a retirar el permiso de una licencia en vigencia. Si así sucediera, el conflicto entre las partes se resolvería en un pleito convencional.

[WWW.009].

2.4. DIFERENCIAS ENTRE SOFTWARE LIBRE, CÓDIGO ABIERTO Y FREeware

Con frecuencia los términos freeware, Software Libre y Código Abierto se utilizan de forma indistinta para referirse a un software que, en teoría:

- Es gratuito,
- No tiene propietario y
- Carece de soporte.

2.4.1. Software Libre (Free Software)

El Software Libre o GNU, nació por la necesidad de contrarrestar las privatizaciones de software que se llevaron a cabo sistemáticamente en USA y en el mundo. Los grandes monopolios y mercaderes empezaron a robar el software que se compartía libremente, establecido en: las universidades, organizaciones y programadores que aportaban su software de forma voluntaria y sin ánimo de lucro. Nace el proyecto GNU, con la concepción de crear un sistema operativo completamente libre. Esto implicó el planteamiento de 5 leyes básicas para que el Software sea GNU o sea libre. Las 4

Libertades:

- Libertad 1. Libertad para ejecutar el programa con cualquier propósito.
- Libertad 2. Libertad para estudiar y modificar el programa.
- Libertad 3: La Libertad de (re)distribuir copias.
- Libertad 4. Libertad publicar las modificaciones.

Y la quinta ley, la que protege las 4 libertades y obliga su cumplimiento es la licencia Copyleft. Esta licencia es la que ratifica que el software sea libre y no como el antiguo software compartido, impidiendo la expropiación del Software Libre a manos de privatizadores del software, esta licencia es la que respalda al GNU legalmente.

	Características	Delito	Resolución
Software Libre	Nació por la necesidad de contrarrestar las privatizaciones de software que se llevaron a cabo sistemáticamente en USA y en el mundo.	Los grandes monopolios y mercaderes empezaron a robar el software que se compartía libremente, establecido en: las universidades, organizaciones y programadores que aportaban su software de forma voluntaria y sin ánimo de lucro.	Proyecto GNU, que crea un Sistema Operativo Libre. Planteamiento de 5 leyes básicas. Libertad 1. Libertad para ejecutar el programa. Libertad 2. Libertad para estudiar y modificar el programa. Libertad 3: Libertad de (re)distribuir copias. Libertad 4. Libertad publicar las modificaciones. La quinta ley protege las 4 libertades y obliga su cumplimiento.

Tabla 2.5. Características del Software Libre.

2.4.2. Código Abierto (Open Source)

La iniciativa Open Source nació a partir de la FSF (Free Software Foundation) y aunque gran parte de las características de los programas Open Source son virtualmente equivalentes a las del Software Libre ambos movimientos tienen puntos de vista totalmente diferentes.

El Software Libre no necesariamente es gratuito, de hecho su denominación de Libre se debe a que se tratan de programas de Código Abierto (Open Source) y es ahí donde reside la esencia de su libertad: los programas bajo licencias GPL, una vez adquiridos, pueden ser usados, copiados, modificados y redistribuidos libremente.

Para que un software sea definido como software libre, o bien, para que lo sea de código

abierto, o ambos, debe cumplir ciertas reglas o normas:

SOFTWARE LIBRE	CÓDIGO ABIERTO
LAS 4 LIBERTADES DEL SOFTWARE LIBRE	LAS 10 PREMISAS DE SOFTWARE DE CÓDIGO ABIERTO
Ejecutar el programa con cualquier propósito (Libertad 0) (privado, educativo, público, comercial, militar, etc.)	Libre redistribución: el software debe poder ser regalado o vendido libremente.
Estudiar y modificar el programa (Libertad 1) (para lo cual es necesario poder acceder al código fuente)	Código fuente: el código fuente debe estar incluido u obtenerse libremente.
Distribuir el programa de manera que se pueda ayudar al prójimo (Libertad 2)	Trabajos derivados: la redistribución de modificaciones debe estar permitida.
Distribuir las versiones modificadas propias (Libertad 3)	Integridad del código fuente del autor: las licencias pueden requerir que las modificaciones sean redistribuidas sólo como parches.
	Sin discriminación de personas o grupos: nadie puede dejarse fuera.
	Sin discriminación de áreas de iniciativa: los usuarios comerciales no pueden ser excluidos.
	Distribución de la licencia: deben aplicarse los mismos derechos a todo el que reciba el programa.
	La licencia no debe ser específica de un producto: el programa no puede licenciarse solo como parte de una distribución mayor.
	La licencia no debe restringir otro software: la licencia no puede obligar a que algún otro software que sea distribuido con el software abierto deba también ser de código abierto.
	La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software.

Tabla 2.6. Resumen de características de software libre y Código Abierto [WWW.010].

NOTA: Es importante señalar que las "libertades 1 y 3" del software libre obligan a que se tenga acceso al código fuente. La "libertad 2" hace referencia a la libertad de modificar y redistribuir el software libremente bajo algún tipo de licencia de software libre que beneficie a la comunidad.

2.4.3. Formas de Distribución, Freeware y Shareware

Ambos términos hacen referencia a la forma en que una aplicación específica es

distribuida y a las condiciones bajo las cuales el usuario deberá pagar por su utilización.

Bajo el término freeware se suelen incluir aquellos programas que pueden redistribuirse sin pagar por ellos; sin embargo, aún cuando el freeware garantiza dos de las cuatro libertades exigidas por la definición de Software Libre de la FSF no existe la libertad de análisis ni la de modificación (puesto que el código fuente no se suele distribuir). Por esa razón, el freeware no es Software Libre (ni Código Abierto).

El shareware es similar pues permite la redistribución gratuita del software, sin embargo, exige el pago del software por parte del usuario si éste desea hacer un uso continuado del mismo; como se puede comprobar, el shareware además de las limitaciones del freeware elimina la libertad de ejecución estando, por tanto, fuera de la categoría de Software Libre, este tipo de software (programas) de libre distribución, es decir, que puede ser copiado libremente, para facilitar que el usuario pruebe el programa para saber si sirve a sus necesidades. Sin embargo, si decide utilizarlo una vez pasado del periodo de prueba (15 ó 30 días, generalmente), debe comprarlo o dejar de utilizarlo. Al proceso de compra se le denomina registro, ya que incluye el envío de datos del comprador al vendedor, además del pago en sí mismo. El precio de estos programas es muy inferior al de los programas comerciales, dado que, entre otras razones, la venta se realiza directamente de fabricante a usuario, eliminando los gastos y márgenes comerciales de la cadena de distribución **[WWW.011]**

A diferencia de lo que pueda pensarse inicialmente, existen muchos programas shareware que no tienen nada que envidiar, e incluso superan al software comercial.

Fuera ya de las licencias podemos encontrar diferentes formas de distribución de software, entre ellas el Freeware, el Shareware o el Adware. Para una mejor visualización y comprensión lo resumiremos en el siguiente **Gráfico 2.11**:

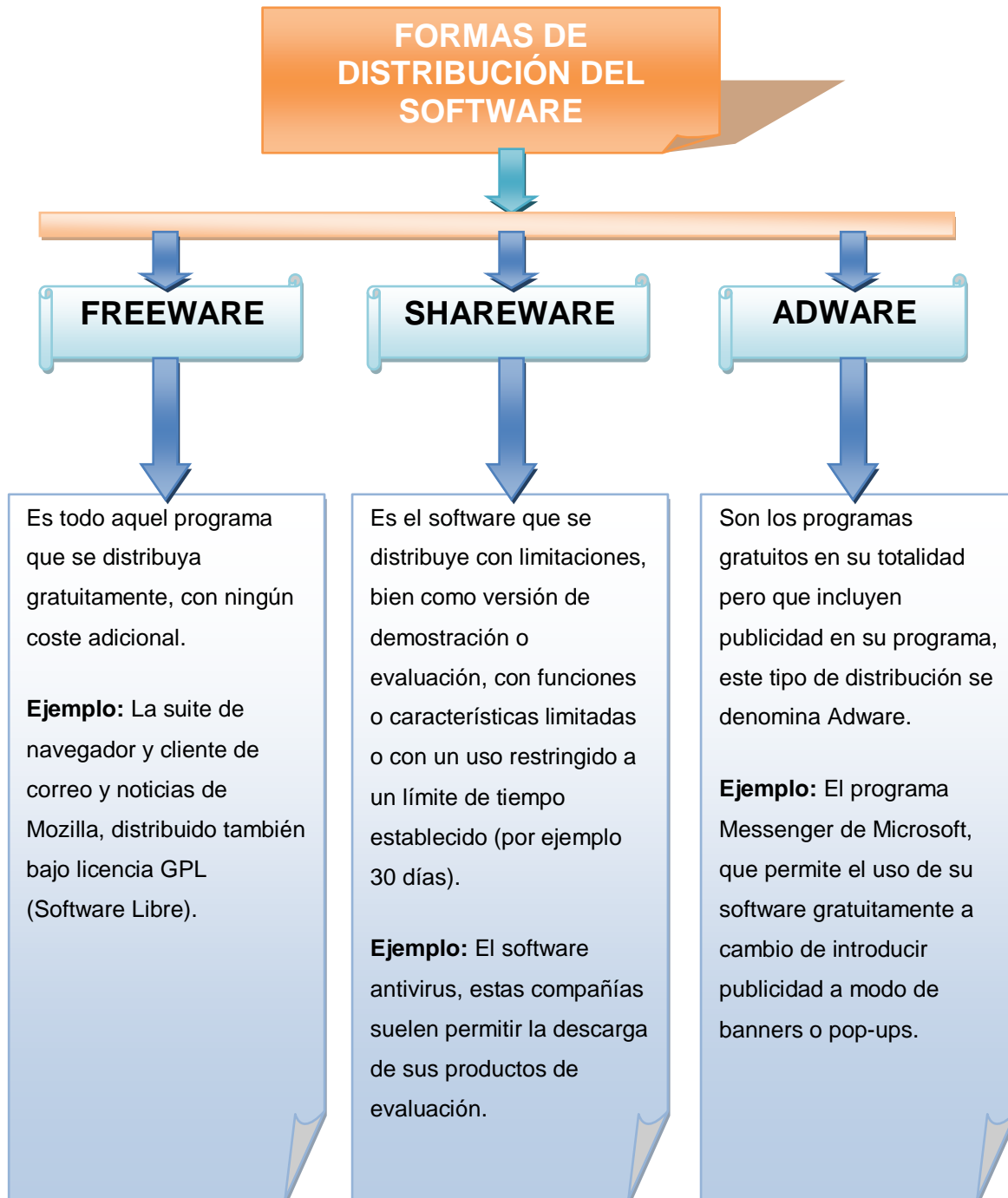


Gráfico 2.11.: Formas de Distribución del software.

Fuente: Diseño Propio

2.5. EL SOFTWARE LIBRE EN ECUADOR

2.5.1. Antecedentes

El Presidente de la República del Ecuador, Rafael Correa difunde un mensaje dirigido a

17 países del continente en el que hace un llamado a la utilización del software libre como medio para garantizar la soberanía y como paso para la integración y liberación de América Latina.

Este llamado lo realizó con el fin de que todos utilicemos los programas informáticos que pueden ser distribuidos, copiados, estudiados y modificados libremente.

En este mensaje, el cual ya se difunde en youtube (<http://www.youtube.com/watch?v=Hy5yAk4dYOk>), hace referencia a la utilización de software libre tanto a nivel público cuanto a nivel privado, pues dice que de esa manera garantizaremos la soberanía de nuestros estados, dependeremos de nuestras propias fuerzas, no de fuerzas externas a la región; seremos productores de tecnología, no simples consumidores; seremos dueños de los códigos fuentes; y podemos desarrollar muchos productos que, incluso, con una adecuada articulación de nuestros esfuerzos, puede ser de suma utilidad para las empresas públicas y privadas de la región.

Es por eso que el Gobierno ecuatoriano ya lo estableció como una política de Gobierno y de Estado.

Un gran paso para Ecuador, y todos esperamos que seguir avanzando, mejorando la integración de nuestra población a las tecnologías de la información, ya que actualmente tienen uno de los precios más elevados de conexión a internet en América Latina.

2.5.2. El Decreto 1014.

El día jueves 10 de Abril del 2008 se emitió el decreto 1014 por parte de la presidencia del Ec. Rafael Correa Delgado que promueve el uso de software libre en las instituciones públicas del Ecuador.

La URL donde se pueden descargar el documento del decreto es: **[WWW.012]**
Este es un pequeño extracto de lo que dice el documento:

Art. 1: Establecer como política pública para las entidades de administración Pública central la utilización del Software Libre en sus sistemas y equipamientos informáticos.

Art. 2: Se entiende por software libre, a los programas de computación que se pueden utilizar y distribuir sin restricción alguna, que permitan el acceso a los códigos fuentes y que sus aplicaciones puedan ser mejoradas.

Estos programas de computación tienen las siguientes libertades:

- Utilización de programa con cualquier propósito de uso común.
- Distribución de copias sin restricción alguna
- Estudio y modificación de programa (Requisito: código fuente disponible)
- Publicación del programa mejorado (Requisito: código fuente disponible)

Art. 3: Las entidades de la administración pública central previa a la instalación del software libre en sus equipos, deberán verificar la existencia de capacidad técnica que brinde el soporte necesario para este tipo de software.

Art. 4: Se faculta la utilización de software propietario(no libre) únicamente cuando no exista una solución de software libre que supla las necesidades requeridas, o cuando este en riesgo de seguridad nacional, o cuando el proyecto informático se encuentre en un punto de no retorno.

[WWW.013].

2.5.3. Gratas Consecuencias al Usar Software Libre

Con este antecedente, se hará mención de algunos de los motivos del por qué el crear Software Libre en Ecuador puede traer gratas consecuencias, las cuales se resumen en el siguiente cuadro.

CONSECUENCIA	DESCRIPCIÓN
GENERAR APOYO A LA INDUSTRIA LOCAL DE SOFTWARE	Esta es la más importante consecuencia, que se refiere a la de generar un apoyo a la industria local y será el principal proveedor de soluciones de software. Este impulso hará que la industria local no solo brinde servicios asociados a este tipo de software sino que cree productos de Software Libre.
IMPULSO AL TRABAJO COMUNITARIO	Esta es otra consecuencia interesante, y poco considerada, ya que el Software Libre se basa en gran parte en aportes de la comunidad. Esto ha creado una nueva cultura de compartir conocimientos que produce una ola imparable de innovación: si dos cabezas piensan mejor que una, imagínense millones de cabezas pensando a la vez. En Ecuador este es un concepto extremadamente revolucionario que puede traer muchos beneficios, brinda un impacto cultural que nos permitirá en realidad ser uno solo ante el mundo, de una forma práctica.
APORTES A LA CREACIÓN DE SOFTWARE LIBRE	Si en Ecuador se aporta en la creación de Software Libre, podremos contar con aplicaciones adaptadas a nuestra realidad. Esto es algo poco visto, ya que normalmente nos hemos acostumbrado a ser usuarios de terminología importada que viene como parte del software creado afuera.

Tabla 2.7. Consecuencias de la creación de Software Libre en Ecuador.

Se debe aprovechar este momento para ser proveedores y no solo consumidores de lo que será el futuro en la generación de software, y todo esa responsabilidad está en los desarrolladores de software o de al menos los que de una u otra manera están inmersos en este mundo e interesados en herramientas de Software libre para el diseño y desarrollo de Aplicaciones.

Actualmente en nuestro país el sitio web de la Subsecretaría de Informática www.informatica.gov.ec se encuentra disponible la Estrategia de Implantación de Software Libre en la Administración Pública Central, en la que se detalla el proceso. De las 30 instituciones que iniciaron con esta migración a Junio de 2009 el 90% de sitios web están desarrollados con Software Libre, 70% usan correo electrónico con aplicativos en Software Libre, 95% usan sistemas operativos basados en Software Libre en cuanto a servidores y 30% usan suite de oficina basada en Software Libre.

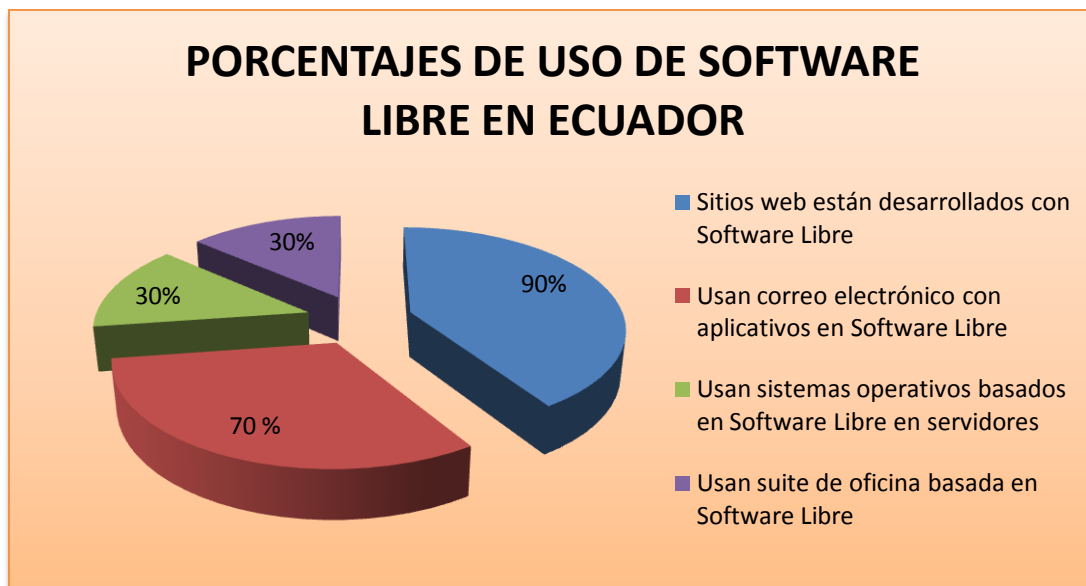


Gráfico 2.12.: Porcentaje del total de 30 instituciones que iniciaron migración a Junio 2009.

Fuente: <http://www.somoslibres.org/>

La Subsecretaría de Informática del Ecuador estableció como política el uso del Open Document Format ODF, lo que permite estandarizar la documentación en la Administración Pública Central [WWW.014].

La migración a Software Libre es un proceso que se debe realizar de manera planificada, lo que permitirá mostrar resultados a mediano y largo plazo, que son los objetivos del Gobierno Actual.

El modelo de negocio que persiguen las empresas de software libre es brindar soporte técnico y adecuar las funciones de los programas a las necesidades del cliente. Este servicio más tarde se revierte en fuentes de trabajo para los programadores nacionales.

El propio **Richard Stallman** en algún momento comentó sobre la aceptación del movimiento GNU/Linux en el gobierno de Ecuador. “Ecuador es un ejemplo actual de este importante progreso” -dijo Stallman-. “Su presidente, Rafael Correa, ha decidido migrar las agencias del estado al software libre, y ha establecido una política firme que está logrando la migración con éxito, tal es así que ahora continuará con la migración de sus escuelas públicas” [WWW.015].

2.6. SOFTWARE MÉDICOS EN EL MERCADO

2.6.1. Introducción

Los médicos suelen tratar con grandes cantidades de pacientes y dependiendo de la especialidad es la cantidad de síntomas y casos diferentes que deben atender acordarse de el expediente y síntomas de un paciente debe ser cansado cuando se atiende a 30 personas al día, muchos médicos deciden apoyarse en software medico para registrar a sus pacientes y tener un expediente rápido de cada paciente con cada visita.

Pero, estos programas de gestión médica la mayoría son costosos y no todos los médicos o entidades médicas están dispuestos a pagar por ellos, por suerte si nos tomamos un tiempo y buscamos en Internet Software Médicos podemos encontrar algunos muy económicos y accesibles, que se encuentran a la venta.

Aquí realizamos una muestra de software médicos en el mercado, pero no todos son libres, pero si son un poco más económicos:

2.6.1.1. ClearHealth

ClearHealth Advantage es un sistema empresarial escalable para satisfacer las necesidades más exigentes. La suite ClearHealth médica ofrece ahora una nueva generación de conjunto de capacidades para los grandes y pequeños consultorios.

2.6.1.2. EGADSS

EGADSS es una herramienta de código abierto que está diseñado para trabajar en conjunto con la atención primaria de Registro Médico Electrónico (EMR) para proporcionar sistemas de punto de pacientes específicos de los recordatorios de atención con el fin de ayudar a los médicos ofrecer atención de alta calidad. EGADSS está diseñado como un sistema independiente que responda a las solicitudes de los actuales registros médicos electrónicos, tales como Wolf, Medio de acceso, y MedOffIS para proporcionar una guía específica de pacientes clínicos sobre la base de su colección interna de las directrices. La centralización de la gestión directiva en virtud de una herramienta (EGADSS) permite un mantenimiento más fácil del contenido. EGADSS se centrará, inicialmente, en la atención preventiva recordatorios.

2.6.1.3. IndivoHealth

Indivo se refiere al Registro Personal de Salud Controlado (CPDH). El CPDH permite a un individuo a poseer y administrar una solución completa, copia digital segura de su información de salud y bienestar. Indivo integra la información de salud a través de sitios de la atención y el tiempo. Indivo es gratuito y de código abierto, utiliza estándares abiertos, no comprometido, y es utilizado activamente en diversos lugares.

2.6.1.4. OpenEMR

OpenEMR es un sistema de gestión médica libre, que administra historias clínicas, recetas por escrito, y la aplicación de facturación médica. Estos programas también se refieren a los registros de salud electrónicos. OpenEMR está licenciado bajo la GNU General Public License (GPL General). Se trata de un reemplazo gratuito de código abierto para aplicaciones médicas, tales como Medical Manager, Pro Salud, y Misys.

2.6.1.5. Tolven

Tolven se centra en ofrecer los siguientes productos:

- Un sistema electrónico de Registro de Salud Personal (ePHR) que permitirá a los consumidores grabar y compartir de forma selectiva la información de salud acerca de sí mismos y sus seres queridos de una manera segura.
- Un sistema electrónico de registro clínico de la Salud (CEDH), que permite a los proveedores de atención médica para acceder de forma segura la información de salud reunidos a partir de cualquier número de fuentes de confianza en relación con un paciente de una manera estructurada y de fácil acceso.
- Un médico de Informática plataforma que permite a todos los datos de salud que se almacenan y se accede a través de las soluciones ePHR y el CEDH. La plataforma se basa en tecnologías estándar de la industria y modelos de datos.
- Una solución de la Salud de Google Analytics que permite a todos los datos almacenados en la Plataforma Tolven a extraer o analizado con fines estadísticos.

[WWW.016]

2.6.2. Empresas Desarrolladoras

Existe una alta competencia entre las empresas desarrolladoras de software, cada una pretende abarcar la mayor parte del mercado, a través de la personalización de programas que se adapten a las necesidades de la mayoría de las empresas consumidoras.

En el caso de las grandes empresas competidoras que ya tienen sus clientes formados, se encuentran mucho más consolidadas en el mercado, por lo que la rivalidad existente no es tan grande.

- MEDICAL producto de Intermix Soluciones en Internet.
- Programa historia médica producto publicado en mercado libre.
- iHospital y Nemesis-Clinic productos de Ocitel S.A.
- SISHOSP producto de Prosoft.
- Info-Medic Win producto de Gestión Médica Integrada (IGM) – España.
- MedFile.
- Galenus Pro.

2.6.3. Industria de Software en Ecuador

Cuando se habla de las Tecnologías de la Información, más de uno nos habremos impresionado sobre su espectacular impacto en el desarrollo de los países y de las empresas; casos de éxito a nivel mundial son presentados a diario en todos los medios de comunicación y ya no es posible concebir un ambiente en el que la Tecnología no haya generado una influencia directa en su eficiencia. Esta realidad no debe quedarse únicamente en un discurso internacional, todos los presentes somos parte de esta industria y lejos de ser elementos pasivos somos responsables de aplicar estas herramientas en las instituciones y empresas que formamos parte.

El Software es la industria para todas las industrias, somos una herramienta para generar competitividad en todos los sectores, y por qué no, un sector con potencial de desarrollo internacional que genere bienestar para el gran talento humano que existe en el país y que se ha demostrado a través de un sinnúmero de historias de éxito de varias empresas

ecuatorianas.

Es por ello que a través de la Corporación Andina de Fomento CAF, MICIP y Corporación de Promoción de Exportaciones e Inversiones CORPEI, se firma el convenio con la Asociación Ecuatoriana de Software AESOFT, del Proyecto de Mejoramiento de la Competitividad de la cadena del software del Ecuador.

Siendo el objetivo principal del programa, contribuir al cluster y a elevar la competitividad del sector de software ecuatoriano.

Para esto se han definido dos áreas principales: Impulsar la adopción de modelos de producción con estándares internacionales a través del apoyo en la Capacitación, implementación y certificación del modelo CMMi en empresas desarrolladoras de software. Adicionalmente se desarrollarán actividades orientadas al fortalecimiento de la asociatividad de las empresas del Sector y a fortalecer el nivel de los profesionales vinculando el sector empresarial y universitario.

[WWW.017]

2.6.4. La Importancia del sector del software en Ecuador.

Es claro que las tecnologías de la información jugarán un papel protagónico durante el próximo siglo. Ya se encuentran en la base de la economía de muchas naciones desarrolladas y la única manera de competir en el mercado moderno e incorporándolas en las economías de los países en desarrollo.

El mercado tiende a ser cada vez más globalizado. Las fronteras están cayendo. En menos de un minuto, se toma un avión para transportarse a otro extremo del mundo y en este ámbito la información es vital para la competencia. Actualmente la Industria del Software del Ecuador, cuenta con las siguientes cifras:

- Ventas de Software: 62 \$ Millones
- No. de empleos directos fijos: 2.600
- No. de empleos directos a destajo: 633
- No. de empleos indirectos: 3.988

- Aporte fiscal: 21.6 \$ Millones
- Exportaciones: 10.7 \$ Millones

[WWW.017]

El potencial de la industria de software es muy grande a las puertas del siglo XXI. Muchas de las industrias de hoy, compiten gracias a la forma en cómo manejan la información.

Cuentan con programas para la producción de una planta de tal forma que se puedan posponer o evitar inversiones en equipo industrial. La información aumenta la capacidad de la planta, se administra de forma diferente y se obtiene un mayor beneficio.

Es necesario desarrollar, crear, innovar, convenios, alianzas y apoyos que beneficien en buen nivel al sector del software.

[WWW.017]

2.6.5. Perfil del Cliente.

Los usuarios potenciales de este software son:

- Clínicas Médicas que cuenten con un equipo de trabajo amplio.
- Doctores de Centros Médicos o Consultorios particulares que deseen ampliar su mercado.
- Centros Clínicos privados que no cuentan con un avance tecnológico para llevar el historial médico de sus pacientes.
- Instituciones Médicas que deseen que el prestigio y status se forme o mantenga.
- Centros Clínicos que requieran un renombre y aumento de mercado.

Al ser la información de los pacientes compartida por todos los Doctores del Centro Médico, esto les permitirá optimizar la atención que se le brinde al paciente, ya que tendrá acceso a la información en el momento preciso.

La permitirá al Doctor disponer de más tiempo para atender a más pacientes, como resultado de la agilidad que se adquiere tanto al ingresar los datos de las historias clínicas

como para la búsqueda de información.

La mayoría de software médicos del mercado persiguen o brindan beneficios como los siguientes:

- Reducción de tiempo.
- Legibilidad de la información.
- Organización de la información.
- Reducción de papelería.
- Disponibilidad de la información en tiempo real.
- Disminuir la carga de trabajo.
- Seguridad de la información.
- Flexibilidad en el Sistema.
- Ayudar al usuario con el correcto manejo de los recursos.
- Manejar eficientemente el volumen creciente de transacciones.

2.7. ANÁLISIS DE METODOLOGÍAS PARA EL DESARROLLO DE SOFTWARE

Los desarrolladores de software al inicio de un proyecto se preguntan ¿Qué metodología voy a aplicar?, y por supuesto que esta pregunta es indispensable, cómo un arquitecto o un ingeniero civil antes de hacer una casa o cualquier construcción, mientras más grande sea, más indispensable se torna “UN PLANO” en donde se va a apoyar para la ejecución de tal trabajo.

Es lo mismo o parecido a la ejecución del trabajo de este arquitecto, para los desarrolladores de software, el mismo que debería elaborar una especie de plano en el cual apoyarse y seguir el plan de desarrollo para no tener inconvenientes en su camino a la culminación del proyecto.

Pero muchas veces este aspecto no es tomado muy en cuenta, sobre todo cuando se trata de proyectos pequeños de dos o tres meses, pues tal vez no se vea necesario en estos proyectos o no se van a notar los inconvenientes. En estos proyectos normalmente se separan rápidamente el aplicativo en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo.

En cambio si los proyectos a desarrollarse son de mayor envergadura, en este caso deberíamos basarnos en una metodología de desarrollo, por lo que buscaríamos cuál de las que existen sería la más apropiada para nuestro caso. Pero en la realidad no encontramos la más adecuada y terminamos por hacer o diseñar nuestra propia metodología, no podríamos decir que esto está muy mal, pero cualquiera que sea, sea uno existente o inventado por uno mismo debe cumplir con el objetivo propuesto al inicio, para satisfacer al cliente y a uno mismo como diseñador.

A veces diseñamos o planeamos nuestro software de una manera rígida, tratando de tomar en cuenta los requerimientos del cliente, esto muy a menudo ocasiona problemas a la hora que nos pide realizar cambios, pues altera toda nuestra planificación y se hace muy difícil complacer al cliente. Obviamente para evitar estos incidentes debemos haber llegado a un acuerdo formal con el cliente, al inicio del proyecto, de tal manera que cada cambio o modificación no perjudique al desarrollo del mismo, por lo que esto se hace indispensable, y que por lo menos el proyecto culmine de una manera satisfactoria para ambas partes (cliente y desarrollador).

Buscando en el Internet, existen diversas metodologías para realizar un proyecto de software, cada una de ellas muy buenas y que podrían adaptarse a un tipo de proyecto, pero las que nos pareció más interesantes y las más comunes son: RUP, XP y MSF, por lo que vamos a centrar nuestro análisis en estas tres en el siguiente cuadro:

METODOLOGÍA	ENFOQUE	VENTAJAS/DESVENTAJAS
Rational Unified Process (RUP)	<p>La metodología RUP, divide en 4 fases el desarrollo del software:</p> <ul style="list-style-type: none"> • Inicio, Determinar la visión del proyecto. • Elaboración, Determinar la arquitectura óptima. • Construcción, Aquí el objetivo es llevar a obtener la capacidad operacional inicial. • Transmisión, El objetivo es llegar a obtener el "release"⁷ del proyecto. <p>Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en</p>	<p>Ventajas:</p> <ul style="list-style-type: none"> • Útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida. • Ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo o de la adaptabilidad de un sistema operativo. <p>Desventajas:</p> <ul style="list-style-type: none"> • RUP para algunos

⁷ Nueva versión de una aplicación informática.

	reproducir el ciclo de vida en cascada a menor escala.	proyectos es demasiado complejo.
Extreme Programming (XP)	Actualmente, esta metodología se ha convertido en una de las metodologías de desarrollo de software más exitosas y más utilizadas para proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer. Esta metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.	<p>Ventajas:</p> <ul style="list-style-type: none"> • La comunicación, entre los usuarios y los desarrolladores • La simplicidad, al desarrollar y codificar los módulos del sistema • La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales <p>Desventajas:</p> <ul style="list-style-type: none"> • Para algunos proyectos es demasiado ligero
Microsoft Solution Framework (MSF)	<p>MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.</p> <p>Sus características son:</p> <ul style="list-style-type: none"> • Adaptable: es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar. • Escalable: puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más. • Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente. • Tecnología Agnóstica: porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología. <p>Se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.</p>	<p>Ventajas</p> <ul style="list-style-type: none"> • Se puede tener mayor soporte por ser Microsoft y mantenimiento y la mayoría de los usuarios finales están más acostumbrados con este producto. • Sirve para grandes y pequeños proyectos y de cualquier tecnología. <p>Desventajas</p> <ul style="list-style-type: none"> • La principal desventaja es que se torna un trabajo bastante largo, ya que para cada fase se debe documentar profundamente todo lo que se haga, pero no deja de ser un modelo que tiene buenos resultados.

Tabla 2.8: Análisis de Metodologías para el Diseño de Software.

En el análisis del **Cuadro 2.8**, los tres métodos analizados tienen una gran importancia, y son muy interesantes en la aplicabilidad de un determinado tipo de proyecto, pues todas tienen ventajas y desventajas se vuelven notables a la hora de su aplicación.

Tomando en cuenta este análisis, a la hora de elección de la metodología para el diseño del software, vimos conveniente elegir el método RUP, por las siguientes razones:

- Por ser el método más común en nuestro medio.
- RUP era el método que más se adaptaba a nuestro proyecto, por su extensión, pues no era una aplicación pequeña y necesitábamos un plan que se asemeja mucho al mismo.
- También por la documentación existente en el Internet.

No se puede lograr llegar al propósito planteado sin seguir las disciplinas propuestas por este método, denominadas **Disciplinas de Desarrollo**, pues al inicio lo único que se sabe del proyecto a desarrollarse es a lo que se quiere llegar, pero no como llegar.

Lo primero es entender las necesidades del negocio, en este caso la del Subcentro de Salud, RUP lo identifica como INGENIERÍA DE NEGOCIOS, para luego trasladar las necesidades del Subcentro a un modelo de sistema automatizado, conocido por RUP como REQUERIMIENTOS.

Estos requerimientos una vez analizados se trasladan a su vez a la Arquitectura del Software previamente diseñado, esta disciplina conocida como ANÁLISIS Y DISEÑO.

El siguiente paso se relaciona con la creación del Software que se ajusta a la arquitectura del mismo y que cumpla con el comportamiento deseado, también conocido por RUP como IMPLEMENTACIÓN.

Y el último paso, se refiere con asegurarse que el comportamiento requerido sea el correcto y que la mayoría de lo solicitado esté presente en las PRUEBAS.

Además de las anteriores, no se puede culminar con éxito el objetivo planteado sin apearse a las **Disciplinas de Soporte**, las cuales permiten seguir el desarrollo sin inconvenientes.

Estas se refieren a que se debe acostumbrarse a guardar las versiones de software cada tiempo que se piense conveniente a esta disciplina RUP se conoce como CONFIGURACIÓN Y ADMINISTRACIÓN DEL CAMBIO.

Por supuesto que se debe tener una buena administración de los horarios de trabajo y también de los recursos que se emplea conocido como ADMINISTRACIÓN DEL PROYECTO.

Para una buena concentración en el proyecto, se debe seleccionar un ambiente de desarrollo que favorezca esta actividad, es decir un ambiente de trabajo sin interrupciones y con lo necesario para desarrollar nuestro proyecto con normalidad y cumpla los objetivos en las diferentes etapas del desarrollo, conocido esta disciplina como ADMINISTRACIÓN DEL AMBIENTE DE DESARROLLO.

Al final del trabajo se debe sacar un producto, esto se refiere a la salida del proyecto, por lo que haremos todo lo necesario para que este objetivo se cumpla, conocido como DISTRIBUCIÓN.

Todas estas iteraciones anotadas, se debe clasificarlas y ordenarlas según su prioridad, las mismas que se convierten en entregables al cliente, y que benefician al proyecto para su retroalimentación en cada entregable, pues las mismas nos sirven para seguir creciendo el proyecto.

Como toda metodología RUP tiene sus elementos indispensables para el desarrollo del proyecto software, estas son:

- **Las Actividades:** Estos son los procesos que se llegan a determinar en cada iteración.
- **Los Trabajadores:** Es decir el recurso humano, vienen a ser las personas o entes involucrados en cada proceso, y
- **Los Artefactos:** Es decir un documento, un modelo, o un elemento de modelo.

En conclusión se puede decir que lo más importante antes de elegir la metodología para la implementación del software, es determinar el alcance que tendrá y luego de ahí ver cuál es la que más se acomoda en la aplicación a desarrollar.

CAPÍTULO 3

DISEÑO DE LA SOLUCIÓN



- 3.1. Inicio
- 3.2. Elaboración
- 3.3. Alcance del Sistema

3.1. INICIO

3.1.1. Modelado Actual de Actividades.

3.1.1.1. Proceso de Atención al Paciente.

El Subcentro de Salud de San Antonio de Ibarra en los últimos años ha llegado a tener problemas en brindar una atención ágil a los usuarios, debido a que hay mucha aglomeración de pacientes que quieren hacerse atender, muchas veces hay que madrugar a las 3 o 4 de la mañana para alcanzar turnos para ese día, esto ha ocasionado molestias en la gente ya que a veces se va ya a las 8 de la mañana y se encuentra con la ingrata noticia que se acabaron los turnos para ese día.

Otro de los grandes problemas es que si el paciente no sabe su número de Historia clínica, no puede reservar su turno, pues las enfermeras de turno deben saber este número para poder reservarle el turno, este es otra de las molestias con las que se encuentra el usuario del Subcentro de Salud, pues para cada paciente o usuario frecuente tiene una carpeta donde se registran sus datos, las mismas que están guardadas en un anaquel ordenadas según su historia clínica.

