



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
Facultad 5

**SISTEMA PARA LA GESTIÓN DE LA INFORMACIÓN DE LOS MEDICAMENTOS, TURNOS
MÉDICOS Y VACUNAS EN EL HOSPITAL ERNESTO CHÉ GUEVARA.**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS**

Autores: Isnauty Pérez Téllez
Yasmany González Sañudo

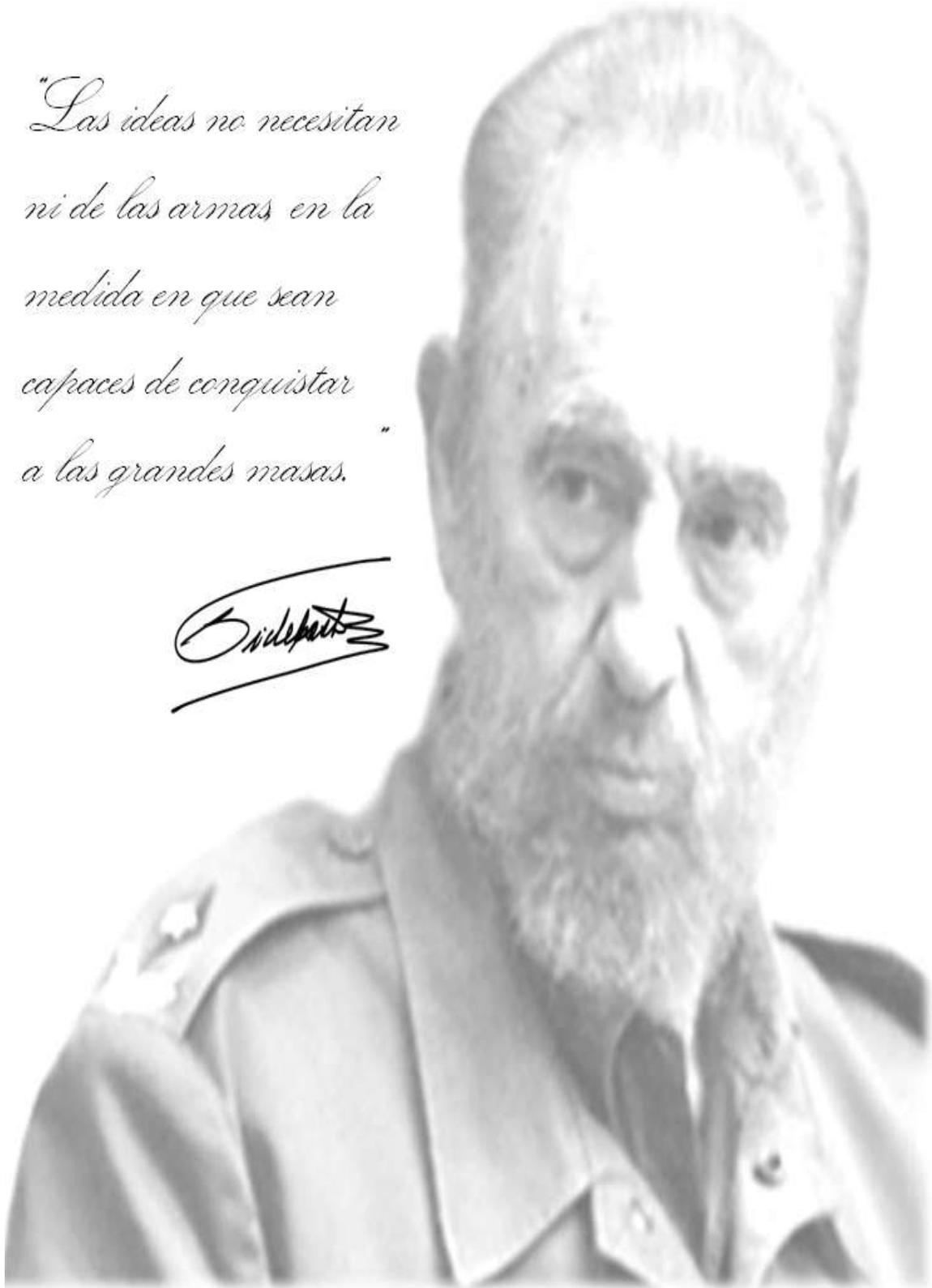
Tutor: Ing. Saily Salas Hechavarria

Consultantes: Ing. Roxana Martínez Ferrer
Ing. Danay Leyva Abrahantes

La Habana, 8 de junio de 2012

*"Las ideas no necesitan
ni de las armas, en la
medida en que sean
capaces de conquistar
a las grandes masas."*

Ortiz



Dedicatoria de Isnaudy:

*A las personas que más quiero en la vida y las que nunca me van a fallar:
A mis padres porque sin ellos hoy no estaría hoy aquí, porque ellos son los que me
dan fuerzas para seguir adelante cada día.*

A mi hermana que siempre está a mi lado en las buenas y en las malas.

A toda mi familia por confiar en mí.

A mi padrino que con tanta paciencia me ha cuidado siempre.

*En especial a mi novia **Lisandra** que es una de las cosas más bellas que me ha
dado la vida, gracias por los momentos inolvidables que pasé a tu lado.*

Dedicatoria de Yasmany:

*Primeramente a mis padres, a quienes le debo todo lo que soy y los logros
alcanzados porque han sabido dar todo de ellos para mi bienestar de manera
incondicional desde que nací y aún más durante mi carrera.*

*A mis abuelos, especialmente a Grasiela que lamentablemente no pudo estar aquí
conmigo pero que se dedicó por completo a mí, a educarme, mimarme y hasta mal
criarme.*

*A mis compañeros de grupo, por enseñarme el valor de la amistad,
No por mencionarla al final es menos importante, a **Prianna** que me ha dado todo
su cariño desde el primer día, de quien he aprendido el verdadero significado de la
palabra "amor y me ha regalado la alegría más grande del mundo **Brianna**.*

Agradecimientos:

A las tutoras por toda la ayuda y el apoyo que nos han brindado.

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo “Sistema para la gestión de la información de los medicamentos, turnos médicos y vacunas en el hospital de la Universidad de Ciencias Informáticas” y autorizamos a la Facultad 5 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año _____

Autores:

Tutor:

Consultantes:

RESUMEN

El surgimiento y desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) y su incorporación al área de la salud, ha alcanzado un espacio importante en la sociedad. Actualmente para la gestión de la información de los servicios que se brindan en los hospitales del mundo se han desarrollado diferentes sistemas que facilitan el trabajo en estos centros. Cuba no ha quedado atrás en este sector, ya que la empresa cubana de desarrollo de software “SOFTEL” ha implementado herramientas que se encargan de manipular la información en los centros de salud. Estas herramientas han servido de gran ayuda en varios hospitales de Cuba, incluyendo el hospital Ernesto Ché Guevara.

Sin embargo, existen insuficiencias para la gestión de la información de los servicios que se brindan en el hospital debido a que la herramienta no se está utilizando por problemas tecnológicos y de soporte. Atendiendo a estas insuficiencias se desea desarrollar el Sistema de Gestión Hospitalaria (SIGESH) que permita la gestión de la información de los medicamentos, vacunas y turnos médicos en dicho centro hospitalario.

El resultado principal está encaminado a facilitar la gestión de la información de los servicios que se brindan en el hospital, sin necesidad de tener que utilizar herramientas externas que no son desarrolladas por la universidad, a la cual se le pueda realizar cambios en dependencia de las necesidades que existan en ese momento.

Palabras clave: gestión, herramientas, información, software.

ABSTRACT

The emergence and development of Information Technologies and Communications (TIC) and its incorporation into the area of health, has achieved an important place in society. Currently for the management of information services provided in hospitals of the world have developed different systems that make working in these centers. Cuba has not lagged behind in this sector, since the Cuban enterprise software development "SOFTEL" has implemented tools that are responsible for manipulating information in health centers. These tools have been helpful in several hospitals in Cuba, including the hospital Ernesto Ché Guevara.

However, there are inadequacies in the information management services provided at the hospital because the tool is not being used by technology problems and supportive. In response to these shortcomings is to be developed Hospital Management System (SIGESH) to enable information management of medicines, vaccines and medical appointments in the hospital.

The main result is intended to facilitate information management services provided at the hospital, without having to use external tools that are developed by the university, which you can make changes depending on the needs that exist at that time.

Keywords: management, tool, information, software.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	5
INTRODUCCIÓN.....	5
1.1 <i>Conceptos asociados al dominio del problema</i>	5
1.2 <i>Soluciones similares</i>	7
1.3 <i>Lenguajes de programación</i>	10
1.4 <i>Herramientas de diseño y desarrollo</i>	12
1.5 <i>Servidor web</i>	14
1.6 <i>Gestor de bases de datos</i>	15
1.7 <i>Metodologías, herramientas y lenguajes de modelado en el proceso de desarrollo de software</i>	17
1.7.1 <i>Metodología de desarrollo de software</i>	17
1.7.2 <i>Lenguaje de modelado</i>	20
1.7.3 <i>Herramientas CASE para el modelado</i>	20
CONSIDERACIONES PARCIALES.....	22
CAPÍTULO 2 DEFINICIÓN DEL SIGESH.....	23
INTRODUCCIÓN.....	23
2.1 <i>Objetos a informatizar</i>	23
2.2 <i>Propuesta de solución del sistema</i>	23
2.3 <i>Actores del sistema</i>	23
2.4 <i>Funcionalidades del sistema</i>	24
2.4.1 <i>Lista de reserva del producto</i>	24
2.5 <i>Requisitos no funcionales</i>	27
2.6 <i>Descripción de las historias de usuario</i>	28
2.7 <i>Plan de iteraciones</i>	30
CONSIDERACIONES PARCIALES.....	32
CAPÍTULO 3 DESARROLLO Y PRUEBAS DEL SIGESH.	33
INTRODUCCIÓN.....	33
3.1 <i>Patrones de diseño</i>	33
3.2 <i>Diseño del sistema</i>	35
3.2.1 <i>Tarjetas CRC</i>	35
3.3 <i>Arquitectura de la solución propuesta</i>	38
3.3.1 <i>Patrón de arquitectura</i>	38
3.3.2 <i>Estructura del SIGESH</i>	40
3.3.3 <i>Descripción de los módulos</i>	41
3.4 <i>Diseño del esquema de base de datos del sistema</i>	44

3.5	<i>Estándar de codificación utilizada</i>	47
3.6	<i>Distribución física del SIGESH</i>	49
3.7	<i>Validación de las funcionalidades</i>	50
3.7.1	<i>Resultados de las Pruebas</i>	56
CONSIDERACIONES PARCIALES.....		57
CONCLUSIONES GENERALES		58
RECOMENDACIONES		59
REFERENCIAS BIBLIOGRÁFICAS		60
GLOSARIO DE TÉRMINOS		63

ÍNDICE DE FIGURAS

Figura 1 Tarjeta CRC. Paciente	35
Figura 2 Tarjeta CRC. Vacuna.....	36
Figura 3 Tarjeta CRC. Consulta.....	36
Figura 4 Tarjeta CRC. Producto	37
Figura 5 Tarjeta CRC. Vacunación.....	38
Figura 6 Funcionamiento del patrón modelo-vista-controlador	39
Figura 7 Estructura del SIGESH.....	41
Figura 8 Estructura de directorios de un módulo	42
Figura 9 Modelo físico de datos del esquema hospital	46
Figura 10 Modelo físico de datos del esquema seguridad.....	47
Figura 11 Diagrama de despliegue del sistema SIGESH.....	50

ÍNDICE DE TABLAS

Tabla 1 HU1.Gestionar medicamento en la Tarjeta de Estiba.....	28
Tabla 2 Plan de Iteraciones de las HU.....	30
Tabla 3 Casos de prueba. Gestionar Medicamento en Tarjeta de Estiba	52
Tabla 4 Casos de prueba. Modificar medicamento en Tarjeta de Estiba	53
Tabla 5 Casos de prueba. Eliminar medicamento en Tarjeta de Estiba.....	55
Tabla 6 No conformidades.....	56

INTRODUCCIÓN

La utilización de las Tecnologías de la Información y las Comunicaciones (TIC) en instituciones, empresas y centros educacionales como apoyo a los servicios que brindan, están teniendo gran auge a nivel mundial. Su desarrollo y aplicación ha abierto el camino hacia una mejor comunicación entre los involucrados, ya que facilita el trabajo y permite de manera más eficaz gestionar toda la información que se desea controlar.

