

**UNIVERSIDAD CENTRAL DE VENEZUELA  
FACULTAD DE CIENCIAS  
ESCUELA DE COMPUTACIÓN  
CENTRO DE ENSEÑANZA ASISTIDA POR COMPUTADOR - CENEAC**

**Desarrollo de un prototipo de  
sistema de inventario y manejo  
de historias médicas físicas**



**Trabajo Especial de Grado presentado ante la Ilustre  
Universidad Central de Venezuela por el  
Br. João C. Alves Castillo (CI. 18.249.125)  
Para optar al título de Licenciado en Computación**

**Tutora: Profa. Yusneyi Y. Carballo Barrera**

**8 de Febrero de 2013**

UNIVERSIDAD CENTRAL DE VENEZUELA  
FACULTAD DE CIENCIAS  
ESCUELA DE COMPUTACIÓN

## **ACTA**

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado "**Desarrollo del prototipo de un sistema de inventario y manejo de historias médicas físicas**" y presentado por el **Br. João C. Alves Castillo (CI. 18.249.125)**, a los fines de optar al título de **Licenciado en Computación**, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día \_\_\_ de \_\_\_\_\_ de \_\_\_\_\_, a las \_\_\_\_\_ horas, para que su autor lo defendiera en forma pública, lo que este hizo en \_\_\_\_\_ de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondió a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de \_\_\_\_ puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día \_\_\_ de \_\_\_\_\_ de 2013.

---

**Profa. Yusneyi Carballo Barrera**  
**(Tutora)**

---

**Profa. Concetinna Di Vasta**  
**(Jurado)**

---

**Prof. Wuilfredo Rangel**  
**(Jurado)**

Cuando finalmente me siento en ese el último escalón que me separa de obtener el tan anhelado título universitario, se me hace increíble solo pensar donde estoy parado y recordar todos los obstáculos que tuve que superar, para finalmente cumplir el objetivo inicial que un día me planteé cuando estaba inscribiendo mi primer semestre en esta magnífica casa de estudios

Son muchas las personas que me han ayudado y que deberían aparecer mencionadas en este trabajo, amigos de la carrera que en ciertas etapas se convirtieron en mi mano derecha, profesores que han dejado huella y han sembrado la esperanza de lograr un mejor futuro a partir de la educación universitaria, familiares que me han dado aliento en los momentos en que me daba por vencido, en fin, son tantas personas que sería difícil poder mencionarlas en este pequeño espacio.

Debo agradecer infinitamente a Dios y a mis Santos por permitirme llegar hasta esta etapa de la carrera y culminar satisfactoriamente este trabajo, pues fueron mi refugio en muchas ocasiones cuando me faltaba ánimo y también fueron los proveedores de las oportunidades y el éxito con el que logré alcanzar muchos objetivos.

A mis padres, Evelyn Castillo y Juan Alves, los cuales son motivo de inspiración y han estado conmigo incondicionalmente en todos los momentos en los cuales los he necesitado, dándome ánimo en los momentos en que desmayaba y celebrando junto a mi cada uno de mis triunfos en el arduo camino hacia el Aula Magna, definitivamente este título no es solo mío, es una gran recompensa y satisfacción que les quiero regalar a ellos por todas esas cosas que han hecho incondicionalmente a lo largo de este trayecto universitario para lograr mi bienestar.

A mi hermana, Joalyn Alves, por siempre brindarme su ayuda incondicional para resolver los problemas que se me han presentado a lo largo de este camino y por ser ese gran ejemplo a seguir que me motiva y me inspira a ser mejor persona en todos los aspectos de la vida.

A mi abuela, Doris Lara, quien no se encuentra físicamente conmigo, pero no me cabe la menor duda que espiritualmente esta siempre a mi lado y me acompañará a recibir el título y la medalla en mi acto de graduación.

A mi tutor organizacional, Lic. Carlos Rivera, por brindarme la oportunidad de realizar este interesante trabajo de investigación, y quien siempre me apoyó a lo largo del mismo.

A mi tutora, la Profesora Yusneyi Carballo, por la confianza que depositó en mí, permitiéndome desarrollar esta investigación, estar siempre pendiente de mi trabajo y ser ese empujón que necesitaba en esta última etapa de mi carrera, además de ser una persona muy admirable no solo en el ámbito profesional, sino por su sencillez y calidad humana.

También debo agradecer a mis tías, tíos, demás familiares y amigos, los cuales como explique anteriormente, no menciono directamente pero les agradezco por haber contribuido con su pequeño grano de arena a concretar mis objetivos profesionales y hoy estar escribiendo estas líneas.

Universidad Central de Venezuela  
Facultad de Ciencias  
Escuela de Computación  
Centro de Enseñanza Asistida por Computador - CENEAC

**Desarrollo del prototipo de un sistema de inventario y manejo de historias  
médicas físicas**

**Autor:** Br. João C. Alves Castillo

**Tutora:** Profa. Yusneyi Carballo Barrera

**Fecha:** 8 de Febrero de 2013

**RESUMEN**

El presente Trabajo Especial de Grado describe la problemática actual existente en los procesos de manejo y almacenamiento de historias médicas físicas en un centro de salud de la ciudad de Caracas.

Se plantea en este presente Trabajo Especial de Grado el desarrollo de un prototipo que automatice el manejo y almacenamiento de historias médicas de manera eficaz. De esta manera se pretende cubrir las necesidades expuestas por el centro de salud, beneficiando tanto al personal encargado de la realización de cada uno de los procesos, como a los clientes finales (los pacientes), ya que muchas veces se ven afectados por los problemas generados debido a la falta de automatización en el manejo y almacenamiento de historias médicas físicas.

En el desarrollo de este prototipo se utiliza como método de desarrollo, una adaptación del método XP, el patrón de desarrollo MVC, el lenguaje de programación JEE, el entorno de desarrollo Eclipse, el sistema manejador de base de datos PostgreSQL, entre los más importantes.

Como resultado de este trabajo se tiene el desarrollo del prototipo de un sistema que resuelve eficazmente los problemas de manejo y almacenamiento de historias médicas presentes en el centro de salud, incorporando para ello módulos que automatizan los procesos de entrada, salida, solicitudes y peticiones de historias médicas físicas, destacando la realización de una portada para identificar de manera única a cada historia dentro del centro de salud y la posibilidad de realizar inventario de las mismas.

**Palabras Claves:** Historia médica, almacenamiento, XP (Programación eXtrema), JEE (Java Enterprise Edition), MVC (Modelo-Vista-Controlador).

---

**Índice**

INTRODUCCIÓN.....	I
CAPÍTULO 1. PROBLEMA DE INVESTIGACIÓN .....	1
1.1 PLANTEAMIENTO DEL PROBLEMA .....	1
1.2 PLANTEAMIENTO DE LA SOLUCIÓN .....	2
1.3 OBJETIVO GENERAL .....	4
1.4 OBJETIVOS ESPECÍFICOS.....	5
1.5 ALCANCE DE LA APLICACIÓN.....	5
CAPÍTULO 2. MARCO TEÓRICO CONCEPTUAL .....	7
2.1 HISTORIA MÉDICA .....	7
2.2 ORIGEN E INICIO DE LA HISTORIA MÉDICA .....	7
2.3 CONTENIDO DE LA HISTORIA MÉDICA.....	7
2.4 CARACTERÍSTICAS DE LAS HISTORIAS MÉDICAS.....	9
2.5 FINALIDAD DE LA HISTORIA MÉDICA .....	10
2.6 PRIVACIDAD Y CONFIDENCIALIDAD DE LA HISTORIA MÉDICA.....	11
2.7 GESTIÓN Y ALMACENAMIENTO DE HISTORIAS MÉDICAS FÍSICAS .....	12
CAPÍTULO 3. MARCO METODOLÓGICO.....	14
3.1 MÉTODO ÁGIL .....	14
3.2 PROGRAMACIÓN EXTREMA (XP).....	15
3.2.1 OBJETIVOS DE LA PROGRAMACIÓN EXTREMA (XP) .....	15
3.2.2 PRINCIPIOS BÁSICOS DE LA PROGRAMACIÓN EXTREMA (XP).....	16
3.2.3 ACTIVIDADES DE LA PROGRAMACIÓN EXTREMA (XP).....	18
3.2.4 HISTORIAS DE USUARIOS.....	19
3.2.5 VENTAJAS DE LA PROGRAMACIÓN EXTREMA (XP) .....	19
3.2.6 DESVENTAJAS DE LA PROGRAMACIÓN EXTREMA (XP) .....	20
3.3 PATRÓN DE DISEÑO MODELO – VISTA – CONTROLADOR (MVC) .....	20
3.4 JAVA ENTERPRISE EDITION (JEE) .....	21
3.4.1 ARQUITECTURA DE JEE.....	22
3.4.2 VENTAJAS DEL USO DE JEE .....	23
3.5 JSF (JAVA SERVER FACES) .....	24
3.6 RICHFACES (RED HAT) .....	25
3.7 SPRING.....	26
3.8 HIBERNATE .....	27
3.9 MOTOR DE GENERACIÓN DE REPORTES JASPERREPORT .....	27
3.10 SPRING SECURITY .....	28
CAPÍTULO 4. MARCO APLICATIVO .....	29
4.1 ADAPTACIÓN DE LA PROGRAMACIÓN EXTREMA (XP).....	29