También el tiempo que tarda al buscar las historias de cada paciente, hace que la cola de espera para cada paciente no avance rápido, haciendo esperar al mismo un buen rato, perdiendo un tiempo que se pudiera invertir en una atención más ágil a los pacientes que están esperando su turno en la sala de espera.

Actualmente, han incorporado más profesionales médicos para atender a los pacientes de este Subcentro, por lo que ha ayudado en algo su atención, pues ahora ya se han eliminado las desesperantes madrugadas a las que estaban acostumbrados algunos de los pacientes, pero todavía no en su totalidad, este incremento de profesionales agiliza bastante la atención al paciente, pero no en su totalidad, pues todavía se sigue manteniendo la reserva de turnos como se lo hacía antes de incrementar más médicos, ya que al pedir el turno la enfermera tiene que ir a la gran montaña de carpetas a buscar el número de historia clínica, perdiendo un valioso tiempo y desesperando aún más al paciente, pero este es el caso de que si se sabe el número de historia clínica, en el caso de que no supiera su número de historia, no será atendido o tiene que venir en otro rato que haya más tiempo para dedicarle buscando ese número según su apellido, pues para las profesionales de la salud les resulta bastante tedioso el estar buscando por el apellido

las historias, en el siguiente gráfico se muestra el proceso de atención en el Subcentro de Salud de San Antonio de Ibarra:

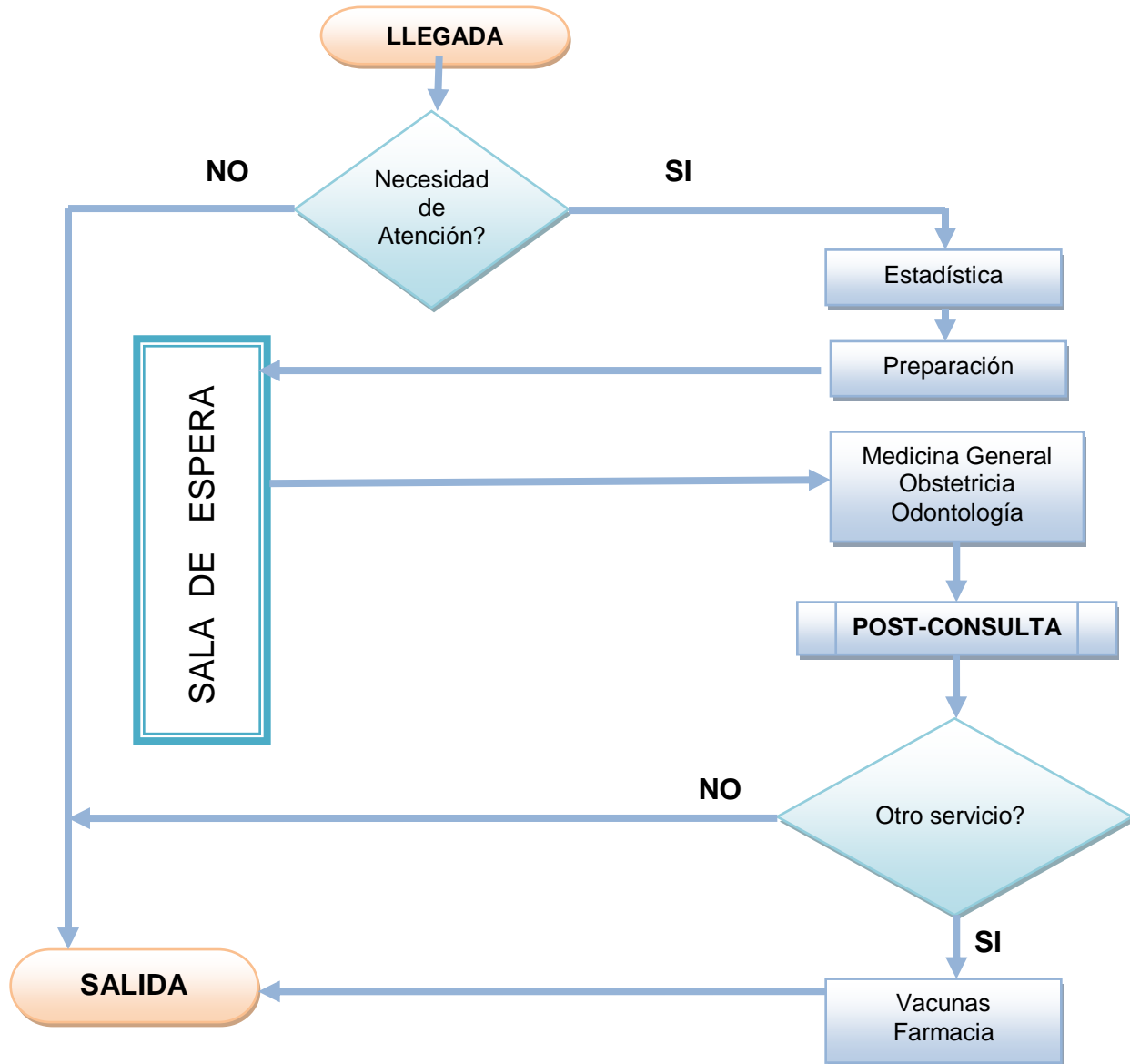


Grafico 3.1.: Proceso de atención al usuario.

Fuente: Subcentro de Salud San Antonio

1. Ingreso - llegada del usuario al Subcentro, si requiere atención va al paso 2, en caso contrario al paso 8.
2. Ventanilla de ESTADÍSTICA Atención en estadística para obtención de Turno e identificación de Historia Clínica o referencia.
3. PREPARACIÓN.- Atención al paciente, toma signos vitales.
4. SALA DE ESPERA.- para esperar el turno de atención según la consulta requerida

5. CONSULTORIO MÉDICO.- Según el servicio requerido
6. POST CONSULTA.- Farmacia, refuerzo o indicaciones médicas, entrega de medicación, si no requiere otro servicio ir al paso 8.
7. Asistencia a vacunas o farmacia, según sea el caso
8. Salida

En **el Gráfico 3.1.**, se muestra los 8 pasos necesarios para poder ser atendido antes de retirarse del Subcentro, se ha identificado los mayores problemas en el paso 2, pues aquí se acumula bastante los pacientes en espera de que le den un turno, estos pasos se explicarán más detenidamente:

Cuando el usuario llega al Subcentro, puede ser que requiera un servicio del Subcentro o solo fue a realizar una pregunta rápida, en caso que requiera uno de los servicios médicos tiene que colocarse en la cola que están haciendo los demás usuarios en la ventanilla de estadística, pero en caso que no hay nadie en la cola, simplemente debe ir a la ventanilla y dar su número de Historia para que una de las enfermeras le den buscando su carpeta, en donde aún mantienen una forma anticuada, pues la enfermera de turno se dirige a un anaquel en donde debe buscar de entre aproximadamente 20.000 carpetas la que necesite, aquí es donde se invierte un buen tiempo, pero por la rutina diaria al que ya están acostumbrados, no se demoran mucho.

Una vez que ya tenga asignado el turno con el respectivo médico profesional, tiene que esperar en la Sala de Espera hasta que le llamen a Preparación, aquí le llamarán según el turno que le asignaron, y en caso de que no se encuentra el paciente le volverán a llamar más luego pero más luego.

Cuando ya le llamaron tiene que dirigirse a Preparación, y es en donde le toman los signos vitales, la temperatura, la presión, el peso y estatura, y todos estos datos actualizan a su carpeta.

Después de que haya pasado por Preparación está listo para que el médico asignado le atienda, pero antes tiene que dirigirse nuevamente a la Sala de Espera para que le llame el médico asignado según el número de su turno.

Cuando ya fue atendido por el médico, debe dirigirse a PostConsulta, esto es si es que necesita alguna inyección, o indicaciones adicionales que hay que si son necesarias el

mismo médico o las enfermeras le dan, o tal vez sea de retirar alguna receta de farmacia.

En caso de que no sea necesario de ninguna indicación adicional, solo tendrá que entregar la carpeta en farmacia o Estadística, para luego retirarse.

En resumidas cuentas las que vimos son todas las actividades que se realizan en el Subcentro, a donde aproximadamente 120 personas diarias acuden a ser atendidas, por lo que el servicio de atención en Estadística especialmente debe agilizar su atención para atender a los usuarios en una forma más rápida.

De acuerdo a la Información de los usuarios sobre la calidad de la atención, el tiempo de espera inicialmente es de 3 a 5 horas como promedio, el lugar en que se demora más es en sala de espera y en estadística.

3.1.1.2. Proceso de Creación y Modificación de Historias Clínicas.

Para la creación, modificación de Historias clínicas en esta institución pública el usuario que va a ventanilla y es un usuario por primera vez que llega a este centro de atención médica, debe crearse y dar sus datos, pero este trámite debe hacerse cuando sus funcionarios no estén ocupados asignando turnos a los pacientes, en caso de que estén en esta labor le dirán que regrese más luego o tendrá que esperar, pues las enfermeras deben atender en la asignación de turnos primero, esto quiere decir, a los pacientes que ya tienen historias clínicas ya creadas.

Es lo mismo en el caso de que necesitar modificar o actualizar sus datos en su historia, pues esta labor debe hacerse cuando hay tiempo suficiente para buscar su historia y proceder a actualizarla.

Todo este proceso lo realizan en ESTADISTICA, en donde mantienen todas las carpetas de las historias clínicas, las mismas que están guardadas en un anaquel ordenadas por orden alfabético, expuestas a pérdidas o en el caso de un incendio convertirse en cenizas, por lo que sería fatal para los usuarios y el Subcentro.

En la actualidad ya se ha empezado a crear archivos hechos en Excel, en donde se guardan el N° de Historia Clínica, Nombres y Apellidos de los usuarios, este proceso está avanzado en más de un 50% del total de Historias Clínicas, pero vemos que faltan los

demás datos de una historia y para guardar en el mismo archivo de Excel se les hace complicado, aunque no imposible. Este archivo últimamente ha servido de gran ayuda el proceso de atención, pero aún no se encuentran satisfechos, pues para averiguar los demás datos como es el teléfono, dirección, cédula, etc., se necesita acudir y buscar en el anaquel en donde encontramos una gran montaña de carpetas.

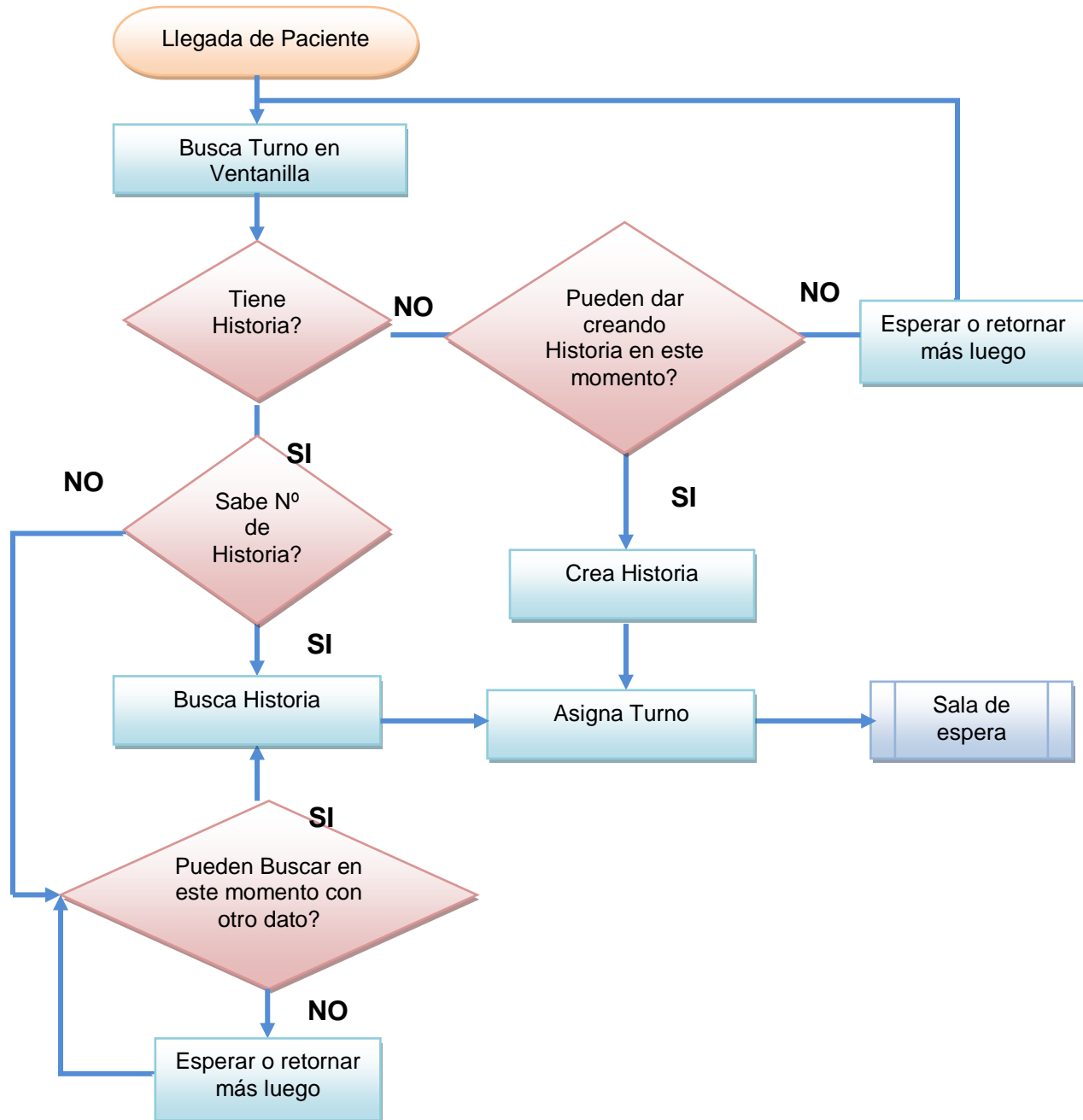


Gráfico 3.2.: Proceso de adquisición de Turnos y/o creación (modificación) de Historias.

Fuente: Subcentro de Salud San Antonio

Nótese que hemos supuesto el caso de que el usuario necesita ser atendido en este centro de salud, no hemos puesto el caso donde los pacientes en el caso de no querer esperar solo se retiran y vuelven otro día o simplemente no vuelven, es por eso que no hemos puesto la “salida”, y además, después de haber hecho todo este trámite tienen que dirigirse a la SALA DE ESPERA, de donde le llamarán a la Preparación o al médico como se muestra en el **Gráfico 3.1**.

Según las profesionales enfermeras que se encuentran laborando actualmente en el Subcentro de Salud San Antonio, lo más urgente a atender es ordenar las historias clínicas para poder agilizar el trabajo de búsquedas, creación y modificación de historias, pues en este momento no les interesa aún los demás departamentos.

3.1.1.3. Proceso de Registro de Medicamentos en farmacia.

Analizando el trabajo que realiza en Farmacia de este Subcentro, vemos que también necesitan una pequeña ayuda, para poder ordenar las entradas y salidas de medicamentos, pues de igual manera esta información se mantiene en carpetas en donde tienen formularios para archivar estas actividades, pero en realidad aquí no entra una gran diversidad de medicamento, solo lo básico para atender afecciones comunes, en su mayoría las profesionales de la salud ya se memorizan estos fármacos debido a la rutina diaria.

El proceso de registro de medicamentos se los realiza cada mes, pues cada mes reciben nuevos medicamentos para poder equipar stock de farmacia, el proceso de registro de medicamentos en forma mensual lo resumimos en el siguiente flujo de trabajo:

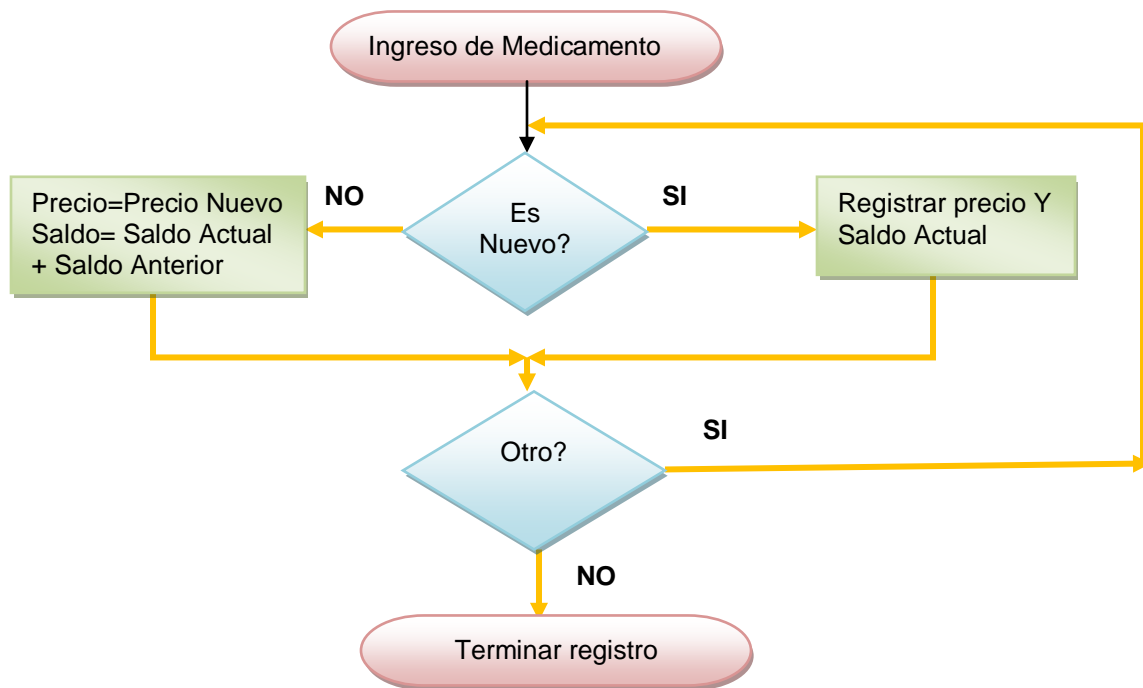


Figura 3.3: Proceso de Registro mensual de medicamento.

Fuente: Diseño Propio

En este proceso lo que se toma mucho en cuenta es en cuanto se refiere a precios y a los saldos. Los precios de los medicamentos pueden variar de mes a mes, por lo que se toma en cuenta este aspecto, pues si un medicamento ya viene con un nuevo precio, inmediatamente se los actualiza a los demás productos del saldo anterior, en el caso de ser un producto nuevo que el gobierno envía, solo se registra el precio y el saldo actual, es decir la cantidad de unidades que ha enviado.

En cuanto al Saldo, esto se refiere a las unidades de envases, cajitas de tabletas que se reciben. El Saldo Anterior se refiere al número de unidades que ha sobrado del mes anterior y que todavía existe en stock, y el Saldo Actual se refiere al número de unidades que ingresan en el mes de registro (este mes).

NOTA: Nótese que en Farmacia del Subcentro de Salud San Antonio no se está vendiendo los medicamentos que envía el Gobierno ya que actualmente estos medicamentos se los regala a los pacientes, pero de todas maneras se debe registrar los precios, para poder rendir cuentas si esto fuera necesario.

En cambio el proceso de registro de salidas de medicamentos diarios se puede resumir en el siguiente gráfico:

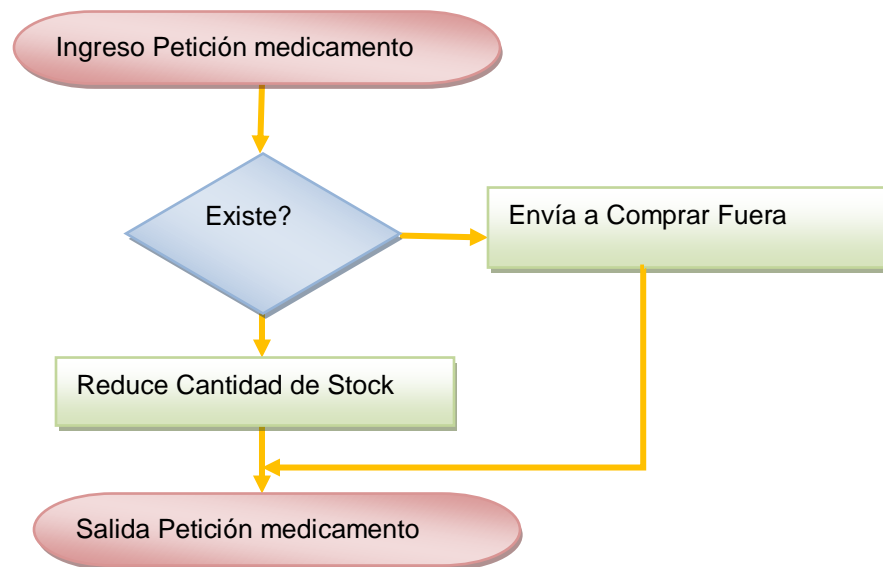


Figura 3.4.: Proceso de Registro diario de Medicamento en Farmacia.

Fuente: Diseño Propio

Hay que tomar en cuenta que el registro diarios de medicamentos solo se refiere a la salida del medicamento, aquí no hay entradas, esto se lo hace en el registro mensual (Figura 3.4). También quisiéramos hacer notar que este proceso se lo hace para cada medicamento solicitado por el usuario, es decir el medicamento recetado por el doctor quien lo atendió en la consulta.

NOTA: Hay que tomar en cuenta que los medicamentos que salen de stock de farmacia del Subcentro San Antonio, son exclusivamente para los pacientes que se hicieron atender en esta dependencia médica; mas no se encuentra a la venta para personas de fuera, pues para ir a retirar los medicamentos se debe ir con la carpeta y sus debidas recetas, pues el paciente o representante del paciente debe firmar en la parte posterior de la receta como constancia de haber recibido.

3.1.1.4. Proceso de Creación y Modificación de Historias Clínicas.

En cuanto se refiere a los datos de los médicos que están atendiendo en esta dependencia, pues no llevan ningún registro ordenado de los mismos, pues no hay un historial de médicos que atienden o estuvieron atendiendo aquí. Tal vez fue un poco de descuido de parte de las personas que trabajan en la Institución o porque no consideraban mantener su propio registro de médicos que prestan su servicio en la Institución que no llevan un registro ordenado de estos datos.

3.1.2. Análisis de Requerimientos.

3.1.2.1. Introducción.

Como ya se analizó anteriormente con respecto a la forma de trabajar en la actualidad del Subcentro de Salud San Antonio, ya se identifican los problemas más urgentes en el Subcentro en cuanto al tratamiento de información y de esta manera poder identificar los requerimientos del sistema actual de trabajo en esta dependencia pública.

El modelo actual del proceso de trabajo, es un modelo que se ha mantenido un buen tiempo en la vida del Subcentro y que en sus inicios no tenía ningún problema, pues no habían muchas historias clínicas y se atendía normalmente sin ninguna queja al respecto, pero en los últimos años, el número de historias clínicas ya han superado los 20.000, por lo que se atienden a un promedio de 120 pacientes diarios, haciendo que este modelo de atención colapse. En realidad, este modelo no es el problema sino una parte de este modelo, porque su estructura es óptima y se adapta al Subcentro según su capacidad física.

Un sistema que brinde un tratamiento ordenado a la información de pacientes ayudaría en gran medida la atención a los usuarios, pues de esta manera se agiliza la búsqueda, modificación y creación de historias clínicas de los usuarios.

Implementando este sistema, toda la información de historias clínicas ya estarán ordenadas en la base de datos, por lo que mediante el sistema se puede implementar varias formas de búsqueda, sin esperar que el paciente se memorice o traiga anotado su número de historia clínica, lo que muchas veces por el apuro o falta de conocimiento del proceso de atención en el Subcentro no lo llevan o ven innecesario memorizarse, o simplemente lo pierden pensando que tal vez no lo van a volver a utilizar. Todo esto hace que el Subcentro no brinde una atención óptima.

Con la implementación del sistema el paciente ya no tendrá que memoriza su número de historia, o llevarlo anotado para poder pedir un turno de atención en este Subcentro de Salud, pues en la actualidad si no lleva este dato lo dejarán para los últimos turnos o le harán esperar un buen tiempo, causando molestias de parte de los usuarios. Para enmendar este problema al sistema podemos implementar distintos modos de búsqueda:

- Por número de historia clínica
- Por cédula de identidad
- Por Apellidos
- Por Nombres
- Por Dirección, etc.
- O una combinación de las anteriores

Es decir, el paciente debe dar cualquiera de los datos anteriores o la combinación de ellos para ser atendido, por lo que tiene mayor probabilidad de ser atendido, y se demorará el mismo tiempo para todos los usuarios. De esta forma tanto el paciente como la profesional de la salud que se encuentra atendiendo en ventanilla van a estar satisfechos.

Lo mismo pasaría en la creación de nuevas historias clínicas ya que este trabajo se lo haría en un menor tiempo posible, así como en el caso de la modificación de las mismas.

Se debe hacer notar que se pone más énfasis con respecto a la automatización del tratamiento de información, pues este es el problema al que se enfrentan a diario las personas que atienden en ventanilla, ya que a diario existen pacientes con urgencia de atención médica, y debido a ello existe lentitud de búsquedas de información, creación, modificación de las historias clínicas haciendo que la cola de espera no avance.

Pero además de solucionar el tratamiento de información de historias clínicas, se puede implementar otros módulos más, por ejemplo:

- Tratamiento a las entradas y salidas de medicamentos.
- Tratamiento a la información de médicos.
- Tratamiento a la información de turnos asignados.
- Controlar acceso al sistema con Creación de usuarios.
- Dar privilegios a los usuarios.
- Sacar copias de las Bases de Datos (Backup)
- Restaurar la Base de Datos (Restore)

Por supuesto que se puede incrementar algunos módulos más según los requerimientos que se presenten en el quehacer diario del Subcentro de Salud.

3.1.2.2. Requerimientos de Usuarios del Sistema

3.1.2.2.1. Requerimientos de Usuarios-Pacientes.

La atención al paciente en cualquier entidad pública de servicio debe ser lo más ágil posible, para que no haya aglomeraciones de personas, y para que nosotros como usuarios no tengamos ningún problema en la atención que nos brinde y que nuestra salud en este caso sea atendida lo más pronto posible, tomando en cuenta estos aspectos, los requerimientos de los pacientes serían:

- Atención ágil en la asignación de turnos.
- Atención rápida en la creación de Nuevas Historias.
- Atención ágil en la modificación de datos.
- Recepción rápida de medicamentos.
- A todo esto, recibir un buen trato.

Se puede ver que si existe todo esto conlleva a recibir un buen trato de parte y parte es decir entre paciente y profesional de salud.

3.1.2.2.1. Requerimientos de Usuarios-Profesionales de la Salud.

La necesidad de contar con una herramienta más de trabajo en su rutina diaria de parte del profesional de salud como es un sistema informático que brinde un tratamiento automatizado a las historias clínicas es grande, pues con esta herramienta ya no estarían tan apretados de tiempo, pues de parte del profesional de la salud sus requerimientos podremos resumirlo así:

- Contar con una herramienta que brinde un tratamiento automatizado a las historias clínicas de los pacientes.
- Que la misma herramienta ayude a la consulta de medicamentos en stock de farmacia.
- Contar con la misma herramienta que ayude a guardar información personal de los médicos que pasan por este Subcentro de Salud.
- Que los datos estén guardados de una manera segura, y que el mismo sistema pueda sacar copias de seguridad o restaurar copias si el caso requiera.

Todo esto se resume a que se quiere contar con un sistema informático que sirva como una herramienta de apoyo para agilizar la ejecución de su trabajo diario.

3.1.2.2. Requerimientos para la Instalación y Uso del Sistema Informático.

3.1.2.2.1. Requerimientos de Hardware.

La Institución no cuenta con recursos suficientes como para adquirir equipos de gran valor, pues la asignación de recursos por parte del Gobierno no alcanza como para hacer un centro de cómputo de primera, pero afortunadamente podemos contar con herramientas de programación libres y que no necesitan mucho recurso para su funcionamiento, es así que los requerimientos de Hardware para que el Sistema Funcione puede ser:

- Computadora Pentium IV o superiores, para un funcionamiento óptimo.
- Hub o Switch para implementar una red interna, para que el mismo sistema pueda funcionar como cliente en varios equipos dentro del Subcentro de Salud.
- En caso de una red inalámbrica, un Router con Wireles para su implementación sin cables.

3.1.2.2.2. Requerimientos de Software.

En cuanto al Software que se requiere tener instalado para que el sistema funcione, de igual manera las herramientas a utilizar son flexibles, libres y no consumen mucho recurso en una computadora, como los siguientes requerimientos.

- Firefox 2.0 en adelante o Internet Explorer 6.0 en adelante
- Paquete de Aplicaciones WampServer 2.0i, en donde contiene:
 - PHP 5.3.3
 - Apache 2.2.17
 - Mysql 5.1.53 (versión 64 bits)
 - Mysql 5.5.8 (versión 32 bits)
 - PhpMyadmin 3.2.0.1
- Adobe Reader desde la versión 5 en adelante.

3.2. ELABORACIÓN DE LA SOLUCIÓN

3.2.1.- Flujo de Trabajo de Requerimientos.

En la sección anterior se resumió los diferentes requerimientos que actualmente se presenta en el Subcentro de Salud de San Antonio. Y como ya se vio en el Gráfico 3.1, este es el modelo actual a breves rasgos, ahora se va a diseñar lo mismo pero incrementando la solución informática.

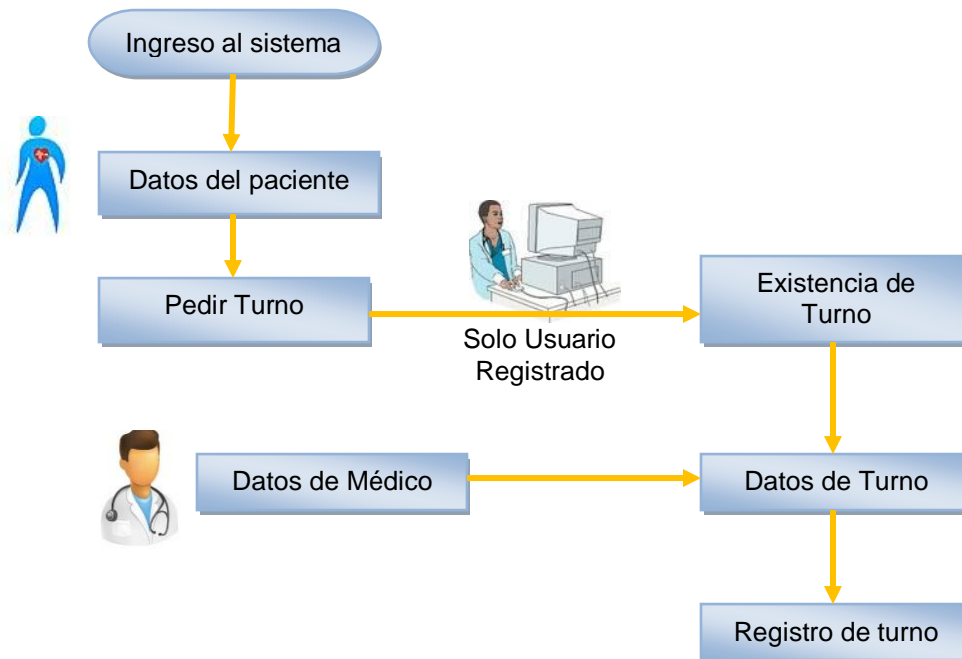


Gráfico 3.5.: Proceso de Trabajo en el Subcentro de Salud, incrementando el Sistema.

Fuente: Diseño Propio

En este proceso de trabajo ya interviene el Sistema, con la cual el usuario no esperará mucho para que ya consiga su turno.

Aquí se va a ver un poco más detenidamente el proceso que se realiza en el Gráfico 3.6. Pues el usuario cuando entra a pedir su turno a ventanilla (Departamento de ESTADISTICA en el Subcentro), tendrá que proveer sus datos los cuales no solo puede ser su número de Historia como anteriormente lo hacía, este en el caso que desconozca este dato tendrá que dar sus apellidos o nombres o dirección o su número de cédula, o si es que se acordase el número de historia, o a su vez una combinación de estos datos, es decir como por ejemplo: nombres y apellidos, o solo un nombre y dos apellidos o viceversa, o cédula y nombres y apellidos, etc. Esto con la finalidad de hacer más certera

su búsqueda, de esta manera se eliminan los rechazos en el momento de conseguir un turno por lo que se brinda una ayuda tanto al profesional de la salud como al paciente.

En el momento que ya se da el turno se registrará también los datos del médico con el cual se va a atender, dependiendo del tipo de atención que requiera.

3.2.2.- Análisis del Modelo óptimo de Trabajo.

Ahora se analizará más detenidamente el Modelo que debería ser la más óptima, por supuesto que el principal actor va a ser el Sistema.

El requerimiento más urgente es el del tratamiento automatizado de la información de historias clínicas de los pacientes que llegan al Subcentro a hacerse atender:

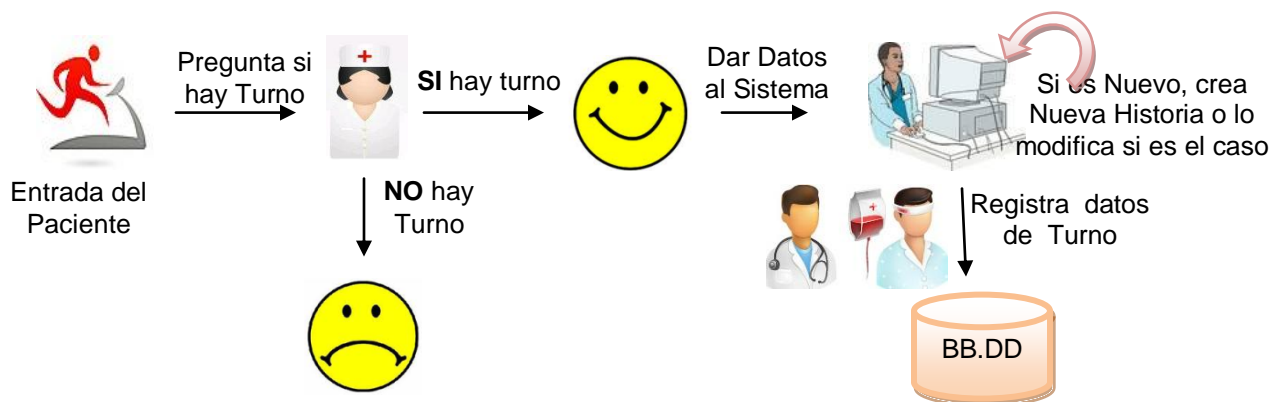


Gráfico 3.6.: Proceso de Petición de turno contando con el sistema.

Fuente: Diseño Propio

En el Gráfico anterior se muestra como llega el usuario al Subcentro y lo primero que debe hacer es llegar a ventanilla (ESTADISTICA), y averiguar si es que todavía hay turnos para poder ser atendido, entonces la enfermera le confirmará si es que hay o no. En el caso de que no haya turnos lamentablemente tendrá que retirarse, pero en el caso de que haya turnos tendrá que dar sus datos para que lo busque en el sistema, estos datos pueden ser: cédula o nombres o apellidos o número de historia o en el último caso la dirección o una combinación de ellos, para ver si es que el paciente ya está registrado en el sistema, como ya se dijo anteriormente.

Una vez que ya pasó por todo el proceso anterior, se registra el número de turno y los datos del paciente y del médico que va a atenderlo. Pero puede haber el caso que el usuario también quiere que le modifique algún dato de su historia, no perderá su turno y lo modificará en ese momento. Otro caso que puede ocurrir es que el usuario sea nuevo, en

ese caso también no perderá su turno ya que podrá crearlo en ese mismo momento en una forma rápida.

Cualquiera que sea el caso, el turno será registrado como ya se indicó anteriormente, con los datos del paciente y el médico.

De esta manera se evitará el problema que siempre tuvieron; el que si no llevaban su número de historia, no era atendido o le dejaban al último, ya que a veces resultaba injusto porque el usuario se sacrificaba madrugando todo para recibir esta ingrata noticia.

Cuando ya logran obtener el turno, solo tendrán que esperar para que les realicen la toma de signos vitales, peso, etc., a esto se llama PREPARACIÓN, antes de que el médico le atienda.