En el ascenso de sectores económicos y sociales, la presencia de sistemas informáticos se ha vuelto ineludible. El sector de la salud no está ajeno a dicho desarrollo y hoy en día en el mundo existen varios sistemas que permiten gestionar la información de los centros de salud, entre los que se destacan: **“SIHOWIN”** y **“SOFTMED”**.

Estos sistemas informáticos permiten llevar el control de la administración de hospitales y clínicas, para operar en red de cómputo. Automatizan el manejo y vigilancia a pacientes, almacenes, quirófanos, balance de cargas de trabajo en áreas médicas y de enfermería, facturación, compras, pago a proveedores, cobranza, auditoría interna, cambio de turno de enfermería, hospitalización y pólizas contables. Aunque estos sistemas son utilizados para este tipo de servicio, al ser desarrollados por países capitalistas son privativos y no cumplen con las necesidades específicas de un centro médico como el hospital Ernesto Ché Guevara.

En estudios realizados se pudo constatar que el centro de salud está atravesando una etapa de cambios, en la que se decidirá cuales áreas no continuarán formando parte del mismo, aunque existen algunas como el almacén de medicamentos, vacunación y admisión, que por su importancia seguirán brindando sus servicios, además requieren de un estricto control.

Los departamentos de Economía y Estadísticas de dicho centro son los encargados de llevar a cabo la gestión de la información de los medicamentos, turnos médicos y vacunas. Por su parte, en Economía se

gestiona la distribución de los medicamentos realizada por el almacén de fármacos, mientras Estadísticas se encarga de los reportes emitidos por las áreas de vacunación y admisión.

En entrevistas realizadas a diferentes trabajadores del hospital se confirmó que aunque existe un sistema desarrollado por “SOFTTEL” para facilitar la gestión de la información de los servicios que se brindan en el centro hospitalario, no se está utilizando por problemas tecnológicos y de soporte, además no cumple con la necesidades específicas del centro de salud pues solo controla el área del almacén y al no ser desarrollado por la Universidad de las Ciencias Informáticas(UCI) no se puede realizar ninguna modificación sobre el software.

Como consecuencia, existe atraso en la entrega de los reportes emitidos por los departamentos del hospital, ya que se procesa gran cantidad de información de forma manual, dificultando el almacenamiento y consulta de la misma. Esto provoca que el trabajo del personal sea muy engorroso, se gaste gran cantidad de recursos de oficina innecesariamente y resulte complicado llevar un control actualizado de los medicamentos, vacunas y turnos médicos.

Por otra parte, los pacientes muestran inconformidad al solicitar turnos médicos ya que existe demora en el servicio y en ocasiones tienen que retirarse sin ser atendidos. Además, se evidencia descoordinación y atraso en la vacunación de los pacientes, lo que conlleva a que se repitan u omitan algunas vacunas, así como no se apliquen las reactivaciones en la fecha correspondiente.

A raíz de la problemática antes expuesta, se plantea el siguiente **problema a resolver**: ¿Cómo informatizar la gestión de la información de los servicios en el hospital Ernesto Ché Guevara?.

El problema planteado se enmarca en el **objeto de estudio**: Proceso de gestión de la información en los centros de salud.

En la investigación se concibe como **campo de acción**: Proceso de desarrollo de sistemas para la gestión de la información de los servicios que se prestan en el hospital Ernesto Ché Guevara.

Para dar solución al problema identificado se define el siguiente **objetivo general**: Desarrollar un sistema informático que permita la gestión de la información de los medicamentos, turnos médicos y vacunas en el hospital Ernesto Ché Guevara.

Para dar cumplimiento al objetivo general se trazaron las siguientes **tareas de investigación**:

- Sustentación teórico metodológico sobre los aspectos característicos para la gestión de la información de los medicamentos, turnos médicos y vacunas en el hospital Ernesto Ché Guevara.
- Determinación del negocio para un mejor entendimiento del sistema a desarrollar.
- Caracterización de las tecnologías y estándares más utilizados para la creación de sistemas para la gestión de medicamentos, turnos médicos y vacunas.
- Diseño de un sistema que permita gestionar la información de los medicamentos, turnos médicos y vacunas en el hospital Ernesto Ché Guevara.
- Diseño del sistema de almacenamiento de datos para el manejo y control de la información.

Desde el punto de vista metodológico se emplearon los siguientes métodos científicos:

Métodos teóricos

Análítico-Sintético: Utilizado para analizar elementos bibliográficos y definiciones sobre gestión de la información de servicios en hospitales, con el objetivo de arribar a conclusiones que sustenten la necesidad de la investigación.

Análisis Histórico-Lógico: Es utilizado para analizar la evolución histórica de soluciones similares, las tendencias más recientes de la gestión de la información de los servicios en hospitales y basado en esos datos, tomarlos como guía para el desarrollo de la presente investigación.

Modelación: Se utiliza en la modelación de los diagramas que propone la metodología de desarrollo de software seleccionada para llevar a cabo la solución.

Métodos empíricos

Observación: Se realiza el estudio de las características y comportamientos de los usuarios de soluciones similares, permitiendo así la formulación global de la investigación.

Entrevista: Se aplica a trabajadores del hospital Ernesto Ché Guevara para detectar deficiencias y necesidades en la gestión de la información en este centro. Su instrumento (guía de entrevista) está ubicado en los anexos. Ver anexo 1. Entrevistas realizadas a expertos del hospital.

Estructura capitular

Capítulo 1: Fundamentación Teórica

En este capítulo se abordan los principales aspectos relacionados con los servicios que se brindan en los centros de salud. Además se realiza un estudio del estado del arte para la creación de sistemas para la gestión de la información de los servicios que se brindan en los hospitales. Por último se precisan los conceptos teóricos que rigen la investigación, se describen las técnicas, tecnologías, herramientas y se seleccionan las que se van a utilizar en el desarrollo de la solución propuesta.

Capítulo 2: Definición del SIGESH

En este capítulo se identifican los actores del sistema, las funcionalidades y los requisitos no funcionales, así como los procesos a informatizar. Además se describen las historias de usuario para cada funcionalidad y se realiza el plan de iteraciones.

Capítulo 3: Desarrollo y pruebas del SIGESH

En este capítulo se detallan los elementos necesarios para la implementación. Se elaboran las tarjetas Clase-Relación-Colaborador(CRC), se seleccionan los patrones de diseño y arquitectura a utilizar para la realización del sistema SIGESH. Además se realiza el diagrama de despliegue y las validaciones de las funcionalidades del sistema para comprobar que este se ajusta a las necesidades del cliente.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

Introducción

Las nuevas TIC, como los sistemas de gestión, son herramientas que facilitan los procesos de gestión de la información de los servicios que se brindan en los hospitales. En el presente capítulo se exponen los conceptos relacionados con la fundamentación teórica de la presente investigación y la existencia de soluciones informáticas similares. Además se realiza una descripción de las tecnologías a utilizar para dar solución al problema planteado, que incluye: lenguajes de programación, entorno de desarrollo integrado, metodología de desarrollo de software, herramientas de modelado, entre otras que se detallan más adelante.

1.1 Conceptos asociados al dominio del problema

Los conceptos asociados al dominio se seleccionan a partir de las definiciones o conceptos más actualizados establecidos por organizaciones tanto nacionales como internacionales dedicadas al desarrollo de sistemas de gestión de información de hospitales. Estos conceptos proporcionan un mejor entendimiento del problema a tratar en la investigación.

Hospital

Según el Diccionario de Ciencias Médicas Dorland, el Hospital es un establecimiento público o privado en el que se curan los enfermos, en especial aquellos carentes de recursos. Es una institución organizada, atendida y dirigida por personal competente y especializado, cuya misión es la de proveer, en forma científica, eficiente, económica o gratuita, los complejos medios necesarios para la prevención, el diagnóstico y el tratamiento de las enfermedades y para dar solución, en el aspecto médico, a los problemas sociales. (Flores, 2009)

Gestión de información

La gestión de información es el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, así como para mejorar los procesos, productos y servicios de la organización. Incluye además, operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma. (Curto, 2006)

Gestión de información en hospitales

Un Sistema de Información Hospitalario (del inglés HIS), también llamado Sistema de Información Clínica (del inglés CIS), es un sistema de información integrado, diseñado para manejar los aspectos administrativos, financieros y clínicos de un hospital.

Los sistemas de gestión hospitalaria constituyen un sistema jerárquico formado por otros subsistemas, dentro de los cuales se identifican los sistemas de información epidemiológica, sistemas de información económica financiera y administrativa, sistemas de información socio-demográfica y social, y sistemas de información de carácter clínico. (Nogueira, 2009)

Servicio

Es un conjunto de actividades que buscan responder a necesidades de un cliente. Se define un marco donde las actividades se desarrollarán con la idea de fijar una expectativa en el resultado de éstas. Es el equivalente no material de un bien. Un servicio se diferencia de un bien (físico o intangible) en que el primero se consume siempre en el momento en que es prestado.

Stanton William, Etzel Michael y Walker Bruce en el libro "Fundamentos de Marketing" definen servicios como "Actividades identificables e intangibles que son el objeto principal de una transacción ideada para brindar a los clientes satisfacción de deseos o necesidades" (Stanton William, 2004)

Vacuna

Es un preparado a base de virus o bacterias muertas o debilitadas, o partes de ellas, que se introducen en el cuerpo para que éste lo reconozca y desarrolle anticuerpos. Los anticuerpos son partículas de nuestro sistema inmunológico que reconocen a los cuerpos extraños al organismo y los neutralizan.

Con una vacuna el sistema inmunológico del organismo adquiere una "memoria celular" que le permite responder pronto ante una infección que no se ha padecido previamente de forma natural, evitando la enfermedad. (Pino, 2009)

Medicamento

Los medicamentos son una sustancia o combinación de estas que pueden administrarse a seres humanos con el fin de restaurar, corregir o modificar las funciones fisiológicas ejerciendo una acción farmacológica, inmunológica o metabólica, o de establecer un diagnóstico médico. (Vidal, 2009)

Aplicación informática

Se conoce como software o programa informático al equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos del sistema, llamados hardware.

Una aplicación informática es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajo. Suele resultar una solución informática para la automatización de ciertas tareas complicadas como pueden ser la contabilidad, la redacción de documentos, o la gestión de un almacén. Algunos ejemplos de programas de aplicación son los procesadores de textos, hojas de cálculo, base de datos, entre otros. (EcuRed, 2011)

1.2 Soluciones similares

Mediante investigaciones realizadas se pudo constatar que existen varios sistemas que permiten gestionar la información de los centros de salud, ejemplo de ellos son: “**SIHOWIN**”, “**SOFTMED**”, “**SCMH4**” y “**GALEN LAB**”.

SIHOWIN

Es un sistema de cómputo diseñado para aquellos hospitales, clínicas o sanatorios que necesiten llevar el control de sus pacientes, almacenes, cuartos, médicos y cobranza, siendo una herramienta poderosa y de fácil manejo. Entre sus características se encuentran:

- Opera en red.
- Transacciones en línea, Ambiente gráfico.
- Exportación ASCII y Office.
- Catálogos precargados y Reportes pre configurados.
- Parametrizable.
- Sin límite de usuarios por Instalación. (GACSE SYSTEMS, 2008)

SOFMED

Es una herramienta desarrollada para satisfacer las necesidades de distintas especialidades médicas, entre ellas: Medicina General, Ginecología, Otorrinolaringología, Pediatría, Neonatología, Traumatología, Endocrinología, Anatomía Patológica Y Urología.

Características de SOFMED:

- Creado en Access y Visual Basic para aplicaciones.
- Integración total con Word, Excel y posibilidad de enviar por Fax.
- Definición de fechas y horario de atención por parte del usuario.
- Definición de precios por procedimiento para cada entidad.
- Manejo de los ingresos y egresos del consultorio.

- Agenda telefónica y recordatorio. (División Scientia Pharma, 2005)

SCMH4

Es una herramienta diseñada para gestionar los medicamentos en los almacenes de los centros de salud a través de Tarjetas de Estiba, proporcionando un mayor control de las entradas y salidas que se realizan en el almacén, la misma esta compuesta por un menú principal con cinco opciones generales.

- **Menú Control:** Permite la configuración de usuarios, chequeo de la información y movimientos históricos.
- **Menú Catálogos:** Permite la gestión de los costos, destinos y productos.
- **Menú Operaciones:** Permite llevar el control de la entrada y salida de medicamentos, así como el informe diario.
- **Menú Consultas:** Permite llevar el control de los movimientos del día, vencimientos y el inventario de productos.
- **Menú Ayuda:** Ofrece una ayuda detallada de las funcionalidades del sistema. (González, 2006)

GALEN LAB

Está dirigido a la gestión de los medios de diagnóstico, facilitando la solicitud de exámenes, registro y evaluación de los resultados obtenidos, así como la generación de información estadística.