---

4.1.1	PLANIFICACIÓN .....	29
4.1.2	DISEÑO .....	31
4.1.3	CODIFICACIÓN .....	31
4.1.4	PRUEBAS .....	32
4.2	ANÁLISIS GLOBAL DEL PROTOTIPO DE SISTEMA .....	32
4.2.1	HISTORIAS DE USUARIO .....	32
4.2.2	METÁFORA DEL PROTOTIPO DE SISTEMA.....	35
4.3	ESPECIFICACIONES TÉCNICAS DEL PROTOTIPO DE SISTEMA .....	35
4.3.1	ARQUITECTURA DE DESARROLLO .....	36
4.4	PLAN DE ITERACIONES .....	37
	RESULTADOS .....	59
	CONCLUSIONES Y RECOMENDACIONES .....	61
	REFERENCIAS BIBLIOGRÁFICAS Y DÍGITALES .....	63
	ANEXO 1: CONVENCIONES DE CÓDIGO PARA EL LENGUAJE DE PROGRAMACIÓN JAVA.....	65
	ANEXO 2: TABLAS DEL MODELO RELACIONAL .....	70

---

**Índice de Imágenes**

Imagen 1: Formato de portada de una historia médica física .....	3
Imagen 2: Interacción de los componentes del Modelo – Vista – Controlador .....	21
Imagen 3: Esquema de un modelo de 3 o 4 capas .....	23
Imagen 4: Metáfora del prototipo de sistema .....	35
Imagen 5: Arquitectura propuesta, considerando el patrón MVC .....	36
Imagen 6: Modelo Entidad/Relación del prototipo de sistema .....	38
Imagen 5: Menú principal del prototipo de sistema .....	40
Imagen 8: Elementos de la portada de la historia médica física .....	43
Imagen 9: Pantalla para definir un nuevo almacén en el prototipo .....	43
Imagen 10: Listado de almacenes definidos en el prototipo .....	44
Imagen 11: Pantalla para registrar la entrada de historias médicas al almacén/sucursal .....	47
Imagen 12: Pantalla para registrar la salida de historias médicas del almacén/sucursal .....	47
Imagen 13: Pantalla para administrar las peticiones (solicitudes) que llegan a un almacén/sucursal..	50
Imagen 14: Pantalla para registrar las solicitudes de historias médicas físicas a otros almacenes/sucursales.....	51
Imagen 15: Interfaz para el control de historias médicas en préstamo.....	54
Imagen 16: Interfaz para realizar el inventario de historias médicas .....	54
Imagen 17: Interfaz para el control de historias médicas en tránsito .....	57
Imagen 18: Interfaz para el control de historias médicas extraviadas .....	57



---

**Índice de Tablas**

Tabla 1: Modelo de tabla para la bitácora de desarrollo.....	30
Tabla 2: Formato para registrar las Historias de Usuario .....	30
Tabla 3: Formato para el registro de pruebas .....	32
Tabla 4: Planificación de la iteración 1 .....	39
Tabla 5: Bitácora de desarrollo de la iteración 1 .....	39
Tabla 6: Casos de prueba. Iteración 1 .....	41
Tabla 7: Planificación de la iteración 2 .....	42
Tabla 8: Bitácora de desarrollo de la iteración 2.....	42
Tabla 9: Casos de prueba. Iteración 2 .....	45
Tabla 11: Planificación de la iteración 3 .....	45
Tabla 11: Bitácora de desarrollo de la iteración 3 .....	46
Tabla 12: Casos de prueba. Iteración 3.....	48
Tabla 13: Planificación de la iteración 4 .....	49
Tabla 14: Bitácora de desarrollo de la iteración 4 .....	49
Tabla 15: Casos de prueba. Iteración 4.....	52
Tabla 16: Planificación de la iteración 5 .....	52
Tabla 17: Bitácora de desarrollo de la iteración 5 .....	53
Tabla 19: Casos de prueba. Iteración 5.....	55
Tabla 19: Planificación de la iteración 6 .....	56
Tabla 20: Bitácora de desarrollo de la iteración 6 .....	56
Tabla 21: Casos de prueba. Iteración 6.....	58

## **INTRODUCCIÓN**

Las historias médicas son documentos físicos y legales donde se encuentra toda la información de un paciente, junto a los diferentes diagnósticos, tratamientos, evolución y diversos resultados de exámenes que el paciente se ha realizado a lo largo de la consulta en un centro de salud. En la mayoría de los casos, toda esta información se encuentra ubicada en una carpeta identificada según los estándares del centro de salud.

Tomando en cuenta la importancia de este documento (historia médica), surge la necesidad de gestionar eficientemente su uso, en cuanto a:

1. Control de la ubicación física.
2. Disponibilidad de la historia médica en el centro de salud en el momento en que el médico requiere consultar su información.

Este último punto cobra mayor importancia, cuando el médico presta servicio a sus pacientes en diferentes lugares o el centro de salud posee varias sucursales. Incluso, suele suceder que no se dispone de suficiente espacio para almacenar todas las historias médicas físicas de los pacientes en el mismo centro de salud y se recurre al uso de un almacén externo para dicho fin.

Cuando se está en presencia de alguna de las situaciones anteriores, es de suma importancia determinar la ubicación exacta de la historia médica del paciente y gestionar de manera adecuada el movimiento de este documento entre el profesional de la salud y el archivo donde se almacena.

El centro de salud que nos servirá de caso de estudio, es una unidad de oncología médica con sede en Caracas, creada y diseñada para brindar servicios de alta calidad en el diagnóstico y tratamiento del cáncer. Con el propósito de prestar servicios de calidad a un mayor número de pacientes en el área metropolitana, el centro de salud está por inaugurar una nueva sede en la ciudad capital, lo cual incrementa la necesidad de contar con sistemas automatizados para la gestión de las historias médicas físicas entre ambas sedes y el almacén externo del que disponen actualmente. Es importante destacar que en la actualidad, no cuentan con herramienta alguna que permita gestionar el control de historias médicas físicas,

Para resolver la problemática anteriormente planteada, se estudiaron previamente en el trabajo de seminario, todos los procesos asociados a la gestión y almacenamiento actual de las historias médicas físicas, lo cual permitió realizar un exhaustivo análisis de los mismos, con el objetivo de elaborar una propuesta viable para el desarrollo de un prototipo que permitiera la gestión y el control de las mismas.

Partiendo de lo comentado anteriormente, se desarrolló un prototipo de sistema web, que permite gestionar y controlar de manera eficaz el uso de las historias médicas físicas en el centro de salud seleccionado como caso de estudio, garantizando de esta manera la satisfacción de los pacientes y a su vez, facilitando las labores de los médicos que constituyen el grupo médico.

Este trabajo está estructurado en cuatro capítulos, los cuales se describen a continuación:

Capítulo 1: Problema de investigación, planteamiento general de la solución, objetivo general del TEG, objetivos específicos y alcance.

Capítulo 2: Presentación del marco teórico conceptual, es decir, los términos y conceptos necesarios para entender la problemática planteada, pasando desde puntos básicos como: ¿qué es una historia médica?, ¿cómo se origina e inicia una historia médica?, ¿cuál es el objetivo, contenido y características, de una historia médica?, hasta puntos más complejos referentes a la confidencialidad, privacidad, gestión y almacenamiento de las mismas.

Capítulo 3: Marco metodológico, el cual consta de la descripción y análisis de los métodos que se emplearán en el estudio de investigación, describiendo el método XP, el patrón de desarrollo MVC, el lenguaje de programación Java JEE, el entorno de desarrollo Eclipse, el sistema manejador de base de datos PostgreSQL, y demás elementos utilizados en el desarrollo del prototipo de sistema.

Capítulo 4: Marco aplicativo, en este capítulo se presenta la adaptación de las actividades que cumplen con los principios de la Programación Extrema, las cuales se llevan a cabo durante el proceso de desarrollo. Además, se describe cada una de las iteraciones especificando la planificación y ejecución de las tareas correspondientes.

Finalmente, se presentan los resultados de este TEG, las conclusiones, referencias bibliográficas utilizadas y anexos que suministran información más detallada sobre el desarrollo de la solución propuesta.

## **CAPÍTULO 1. PROBLEMA DE INVESTIGACIÓN**

En este capítulo se presenta el problema que motiva este Trabajo Especial de Grado, una descripción general de la solución planteada, el objetivo general y los objetivos específicos y el alcance del presente trabajo.