Ahora se analizará también el caso de registro de medicinas (Farmacia), en donde actualmente tienen formularios de registro mensual y diario de medicamentos en los que registran las salidas de medicamentos (registro diario), y la entrada de medicamentos (registro mensual). Tal vez resulte que en este caso no ayude el sistema, tanto como en el tratamiento de información de historias clínicas, pues las personas que laboran en el Subcentro ya están acostumbradas en manejar estas fichas, pero a la larga ya podrán apreciar las bondades que presta este sistema también en este caso, esto se resume en el siguiente gráfico:

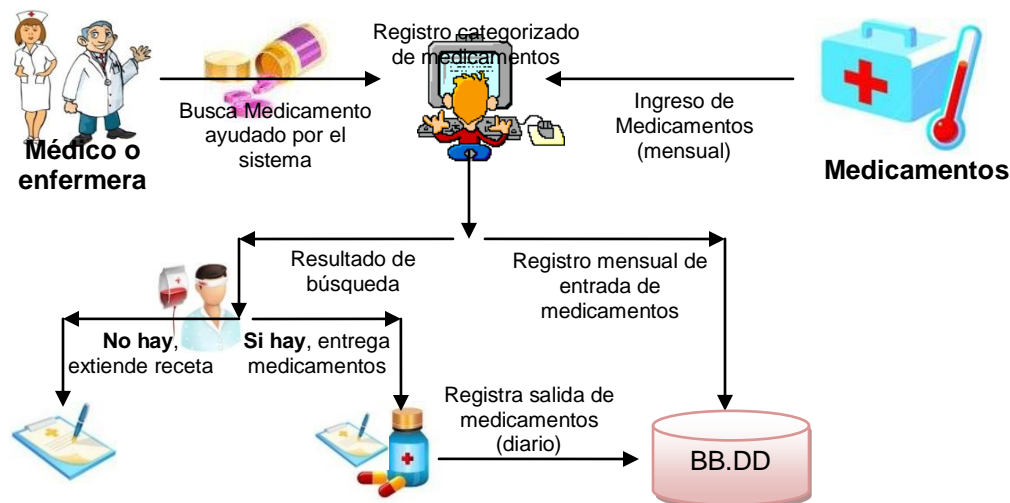


Gráfico 3.7.: Proceso de registro de salida de medicamentos (diario), y entrada de medicamentos (mensual).

Fuente: Diseño Propio

Los principales actores en este proceso son el profesional de la salud (médico o enfermera), el mismo que necesita averiguar la existencia de un respectivo medicamento en farmacia del Subcentro o algún medicamento genérico que pueda recetar, esto con el fin de que pueda llevarse el mismo desde el Subcentro y no tenga que ir a conseguir afuera, ya que los medicamentos aquí no se cobra en su mayoría, y muchas veces los pacientes que acuden a esta casa de salud son de muy pocos recursos económicos, el sistema arrojará un resultado , el mismo que puede ser el caso de que SI HAY, por lo que el paciente llevará los medicamentos y sus respectivas indicaciones para suministrarse, o puede ser el caso de que NO HAY, en este caso tendrá que llevarse la receta e indicaciones para que pueda conseguirse en otro lado. El proceso de salida de medicamentos será registrado a la base de datos.

El otro actor principal es los medicamentos que ingresan mensualmente, los mismos que serán registrados en la base de datos, estos normalmente lo hacen cada mes, pues, son medicamentos que reciben del estado, los mismos que son gratuitos en la actualidad, pero de todas maneras se registran sus precios, esto con el fin de legalizar su entrada y pueda que en algún momento realizar un reporte de los mismos para cualquier fin.

Todo este proceso, ayudaría al Subcentro a tener organizado los datos de medicinas existentes en stock de farmacia y en una forma categorizada de los mismos, es decir, podríamos ordenar medicamentos por categorías, por lo que una enfermera inexperta, o un practicante que no tiene mucha experiencia en medicamentos lo pueda encontrar de una forma rápida, por ejemplo así:

ANTIBIÓTICOS:

- Tetraciclina,
- Doxicilina,
- Estreptomina.

ANTIDEPRESIVOS:

- Litio,
- Paroxetina,
- Fluoxetina,
- Sertralina.

etc.....

El sistema dará la oportunidad de crear sus propias categorías o eliminarlas, esto hará que cualquier usuario con o sin experiencia pueda encontrar con mayor rapidez el medicamento a buscar.

Otro de los requerimientos que se pudo observar en el Subcentro de Salud San Antonio, se trata sobre el registro de médicos que pasan por esta casa de salud, pues hasta ahora no le han tomado mucha importancia a estos datos, por lo que en los archivos del Subcentro no se encuentra este tipo de registros.

En el sistema resulta también importante incrementar este módulo, pues se piensa que resulta de gran importancia mantener el historial de médicos que han pasado por el Subcentro, y de esta manera se mantiene un historial de médicos, en donde se registra sus datos personales básicos y su especialidad, de la siguiente manera:



Gráfico 3.8.: Proceso de registro de datos personales de Médicos.

Fuente: Diseño Propio

Los casos analizados anteriormente son los puntos que son los más urgentes a atender. Pero podemos implementar módulos adicionales según nuevos requerimientos que se presenten, así como, implementar un módulo de reportes, de registro de turnos, y un módulo para crear o recuperar backups de la Base de Datos.

3.2.3. Diseño

Para el Diseño de Producto se va a utilizar herramientas en su mayoría “Libres”, para no tener problema con las licencias, las ventajas y características de este tipo de herramientas ya se enunciaron en el capítulo 2 con bastante amplitud, en esta ocasión se va a indicar las bondades de las herramientas seleccionadas para realizar el producto.

3.2.3.1. Wamp Server 2i

WampServer es un entorno de desarrollo web bajo Windows, permite crear aplicaciones web con Apache, PHP y MySQL como base de datos. Además incluye PHPMyAdmin para facilitar el manejo de las bases de datos.

[WWW.018]

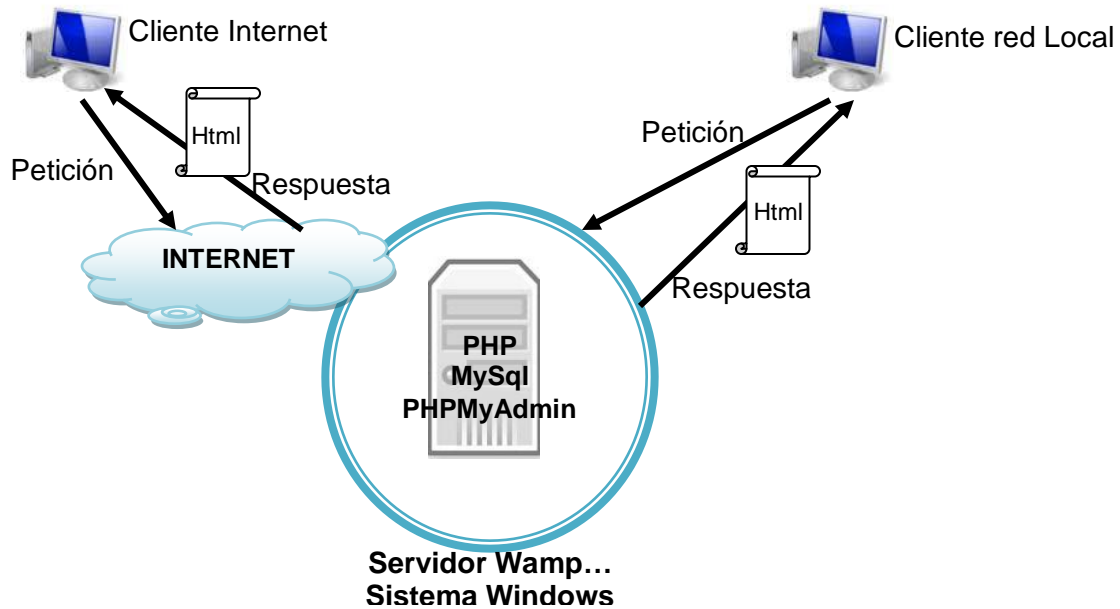




Gráfico 3.9.: Arquitectura de servidor Wamp.

Fuente: Diseño Propio

WampServer se instala con mucha facilidad y su utilización es muy intuitiva. Así, se podría configurar el servidor incluso sin tocar ni un solo fichero de configuración.

Además este paquete permite reproducir nuestro servidor de producción en forma local, pues el Subcentro de Salud desea mantener el Sistema solo a nivel Local funcionando el sistema en una intranet dentro del Subcentro.

Cuando se instala, y el servidor está trabajando se puede notar a la izquierda de nuestra pantalla un ícono como el siguiente, pueden estar en diferentes estados, así:

- Servidor Desactivado 
- Servidor Activado 

En cualquiera de estos estados localmente trabaja normalmente, solo que para otras máquinas no podrá trabajar con normalidad, pues se restringe algunas cosas, como es la de descargar información (en caso de estar servidor desactivado), etc.

Al hacer clic izquierdo en el icono cualquiera de los dos anteriores, nos aparecerá un menú desplegable de administración, en donde podemos configurar los servicios de la aplicación.

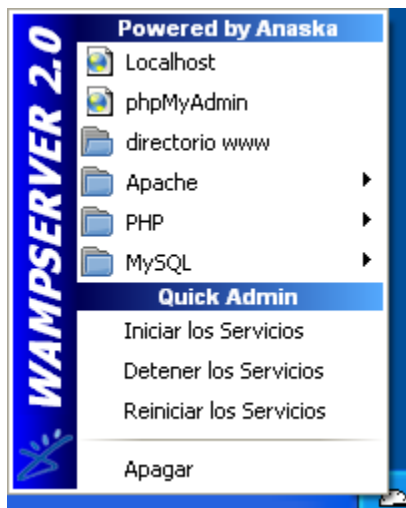


Gráfico 3.10.: Menú de administración de WampServer.

Fuente: Pantalla capturada

En este menú se puede encontrar las siguientes herramientas:

Localhost: Permite acceder con el navegador a los contenidos del servidor web. Es lo mismo que abrir un navegador e ir a la dirección <http://127.0.0.1> que es equivalente a ir a <http://localhost>.

phpMyAdmin: Con esta opción permite acceder con el navegador al administrador de bases de datos phpMyAdmin. Es equivalente a abrir un navegador e ir a la dirección <http://localhost/phpmyadmin>.

Directorio www: Se puede acceder con el explorador de archivos al DocumentRoot, carpeta `c:\wamp\www`, para copiar contenidos. En esta carpeta es donde debemos copiar las aplicaciones como en este caso el Sistema del Subcentro de Salud.

Apache: Para acceder a la configuración del Servidor Apache.

PHP: Para acceder a la configuración de PHP.

MySQL: Para acceder a la configuración de MySQL.

Iniciar los servicios/Detener los servicios: Esta opción permite iniciar o detener Apache y MySQL

Reiniciar los servicios: Esta herramienta permite reiniciar Apache y MySQL. Útil si hemos realizado alguna modificación en la configuración y queremos aplicar los cambios.

Existe una aplicación similar llamada XAMP, pero esta se instala en Linux, es la única diferencia, la forma de configurar y más no difieren en mucho.

Cabe señalar que esta herramienta es libre, es decir no se paga ninguna licencia por su utilización, razón suficiente como para decidirse por ella.

Para poder administrar las Bases de Datos en forma visual se puede acceder desde este menú de administración de Wamperver (Grafico 3.10), así:

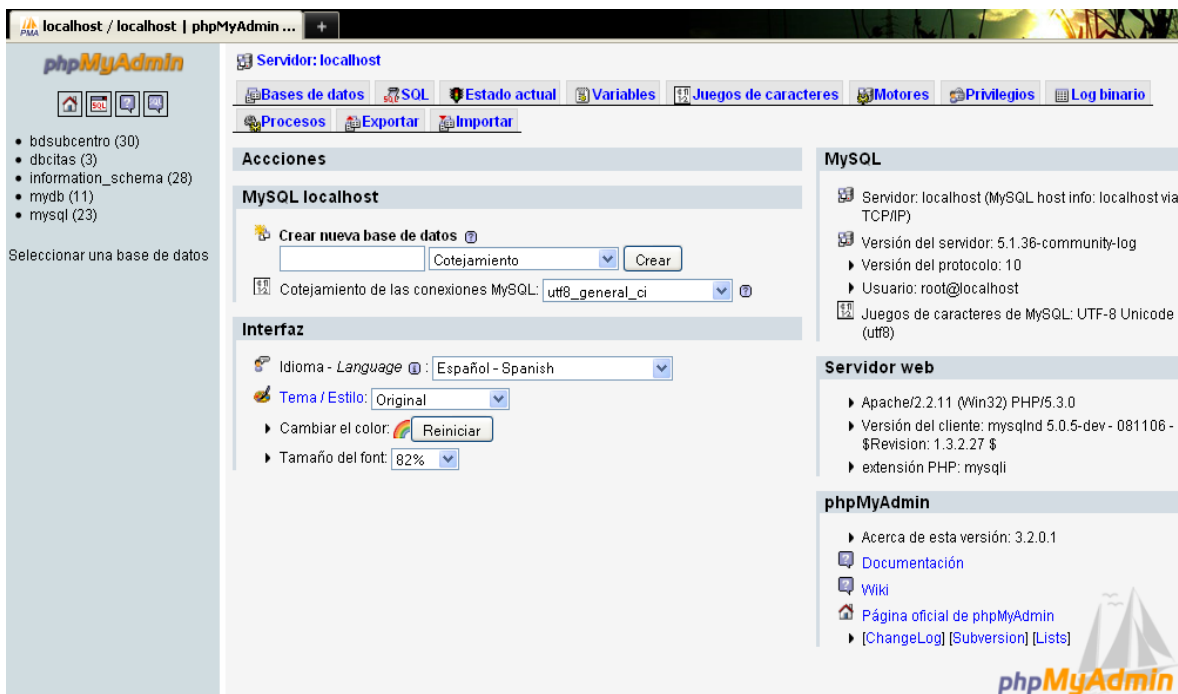


Gráfico 3.11.: Ventana de phpMyAdmin.

Fuente: Pantalla capturada

Aquí se puede crear bases de datos, eliminarlos, o crear, modificar o eliminar registros de las bases de datos, es decir se puede administrar todas las bases de datos existentes y en forma visual.

3.2.3.2. Framework Symfony 1.4.

3.2.3.2.1. Conociendo Symfony

Una vez que ya se tenga instalado el paquete de aplicaciones WampServer, ya se tiene también Apache, MySQL y PHP, por lo que como paso siguiente sería conseguir un Framework que facilita la vida a un programador de PHP, por lo que la decisión fue por Symfony, pues es un paquete libre y nuevo, para conocer más sobre esta aplicación se muestra un poco de historia y se mostrará un poco de sus características.

La primera versión de Symfony fue publicada en Octubre de 2005 por Fabien Potencier, fundador del proyecto, es el presidente de Sensio (<http://www.sensio.com/>), una empresa francesa de desarrollo de aplicaciones web conocida por sus innovaciones en este campo. En el año 2003, Fabien realizó una investigación sobre las herramientas de software libre disponibles para el desarrollo de aplicaciones web con PHP. Fabien llegó a la conclusión de que no existía ninguna herramienta con esas características. Después del lanzamiento de la versión 5 de PHP, decidió que las herramientas disponibles habían alcanzado un grado de madurez suficiente como para integrarlas en un framework completo. Fabien empleó un año entero para desarrollar el núcleo de Symfony, basando su trabajo en el framework Mojavi (que también era un framework que seguía el funcionamiento MVC), en la herramienta Propel para el mapeo de objetos a bases de datos (conocido como ORM, de "*object-relational mapping*") y en los *helpers* empleados por Ruby on Rails en sus plantillas [WWW.019].

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de

negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, 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 [WWW.020].

Es decir que Symfony es un conjunto de librerías realizadas con el fin de reutilizarlas o hacer el llamado en un punto cualquiera de la aplicación que se está construyendo, esto con el fin de no repetir código y ahorrar líneas de código.

Además que Symfony está desarrollado con PHP 5, y actualmente va cobrando auge en el mundo de los desarrolladores de aplicaciones web, y ha sido compatible con algunas bases de datos como es; MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Solo se aplica en plataforma Windows, sino también en la familia de sistemas operativos *nix, es decir Unix, Linux, etc.

Para poder descargar el código fuente se debe acceder a la página (<http://www.symfony-project.org/>), en donde se puede elegir el paquete según el sistema operativo en el que vamos a trabajar.

Se debe tomar en cuenta las siguientes consideraciones antes de decidirse a utilizar un framework de aplicaciones o no:

- Si el proyecto web va a consistir 5 o 10 páginas diferentes, acceso simple a bases de datos y no es importante asegurar un gran rendimiento o una documentación adecuada, se debería utilizar solo PHP sin framework.
- Ahora en el caso de que se vaya a desarrollar aplicaciones web complejas con mucha lógica de negocio, no es recomendable utilizar solo PHP. Para asegurar el mantenimiento y las ampliaciones futuras de la aplicación, es necesario que el código sea ligero, legible y efectivo, en esos casos si sería aconsejable utilizar Symfony.
- Además, para desarrolladores web profesionales que conozcan todas las ventajas de utilizar un framework de desarrollo de aplicaciones web y solo necesitan un

framework que sea maduro, bien documentado y con una gran comunidad que lo apoye, también en este caso deberían utilizar Symfony.

Para trabajar con Symfony es necesario tener conocimientos de PHP 5, además tener instalado por lo menos la versión PHP v5.2 como mínimo para que funcione Symfony.

Además se debe tener conocimientos básicos de algunos términos, los cuales tienen relación con este tipo de programación, esto se lo resume en el siguiente cuadro:

TÉRMINOS	SIGNIFICADO
OOP	La programación orientada a objetos se refiere a una aplicación en la que se puede considerar como una colección de unidades individuales, llamadas objetos, que interactúan entre sí. Los programas tradicionales pueden considerarse como una colección de funciones o como una lista de instrucciones de programación, Symfony hace un uso continuo de los mecanismos orientados a objetos disponibles en PHP 5.
ORM	Las bases de datos siguen una estructura relacional. PHP 5 y Symfony por el contrario son orientados a objetos. Por este motivo, para acceder a la base de datos como si fuera orientada a objetos, es necesaria una interfaz que traduzca la lógica de los objetos a la lógica relacional. Esta interfaz se denomina " <i>mapeo de objetos a bases de datos</i> " (ORM, de sus siglas en inglés " <i>object-relational mapping</i> "). Un ORM consiste en una serie de objetos que permiten acceder a los datos y que contienen en su interior cierta lógica de negocio.
RAD	El uso de lenguajes de script como es PHP permiten seguir otras estrategias de programación, como RAD (desarrollo rápido de aplicaciones) y el desarrollo ágil de software. Una de las ideas centrales de esta metodología es que el desarrollo empieza lo antes posible para que el cliente pueda revisar un prototipo que funciona y pueda indicar el camino a seguir. A partir de ahí, la aplicación se desarrolla de forma iterativa, en la que cada nueva versión incorpora nuevas funcionalidades y se desarrolla en un breve espacio de tiempo.
DRY	Cuando se modifican los requisitos o cuando se añade una nueva funcionalidad, normalmente se debe reescribir parte del código existente (refactorización), sucede a menudo durante el desarrollo de una aplicación web, los bloques de código repetidos se refactorizan en un único lugar, aplicando el principio DRY: No te repitas (<i>Don't Repeat Yourself</i>).
KISS	Las consecuencias de estas metodologías para el desarrollador son numerosas. El programador no debe pensar acerca de las versiones futuras al incluir una nueva funcionalidad. Los métodos utilizados deben ser lo más sencillos y directos posibles. Estas ideas se resumen en el principio denominado KISS: ¡Hazlo sencillo, idiota! (<i>Keep It Simple, Stupid</i>).
TDD	Para asegurar que la aplicación sigue funcionando correctamente a pesar de los cambios constantes, se necesita una serie de pruebas unitarias que puedan ser automatizadas. Si están bien escritas, las pruebas unitarias permiten asegurar que nada ha dejado de funcionar después de haber refactorizado parte del código de la aplicación. Algunas metodologías de desarrollo de aplicaciones obligan a escribir las pruebas antes que el propio código, lo que se conoce como TDD: desarrollo basado en pruebas (<i>test-driven development</i>).
YAML	YAML es un lenguaje muy sencillo que permite describir los datos como en XML, pero con una sintaxis mucho más sencilla. YAML es el acrónimo de " <i>YAML Ain't Markup Language</i> " ("YAML No es un Lenguaje de Marcado") y se

	pronuncia "yamel". Aparece en el 2001 y existen utilidades para procesar YAML en una gran variedad de lenguajes de programación, Symfony utiliza el formato YAML como el lenguaje preferido para almacenar su configuración.
PEAR	PEAR es un "framework y sistema de distribución para componentes PHP reutilizables". PEAR permite descargar, instalar, actualizar y desinstalar scripts de PHP. PEAR es un proyecto creado por la comunidad de usuarios de PHP, está desarrollado con PHP y se incluye en las distribuciones estándar de PHP.

Cuadro 3.1: Resumen de Términos Básicos que se relacionan con Symfony.

Symfony es un framework para desarrollar aplicaciones web creadas con PHP 5. Añade una nueva capa por encima de PHP y proporciona herramientas que simplifican el desarrollo de las aplicaciones web complejas, para entender este tipo de programación solamente es necesario estar familiarizado con los conceptos básicos de la programación moderna, sobre todo la programación orientada a objetos (OOP), el mapeo de objetos a bases de datos (ORM) y el desarrollo rápido de aplicaciones (RAD). El único requisito técnico obligatorio es el conocimiento de PHP 5.

3.2.3.2.2. Desarrollo de proyectos con Symfony

Para empezar a desarrollar proyectos con Symfony, además de conocer los anteriores términos hay que tener un conocimiento por lo menos básico de HTML, CSS, con esta forma de programación aprenderemos a ser un poco más ordenados en relación a lo que tiene que ver al lugar donde van a ir código de programación.

Pues hay veces que utilizaremos solo código HTML, en otros casos deberíamos utilizar solo PHP o una mezcla de ellos, además de hacer referencia a archivos CSS para poder dar estilos a nuestras páginas.

Lo primero que debemos hacer después de descargar el paquete symfony desde su sitio oficial que está empaquetado, es desempaquetar, una vez que tengamos en nuestro computador debemos grabar esta carpeta en la carpeta de nuestro proyecto, o puede ser en la carpeta www de la carpeta wamp en el caso de Windows c:\wamp\www\, es recomendable grabar en una carpeta distinta si vamos a trabajar en forma local, es decir sin subir al sitio en Internet, en el caso que se va a subir al ciberespacio, pero esto hay que hacerlo después de haber creado el proyecto con symfony.

Antes de continuar con la creación de un proyecto, nos parece importante entender como es la estructura del código Symfony. Todo proyecto creado por Symfony consta de una

serie de carpetas y archivos que contienen el código base del framework. El árbol de carpetas de un proyecto es:

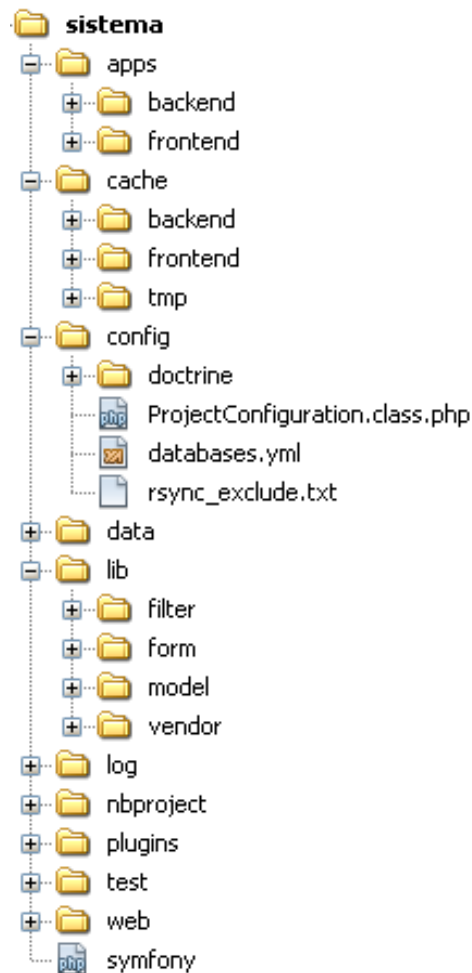


Gráfico 3.12.: Estructura de archivos de un Proyecto Symfony creado.

Fuente: Pantalla capturada

Además, un proyecto creado con symfony están divididos en:

- Aplicaciones: los mismos que pueden tener varios módulos, normalmente pueden haber dos aplicaciones Backend y Frontend.
- Módulos: Estos a su vez suelen coincidir con las tablas de la base de datos creada según sea el propósito del proyecto.

Para entender un poco más mejor a continuación se explicará cómo crear un pequeño proyecto, para ello haremos un blog muy simple **[WWW.021]**, y le pondremos el nombre blog, ésta a su vez, estará formado por dos aplicaciones:

- frontend (o la parte visible para los usuarios comunes) y
- backend (la parte de administración).

Y cada una de las aplicaciones tendrá dos módulos: post y comment.

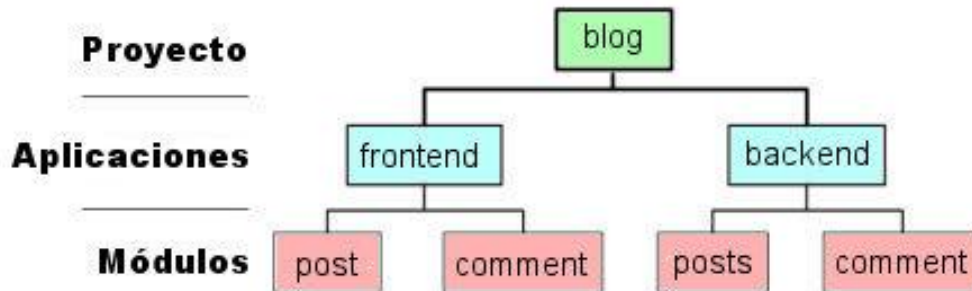


Gráfico 3.13.: Estructura del Proyecto Symfony del ejemplo blog.

Fuente: [WWW.021]

Lo primero que se debe hacer es modelar la base de datos, para eso se utilizará herramientas libres como es MySQL Workbench, en donde se puede realizar en una forma fácil y visual, este modelo constará de una tabla de posts y una tabla de comentarios con una relación 1:n, puesto que un post puede tener n comentarios.

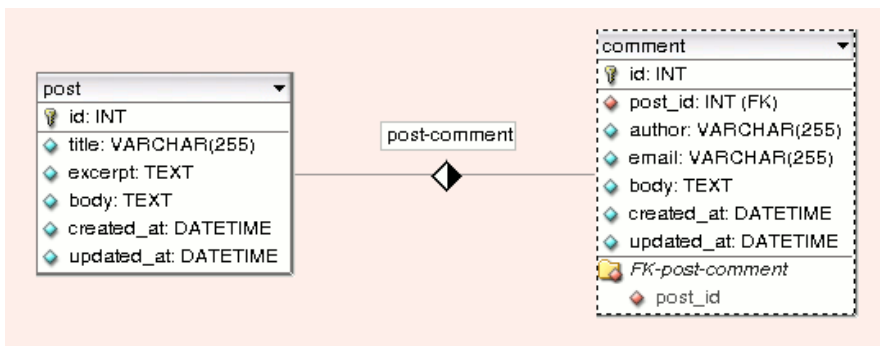
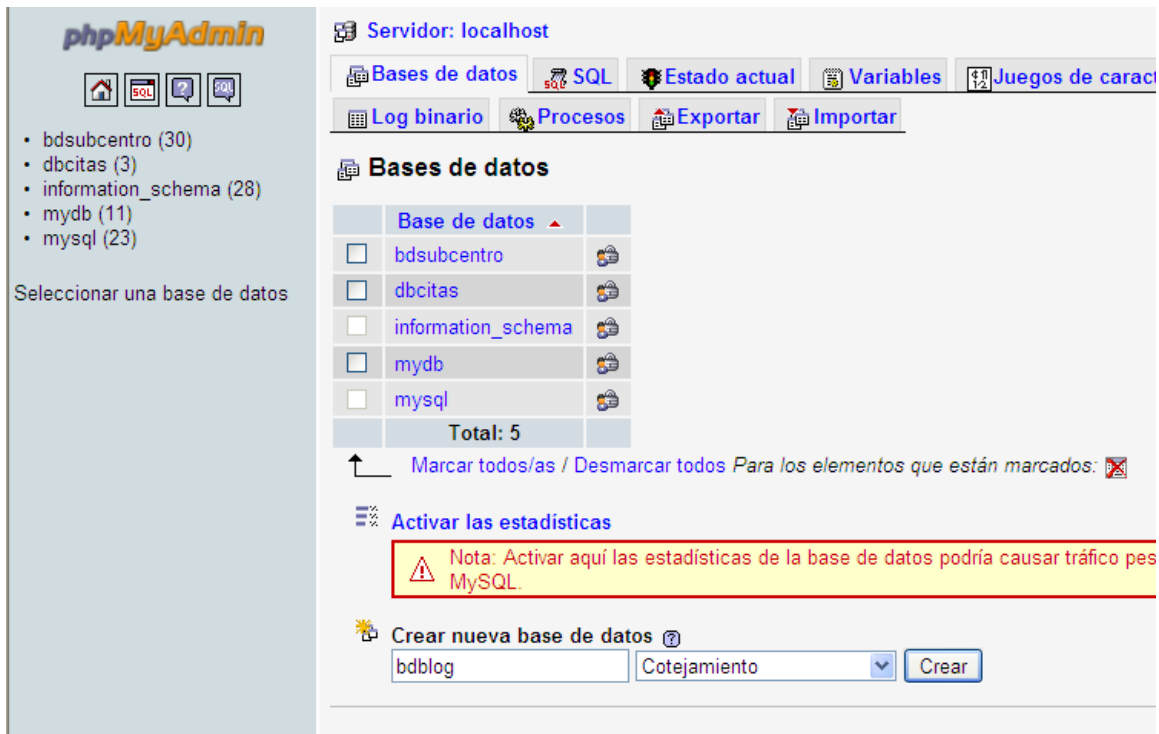


Gráfico 3.14.: Modelo de Base de Datos para el ejemplo blog.

Fuente: [WWW.021]

Pero esta herramienta solo se encarga de modelar la base de Datos, para crear la Base de Datos en MySQL, se accederá a la pantalla de phpMyAdmin, en donde se creará físicamente a la Base de Datos.



Después que se tenga la base de datos y se tenga claro como es la estructura de un proyecto symfony, se creará el código respectivo.

Para esto se creará una carpeta en donde a su vez se creará las demás carpetas y también copiar la carpeta Symfony descargada desde su sitio oficial, todo este trabajo se lo puede realizar en forma visual ayudados de la herramienta Netbeans, de la siguiente manera:

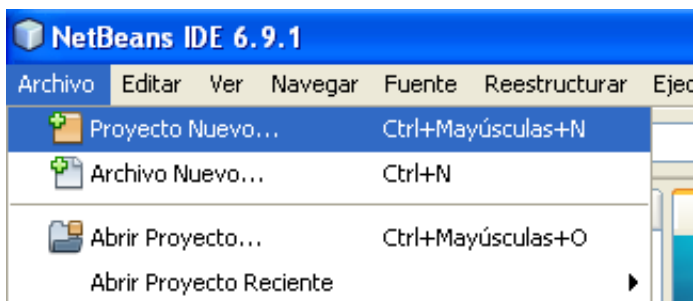


Gráfico 3.15.: Creación de un nuevo proyecto de forma visual con Netbeans.

Fuente: Pantalla capturada

Esta forma de crear el proyecto reemplaza al siguiente código:

```
symfony generate:project blog
```

Cualquiera sea la forma en la que se crea el proyecto, se habrá creado todo el árbol de directorios y archivos necesarios para poner en marcha nuestra web con este framework.

Luego, se creará los módulos, pero se utiliza Netbeans como es en el presente caso, al crear el proyecto también ya se crea estas dos aplicaciones por defecto.

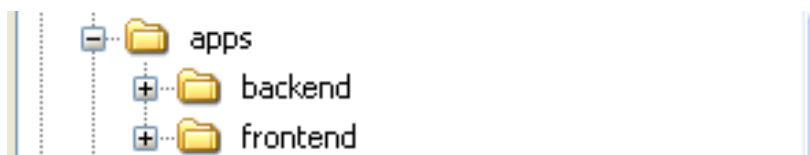


Gráfico 3.16.: Visualización de los proyectos creados por defecto con Netbeans.

Fuente: Pantalla capturada

En cambio si no utiliza Netbeans, tendrá que crearlos manualmente, aplicando un código similar al siguiente:

- `symfony generate:app --escaping-strategy=on --csrf-secret=UniqueSecret frontend`
- `symfony generate:app --escaping-strategy=on --csrf-secret=UniqueSecret backend`

Al llamar a la tarea `generate:app`, también se ha pasado dos opciones relacionadas con la seguridad:

- `--escaping-strategy`: Permite escapar la salida para evitar ataques XSS
- `--csrf-secret`: Permite tokens de sesión en los formularios para prevenir los ataques CSRF

El siguiente paso, será conectar esta aplicación con la base de datos, para ello, Symfony dispone de un archivo de configuración **databases.yml** en el que le se indica donde está localizada, a que base de datos y con qué usuario y contraseña lo deberá hacer.

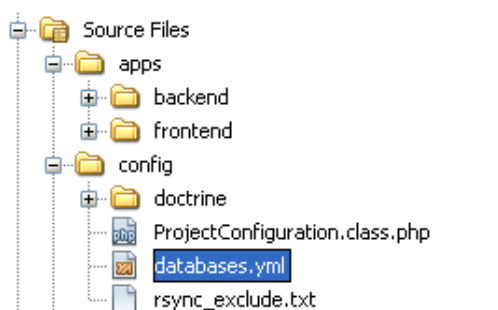


Gráfico 3.17.: Localización del archivo de configuración databases.yml.

Fuente: Pantalla capturada

Este archivo normalmente contiene el siguiente un código similar al siguiente:

```
all:
doctrine:
class: sfDoctrineDatabase
param:
dsn: 'mysql:host=localhost;dbname=bdblog'
username: root
password: null
```

Con esto, ya se tiene configurado el proyecto con la base de datos, luego con Doctrine se generará el archivo *schema.yml* a partir de la base de datos:

```
symfony doctrine:build-schema
```

Después de esto se crea el modelo, para que se genere clases para manejar la base de datos mediante objetos, de la siguiente manera:

```
symfony doctrine:build-model
```

De una manera similar se crea con doctrine, los filtros, los formularios, etc. Para no realizar de manera separada todas estas acciones, se utiliza un solo comando, de la siguiente manera:

```
Symfony doctrine:build --all
```

Solo que con el comando anterior hay que tener cuidado si es que ya tiene creado la base de datos anteriormente, ya que esta forma es una forma de resetear la base de datos y dejarla en blanco, o los vuelve a reconstruir la misma.

Como se mencionó anteriormente, se utilizará Netbeans, por lo que estas líneas de código lo podemos ejecutar de forma visual.

Si todo salió bien, ya podemos irnos con nuestro navegador a ver la pantalla de Symfony en donde dice que el proyecto ha sido creado, escribiendo la dirección **http://localhost/blog/** en nuestro navegador, de esta forma:



Gráfico 3.18.: Pantalla de Symfony, luego de haber creado nuestro nuevo proyecto symfony.

Fuente: Pantalla capturada

De aquí en adelante no es más que crear módulos en nuestras aplicaciones como es el caso de Backend y Frontend, además de insertar código para manipular y controlar nuestro objetivo en el sistema, pero estas líneas de código PHP van a ser mínimos [WWW.022].

3.2.3.3. Entorno de Desarrollo Netbeans

Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y en Abril del 2009 fue comprado por Oracle (<http://es.debugmodeon.com/debate/oracle-compra-sun/3#comments>). NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso [WWW.023].

En el apartado anterior se explicaba la creación de un pequeño proyecto “blog”, se utiliza este entorno de desarrollo para facilitar la programación y el desarrollo del sistema, pues

este ayuda bastante y permite ahorrar en gran cantidad el diseño, pues aquí se puede administrar de mejor manera las carpetas y archivos del proyecto y de una manera ordenada.

Este entorno fue creado inicialmente para el lenguaje Java, pero luego se crearon IDEs para que podamos trabajar con PHP. Con esta herramienta todo este trabajo se puede realizar de forma visual, ha resultado más cómodo a la hora de desarrollar en PHP e incluso para HTML y CSS.

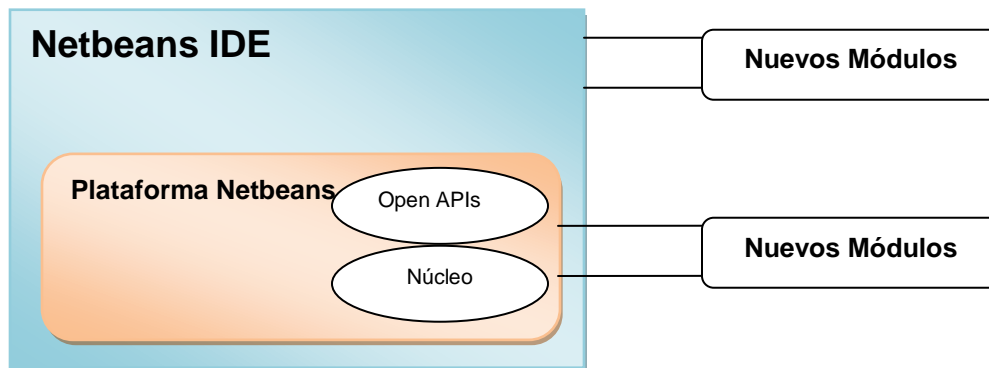


Gráfico 3.19.: Arquitectura de Netbeans.