Diseñado para ser utilizado por los técnicos, médicos y personal administrativo de medios de diagnóstico para optimizar el trabajo y elevar la eficiencia. Cuenta con un sistema de ayuda en línea que brinda al usuario toda la información que necesite sobre el proceso que está efectuando y garantiza un estricto control de la información, diferenciando para cada uno de ellos las opciones disponibles según su responsabilidad. Entre sus funcionalidades se encuentran:

- Solicitud de exámenes.
- Asignación de turnos para realización de exámenes.

- Buscar solicitud de examen.
- Registrar nuevo paciente.
- Verificar muestras.
- Obtener lista de muestras pendientes.
- Obtener la Lista de trabajo interna.
- Obtener lista de trabajo externa.
- Calcular exámenes.
- Obtener informe de resultados.
- Reportes generales y estadísticos. (SOFTEL, 2004)

“**SIHOWIN**”, “**SOFTMED**” y “**GALEN LAB**” no son herramientas multiplataforma característica que limita su uso, además son propietarias y esto representa un gran inconveniente ya que el usuario que adquiere este software depende 100% de la empresa propietaria lo cual imposibilita su redistribución, copia o modificación. “**SCMH4**” es una herramienta de escritorio por tanto depende del sistema operativo que utilice el ordenador y sus características (video, memoria). Su acceso se limita al ordenador donde está instalada.

A pesar de la variedad de funcionalidades que presentan estos sistemas y su gran utilidad en centros de salud, no cumplen con las necesidades específicas que presenta el hospital Ernesto Ché Guevara, por lo que se decidió implementar un nuevo sistema que centralice la información de los servicios y que se ajuste a las características del centro de salud.

1.3 Lenguajes de programación

Un lenguaje de programación es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

HTML

HTML (del inglés HyperText Markup Language) es el lenguaje de marcado de hipertexto predominante para la elaboración de páginas Web. Es un lenguaje muy simple que se utiliza para describir la estructura y el contenido de las páginas en forma de texto incluyendo imágenes, animaciones y otros medios soportados. El formato de los documentos se marca mediante etiquetas que indican el comienzo y el final de los elementos que componen el mismo, teniendo cada uno de ellos un significado estructural diferente. Aunque HTML es un estándar muy usado se considera un lenguaje estático, basado en etiquetas y muy limitado al no poder ser ampliado. (Mora, 2010)

CSS

CSS es un lenguaje utilizado en la presentación de documentos HTML. Un documento HTML viene siendo coloquialmente “una página web”. Entonces podemos decir que el lenguaje CSS sirve para organizar la presentación y aspecto de una página web. Este lenguaje es principalmente utilizado por parte de los navegadores web de internet y por los programadores web informáticos para elegir multitud de opciones de presentación como colores, tipos y tamaños de letra. (Sierra, 2008)

JavaScript

JavaScript es un lenguaje de programación del lado del cliente muy utilizado en el desarrollo web tanto para la realización de pequeñas tareas, como para la gestión de complejas aplicaciones.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador que lo soporte sin necesidad de procesos intermedios. A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. (Pérez, 2009)

PHP

PHP (Hypertext Preprocessor), es un lenguaje interpretado, diseñado para la creación de páginas web dinámicas con acceso a información almacenada en bases de datos. La naturaleza de código abierto (open-source) de PHP lo muestra como una alternativa de fácil acceso para todos. PHP contiene funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos, posee manejo de excepciones, posibilita la generación de archivos PDF, así como imágenes en diferentes formatos.

Una de sus ventajas más notable resulta el consumir pocos recursos. Soporta en cierta medida la orientación a objetos y es capaz de conectarse con la mayoría de los manejadores de bases de datos que se utilizan en la actualidad. Por otra parte, entre sus desventajas se pueden mencionar que para la realización de proyectos complejos requiere un alto nivel de experiencia, y que por determinadas limitaciones del lenguaje existe un pequeño grupo de tareas que resultan complejas de realizar. (Alvarez, 2001)

1.4 Herramientas de diseño y desarrollo

IDE NetBeans

NetBeans es un entorno de desarrollo integrado, extensible para el desarrollo sobre varios lenguajes, aunque se realizó fundamentalmente para Java. Es un producto de código abierto desarrollado por la compañía Sun Microsystems en el año 2000. Se basa en una filosofía modular, lo que permite el desarrollo de múltiples proyectos con el uso de varias tecnologías. Entre sus principales ventajas se encuentran:

- Administración de interfaces de usuario.
- Integración a múltiples framework.
- Administración de almacenamiento.
- Gran cantidad de módulos y extensiones para múltiples lenguajes y tecnologías.

- Fuerte comunidad de respaldo. (Domínguez, 2005)

Ext JS Framework

Ext JS es una librería de JavaScript ligera, de alto rendimiento para el desarrollo rápido de aplicaciones web *cross-browser*. Presenta una interfaz de usuario personalizable, bien diseñada y contiene un modelo de componentes extensibles.

Tiene disponible licencias comerciales y de código abierto. Es compatible con la mayoría de los navegadores y su familia de productos es utilizada por miles de compañías en el mundo. 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. (EcuRed, 2007)

Symfony Framework

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. Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en el sistema operativo Windows).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.

- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales, es adaptable a las políticas y arquitecturas propias de cada empresa, ya que es lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer y documentar, además de que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros. (Potencier, 2006)

1.5 Servidor web

Un servidor web es un programa que implementa el protocolo HTTP y se encarga de mantenerse a la espera de peticiones llevadas a cabo por un cliente conocemos como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML, entre los que se encuentran: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos (Gerner, y otros., 2006).

Apache

El servidor web Apache es un servidor de código abierto, libre de pagos de licencias, posee una increíble velocidad de respuesta soportando mayores peticiones por segundo que su homólogo más cercano IIS, la elevada seguridad que le provee a sus clientes es una de sus grandes características y en la actualidad domina ampliamente el mercado de la Web con respecto a los demás servidores HTTP.

El servidor Apache se desarrolla dentro del proyecto HTTP Server de la Apache Software Foundation. Este presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en

su configuración. En la actualidad, Apache es el servidor HTTP es usado en el 70% de los sitios web en el mundo y creciendo aún su cuota de mercado. (Gerner, y otros., 2006)

XAMPP

XAMPP es un servidor independiente de plataforma que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. Oficialmente, los diseñadores de XAMPP sólo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores testear su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica, sin embargo, XAMPP es utilizado actualmente como servidor de sitios Web, ya que, con algunas modificaciones es lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes. (EcuRed, 2010)

WAMP

Entorno de desarrollo web, es el acrónimo usado para describir un sistema de infraestructura de Internet. El uso de WAMP permite servir páginas HTML a Internet, además de poder gestionar datos en ellas. Al mismo tiempo un WAMP proporciona lenguajes de programación para desarrollar aplicaciones web. Además, tiene funcionalidades que lo hacen muy completo y fácil de usar, entre las que se destacan:

- Gestionar sus servicios de Apache y MySQL.
- Cambiar de línea / fuera de línea (dar acceso a todos o sólo local).
- Instalar y cambiar de Apache, MySQL y PHP emisiones.
- Gestión de la configuración de sus servidores.
- Acceder a sus registros.
- Acceder a sus archivos de configuración. (EcuRed, 2010)

1.6 Gestor de bases de datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

PostGreSQL

PostGreSQL es un sistema de gestión de bases de datos que presenta un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.

Principales características:

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes, cadenas de bits, etc. También permite la creación de tipos propios.
- Incorpora una estructura de datos para arreglos.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios y los permisos asignados a cada uno de ellos. (Scribd, 2008)

MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Este gestor de bases de datos

es uno de los más usados en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Existen varias librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. Entre sus características se destacan:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API en gran cantidad de lenguajes (C, C++, Java, PHP).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo buen nivel de seguridad en los datos. (MySQL, 2010)

1.7 Metodologías, herramientas y lenguajes de modelado en el proceso de desarrollo de software

Según el libro titulado **El proceso unificado de desarrollo de software**: "El proceso de desarrollo de software es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo".

Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto de software que reúna los requisitos del cliente. No existe un proceso de software único que sea efectivo para todos los contextos. Debido a esta diversidad, es difícil automatizar todo un proceso de desarrollo de software por lo que se hace necesario contar con una metodología que garantice la satisfacción del cliente para que se puedan alcanzar los objetivos del proyecto. (Jacobson, y otros., 2000)

1.7.1 Metodología de desarrollo de software

Un proceso de software detallado y completo suele denominarse Metodología, define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Las metodologías se desarrollan con el objetivo de dar solución a los problemas existentes en la producción de software, que cada vez son más complejos. Estas engloban procedimientos, técnicas, documentación y herramientas que se utilizan en la creación de un producto de software. No existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable. (Jacobson, y otros., 2000)

Otras metodologías, denominadas Metodologías Ágiles, están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos de desarrollo pequeños, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso.

SXP

La metodología SCRUM Extreme Programming (SXP) es la unión de las metodologías SCRUM y Extreme Programming (XP) que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepamos por dónde andamos. Por su parte XP más bien es una metodología encaminada para el desarrollo, 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.

Consta de 4 fases principales:

- Planificación-Definición donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- Desarrollo, es donde se realiza la implementación del sistema hasta que esté listo para ser entregado.

- Entrega, puesta en marcha.
- Mantenimiento, donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto (LRP), definición de las Historias de Usuario (HU), diseño, implementación, pruebas, de donde se generan artefactos para documentar todo el proceso. Además existe una refactorización continua, lo que nos permite mejorar el diseño cada vez que se le añade una nueva funcionalidad. (EcuRed, 2011)

RUP

RUP es uno de los procesos más generales, estructurado y adaptable a las características y necesidades de cualquier proyecto. Propone 6 disciplinas ingenieriles (Modelamiento del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba, Despliegue) y 3 de apoyo (Gestión de Configuración y Cambios, Gestión de Proyecto, Ambiente) que tienen lugar en algunas o todas las fases que define (Inicio, Elaboración, Construcción y Transición). Este proceso unificado tiene tres características distintivas:

- Dirigido por Casos de Uso
- Centrado en la Arquitectura.
- Iterativo e Incremental. (EcuRed, 2011)

RUP permitirá trabajar con precisión y calidad, perfeccionado el software en cualquier momento del desarrollo, logrando así, un sistema con la robustez necesaria, independientemente del tiempo disponible. Si se maneja bien el conjunto de procesos, es posible realizar todos los artefactos que propone RUP independientemente del tamaño del equipo de desarrollo.

Con el uso de la metodología SXP, se logra un estrecho vínculo entre los colaboradores y los desarrolladores, permitiendo el intercambio de información que facilita el trabajo de ambas partes, sin embargo, RUP posee un alto grado de complejidad lo que resulta un poco complicado ya que requiere de un amplio conocimiento del proceso, siendo recomendable para proyectos de gran alcance. Por otra parte

la documentación que se obtiene con SXP es muy flexible y está sujeta a cambios repentinos que puedan ocurrir a lo largo del desarrollo del producto.

1.7.2 Lenguaje de modelado

El modelado de sistemas software es una técnica para tratar con la complejidad inherente a estos sistemas. El uso de modelos ayuda al ingeniero de software a "visualizar" el sistema a construir. Además, los modelos de un nivel de abstracción mayor pueden utilizarse para la comunicación con el cliente.

UML

UML es un lenguaje de modelado visual que se utiliza para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Con este lenguaje es posible diseñar, hojear, configurar, mantener, y controlar la información de los sistemas de software.

UML incluye conceptos semánticos, notación y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como, generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. (Jacobson, y otros., 2000)

1.7.3 Herramientas CASE para el modelado

La mayoría de las herramientas CASE (del inglés Computer Aided Software Engineering) hacen uso de UML. Estas herramientas se consideran aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software de forma tal que se reduzca el costo de las mismas en términos de tiempo y dinero. Entre las más utilizadas a nivel mundial se encuentran Visual Paradigm y Rational Rose Enterprise Edition.

Rational Rose Enterprise Edition

Rational Rose Enterprise es el producto más completo de la familia Rational Rose, es una herramienta CASE propietaria, desarrollada por Rational Corporation, basada en el Lenguaje Unificado de Modelado.