### **1.1 PLANTEAMIENTO DEL PROBLEMA**

El centro de salud que sirvió de caso de estudio, es una unidad de oncología médica con sede en Caracas, creada y diseñada para brindar servicios de alta calidad en el diagnóstico y tratamiento del cáncer con más de 20 años de servicio y fama mundial, la cual ha decidido abrir nuevas unidades de atención a sus pacientes en diferentes lugares de la ciudad capital.

Actualmente presentan un problema de eficiencia en el manejo de las historias médicas físicas de los pacientes que atienden. Una historia médica física, es una carpeta organizada que contiene los datos personales del paciente y documentos con información médica importante (informes médicos, diversos exámenes médicos, etc.).

A lo largo del tiempo de servicio y por motivos de espacio físico, el centro de salud se ha visto en la necesidad de ubicar todas sus historias médicas en un almacén situado en un lugar distinto a su sede principal. Cada vez que se requiere la historia médica de un paciente, esta es solicitada al almacén y enviada a la sede principal. Cabe destacar en este punto que la solicitud de historias al almacén, se realiza por correo electrónico y, si se trata de una urgencia, se realiza por teléfono. Esta situación ha llegado a crear conflictos, como por ejemplo: se realiza la solicitud de una historia requerida en la sede principal al almacén, y el almacén no posee la historia; en esta situación la sede principal tiene las siguientes alternativas de la posible ubicación de la historia médica:

- La historia médica del paciente en cuestión había sido utilizada en la sede y no había sido devuelta al almacén.
- La historia médica se la llevo un médico para estudiar más a fondo un caso, realizar informes médicos o realizar trabajos de investigación.
- La historia médica ya fue solicitada al almacén y va en camino a la sede (está en tránsito).

Si la historia médica no se encuentra en alguno de los casos anteriores, se presenta un problema mayor, la historia médica esta pérdida. Si esto ocurre se pierde información sumamente importante del paciente acarreado con ello el descontento del mismo y los problemas que tendrá el

médico al momento de tratarlo, ya que este debe recordar el caso y hasta repetir exámenes médicos si es necesario.

Con el fin de garantizar la satisfacción de todos sus pacientes y a su vez facilitar la labor de los médicos que trabajan en esta unidad oncológica, se realizan actividades exhaustivas para garantizar la seguridad de los documentos y su disponibilidad en todo momento, las cuales consumen un tiempo considerable para el personal responsable de estas en el centro de salud.

La gerencia del centro de salud tomada como caso de estudio, tiene en vista la próxima apertura de un importante centro médico privado en Caracas (nueva sede), razón por la cual crece su preocupación ante el problema planteado ya que una nueva sede implica directamente un aumento considerable tanto en la cantidad de historias, como en los movimientos que tendrán estas entre los lugares donde se presta servicio al paciente y el almacén donde se guardan las historias médicas físicas hoy en día. Cabe destacar en este punto que el centro de salud tiene actualmente 15.000 historias aproximadamente y atiende alrededor de 200 pacientes diarios.

## **1.2 PLANTEAMIENTO DE LA SOLUCIÓN**

Para resolver la necesidad del centro de salud seleccionado como caso de estudio, se plantea como solución, el desarrollo de un prototipo de sistema web, con una interfaz sencilla, intuitiva y fácilmente usable por el personal, tomando en cuenta las siguientes consideraciones:

Cada historia médica tendrá una portada que contendrá datos básicos del paciente (nombre, cédula y número de historia) y un código de barras generado por el prototipo del sistema, el cual permitirá (de manera sencilla y evitando el riesgo de transcripción del operador del prototipo de sistema) el registro de la entrada y salida de cada historia médica de los lugares definidos como almacenes en el prototipo de sistema. Además del control de entradas y salidas de historias desde y hacia los distintos almacenes, se debe controlar el motivo de su solicitud (principalmente, envío al centro para consulta médica y préstamo al personal médico por motivos de investigación, elaboración de informes médicos, entre otras). La portada tendrá un formato similar al mostrado en la imagen 1.

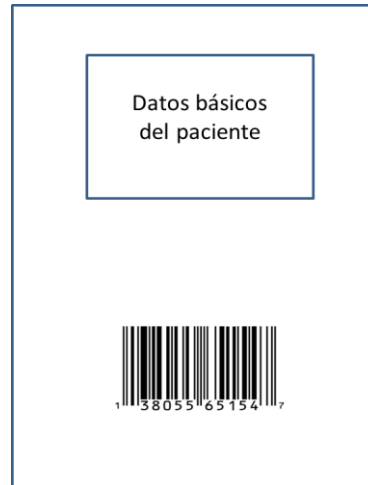


Imagen 1: Formato de portada de una historia médica física

Con el uso de esta portada se podrá identificar de manera unívoca cada una de las historias médicas físicas, facilitando el registro de los movimientos de estas entre los distintos almacenes. Es importante destacar que la salida de una historia de un lugar a otro, determina el inicio de un estatus denominado "en tránsito", es decir, se encuentra en manos de la persona encargada de transportar las historias de un lugar a otro. Esta es una acotación importante ya que si una historia sale de un lugar y no tiene entrada en su lugar destino, su estado aparecerá en el prototipo de sistema como "en tránsito" y se podrá determinar quién fue la persona responsable de la pérdida de la misma.

Habiendo ingresado a un almacén definido en el prototipo de sistema, se podrá determinar la ubicación de una historia médica. Una historia física puede encontrarse en:

- Alguna de las sedes de la Institución de Salud.
- Almacén Principal.
- En manos de un médico, en este caso se desea saber cuál es el médico.
- En tránsito, es el caso en que la historia se encuentra viajando de un lugar a otro. En este caso se desea conocer el origen y destino de la historia médica física, así como el responsable de su traslado.

Un usuario ubicado en cualquiera de las sedes y con los permisos para tal fin, tiene la opción de "Solicitar historia médica", esta opción le permite registrar la solicitud de una historia médica o un grupo de historias médicas al almacén que las posea al momento de realizar esta. Cada vez que se registra una solicitud, el encargado del almacén en el que se encuentra la historia o grupo de historias, le aparecerá en la opción "Petición" el detalle de lo solicitado.

Así mismo, un usuario en cualquier sede/almacén con permisos para tal fin, tienen la opción de registrar el envío de historias médicas mediante la opción "Salida de historias", esta opción permite

registrar la salida de una historia médica o un grupo de historias médicas al lugar desde el que se hizo la solicitud.

Por cada salida de una historia o grupo de historias de un almacén, se debe generar una orden de salida en la que quede constancia de las historias que están saliendo. Así mismo, debe contener información sobre el motivo de la salida de la historia o grupo de historias: Solicitud desde una sede, solicitud de préstamo por un médico tratante para investigación, entre otras. Cuando se trate de salida por concepto de préstamo, se debe registrar en el prototipo de sistema, además de las historias que están saliendo, el nombre del médico que realizó la solicitud.

Para registrar la entrada de una historia en cualquiera de los lugares definidos (sedes o almacenes) bastará solamente con seleccionar la opción de "Entrada de Historia Médica" y a través del uso de un escáner de código de barras, identificar la historia en cuestión (a través de la lectura del código de barras incluido en la portada de cada historia) y hacer el registro de la llegada.

Por último se desarrolla una opción que le permita al usuario realizar un inventario de las historias existentes en un almacén/sede. Esta opción sencillamente permitirá que el usuario, mediante el uso de un escáner de código de barras, capture el código de las historias médicas existentes en el almacén/sede y posterior a ello el prototipo de sistema realice una comparación entre los códigos escaneados de las historias físicas y la base de datos, con el único objetivo de ofrecer al usuario un resumen de las historias sobrantes o faltantes en dicho almacén... En este resumen se ofrecerá información acerca de:

- Total de historias escaneadas
- Total de historias extraviadas (historias que se encuentran en la base de datos de ese almacén/sede y no fue verificada físicamente con el escáner de código de barras)
- Total de historias sobrantes en el almacén (historia que fue leída con el escáner de código de barras y no existía en la base de datos del almacén/sede),
- entre otra información relevante que así disponga el cliente.

Cabe destacar que la diferencia entre un almacén y una sede reside únicamente en que los almacenes son lugares estrictamente destinados para guardar las historias médicas físicas, mientras que las sedes pueden almacenar historias médicas y también son los lugares donde el personal médico atiende a sus pacientes.

### **1.3 OBJETIVO GENERAL**

Desarrollar un prototipo de sistema web, que permita gestionar de manera eficaz el uso y almacenamiento de las historias médicas físicas en el centro de salud seleccionado como caso de estudio.

#### **1.4 OBJETIVOS ESPECÍFICOS**

1. Diseñar e implementar el modelo de la base de datos, que permita reflejar la información recaudada en la fase de análisis de la aplicación.
2. Diseñar y desarrollar un prototipo de sistema web bajo las tecnologías previamente seleccionadas en el trabajo de seminario y según los requerimientos del centro de salud seleccionado como caso de estudio, adaptando y aplicando el método XP y los lineamientos de diseño web relacionado con usabilidad y accesibilidad.
3. Realizar las pruebas pertinentes que permitan garantizar el correcto funcionamiento del prototipo.