Fuente: <http://www.slideshare.net/ralphkui/la-arquitectura-de-netbeans-v2>

NetBeans tiene diversas versiones dependiendo del lenguaje principal soportado. Es así que se tiene NetBeans para: Java, Ruby, C/C++, PHP y una versión que soporta todos los anteriores. En este caso se utiliza la versión PHP, la cual se puede descargar desde www.netbeans.org/downloads/.

Dentro de los pasos que se debe seguir para realizar la instalación está el de aceptar las licencias bajo las cuales ha sido liberado NetBeans. En resumen son las siguientes:

- GPLv2
- CDDL (Common Development and Distribution License)
- La licencia incluida en el mismo programa

Una vez finalizada la instalación se crea accesos directos en el escritorio y el menú del sistema. Al ejecutar el programa se abre la vista principal desde donde se puede iniciar un nuevo proyecto PHP:

Además de ser una aplicación libre y gratuita, algunas características adicionales en las

que NetBeans supera a su competencia. Por ejemplo:

- La simple configuración de nuevos proyectos, tanto para trabajo local como remoto.
- La información de cada función en el autocompletado de código.
- Los ayudantes para HTML (por ejemplo para formularios, tablas, etc.)
- Las extensiones y tutoriales disponibles en el sitio oficial del programa (www.netbeans.org)

[WWW.024]

3.2.3.3.1. Descarga e instalación de Netbeans.

La instalación de NetBeans, en cualquiera de los sistemas operativos Windows, Linux o Solaris, requiere de la instalación previa del JDK (Java Development Kit). Actualmente la versión más reciente es JDK 6 update 20, mientras que en lo que se refiere al IDE, la versión más moderna es NetBeans 6.9 [WWW.025].

Lo primero que se debe hacer es descargar e instalar Java, ya que como se dijo, Netbeans es hecho en su totalidad bajo Java, para eso tenemos que visitar la dirección <http://java.sun.com/javase/downloads/index.jsp>, desde donde descargamos la última versión de JDK, y por supuesto hay que asegurarse de que bajamos para la plataforma correcta, es decir, si vamos a trabajar bajo Windows o Linux, en relación al lenguaje sólo está disponible la opción Multilenguaje.

Para instalar JDK hacemos doble clic sobre

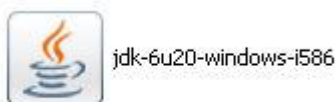


Gráfico 3.20.: Ícono del Instalador de Java.

Fuente: Pantalla capturada

Por lo que el Asistente de instalación nos guiará durante todo el proceso de instalación de Java.

Una vez instalado Java en nuestra máquina, ya estamos listos para proceder con la

instalación de nuestro entorno de desarrollo Netbeans.

Para esto lo primero que tenemos que hacer es irnos a la página principal de NetBeans <http://netbeans.org/downloads/index.html>.

Descargar NetBeans IDE 6.9

Idioma del IDE: **Español** Plataforma: **Windows 2000/XP/Vista**

Descarga de paquetes Netbeans IDE en idiomas aportados por la comunidad.¹

Tecnologías *	Java SE	JavaFX	Java	Ruby	C/C++	PHP	All
NetBeans Platform SDK	•	•	•				•
Java SE	•	•	•				•
JavaFX		•					•
Java Web y EE			•				•
Java ME			•				•
Java Card™ 3 Connected			•				•
Ruby				•			•
C/C++					•		•
Groovy			•				•
PHP						•	•
Servidores incluidos							
GlassFish Server Open Source Edition 3.0.1			•	•			•
Apache Tomcat 6.0.26			•				•

Download Libre, 66 MB Download Libre, 133 MB Download Libre, 225 MB Download Libre, 94 MB Download Libre, 43 MB Download Libre, 38 MB Download Libre, 321 MB

Gráfico 3.21.: Pantalla para seleccionar idioma y lenguajes antes de descargar Netbeans.

Fuente: [WWW.026].

En el Gráfico 3.19 se puede elegir inicialmente tanto el lenguaje como el sistema operativo en que deseamos trabajar, por otra parte también se puede seleccionar las opciones que se ajusten a los proyectos con los que se va a trabajar, así como los servidores web que permitirán ejecutar las aplicaciones y finalmente también puede seleccionarse las distintas plataformas con las que se desee trabajar según el proyecto: J2SE, J2EE y J2ME, se instalará con solo doble clic en el icono que aparece luego de descargarse:



Gráfico 3.22.: Ícono del Instalador de Netbeans 6.9.1 para PHP en Windows.

Fuente: Pantalla capturada

Por lo que aparecerá la siguiente pantalla:



Gráfico 3.23.: Pantalla de Bienvenida en la instalación de Netbeans.

Fuente: Pantalla capturada

Al hacer clic en “Personalizar”, personalizamos esta instalación, como es el caso de que usted quiera instalar Apache Tomcat Ruby, Php, etc., con lo que aparecerá:

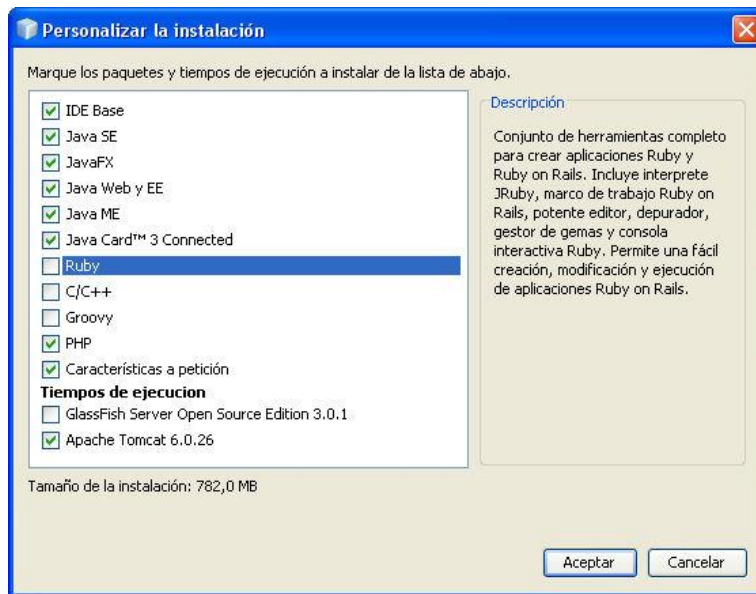


Gráfico 3.24.: Pantalla para personalizar Instalación de Netbeans.

Fuente: Pantalla capturada

Aquí se puede seleccionar o deseleccionar algunos paquetes para no ocupar mucho espacio en la instalación, luego aceptar:

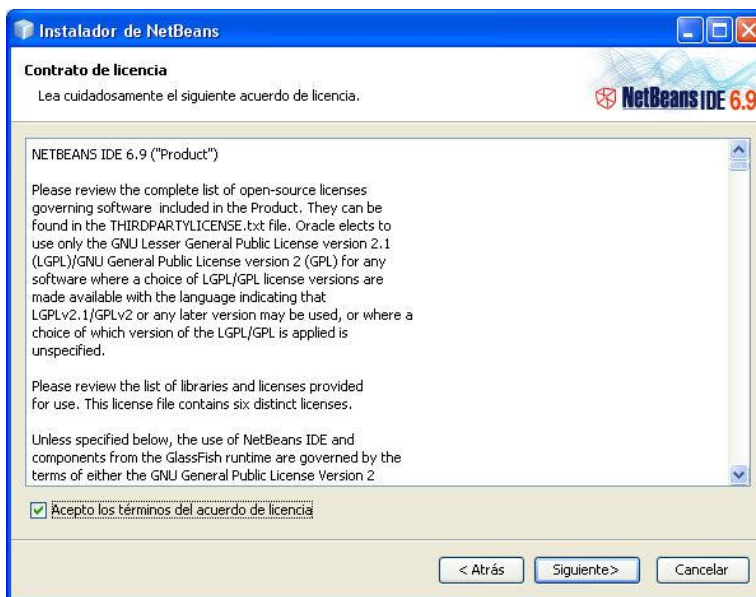


Gráfico 3.25.: Pantalla de Acuerdo de Licencia en la Instalación de Netbeans.

Fuente: Pantalla capturada

Hay que aceptar los términos de la licencia y clic en "Siguiente"

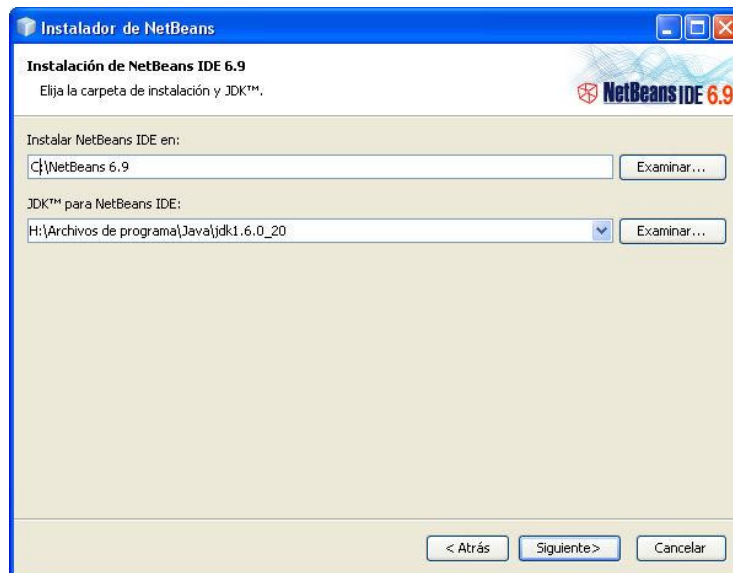


Gráfico 3.26.: Pantalla donde nos permite seleccionar carpeta para instalar Netbeans.

Fuente: Pantalla capturada

También hay que fijar el directorio y subdirectorios donde se desea instalar y clic en “Siguiete”.

Al final de la instalación después de varias ventanas, según los paquetes que haya escogido, le pedirá hacer clic en Instalar, por lo que se tendrá una pantalla como la siguiente:

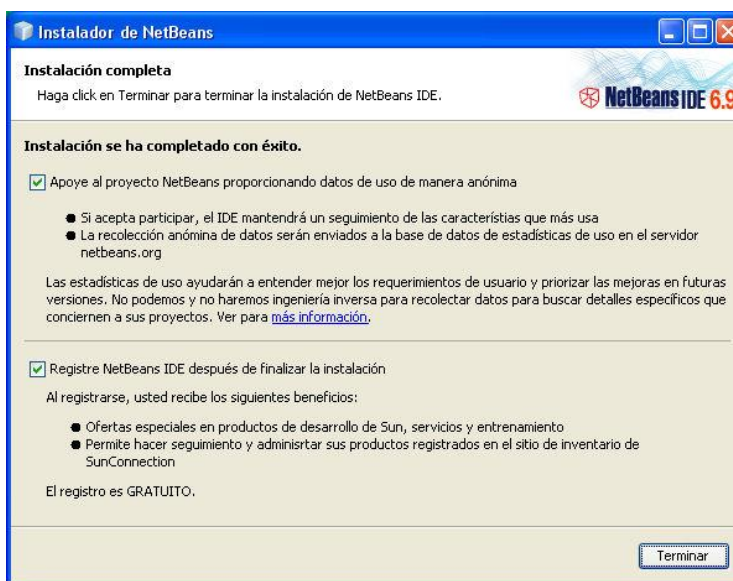


Gráfico 3.27.: Pantalla final en la instalación de Netbeans.

Fuente: Pantalla capturada

Si se desea se puede crear una cuenta, y cerrar el navegador, y con esto queda terminada la instalación de NetBeans 6.9.

Para comprobar que ya está instalado se puede observar en el escritorio un icono similar al siguiente:



Gráfico 3.28.: Ícono de Acceso directo a Netbeans 6.9 después de la Instalación.
Fuente: Pantalla capturada

En donde, si se hace doble clic sobre este, aparecerá el IDE

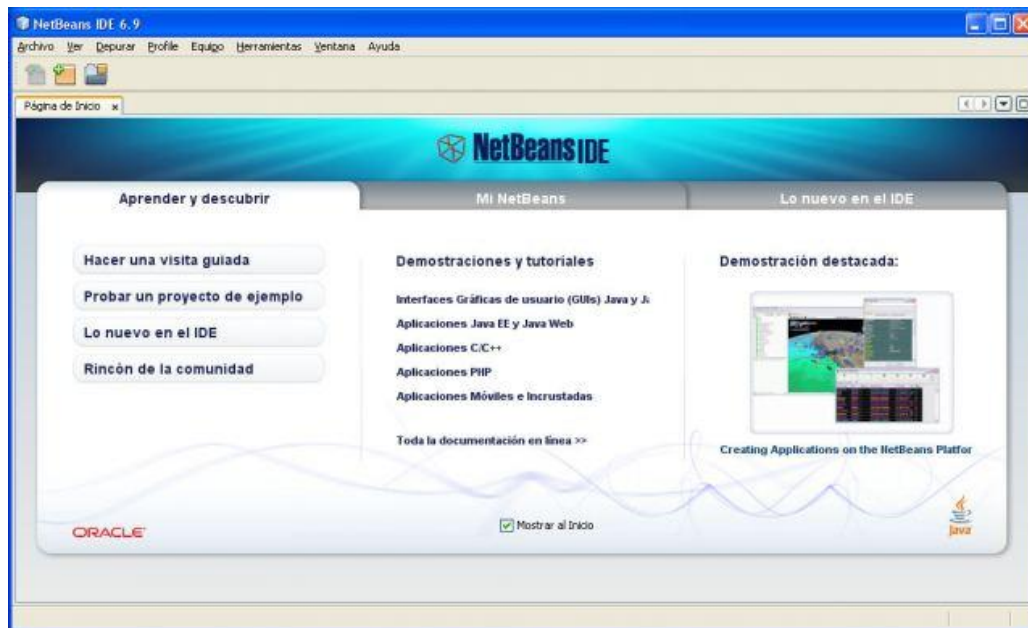


Gráfico 3.29.: Primera Pantalla en la ejecución del Netbeans IDE 6.9.
Fuente: Pantalla capturada

Hay que tomar en cuenta que paquetes no más se va a instalar, para que no ocupe mucho espacio en esta instalación..

3.2.3.3.2. Configuración de Symfony en Netbeans.

La ventaja de haber escogido Symfony para hacer el sistema es que hay una comunidad muy activa que proporciona muy buena documentación sobre cómo utilizarlo, además en internet se encuentra el manual oficial traducido al castellano (<http://netbeans.org/>).

Como tantas veces ya se ha dicho en este documento, Symfony se basa en el patrón **MVC (Modelo Vista Controlador)**, es decir, la estructura lógica de los proyectos creados se divide en:

- **Aplicaciones:** es decir el proyecto a desarrollarse,
- **Clases:** las clases los tipos de funcionalidades de las que consta la aplicación (gestión de usuarios, gestión de noticias, gestión de comentarios, etc.) y,
- **Módulos:** que es donde se detalla cómo hacer cada funcionalidad de la clase a la que están asignados (alta, borrado y edición de usuarios, etc.).

Al iniciar con el uso de este framework, no se sabe como empezar, por lo que no se conoce la sintaxis y la metodología para programar las operaciones, pero la práctica hace que se familiarice poco a poco corrigiendo errores, experimentando código, etc., por supuesto que hay que haber tenido bases de programación para que con el tiempo pueda ir adquiriendo destrezas.

En el caso de estar familiarizados con la programación visual, para que nuestro trabajo sea lo más sencillo posible, se debe decidir por el programa o IDE que se va a utilizar para el proyecto web. NetBeans 6.9.1, la versión más actual que encontramos a la fecha de iniciación de nuestro trabajo, ya que además de ser opensource y totalmente gratuito, pues en esta versión nos ofrece un asistente para crear un proyecto PHP utilizando el Symfony como framework.

La configuración de Netbeans para trabajar con Symfony, se hace de la siguiente manera:

En primer lugar se debe tener un “LAMP”⁸ (Linux-Apache-MySQL- PHP/Python/PERL) instalado en nuestro equipo, en nuestro caso utilizamos WAMP SERVER 2.0.

⁸ Conjunto de aplicaciones para configurar sitios web o servidores dinámicos con un esfuerzo reducido.

Luego se descarga la última versión estable de Symfony desde su web, para el inicio de nuestro proyecto la última versión estable fue Symfony 1.4 (<http://www.symfony.es/>).

Una vez descargado el paquete que normalmente viene empaquetado, se lo descomprime en el directorio de nuestra web, para el presente proyecto se trabaja en forma local, se crea una carpeta en la unidad C:/, de la siguiente manera: "C:\SubcentroSalud\sistema\", entonces la dirección completa donde se lo descomprime va a ser algo así: C:\SubcentroSalud\sistema\lib\vendor\symfony.

El siguiente paso es abrir Netbeans, para luego crear un nuevo proyecto PHP, el asistente guiará paso a paso, en donde solo se da el nombre al proyecto y localización de los archivos del mismo, pero en el paso 4 donde se debe elegir el framework (PHP frameworks), obviamente se elige Symfony. Al seleccionarlo aparecerán varias opciones de configuración se tiene que pulsar sobre Options, a la derecha de la ventana, como se indica en el **Gráfico 3.28**:

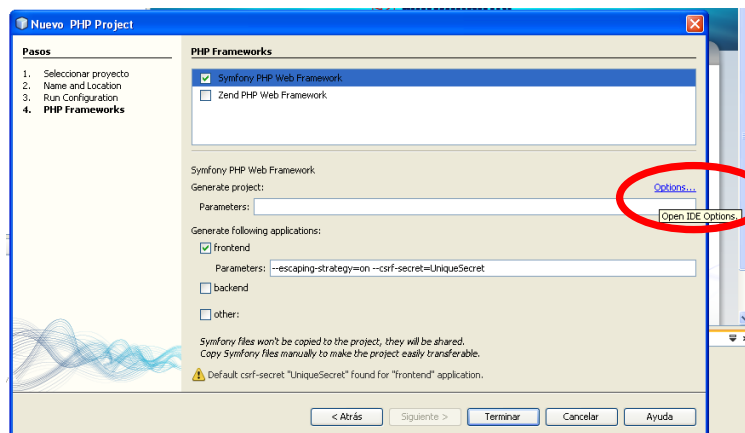


Gráfico 3.30.: Selección "Options" para configurar correctamente symfony.

Fuente: Pantalla capturada

Luego de hacer clic en Options..., se abrirá el panel de configuración de PHP, en esta ventana se hace clic en la pestaña de PHP en donde se indica la ruta donde se encuentra el ejecutable de PHP del LAMP que hayamos instalado, en este caso se encuentra en la siguiente ruta:

C:\wamp\bin\php\php5.3.0\php.exe

Además, en esta misma ventana se tendrá que indicar la ruta de instalación de Symfony pulsando el botón **Add Folder**, que será la carpeta que se ha descomprimido en el directorio del proyecto

C:\SubcentroSalud\sistema\lib\vendor\symfony

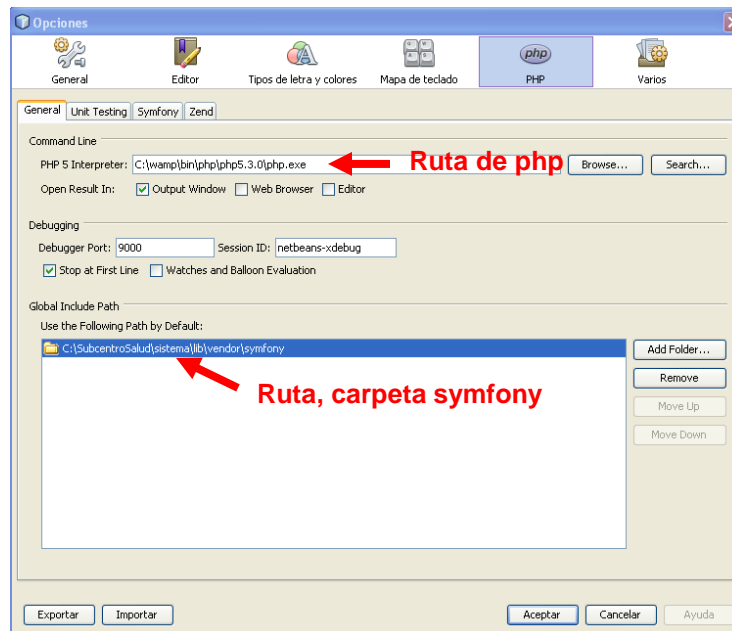


Gráfico 3.31.: Selección de rutas de instalación de PHP y Symfony.

Fuente: Pantalla capturada

Ahora, en la pestaña Symfony, de esta misma pantalla se tendrá que indicar la ruta del fichero binario del framework, la cual en este caso se lo encuentra en la siguiente dirección:

C:\SubcentroSalud\sistema\lib\vendor\symfony\data\bin\symfony

Como se muestra en el siguiente gráfico:

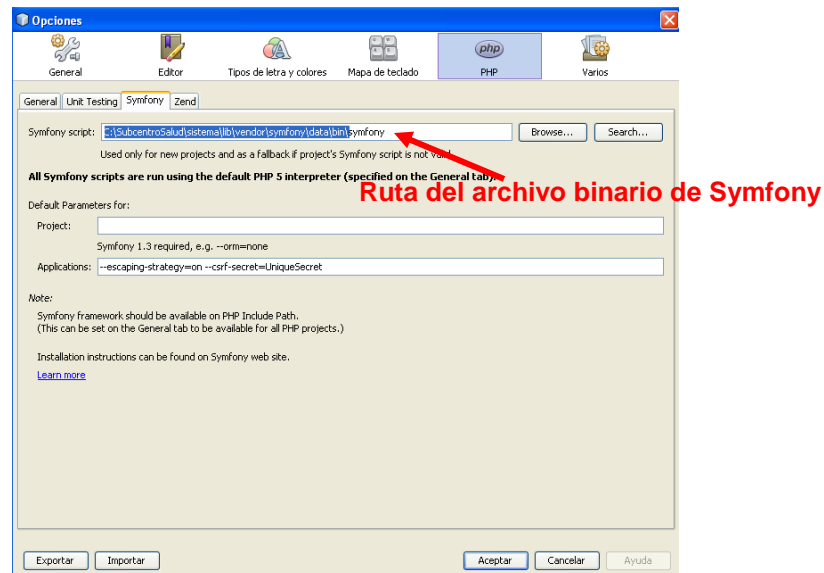


Gráfico 3.32.: Selección del archivo binario de Symfony.

Fuente: Pantalla capturada

Y por último, se hace clic en **Aceptar**, pero antes de terminar, hay que seleccionar el checkbox **Backend**, en donde debe quedar con un visto como se indica en el siguiente gráfico, esto con el fin de que el proyecto que se está realizando maneje la base de datos, luego de esto se pulsa en **Terminar**.

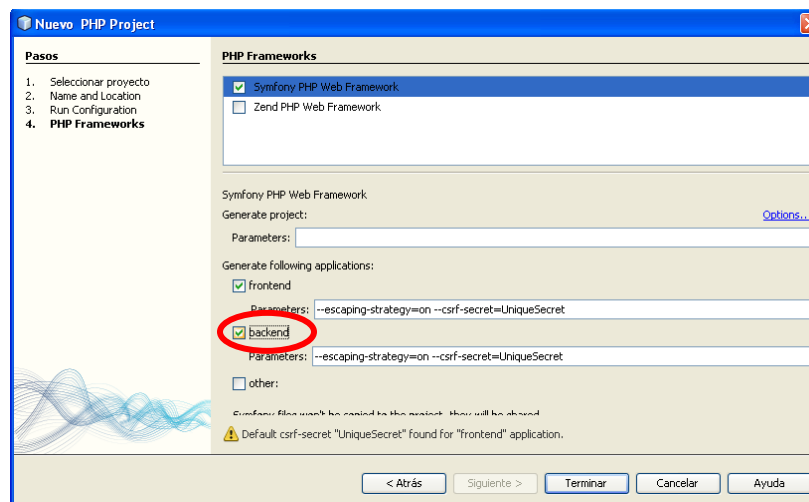


Gráfico 3.33.: Selección "backend" al final de la configuración de symfony.

Fuente: Pantalla capturada

Si no hay ningún inconveniente en el transcurso de todos los pasos que el asistente sirvió de guía, ya tendremos configurado e instalado nuestro proyecto **Symfony** en **NetBeans**.

3.2.3.3.3. Configuración de Apache.

Con los pasos ya detallados en el apartado anterior no está terminado todo, pues esta aplicación debe ejecutarse en el Apache Web Server la misma que se debe ya tenerlo instalado, Symfony recomienda ejecutar la aplicación de desarrollo en otro host, en este caso el host será local y virtual, de tal manera que se ejecute bajo otro nombre de host y en otro puerto, por ejemplo en el puerto 80.

En internet se encuentra varios ejemplos de cómo crear un host virtual para ejecutar el proyecto de forma local, aquí se muestra un ejemplo:

```
NameVirtualHost *:80
<VirtualHost *:80>

ServerName miApp
DocumentRoot "C:\Users\Juanjo\Documents\NetBeansProjects\miApp\ web"
DirectoryIndex index.php
<Directory "C:\Users\Juanjo\Documents\NetBeansProjects\miApp\ web">
AllowOverride All
Allow from All
</Directory>

#solo para la barra de depuración:

Alias /sf "C:\symfony-1.4.11\data\web\sf"
<Directory "C:\symfony-1.4.11\data\web\sf">
AllowOverride All
Allow from All
</Directory>

</VirtualHost>
```

[WWW.027]

Como paso siguiente, se coge cualquiera de estos códigos y se lo edita para acoplarlo al proyecto, quedando de la siguiente manera:

```
# Esta es la configuración de Sistema para Subcentro de Salud
#Listen 127.0.0.1:80

<VirtualHost 127.0.0.1:80>
ServerName sistema
DocumentRoot "C:\SubcentroSalud\sistema\web"
DirectoryIndex index.php
Alias /sf "C:\SubcentroSalud\sistema\lib\vendor\symfony\data\web\sf"
<Directory "C:\SubcentroSalud\sistema\lib\vendor\symfony\data\web\sf">
```



```
AllowOverride All
Allow from All
</Directory>
<Directory "C:\SubcentroSalud\sistema\web">
  AllowOverride All
  Allow from All
</Directory>
</VirtualHost>
```

Como ya están establecidas por Symfony las rutas de la carpeta del proyecto, solo queda seguir los siguientes pasos:

1. Copiar el contenido (Ctrl+C) del código anterior (del VirtualHost),
2. Abrir el archivo httpd.conf que se encuentra en la siguiente ruta:
C:\wamp\bin\apache\Apache2.2.11\conf,
3. Lo pegamos (Ctrl+V), normalmente en la parte final del contenido de este archivo,
4. Guardarlo,
5. Y por último reiniciar los servicios de Apache.

Si todo ha salido bien, y si la configuración de Apache fue la correcta, se abre el navegador en donde se anota la siguiente dirección: <http://localhost:80>, pero también se lo puede hacer desde Netbeans, digitando la tecla F6 para ejecutar el proyecto. Y de esta manera debe salir una pantalla similar a la siguiente:

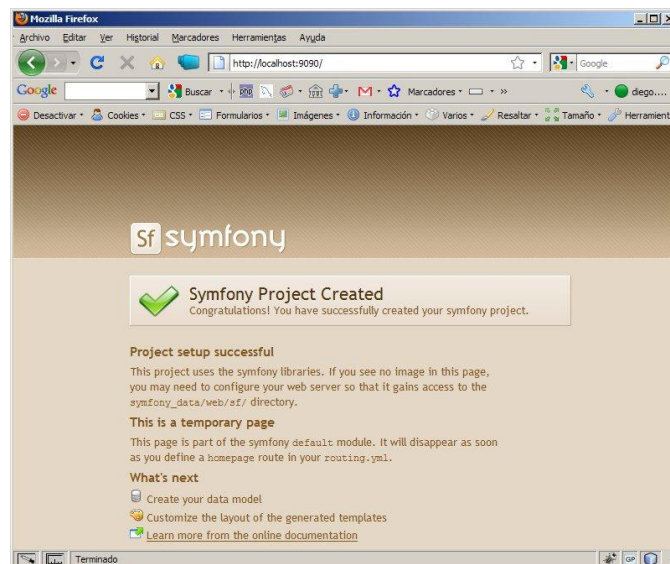


Gráfico 3.34.: Pantalla de Bienvenida de Symfony.

Fuente: Pantalla capturada

3.2.3.3.4. Ejecución de Comandos Symfony en Netbeans.

Los manuales que se encuentra en Internet, con lo referente a la creación de proyectos con Symfony, lo hacen la mayoría en modo de líneas de comandos, desde la creación de proyectos, de módulos, de archivos, etc., pero se puede intuir o relacionar esta forma de trabajar en modo de línea de comandos y en forma visual como lo se lo hace con Netbeans.

En la siguiente pantalla se observa cómo se puede ejecutar comandos de Symfony fácilmente en forma visual:

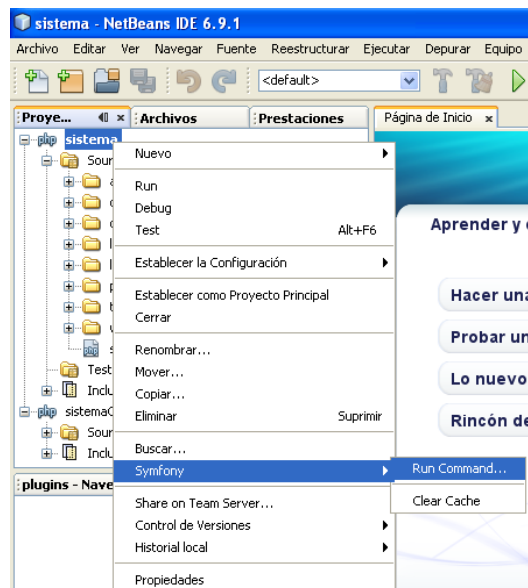


Gráfico 3.35.: Ejecución de comandos Symfony en Netbeans 6.9.1.

Fuente: Pantalla capturada

La forma más fácil para lograr esto es, haciendo clic derecho sobre la carpeta del proyecto en donde se está trabajando, luego clic en Symfony en donde aparece la opción "Run Command...", luego de hacer clic en esta opción aparecerá la siguiente pantalla:

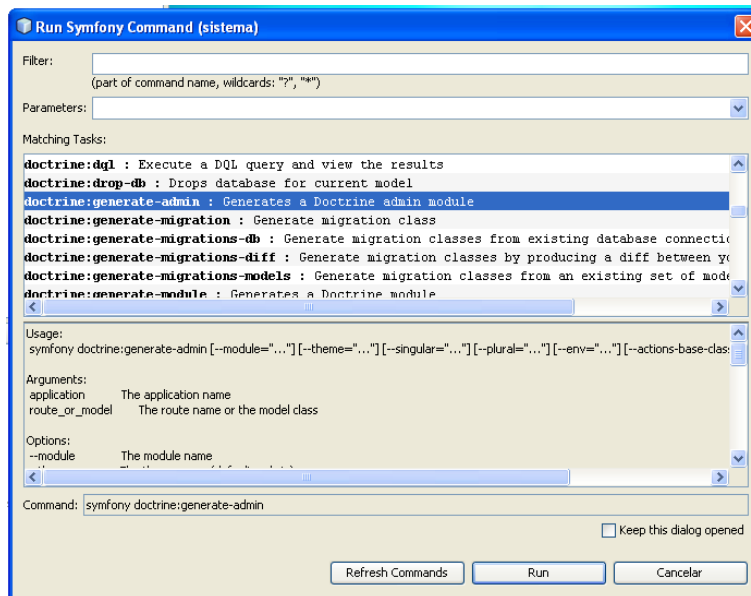


Gráfico 3.36.: Selección de comandos Symfony en Netbeans 6.9.1.

Fuente: Pantalla capturada

Aquí se presenta un lista de comandos disponibles dependiendo del “ORM”⁹ que se esté trabajando (Propel o Doctrine), en este caso Doctrine, también aquí se encuentra comandos de los “PLUGINS”¹⁰ que tendremos instalados, además se puede ver en la pantalla anterior que presenta una ayuda del comando seleccionado, ayudando más aún en la creación del nuevo proyecto web.

De esta manera se ejecutar los comandos de Symfony de una manera visual, y se evita trabajar en línea de comandos.

3.3. ALCANCE DEL SISTEMA.

En cuanto al alcance o cobertura que tiene el sistema, este análisis se lo va a realizar de la siguiente manera:

USUARIOS: El usuario más frecuente va a ser la enfermera/o, en donde a la cabeza va a estar la actual directora del Subcentro la Lcda. Maritza Álvarez (en la actualidad), la misma que tiene a su cargo toda la información de historias clínicas y otro tipo de información del Subcentro.

⁹ Técnica de programación que nos permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

¹⁰ Es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande

La misma directora se encarga de delegar el cargo a las otras enfermeras de planta que son en un número de 3, estas a su vez pueden delegar a estudiantes de enfermería que realizan pasantías en esta casa de salud, los mismos que varían en cada semestre.

Pero también, un médico puede acceder a esta base de datos con su propia cuenta o con la cuenta de una enfermera o pasante, tal vez no entre con mucha frecuencia, pero si lo hará en su respectivo momento.

Esto se lo resume en el siguiente gráfico:

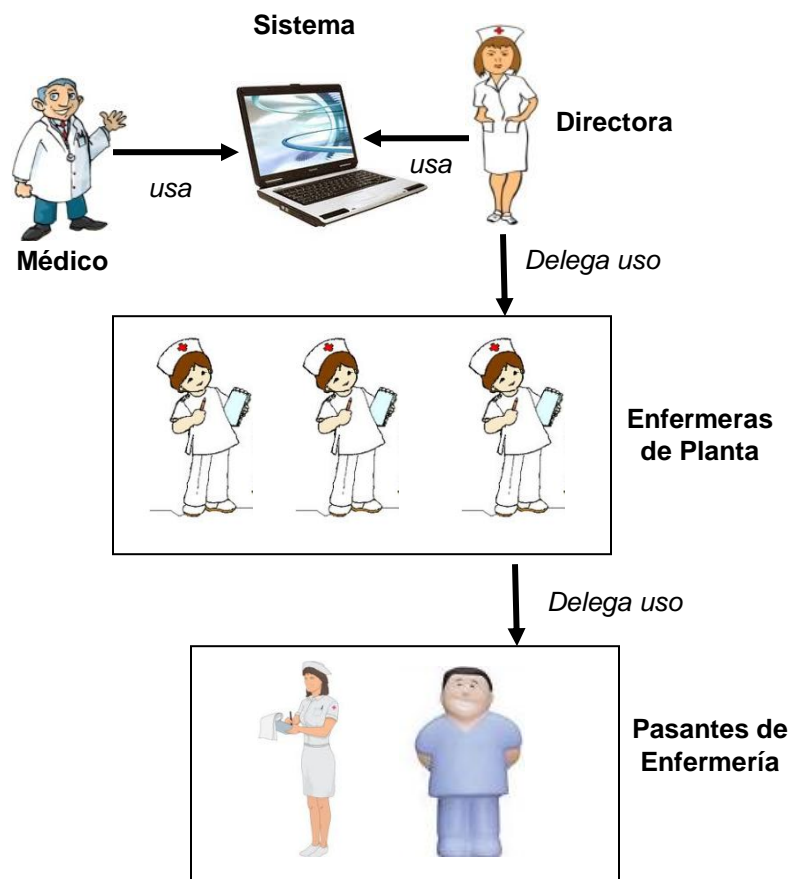


Gráfico 3.37.: Esquema de cobertura de usuarios del sistema.

Fuente: Diseño Propio.

HARDWARE: Actualmente el Subcentro no cuenta con una buena infraestructura, solo cuenta con tres equipos de cómputo, una impresora y un modem que últimamente el gobierno ha instalado para proveer internet, la misma que solo tiene en una máquina.

Implementando el sistema se puede proyectarlo a utilizarlo en máquinas clientes en todos sus 5 consultorios médicos que tiene actualmente el Subcentro, a más del departamento de FARMACIA, el único problema es que el Subcentro no cuenta con recursos para adaptar el sistema en todo el espacio físico de esta casa de salud, por lo que optamos en instalar en el punto más crítico del trabajo del Subcentro esto es en el departamento de ESTADÍSTICA en donde se mueve a diario toda la información necesaria para su trabajo.