Permite crear los diagramas que se van generando durante el proceso de desarrollo del software y posee un gran número de estereotipos predefinidos que facilitan el proceso de modelación. Rational Rose Enterprise Edition posee características adicionales como:

- Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5.
- Soporte Enterprise Java Beans 2.0.
- Capacidad de análisis de calidad de código.
- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- Integración con otras herramientas de desarrollo de Rational.
- Capacidad para integrarse con cualquier sistema de control de versiones. (GSInnova, 2010)

Visual Paradigm

Herramienta CASE propietaria con licencia gratuita, que soporta el ciclo de vida completo del desarrollo de software. Visual Paradigm ofrece:

- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.

- Disponibilidad de integrarse en los principales IDEs.
- Disponibilidad en múltiples plataformas. (Sierra, 2008)

Aunque Rational Rose Enterprise Edition es una de las herramientas más potente para el modelado visual y una de las más utilizadas en el área de la informática, es un software propietario y no es multiplataforma, inconvenientes que limitan su uso. Considerando estos aspectos, se selecciona Visual Paradigm como herramienta para el modelado de la solución que se propone, siendo esta una herramienta multiplataforma que cuenta además, con versiones gratuitas y provee fácil integración con el resto de las herramientas de desarrollo.

Consideraciones parciales

El estudio realizado sobre los sistemas de gestión de la información de los servicios en hospitales, confirmó que no existe ninguno que satisfaga todas las necesidades del hospital. Debido a que se necesita gestionar la información de los servicios que se brindan en el centro de salud de la universidad, se propone el desarrollo del sistema “SIGESH” que permitirá gestionar la información que se maneja en el centro hospitalario, además de centralizar dicha información en un único servidor.

El perfil tecnológico para la realización de esta propuesta estará compuesto por Symfony, Ext JS como frameworks y NetBeans como de entorno de desarrollo. Para la lógica del cliente y el control de los procedimientos se utilizará JavaScript, así como HTML y CSS para definir la interfaz de usuario. Como gestor de base de datos se utilizará PostgreSQL y como servidor de base de datos Apache.

Como metodología de desarrollo se utilizará SXP, ya que se adecua a las características del proyecto. La herramienta CASE será Visual Paradigm por ser apropiada y potente para el desarrollo de la solución propuesta.

CAPÍTULO 2 DEFINICIÓN DEL SIGESH

Introducción

En el presente capítulo se describen las principales funciones y los requisitos no funcionales identificados durante la investigación. Se muestran las historias de usuarios que fueron descritas por el cliente y son definidas las iteraciones por los programadores. Además, se muestran los procesos que serán informatizados, realizando así la primera propuesta de solución del sistema.

2.1 Objetos a informatizar

Durante el proceso de desarrollo del software existen varias acciones que deben ser informatizadas por lo engorrosa que se hace la gestión de la información en el centro de salud. Con la implementación de este sistema se informatizarán las actividades de gestión de la información de los medicamentos en el almacén, el departamento de economía, estadística, vacunaciones y admisión y turnos médicos, integrando en un servidor central toda esta información de manera que este centralizada. Con el objetivo de tener un mayor control de toda la información de los servicios que se brindan en el hospital y mejorar la calidad de los mismos.

2.2 Propuesta de solución del sistema

Con la realización del trabajo se dispone a desarrollar un sistema que permita la gestión (insertar, modificar, eliminar y consultar) la información referente a los servicios que se brindan en el hospital Ernesto Ché Guevara. Además, permitirá centralizar y gestionar de manera más eficaz los servicios que se brindan el centro de salud de la universidad, permitiendo generar reportes de los resultados almacenados. Se habilitará una cuenta de administrador para limitar el acceso a las funcionalidades que brinda el sistema a quienes no lo requieran, todos los usuarios que requieran acceder al sistema, deben ser autenticados contra un repositorio o base de datos única de identidades (rol de usuario), acción que permitirá habilitar los permisos que tendrán los mismos al ingresar al sistema.

2.3 Actores del sistema

Actores

Un actor es alguien o algo, externo al sistema que de cierta forma interactúa con este. Los actores no son solamente roles que juegan personas, sino también organizaciones, software y máquinas. (Portillo, 2008)

A continuación se reflejan los actores que componen el sistema en desarrollo:

Administrador: Posee los permisos necesarios para gestionar los usuarios y privilegios asociados a estos.

Almacenero: Posee los permisos necesarios para gestionar los medicamentos en la Tarjeta de Estiba y exportar reportes de los mismos.

Asistente de estadística: Posee los permisos necesarios para consultar el Registro de Vacunación así como exportar reportes del Registro de Vacunación y la Tarjeta de Vacunación de un paciente determinado.

Económica: Posee los permisos necesarios para consultar los medicamentos en la Tarjeta de Sub-mayor de Inventario y exportar reportes de los mismos.

Enfermera: Posee los permisos necesarios para gestionar los Reportes de Vacunación así como exportar el Reporte de Vacunación y la Tarjeta de Vacunación de un paciente determinado.

Recepcionista: Posee los permisos necesarios para gestionar los Turnos Médicos y exportar reportes de los mismos.

2.4 Funcionalidades del sistema

Un requerimiento es una descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso. (Arias, 2006)

2.4.1 Lista de reserva del producto

Después de haber descrito el proceso de gestión hospitalaria, se definen y priorizan las funcionalidades que va a tener el sistema y se describen los requisitos no funcionales que debe tener el software. Esta

lista puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente. El objetivo es asegurar que el producto definido al terminar la misma es el más correcto, útil y competitivo posible.

A continuación se muestran las principales funcionalidades que debe tener el Sistema de Gestión de Hospitales, agrupándose según su prioridad:

Prioridad muy alta

1. Gestionar medicamento en Tarjeta de Estiba.
 - 1.1. Registrar medicamento en Tarjeta de Estiba.
 - 1.2. Modificar medicamento en Tarjeta de Estiba.
 - 1.3. Eliminar medicamento de Tarjeta de Estiba.
 - 1.4. Listar medicamento de Tarjeta de Estiba.

2. Gestionar Reporte de Vacunación.
 - 2.1. Crear Reporte de Vacunación.
 - 2.2. Modificar Reporte de Vacunación.
 - 2.3. Eliminar Paciente del Reporte de Vacunación.
 - 2.4. Listar Pacientes del Reporte de Vacunación.

3. Gestionar turno médico.
 - 3.1. Crear turno médico.
 - 3.2. Modificar turno médico.
 - 3.3. Eliminar turno médico.
 - 3.4. Listar Pacientes con turno médico.

4. Gestionar paciente.
 - 4.1. Registrar datos del paciente.
 - 4.2. Modificar datos del paciente.
 - 4.3. Eliminar paciente.
 - 4.4. Listar pacientes.

5. Gestionar vacuna.
 - 5.1. Registrar datos de la vacuna.
 - 5.2. Modificar datos de la vacuna.
 - 5.3. Eliminar vacuna.
 - 5.4. Listar vacunas.
6. Consultar Registro de Vacunación.
7. Consultar medicamentos existentes en el departamento de Economía.

Prioridad alta

8. Exportar Tarjeta de Estiba.
9. Exportar Reporte de Vacunación.
10. Exportar Tarjeta de Sub-mayor de Inventario.
11. Exportar Registro de Vacunación.
12. Exportar listado de interconsultas.

Prioridad media

13. Gestionar usuario.
 - 13.1. Crear usuario.
 - 13.2. Actualizar usuario.
 - 13.3. Eliminar usuario.
 - 13.4. Listar usuarios.
14. Asignar permiso al usuario.
15. Denegar permiso al usuario.
16. Autenticar usuario.

17. Cambiar contraseña.

2.5 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. (Arias, 2006)

Los requisitos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, mediante esta valoración se puede marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

RnF 1. Accesibilidad: El sistema debe garantizar que el acceso a la información se realice de acuerdo al rol que desempeñan los usuarios.

RnF 2. Disponibilidad: El sistema debe estar disponible las 24 horas del día.

RnF 3. Portabilidad: El sistema debe ser multiplataforma.

RnF 4. Requisitos de apariencia o interfaz externa: La interfaz del sistema debe ser sencilla e intuitiva, para que los usuarios puedan utilizar la aplicación con facilidad.

RnF 5. Requisito de software:

Cliente web:

- Sistema Operativo: Windows (Windows 95 o superior).
- Linux (cualquiera de sus distribuciones).
- Navegador Web compatible con HTML 2.0 y CSS.
- Internet Explorer 4.2 (o superior) y compatibles.

Servidor web: Servidor Web Apache 2.2.X o superior.

Servidor de bases de datos: PostgreSQL.

RnF 6. Requisitos de hardware:

Cliente web:

- Procesador Pentium 3 (o superior).
- Memoria RAM mínima de 256 MB.

Servidor web:

- Procesador Pentium 4 (o superior).
- Memoria RAM \geq 1 GB.

Servidor de bases de datos:

- Procesador Pentium 4 (o superior).
- Memoria RAM \geq 1 GB.

2.6 Descripción de las historias de usuario

Con el uso de la metodología SXP la gestión de requerimientos del sistema es extremadamente simple. El cliente describe y prioriza sus necesidades mediante historias de usuario (HU), las cuales constituyen descripciones cortas y escritas sin terminología técnica. Los programadores estiman el esfuerzo asociado y las dependencias entre ellas. Para planificar el trabajo desde el punto de vista técnico, las HU son divididas en tareas para las cuales también se realiza una estimación. (Escribano, 2002)

A continuación se describe la HU Gestionar medicamento en la Tarjeta de Estiba, ver tabla 1. Las descripciones correspondientes al resto de las HU se muestran en el Anexo 2. Descripción de las historias de usuario, ver tablas 7-22.

Tabla 1 HU1.Gestionar medicamento en la Tarjeta de Estiba

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Gestionar

	medicamento en la Tarjeta de Estiba
Modificación de Historia de Usuario Número: Ninguna	
Programador: Isnaudy Pérez Téllez	Iteración Asignada: Primera
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2 semanas
Riesgo en Desarrollo: Muy Alta	Puntos Reales: 1 semana
<p>Descripción: El sistema debe permitir registrar, modificar, eliminar y listar los medicamentos existentes en la Tarjeta de Estiba. Para registrar un medicamento se muestra su respectivo formulario con los campos donde se va a registrar el medicamento: código, unidad de medida, precio, clave, máximo, subgrupo, nombre, proveedor, grupo farmacológico, forma farmacéutica, forma de presentación, fabricante. Si se va a modificar un medicamento se debe seleccionar del listado de medicamentos existentes el que se desea modificar, se muestra un formulario con los campos que se pueden modificar: código, unidad de medida, precio, clave, máximo, subgrupo, nombre, proveedor, grupo farmacológico, forma farmacéutica, forma de presentación, fabricante. En caso de eliminar un medicamento se debe seleccionar del listado de medicamentos existentes el que se desea eliminar. Para listar los medicamentos de debe especificar un criterio de búsqueda (nombre, fecha).</p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> • El usuario debe ser trabajador del hospital, además debe estar autenticado en el sistema con el rol de Almacenero. • El Almacenero es el encargado de la gestión de los medicamentos en la Tarjeta de Estiba. 	
Prototipo de interfaz:	

Código	Subgrupo	Nombre	Clave	Precio	Mínimo	Máximo	Fondo ...	Existen...	Reten...	
<div style="display: flex; justify-content: space-between; align-items: center;"> + Adicionar ✎ Modificar ✖ Eliminar <input type="text" value="Escriba un nombre..."/> filtro por subgrupo filtro por farmacología </div>										
Subgrupo: Tabletas que se toman (5 Productos)										
1	38010110282883	Tabletas que se tom...	Ácido bórico x 500G	ACI-234	\$12	32	40	10	3	0
2	77849766303939	Tabletas que se tom...	Amoxicilina - 400	AM-12R	\$2.7	720	900	225	40	0
3	7643124567765432	Tabletas que se tom...	Mentol	234	\$4	453	567	141	323	0
4	2344564535456456	Tabletas que se tom...	Oxacilina-250	123-3A	\$2.5	400	500	150	230	100
5	9283740293840	Tabletas que se tom...	Aspirina 400	TRASC	\$5.5	960	1200	300	400	0

2.7 Plan de iteraciones

El Plan de iteraciones tiene como entrada el conjunto de historias de usuario definidas previamente. Para colocar una historia en cada iteración es necesario tener en cuenta la prioridad definida por el cliente en cada caso. La duración de la historia de usuario es el valor real que se le asignó en la misma, esta duración se expresa en semanas. Como resultado de la priorización de las HU se estima la siguiente planificación: (Escribano, 2002)

Tabla 2 Plan de Iteraciones de las HU

Iteraciones	Descripción de la iteración	Orden de la HU a implementar	Duración de cada HU (semanas)	Duración total (semanas)
Primera	En esta iteración se implementan las HU con prioridad muy alta.	HU1. Gestionar medicamento en Tarjeta de Estiba.	3	15
		HU2. Gestionar Reporte de Vacunación.	2	
		HU3. Gestionar turno médico.	3	

		HU4. Gestionar paciente.	3	
		HU5. Gestionar vacuna.	2	
		HU6. Consultar Registro de Vacunación.	1	
		HU7. Consultar medicamentos existentes en el departamento de economía.	1	
Segunda	En esta iteración se implementan las HU que tienen prioridad alta	HU8. Exportar Tarjeta de Estiba.	1	5
		HU9. Exportar Reporte de Vacunación.	1	
		HU10. Exportar Tarjeta de Sub-mayor de Inventario.	1	
		HU11. Exportar Registro de Vacunación.	1	
		HU12. Exportar listado de interconsultas.	1	
Tercera	En esta iteración se implementan las	HU13. Gestionar usuario.	1	5

HU que tienen prioridad media.	HU14. Asignar permisos al usuario.	1	
	HU15. Denegar permisos al usuario.	1	
	HU16. Autenticar usuario.	1	
	HU17. Cambiar Contraseña.	1	
Total			25 semanas

Consideraciones parciales

Las características del sistema fueron descritas a través de los artefactos correspondientes.