#### **1.5 ALCANCE DE LA APLICACIÓN**

El prototipo de sistema sugerido, debe cumplir los requerimientos planteados por la gerencia del centro de salud que sirve como caso de estudio, y además debe permitir su uso en otras instituciones de salud que pudieran tener necesidades similares. En este sentido, el alcance de esta investigación se circunscribe a desarrollar soluciones para las siguientes funcionalidades:

1. Permitir el acceso a la aplicación mediante el uso de un nombre de usuario y contraseña, los cuales deben ser previamente definidos en un módulo de configuración y asignación de permisos.
2. Permitir al personal encargado, realizar la generación e impresión de la portada de una historia médica. Esta portada contiene los datos básicos del paciente: nombre, apellido, cédula, número de historia, y adicionalmente el código de barras asociado a la historia, el cual debe ser generado por la aplicación.
3. Permitir al personal encargado, definir los almacenes de historias médicas que maneja físicamente el centro de salud en cuestión.
4. Registrar el ingreso de historias médicas físicas a un almacén/sede, mediante el código de barras generado en la portada, el cual es asociado a una historia médica física.
5. Registrar la salida de historias médicas físicas a un almacén/sede mediante el código de barras generado en la portada, el cual es asociado a una historia médica física.
6. Realizar solicitudes de historias médicas físicas requeridas por algún motivo (consulta, estudio, realización de informe médico) desde un almacén/sede al almacén/sede en donde se ubica la historia médica.
7. Recibir peticiones de historias médicas provenientes de una sede.



8. Permitir la realización de una consulta rápida mediante el escaneo del código de barras de la portada de una historia médica, para mostrar información relacionada con: ubicación física de la historia y estatus actual de la misma (por ejemplo: "portada impresa", "en almacén", "en tránsito").
9. Permitir la realización de un inventario de historias médicas físicas, el cual debe consistir en cotejar o comparar las historias médicas registradas en el prototipo de sistema mediante la aplicación web, con la existencia física y tangible de las historias médicas que se encuentran en ese almacén o sede. Luego de realizar este inventario, imprimir un reporte con la información suministrada por el mismo.

## **CAPÍTULO 2. MARCO TEÓRICO CONCEPTUAL**

El propósito de este capítulo es demostrar las bases conceptuales que sirven de fundamento para el desarrollo del prototipo de sistema. A lo largo del mismo se presentan todos aquellos conceptos necesarios para comprender la problemática actual del centro de salud y de la misma manera sirvan de fundamento para desarrollar exitosamente la solución planteada en el capítulo anterior.

### **2.1 HISTORIA MÉDICA**

Para Pardos (1997), una historia médica, también llamada historia clínica, es un documento médico legal en donde queda constancia escrita de todas las evaluaciones y resultados de exámenes médicos, que fueron realizados a un paciente a lo largo de la evolución y de los tratamientos aplicados por un médico.

La historia médica tiene como objetivo recaudar a lo largo del tiempo todos los datos clínicos que tengan relación con la situación del paciente, su proceso de evolución, tratamiento y recuperación, adicionalmente incluye juicios, documentos, procedimientos, informaciones y expresiones del consentimiento del paciente.

### **2.2 ORIGEN E INICIO DE LA HISTORIA MÉDICA**

La historia médica se origina con el primer contacto que establece el paciente con el sistema de salud, bien sea por enfermedad o cualquier tipo de control médico. Por ejemplo: El nacimiento de un niño es el punto de inicio de la historia médica, ya que el pediatra abre un expediente con todos los datos importantes del paciente (sexo, peso, talla, valoración tras el parto, etc.) más una primera exploración.

A nivel hospitalario la historia médica se origina ante el estudio y atención del paciente por una determinada patología, como un traumatismo, una fiebre o un dolor, o el inicio del seguimiento de un proceso, como por ejemplo el embarazo; en ambos casos pudiendo causar ingreso hospitalario o no.

### **2.3 CONTENIDO DE LA HISTORIA MÉDICA**

El objetivo principal de una historia médica es facilitar el manejo de la información asociada a la asistencia sanitaria, dejando constancia de todos aquellos datos que, bajo criterio médico, permitan el conocimiento veraz y actualizado del estado de salud de un paciente. Para Pardos (1997), generalmente una historia médica incorpora información relacionada con:

- La anamnesis, comprende datos personales, antecedentes familiares, personales y demás datos relevantes suministrados por el paciente.
- La exploración física.
- La evolución.
- Las órdenes médicas.
- La hoja de interconsulta.
- Los informes de exploraciones complementarias.
- Exámenes médicos realizados así como sus respectivas conclusiones
- La evolución y planificación de cuidados de enfermería.
- La aplicación terapéutica de enfermería.

Según Pardos (1997), todos estos datos incorporados en la historia médica son recaudados siguiendo el método clínico, de la siguiente manera:

- **Anamnesis:** Es el primer paso y el punto de encuentro entre el médico y el paciente. La anamnesis se debe realizar de tal modo que el paciente pueda responder a las preguntas de una forma franca y espontánea. Éste debe sentirse en un estado de confianza de tal manera que ningún tema mencionado en presencia del médico será motivo de censura, ridículo o vergüenza. A lo largo de la conversación entre ambos, el médico debe estar muy atento para obtener los datos claves que el paciente comenta de su enfermedad, y de este modo, encaminar la investigación diagnóstica y las tácticas hacia un adecuado diagnóstico.
- **Exploración física:** Se refiere a un conjunto de procedimientos que realiza el médico al paciente, luego que realiza la anamnesis, con el objetivo de obtener un conjunto de datos puntuales o signos que estén relacionados con los síntomas que refiere el paciente. Una vez obtenidos los datos de la exploración física y de la anamnesis, el médico tiene la base para construir un diagnóstico inicial a partir del cual puede solicitar la realización de exámenes más específicos que confirmen el diagnóstico de una enfermedad.
- **Exploración complementaria:** Es una prueba diagnóstica que puede solicitar el médico para confirmar el diagnóstico inicial o para descartarlo. Esta prueba se realiza Luego de la anamnesis y la exploración física, y en conjunto con la exploración física conforman lo que llamamos examen médico.
- **Diagnósticos Presuntivos:** Es el resultado o conclusión que obtiene el médico, de acuerdo a su experiencia y estudios, luego de realizar la anamnesis y el examen físico. Reciben este nombre ya que están sujetos a los resultados de diversos exámenes que puede solicitar el médico para corroborarlos.

- **Tratamiento:** Una vez confirmado el diagnóstico a través de los exámenes médicos realizados, el médico procede a colocar el tratamiento adecuado con la finalidad de curar o aliviar las enfermedades o síntomas presentados por el paciente.

#### 2.4 CARACTERÍSTICAS DE LAS HISTORIAS MÉDICAS

Según Díaz (2009), la historia médica, a lo largo de la historia, se ha caracterizado por lo siguiente:

- **Confidencialidad:** Es una de las características más importantes de la historia médica, y se basa en que el secreto médico es uno de los deberes principales del ejercicio de la medicina, teniendo origen en los tiempos más antiguos de la historia y manteniéndose aún vigente a pesar del paso de los años. El secreto médico, la confidencialidad e intimidad y la historia clínica, son tres cuestiones que se implican recíprocamente y se relacionan. La historia clínica es el soporte documental biográfico de la asistencia sanitaria administrada a un paciente, por lo cual es uno de los documentos más privados que existe para una persona.
- **Seguridad:** Se refiere a que en la historia médica debe constar la identificación de los datos del paciente, así como también de los médicos y demás profesionales de la salud que intervengan a lo largo del proceso en el cual este asiste al centro de salud.
- **Disponibilidad:** Teniendo en cuenta el punto de la confidencialidad e intimidad de la información contenida en la historia médica, se debe garantizar de igual forma la disponibilidad de la misma, facilitándose en los casos legalmente contemplados, su acceso y disponibilidad.
- **Única:** Sería ideal que desde el nacimiento de una persona y a lo largo de toda su vida existiera solo una historia en la cual se pudiera observar el control médico en el tiempo. Esta cualidad no se ve comúnmente, pero por lo menos se espera que en un centro de salud o por especialidad médica, un paciente tenga solo una historia médica.
- **Legible:** La legibilidad debe estar presente en todo momento en las historias médicas, una historia médica desordenada e ilegible puede ser perjudicial tanto para los médicos, porque dificulta su labor al momento de atender un paciente que se presente a su consulta, y a los pacientes por los errores que pueden ser generados por una inadecuada interpretación de los datos que contiene la historia médica.

## 2.5 FINALIDAD DE LA HISTORIA MÉDICA

Como se ha venido tratando en los puntos anteriores, la historia médica tiene como finalidad principal recaudar datos acerca del estado de salud de un paciente con el objetivo de facilitar la asistencia sanitaria. Se puede considerar a la historia médica como un instrumento esencial para el buen ejercicio médico, porque sin su uso es imposible que el profesional de la medicina tenga una visión amplia y completa del paciente a lo largo del tiempo para de esta manera poder prestar asistencia sanitaria.