El esquema ideal de trabajo del sistema se lo resume en el siguiente gráfico:

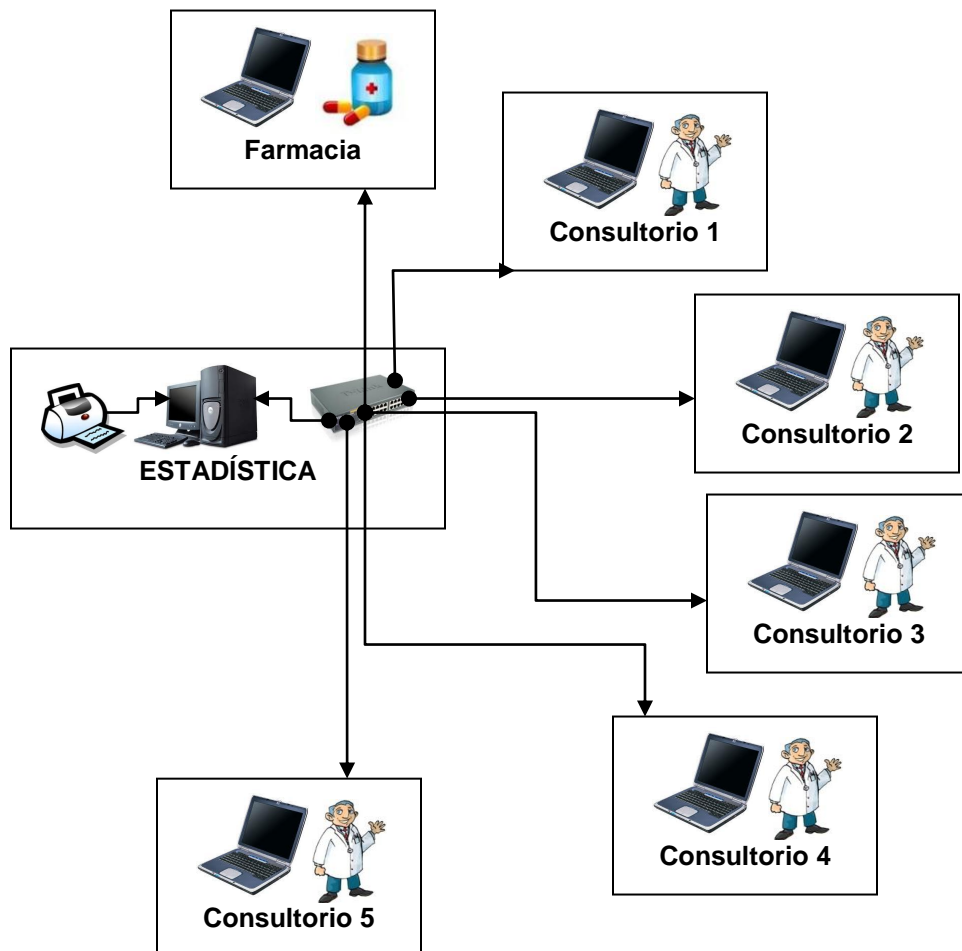
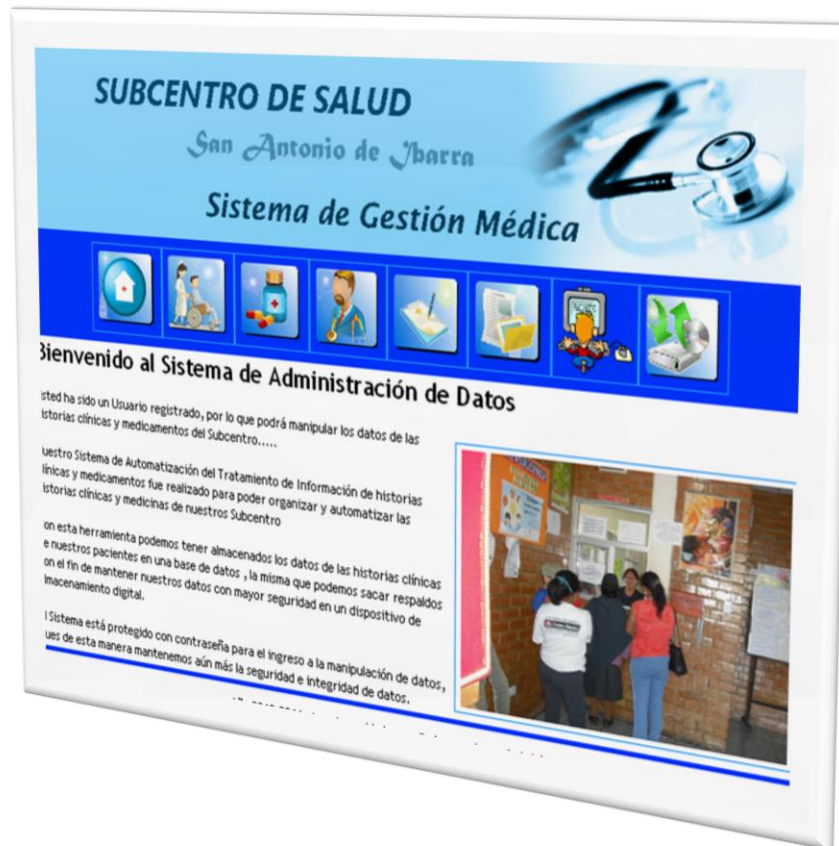


Gráfico 3.38.: Esquema ideal de trabajo en el Subcentro de Salud.

Fuente: Diseño Propio

CAPÍTULO 4

CONSTRUCCIÓN DEL SISTEMA



4.1. Análisis y Recolección de Información

4.2. Diseño

4.3. Arquitectura y Codificación del Sistema

4.4. Implementación

4.5. Instalación y Capacitación

4.1. ANÁLISIS Y RECOLECCIÓN DE INFORMACIÓN

4.1.1. Análisis de Usuarios del Sistema

El presente sistema, en realidad va a trabajar dentro del Subcentro de salud, en una intranet, por lo que los usuarios más comunes van a ser las enfermeras de turno, y los médicos, pero tal vez en casos extraordinarios van a ser personas ajenas al Subcentro, en cuyo caso los llamaremos auxiliares.

Ahora se va a realizar un análisis de estos tipos de usuarios en forma más detenida:

Usuario-Enfermeras/os: Es la persona que más cerca del sistema va a estar, pues, es la persona que todos los días dan los turnos a los pacientes y conocen más de cerca los problemas de aglomeración en las colas que se hacen para la petición de turnos o información.

En el Subcentro de Salud de San Antonio de Ibarra, toda la información más importante de los pacientes se encuentra en el departamento de ESTADÍSTICA, en cuya localización tiene también una computadora y una impresora, que se utiliza en su mayoría para grabar documentos e imprimirlos, documentos como oficios, solicitudes y otros archivos que tienen relación con esta casa de salud y es en donde la mayoría de información se maneja o se centra aquí. Es por esta razón que pensamos que es en donde deberíamos tener el servidor de la base de datos de las historias clínicas de los pacientes.

Además en este departamento se encuentran las enfermeras/os dando turno a los pacientes y son este tipo de usuarios los que están más allegados al archivo de las historias de los pacientes que actualmente aún se maneja en carpetas las mismas que se encuentran ordenadas en un anaquel desde hace varios años.

Usuarios-Médicos: Con esta nomenclatura nos referimos a los médicos que atienden a los pacientes, los mismos que se dedican a chequearlos en cada uno de sus consultorios, los mismos que no van a tener tantos conocimientos acerca de la organización de los archivos tanto en los anaqueles, pero por A o B circunstancias van a necesitar entrar a manejar el sistema, tanto para averiguar acerca de la existencia de medicamentos en

farmacia, para la extensión de recetas impresas mediante el sistema, o para ver o revisar la historia clínica del paciente al que está atendiendo en ese momento.

Este tipo de usuario, si bien es cierto que no va a relacionarse tan estrechamente con el sistema, pero si le va a ser útil en su momento.

Usuario-Auxiliar: Este tipo de usuario no va a ser tan común, ya que no se va a presentar tan a menudo, pues nos referimos al caso en que la enfermera en turno o algún médico permitirá el acceso de los datos por algún motivo excepcional, pues por mantener la integridad de los datos, esta información solo está dedicada en especial a las/los enfermeras/os y a los médicos, ya que ellos están más relacionados con esta información.

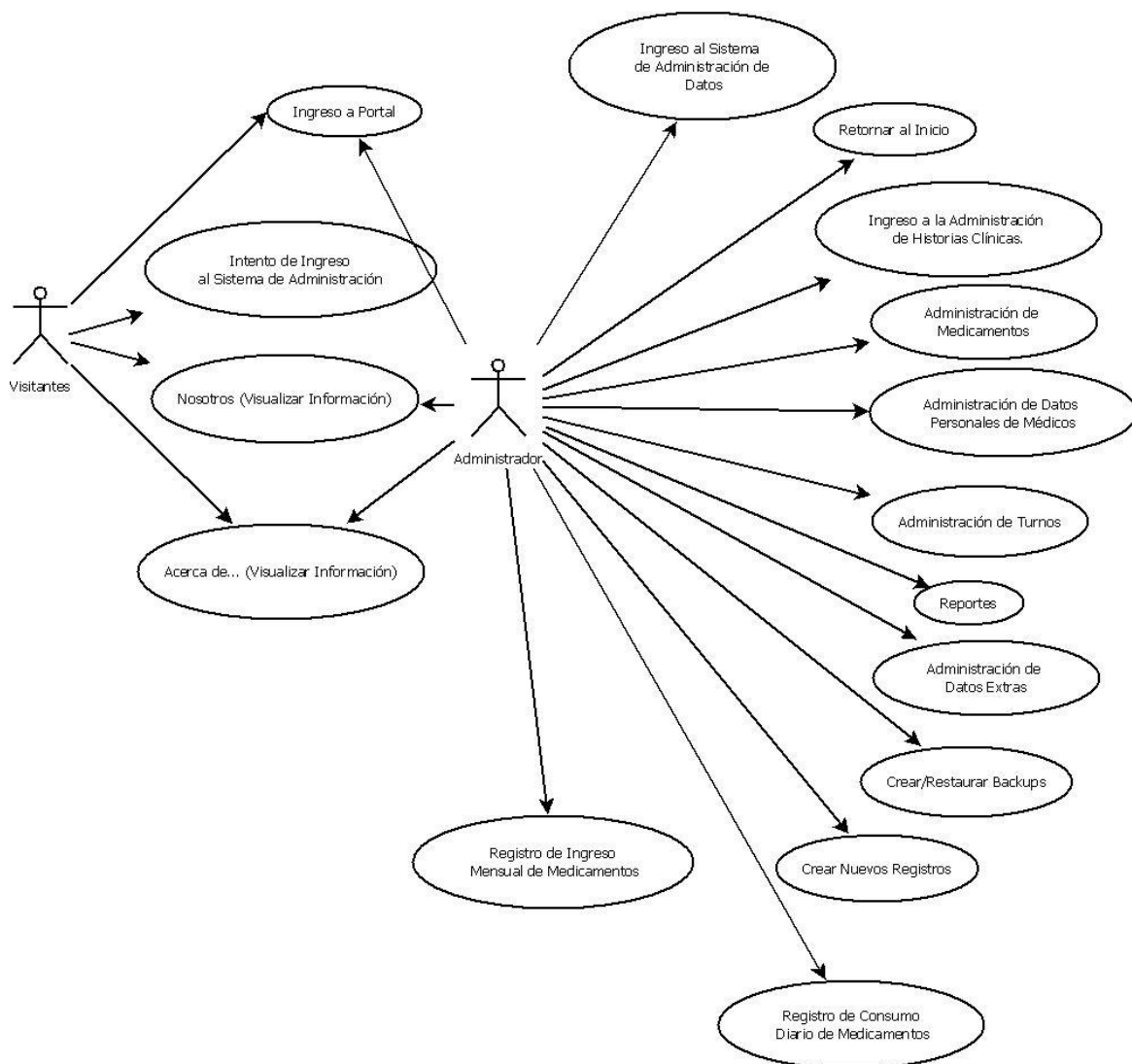


Gráfico 4.1.: Esquema de Relación Usuario-Sistema.

Fuente: Diseño Propio

4.1.2. Construcción del Producto (Casos de Uso).

4.1.2.1. Caso de Uso Gu: Gestionar de Usuarios.

Descripción Breve: El proyecto tiene dos tipos de usuarios:

- **Administradores:** Se refiere al usuario que puede realizar cualquier operación con la base de datos, es decir crear, eliminar, modificar, sacar respaldos (Backups), recuperar respaldos, este usuario puede ser las enfermera encargadas, médicos o auxiliares, previo a su registro como usuario aceptado para este tipo de trabajos.
- **Usuarios:** El usuario común puede ser un paciente, o algún usuario que puede visitar la página solo para curiosear o investigar algún aspecto sobre el sistema, en realidad este tipo de usuario no puede realizar mayor cosa, ya que el sistema está hecho solo para la administración de la base de datos de pacientes en su mayoría.

Flujo Básico de Eventos:

- Para ingresar al sistema debe ingresar a la dirección <http://sistema/>, en su navegador web (de preferencia Mozilla Firefox), aquí podrá observar una descripción breve del Subcentro de salud “San Antonio”, en donde además de mostrar imágenes me muestra iconos de enlace para poder ingresar al sistema, mostrar información en el enlace “Nosotros” y otro icono llamado “Acerca de..”.

4.1.2.2. Caso de Uso Fr1: Mostrar Información General

Descripción Breve: En la página principal, los usuarios se informaran en forma general sobre el Subcentro de Salud, pues al ingresar a la página web del Subcentro se mostrará un “Slide”¹¹ de Fotografías del Subcentro y en donde se especifica su ubicación en la parte de abajo de las imágenes.

Flujo Básico de Eventos:

Aquí se muestra los íconos de acceso a las siguientes opciones:

¹¹ Presentación de imágenes o fotografías.

- Ingreso al Sistema (Caso de Uso Fr2)
- Nosotros (Caso de Uso Fr3)
- Acerca de...(Caso de Uso Fr4)

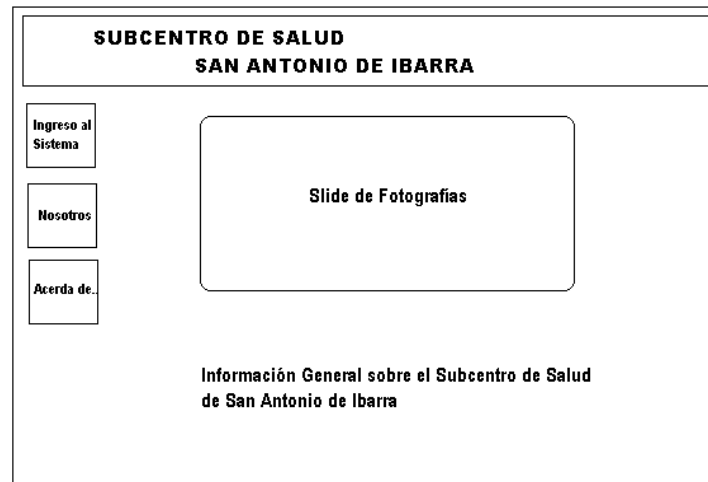


Gráfico 4.2.: Esquema de presentación del Caso de Uso Fr1.

Fuente: Diseño Propio

Precondiciones:

- No existe una condición anterior específica para ingresar a este apartado, pues cualquier usuario puede ingresar a ver esta información.

4.1.2.3. Caso de Uso Fr2: Intentar Ingresar al Sistema

Descripción Breve: El Usuario Intenta ingresar al Sistema de Gestión de Información, solo tendrá éxito si es un usuario registrado previamente en la base de datos.

Flujo Básico de Eventos:

- Cuando el Usuario hace clic en el ícono de acceso al Sistema nos presentará una pantalla en donde le pedirá el nombre de usuario y la contraseña.
- En caso de ingresar los datos correctos o ser un usuario registrado, solo en ese caso podrá ingresar al Sistema de Gestión de Información del Subcentro, en caso contrario solo quedará aquí.

La tabla de usuarios registrados tiene la siguiente estructura:

sf_guard_user

Campo	Tipo	Nulo	Predeterminado
id	bigint(20)	No	
first_name	varchar(255)	Sí	NULL
last_name	varchar(255)	Sí	NULL
email_address	varchar(255)	No	
username	varchar(128)	No	
algorithm	varchar(128)	No	sha1
password	varchar(128)	Sí	NULL
is_active	tinyint(1)	Sí	1
is_super_admin	tinyint(1)	Sí	0
last_login	datetime	Sí	NULL
created_at	datetime	No	
updated_at	datetime	No	

Índices:

Nombre de la clave	Tipo	Único	Campo
PRIMARY	BTREE	Sí	id
email_address	BTREE	Sí	email_address
username	BTREE	Sí	username
is_active_idx_idx	BTREE	No	is_active

NOTA: Hay que aclarar que esta tabla viene con el plugin sfDoctrineGuardPlugin utilizado en esta aplicación, de uso libre bajado de la dirección <http://www.symfony-project.org/plugins/>, la misma que viene con otras tablas, como es: sf_guard_group, sf_guard_group_permission, sf_guard_permission, sf_guard_user_group, sf_guard_user_permission, las mismas que se utiliza para crear grupos de usuarios con sus respectivos permisos, para implementar la seguridad de acceso a datos.

**SUBCENTRO DE SALUD
SAN ANTONIO DE IBARRA**

Seguridad de Entrada al Sistema

Usuario:

Password:

Gráfico 4.3.: Esquema de presentación del Caso de Uso Fr2.

Fuente: Diseño Propio

Precondiciones:

- No existe una condición anterior específica para ingresar a intentar ingresar al sistema, solo si ingresó Correctamente los datos de usuario, al hacer clic en aceptar se enlazará con el caso de uso Bk1, que más adelante detallaremos.

4.1.2.4. Caso de Uso Fr3: Ingresar a Información “Nosotros”.

Descripción Breve: El Usuario Ingresa a ver detalles de Información acerca de Nosotros (Subcentro de Salud).

Flujo Básico de Eventos:

- Al hacer clic en el ícono “Nosotros”, enlaza a la página donde se presentan información de los tipos de servicios que prestan, información general sobre el edificio y otros detalles más, con una imagen actual del Subcentro de Salud.
- Para poder dirigirse a otra dirección solo se hace clic en el icono respectivo.
- Para retornar a la página principal, solo se hace clic en el logo del Subcentro de Salud ubicado en la parte superior.

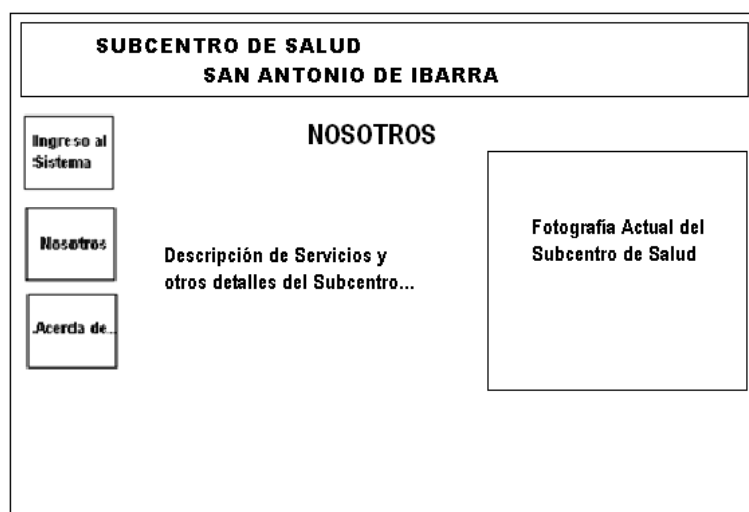


Gráfico 4.4.: Esquema de presentación del Caso de Uso Fr3.

Fuente: Diseño Propio

4.1.2.5. Caso de Uso Fr4: Ingresar a Información Acerca del Sistema.

Descripción Breve: El Usuario Ingresa a ver detalles de Información acerca del Sistema (Acerca de...), aquí se presenta el nombre de la persona que realizó, la Institución, Software que se utilizó, etc., acompañado de una imagen.

.Flujo Básico de Eventos:

- Cuando el Usuario hace clic en este ícono, le presentamos una página donde se describe detalles acerca del sistema.
- Para poder dirigirse a otra dirección solo se debe hacer clic en el icono respectivo.
- Para retornar a la página principal, solo hacemos clic en el logo del Subcentro de Salud ubicado en la parte superior.

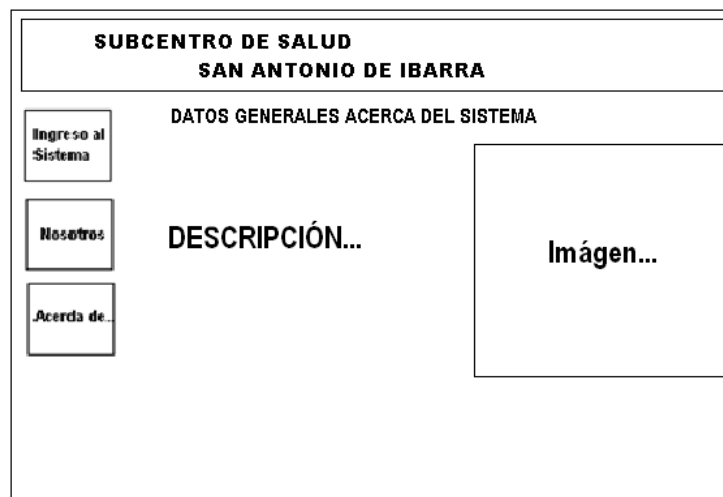


Gráfico 4.5.: Esquema de presentación del Caso de Uso Fr4.

Fuente: Diseño Propio

Precondiciones:

- No existe una condición anterior específica para ingresar a este apartado, pues cualquier usuario puede ingresar a ver esta información.

4.1.2.6. Caso de Uso Bk1: Ingresar al Sistema

Descripción Breve: El Usuario Ingresa al Sistema de Gestión de información del Subcentro.

.Flujo Básico de Eventos:

- Una vez que haya ingresado los datos correctos de Usuario y Password detallados anteriormente en el Caso de Uso Fr2, se ingresa al Sistema de Gestión de información del Subcentro, en donde se presentará nuevamente otro tipo de pantalla y en donde se encuentra íconos con enlaces a las diferentes actividades de gestión de información,
- Al hacer clic en el icono “Retorno al Inicio” se enlaza con el Caso de Uso Bk2.
- Cuando se hace clic en el icono “Pacientes/historias Clínicas” se enlazará con el Caso de Uso Bk3.
- Al hacer clic en el icono “Administración de Medicamentos”, se va al Caso de Uso Bk4.
- Cuando se hace clic en el icono de Administración de Datos personales de Médicos nos iremos al Caso de Uso Bk5.
- Al hacer clic en el icono de Administración de Turnos, vamos al Caso de Uso Bk6.
- Cuando se hace clic en el icono Reportes se enlaza con el Caso de Uso Bk7.
- Al hacer clic en el icono Mantenimiento de datos extras se va al Caso de Uso Bk8.
- Y cuando se hace clic en el icono Crear/Restaurar Copias de Seguridad de la Base de Datos se va al Caso de Uso Bk9.

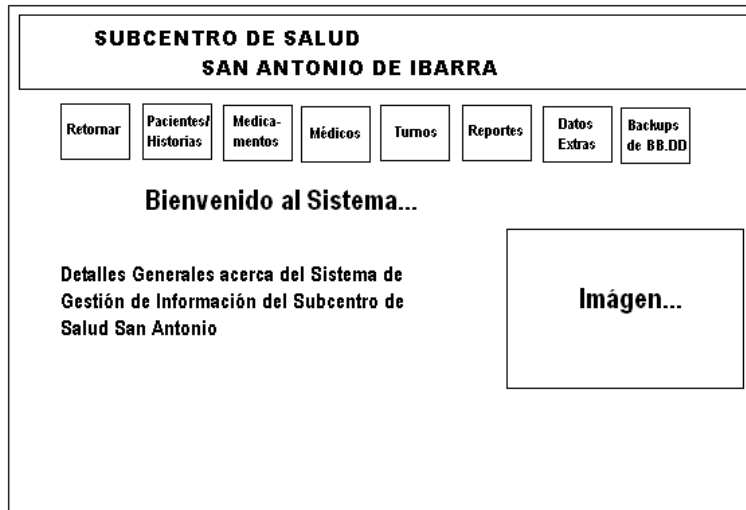


Gráfico 4.6.: Esquema de presentación del Caso de Uso Bk1.

Fuente: Diseño Propio

Precondiciones:

- El usuario debe ser registrado en la tabla de usuarios con sus respectivos permisos.
- Debió haber ingresado los datos de usuario y contraseña en la pantalla del caso de uso **Fr2** en forma correcta.

4.1.2.7. Caso de Uso Bk2: Regresar al inicio

Descripción Breve: El Usuario Regresa al Inicio del sistema.

.Flujo Básico de Eventos:

- Como se presenta en el esquema del **Gráfico 4.5.**, al hacer clic en el ícono Retornar, regresará al inicio del Sistema como se indica en el esquema del Caso de Uso Fr1, en el caso de que no tenga nada más que hacer en el Sistema de Gestión de Información del Subcentro de Salud.

4.1.2.8. Caso de Uso Bk3: Administrar Información de Historias Clínicas.

Descripción Breve: El Usuario Ingresa a la administración de Información de historias Clínicas.

Flujo Básico de Eventos:

- *Muestra el Listado de Pacientes:* El listado de pacientes con sus campos básicos, así como el número de Historia, cédula, Nombres, Apellidos y Dirección, a más de acciones básicas que podemos hacer con cada registro (Eliminar o Editar) en bloques de 10, pues al listar todos de una sola, la página se alargaría demasiado ya que existen una totalidad de más de 20.000 historias clínicas.
- *Realizar Búsqueda de Pacientes:* En este bloque se da opciones de cómo puede realizar la búsqueda de un determinado Paciente, esta parte es muy importante para poder trabajar en una forma ágil en el Subcentro, pues cada vez que se va a dar un turno, lo primero que hacen las enfermeras o personas encargadas de dar turnos es buscar el número de Historia del paciente para ver si está registrado en la Base de Datos. En este bloque se puede dar opciones al pedir sus datos; Puede ser su Número de Historia, su Cédula, sus Nombres, sus Apellidos o su Dirección o una combinación de ellos para que su búsqueda sea más acertada.
- Al hacer clic en el icono o enlace “Crear Nuevo Registro” se enlaza con el Caso de Uso Bk10.
- Cuando se hace clic en el icono Salir se va a regresar al Caso de Uso Bk2.

**SUBCENTRO DE SALUD
SAN ANTONIO DE IBARRA**

Nuevo
Registro

Salir

LISTADO DE PACIENTES

Historia	Cédula	Nombres	Apellidos	Dirección	Acciones
					Modificar
					Eliminar
⋮					

BÚSQUEDA DE PACIENTES

Historia:

Cédula:

Apellidos:

Nombres:

Dirección:

Limpiar/Restablecer

« »

Acciones: Eliminar seleccionados

Ejecutar Accion

+ Crear Nuevo Paciente

Gráfico 4.7.: Esquema de presentación del Caso de Uso Bk3.

Fuente: Diseño Propio

La estructura de la tabla para registrar historias clínicas fue basada a la hoja del formulario para registrar las nuevas historias clínicas que se están manejando actualmente en el Subcentro de Salud “San Antonio” y tiene la siguiente estructura.

pacientes

Campo	Tipo	Nulo	Predeterminado	Enlaces a
id_historia	int(11)	No		
id_historiaclin	varchar(6)	No		
cedula	varchar(15)	Sí	NULL	
pasaporte	varchar(45)	Sí	NULL	
apellidos	varchar(60)	Sí	NULL	
nombres	varchar(60)	Sí	NULL	
direccion	varchar(80)	Sí	NULL	
telefono	varchar(15)	Sí	NULL	
fecha_nac	date	Sí	NULL	
lugar_nac	varchar(60)	Sí	NULL	
sexo	varchar(1)	Sí	NULL	
estado_civil	varchar(3)	Sí	NULL	
fecha_admision	date	Sí	NULL	
antecedentes	text	Sí	NULL	
ocupacion	varchar(60)	Sí	NULL	
lugar_trabajo	varchar(60)	Sí	NULL	
tiposegsalud	varchar(60)	Sí	NULL	
referidode	varchar(60)	Sí	NULL	
nombrepresentesco	varchar(60)	Sí	NULL	
direc_parentesco	varchar(60)	Sí	NULL	
telf_parentesco	varchar(15)	Sí	NULL	
comentarios	varchar(250)	Sí	NULL	
barrios_id_barrio	int(11)	Sí	NULL	barrios -> id_barrio
parroquias_id_parroquias	int(11)	Sí	NULL	parroquias -> id_parroquias
cantones_id_canton	int(11)	Sí	NULL	cantones -> id_canton
provincias_id_provincia	int(11)	Sí	NULL	provincias -> id_provincia
zonas_id_zona	int(11)	Sí	NULL	zonas -> id_zona
nacionalidades_id_nacionalidad	int(11)	Sí	NULL	nacionalidades -> id_nacionalidad
grupos_culturales_id_grupocultural	int(11)	Sí	NULL	grupos_culturales -> id_grupocultural
parentesco_id_parentesco	int(11)	Sí	NULL	parentesco -> id_parentesco
instruccion_id_instruccion	int(11)	Sí	NULL	instruccion -> id_instruccion

Índices:

Nombre de la clave	Tipo	Único	Campo
PRIMARY	BTREE	Sí	id_historia
id_historiaclin_idx	BTREE	No	id_historiaclin
barrios_id_barrio_idx	BTREE	No	barrios_id_barrio
parroquias_id_parroquias_idx	BTREE	No	parroquias_id_parroquias
cantones_id_canton_idx	BTREE	No	cantones_id_canton
provincias_id_provincia_idx	BTREE	No	provincias_id_provincia
zonas_id_zona_idx	BTREE	No	zonas_id_zona

nacionalidades_id_nacionalidad_idx	BTREE	No	nacionalidades_id_nacionalidad
grupos_culturales_id_grupocultural_idx	BTREE	No	grupos_culturales_id_grupocultural
parentesco_id_parentesco_idx	BTREE	No	parentesco_id_parentesco
instruccion_id_instruccion_idx	BTREE	No	instruccion_id_instruccion

Precondiciones:

- El usuario debió haberse creado con privilegios suficientes para ingresar a este módulo.
- Se tiene casos en donde el usuario estará restringido en algunas actividades, como es Eliminar o modificar debido a que puede que sus privilegios no le permitan.

4.1.2.9. Caso de Uso Bk4: Administrar Datos de Medicamentos.

Descripción Breve: El Usuario Ingresa a la Administración de datos básicos de Medicamentos que se manejan en el Subcentro.

Descripción Básica de Eventos:

- *Listado de Medicamentos:* El listado de medicamentos con sus campos básicos, así como: Nombre de Medicamento, Descripción y Categoría, a más de acciones básicas que podemos hacer con cada registro (Eliminar o Editar) en bloques de 10, pues pueden existir una gran cantidad de medicamentos.
- *Búsqueda de Medicamentos:* En este bloque se da opciones de cómo puede realizar la búsqueda de un determinado Medicamento, esto con el fin de averiguar si está registrado en la Base de Datos y ver si existe en stock de Farmacia. De la misma manera que las anteriores en este bloque se puede dar opciones de búsquedas, como es: Nombre del Medicamento, Descripción y Categoría del Medicamento o una combinación de ellos para que la búsqueda sea más exacta.
- Al hacer clic en el ícono de Administración de Medicamentos, el usuario ingresará a manipular datos básicos de medicamentos que existen en Farmacia, por lo que de esta manera se puede estar informados si un respectivo medicamento existe en stock, los mismos que la mayoría de ellos le regalan al paciente.
- Cuando se hace clic en el Ícono Consumo Mensual de Medicamentos se va al Caso de Uso Bk11.

- Al hacer clic en el Ícono Consumo Diario de Medicamentos se va al Caso de Uso Bk12.
- Cuando se hace clic en el Enlace Crear Nuevo Registro se enlazará con el Caso de Uso Bk10.
- Al hacer clic en el Ícono Salir se regresará al Caso de Uso Bk2.

Gráfico 4.8.: Esquema de presentación del Caso de Uso Bk4.

Fuente: Diseño Propio

La estructura de la tabla de medicamentos es la siguiente:

medicamento

Campo	Tipo	Nul o	Predetermina do	Enlaces a
id_medicamento	int(11)	No		
nombre_med	varchar(60)	Sí	NULL	
descripcion	varchar(250)	Sí	NULL	
categoria_medicamento_id_categoria	int(11)	Sí	NULL	categoria_medicamento -> id_categoria

Índices:

Nombre de la clave	Tipo	Único	Campo
PRIMARY	BTREE	Sí	id_medicamento
categoria_medicamento_id_categoria_idx	BTREE	No	categoria_medicamento_id_categoria

4.1.2.10. Caso de Uso Bk5: Administración de datos personales de médicos.

Descripción Breve: El Usuario Ingresará a la Administración de Datos Personales de

Médicos, en donde podrá modificar o eliminar o simplemente activarlos (en el caso de que ingresen a prestar su servicio) o desactivarlos (en el caso de que deje de prestar sus servicios).

Descripción Básica de Eventos:

- *Listado de Médicos:* El listado de médicos con sus campos básicos: Cédula, Apellidos, Nombres, Teléfono, Celular, Especialidad, a más de acciones básicas que podemos hacer con cada registro (Eliminar o Editar) en bloques de 10, pues de igual manera pueden existir una gran cantidad de médicos registrados, ya que los médicos no son permanentes y cada tiempo los cambian.
- *Búsqueda de Médicos:* En este bloque se da opciones de cómo puede realizar la búsqueda de un determinado Médico, esto con el fin de averiguar si está registrado en la Base de Datos.
- De igual manera que las anteriores se puede realizar las búsquedas por: Cédula, Apellidos, Nombres, y Especialidad o una combinación de ellos para que su búsqueda sea más exacta.
- Al hacer clic en el Ícono Retornar se Regresa al Caso de Uso Bk2.
- Cuando hagamos clic en el Enlace/ícono Crear Nuevo Registro se retornará al Caso de Uso Bk10.

**SUBCENTRO DE SALUD
SAN ANTONIO DE IBARRA**

Retomar Nuevo Registro

LISTADO DE MÉDICOS

Cédula	Apellidos	Nombres	Teléfono	Celular	Especialidad	Acciones
						<small>Modificar</small> <small>Eliminar</small>
⋮						

BÚSQUEDA DE MÉDICOS

Historia:

Cédula:

Apellidos:

Nombres:

Dirección:

Limpiar/Restablecer

⏪ ⏩

Acciones: Eliminar seleccionados + Crear Nuevo Médico

Gráfico 4.9.: Esquema de presentación del Caso de Uso Bk5.

Fuente: Diseño Propio

La tabla para registrar datos personales de médicos tiene la siguiente estructura:

medicos

Campo	Tipo	Nulo	Predeterminado
id_medico	int(11)	No	
cedula	varchar(11)	No	
apellidos	varchar(60)	Sí	NULL
nombres	varchar(60)	Sí	NULL
telf_casa	varchar(15)	Sí	NULL
telf_celular	varchar(15)	Sí	NULL
sexo	varchar(1)	Sí	NULL
especialidad	varchar(45)	Sí	NULL

Índices:

Nombre de la clave	Tipo	Único	Campo
PRIMARY	BTREE	Sí	id_medico
cedula_idx	BTREE	No	cedula

Precondiciones:

- El usuario debió haberse creado con privilegios suficientes para ingresar a este módulo.
- Tenemos casos en donde el usuario estará restringido en algunas actividades, como es Eliminar o modificar debido a que puede que sus privilegios no le permitan.

4.1.2.11. Caso de Uso Bk6: Administración de Turnos.

Descripción Breve: El Usuario Ingresa a la Administración de Turnos, en donde puede eliminar o modificar turnos entregados en ese día.

Descripción Básica de Eventos:

- *Listado de Medicamentos:* El listado de turnos con sus campos básicos, así como: Nº de Turno, Fecha, Historia Clínica, a más de acciones básicas que podemos hacer con cada registro (Eliminar o Editar) en bloques de 10, pues pueden existir una gran cantidad de Turnos Registrados, pues el usuario deberá borrar cada cierto tiempo ya que a diario se registran gran cantidad de turnos.
- *Búsqueda de Turnos:* En este bloque se da opciones de cómo puede realizar la búsqueda de un determinado Turno dado en una determinada fecha, esto con el

fin de averiguar si está registrado en la Base de Datos, si es que aún no se han eliminado.

- De igual manera que las anteriores también se puede realizar búsquedas por: N° de Turno, Fecha, Historia Clínica o una combinación de ellos para que la búsqueda sea más exacta.
- Cuando se hace clic en el Ícono Retornar se Regresa al Caso de Uso Bk2
- Al hacer clic en el Enlace/ícono Crear Nuevo Registro se podrá enlazarse con el Caso de Uso Bk10.