Se identificaron los procesos a informatizar dentro del sistema, los actores, las funcionalidades y los requisitos no funcionales.

Se realizó una planificación y estimación del esfuerzo de las historias de usuario a implementar en cada iteración y la descripción de cada una de ellas.

A partir de los artefactos descritos se puede comenzar la construcción del sistema, que incluye la arquitectura, la implementación y las pruebas de las historias de usuario descritas.

CAPÍTULO 3 DESARROLLO Y PRUEBAS DEL SIGESH.

Introducción

En el presente capítulo se realiza el diseño del sistema a través de la realización de tarjetas CRC. Además se describe la arquitectura utilizada, así como los patrones de diseño para el desarrollo del SIGESH. También se describe la estructura modular del sistema propuesto. Se presenta el modelo físico de la base de datos, el estándar de codificación utilizado, el diagrama de despliegue y las pruebas que se le aplican al sistema.

3.1 Patrones de diseño

Patrones de diseño del inglés Design Patterns son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, están basadas en la experiencia y se ha demostrado que funcionan. (Gracia, 2005)

GRASP

En el diseño del software orientado a objeto el uso de los patrones GRASP acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades) tiene una gran importancia, debido a que solucionan a muchos de los problemas que se pueden presentar a la hora de programar. (Larman, 2003)

El patrón Bajo acoplamiento establece la poca dependencia entre los componentes, ya que mientras menos acoplamiento, más reusable y flexible se vuelve el sistema. (Larman, 2003)

El uso de este patrón en el SIGESH se evidencia en los módulos Vacunas, Medicamentos, Estadísticas y Admisión, ya existe la menor dependencia posible entre ellos.

El patrón Alta Cohesión el cual plantea que la información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase. (Larman, 2003)

En el SIGESH el patrón Alta Cohesión se evidencia en la clase Vacuna, Paciente, Consulta y Producto, ya que la información contenida en las mismas se corresponde con el nombre de la clase: Ejemplo en la clase Producto se controla la información de los productos existentes en el almacén de medicamentos.

Experto: es un patrón de asignación de responsabilidades que delega las responsabilidades a quién contiene la información necesaria para cumplirlas. (Larman, 2003)

Tal es el caso de las clases que genera la librería Propel con todas las operaciones o métodos útiles que consolidan un desarrollo ágil y una abstracción del sistema gestor de bases de datos empleado.

GOF

Fachada: el mismo provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema. (Gamma, y otros., 1994)

El uso de este patrón se evidencia en el SIGESH, ya que desde la interfaz del SIGESH se accede a las diferentes vistas de los subsistemas que lo componen y a partir de estos subsistemas se accede a un grupo de módulos que forman parte del mismo.

Decorator (Envoltorio): Normalmente en las aplicaciones web existen contenidos que son comunes a todas las páginas que conforman la aplicación. (Gamma, y otros., 1994)

SIGESH, haciendo uso de este patrón de diseño estructural delimita el código común en todas las páginas (definido en un archivo global para todas las vistas denominado layout) del código HTML generado como respuesta a una petición determinada (denominado plantilla). El contenido de la plantilla se integra en el layout, por lo que se puede afirmar que el layout decora la plantilla.

Singleton (Única instancia): Es un patrón creacional que garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. (Gamma, y otros., 1994)

A través de la clase sfContext::getInstance() en el SIGESH, se obtiene una referencia a todos los objetos del núcleo del framework, por lo que es posible por ejemplo acceder desde las clases de formularios a métodos definidos en la clase que maneja las peticiones.

3.2 Diseño del sistema

Para el diseño de aplicaciones, la metodología SXP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar se usan otras técnicas como las tarjetas CRC.

3.2.1 Tarjetas CRC

Estas tarjetas son una técnica de modelado orientado a objeto que permite identificar clases y sus responsabilidades, Se hacen con el objetivo de identificar jerarquías de generalización/especificación o jerarquías de agregación entre las clases, de manera que ayuda al refinamiento de clases. Se definen las tarjetas CRC con la finalidad de obtener un diseño simple y no incurrir en la implementación de características que no son necesarias. (Escribano, 2002)

A continuación se describen las tarjetas CRC, vacuna, paciente, consulta, producto y vacunación, ver figuras 1-5. Las descripciones correspondientes al resto de las tarjetas CRC se muestran en el Anexo 4. Tarjetas CRC, ver figuras 12-19.

CRC paciente	
Descripción: Contienen información del estado del Paciente	
Atributos:	
Nombre	Descripción
id	identificador
nombre	
tipoDePaciente	
apellidos	
facultad	
fecha	
hora	
grupo	
Responsabilidades:	
Nombre	Collaborator
addPaciente	tiposDePaciente
getPacientes	
setPaciente	
delPaciente	

Figura 1 Tarjeta CRC. Paciente

CRC vacuna	
Descripción: Contienen información del estado del Vacuna	
Atributos:	
Nombre	Descripción
id	identificador
nombre	
tipoDeVacuna	
farmacologia	
contraindicaciones	
precauciones	
reaccionesAdversas	
psicologia	
composicion	
advertencias	
Responsabilidades:	
Nombre	Collaborator
addVacuna	tipoDeVacuna
getVacuna	
delVacuna	
setVacuna	

Figura 2 Tarjeta CRC. Vacuna

CRC consulta	
Descripción: Contienen información del estado de la Consulta	
Atributos:	
Nombre	Descripción
id	identificador
paciente	
especialidad	
hora	
asistencia	
reconsulta	
fecha	
Responsabilidades:	
Nombre	Collaborator
addConsulta	especialidad
getConsultas	paciente
setConsulta	
delConsulta	
exportarReporteConsulta	

Figura 3 Tarjeta CRC. Consulta

CRC producto	
Descripción: Contienen información del estado del Producto	
Atributos:	
Nombre	Descripción
id	identificador
codigo	
precio	
nombre	
maximo	
minimo	
fondoFijo	
existencia	
retenidos	
subgrupo	
grupoFarmacologico	
formaFarmaceutica	
unidadDeMedida	
proveedor	
fabricante	
precetacion	
Responsabilidades:	
Nombre	Collaborator
addProducto	subgrupo
getProductos	grupofarmacologico
delProducto	formaFarmaceutica
setProducto	unidadDeMedida
exportarReporteProducto	proveedor
exportarReporteTarjeta	fabricante
	formaDePresentacion
	productoMovimiento

Figura 4 Tarjeta CRC. Producto

CRC vacunacion	
Descripción: Contienen información del estado de la Vacunación	
Attributes:	
Nombre	Descripción
id	identificador
vacuna_fk	llave foránea
grupo_fk	llave foránea
paciente_fk	llave foránea
primera_dosis	
segunda_dosis	
tercera_dosis	
reactivacion	
Responsabilidades:	
Nombre	Collaborator
addVacunacion	grupoDeRiesgo
getVacunaciones	paciente
setVacunacion	vacuna
delVacunacion	
getGruposDeRiesgo	
exportarReporteVacunacion	

Figura 5 Tarjeta CRC. Vacunación

3.3 Arquitectura de la solución propuesta

La IEEE define la arquitectura de software como: organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. (Reynoso, y otros., 2004)

3.3.1 Patrón de arquitectura

La representación y diseño explícitos de la arquitectura de software, se han convertido en temas dominantes de la ingeniería de software. En su libro dedicado a la arquitectura de software, Bass y sus colegas la identifican como un elemento de gran importancia durante la construcción del software, de la cual depende en gran medida el éxito del desarrollo. (Reynoso, y otros., 2004)

Para la arquitectura del SIGESH se utilizó el **Modelo – Vista – Controlador (MVC)** pues symfony está basado en este patrón. El MVC es un patrón de arquitectura de software que separa los datos de una

aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. A continuación se darán a conocer algunas características del MVC:

- Separa la lógica de la aplicación y la presentación en varias capas.
- Facilita la labor de todo el equipo: diseñadores gráficos, programadores, diseñadores de base de datos, entre otros roles.
- Existen marcos de trabajo ya programados que facilitarán el trabajo de los miembros del equipo de trabajo.
- El Modelo, la Vista y el Controlador son tratados como entidades separadas, habiendo un nivel de abstracción y separación bien definido entre ellos, pero con una vía de comunicación flexible.
- Los cambios realizados en el Modelo se reflejan de forma casi automática en cada una de las Vistas, lo cual es un mecanismo ideal para producir aplicaciones que necesitan tener el estado de los datos en Tiempo Real. (Bahit, 2010)

Modelo – Vista – Controlador (MVC)



Figura 6 Funcionamiento del patrón modelo-vista-controlador

El Modelo

Es el componente que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas y notificar a las Vistas cuando cambia el Modelo. (Potencier, y otros., 2008)

En el SIGESH la capa del modelo está dividida en dos capas, una para la abstracción del gestor de bases de datos y la otra es la capa de acceso a los datos, con lo que logra aplicaciones independientes a los sistemas gestores de base de datos.

La Vista

Es el componente que maneja la presentación visual de los datos contenidos por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al mismo. (Potencier, y otros., 2008)

En el SIGESH la capa de la vista, está separada en tres partes: los elementos comunes a todas las páginas de la aplicación denominada “layout”, los “grid” encargados de visualizar las variables definidas en el controlador y la lógica de las vistas definida a través de los “form” (fragmentos de vistas que contienen lógica de aplicación y pueden ser reutilizables).

El Controlador

Es el componente que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al mismo. (Potencier, y otros., 2008)

La capa referente al controlador dentro del SIGESH, se divide en un controlador frontal y acciones, ubicando el controlador frontal en el directorio web del proyecto.

3.3.2 Estructura del SIGESH

Un proyecto desarrollado con symfony se compone de una o más aplicaciones y estas a su vez están organizadas en módulos. Los módulos encapsulan código estrechamente relacionado y representan operaciones que se realizan en los mismos. Symfony implementa una estructura jerárquica de directorios que permite realizar configuraciones a distintos niveles: proyecto, aplicación o módulo; con el que se logra un nivel de configuración muy preciso.

Para darle cumplimiento al problema planteado se creó un proyecto en symfony que consta de una aplicación llamada SIGESH:



Figura 7 Estructura del SIGESH

3.3.3 Descripción de los módulos

El SIGESH es una aplicación con la que interactúan los usuarios del sistema, dedicada a la gestión de la información de los medicamentos, vacunas y turnos médicos del hospital Ernesto Ché Guevara. Está compuesto por un total de 5 módulos (Admisión y Turnos, Informe, Medicamentos, Configuraciones y Vacunaciones), a los cuales se les establecen los privilegios de acceso que los usuarios deberán tener para operar sobre los mismos.

Los módulos son los pequeños subsistemas que vemos en el árbol de funcionalidades del proyecto (ver figura 8). Cada módulo es una pequeña aplicación independiente, con su propia configuración, vistas y sistemas de seguridad.

```
apps/  
  [nombre aplicacion]/  
    modules/  
      [nombre modulo]/  
        actions/  
          actions.class.php  
        config/  
        lib/  
        templates/  
          indexSuccess.php
```

Figura 8 Estructura de directorios de un módulo

Módulos:

Admisión y Turnos: Este módulo gestiona los pacientes y los turnos médicos a las que ellos asistirán.