Si bien, el objetivo primordial de la historia médica es el asistencial, no debemos obviar otros aspectos importantes sobre esta, como lo son los siguientes:

- **Docencia e investigación:** Mediante las historias médicas, los galenos pueden realizar estudios e investigaciones sobre determinadas patologías, esto con el fin de encontrar nuevos y mejores tratamientos, conocer a profundidad una enfermedad y si es posible mejorar la calidad de vida de este paciente y de todos los demás que presente la misma patología, todo esto respetando la confidencialidad de los datos personales existentes en la historia médica del paciente.
- **Evaluación de la calidad asistencial:** La historia médica es considerada legalmente como un derecho que tiene el paciente para garantizar una asistencia médica de calidad. En ella se encuentra reflejada la relación existente entre el médico y el paciente, así como también todo el servicio médico que es prestado a un paciente en una circunstancia específica. Su estudio y valoración pueden hacer de ella un instrumento que permitirá evaluar la calidad asistencial que es prestada a los pacientes.
- **Epidemiología:** Los datos e información contenidos en las historias médicas pueden servir de base para estudios realizados por juntas médicas, entes gubernamentales en el área de salud y cualquier organismo calificado para el estudio y análisis de los mismos.
- **Administrativa:** La historia médica es un elemento primordial para el control y gestión de todos los servicios médicos que prestan los centros de salud.
- **Médico-Legal:** Este es uno de los puntos más importantes y delicados al hablar de la historia médica, ya que en el marco de la ley existe la obligación de realizarla para poder usarla en casos de:
  - **Responsabilidad médica profesional como elemento de prueba**, al convertirse por orden judicial en la prueba material principal de todos los procesos de responsabilidad

profesional médica, constituyendo un documento médico-legal fundamental y de primer orden. En tales circunstancias, la historia médica es el elemento que permite la evaluación de la calidad asistencial tanto para la valoración de la conducta del médico como para verificar si cumplió con el deber de informar, de realizar la historia clínica de forma adecuada y eficaz para su finalidad asistencial, puesto que el incumplimiento de tales deberes también constituyen causa de responsabilidad profesional.

- **Declaraciones sobre actos clínicos y conducta profesional como testimonio documental** de ratificación o veracidad de dichos actos.
- **Elaboración de informes médico-legales sobre responsabilidad médica profesional como instrumento de dictamen pericial.** El objetivo de estudio en los informes periciales sobre la responsabilidad médica profesional es la historia clínica, a través de la cual se valoran aspectos como: enumeración de todos los documentos que la integran, reconstrucción de la historia clínica, análisis de cada uno de los procedimientos médicos realizados en el paciente, personal que intervino durante el proceso asistencial, etc.

En el caso de no realizar historia médica o no cumplir debidamente con los requerimientos de la misma, el médico o centro de salud puede acarrear con alguna de las siguientes repercusiones:

- Negligencia médico-asistencial por incumplimiento de la normativa legal.
- Posible riesgo de responsabilidad por perjuicios al paciente y a la institución.
- Riesgo médico-legal objetivo por carencia del elemento de prueba fundamental en reclamaciones por mal praxis o negligencia médica.

## **2.6 PRIVACIDAD Y CONFIDENCIALIDAD DE LA HISTORIA MÉDICA**

Todo el personal médico que directamente trata con los datos personales del paciente que están contenidos en la historia médica o, tienen alguna relación laboral con ellos, están obligados a mantener el secreto y la confidencialidad de la información manipulada.

Mantener la confidencialidad y la discreción en la manipulación de los datos que contiene la historia médica son aspectos obligados por el código deontológico de la medicina y también por la legislación en materia de protección de datos y la legislación penal.

La confidencialidad de los datos en la historia médica no es sólo responsabilidad del médico tratante del paciente, sino también es obligación de los facultativos, enfermeros, fisioterapeutas, terapeutas ocupacionales, podólogos, asistentes personales, personal administrativo y cualquier otro tipo de personal que tenga acceso a la historia médica de un paciente por cualquier motivo laboral.

La privacidad y confidencialidad de los datos suministrados por el paciente deben estar custodiados de forma adecuada y permaneciendo accesible únicamente a personal autorizado a su uso. Todos los aspectos relacionados con estos dos conceptos deben ser controlados desde el inicio y creación de la historia médica, en la anamnesis y la exploración física, hasta la realización de los diversos exámenes médicos, información a los familiares del paciente, reuniones médicas acerca del caso, etc.

## 2.7 GESTIÓN Y ALMACENAMIENTO DE HISTORIAS MÉDICAS FÍSICAS

Como se comentó en párrafos anteriores, la historia médica es un expediente que reúne la información del caso clínico de un paciente. Cuando un paciente asiste a un centro de salud a consultarse pueden presentarse algunos de los siguientes casos:

- **Paciente por primera vez:** En este caso se abre la historia médica, el paciente típicamente llena una ficha manualmente en la cual se encuentran datos básicos del mismo y los antecedentes médicos. Posteriormente, los profesionales de la salud se encargan de completar la información relativa al caso y la almacenan en una carpeta identificada con los datos del paciente.
- **Paciente recurrente:** Cuando el paciente acude por segunda vez o regularmente al centro de salud. El proceso se inicia con la búsqueda de la historia en los archivos del centro de salud, la misma se realiza generalmente por apellido, cédula de identidad o incluso el año de la primera consulta. El proceso continúa con la actualización de la historia médica según sea el motivo de la visita a la Institución.

La cantidad de pacientes que puede atender un médico mensualmente, nos puede dar una idea del gran tamaño que pueden llegar a tener estos archivos físicos a lo largo del tiempo. En un centro de salud donde operan numerosos médicos de diferentes especialidades, la cantidad de expedientes puede llegar a ser inmanejable, ocupando grandes espacios físicos y haciendo necesaria la participación de una cantidad importante de personas para su gestión.

Muchos centros hospitalarios y grupos médicos, se han visto en la necesidad de tener archivos externos, ubicados en un lugar distinto en el que operan, en los cuales almacenan las historias. Generalmente se realiza la separación de las historias según la frecuencia de uso, esto permite mantener en el lugar de operación todas aquellas historias médicas cuyo uso es más frecuente, y almacenar incluso en otra localidad, las historias médicas que sobrepasen un lapso de tiempo estipulado, ya que por lo general son pacientes que probablemente no volverán a consultarse.

Para Muniagurria (1998), entre los métodos que existen para el almacenamiento de historias médicas se encuentran:

- El correlativo, el cual consiste en ordenar las historias por números correlativos empezando desde el uno.
- Por dígito terminal, en el cual podemos organizar las historias a partir de algún dato del paciente, como por ejemplo la fecha de nacimiento, apellido, etc.
- Por valor posicional dinámico asistido por computadora, en el cual el almacenamiento de la historia es dirigido por la computadora según ciertos datos suministrados.

Cualquiera de los métodos empleados por los centros de salud o consultorios médicos deben garantizar la unicidad de la historia médica, así como también la confidencialidad de los datos contenidos en la misma, procurando que su acceso solo sea delegado a personal autorizado.



## **CAPÍTULO 3. MARCO METODOLÓGICO**

En este capítulo se presenta las técnicas y métodos utilizados para la construcción del prototipo de sistema, como lo es la Programación Extrema como método ágil de desarrollo. Además se describen las herramientas tecnológicas que se requieren para la su implementación, como el lenguaje de programación Java JEE, el entorno de desarrollo Eclipse, el sistema manejador de bases de datos PostgreSQL, entre otros.

### **3.1 MÉTODO ÁGIL**

Para Amaro y Valverde (2007), un Método Ágil es un marco de trabajo conceptual de la ingeniería de software que produce iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto. La mayoría de los métodos de desarrollo Ágil minimizan riesgos, desarrollando software en cortos lapsos de tiempo. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación.

La razón principal por la cual se decidió utilizar un método ágil en lugar de un método tradicional fue que el gerente del centro de salud que sirvió como caso de estudio, no tenía bien definidos los requisitos del prototipo de sistema, aunado a ello, se ha trabajado en otros proyectos con este cliente y se ha concluido que es común que existan continuos cambios a lo largo de la fase de desarrollo de los proyectos. Adicionalmente podemos señalar que existen otras características importantes que hacen atractivo el uso de un método ágil para el desarrollo de este prototipo de sistema, como lo son:

- Capacidad de respuesta a cambios de requisitos a lo largo del desarrollo.
- Entrega continua y en plazos breves de software funcional.
- Trabajo conjunto entre el cliente y el equipo de desarrollo.
- Importancia de la simplicidad, eliminado el trabajo innecesario.
- Atención continua a la excelencia técnica y al buen diseño.
- Mejora continua de los procesos y el equipo de desarrollo.