Gráfico 4.10.: Esquema de presentación del Caso de Uso Bk6.
 Fuente: Diseño Propio

La estructura de la tabla de turnos es la siguiente:

turnos

Campo	Tipo	Nulo	Predeterminado	Enlaces a
id	bigint(20)	No		
n_turno	int(11)	Sí	NULL	
fecha	datetime	Sí	NULL	
tipo_turnos_id_tipo_turno	int(11)	No		tipo_turnos -> id_tipo_turno
pacientes_id_historia	int(11)	No		pacientes -> id_historia
medicos_id_medico	int(11)	No		medicos -> id_medico

Índices:

Nombre de la clave	Tipo	Único	Campo
PRIMARY	BTREE	Sí	id
tipo_turnos_id_tipo_turno_idx	BTREE	No	tipo_turnos_id_tipo_turno
pacientes_id_historia_idx	BTREE	No	pacientes_id_historia
medicos_id_medico_idx	BTREE	No	medicos_id_medico

Precondiciones:

- El usuario debió haberse creado con privilegios suficientes para ingresar a este módulo.
- Los casos en donde el usuario estará restringido en algunas actividades, como es Eliminar o modificar debido a que puede que sus privilegios no le permitan.

4.1.2.12. Caso de Uso Bk7: Visualizar Reportes.

Descripción Breve: El Usuario Ingresa a Reportes, en donde podrá ver reportes estadísticos en pantalla o enviar a la impresora.

Descripción Básica de Eventos:

- Cuando se hace clic en el icono Estadísticas se entrará a la pantalla en donde se puede visualizar reportes estadísticos en:
 - Pastel, si es que se hace clic en el botón de opción Pastel.
 - Barras, si es que se escoge el botón de opción Barras.

Pero también debemos escoger el criterio de Grafico, es decir:

- Sexo, si es que se va a ver el gráfico con este criterio.
- Barrios, si se va a mostrar el gráfico con este criterio.
- Edades, de la misma manera, se visualizará el gráfico de este criterio.
- Motivos de Consulta, se mostrara el gráfico estadístico con este criterio.

También hay que hacer notar que se puede combinar entre dos criterios, pero solo con el gráfico de Barras, en pastel no acepta esta situación, por lo que muestra un mensaje de error.

- Al hacer clic en el icono Reporte de Turnos, este Reporte sacará los turnos dados en el día, a un documento PDF, con el que podemos guardarlo o imprimirlo directamente.
- Y, al hacer clic en el icono Salir de Reporte, regresa al Caso de Uso BK2

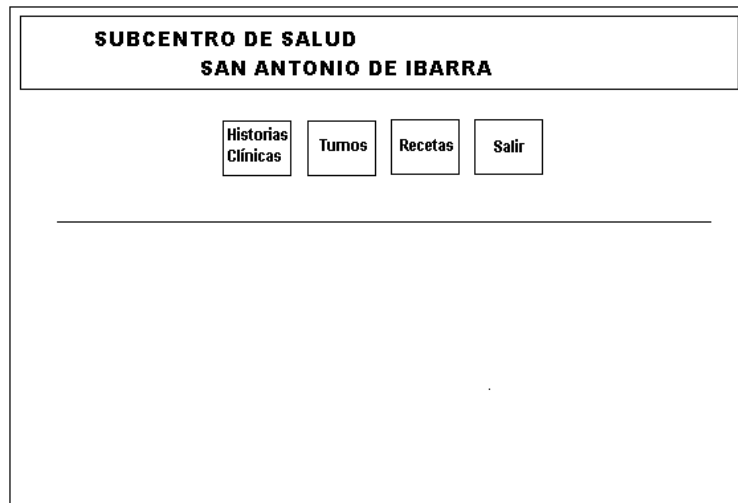


Gráfico 4.11.: Esquema de presentación del Caso de Uso Bk7.

Fuente: Diseño Propio

4.1.2.13. Caso de Uso Bk8: Administrar datos extras para la base de datos.

Descripción Breve: En este bloque el Usuario Ingresa a la Administración de Datos extras los mismos que se utilizan en la Base de Datos, estos datos son

Datos extras para Historias Clínicas.

- Nuevo Barrio
- Nueva Parroquia
- Nuevo Cantón
- Nueva Provincia
- Nueva Zona
- Nueva Nacionalidad
- Nuevo Grupo Cultural
- Nuevo Tipo de Parentesco
- Nuevo Tipo de Instrucción

Otros Datos Extras:

- Nueva categoría de Medicamentos
- Nuevo Motivo de Consulta.

Descripción Básica de Eventos:

- Al Hacer clic en el icono Nuevo Barrio, se crea un Nuevo barrio para hacer referencia desde la tabla de pacientes en la creación de un nuevo registro.
- Cuando se hace clic en el icono Nueva Parroquia, creamos una nueva Parroquia para hacer referencia desde la tabla de pacientes en la creación de un nuevo registro.
- Al hacer clic en el icono Nuevo Cantón, se crea un nuevo cantón, para hacer referencia desde la tabla de pacientes en la creación de un nuevo registro.
- Cuando se hace clic en el icono Nueva Provincia, se registra una nueva provincia en su respectiva tabla, para hacer referencia desde la tabla de pacientes en la creación de un nuevo registro.
- Al hacer clic en el icono Nueva Zona, se crea un nuevo registro en la tabla Zonas, para hacer referencia desde la tabla de pacientes en la creación de un nuevo registro.
- Cuando se hace clic en el icono Nueva Nacionalidad, se crea un nuevo registro en la tabla Nacionalidades, para hacer referencia desde la tabla de pacientes en la creación de un nuevo registro.
- Al hacer clic en el icono Nuevo Grupo Cultural, se crea un nuevo registro en la tabla grupos_culturales, para hacer referencia desde la tabla de pacientes en la creación de un nuevo registro.
- Cuando se hace clic en el icono Nuevo Tipo de Parentesco, se crea un nuevo registro en la tabla Parentesco, para hacer referencia desde la tabla de pacientes en la creación de un nuevo registro.
- Al hacer clic en el icono Nuevo Tipo de Instrucción, se crea un nuevo registro en la tabla Instrucción, para hacer referencia desde la tabla de pacientes en la creación de un nuevo registro
- Cuando se hace clic en el icono Nueva Categoría de Medicamentos, se creará una nueva categoría para los medicamentos, esto con el fin de hacer la referencia desde la tabla Medicamentos en la creación de un nuevo registro.
- Al hacer clic en el icono Crear Nuevos Motivos de Consultas, se creará un nuevo Motivo de consultas, este tipo de registros servirá para hacer la referencia desde la tabla Consultas en la creación de nuevos registros.

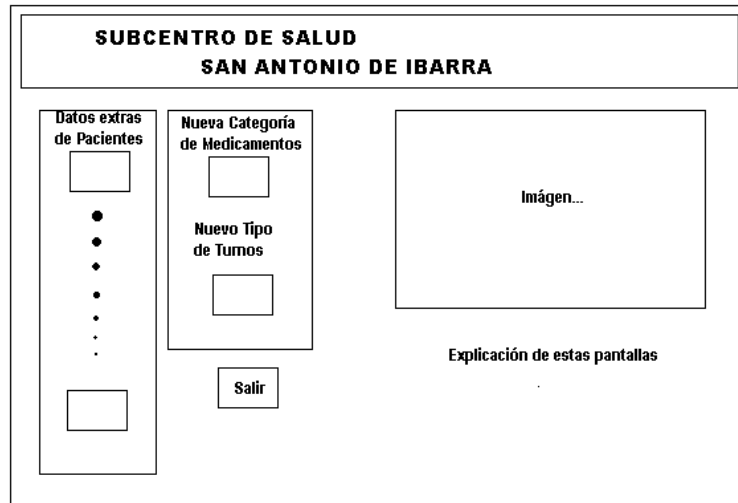


Gráfico 4.12.: Esquema de presentación del Caso de Uso Bk8.

Fuente: Diseño Propio

Subflujos:

- Para crear los datos anteriores se lo toma de esta forma, ya que estos datos pueden variar en su cantidad, por lo que se dedica una tabla, las mismas que todas tendrán o estarán conformadas por los siguientes campos:
 - Nombre
 - Comentarios

Y el Esquema de presentación en su página va a ser similares en todas:

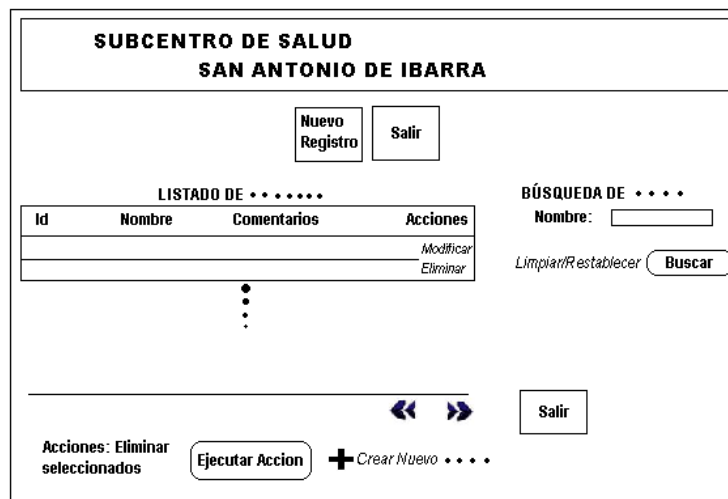


Gráfico 4.13.: Esquema de presentación común para todas las acciones de las pantallas de Mantenimiento de datos extras.

Fuente: Diseño Propio

Tomamos la estructura de la tabla barrios como ejemplo, para mostrar la estructura en todas las tablas anteriormente enunciadas.

barrios

Campo	Tipo	Nulo	Predeterminado
id_barrio	int(11)	No	
nombre	varchar(80)	Sí	NULL
comentarios	varchar(250)	Sí	NULL

Índices:

Nombre de la clave	Tipo	Único	Campo
PRIMARY	BTREE	Sí	id_barrio

4.1.2.14. Caso de Uso Bk9: Crear y restaurar Backups.

Descripción Breve: Al ingresar a este módulo el usuario Ingresa a Crear una copia de la Base de Datos a la fecha actual, o Restaurar copias a una fecha anterior.

Descripción Básica de Eventos:

- Al hacer clic en el icono *Crear Backup*, da la posibilidad de crear un archivo de seguridad de la Base de datos en cuyo nombre estará compuesto por la fecha y hora actuales a la creación de los mismos.
- Cuando hacemos clic en el icono *Restaurar Backup*, se puede restaurar un archivo de copia de la Base de Datos a una determinada Fecha, en el caso que por A o B circunstancias se ha dañado la actual.
- Y, al hacer clic en el icono *Salir* se retornará al Caso de Uso Bk2.

Precondiciones:

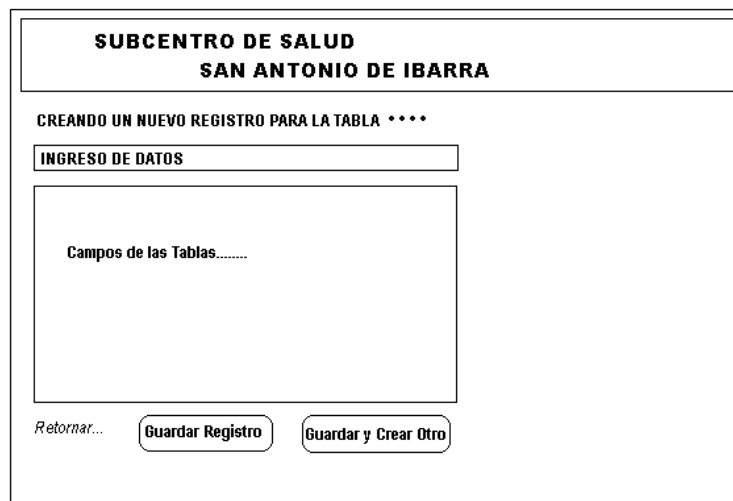
- El usuario debe ser un usuario registrado con privilegios suficientes para ingresar a este módulo.

4.1.2.15. Caso de Uso Bk10: Crear nuevos registros.

Descripción Breve: El Usuario Ingresa a la creación de Nuevos Registros, el enlace que me permitirá crear nuevos registros pueden ser íconos o enlaces, dependiendo en que caso de Uso nos encontremos.

Descripción Básica de Eventos:

- Se representa en forma global este caso de uso, ya que todas estas páginas van a tener el mismo esquema y el propósito va a ser el mismo, solo que los registros van a ser grabados en sus respectivas tablas, y además que por supuesto que sus registros van a ser diferentes, es así que este caso de uso sirve para las siguientes tablas:
 - Historias Clínicas.
 - Medicamentos
 - Médicos
 - Ingreso Mensual de Medicamentos
 - Ingreso Diario de Medicamentos
 - Turnos



El gráfico muestra un formulario web con el siguiente contenido:

- Encabezado: **SUBCENTRO DE SALUD SAN ANTONIO DE IBARRA**
- Subencabezado: **CREANDO UN NUEVO REGISTRO PARA LA TABLA ******
- Formulario: **INGRESO DE DATOS**
- Área de campos: **Campos de las Tablas.....**
- Botones de acción: **Retomar...**, **Guardar Registro**, **Guardar y Crear Otro**

Gráfico 4.14.: Esquema de presentación del Caso de Uso Bk10.

Fuente: Diseño Propio

Precondiciones:

- El usuario debió haberse creado con privilegios suficientes para poder crear nuevos registros o ingresar al respectivo módulo.

4.1.2.16. Caso de Uso Bk11: Registrar ingreso mensual de medicamentos.

Descripción Breve: Cuando el usuario ingresa a registrar el Ingreso Mensual de Medicamentos, ingresa datos de un determinado medicamento, ya que cada mes viene al Subcentro una determinada cantidad de medicamentos a farmacia, estos deben ser registrados para mantener el control de entradas de medicamentos.

Descripción Básica de Eventos:

- *Listado de Registros de Ingresos Mensuales:* El listado de registros de Ingresos Mensuales de Medicamentos con sus campos básicos, así como: Fecha, Medicamento, Valor Factura, Consumo y Saldo Final a más de acciones básicas que podemos hacer con cada registro (Eliminar o Editar) en bloques de 10, para controlar que no se alargue y se desordene la página.
- *Búsqueda de Registros:* En este bloque se podrá realizar búsquedas por: Un rango de Fechas y por Medicamento o la combinación de las dos, en el caso de que se quiera averiguar un determinado registro de algún ingreso mensual de una determinada fecha.
- Cuando se hace clic en el Ícono Retornar se retornará al Caso de Uso Bk2.
- Al hacer clic en el Ícono/Enlace Crear Nuevo Registro se va a dirigir al Caso de Uso Bk10.

**SUBCENTRO DE SALUD
SAN ANTONIO DE IBARRA**

Retornar
Nuevo Registro

REGISTRO MENSUAL DEL INGRESO DE MEDICAMENTOS

Fecha	Medicamento	Valor fact.	Consumo	Saldo Fin.	Acciones
					<small>Modificar</small> <small>Eliminar</small>
⋮					

BÚSQUEDA DE REGISTROS

Fecha: Desde:
Hasta:

Medicamento

Limpia/Restablecer

⏪ ⏩

Acciones: Eliminar seleccionados

+ Crear Nuevo Registro

Gráfico 4.15.: Esquema de presentación del Caso de Uso Bk11.

Fuente: Diseño Propio

La estructura de la tabla para registrar los medicamentos es la siguiente:

medicamento

Campo	Tipo	Nulo	Predeterminado	Enlaces a
id_medicamento	int(11)	No		
nombre_med	varchar(60)	Sí	NULL	
descripcion	varchar(250)	Sí	NULL	
categoria_medicamento_id_categoria	int(11)	Sí	NULL	categoria_medicamento -> id_categoria

Índices:

Nombre de la clave	Tipo	Único	Campo
PRIMARY	BTREE	Sí	id_medicamento
categoria_medicamento_id_categoria_idx	BTREE	No	categoria_medicamento_id_categoria

Precondiciones:

- Para realizar el ingreso mensual de medicamentos, el usuario debió haberse creado con privilegios suficientes para poder crear nuevos registros.
- Debe estar dentro del módulo Administración de Datos de Medicamentos.
- Debió haber hecho clic en el icono Consumo Mensual de Medicamentos.

4.1.2.17. Caso de Uso Bk12: Registrar consumo diario de medicamentos.

Descripción Breve: El Usuario Ingresa a registrar la salida diaria de medicamentos, estos deben ser registrados para mantener el control de salida de medicamentos..

Descripción Básica de Eventos:

- *Listado del Consumo Diario de Mensuales:* Este listado también tienen sus campos básicos, así como: Fecha, Cantidad, Valor Unitario, Medicamento y Descripción, a más de acciones básicas que se puede hacer con cada registro (Eliminar o Editar) en bloques de 10, para controlar que no se alargue y se desordene la página.
- *Búsqueda de Registros:* En este bloque se podrá realizar búsquedas por: Un rango de Fechas y por Medicamento (similar al caso de Uso Bk11) o la combinación de las dos, en el caso de que se quiera averiguar un determinado registro de una determinada fecha.
- Cuando se hace clic en el Ícono Retornar, el control va a retornar al Caso de Uso Bk2.
- Al hacer clic en el Ícono/Enlace se va a crear un Nuevo Registro (Caso de Uso Bk10).

**SUBCENTRO DE SALUD
SAN ANTONIO DE IBARRA**

Retornar
Nuevo Registro

LISTADO DE REGISTROS DE CONSUMO DIARIO DE MEDICAMENTOS

Fecha	Cantidad	Val. Unit	Descripción	Medicamento	Acciones
					<small>Modificar</small> <small>Eliminar</small>
⋮					

BÚSQUEDA DE REGISTROS

Fecha: Desde:

Hasta:

Medicamento

Limpiar/Restablecer Buscar

⏪ ⏩

Acciones: Eliminar seleccionados

Ejecutar Accion

+

Crear Nuevo Registro

Gráfico 4.16.: Esquema de presentación del Caso de Uso Bk12.

Fuente: Diseño Propio

La estructura de la tabla en donde se va a registrar las salidas diarias de medicamentos tiene la siguiente estructura:

cons_diario_medicam

Campo	Tipo	Nulo	Predeterminado	Enlaces a
id	bigint(20)	No		
fecha	date	Sí	NULL	
cantidad	int(11)	Sí	NULL	
valor_unitario	float	Sí	NULL	
descripcion	varchar(250)	Sí	NULL	
medicamento_id_medicamento	int(11)	No		medicamento ->

id_med medicamento

Índices:

Nombre de la clave	Tipo	Único	Campo
PRIMARY	BTREE	Sí	id
medicamento_id_med medicamento_idx	BTREE	No	medicamento_id_med medicamento

Precondiciones:

- Para realizar el registro diario de consumo de medicamentos, el usuario debió haberse creado con privilegios suficientes para poder crear nuevos registros.
- Debe estar dentro del módulo Administración de Datos de Medicamentos.
- Debió haber hecho clic en el icono Consumo Diario de Medicamentos.

4.1.3. Redefinición de Análisis y Diseño.

Después de haber analizado como se realiza el proceso de atención al paciente actualmente, esto lo esquematizamos en el Gráfico 3.1, este proceso de atención se ha mantenido desde hace un buen tiempo atrás, pero los requerimientos de salud aumenta con el crecimiento de la población San Antonense, por lo que este proceso no rinde como se quisiera.

En este documento no proponemos cambiar este proceso, solo se propone introducir una herramienta más para que se adopte en el punto crítico de atención, que en este caso es ESTADÍSTICA, en donde se maneja todos los datos importantes del Subcentro (Historias Clínicas de Pacientes) y en donde dan los turnos, motivo por el cual el paciente acude a esta casa de salud y en donde se localiza y se ha detectado la ineficiente atención en cuanto a extender turnos.

Se tiene la seguridad que una herramienta de este tipo ayudará a brindar una mejor y ágil atención al paciente.

Un Sistema de Cómputo que en el mejor de los casos trabajaría en red para que se pudiera acceder a la base de datos desde cualquier punto del Subcentro, pues el sistema no solo solucionará el gran problema de almacenamiento y búsquedas de Historias Clínicas, sino que también tratará de solucionar el problema de búsquedas de Medicamentos en Farmacia, pues al no saber el médico de la existencia de un medicamento en stock de farmacia tiene que dirigirse personalmente a averiguarlo en este sector del Subcentro, haciendo perder un buen tiempo para la atención de pacientes,

impacientándolos aún más. Con este sistema, lo ideal sería que desde su consultorio podría averiguarlo, si es que aquí tuviera instalado un terminal conectado a la Base de datos del Sistema.

Pueden haber otros puntos que solucionaría con la adopción de este sistema como una herramienta de trabajo, pues tendríamos ordenado/almacenado los datos personales de cada médico que pasa por esta casa de salud, ya que ningún médico está permanentemente varios años aquí, todos llegan a cumplir su trabajo solo por un determinado tiempo estos pueden ser meses o años en el mejor de los casos. No podríamos decir que este es un problema, pero esto serviría como datos historiales para el Subcentro, ya que actualmente no cuenta con un registro propio de los datos personales de médicos que han pasado por el Subcentro de Salud, estos datos solo son manejados por la dirección de salud.

Como la mayoría de sistemas, estos deben emitir reportes necesarios para el Subcentro, como son de datos personales de pacientes (Historias Clínicas), de recetas o turnos, nosotros creemos que una buena forma de emitir reportes es enviándolo a un archivo PDF, ya que a la falta de impresora se puede llevar grabándolo en un dispositivo de almacenamiento (disco duro, Flash memory, CD, etc.), pero desde este mismo documento podemos enviarlo a imprimir en el caso de que contemos con la impresora.

Por supuesto que para mantener la integridad y seguridad en los datos grabados en nuestra base de datos, también debe incrementarse un módulo que permita Crear una copia de seguridad cada vez que el usuario del sistema crea conveniente, es decir cada lapso de tiempo, el usuario más experimentado debería adoptar esta costumbre como una especie de política, pues nadie sabe lo que puede pasar con la base de datos o con el sistema o con el mismo equipo de cómputo que alberga la base de datos, con esto aseguramos que nuestros datos estén seguros. Pero de la misma forma que el sistema sea capaz de crear la base de datos, también debe ser capaz de restablecer una copia de seguridad, es por eso que en el sistema debe tener esta opción en donde también debe ser capaz de elegir el archivo al que quiere restablecer, pues es aconsejable que se restaure a la última copia de seguridad, pero también puede ser que a otra copia, dependiendo del daño que tuvo el sistema, esta decisión debe tomarlo el usuario más experimentado o que tenga más conocimiento de los datos o del manejo del sistema.

Por último, este sistema se podría subir a la web para su uso desde cualquier lugar conectado a Internet, pero por motivos de seguridad en los datos de Historias clínicas de pacientes en especial y por falta de recursos económicos, no se proyecta subir a la web, pero el mismo sistema si debe ser capaz de trabajar en red por lo menos dentro del Subcentro.

Como dijimos anteriormente, se propone introducir una herramienta más que ayude en el trabajo diario del Subcentro, es decir un Sistema Computacional, pues el presente proceso de atención es el óptimo aún, y se podría ayudar aún más incrementando el sistema, pero se debe cambiar la mentalidad de las personas que van a manejar el sistema, pues están ya acostumbradas a estar sin esta herramienta y se muestran reacias a adoptar al sistema para realizar su trabajo diario.

El siguiente esquema muestra el trabajo del sistema en el Subcentro de Salud:

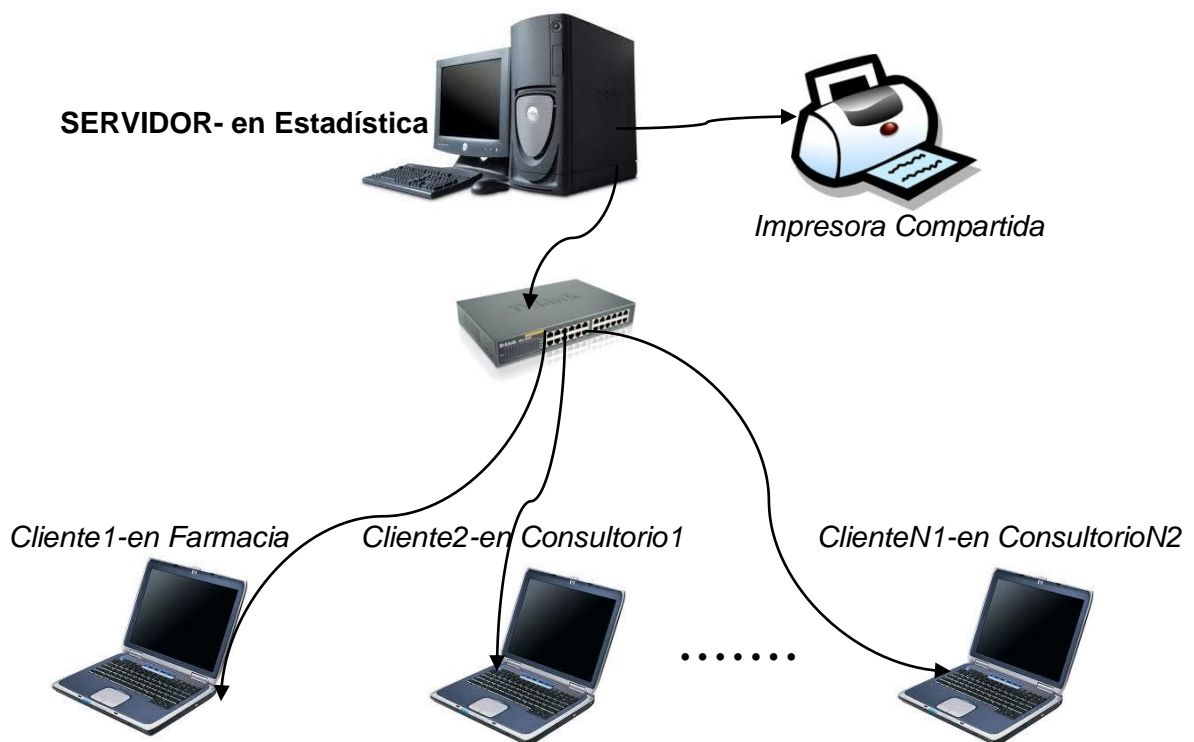


Gráfico 4.17.: Esquema de Trabajo del Sistema de Computo en el Subcentro de Salud.

Fuente: Diseño Propio

4.1.4. Ficha de Historias Clínicas

Para la creación de las fichas de Historias Clínicas, están basadas a las fichas ya existentes en el Subcentro que se manejan en forma manual.

Esta ficha es la de mayor tamaño en cuanto al número de campos que tiene, en comparación con las demás fichas que se ingresa en la base de datos, ya que se puso mayor énfasis en esta para que el sistema sea de mayor ayuda en el Subcentro, los campos de esta se refiere a datos personales del paciente y que pueden o no ser llenados en su totalidad, por ejemplo del pasaporte, esto sería solo el caso de que el paciente no tenga nuestra nacionalidad, en este caso la cédula de identidad va a quedar en blanco, pero el caso más común es que se ingresa la cédula de identidad y el campo pasaporte va a estar en blanco.

Los campos a ingresar en la ficha de Historias Médicas lo mostramos en el Gráfico 4.17:

INGRESO DE DATOS	
Historia	<input type="text"/>
Cédula	<input type="text"/>
Pasaporte	<input type="text"/>
Apellidos	<input type="text"/>
Nombres	<input type="text"/>
Dirección	<input type="text"/>
Teléfono	<input type="text"/>
Fecha nac	<input type="text"/> - <input type="text"/> - <input type="text"/>
Lugar nac	<input type="text"/>
Sexo	<input type="text"/>
Estado civil	<input type="text"/>
Fecha admision	<input type="text"/> - <input type="text"/> - <input type="text"/>
Antecedentes	<input type="text"/>
Ocupacion	<input type="text"/>
Lugar trabajo	<input type="text"/>
Tiposeguridad	<input type="text"/>
Referidode	<input type="text"/>
Nombrepresentesco	<input type="text"/>
Direc presentesco	<input type="text"/>
Telf presentesco	<input type="text"/>
Comentarios	<input type="text"/>
Barrio	<input type="text"/>
Parroquia	<input type="text"/>
Cantón	<input type="text"/>
Provincia	<input type="text"/>
Zona	<input type="text"/>
Nacionalidad	<input type="text"/>
Grupo Cultural	<input type="text"/>
Parentesco	<input type="text"/>
Instrucción	<input type="text"/>

Gráfico 4.18.: Campos para ingreso de datos en la Ficha de Historias Clínicas.

Fuente: Pantalla capturada del Sistema.

- **Historia:** Este va a ser la identificación del paciente para el Subcentro de Salud.
- **Cédula:** Va a ser el número de cédula de identidad del paciente, en caso de ser extranjero, o también en el caso de ser menor de edad y no contar con cédula de identidad este campo va a quedar vacío.

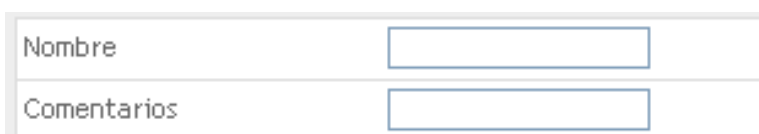
- **Pasaporte:** Este campo será llenado en el caso de un paciente extranjero y que solo cuenta con pasaporte, en este caso el campo cédula va a quedar en blanco.
- **Apellidos:** El dato de los apellidos del paciente.
- **Nombres:** Los nombres del paciente.
- **Dirección:** La dirección de residencia del paciente.
- **Teléfono:** Aquí puede llenarse con el dato de teléfono convencional o celular.
- **Fecha nac:** Se refiere a la fecha de nacimiento del paciente, este dato se puede seleccionar de la lista de selección, en el campo de año notaremos que vamos a empezar desde el año 1910, porque pensamos que es muy difícil que existan historias de pacientes que hayan nacido más antes de este año.
- **Lugar nac:** Este campo se utiliza para ingresar el lugar donde nació el paciente.
- **Sexo:** De la misma manera aquí presentamos una lista de selección en donde podemos seleccionar M (Masculino) o F (Femenino).
- **Estado Civil:** También es un dato que seleccionaremos desde una lista de alternativas, las cuales son: SOL(Soltero/a), CAS(Casado/a), DIV(Divorciado/a), VIU(Viudo/a), UL(Unión Libre), Más(En el caso de que su estado civil no encuentre en la lista presentada).
- **Fecha de Admisión:** En este campo también podemos ingresar este dato dando la oportunidad de seleccionar desde la lista de alternativas, en el campo año damos la oportunidad de seleccionar desde el año de creación del Subcentro (1976).
- **Antecedentes:** Este campo se utiliza para poder ingresar los antecedentes médicos del paciente, un dato útil para el médico o enfermera.
- **Ocupación:** Para ingresar el dato donde indica la ocupación del paciente.
- **Tiposeguridad:** Es un campo donde indicamos si tiene algún tipo de seguro médico.
- **Redeferidode:** Aquí ingresaremos el dato en donde indicamos de donde viene el paciente, es decir si le enviaron desde otra casa de salud o algún médico particular.
- **Nombrepresentesco:** El campo en donde ingresamos el nombre de algún familiar para poder tener alguna referencia del paciente.
- **Direc Parentesco:** Dato de la dirección del familiar que ingresamos como parentesco del paciente.

- **Telf. Parentesco:** El número de teléfono del familiar.
- **Comentarios:** Un campo en donde podemos anotar un dato adicional referente al paciente.

Los siguientes campos, se refieren a datos adicionales del paciente, estos se crean en nuevas tablas, y también en otras pantallas, aquí solo se da la oportunidad de elegir las de entre las que ya se tiene creadas:

- Barrio
- Parroquia
- Cantón
- Provincia
- Zona
- Nacionalidad
- Grupo Cultural
- Parentesco
- Instrucción

Todos estos datos pueden crecer, es decir se puede crear varias clases de cada uno de estos tipos de datos, es por eso que se da una tabla a cada uno de ellos en los que puede haber un número variable de este tipo de datos, es por eso que en este formulario solo se puede hacer seleccionar de entre algunos de ellos, para el ingreso de estos datos se tiene el siguiente formulario el cual es igual para todos:



The image shows a screenshot of a web form. It consists of two rows. The first row has the label 'Nombre' on the left and a rectangular input box on the right. The second row has the label 'Comentarios' on the left and another rectangular input box on the right. The form is enclosed in a light gray border.

Gráfico 4.19.: Campos para ingreso de datos en datos extras para Historias Clínicas.

Fuente: Pantalla capturada del Sistema.

- **Nombre:** Aquí se ingresa el nombre del dato extra, estos pueden ser Barrio, parroquia, cantón, provincia, etc., esto depende la opción que se escoge al ingresar, es decir el dato extra.
- **Comentarios:** Este campo servirá para ingresar algún detalle extra que queremos ingresar sobre el dato que se está ingresando.

4.1.5. Ficha de Información de medicamentos

Para la información de medicamentos entrantes y salientes en el Subcentro, se los divide en dos tipos:

- Los medicamentos entrantes, y
- Los medicamentos salientes

Estos se los analiza más detenidamente, de la siguiente manera:

4.1.5.1. Medicamentos entrantes

Son medicamentos entrantes aquellos que ingresan cada mes al Subcentro, es decir los que provee el gobierno a las casas de salud, estos medicamentos en realidad no tienen ningún costo, en su mayoría, pues son donados al Subcentro para que a su vez sea donado al paciente.

Pero para el ingreso del medicamento hay que registrar su precio, aunque no sea cobrado, esto con fines de formalidad y respaldo para el Subcentro.

Para este tipo de registro, no existe físicamente un formulario como en las historias clínicas del paciente, pero estos son registrados en un archivo de Excel con el cual se maneja todas las entradas de los medicamentos. Los campos que se registra en la base de datos se los crea de la siguiente manera:

Fecha	2011 - 05 - 02
Saldo Anterior	<input type="text"/>
Ingreso Actual	<input type="text"/>
Valor/Fact.	<input type="text"/>
Total	<input type="text"/>
Consumo	<input type="text"/>
Precio Actual	<input type="text"/>
Precio Anterior	<input type="text"/>
Diferencia(Precios)	<input type="text"/>
Venta/Mes	<input type="text"/>
Saldo Fin.	<input type="text"/>
Comentario	<input type="text"/>
Medicamento	<input type="text"/>

Gráfico 4.20.: Campos para el registro mensual de ingreso de medicamentos.

Fuente: Pantalla capturada del Sistema.

Estos campos son los mismos que se maneja con el archivo de Excel creado en el Subcentro de Salud.

Cada campo de este formulario son:

- **Fecha:** Este campo por defecto va a asignarse la fecha actual, esto es para que el usuario no tenga que estar asignando otra fecha a la de “hoy”, pero si eso fuera necesario se puede cambiar la fecha sin ningún problema. En este campo se guardará la fecha del ingreso de medicamentos.
- **Saldo Anterior:** Este campo se refiere al saldo de medicamentos del mes anterior, esto es las unidades de medicamentos que ha sobrado.
- **Ingreso Actual:** En este campo se registra las unidades de medicamentos que ingresan en el presente mes al registro.
- **Valor/Fact.:** Este campo va a registrar el valor de la factura del medicamento que está ingresando actualmente, esto es la cantidad de dinero al que representa el ingreso del mismo.
- **Total:** Este campo sirve para registrar la cantidad total de medicamentos en unidades que nos quedará disponibles para el presente mes, esto es la suma del saldo anterior más el ingreso actual del medicamento al que se está registrando.
- **Consumo:** El consumo se refiere a la cantidad de unidades que ha consumido en el presente mes.
- **Precio Actual:** Se refiere al registro del precio con el que vinieron los medicamentos nuevos.
- **Precio Anterior:** En este campo registrará los precios al que estuvo el medicamento en el mes anterior .
- **Diferencia (precios):** Este campo es el que contendrá la diferencia de los precios, es decir *Precio Actual – Precio Anterior*.
- **Venta/Mes:** Es la cantidad de dinero del egreso del medicamento en registro.
- **Saldo Fin.:** Este campo es el que va a contener el saldo del medicamento para el presente mes.
- **Comentario:** Normalmente este campo se lo utiliza para anotar algún detalle extra acerca de este registro, en este caso del medicamento.
- **Medicamento:** Aquí se seleccionará el nombre del medicamento al que estamos registrando.

NOTA: Nótese que se está registrando de un solo medicamento, este mismo trabajo debemos hacerlo para cada medicamento que ingresa al Subcentro en cada mes, además el ingreso del medicamento no puede ser obligatoriamente cada mes, puede ser que uno de esos medicamentos no vinieron por escasez de ellos.

4.1.5.2. Medicamentos salientes

En este caso se refiere a los medicamentos que salen diariamente, es decir los medicamentos que a diario se donan a los pacientes en su mayoría.

Fecha	2011 - 05 - 02
Cantidad	<input type="text"/>
Valor Unitario	<input type="text"/>
Descripcion	<input type="text"/>
Medicamento	<input type="text"/>

Gráfico 4.21.: Campos para el registro diario de salida de medicamentos.

Fuente: Pantalla capturada del Sistema.

De igual manera que el formulario anterior, aparece por defecto la fecha actual, pero si fuera necesario se pudiera cambiar.