Funcionalidades del módulo:

- Registrar datos del paciente.
- Modificar datos del paciente.
- Eliminar paciente.
- Listar pacientes.
- Crear turno médico.
- Modificar turno médico.
- Eliminar turno médico.
- Listar Pacientes con turno médico.
- Exportar listado de interconsultas.

Informe: Este módulo permite consultar y exportar los reportes enviados desde el almacén y el centro de vacunación.

Funcionalidades del módulo:

- Consultar Registro de Vacunación.
- Consultar medicamentos existentes en el departamento de economía.
- Exportar Tarjeta de Sub-mayor de Inventario.
- Exportar Reporte de Vacunación.
- Exportar Registro de Vacunación.

Medicamentos: Este módulo gestiona las tarjetas de estiba de los medicamentos.

Funcionalidades del módulo:

- Registrar medicamento en Tarjeta de Estiba.
- Modificar medicamento en Tarjeta de Estiba.
- Eliminar medicamento de Tarjeta de Estiba.
- Listar medicamento de Tarjeta de Estiba.
- Exportar Tarjeta de Estiba.

Configuraciones: Este módulo gestiona todo lo relacionado con los roles, las cuentas de usuario y sus permisos, además presenta un registro de todas las acciones realizadas en el sistema.

Funcionalidades del módulo:

- Crear usuario.
- Actualizar usuario.
- Eliminar usuario.
- Listar usuarios.
- Asignar permiso al usuario.
- Denegar permiso al usuario.
- Autenticar usuario.
- Cambiar contraseña.

Vacunaciones: Este módulo adiciona los pacientes y gestiona las vacunas recibidas por estos, además de sus tarjetas de vacunación.

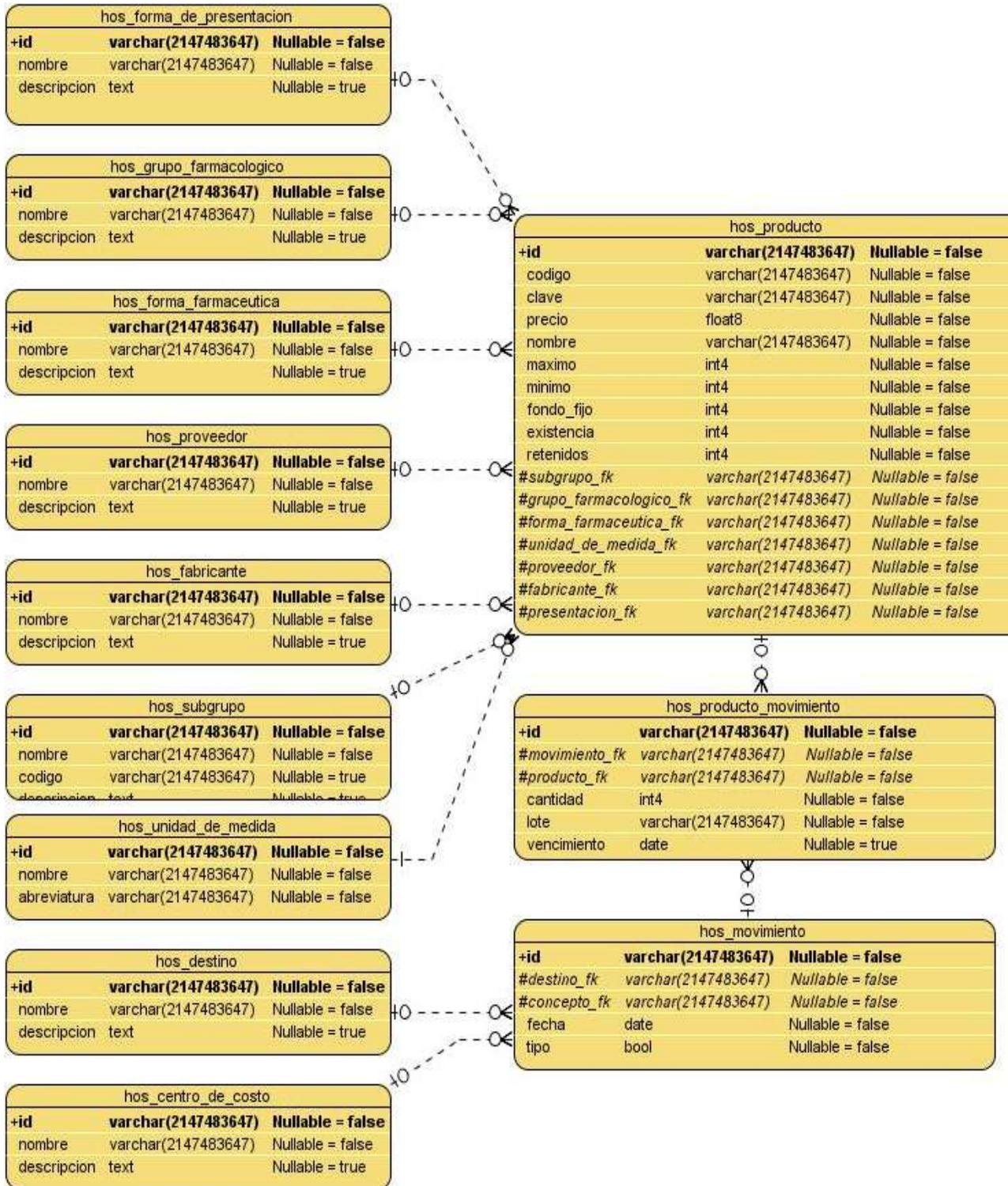
Funcionalidades del módulo:

- Registrar datos del paciente.
- Registrar datos de la vacuna.
- Modificar datos de la vacuna.
- Eliminar vacuna.
- Listar vacunas.
- Crear Reporte de Vacunación.
- Modificar Reporte de Vacunación.
- Eliminar Paciente del Reporte de Vacunación.
- Listar Pacientes del Reporte de Vacunación.

3.4 Diseño del esquema de base de datos del sistema

La base de datos del SIGESH se encuentra dividida por esquemas para lograr una mayor integridad y seguridad de los datos. En este epígrafe se muestra el diseño de los esquemas de base de datos Hospital y Seguridad propuesto a través del modelo físico de datos.

El esquema Hospital es el encargado de almacenar los datos relacionados con los procesos del negocio. Por su parte, el esquema Seguridad es el encargado de almacenar los datos relacionados con la seguridad del sistema, ya sea autenticación o permisos.