En el siguiente punto se describe el método ágil que se seleccionó para el desarrollo del prototipo de sistema, "Programación Extrema" o mejor conocida por sus siglas en inglés "XP" (eXtreme Programming).

### 3.2 PROGRAMACIÓN EXTREMA (XP)

Según Robles (2008), la programación extrema es un método ágil cuyo objetivo es potenciar las relaciones existentes entre cliente y desarrollador 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. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se puede decir que este método es indicado para proyectos con requisitos indefinidos y cambiantes en el tiempo, y donde existe un alto riesgo técnico.

Este modelo de programación se basa en una serie de metodologías de desarrollo de software en la que se da prioridad a los trabajos que dan un resultado directo y que reducen la burocracia que hay alrededor de la programación.

XP fue creada para el desarrollo de aplicaciones donde el cliente no tiene claro el panorama de lo que desea, razón por la cual hay un cambio constante en los requisitos que debe cumplir la aplicación. Este tipo de método es el preciso para este tipo de casos ya que se adapta a las necesidades del cliente y permite que la aplicación se vaya reevaluando en períodos cortos de tiempo.

Los creadores de XP tenían como principal objetivo desarrollar un método que hiciera que los desarrollos fueran más sencillos permitiendo al programador desarrollar más rápido, reduciendo la cantidad de entregables previos al desarrollo a los que estaban acostumbrados a usar en otros métodos.

En resumen se puede decir que XP está diseñada para el desarrollo de aplicaciones que requieran un grupo de programadores pequeño, donde la comunicación sea más factible que en grupos de desarrollo grandes. La comunicación es un punto importante y debe realizarse entre los programadores, los jefes de proyecto y los clientes.

#### 3.2.1 OBJETIVOS DE LA PROGRAMACIÓN EXTREMA (XP)

Los objetivos de programación extrema son muy simples y podemos resumirlos en los siguientes puntos:

- **La satisfacción del cliente:** La comunicación e interacción con el cliente a lo largo del proceso de desarrollo hace que este método responda efectivamente a las necesidades del cliente, acercando mucho el resultado final del desarrollo a la idea que tenía el cliente sobre la aplicación que deseaba.

- **Potenciar al máximo el trabajo en grupo:** El equipo de desarrollo no solo está integrado por los programadores de la aplicación, también están involucrados los jefes de proyecto y el cliente final, los cuales son parte de este equipo y cada uno juega un papel fundamental para obtener los resultados deseados.

### 3.2.2 PRINCIPIOS BÁSICOS DE LA PROGRAMACIÓN EXTREMA (XP)

Robles (2008), expone que la Programación Extrema se basa en 12 principios básicos según lo proponen sus autores, a saber:

- **El principio de pruebas:** Se tiene que establecer un período de pruebas de aceptación del programa donde se definen las entradas al sistema y los resultados esperados. Es muy recomendable automatizar estas pruebas para poder hacer varias simulaciones del sistema en funcionamiento. Para hacer estas simulaciones, se pueden utilizar Ambientes de Prueba.
- **Proceso de planificación:** En esta fase, el usuario debe escribir sus necesidades, definiendo las actividades que realizará el sistema. Se crea entonces un documento llamado "Historias del Usuario".
- **El cliente en el sitio:** Se le da el poder para determinar los requerimientos, definir la funcionalidad, señalar las prioridades y responder las preguntas de los programadores. Esta fuerte interacción directa con el desarrollador disminuye el tiempo de comunicación y la cantidad de documentación, junto con los altos costes de su creación y mantenimiento. Este representante del cliente debe estar con el equipo de trabajo durante toda la realización del proyecto.
- **Programación en parejas:** Requiere que todos los programadores XP escriban su código en parejas, compartiendo un solo computador. De acuerdo con los experimentos, este principio puede producir mejores aplicaciones, de manera consistente, a iguales o menores costos. Aunque la programación en pareja puede no ser para todo el mundo, las estadísticas del uso de XP demuestran un gran éxito.
- **Integración continua:** Permite al equipo hacer un rápido progreso, implementando las nuevas características del software. En lugar de crear "builds" (o versiones) estables de acuerdo a un cronograma establecido, los equipos de programadores XP pueden reunir su código y reconstruir el sistema varias veces al día. Esto reduce los problemas de integración comunes en proyectos largos y estilo cascada.

- **Refactorización:** Permite a los equipos de programadores XP mejorar el diseño del sistema a través de todo el proceso de desarrollo. Los programadores evalúan continuamente el diseño y recodifican lo necesario. La finalidad es mantener un sistema enfocado a proveer el valor de negocio mediante la minimización del código duplicado y/o ineficiente.
- **Entregas pequeñas:** Colocar un sistema sencillo en producción rápidamente que se actualiza de forma expedita y constante permitiendo que el verdadero valor de negocio del producto sea evaluado en un ambiente real. Estas entregas no pueden pasar las 2 o 3 semanas como máximo.
- **Diseño simple:** Se basa en la filosofía de que el mayor valor de negocio es entregado por el programa más sencillo que cumpla los requerimientos. El Diseño Simple se enfoca en proporcionar un sistema que cubra las necesidades inmediatas del cliente, ni más ni menos. Este proceso permite eliminar redundancias y rejuvenecer los diseños obsoletos de forma sencilla.
- **Metáfora:** Es una descripción general del sistema, que se establece al comenzar el proyecto, que fortifica la integridad conceptual, ayuda a guiar el proceso de desarrollo y mantiene una visión unificada entre los actores. La Metáfora determina un estándar en el vocabulario que será utilizado por los programadores y el cliente, que luego ayudará a establecer las clases y métodos del sistema.
- **Propiedad colectiva del código:** Un código con propiedad compartida. Nadie es el propietario de nada, todos son el propietario de todo. Este método difiere en mucho a los métodos tradicionales en los que un simple programador posee un conjunto de código. Los defensores de XP argumentan que mientras haya más gente trabajando en una pieza, menos errores aparecerán.
- **Estándar de codificación:** Define la propiedad del código compartido así como las reglas para escribir y documentar el código y la comunicación entre diferentes piezas de código desarrolladas por diferentes equipos. Los programadores las han de seguir de tal manera que el código en el sistema se vea como si hubiera estado escrito por una sola persona.
- **La semana de 40 horas:** la programación extrema sostiene que los programadores cansados escriben código de menor calidad. Minimizar las horas extras y mantener los programadores frescos, generará código de mayor calidad.

### 3.2.3 ACTIVIDADES DE LA PROGRAMACIÓN EXTREMA (XP)

La Programación Extrema está compuesta, según Robles (2008), por una serie de actividades o prácticas que pueden agruparse en cuatro grandes bloques o fases.

- **Planificación:**

Se escriben las mencionadas Historias de Usuario, las cuales deben ser hechas por el cliente, de tal manera, que entenderá perfectamente cada una de ellas y se podrá hacer una estimación de tiempo para el desarrollo del mismo.

Al reunir suficientes historias, se forma el llamado “Plan de Liberación”, el cual define de forma específica los tiempos de entrega de la aplicación, para recibir retroalimentación por parte del usuario. Por regla general, cada historia suele necesitar de una a tres semanas de desarrollo. Son muy importantes y tienen que ser una constante las reuniones periódicas durante esta fase de planificación. Pueden ser a diario, con todo el equipo de desarrollo para identificar problemas, proponer soluciones y señalar aquellos puntos a los que se les ha de dar más importancia por su dificultad o por su punto crítico.

- **Diseño:**

Es en esta fase donde se pone en práctica el principio del Diseño Simple. Si alguna parte del sistema es de desarrollo complejo, lo apropiado es dividirla en varias. Si hay fallos en el diseño, deben ser corregidos lo antes posible.

Se crea la Metáfora del sistema, que se basa en un gráfico que muestre su arquitectura y una descripción de las funcionalidades, de manera que pueda ser comprendida por cualquier persona.

Se escriben las tarjetas CRC (Clase, Responsabilidad y Colaboración) que ayudan al equipo a definir actividades durante el diseño del sistema. Cada tarjeta representa una clase en la programación orientada a objetos y define sus responsabilidades (lo que ha de hacer) y las colaboraciones con las otras clases (cómo se comunica con ellas).

La Refactorización entra también en esta fase, no se debe titubear al momento de cambiar un diseño si se considera necesario, o en eliminar un código que ya no se necesite. El propósito es minimizar en lo posible, ineficiencias del código o duplicados del mismo.

- **Codificación:**

El equipo de trabajo procede a plasmar las ideas a través del código, y el representante del cliente debe mantenerse presente en todo el desarrollo del proyecto, para escribir las Historias de Usuario, participar en la elección de las que entrarán en el Plan de Liberación, y hacer las pruebas de funcionalidad. La idea es usar el tiempo del cliente para hacer estas tareas, en vez de para crear una minuciosa y detallada descripción de requisitos.