Los campos de este tipo de registro son:

- **Fecha:** La fecha de registro (por defecto aparece la actual).
- **Cantidad:** La cantidad de unidades que ha salido en ese día.
- **Valor Unitario:** Precio por unidad del producto.
- **Descripción:** Registro de un dato extra.
- **Medicamento:** Se selecciona el nombre del medicamento que se está registrando.

Cabe destacar que para el registro diario de medicamentos conviene realizar el registro al final del día de atención a los clientes, pues aquí ya tendremos el resumen de salida de ellos, aunque no es obligatorio.

4.1.6. Ficha Datos Personales de Médicos

Esta ficha se lo utiliza para registrar los datos personales de los médicos que pasan por el Subcentro de salud, esto con el fin de que el Subcentro mantenga un historial propio de esta clase de datos.

Pero este registro se lo puede utilizar también para registrar datos personales de enfermeras, pasantes, auxiliares que trabajan en esta casa de salud, para poder mantener un registro más completo.

Cédula	<input type="text"/>
Apellidos	<input type="text"/>
Nombres	<input type="text"/>
Teléfono	<input type="text"/>
Celular	<input type="text"/>
Sexo	<input type="button" value="v"/>
Especialidad	<input type="text"/>

Gráfico 4.22.: Campos para el registro de datos personales de médicos y/o enfermeras.

Fuente: Pantalla capturada del Sistema.

Ahora se explica cada campo de este formulario:

- **Cédula:** Para registrar el número de cédula de identidad del profesional de la salud.
- **Apellidos:** Los apellidos completos del profesional.
- **Nombres:** Los nombres completos.
- **Teléfono:** Para ingresar el número de teléfono convencional.
- **Activo:** Para registrar el dato del médico esta activo o no, es decir si aún se encuentra prestando sus servicios profesionales o no, en esta casa de salud.
- **Sexo:** Aquí se da la oportunidad de escoger entre la letra “M” (Masculino) y “F” (femenino), para determinar el sexo del profesional de la salud.
- **Especialidad:** Aquí se debe registrar la especialidad que tenga el médico/a del cual se está realizando el registro.

Como ya se ha dicho anteriormente este tipo de registro permitirá tener un historial de registros de los profesionales de la salud que han pasado por el Subcentro y que están prestando sus servicios actualmente, el mismo que nos permitirá consultar los datos personales de los mismos en el momento que se requiera.

4.1.7. Ficha Datos Básicos de Turnos

- Este tipo de registro se lo va a realizar para los turnos que se extienden a diario en el Subcentro, en donde se registra datos básicos para el turno que se da a un determinado paciente, el mismo que servirá mantener un historial de turnos extendidos en fechas determinadas.
- Este registro permitirá a su vez realizar un estudio de requerimientos de atención médica en un rango determinado de fechas, o algún tipo de estudio que se requiera sobre los turnos datos en algún mes de un determinado año.

Nº Turno	<input type="text"/>
Fecha	2011 ▾ - 05 ▾ - 03 ▾
Tipo	<input type="text"/> ▾
Historia	<input type="text"/> ▾
Médicos id	<input type="text"/> ▾
medico	

Gráfico 4.23.: Campos de Datos Básicos de Turnos.

Fuente: Pantalla capturada del Sistema.

Los campos que se incrementa en este tipo de registros son:

- **Nº de Turno:** Aquí se registra el Nº de Turno que le toca al paciente.
- **Fecha:** Por defecto está con la fecha actual, pero se puede cambiar según el requerimiento.
- **Médicos id medico:** Para este caso también se da la oportunidad de escoger de entre los médicos registrados en la base de Datos (Activos).

NOTA: Hay que tener en cuenta que el registro de estos datos va a engordar rápido a la Base de Datos, pues a diario se extienden un aproximado de 120 turnos, tomando en cuenta que actualmente atienden 8 médicos, por lo que hay que dar de baja a los más antiguos y mantenerlos en los backups que el mismo sistema puede sacarlos.

4.2. DISEÑO

4.2.1. Delimitación del Sistema

Esta parte se lo divide en los siguientes tres aspectos para un fácil entendimiento en cuanto a su delimitación, alcance y restricciones encontradas durante la ejecución del proyecto:

4.2.1.1 Delimitación

Se refiere al tiempo que requirió la ejecución y diseño de este nuevo producto y las actividades que realizó en ellas los mismos que ya lo realizamos en el **Capítulo 3** de este texto, en esta parte se va a realizar un resumen.

En cuanto al tiempo que se requirió fue 8 1/2 meses, distribuidos de la siguiente manera:

ACTIVIDAD	TIEMPO EMPLEADO
ANÁLISIS PRELIMINAR: Este análisis nos sirvió para poder detectar las urgentes necesidades a atender en el Subcentro, en donde pudimos observar el trabajo en forma manual que se realiza en el Subcentro.	1 mes
ESTUDIO DE LA HERRAMIENTA: Esta parte fue la que nos ocupó más tiempo de lo previsto, pues al no tener conocimiento sobre la herramienta, en nuestro caso el framework (Symfony) para PHP, se tuvo algunos inconvenientes para empezar el desarrollo.	3 meses
DESARROLLO DEL SOFTWARE: Una vez que ya se tuvo la idea de la aplicación de nuestra herramienta, en esta parte ya se empieza con el diseño del nuevo producto, previamente analizado la arquitectura del mismo.	3 meses
PRUEBAS Y CORRECCIONES: Para detectar posibles problemas en cuanto a la aplicación en la realidad, se realizó pruebas con datos reales migrando la base de datos existente en Excel a nuestra herramienta Mysql, en donde se detectó varias falencias acerca de nuestro sistema, las cuales corregimos la mayoría de ellas.	1 mes
INSTALACIÓN Y CAPACITACIÓN: En la instalación en el equipo de cómputo del Subcentro se encontró un inconveniente, pues la máquina en la cual estaba destinada para el Servidos de Datos estaba congelada con la herramienta Deep Freezer, por lo que se tuvo que descongelarla para poder proceder con las configuraciones necesarias, para luego proceder con la capacitación de su uso a las personas que más cerca van a estar con el sistema.	1/2 mes

Cuadro 4.2. Distribución de tiempo ocupado en la ejecución del proyecto.

4.2.1.2. Alcance o Cobertura

El sistema está orientado a las enfermeras y médicos que prestan los servicios profesionales en el Subcentro de Salud de San Antonio de Ibarra, los mismos que tendrán acceso a la Base de Datos mediante una intranet en el edificio de esta casa de salud.

En cuanto a la base de datos, en esta podremos registrar información necesaria de:

- Historias Clínicas de Pacientes.
- Datos Personales de Médicos/as y/o enfermeras/os que prestan sus servicios en esta casa de salud.
- Datos necesarios de medicamentos categorizados y el registro de las entradas y salidas de los mismos.
- Datos básicos de turnos que se dan a diario a los pacientes.

Todo esto sumado la posibilidad de restaurar o crear copias de seguridad de los datos en un respectivo tiempo.

4.2.1.3. Restricciones y/o Limitaciones

La facilidad de acceso a la información necesaria para la continuación del desarrollo de nuestro proyecto, permitió seguir avanzando sin ninguna restricción en cuanto a este aspecto.

Un inconveniente que se tuvo en el desarrollo del proyecto fue a la hora de su instalación y configuración en el equipo donde tendría que trabajar como servidor, y el temor que tenían las personas, usuarios de este equipo de infectar su máquina.

En cuanto a la limitación del sistema se puede analizarlo desde dos puntos de vista:

CON RESPECTO A SEGURIDADES:

El sistema está diseñado en su mayor parte para las personas que trabajan en esta casa de salud, ya que la información que se maneja solo las conciernen a ellos, es decir en su

mayoría son las historias clínicas de pacientes, medicamentos y datos personales de los médicos que prestan su servicio aquí.

Un paciente, o una persona particular a esta casa de salud, solo podrá ingresar a ver la presentación, aspectos del sistema, y datos básicos del Subcentro de Salud san Antonio, pues para ingresar a la administración de información de la base de datos deberá tener una clave de acceso, el mismo que solo deben tenerlos personas autorizadas a usar el sistema, con este sistema mantenemos seguridad e integridad los datos muy importantes para el trabajo diario del Subcentro.

CON RESPECTO AL SOFTWARE:

En cuanto al sistema va a tener sus limitaciones con respecto al alcance que brinda a los usuarios, pues en su mayoría se pone más énfasis en las historias clínicas de los pacientes más no en turnos, medicinas, tipos de atención que se pueden dar y datos específicos para ellos, aunque el sistema si acoge estos temas pero de una manera superficial.

En cuanto a Reportes, este es otro aspecto que está limitado, pues tenemos pocos tipos de reportes, los más necesarios actualmente para el Subcentro, pero podríamos diversificarlo más aún, por lo que el usuario vería una gran desventaja en este aspecto.

Pero el tipo de programación que brinda Symfony, permite crear software escalable, por lo que podemos aumentar módulos para cubrir con la mayoría de requerimientos, es una de las grandes ventajas que se puede encontrar en esta herramienta.

4.2.2. Diseño de Tablas de la Base de Datos

Las tablas de las bases de datos que se diseñó para nuestro sistema están orientadas exclusivamente para el Subcentro y cubrir las necesidades más urgentes de esta casa de salud.

Para esto se define el modelo de datos, para esto es necesario un ORM (Doctrine o Propel) para interactuar con la base de datos y crear el primer módulo de la aplicación.

La ventaja de utilizar Symfony es que esta herramienta se encarga de la mayor parte del trabajo, por lo que es fácil crear un módulo web completamente funcional sin tener que escribir mucho código PHP.

4.2.2.1. El Modelo Relacional

Los escenarios que se presentaron en el apartado **3.3.1.**, en donde se refiere a los Casos de Uso del Proyecto, aquí se describen los objetos principales que componen el proyecto:

- Administración de información de Historias Clínicas (tabla PACIENTES)

A su vez a esta tabla forman parte las siguientes tablas:

- Tabla PROVINCIAS
 - Tabla CANTONES
 - Tabla PARROQUIAS
 - Tabla BARRIOS
 - Tabla ZONAS
 - Tabla NACIONALIDADES
 - Tabla GRUPOS_CULTURALES
 - Tabla INSTRUCCIÓN
 - Tabla PARENTESCO
- Administración de Medicamentos (tabla MEDICAMENTO)
De la misma manera, de esta tabla también forman parte las siguientes tablas:
 - Tabla CATEGORIA_MEDICAMENTO
 - Tabla CONS_DIARIO_MEDICAM
 - Tabla REGISTRO_MEDICAMENTOS
 - Administración de Datos Personales de Médicos (tabla MEDICOS)
 - Administración de Turnos (tabla TURNOS)
 - Administración de Consultas de los pacientes (tabla CONSULTAS)

El diagrama *Entidad_Relación* se muestra en el **Gráfico 4.6.**, en donde se observa todas las tablas y las relaciones entre ellas.

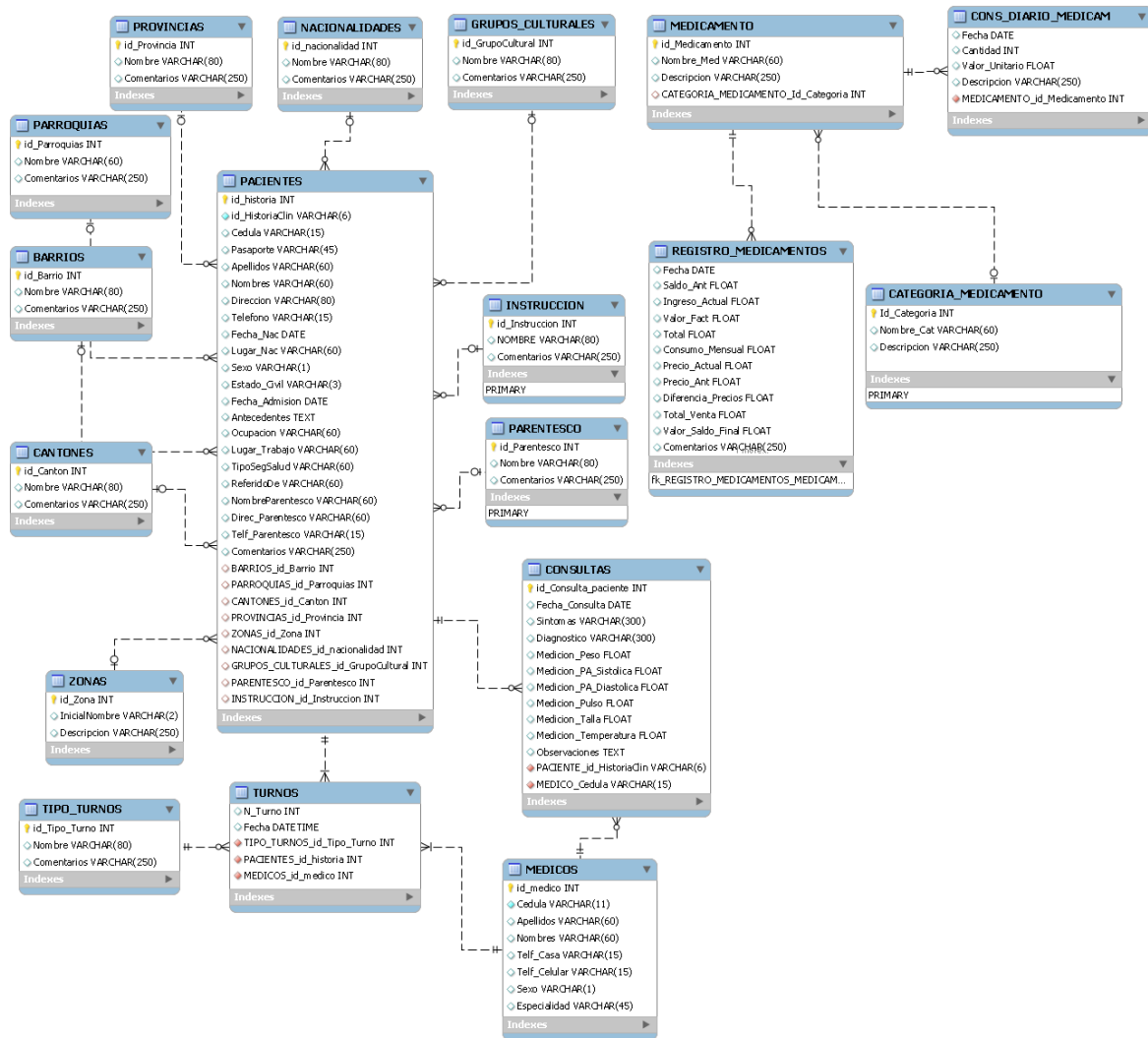


Gráfico 4.24.: Diagrama Entidad_Relación de nuestra Base de Datos.

Fuente: Diseño Propio ayudado por MySQLWorkbench 5.1.18.

4.2.2.2. El Esquema

Todos los datos de Historias Clínicas, Médicos, Medicamentos, Turnos, Consultas, se lo guardan en una base de datos relacional.

Ahora, como Symfony es un framework orientado a objetos, esta herramienta nos da la posibilidad de trabajar con objetos. Por ejemplo, preferimos utilizar objetos a tener que escribir sentencias SQL para obtener los registros de la base de datos, ya que de esta manera estaremos más apegados a la programación orientada a objetos.

Por lo que para poder trabajar con objetos en una base de datos relacional, es necesario

realizar un “mapeo”¹² o conversión entre la información de la base de datos y los objetos PHP, para esto lo podemos realizarlas con unas herramientas llamadas ORM, pues Symfony incluye por defecto dos de las más utilizadas: Propel y Doctrine, para la versión de Symfony (*Versión 1.4*) que se utilizó, viene por defecto Doctrine.

El ORM procede a partir de la descripción de cada tabla y de las relaciones entre tablas a crear clases PHP las mismas que son necesarias para trabajar con objetos.

Para realizar esta actividad se lo puede realizar de dos formas:

1. Creando la descripción del esquema de datos: mediante una base de datos existente, o
2. Creando el esquema manualmente, para esto hay que saber un poco del lenguaje YAML, necesario para crear tablas y relaciones entre ellas.

Pero también hay la posibilidad de simplificar aún más el trabajo de crear un esquema de datos, pues existen aplicaciones para crear bases de datos gráficamente como es el caso de MySQL Workbench en donde se realiza el diagrama entidad relación en forma gráfica, y para generar archivos tipo shema.xml, además que es una herramienta libre.

Al crear un esquema de este estilo, con formato “YAML”¹³ se puede hacer funcionar con la mayoría de gestores de bases de datos.

En el presente caso se creó el archivo *shema.yml* de la siguiente manera:

1. Utilizando MySQL Workbench para realizar el modelo de datos en forma visual cuyo diagrama se muestra en el **Gráfico 4.23**.
2. De la misma herramienta se crea la Base de Datos con la opción */Database/Forward Engineer...*, la cual crea la Base de Datos y de esta manera ya lo vamos a encontrar con PHPMyAdmin.
3. Una vez creado la Base de Datos, primeramente se configura un archivo que se encuentra en la carpeta de nuestro proyecto en la dirección *sistema/config/databases.yml*, el cual se edita para cambiar algunos aspectos,

¹² Técnica que crea una base de datos orientada a objetos virtual, sobre la base de datos relacional.

¹³ Formato para serializar datos, es fácil de leer y es compatible con todos los lenguajes de programación

como es el nombre de la Base de Datos, el tipo de base de datos, el username y el password, como muestra el siguiente código:

all:

doctrine:

class: sfDoctrineDatabase

param:

dsn: 'mysql:host=localhost;dbname=bdsbcentro'

username: root

password: null

4. Para luego crear el esquema de la Base de Datos con el comando `./symfony doctrine:build-schema`, la misma que se puede realizar en forma gráfica, ya que utilizamos Netbeans como se explicó anteriormente en la ejecución de comandos Symfony.
5. De la misma manera se crea el modelo de datos, con el comando `./symfony doctrine:build-model`, este comando lo que hace es crear el modelo de clases basado en el esquema creado, esto hace que los objetos del modelo relacional se pueda manejarlos con clases.

Después de realizar los pasos anteriores se tiene el siguiente código en nuestro archivo `shema.yml`:

Barrios:

connection: doctrine

tableName: barrios

columns:

id_barrio:

type: integer(4)

fixed: false

unsigned: false

primary: true

autoincrement: true

nombre:

type: string(80)

fixed: false

unsigned: false

primary: false

notnull: false

autoincrement: false

comentarios:

type: string(250)

fixed: false

```
unsigned: false
primary: false
nonnull: false
autoincrement: false
relations:
  Pacientes:
    local: id_barrio
    foreign: barrios_id_barrio
    type: many
```

La estructura del esquema (*shema.yml*) de datos se refiere a la traducción del diagrama entidad-relación al formato YAML.

Como ya se dijo anteriormente YAML es un lenguaje muy sencillo que permite describir datos: cadenas de texto, número enteros, fechas, *arrays* simples y *arrays* asociativos.

La “TABULACIÓN”¹⁴ es muy utilizada por YAML para organizar su estructura, los elementos que forman una secuencia utilizan un guión medio y los pares clave/valor de los *arrays* asociativos se separan con dos puntos. YAML también dispone de una notación abreviada para describir la misma estructura con menos líneas: los *arrays* simples se definen con [...] y los *arrays* asociativos se definen con {...}.

Symfony utiliza YAML en la mayoría de sus archivos de configuración, lo más importante que se debe saber al modificar un archivo YAML es que la tabulación siempre se realiza con ESPACIOS EN BLANCO y nunca con el tabulador.

Todas las tablas y columnas de la base de datos se describe en el archivo *schema.yml*, aquí se encuentra la información descrito con la siguiente información:

- **Nombre de la Tabla**
 - *connection*: Tipo de conexión (ORM), en este caso Doctrine.
 - *tableName*: El nombre de la tabla.
 - *columns*: el nombre de cada columna.
 - *type*: el tipo de columna, que puede ser boolean, tinyint, smallint, integer, bigint, double, float, real, decimal, char, varchar(size), longvarchar, date, time, timestamp, blob y clob.

¹⁴ Espaciado desde el margen izquierdo hacia la derecha.

- `fixed`: en el caso de que demos un valor inicial, `False` en caso contrario.
- `unsigned`: campo con signo o sin él.
- `primary`: si el campo es una clave primaria `True` en caso contrario `false`.
- `notnull`: Si es que queremos que el campo no acepte campos vacíos `True`, caso contrario `False`.
- `autoincrement`: esto es para los valores numéricos, en el caso que se autoincrementen.
- *relations*: en esta sección se define las relaciones con esta tabla, las cuales todas van a tener los siguientes campos:
 - `local`: el nombre del campo de esta tabla con el que se relaciona la clave foránea
 - `foreign`: el nombre de la clave foránea.
 - `type`: en este caso se define el tipo de relación (`many`, `one`).

En el caso de que se requiera trabajar o cambiarse a otro gestor de bases de datos que no sea MySQL, el cambio no afecta al código anterior, claro que este gestor debe soportar este mismo tipo de código. Pues la idea de trabajar con un ORM es que este debe generar automáticamente las sentencias SQL, es muy sencillo adaptar el código a otro tipo de base de datos.

4.2.2.3. EI ORM

La descripción de las tablas y columnas de la base de datos en el archivo *schema.yml*, permite hacer uso de algunas tareas (comandos) incluidas en Doctrine, estas generan automáticamente las sentencias SQL las mismas que son necesarias para crear todas las tablas de la base de datos:

```
./symfony doctrine:build-sql
```

Esta tarea genera en el directorio `data/sql/` las sentencias SQL optimizadas para el sistema gestor de bases de datos que estamos utilizando:

```
# fragmento del archivo data/sql/ schema.sql
```

```
CREATE TABLE barrios (id_barrio INT AUTO_INCREMENT, nombre VARCHAR(80),  
comentarios VARCHAR(250), PRIMARY KEY(id_barrio)) ENGINE = INNODB;
```

El código anterior es similar para cada tabla del sistema que se está desarrollando. Ahora para crear la estructura de tablas en la base de datos, se ejecuta la tarea *doctrine:insert-sql*:

```
./symfony doctrine:insert-sql
```

Para ver la ayuda sobre un comando o tarea en particular, se puede utilizar la tarea *help*, la misma que muestra la explicación en inglés completa de su uso:

```
./symfony help propel:insert-sql
```

Estas explicaciones muestran todos los argumentos y opciones de la tarea (comando), los valores iniciales de cada uno de ellos y también algunos ejemplos de uso.

El mismo ORM también genera automáticamente las clases PHP que relacionan las tablas de la base de datos con los objetos de la aplicación, el comando (tarea) que se encarga de esta situación es la siguiente:

```
./symfony doctrine:build-model
```

Esta tarea genera el directorio *lib/model/doctrine* en donde encontramos los archivos PHP que se utilizan para interactuar con la base de datos.

Si se realiza una revisión a esta dirección vamos a ver que por cada tabla se crea 3 clases, dos de las cuales encontramos en la dirección *lib/model/doctrine*, y uno de ellos en la dirección *lib/model/doctrine/base*. Para explicar mejor esta situación tomemos el caso de la tabla Barrios del sistema, las clases que se crean en la primera dirección es:

- *Barrios.class.php*: Al crear un objeto de esta clase se representa a un único registro de la tabla Barrios, por defecto esta clase está vacía, se puede crear funciones según la conveniencia, para el presente sistema, se aprovecha esta clase para poder ingresar el método mágico `__toString()`, el mismo que permite convertir en cadena de texto el índice de la clave foránea en las tablas relacionadas, así:

```
public function __toString()
{
    return $this->getNombre();
}
```

Con este método lo que se hace es enviar el campo nombre de la tabla Barrios, cuando se haga referencia a la clave primaria de esta, en la tabla con la que se relaciona la misma, en este caso la tabla Pacientes.

- BarriosTable.class.php: Con esta clase se retorna colecciones de objetos, en este caso lo utiliza para retornar registros de la tabla, de la siguiente manera:

```
public static function getInstance()
{
    return Doctrine_Core::getTable('Barrios');
}
```

Ahora, la única clase que se crea en la dirección *lib/model/doctrine/base* es la siguiente:

- BaseBarrios.class.php: A esta se la conoce como la clase padre de, en este caso de Barrios, esta clase se sobrescribe cada vez que se ejecuta el comando (tarea) *doctrine:build --model*, por lo que se debe estar muy seguros a la hora de volverla a reconstruir, por lo que todas las personalizaciones se deben hacer en la clase Barrios (en este caso).

Doctrine, al igual que Propel, tiene una tarea útil en el caso de que no quiera hacer una a una las tareas antes analizadas, como es el caso de crear el esquema, el modelo, sql, crear formas, etc., todo esto lo puede hacer con una sola tarea (comando):

```
./symfony doctrine:build --all
```

Pero hay que tener en cuenta una situación antes de realizar esta acción, este aspecto es que si usted realizó configuraciones en la clase padre, como en el presente caso de la tabla Barrios, *BaseBarrios.class.php*, esta se sobrescribe, por lo que todas las configuraciones necesarias hay que hacerlas en la clase *Barrios.class.php*, esta no se afectará si aplicamos la tarea que se está analizando.

Esta tarea resulta muy útil en el caso de que algún archivo de configuraciones de estas carpetas se haya dañado por algún motivo y no se encuentra el daño, pero es aconsejable sacar una copia antes de realizar esta actividad, porque en caso contrario todo el trabajo hecho hasta ese momento se perderán.

Es aconsejable también que cada vez que se cree los modelos, el esquema o si aplicamos la tarea `./symfony doctrine:build --all`, es decir, cada vez que se añade una clase nueva al proyecto es necesario borrar la cache que utiliza Symfony para esto se ejecuta la tarea:

```
./symfony cache:clear --app=frontend
```

4.3. ARQUITECTURA Y CÓDIFICACIÓN DEL SISTEMA

Ya se ha analizado cómo Symfony simplifica la gestión de bases de datos por abstracción (capa de abstracción) entre los diferentes motores de bases de datos, y cómo también se convierte los elementos relacionales a clases orientadas a objetos.

Ahora es necesario ordenar nuestro código, de manera que los futuros cambios que se puedan realizar sean los más fáciles posibles, por lo que se analizará su arquitectura.

4.3.1. Análisis de Arquitectura del sistema

Al desarrollar sitios web sin ningún framework se puede utilizar un archivo PHP por página HTML, los mismos que probablemente contengan el mismo tipo de estructura, es decir: *inicialización y configuración global, lógica de negocio relacionada con la página solicitada, búsqueda de registros en la base, y finalmente, el código HTML.*

Para separar la lógica del HTML, podríamos utilizar un motor de plantillas o tal vez utilizaríamos una capa de abstracción de la base de datos para separar el modelo de la lógica de negocio.

Bueno, cualquier alternativa que se tome, vamos a terminar un código de gran tamaño que resultaría una pesadilla para cualquier programador volver a analizarlos y dar

mantenimiento, tal vez, si el sistema no es grande resultaría algo más fácil, pero en la práctica un sistema normalmente se vuelve grande, por lo que su mantenimiento resultaría imposible, a menos que ese mantenimiento lo haga el propio diseñador, es decir la persona que lo creó y trabajó en la su codificación.

Hoy en día, gracias a la creación de los famosos frameworks, este trabajo se simplifica, por lo que no necesariamente hay que ser los creadores de ese sitio web para poder dar mantenimiento, todo esto gracias al patrón de diseño MVC (Arquitectura MVC). Esta forma de diseñar sistemas define una manera de organizar su código de acuerdo a su naturaleza, por lo que separa el código en tres capas:

- LA CAPA MODELO: en esta capa se define la lógica de negocio por lo que la base de datos pertenece a esta capa, es así que Symfony guarda todas las clases y archivos relacionados con el modelo en el directorio *lib/model/*.
- LA CAPA VISTA: esta capa en cambio, es la encargada de interactuar con el usuario, por lo que un motor de plantillas es parte de esta capa. En Symfony, la vista es principalmente la capa de plantillas PHP, las mismas que son guardadas en varios directorios *templates/*. Estas carpetas son creadas cada vez que creamos un módulo o alguna aplicación para nuestro proyecto con Symfony, si nosotros abrimos esta carpeta del nuevo módulo o aplicación siempre se verá otra carpeta en su interior con el nombre *templates*, la misma que puede tener archivos php en su interior, como *index.php* o *layout.php*, en los que se puede crear varios más dependiendo de nuestras necesidades.
- EL CONTROLADOR: Es el código que llama al Modelo para obtener algunos datos, los mismos que son pasados a la Vista para la presentación al cliente. En realidad todas las peticiones se canalizan a través de los controladores frontales, este puede ser *index.php* y *frontend_dev.php*, estos controladores frontales delegan todo el trabajo en las acciones, las mismas que se encuentran en la carpeta *actions/* del **módulo** creado, por lo que se denominan módulos a las agrupaciones lógicas de acciones.

Este modelo se explica en forma grafica en el Esquema de la Arquitectura MVC del **Gráfico 2.8**.

4.3.2. El Layout.

Al analizar los esquemas de presentación de los Casos de Uso desde el **Gráfico 4.6** al **Gráfico 4.16** se va a ver que algunas partes visuales del mismo se repiten en la todas las páginas, toda la vida nos enseñaron nuestros profesores en las aulas que tener código duplicado nunca fue buena idea, ya sea código PHP o etiquetas HTML, por lo que tenemos que encontrar alguna forma de evitar esta situación, especialmente para los elementos comunes de las páginas.

Se resuelve este problema temporalmente definiendo una cabecera y un pie que se añaden en cada plantilla, aunque no es la recomendable:

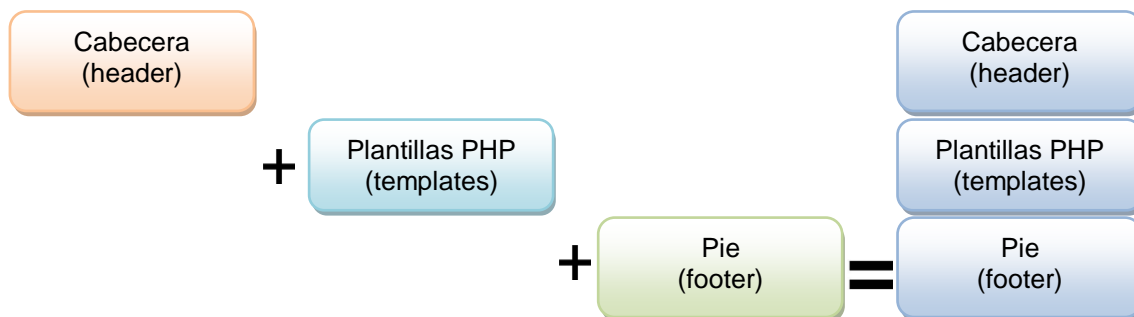


Gráfico 4.25.: Cabecera y Pie de Página, de las páginas.

Fuente: Diseño Propio.

Pero en los archivos de la cabecera y del pie no contienen código HTML válido, es por eso que se debe buscar otra alternativa, por lo que el PATRÓN DE DISEÑO DECORATOR, es el apropiado para evitar este problema, con este patrón el contenido se muestra con una plantilla que después se decora con una plantilla global que en Symfony se llama **layout**.



Gráfico 4.26.: Funcionamiento del layout.

Fuente: Diseño Propio.

En la presente aplicación se puede encontrar al *layout* en el directorio *apps/frontend/templates/* o puede ser también *apps/backend/templates/*, pues por defecto se graba en este directorio el archivo *layout.php*, es aquí donde se graban todas las plantillas globales de la aplicación.

Ayudados con Netbeans se puede visualizarlos en forma gráfica, tanto en la aplicación backend como en la frontend:

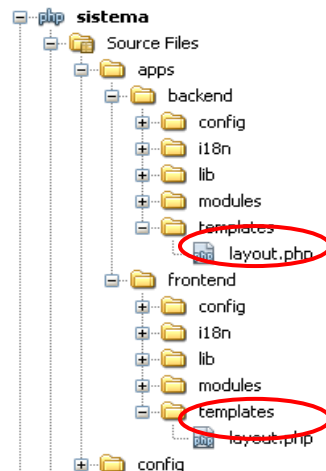


Gráfico 4.27.: Localización de los archivos *layout* en Netbeans.

Fuente: Pantalla capturada.