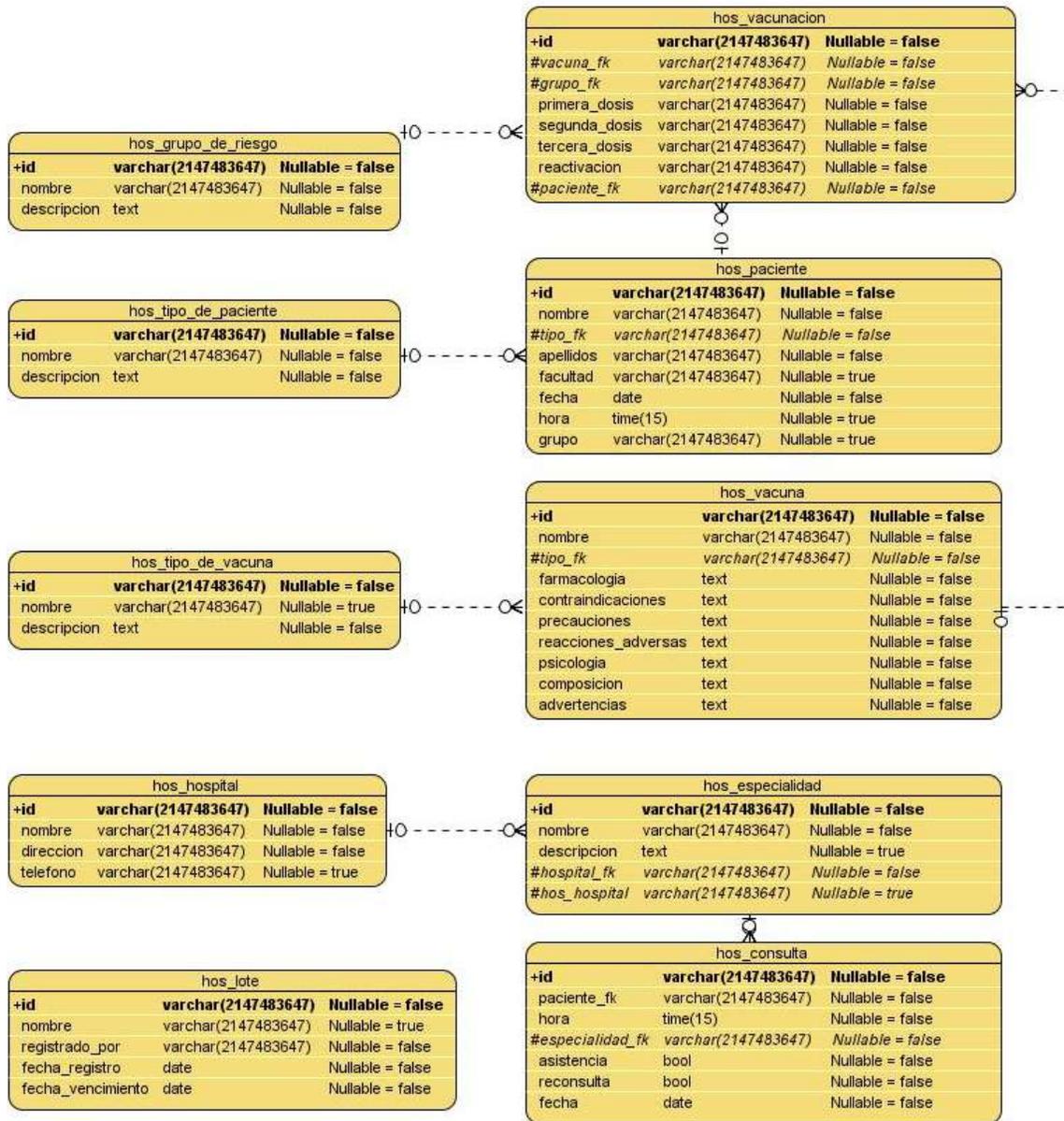


Figura 9 Modelo físico de datos del esquema hospital

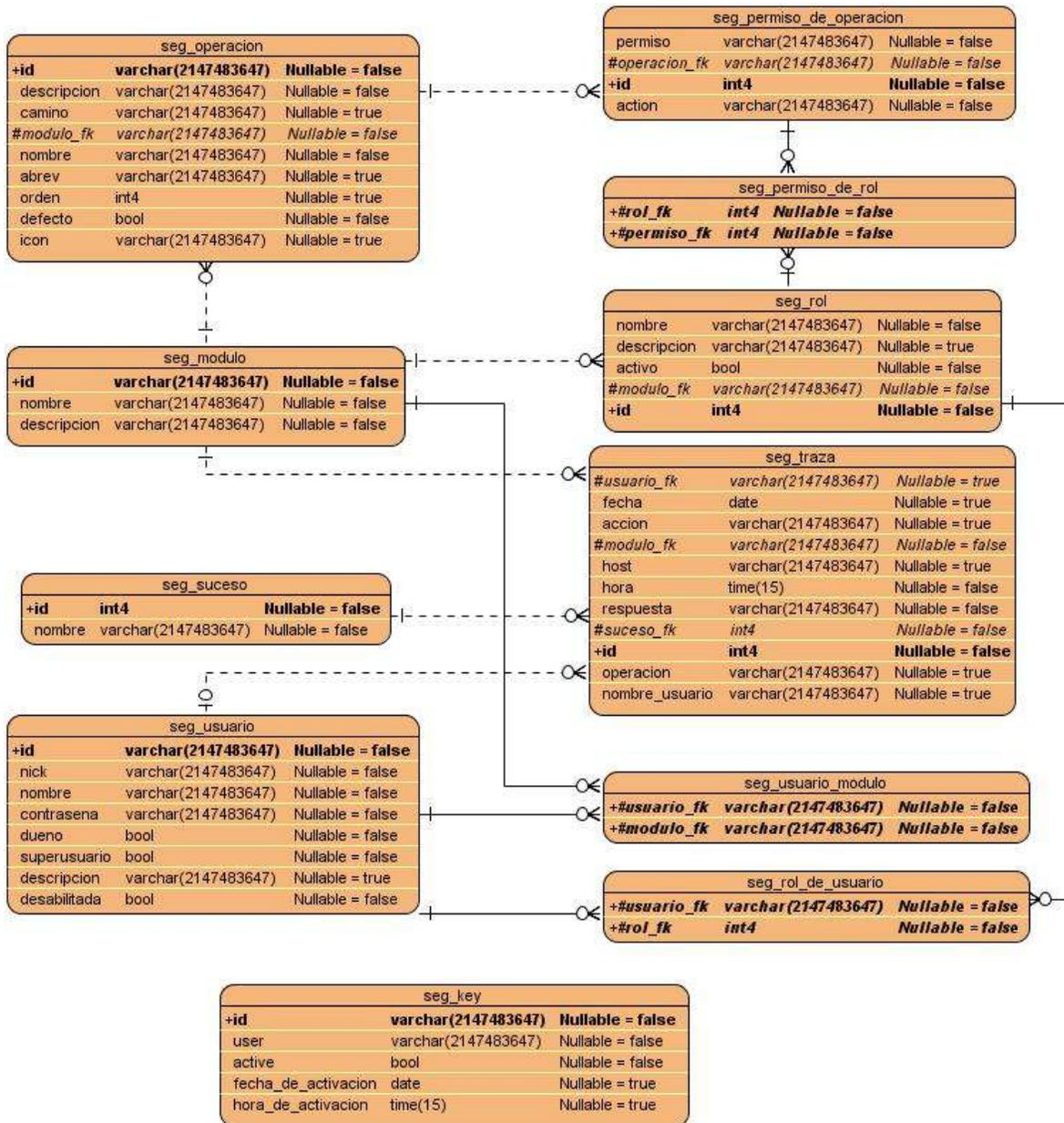


Figura 10 Modelo físico de datos del esquema seguridad

3.5 Estándar de codificación utilizada

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vista a generar un código de alta eficacia es de gran importancia para la calidad del software y para obtener un buen

rendimiento. Además, si se aplica de forma continua un estándar de codificación bien definido es posible que un proyecto de software se convierta en un sistema fácil de comprender y de mantener.

1. La longitud utilizada para una línea de código es de 80 caracteres. Esto significa que se debe mantener cada línea de su código por debajo de los 80 caracteres, siempre que sea posible.
2. No ponga espacios después de abrir un paréntesis ni antes de cerrarlo:

```
if ($Hospital->save()) // correcto
if ( $Hospital->save() ) // incorrecto
```

3. Use la notación camelCase, no guión bajo, para variables, funciones y nombre de métodos, además para la programación orientada a objetos, los métodos de acceso para las instancias o variables estáticas deben ir antepuestos con "get", "set", "add" o "del".

```
setNombre// correcto
$FechaConsulta // incorrecto
$fecha_Consulta // incorrecto
```

4. La llave "{" se escribe en la misma línea que la sentencia condicional. La llave de cierre "}" se escribe siempre en su propia línea. Cualquier contenido dentro de las llaves debe separarse con cuatro espacios en blanco.

```
if($pers){
    $usuario->setNombre($pers->getNombreCompleto()); // correcto
}
if($pers)
{
    $usuario->setNombre($pers->getNombreCompleto()); // incorrecto
}
```

5. Symfony puede declarar su visibilidad explícitamente usando las palabras reservadas `private`, `protected` o `public`.

```
public function execute()
```

6. La declaración de las clases según symfony es `class` nombreDeLaClase seguido de la función Action `extends` claseHereda, nombreDeLaClase puede ser escrito en español, utilizando la notación camelCase.
7. No termine el código de los archivos de librerías con la etiqueta de cierre `?>`. Esto es porque realmente no es necesario y puede traer problemas en las salidas si en algún momento tiene un espacio en blanco al final de dicha etiqueta.

```
<?php  
  
class sesionCaducaAction extends sfAction{ // correcto  
}  
  
<?php  
class sesionCaducaAction extends sfAction{ // incorrecto  
}  
  
?>
```

8. Use minúsculas para constantes tipadas nativas de PHP: `false`, `true` y `null`. Lo mismo para `array()`.
9. Los nombres de las tablas de la base de datos serán lo más descriptivo posible, las del esquema hospital iniciarán con el prefijo `hos`, mientras que las del esquema seguridad iniciarán con el prefijo `seg` y para el caso de nombres compuestos se usará guión bajo:

```
hos_hospital // correcto  
seg_usuario // correcto  
hospital // incorrecto  
segpermisoderol // incorrecto
```

3.6 Distribución física del SIGESH

Diagrama de despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. (Sparx, 2007)

El diagrama de despliegue que aparece a continuación muestra la disposición física de los distintos nodos que componen el sistema en tiempo de ejecución, los vínculos de comunicación y las instancias de los componentes y objetos que residen en los mismos. Las conexiones establecidas en este diagrama son asociaciones de comunicación entre los nodos y se etiquetan con un estereotipo que identifica el protocolo de comunicación o la red utilizada.

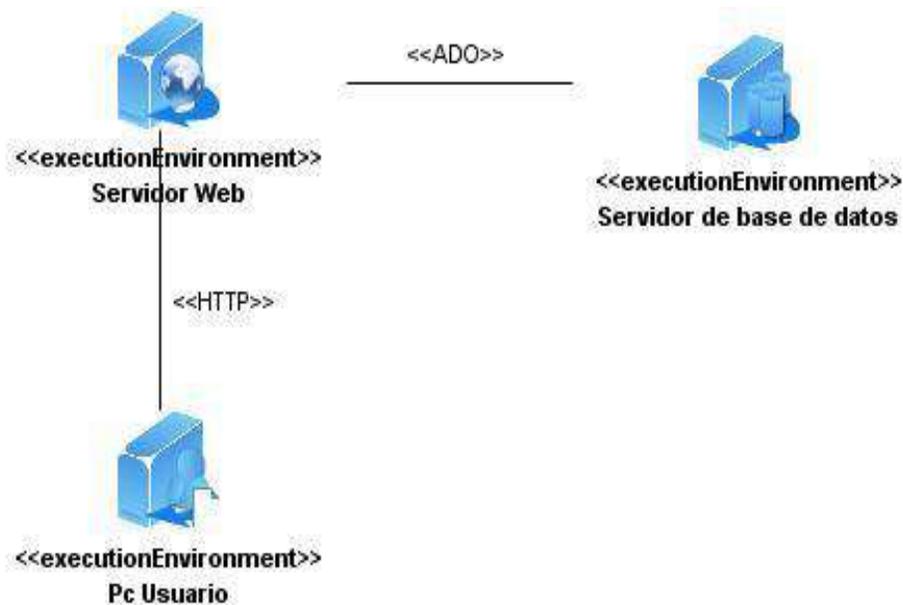


Figura 11 Diagrama de despliegue del sistema SIGESH

3.7 Validación de las funcionalidades

Las pruebas de software son elementos críticos para la garantía de la calidad del software y representan una revisión final de las especificaciones del diseño y la codificación.

Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra definidas por el cliente para cada historia de usuario y tienen como objetivo asegurar que las funcionalidades del sistema cumplan con lo que se

espera de ellas. En efecto, las pruebas de aceptación corresponden a una especie de documento de requerimientos en SXP, ya que marcan el camino a seguir en cada iteración, indicándole al equipo de desarrollo hacia donde tiene que ir y en qué puntos o funcionalidades debe poner el mayor esfuerzo y atención. (Jeffries, 1999).

Método de caja negra

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Los casos de prueba (CP) pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. (EcuRed, 2009)

De acuerdo a la planificación de las iteraciones realizadas (ver tabla 2), en la primera iteración se prueban funcionalidades correspondientes a las historias de usuario que tienen una prioridad Muy Alta para el negocio, en la segunda las de Alta y en la tercera las de Media. Además se realiza una cuarta iteración para comprobar que el sistema funciona correctamente.

A continuación se diseñan y aplican los casos de prueba para la HU1, ver tablas 3-5. Los casos de prueba correspondientes al resto de las HU se muestran en el Anexo 3. Descripción de los casos de prueba, ver tablas 23-43.

Casos de prueba. HU1 Gestionar Medicamento en Tarjeta de Estiba

Esta sección cubre el conjunto de pruebas funcionales relacionadas con la historia de usuario: **Gestionar medicamento en Tarjeta de Estiba.**

Descripción de la funcionalidad: En este CP se comprueba la entrada de un medicamento en la base de datos. Si los datos del medicamento son correctos se debe insertar en la base de datos y mostrar un mensaje indicando que el medicamento fue insertado satisfactoriamente. Si existe error se debe mostrar un mensaje indicando el mismo. Comprobar la modificación de un medicamento seleccionado del listado de medicamentos existentes. Si los datos modificados son correctos, se debe mostrar un mensaje indicando que la modificación fue satisfactoria, en caso contrario mostrar un mensaje indicando el error que existe. Además se debe verificar la eliminación de un medicamento seleccionado en el listado existente en la base de datos. Se debe mostrar un mensaje preguntando si se desea eliminar el

medicamento, en caso de eliminarlo se muestra un mensaje confirmando que la eliminación fue satisfactoria.

Condiciones de ejecución: El usuario debe haber sido autenticado como Almacenero en el SIGESH.

Flujo Central: Medicamentos/Operaciones/Catálogo de productos/Adicionar.

Escenario: “Registrar medicamento en Tarjeta de Estiba”.

Tabla 3 Casos de prueba. Gestionar Medicamento en Tarjeta de Estiba

Clases Válidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Entrada correcta del medicamento (Código: C1, Clave: 01, Nombre: N1, Subgrupo: S1, Precio: 2.00, Máximo: 50, Grupo Farmacológico: GF1, Unidad de Medida: U/M, Forma Farmacéutica: FF1, Forma de Presentación: FP1, Proveedor: P1, Fabricante: F1).	Tras la entrada de un medicamento, si el proceso es satisfactorio, en la base de datos aparecerán los datos del nuevo medicamento: N1. Se muestra un mensaje indicando que el medicamento fue modificado satisfactoriamente.	Satisfactoria	
Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Campos requeridos vacíos (Código, Clave, Nombre, Subgrupo, Precio, Máximo, Grupo Farmacológico, Unidad de Medida, Forma Farmacéutica, Forma de	Se debe enmarcar de color rojo los campos que deben ser llenados obligatoriamente, si se pasa el mouse por encima del campo se debe mostrar un	Satisfactoria	

Presentación, Proveedor, Fabricante).	mensaje de error.		
Entrar en los campos (Código, Precio, Máximo) valores distintos a números reales.	Se debe enmarcar de color rojo los campos que deben ser llenados correctamente, si se pasa el mouse por encima del campo se debe mostrar un mensaje de error.	Satisfactoria	
Entrar en los campos (Código, Precio, Máximo) caracteres distintos de "números", "puntos" o "guión medio".	No debe permitir escribir estos caracteres en el campo.	Satisfactoria	
Entrar en los campos (Grupo Farmacológico, Unidad de Medida, Forma Farmacéutica, Forma de Presentación, Proveedor, Fabricante, Subgrupo) caracteres.	No debe permitir escribir estos caracteres en el campo.	Satisfactoria	
Evaluación de la prueba	Satisfactoria		

Flujo Central: Medicamentos/Operaciones/Catálogo de productos/Modificar.

Escenario: "Modificar medicamento en Tarjeta de Estiba".