Debe regirse bajo el principio del Estándar de Codificación mencionado anteriormente, para asegurar la consistencia y facilitar la comprensión y refactorización del código. Es en esta fase donde se pone en práctica la Integración Continua de los últimos avances del proyecto, las Entregas Pequeñas al cliente, la propiedad colectiva del código y la Programación en Parejas, para asegurar una alta calidad del trabajo.

- **Pruebas:**

Todo el código debe tener pruebas unitarias, y deben ser aprobadas antes de ser lanzado. Cuando se encuentra un error de codificación, se desarrollan pruebas para evitar volver a caer en el mismo.

Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las Historias de Usuario elegidas para la iteración, y el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se pasa la prueba de aceptación, se considera que la historia correspondiente se ha completado.

### **3.2.4 HISTORIAS DE USUARIOS**

Las historias de usuario son la forma de representar las necesidades del cliente que deben ser cubiertas con el desarrollo del sistema. El método XP no propone un estándar para elaborar estas historias de usuario, por el contrario, es el cliente en conjunto con el desarrollador del software los que marcan la pauta. En estas historias de usuario se pueden incorporar diversos elementos como tablas, dibujos a mano alzada, notas, etc. Cabe destacar que estas historias de usuario son el producto de mesas de trabajo entre el cliente y el programador, entrevistas a los usuarios finales, entre otros.

### **3.2.5 VENTAJAS DE LA PROGRAMACIÓN EXTREMA (XP)**

Entre algunas de las ventajas que nos brinda el método de programación XP podemos mencionar las siguientes:

- Puede ser implementado en forma parcial (elegir sólo algunas de las prácticas)
- Puede ser implementado en forma gradual
- Puede adaptarse a las necesidades de cualquier equipo de desarrollo. De hecho, Kent Beck recomienda a los equipos que lo adapten a sus necesidades.
- Exige que se establezca una comunicación más fluida con el cliente y que este tenga mayor participación en el proceso de desarrollo. La consecuencia de esto es que el cliente se involucre más en el desarrollo del producto.
- Actualmente es el método ágil más extendida y documentada
- Se realizan pruebas constantemente del sistema.

### **3.2.6 DESVENTAJAS DE LA PROGRAMACIÓN EXTREMA (XP)**

Como todo método de desarrollo, la programación extrema tiene también sus puntos débiles, los cuales debemos tener en cuenta al momento de seleccionarla para realizar algún desarrollo. Entre las desventajas podemos mencionar:

- XP no es escalable a equipos de muchos desarrolladores, como mucho se puede tener un máximo entre 10 y 15.
- Es una método nuevo y no está ampliamente probada, sin embargo muchos desarrolladores defienden los beneficios que genera desarrollar bajo esta método.
- Requiere un equipo de programadores altamente especializados y con una amplia experiencia.
- Requiere alto compromiso del equipo de desarrollo, lo cual a veces es difícil debido a la precaria situación contractual de la gente que trabaja en consultoras.

### **3.3 PATRÓN DE DISEÑO MODELO – VISTA – CONTROLADOR (MVC)**

Según Caraballo (2009), es muy común encontrar aplicaciones creadas en JEE bajo el patrón arquitectónico MVC. Este patrón se caracteriza por separar los datos de una aplicación, la interfaz de usuario, y la lógica de negocios en tres componentes distintos, los cuales pueden ser descritos de la siguiente manera:

- **Modelo:** Se refiere a la representación específica de la información con la cual el sistema funciona. Básicamente contiene la lógica de negocio real, el dominio de la aplicación, y los Objetos de Acceso a Datos (DAO – *Data Access Objects*) que implementen operaciones como: crear, leer, actualizar, etc.

- **Vista:** Esta es la capa responsable de la lógica de presentación y captura de datos de nuestro sistema al exterior y viceversa.
- **Controlador:** Se encarga de realizar cálculos y une todas las partes o piezas. Recibe eventos del exterior, e interactúa con el modelo y actualiza la información de las vistas.

En la siguiente imagen (Ver imagen 2) podemos observar la interacción de los tres componentes descritos anteriormente.

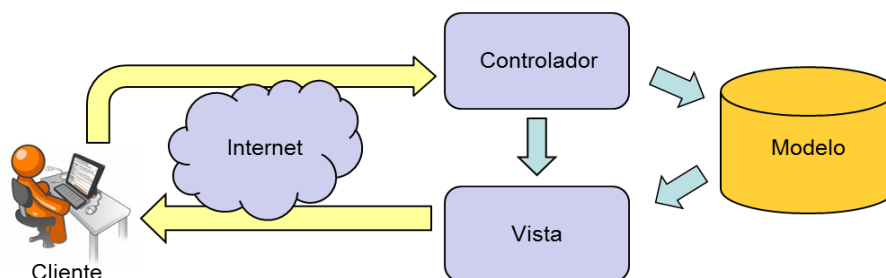


Imagen 2: Interacción de los componentes del Modelo - Vista - Controlador

### 3.4 JAVA ENTERPRISE EDITION (JEE)

JEE es una plataforma abierta y estándar diseñada por la reconocida empresa *Sun Microsystems*, que permite la construcción de aplicaciones empresariales y surge de la necesidad del mercado de desarrollo de software, respecto a contar con medios y herramientas que permitan construir aplicaciones para este rubro.

Esta plataforma de desarrollo busca simplificar las aplicaciones empresariales basándolas en componentes modulares y estandarizados, brindando un completo conjunto de servicios a estos componentes, y manejando muchas de las funciones de la aplicación de manera automática, pudiendo realizar todo esto sin necesidad de recurrir a una programación muy compleja.

Se le denomina plataforma porque proporciona especificaciones técnicas que describen el lenguaje pero, además, provee las herramientas para implementar productos de *software* (aplicaciones) basados en dichas especificaciones.

Con todo lo anteriormente descrito podemos concluir que JEE no es solo una tecnología para el desarrollo de aplicaciones empleando lenguaje Java, sino también un estándar de desarrollo, construcción y despliegue de aplicaciones.



### 3.4.1 ARQUITECTURA DE JEE

Según Lago (2007), la plataforma JEE define un modelo de programación encaminado a la creación de aplicaciones basadas en n-capas. Típicamente una aplicación puede tener cinco capas diferentes:

- **Capa de cliente:** Representa el interfaz de usuario que maneja el cliente. Esta puede admitir numerosos clientes como por ejemplo: HTML (*HiperText Markup Language* – Lenguaje de marcado de hipertexto), *Applet*, aplicaciones Java, etc.
- **Capa de presentación:** Es la representación de todo el conjunto de componentes que generan la información que se presentará en el interfaz de usuario del cliente. Generalmente es creada mediante el uso de componentes basados en *Servlets* y JSP (*JavaServer Pages* – Páginas de servidor Java).
- **Capa de lógica de negocio:** Contiene los componentes de negocio reutilizables. Normalmente se forma a partir de componentes EJB (*Enterprise JavaBeans*).
- **Capa de integración:** En esta capa se pueden encontrar los componentes que permiten hacer más transparente el acceso a la capa de sistemas de información. Por ejemplo este es el lugar ideal para implementar una lógica de Objetos de Acceso a Datos (DAO – *Data Access Objects*).
- **Capa de sistemas de información:** Esta capa engloba a nuestros sistemas de información: bases de datos relacionales, bases de datos orientadas a objetos, etc.

Las ventajas de usar este modelo son amplias, ya que al tener capas separadas existe poco acoplamiento entre las mismas, de manera que es mucho más sencillo hacer modificaciones en ellas sin necesidad que interfieran en las demás.

Todo esto redundará en la obtención de mejoras en cuanto a mantenibilidad, extensibilidad y reutilización de componentes. Otra de las ventajas que obtenemos es que se promueve la heterogeneidad de los clientes, ya que añadir nuevos tipos de cliente (móviles, PCs, etc.) se reduce a incorporar nuevas capas de interfaz de usuario y presentación, sin tener que modificar todo el resto de capas.

A continuación (Ver imagen 3) se muestra de manera gráfica un posible esquema de arquitectura JEE de 3 o 4 capas, aplicación - lógica de negocio - sistemas de información y aplicación - lógica de presentación - lógica de negocio - sistemas de información respectivamente.