Si es necesario personalizarlos, se debe reemplazar el contenido del *layout* por defecto de Symfony por un código que se crea conveniente a la aplicación, así:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <?php include_http metas() ?>
    <?php include metas() ?>
    <?php include_title() ?>
    <link rel="shortcut icon" href="/favicon.ico" />
    <?php include_stylesheets() ?>
    <?php include_javascripts() ?>
  </head>
  <body>
    <div id="wrap">
    <div id="header"> </div>

    <p>
      <?php echo $sf_content ?>
    </p>
```

```
<div id="footer">
Año 2010-2011 <a href="#">utn.edu.ec</a> || desarrollado por: <a href="#">Jorge A.
Ipiales</a>
</div>
<div class="clear"></div>
</div>
</body>
</html>
```

Este código se conforma de llamadas a funciones PHP y referencias a variables PHP, hojas de estilo, imágenes para decorar el layout.

Hay que tomar muy en cuenta la variable `$sf_content`, ya que la crea el propio framework y contiene el código HTML generado por la acción.

Luego de haber puesto el código suficiente para decorar el layout, se verá en el navegador que la presentación del sitio web ya se encuentra con su nuevo aspecto, en el presente caso se accede a la dirección <http://sistema/>, esta es la presentación del frontend,

4.3.3. Las hojas de estilo, imágenes y archivos JavaScript

Al inicio del desarrollo del nuevo sitio web, lo primero que se hace es crear el proyecto, con la tarea `Symfony generate:Project`, la misma que crea por defecto tres directorios para guardar los archivos relacionados con la web:

- `web/images/` para las imágenes,
- `web/css/` para las hojas de estilos y
- `web/js/` para los archivos de JavaScript.

Se puede mostrar gráficamente con la ayuda de netbeans el orden de carpetas que crea por defecto esta tarea, pero que además se puede crear varias otras carpetas según la necesidad:

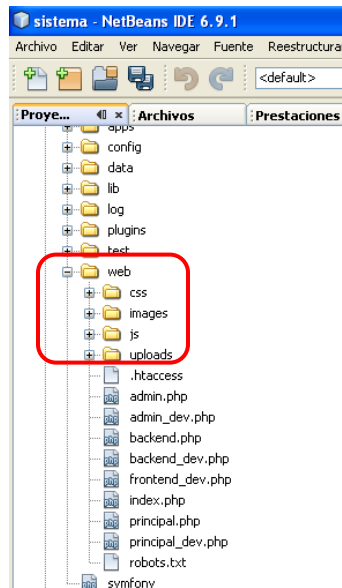


Gráfico 4.28.: Localización de Directorio Web en Netbeans.

Fuente: Pantalla capturada.

De esta forma por defecto Symfony ordena los directorios, pero esto no es una ley que se debe seguir al pie de la letra, porque si se tiene otro criterio de ordenación se puede guardar estos archivos en cualquier otro directorio dentro del directorio *web/*.

Al revisar el código de todos los módulos de la aplicación, no se va a ver un archivo de hojas de estilo en el layout, pero está presente en todas las páginas.

Esto se logra realizando una llamada a la función *include_stylesheets()* que se realiza dentro de la sección *<head>* del *layout*. La función *include_stylesheets()* se conoce con el nombre de **helper**.

Un helper es una función de Symfony a la que se le pueden pasar parámetros y que devuelve código HTML, estas funciones se utilizan casi siempre para mejorar la productividad en el desarrollo, ya que suelen generar fragmentos de código que se utilizan habitualmente en las plantillas. El helper *include_stylesheets()* genera las etiquetas *<link>* necesarias para enlazar las hojas de estilo.

para que el helper sepa cuál es el archivo CSS específico a incluir, en la capa de la vista se puede configurar mediante un archivo de configuración llamado *view.yml* que se

encuentra en el directorio de la aplicación y es generado por la tarea `generate:app`, aquí se encuentra un archivo que va a contener un archivo similar a este:

```
# You can find more information about this file on the symfony website:  
# http://www.symfony-project.org/reference/1_4/en/13-View
```

```
default:  
  http_metas:  
    content-type: text/html  
  metas:  
    #title:    symfony project  
    #description: symfony project  
    #keywords: symfony, project  
    #language: en  
    #robots:   index, follow  
  
  stylesheets: [main.css, estilo_backend.css]  
  javascripts: [report.js]  
  has_layout: true  
  layout:     layout
```

Este archivo (`view.yml`) es empleado para realizar las configuraciones necesarias que utilizará todas las plantillas de la aplicación.

En el código anterior se puede ver (resaltado con negrillas) que la opción `stylesheets` por ejemplo define un array que contiene el nombre de los archivos CSS de las hojas de estilo que se incluyen en cada página de la aplicación (esta información es la que utiliza el helper `include_stylesheets()` para incluir los archivos CSS en las páginas).

Symfony añade automáticamente el prefijo `/css` a las rutas relativas, es por eso que en el archivo `view.yml` anteriormente mostrado ponemos la referencia al archivos `main.css` y no `/css/main.css`, como lógicamente se puede intuir, aunque las dos formas son equivalentes.

Cuando se indican varios archivos css, como es en nuestro caso, Symfony los incluye en el orden en el que se han indicado, así:

```
stylesheets: [main.css, estilo_backend.css]
```

El archivo de configuración `view.yml` también establece el `layout` por defecto que utilizan las páginas de la aplicación su nombre al inicio es `layout`, por lo que Symfony decora todas las páginas con el archivo `layout.php`, pero hay veces que se necesita deshabilitar

esta decoración de las páginas, para eso solo se cambia al valor `false` en la opción `has_layout`.

Pero también, puede ser el caso de que se quiera que cada módulo de la aplicación tenga diferentes estilos o apariencias, por lo que se puede definir su propio archivo de configuración `view.yml`, para esto se modifica la opción `stylesheets` del archivo `view.yml` que anteriormente se muestra y hacer que solo incluya el archivo `main.css`:

apps/backend/config/view.yml

```
stylesheets: [main.css]
```

Ahora, para aplicar un archivo de estilos a un módulo respectivo, se va a coger el ejemplo en el presente proyecto al módulo Barrios, y a este módulo se va a aplicar el archivo CSS `estilo_backend.css` para que sea diferente (aunque en la práctica no se lo hace así). Para lograr esto, se crea un archivo `view.yml` en el directorio `apps/backend/modules/Barrios/config` y se añade el siguiente código:

apps/frontend/modules/Barrios/config/view.yml

```
indexSuccess:
```

```
  stylesheets: [estilo_backend.css]
```

```
showSuccess:
```

```
  stylesheets: [estilo_backend.css]
```

`IndexSuccess` y `showSuccess` en este caso se refiere a nombres de las plantillas asociadas con las acciones `index` y `show`, el archivo `view.yml` en este caso el módulo Barrios utiliza estos nombres para crear las secciones que modifican el aspecto de cada acción.

En cambio, en el caso de que no se define el valor de alguna opción en el archivo `view.yml` del módulo, Symfony lo toma directamente del archivo `view.yml` de la aplicación, es decir de toda la aplicación que contiene al módulo.

Ahora, si se quiere establecer una misma opción para todas las acciones del módulo, esto se lo hace bajo una sección especial llamada `all`.

Las configuraciones se los realizan en archivos apropiados para esta situación, estas configuraciones se las puede realizar en niveles claramente identificados:

- **La Configuración por defecto**, las configuraciones propias de los archivos Symfony.
- **La configuración global del proyecto**, la misma que se encuentra en la carpeta *config*, al entrar a la carpeta del proyecto, en este caso está en *sistema/config/*
- **La configuración local de la aplicación**, en el presente caso lo encontramos en *sistema/apps/[nombre_de_aplicacion]/config/*, con *nombre_de_aplicación* nos referimos a backend o frontend, las aplicaciones que tiene el proyecto.
- **La configuración local del módulo**, que se encuentra en *sistema/apps/[nombre_de_aplicacion]/modules/[nombre_de_modulo]/config/*, de igual manera *nombre_de_modulo*, se refiere al módulo al que se va a configurar.

Para configurar los archivos JavaScript, en el mismo archivo de configuración *view.yml* se utiliza la opción *javascripts* o también se lo puede realizar mediante llamadas al helper **use_javascript()** desde una plantilla.

4.3.4. Las Acciones

Cada vez que se crea un módulo, también se va a crear unos subdirectorios para las plantillas y las acciones. Una acción es la parte del CONTROLADOR de las páginas y las plantillas asociadas, a estas acciones normalmente se los nombran con la siguiente estructura: *[nombre]Success.php*, como por ejemplo *indexSuccess.php* que es la parte de la VISTA, visualmente puede ver de la siguiente manera:

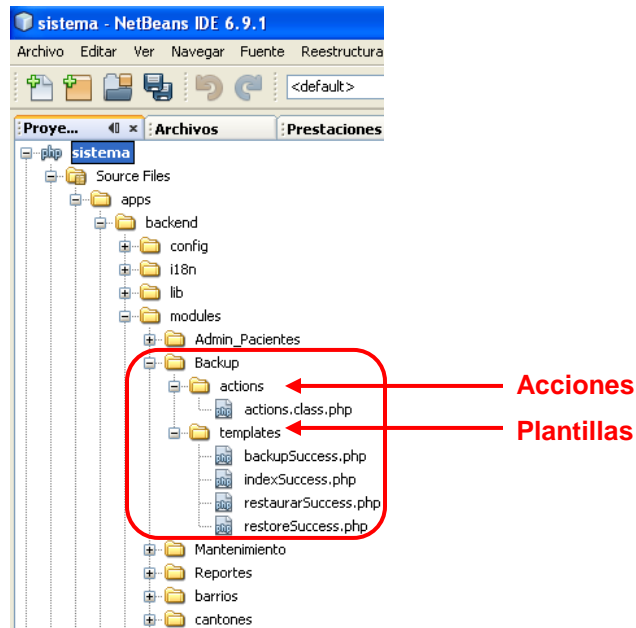


Gráfico 4.29.: Localización de Acciones y Plantillas de Módulos.

Fuente: Pantalla capturada.

Por lo que los niveles de los directorios se los puede también diferenciarlos de la siguiente manera:

```
apps/
  frontend/
  modules/
    job/
      actions/
        actions.class.php
      templates/
        indexSuccess.php
```

De esta manera ya se puede localizar las acciones, a las cuales también se definen como métodos de una clase.

Los métodos para las acciones van a estar siempre en el archivo `actions.class.php` y la estructura de sus nombres van a ser similar a: `execute[nombre_de_metodo]`, por lo que para el método `Index` se los nombra como `executeIndex()` (siempre es la palabra `execute` seguida del nombre de la acción). Lo que hace esta acción es visualizar la plantilla `indexSuccess`.

Dicho de otra manera, el método `executeIndex()` (el CONTROLADOR) realiza llamadas a los métodos de la clase (como por ejemplo `barrios`) del modelo. Normalmente se lo realiza para obtener un array de objetos de tipo `Barrios` en nuestro caso.

Todas las propiedades de este objeto se pasan automáticamente a la plantilla, que es la parte de la VISTA. Para pasar datos del controlador a la vista se crea una propiedad en el objeto mediante `$this->nombreDeLaPropiedad`.

4.3.5. Las Plantillas

Como ya mencionamos anteriormente a breves rasgos, Symfony utiliza una forma de asociar el nombre de la plantilla con su respectiva acción, esta consiste en:

`[Nombre de acción]+Success`

La plantilla llamada `indexSuccess.php`, esta es la que genera todo el código HTML que permite visualizar los elementos que se manipula en la correspondiente acción, como ejemplo se muestra el código de la plantilla `index` del módulo `barrios`:

```
<!-- apps/frontend/modules/barrios/templates/indexSuccess.php -->
<?php use_helper('I18N', 'Date') ?>
<?php include_partial('barrios/assets') ?>

<div id="sf_admin_container">
  <h1><?php echo __('LISTA DE BARRIOS', array(), 'messages') ?></h1>

  <?php include_partial('barrios/flashs') ?>

  <div id="sf_admin_header">
    <?php include_partial('barrios/list_header', array('pager' => $pager)) ?>
  </div>

  <div id="sf_admin_bar">
    <h1><?php echo 'BUSQUEDA DE BARRIOS' ?></h1>
    <?php include_partial('barrios/filters', array('form' => $filters, 'configuration' =>
$configuration)) ?>
  </div>

  <div id="sf_admin_content">
    <form action="<?php echo url_for('barrios_collection', array('action' => 'batch')) ?>"
method="post">
      <?php include_partial('barrios/list', array('pager' => $pager, 'sort' => $sort, 'helper' =>
$helper)) ?>
      <ul class="sf_admin_actions">
```



```
<?php include_partial('barrios/list_batch_actions', array('helper' => $helper)) ?>
<?php include_partial('barrios/list_actions', array('helper' => $helper)) ?>
</ul>
</form>
</div>

<div id="sf_admin_footer">
  <?php include_partial('barrios/list_footer', array('pager' => $pager)) ?>
</div>
</div>
```

Hay que notar que el código que se visualiza es generado automáticamente por la tarea `/symfony doctrine:generate-admin backend barrios` el mismo que genera un módulo de administración (CRUD). Esta se encarga de generar una interfaz completa de administración para una clase del modelo:

Los módulos de administración tienen muchas más funcionalidades que los módulos simples que también se puede generarlos con la tarea `./symfony generate:module frontend article`.

Esta tarea trae muchas ventajas, pues sin ni siquiera tener que escribir una sola línea de código PHP, cada módulo incluye las siguientes características:

- Muestra el listado de registros (objetos) con paginación
- El listado que se muestra se puede ordenar
- Se puede realizar búsquedas de un determinado objeto o dato de la tabla (Filtrar datos)
- Se puede crear un nuevo registro, modificarlos y borrarlos.
- Además introduce la posibilidad de borrar varios registros (objetos) a la vez
- Se aplica la validación en los formularios
- Se muestran mensajes flash para informar al usuario del resultado de las acciones
- Entre otras...

Este generador de la parte administrativa incluye todas las características necesarias para crear una interfaz de administración en forma de módulos generados fácilmente y además se tiene la ventaja de que son configurables.

Muchas veces, para acceder al valor de cada columna también se puede utilizar un método generado automáticamente y que se construye uniendo la palabra `get` junto con el nombre de la columna en formato “*camelCase*”¹⁵, este consiste en eliminar los guiones bajos del nombre original de la columna y escribir en mayúsculas la primera letra de cada palabra, por lo que por ejemplo una columna llamada `created_at` tiene un método asociado llamado `getCreatedAt()`.

4.3.6. Los Plugins de Symfony

Un plugin ofrece una manera de empaquetar y distribuir una parte de archivos del proyecto. Al igual que un proyecto, un plugin puede tener clases, helpers, configuraciones, tareas, módulos, esquemas, e incluso recursos Web (CSS, JavaScript, etc.).**[WWW.028]**.

Son herramientas que colaboradores de la comunidad Symfony los publican en la página oficial de Symfony (<http://www.symfony-project.org/plugins/>), estas ayudan en gran medida a la construcción del nuevo proyecto.

Los plugins son bastante útiles en el caso de que queramos utilizar un mismo código en diferentes proyectos, pues el objetivo de los plugins es facilitar el intercambio de código entre aplicaciones, o incluso entre distintos proyectos, ya que las aplicaciones symfony sólo comparten el modelo, los Plugins dan una manera de compartir más componentes entre aplicaciones.

Podemos distinguir dos clases de Plugins:

Privados:

Cuando se necesita aplicar en los proyectos un mismo esquema, o los mismos módulos, no es necesario volver a codificarlos, a estos se los puede colocar en un plugin. Pues este en realidad es solo un directorio que tranquilamente se los puede mover al proyecto que se quiere aplicar.

¹⁵ El nombre se debe a que las mayúsculas a lo largo de una palabra en CamelCase se asemejan a las jorobas de un camello.

Además a estos se los llama privados ya que no está a disposición del público, pues está restringido solo para el uso personal o de la empresa.

Públicos:

A estos plugins los llamamos públicos por que son los que se encuentran en Internet y se puede descargarlos e instalarlos para usarlos en nuestras aplicaciones, en su estructura no se diferencia en nada con los privados.

Todos los Plugins públicos que se encuentra en la página oficial de Symfony, son públicos, estos además deben tener un archivo de texto que nos indica cómo usarlos y configurarlos para utilizarlos en un determinado proyecto, en caso contrario no se podría utilizarlos como se debe.

Además para poder mantener una organización más entendible todos los plugins deben terminar con *Plugin*, también es una buena costumbre usar el prefijo *sf*, aunque no es obligatorio.

4.3.6. Arquitectura por defecto vs. Arquitectura de los Plugins

En cuanto a la forma de organizar los directorios es decir a su arquitectura, los Plugins difiere un poco a la utilizada Symfony por defecto:

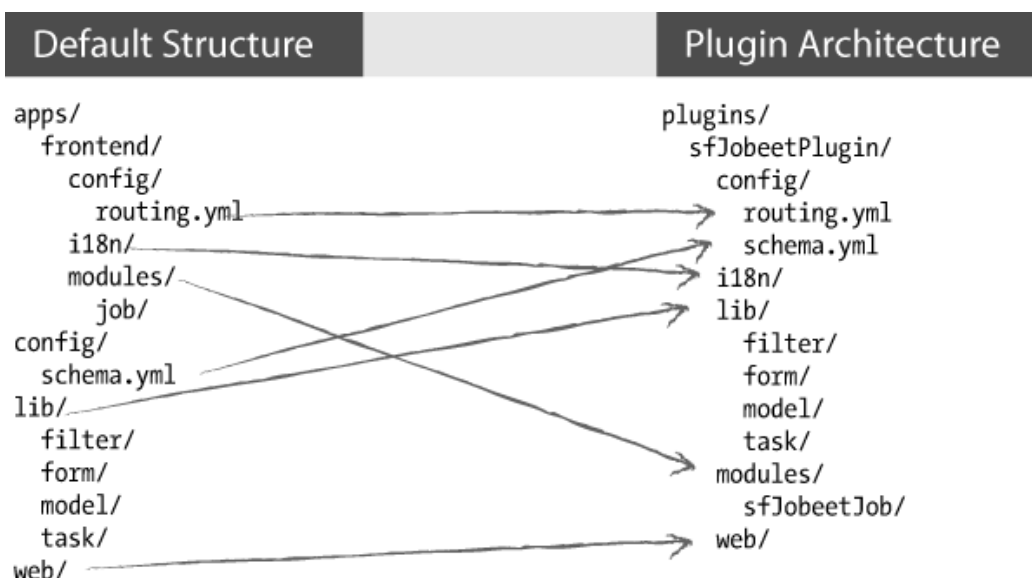


Gráfico 4.30.: Comparación de arquitecturas Symfony por defecto y Plugins.

Fuente: [WWW.029].

Cuando se intenta resolver un problema común o simplemente aumentar una funcionalidad para la aplicación, es necesario adicionar en la sección plugin del sitio web Symfony el nuevo plugin, con lo que se ahorraría bastante tiempo y trabajo, esto resulta de gran utilidad.

Ahora, cuando ya se tiene un plugin, sea este privado o público, en el presente proyecto se lo ha realizado con plugins públicos, hay 3 maneras de instalarlos:

- **Usando la tarea plugin:install** (esto solo funciona si el desarrollador ha creado un plugin package y lo ha subido al sitio web de Symfony)
- **Descargando el package/paquete** y manualmente descomprimirlo bajo el directorio plugins/ (también es necesario que el desarrollador haya subido un package)
- **La creación de un svn:externals** en plugins/ para el plugin (esto solo funciona si el desarrollador ha alojado su plugin en Subversion)

4.4. IMPLEMENTACIÓN

4.4.1. Análisis del Producto Terminado

El sitio construido para el Subcentro de Salud San Antonio, está diseñado exclusivamente para mantener la base de datos de la información de pacientes y otros datos importantes para este lugar en una forma ordenada y fácil para su administración.

Al contar con este sitio se va a encontrar ventajas y tal vez desventajas tomando en cuenta los puntos de vista de los pacientes y usuarios en este caso van a ser los profesionales de la salud.

4.4.1.1. Ventajas y desventajas desde el punto de vista del Paciente

Aquí se va a encontrar varias ventajas con el uso de este producto:

- El paciente ya no va a tener que esperar mucho tiempo en la cola en espera de un turno, pues los turnos se despacharán con mayor rapidez.
- El paciente ya no tendrá que madrugar para poder ver si obtiene un turno con algún médico en especial, esto lo decimos porque para odontología

especialmente, las personas han hecho cola durante mucho tiempo y resulta que ya se acabaron.

- El paciente ya no perderá mucho tiempo a que le atiendan, ya que todo el proceso hasta su atención se acortará.
- Para modificar sus datos o para crearse una nueva historia se lo realizará con mayor rapidez, pues ya no tendrá que volver otro día o esperar al último para que lo puedan atender.
- Para realizar una consulta sencilla, o una vacuna, no tendrá que esperar hasta que las demás enfermeras se desocupen y le atiendan, pues al agilizar el trabajo de dar turnos ya les quedará tiempo para atender pequeñas solicitudes como estas.
- El paciente a la hora que va a retirar sus medicamentos, ya no tendrá que esperar mucho tiempo para que le den averiguando si existe o no tal medicamento, por lo que tampoco se generará una cola larga en farmacia.

Se podría decir que no existe ninguna desventaja al incorporar el sistema como una herramienta más del Subcentro de Salud, ya que para ellos las desventajas si es que lo hubiera no van a ser perceptibles y no los van a tomar en cuenta.

Por todo lo dicho desde el punto de vista de los pacientes, solo las ventajas van a ser muy notables y beneficiosas.

4.4.1.2. Ventajas y desventajas desde el punto de vista del Usuario

Desde el punto de vista del usuario, que en este caso en su mayoría van a ser las enfermeras, los médicos y los pasantes de enfermería que realizan sus prácticas en el Subcentro, vamos a notar también ventajas y vamos a ver si es que existen desventajas desde aquí:

Las ventajas que van a encontrar los usuarios con la incorporación de esta nueva herramienta en su rutina diaria, son:

- Se podrá despachar turnos en forma más rápida, pues las historias clínicas se encontrarán más rápido.
- Las colas generadas se terminarán rápidamente.

- Las consultas de pacientes ya no se harán solo con el número de historia, si no que ya tendrá otras alternativas como es: por número de cédula de identidad, nombres, apellidos, dirección.
- Los médicos podrán consultar la existencia de medicinas en farmacia en forma rápida, sin tener que salir de su consultorio para realizar esta actividad.
- Las enfermeras podrán mantener datos personales de médicos y personal que trabaja en esta casa de salud en un solo lugar.
- Podrá modificar y crear historias más rápidamente.
- El tiempo de trabajo ya no será limitado, es decir que ya les sobraré tiempo para atender otras actividades necesarias.
- Los informes también se podrán realizar de una forma más rápida desde el sistema.
- La recepción y salida de medicamentos se podrá realizar en forma más rápida ordenada y manteniendo guardado estos datos en un solo lugar (en este caso la base de datos),
- Todos los datos que se manipula en el Subcentro estarán almacenados en un solo lugar, y ya no tendrá que perder el tiempo buscando en los anaqueles o en los archivos dispersos en la computadora de Estadística del Subcentro.

Las posibles desventajas que se podría encontrar desde el punto de vista del usuario, serían:

- Acostumbrarse a la nueva herramienta en su quehacer diario del Subcentro.
- Cambiar su forma de pensar en cuanto al uso de nuevas tecnologías.
- Abandonar la vieja forma de trabajar.

4.4.2. Correcciones y Mejoras

Quedan vacíos en algunas opciones que se podría corregir, como es lo que se refiere a la presentación de las páginas, en los que se puede perfeccionarlos incrementando animaciones flash para que se vea aún más animado.

El Sistema, como cualquier otro sistema tiene sus limitaciones, por lo que en base a ellas podemos enunciar algunas mejoras para el sistema.

4.4.2.1. Sitio Web en Internet.

Debido a limitaciones económicas, se decidió utilizar el sistema solamente dentro del Subcentro, por lo que para mejorar el servicio se podría subir el sistema al ciberespacio desde donde un médico o un paciente pueden realizar consultas o reservaciones desde su casa.

De igual manera, una vez subido el sistema al Internet se puede publicar eventos, cursos, charlas, etc., actividades de importancia tanto para los pacientes como para los profesionales de la salud.

De todas maneras desde cualquier punto de vista se obtendrían beneficios la subida de nuestro sistema a este lugar, pero hay que tomar en cuenta que las opciones mencionadas se deberían ser incrementados al sistema.

4.4.2.2. Ampliación de módulos.

Un sistema, normalmente no cubre las necesidades en su totalidad, de igual manera en presente sistema aún queda algunos vacíos, por lo que se podría aumentarlos incrementando nuevos módulos para tratar de satisfacer las expectativas de nuestros usuarios, como por ejemplo:

Módulo para el consultorio de odontología: El departamento de odontología guarda datos especializados del paciente específicamente para este consultorio, por lo que sería de gran ayuda para la doctora.

Módulo para el consultorio de obstetricia: De igual manera se podría realizar un módulo especializado para señoras embarazadas, poder seguir el control de maternidad, y todos los datos que tienen relación con esta especialidad.

Módulo para el control de niños recién nacidos: Este módulo llevará el control de datos médicos de niños recién nacidos, que mensualmente realizan su control y llevar el control de su evolución.

Esto ayudaría aún más al Subcentro de Salud, y cubriría la mayoría de actividades diarias que se realizan en este lugar.

4.4.2.3. Ampliación de Informes.

Al incrementar módulos no afectaríamos la estructura del mismo, por lo que también se puede incrementar y diversificar tipos de informes necesarios para el Subcentro, como por ejemplo:

- Informe de consultas de un paciente en un mes, un año o un rango de tiempo dado.
- Informe de nombres de médicos que prestaron su servicio en un rango de tiempo.
- Informe gráfico de frecuencia de atención médica diaria que se ha dado en un rango de tiempo dado.
- Informe especializado con datos médicos de un paciente, etc.

4.4.2.4. Ampliación de Seguridades.

Las seguridades, es un aspecto muy importante a la hora de mantener datos en un sistema informático, pues de igual manera en el sistema se podría aumentar aún más las seguridades para mantener la integridad de los datos.

La forma de aumentar más seguridades en este sistema sería aumentando más tipos de credenciales para diversificar más tipos de usuarios y restringir ingresos a módulos, con esto se permite el acceso solo a datos que ese usuario puede ver y no a todos, o también se puede restringir actividades como es crear backups, eliminar backups, administrar datos de pacientes, etc.

4.5. INSTALACIÓN Y CAPACITACIÓN

4.5.1. Instalación

Para proceder a la instalación no fue necesario crear un instalador, solo se necesita realizar la configuración respectiva, como es configurar apache para que funcione como localhost, esto en el caso de instalarlo a nivel local, como fue en el caso del Subcentro de salud.

En el caso de subir a un servidor en el Internet en cambio se necesita de una pequeña

aplicación para subir al servidor respectivo, existen algunos programas para realizar esta tarea.

Para proceder a la instalación y configuración respectiva se encontró un problema, debido a que tenían instalado un programa congelador (Deep freezer), el mismo que cada vez que se apaga la máquina regresa a un estado anterior, por lo que se tuvo que descongelarlo, en este caso Apache hay que instalarlo en una unidad a la que no se lo congela, para luego volver a congelar la máquina con la misma herramienta.

4.5.2. Capacitación

Después de la instalación y verificación del buen funcionamiento del sistema, se procede a capacitar a las enfermeras que tienen a su cargo atender en el departamento de Estadística, entre ellas la directora actual del Subcentro.

En esta capacitación se toma en cuenta que la máquina debe permanecer congelada, por lo que necesariamente se debe aprender el funcionamiento de esta aplicación (Deep Freezer), con esto el usuario aprenderá a visualizar los pro y contras de mantener congelada su máquina.

Un aspecto importante en cuanto a la localización de los archivos del sistema, es que aprendan en donde y cuáles son los archivos y carpetas que no se deben tocar, pues alterarían o dañarían su funcionalidad.

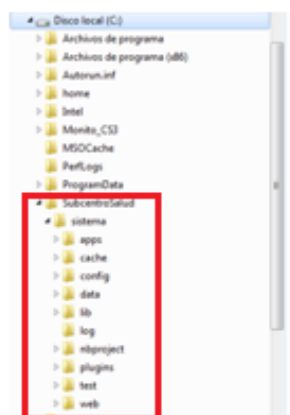


Gráfico 4.31.: Localización de la Carpeta de nuestro sistema.

Fuente: Pantalla capturada.

Es así que si el usuario borra esta carpeta o una parte de ellas el sistema dejará de

funcionar o el resultado no será el deseado.

En cuanto a la funcionalidad del sistema, la actividad de cada módulo se lo relacionó con la actividad manual que a diario se realiza en el Subcentro de Salud por lo que no hay mucho problema en captarlos, solo que se puso más énfasis en la administración de historias clínicas y turnos, pues esos puntos son los que más interesa al personal.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES



5.1. Conclusiones

5.2. Recomendaciones

5.1. CONCLUSIONES

Basado al trabajo realizado con el Subcentro de Salud San Antonio de la Parroquia San Antonio de Ibarra, se puede sacar las siguientes conclusiones:

- ✓ En la actualidad, la mayoría de instituciones públicas o privadas ya cuentan con equipos de computación para realizar su trabajo, pero que por limitaciones económicas o falta de interés aún no cuentan con un sistema informático para realizar un tratamiento adecuado a la información que manejan, muchos de ellos porque el tamaño de información no amerita un sistema como estos, pero que con el pasar del tiempo se ve que poco a poco se les hace necesario debido a su crecimiento.
- ✓ El Subcentro de Salud San Antonio es el eje principal de salud que tiene nuestra parroquia, es por eso que el número de historias han crecido, pues acá acuden pobladores de sus barrios aledaños.
- ✓ El crecimiento de la población hace también que los servicios de salud requieran una urgente actualización en cuanto a herramientas para el tratamiento de información para no generar aglomeraciones.
- ✓ El Subcentro de Salud San Antonio, viene trabajando con un proceso establecido y hasta el momento le ha dado buenos resultados, pero con el crecimiento de sus requerimientos de atención médica está a punto de colapsar. Por lo que es urgente automatizar el tratamiento de su información con un sistema adecuado para ello.

En lo que se refiere a las herramientas que se utilizaron para realizar el sistema dirigido al tratamiento de historias clínicas para el Subcentro de Salud San Antonio, se puede sacar las siguientes conclusiones:

- ✓ Existen herramientas libres en la actualidad que brindan grandes ventajas para el diseñador de sitios web como en nuestro caso fue Symfony.
- ✓ Un framework automatiza las tareas más comunes por lo que el desarrollo de aplicaciones se simplifica en gran medida.
- ✓ Un framework cambia el criterio de un programador pues proporciona una estructura al código fuente, logrando así crear código más legible y más fácil de mantener.

- ✓ Con Symfony podemos trabajar en proyectos web grandes sin mucha dificultad, pues esta herramienta nos simplifican estas tareas y nos facilita nuestro trabajo.
- ✓ En el caso de que se vaya a diseñar proyectos bastante pequeños, solo en estos casos tal vez no se amerite utilizar frameworks.
- ✓ La comunidad Symfony lanza plugins que ayuda a los desarrolladores a acelerar su trabajo, pues estos se bajan desde su página oficial listos para aplicarlos en nuestro sitio, previo unas pequeñas configuraciones para acoplarlos al nuestro. Además podemos diseñar nuestros propios plugins para volver a reutilizarlos en nuestros proyectos, los mismos que nos servirían solo para nosotros (plugins privados), pues atienden nuestros requerimientos.
- ✓ En la actualidad podemos observar que tenemos herramientas libres que nos ayudan a trabajar en nuestros proyectos de desarrollo de software, con los cuales podemos crear software de gran calidad, iguales y en muchos casos superiores al software creado con herramientas pagadas.

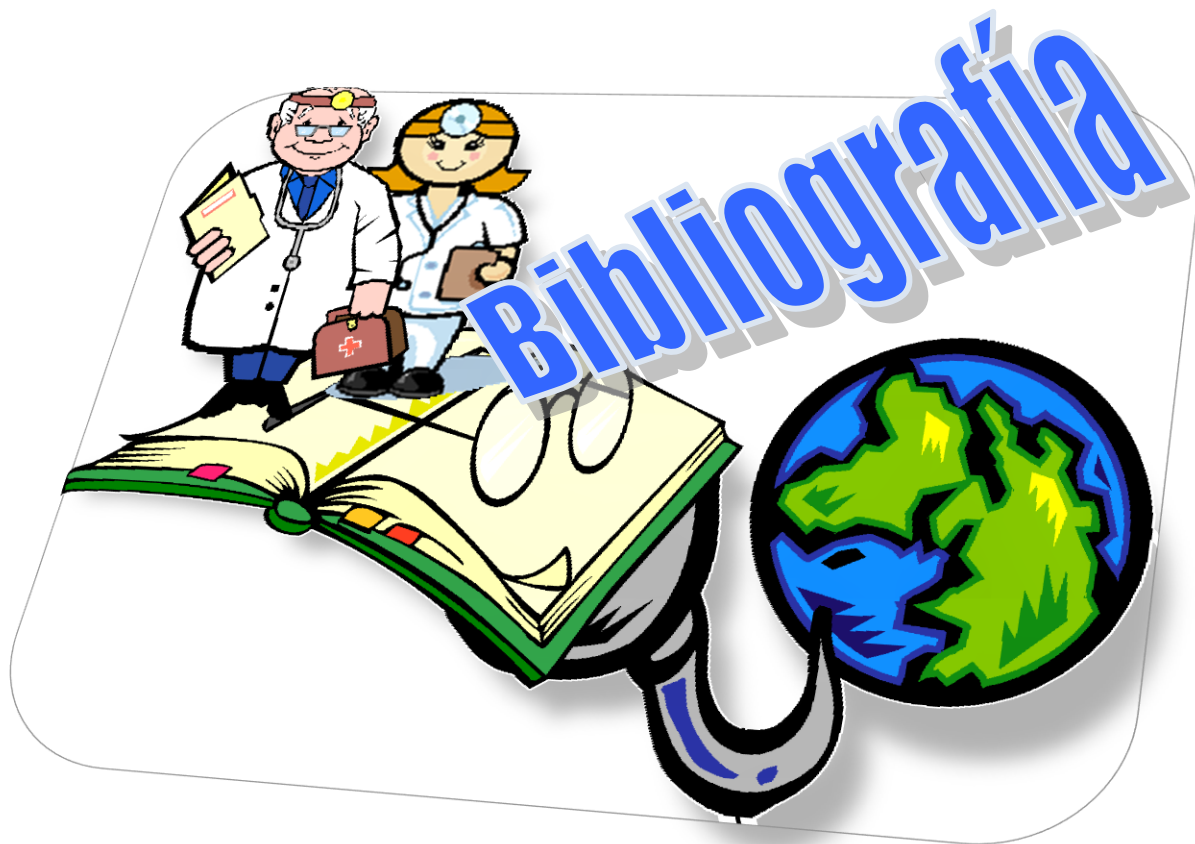
5.2. RECOMENDACIONES

Luego de haber terminado el trabajo en lo que se refiere al diseño del sitio web para el Subcentro de salud San Antonio, y ver el importante aporte a la comunidad brindando su trabajo diario que se realiza en esta casa de salud y de haber obtenido una buena experiencia con la aplicación de Symfony como una nueva herramienta para el desarrollo de sitios, basado en esto podemos recomendar:

- ✓ Se recomienda concientizar a las personas que van a ser usuarios de una herramienta tecnológica no cerrarse en la negativa de su aplicación, pues estas van a ayudar a agilizar su trabajo.
- ✓ Recomendamos subir el sitio web al Internet para que se pueda ampliar el servicio a los médicos y pacientes, implementando más seguridades en ella.
- ✓ Dar un mantenimiento constante al sistema, actualizándolo poco a poco para que a futuro pueda cubrir la mayor parte de actividades del Subcentro y poder obtener un mejor servicio de parte del mismo.
- ✓ Se recomienda mantener una disciplina de sacar copias de seguridad de la Base de datos por si en algún momento falle el sistema.

- ✓ Recomendamos utilizar frameworks para el desarrollo de sitios web, pues este tipo de herramientas nos ayudarán a acelerar nuestro trabajo en el desarrollo de los mismos.
- ✓ Recomendamos también el uso de Symfony para el desarrollo de sitios web de un tamaño considerable, pues para pequeños proyectos no tiene caso su uso.
- ✓ Se recomienda a los estudiantes de nuestra carrera seguir investigando sobre este framework (Symfony), pues ofrece grandes beneficios que aún no se explota en nuestro sector, como por ejemplo el diseño y uso de plugins para sus proyectos es uno de ellos.
- ✓ Recomendamos también la constante actualización de los estudiantes, futuros profesionales de nuestra Facultad, pues cada vez más están saliendo herramientas que ayudan a realizar software de calidad sin emplear mucho tiempo y esfuerzo.

BIBLIOGRAFÍA



INTERNET:

- **INFORMACIÓN DE SAN ANTONIO DE IBARRA**
[WWW.001]
<http://www.sanantonio.gov.ec/es/ubicacion.php>
- **INFORMACIÓN DE TECNOLOGÍAS WEB**
[WWW.002]
[http://es.wikipedia.org/wiki/Portal_\(Internet\)](http://es.wikipedia.org/wiki/Portal_(Internet))
- [WWW.003]
<http://mundogeek.net/>
- **INFORMACIÓN GENERAL DE HERRAMIENTAS LIBRES**
[WWW.004]
[\[http://netbeans.org/index_es.html\]](http://netbeans.org/index_es.html)
- [WWW.005]
<http://www.orcero.org/irbis/blog/busqueda-y-seleccion-de-personal-en-empresas-de-software-libre/>
- [WWW.006]
http://linux.ciberaula.com/articulo/linux_apache_intro/
- [WWW.007]
http://www.webtaller.com/manual-php/introduccion_php.php
- [WWW.008]
http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado
- [WWW.009]
<http://softwarelibre.nireblog.com/post/2008/06/17/generalidades>
- **INFORMACIÓN DE LICENCIAS DE SOFTWARE LIBRE**
[WWW.010]
http://es.wikipedia.org/wiki/FOSS_-_FLOSS
- [WWW.011]
<http://www.aulaclie.es/articulos/licencias.html>
- **INFORMACIÓN DE SOFTWARE LIBRE EN ECUADOR**
[WWW.012]
http://www.estebanmendieta.com/blog/wp-content/uploads/Decreto_1014_software_libre_Ecuador.pdf
- [WWW.013]
<http://drwn.wordpress.com/2008/04/11/decreto-1014-software-libre-en-ecuador/>
- [WWW.014]
<http://www.somoslibres.org/modules.php?name=News&file=article&sid=2767>

[WWW.015]

<http://www.itecuador.com/2009/09/gobierno-de-ecuador-y-el-uso-de-software-libre/>

[WWW.016]

<http://www.dspace.espol.edu.ec/bitstream/123456789/758/1/1398.pdf>

[WWW.017]

http://www.aesoft.com.ec/index.php?option=com_content&task=view&id=15&Itemid=27

- **INFORMACIÓN DE SERVIDORES WEB**

[WWW.018]

<http://www.descargasdirect.com/index.php/component/virtuemart/details/84/133/windows/servidores/wamp-server-2.0i>

- **INFORMACIÓN DE SYMFONY**

[WWW.019]

http://www.librosweb.es/symfony_1_2/capitulo1/symfony_en_pocas_palabras.html

[WWW.020]

http://www.librosweb.es/symfony_1_2/capitulo1/symfony_en_pocas_palabras.html

[WWW.021]

<http://www.tecnoretas.com/programacion/mi-primer-proyecto-con-symfony-127-parte-i/comment-page-1/>

[WWW.022]

<http://www.tecnoretas.com/programacion/mi-primer-proyecto-con-symfony-127-parte-i/comment-page-1/>

[WWW.027]

<http://www.forosdelweb.com/f68/crear-proyecto-symfony-902142/>

[WWW.028]

http://www.symfony-project.org/jobee/1_2/Propel/es/20

[WWW.029]

http://www.symfony-project.org/jobee/1_2/Doctrine/es/20

- **INFORMACIÓN SOBRE NETBEANS**

[WWW.023]

http://netbeans.org/index_es.html

[WWW.024]

<http://www.kickbill.com/?p=572>

[WWW.025]

<http://jtagua.wordpress.com/2010/06/17/descarga-e-instalacion-separada-de-jdk-6-update-20-y-netbeans-6-9-windows/>

[WWW.026]

<http://netbeans.org/downloads/index.html>.

LIBROS

- HOLZNER, Steven, PHP: MANUAL DE REFERENCIA, McGraw-Hill, Año 2009.
- SABANA MENDOZA, Maribel; PHP CON POSTGRES SQL8, Megabyte, Año 2008.
- MERCER, Dave W. & KEND, Allan & NEWICK, Steven D.; FUNDAMENTOS PHP5, Anaya, Año 2005.
- CHARTE OJEDA, Francisco; PROGRAMACIÓN PHP5, grupo Anaya, Año 2007.
- SHULZ, Ralph G.; DISEÑO WEB CON CSS, Marcombo, Año 2009.
- VILLAR, Cruz, DISEÑO WEB PASO A PASO, Megabyte, Año 2009.
- DOMINGUEZ, Silvia & SÁNCHEZ, Enrique & SÁNCHEZ, Gabriela; GUÍA PARA ELABORAR UNA TESIS, Mc Graw Hill, Año 2009.
- POTENCIER, Fabien & ZANINOTTO, François, SYMFONY 1.0, LA GUÍA DEFINITIVA, Creative Commons, Año 2008.
- POTENCIER, Fabien & ZANINOTTO, François, SYMFONY 1.2, LA GUÍA DEFINITIVA, Creative Commons, Año 2008.
- PÉREZ, Javier Eguíluz; INTRODUCCIÓN A AJAX, Creative Commons, Año 2008.