Tabla 4 Casos de prueba. Modificar medicamento en Tarjeta de Estiba

Clases Válidas	Resultado Esperado	Resultado de la Prueba	Observaciones
----------------	--------------------	------------------------	---------------

<p>Seleccionar del listado de medicamentos el que se desea modificar. Modificación correcta del medicamento (Código: C12, Clave: 012, Nombre: N12, Subgrupo: S12, Precio: 2.002, Máximo: 502, Grupo Farmacológico: GF12, Unidad de Medida: U/M1, Forma Farmacéutica: FF12, Forma de Presentación: FP12, Proveedor: P12, Fabricante: F12).</p>	<p>Después de modificar los datos del medicamento, si el proceso es satisfactorio, en la base de datos aparecerán los datos del nuevo medicamento: N12. Se muestra un mensaje indicando que la entrada del medicamento fue satisfactoria.</p>	<p>Satisfactoria</p>	
<p>Clases Inválidas</p>	<p>Resultado Esperado</p>	<p>Resultado de la Prueba</p>	<p>Observaciones</p>
<p>Campos requeridos vacíos (Código, Clave, Nombre, Subgrupo, Precio, Máximo, Grupo Farmacológico, Unidad de Medida, Forma Farmacéutica, Forma de Presentación, Proveedor, Fabricante).</p>	<p>Se debe enmarcar de color rojo los campos que deben ser llenados obligatoriamente, si se pasa el mouse por encima del campo se debe mostrar un mensaje de error.</p>	<p>Satisfactoria</p>	
<p>Entrar en los campos (Código, Precio, Máximo) valores distintos a números reales.</p>	<p>Se debe enmarcar de color rojo los campos que deben ser llenados correctamente, si se pasa el mouse por</p>	<p>Satisfactoria</p>	

	encima del campo se debe mostrar un mensaje de error.		
Entrar en los campos (Código, Precio, Máximo) caracteres distintos de “números”, “puntos” o “guión medio”.	No debe permitir escribir estos caracteres en el campo.	Satisfactoria	
Entrar en los campos (Grupo Farmacológico, Unidad de Medida, Forma Farmacéutica, Forma de Presentación, Proveedor, Fabricante, Subgrupo) caracteres.	No debe permitir escribir estos caracteres en el campo.	Satisfactoria	
Eliminar un medicamento sin seleccionarlo del listado de medicamentos existentes.	Se debe mostrar un mensaje de información indicando que debe seleccionar un medicamento.	Satisfactoria	
Evaluación de la prueba	Satisfactoria		

Flujo Central: Medicamentos/Operaciones/Catálogo de productos/ Eliminar.

Escenario: “Eliminar medicamento en Tarjeta de Estiba”.

Tabla 5 Casos de prueba. Eliminar medicamento en Tarjeta de Estiba

Clases Válidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Seleccionar del listado de medicamentos existentes el que se desea eliminar y	Se debe mostrar un mensaje de confirmación. Si se acepta se muestra un	Satisfactoria	

eliminarlo.	mensaje informando que el medicamento fue eliminado satisfactoriamente.		
Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Eliminar un medicamento sin seleccionarlo del listado de medicamentos existentes.	Se debe mostrar un mensaje de información indicando que debe seleccionar un medicamento.	Satisfactoria	
Evaluación de la prueba	Satisfactoria		

3.7.1 Resultados de las Pruebas

Una evaluación de prueba es una evaluación de los resultados de los esfuerzos de prueba, tales como la cobertura del caso de prueba, la cobertura de código y el estado de los defectos. (Jacobson, y otros., 2000)

Durante el transcurso de las iteraciones de pruebas al sistema, se detectaron 4 no conformidades (NC) en las funcionalidades que responden a las HU de muy alta prioridad, 4 en las de alta y 2 en las de media, como se muestra en la tabla 6.

Tabla 6 No conformidades

Iteraciones	No conformidades			Total
	Muy Alta prioridad	Alta prioridad	Media prioridad	
1ra	3	-	-	3
2da	1	3	-	4
3ra	0	1	2	3
4ta	0	0	0	0

Como se evidencia en la tabla anterior, las NC fueron resueltas satisfactoriamente, contribuyendo así a un mejor funcionamiento del SIGESH.

Consideraciones parciales

Se diseñó el sistema mediante la confección de las tarjetas CRC y se obtuvo el modelo físico de la base de datos.

Se especificó el estándar de código que se siguió para la implementación del SIGESH.

Se describió la arquitectura de desarrollo MVC la cual fue utilizada el marco de trabajo para el desarrollo del SIGESH.

Se realizaron los casos de prueba aplicados a las funcionalidades descritas en las HU del sistema, validando que sus resultados fueran los esperados por el cliente.

CONCLUSIONES GENERALES

El desarrollo de la investigación aportó conocimientos sobre los diferentes servicios que se brindan en el hospital Ernesto Ché Guevara, logrando informatizar los mismos y centralizando la información en un servidor.

El análisis exhaustivo y la selección de las diferentes herramientas, tecnologías y lenguajes, así como la metodología aplicada nos permitió establecer el perfil tecnológico.

Por su parte, la realización de las pruebas realizadas al SIGESH contribuyeron a la obtención de un producto útil para los trabajadores del hospital, el cual responde a las funcionalidades capturas durante el transcurso de la investigación.

RECOMENDACIONES

Una vez concluida la investigación y luego de los resultados obtenidos, se recomienda:

1. Capacitar al personal que trabajará con SIGESH.
2. Comenzar la explotación del nuevo sistema de gestión de la información de los medicamentos, turnos médicos y vacunas en el hospital Ernesto Ché Guevara.
3. Realizar un estudio de los servicios que aún no se han informatizado en el hospital, para desarrollar los módulos correspondientes a estos e incorporarlos a futuras versiones.

REFERENCIAS BIBLIOGRÁFICAS

- Alvarez, Miguel Angel. 2001.** Desarrolloweb.com. [Online] 2001. <http://www.desarrolloweb.com/articulos/392.php>.
- Apache Software Foundation.** Apache Software Foundation. [Online] [Cited: Diciembre 10, 2011.] <http://www.apache.org/>.
- Arias, Michael. 2006.** La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. [Online] 2006. http://www.latindex.ucr.ac.cr/intersedes10/10-art_11.pdf.
- Bahit, Eugenia. 2010.** *Introducción al patrón Arquitectónico MVC*. 2010.
- Curto, Josep. 2006.** Information Management. [Online] 2006. <http://informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion/>.
- División Scientia Pharma. 2005.** SOFTMED SISTEMAS PARA MEDICOS . [Online] 2005. <http://sistemasparamedicos.blogspot.com/>.
- Domínguez, M. Dorado. 2005.** *NetBeans IDE 4.1, la alternativa a Eclipse*. Madrid : Editorial Iberprensa, 2005. pp. 32-34.
- EcuRed. 2009.** Pruebas de software. [Online] 2009. http://www.ecured.cu/index.php/Flujo_de_pruebas_de_un_software.
- EcuRed. 2010.** EcuRed. [Online] 2010. www.ecured.cu/index.php/Photoshop.
- . **2010.** EcuRed. [Online] 2010. <http://www.ecured.cu/index.php/XAMPP>.
- . **2010.** EcuRed. [Online] 2010. <http://www.ecured.cu/index.php/Wamp>.
- . **2011.** EcuRed. [Online] 2011. http://www.ecured.cu/index.php/Aplicaci%C3%B3n_inform%C3%A1tica.
- . **2010.** EcuRed. [Online] 2010. http://www.ecured.cu/index.php/Adobe_Dreamweaver.
- ECUred. 2011.** *Metodología Agíl para proyectos*. [Online] 2011. http://www.ecured.cu/index.php/Metodologia_Agíl_para_proyectos.
- EcuRed. 2007.** Sencha Ext JS. [Online] 2007. http://www.ecured.cu/index.php/Sencha_Ext_JS.
- Escribano., Gerardo Fernández. 2002.** Introducción a Extreme Programming. [Online] 2002. <http://www.dsi.udm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>.
- Flores, Maiqui M. 2009.** Nación & Salud. [Online] 2009. <http://www.nacionysalud.com/node/1354>.
- GACSE SYSTEMS. 2008.** sihowin. [Online] 2008. <http://sihowin.com/sistema/sistema.htm#caract>.

- Gamma, Erich, y otros. 1994.** *Design Patterns: Elements of Reusable Object-Oriented Software*. London : ADDISON-WESLEY, 1994.
- Gerner, Jason, y otros. 2006.** *Professional LAMP: Linux, Apache, MySQL and PHP 5 Web Development*. Indianapolis : Wiley Publishing, Inc., 2006. ISBN: 978-0-7645-9723-7.
- González, Vicente. 2006.** *Manual de ayuda-SCMH*. 2006.
- Gracia, Joaquin. 2005.** Diseño de Software Orientado a Objetos. [Online] 2005.
<http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.
- GSInnova. 2010.** [Online] 2010. <http://www.rational.com.ar/herramientas/roseenterprise.html>.
- Jacobson, Ivar and Booch. 2000.** *El lenguaje unificado de modelado*. s.l. : Addison Wesley, 2000.
- JEFFRIES, Ron. 1999.** Extreme Testing. [online] 1999. [http://www.xprogramming.com/publications/SP99 Extreme for Web.pdf](http://www.xprogramming.com/publications/SP99%20Extreme%20for%20Web.pdf).
- Larman, Craig. 2003.** *UML y Patrones 2da Edición*. s.l. : Prentice Hall, 2003.
- Mora, Sergio Luján. 2010.** *Programación en Internet: Clientes Web*. s.l. : Club Universitario, 2010.
- MySQL. 2010.** Intartius Corporation. [Online] 2010. http://instartius.com/site/?page_id=327.
- Nogueira, Keila García. 2009.** Informática en Salud. [Online] 2009. informatica2009.sld.cu/.
- Nokia Corporation. 2009.** Qt Framework. [Online] 2009. [Cited: Febrero 23, 2012.] www.qt.nokia.com.
- Pérez, Javier Eguíluz. 2009.** *Introducción a JavaScript*. 2009.
- Pino, Dr. Javier Luque. 2009.** Enciclopedia de salud. [Online] 2009.
<http://www.enciclopediasalud.com/categorias/ecologia-biologia-y-biomedicina/articulos/que-es-una-vacuna/>.
- Portillo, José Antonio Pow Sang. 2008.** La especificación de requisitos con casos de uso: Buenas y Malas Prácticas. [Online] 2008. <http://es.scribd.com/doc/14504728/Buenas-Malas-Practicas-CU>.
- Potencier, Fabien. 2006.** Symfony Project. [Online] 2006. [Cited: Noviembre 20, 2011.] www.symfony-project.org.
- Potencier, Fabien and Zaninotto, François. 2008.** *Symfony la guía definitiva*. s.l. : Addison-Wesley, 2008.
- Reynoso, Carlos and Kicillof, Nicolás. 2004.** Estilos y Patrones en la Estrategia Arquitectónica de Microsoft. [Online] 2004. <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>.
- Scribd. 2008.** Scribd. [Online] 2008. es.scribd.com/doc/36570462/postgreSQL-investigacion.
- Sierra, Manuel. 2008.** Aprende a Programar. [Online] 2008.
http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=546:que-es-y-para-que-sirve-el-lenguaje-css-cascading-style-sheets-hojas-de-estilo-dv00203a&catid=46:lenguajes-y-entornos&Itemid=163.
- Softel. 2004.** *Manual de Usuario GALEN LAB, Edición de la versión 5.0*. 2004.

Stanton William, Etzel Michael y Walker Bruce, Mc Graw Hill. 2004. *Fundamentos de Marketing.* 2004.

System Sparx. 2007. [Online] 2007.

http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.

Vidal, Ángel García. 2009. &Análisis. [Online] 2009. http://www.gomezacebo-pombo.com/docs/PUB_117_doc.pdf.

GLOSARIO DE TÉRMINOS

Accesibilidad: de fácil disponibilidad y acceso. Es el grado con que cualquier persona puede acceder a determinado servicio u objeto, independientemente de las capacidades físicas, técnicas o cognitivas que posea.

Cross-browser: Aplicación web que tiene la habilidad de correr en todos los navegadores web.

Framework: estructura conceptual y tecnológica de soporte definido, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

Historias de Usuarios: técnica utilizada en la metodología de desarrollo de software XP, para especificar los requisitos del software.

IDE: es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

IIS: siglas del inglés Internet Information Server es un paquete de servicios para ordenadores que funcionan con Windows.

Java: lenguaje de programación orientado a objetos desarrollado por Sun MicroSystem en los años 90.

Metodología: parte del proceso de investigación o método científico que sigue a la propedéutica y permite sistematizar los métodos y las técnicas necesarias para llevarla a cabo.

PDF (Portable Document Format): formato de archivos desarrollado por Adobe Systems.

Portabilidad: en términos informáticos se refiere a la independencia que posee el código fuente de un sistema informático del sistema operativo que lo utilice, constituyendo un código reusable en varios sistemas operativos.

Prioridad: elemento cualitativo que determina la máxima preferencia; se utiliza en planeación o programación para señalar lo que tiene mayor importancia.

Prototipo de interfaz: representación limitada del diseño de un producto que permite a las partes responsables de su creación experimentar su uso, probarlo en situaciones reales y explorar su uso. Un prototipo puede ser cualquier cosa, desde un trozo de papel con sencillos dibujos a un complejo software.

SCRUM: define un marco de trabajo ágil para la gestión de proyectos, basado en iteraciones y entregas incrementales de desarrollo de un producto o servicio. Especialmente indicado para proyectos con requisitos cambiantes.

Servidor Web: software que suministra páginas Web en respuesta a las peticiones de los navegadores Web.

XP (eXtreme Programing): es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.