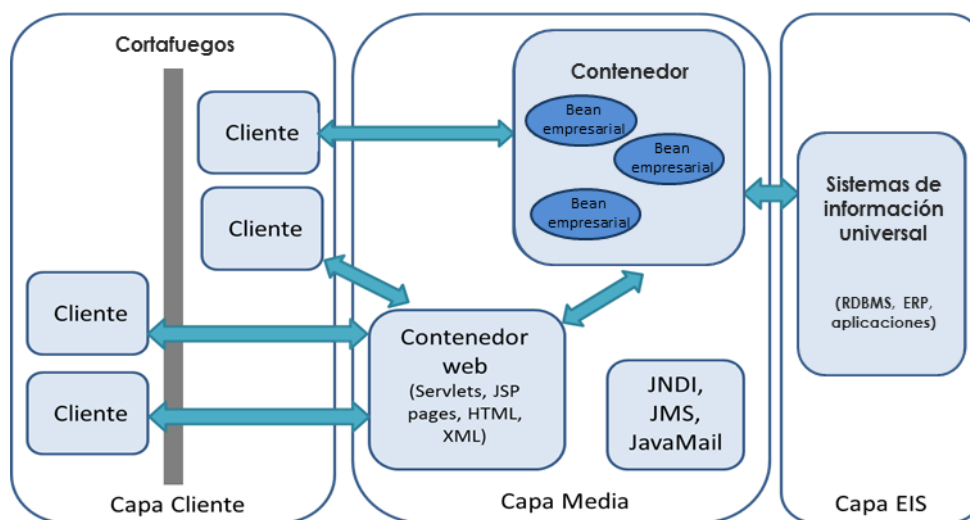


Imagen 3: Esquema de un modelo de 3 o 4 capas <sup>1</sup>

Fuente: Lago, 2007

Como ya se ha hablado a lo largo de este punto, el modelo de desarrollo con JEE está basado en componentes reutilizables, con el objetivo de aumentar la reusabilidad de las aplicaciones. Estos componentes, además, gracias a las especificaciones, son intercambiables entre servidores de aplicaciones, por lo que la portabilidad de nuestros desarrollos es máxima.

### 3.4.2 VENTAJAS DEL USO DE JEE FRENTE A OTRAS PLATAFORMAS DE DESARROLLO DE APLICACIONES EMPRESARIALES

En el mercado actual JEE no es la única plataforma para desarrollar aplicaciones empresariales, existen actualmente varias alternativas, como por ejemplo .Net. La diferencia radica en las ventajas y beneficios que puede brindar JEE frente a las demás arquitecturas de programación similares existentes en el mercado. En esta sección podremos observar las características más importantes que hacen del uso de JEE una ventaja al momento de desarrollar una aplicación.

<sup>1</sup> JSP: JavaServer pages / HTML: HyperText Markup Language (lenguaje de marcado de hipertexto) / XML: eXtensible Markup Language (lenguaje de marcas extensible) / JNDI: Java Naming and Directory Interface (Interfaz de Nombrado y Directorio Java) / JMS: Java Message Service (Servicio de mensajes Java) / RDBMS: Relational Database Management System (Sistema de Gestión de Base de Datos Relacional) / ERP: Enterprise resource planning (Sistemas de planificación de recursos empresariales) / EIS: Enterprise information system (Sistemas de información empresarial)

- **Soporte de múltiples sistemas operativos:** Al ser una plataforma basada en el lenguaje de programación Java, es completamente viable desarrollar aplicaciones basadas en JEE utilizando cualquier sistema operativo donde se pueda ejecutar una máquina virtual Java.
- **Organismo de control:** La plataforma JEE está controlada por un organismo llamado JCP (*Java Community Process* – Proceso de la comunidad Java) formado por más de 400 empresas, asociaciones y particulares cuyo objetivo es asegurar la evolución de las plataformas basadas en Java. Entre las empresas que lo conforman están todas las más importantes del mundo informático (SUN, IBM, Oracle, SAP, HP, AOL, etc.) lo que garantiza que no es una plataforma que nació para evolucionar y mejorar en el tiempo.
- **Competitividad:** es una plataforma de desarrollo de aplicaciones popularmente usada por empresas, las cuales crean soluciones basadas en JEE y ofrecen características como rendimiento, precio, etc., muy diferentes. De esta manera el cliente final tiene una amplia cantidad de opciones a elegir.
- **Madurez:** JEE tiene ya más de cinco años de vida y una gran cantidad de proyectos importantes a sus espaldas, lo que brinda al arquitecto la solidez de trabajar con una plataforma ampliamente probada y aceptada.
- **Soluciones libres:** En la plataforma JEE es posible crear aplicaciones completas basadas única y exclusivamente en productos de software libre, además de la disposición de varias soluciones libres para cada una de las partes de su arquitectura.
- **APIs (*Application Programming Interface* – Interfaz de programación de aplicaciones) bien documentadas:** JEE posee un conjunto numeroso de APIs que podemos usar para desarrollar aplicaciones, las cuales están bien definidas y documentadas, lo cual ahorrara mucho tiempo al momento de emplearlas para desarrollar una aplicación.

### 3.5 JSF (*Java Server Faces*)

Esta tecnología de vista es la encargada de gestionar la sección netamente visual de la aplicación web, corresponde a la capa Vista del modelo MVC y tiene las siguientes características:

- Un conjunto de APIs (*Application Programming Interface* – Interfaz de programación de aplicaciones) para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada de datos, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.

- Un conjunto por defecto de componentes para la interfaz de usuario.
- Dos bibliotecas de etiquetas personalizadas para *JavaServer Pages* que permiten expresar una interfaz *JavaServer Faces* dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- *Beans* administrados.

Dadas estas características, y bajo las especificaciones de la JCP (*Java Community Process* – Proceso de la comunidad Java), organismo que centraliza los diversos requerimientos de los usuarios de esta y otras tecnologías, JSF (*JavaServer Faces*) es una tecnología robusta y confiable en el marco de tecnologías de diseño de interfaces gráficas, además bajo el aval de Oracle (anteriormente *Sun Microsystems*) y la experiencia en el desarrollo a lo largo de varios años por parte del equipo de desarrollo, se decidió utilizar esta tecnología para el diseño de interfaces web.

### 3.6 ***RichFaces (Red Hat)***

Esta tecnología basada en JSF, constituye un marco de trabajo basado en un conjunto extensible de los componentes JSF, que incluye un amplio número de componentes reusables y probados, totalmente integrados con tecnología AJAX (*Asynchronous JavaScript And XML* – JavaScript asíncrono y XML) y que reduce las iteraciones normalmente usadas en proyectos AJAX mediante el uso del Ajax4JSF (*Asynchronous JavaScript And XML for JavaServer Faces* - JavaScript asíncrono y XML para JSF).

De la mano de *Red Hat*, al igual que JBoss, estas dos tecnologías juntas permiten el desarrollo de aplicaciones visuales muy ricas, si bien pueden trabajar por separado, JBoss brinda librerías que mejoran la usabilidad de los componentes.

Entre las características más resaltantes de *RichFaces* tenemos:

- Los temas pueden ser fácilmente intercambiables, incluso desde un XML (*Extensible Markup Language* – Lenguaje de marcas extensible).
- Soporta un gran número de navegadores
- Intensificar el conjunto de beneficios de JSF mientras se trabaja con AJAX.
- Añadir funcionalidad AJAX a las actuales aplicaciones JSF.
- Escribir componentes propios con soporte *built-in* para AJAX.
- Disponer de un paquete de recursos con las clases de la aplicación Java. Además de su núcleo, la funcionalidad de AJAX que se incluye en Ajax4jsf proporciona un avanzado apoyo a la gestión

de diferentes recursos: imágenes, código *JavaScript* y hojas de estilo CSS (*Cascading style sheets* – Hojas de estilo en cascada).

- Crear una interfaz de usuario con un *look-and-feel* basado en *skins*. *Ajax4jsf* proporciona una función de “*skinnability*” que permite definir y administrar fácilmente diferentes esquemas de la interfaz de usuario, con la ayuda de los parámetros definidos en el *skin*. Es posible acceder a los parámetros del *skin* desde el código JSP o desde el código Java (por ejemplo, para ajustar sobre la marcha imágenes basadas en el texto de la interfaz de usuario).
- Soporte AJAX para páginas con componentes JSF que tienen comunicación asíncrona y actualizaciones de porciones de página.
- La capacidad de habilitar AJAX a componentes JSF sin cambios en el propio componente.
- Disponer de una arquitectura abierta y soporte para los estándares en la industria permitiendo mezclar componentes que son de distintas librerías.
- Trabajar en el lado del servidor.
- Soportar *Facelets*.
- Total integración con Tomahawk y Trinidad.

### 3.7 Spring

Risberg (2008), plantea que este marco de trabajo, claro sustituto de los *Enterprise JavaBeans* (EJB), ha tenido un éxito rotundo al permitir llevar las mejores prácticas de programación a un terreno sencillo y estructurado, cosa que solo se creía lograr al adoptar EJB (*Enterprise Java Beans*). *Spring* busca la definición y uso de los componentes de una manera organizada, netamente configurable sobre archivos XML (*Extensible Markup Language* – Lenguaje de marcas extensible) y la reusabilidad como bandera.

En la arquitectura de desarrollo utilizada, *Spring* tiene la tarea de definir el alcance e instanciación de los *beans* de negocios que interactúan con la vista y el modelo, lo cual indica que trabaja en la capa de controlador en la arquitectura MVC, si bien esta es apenas una de las tantas características que provee el *Framework* (o Marco de trabajo), no es la menos importante. *Spring* es capaz de integrarse dentro de una aplicación Web con casi cualquier herramienta de terceros, en nuestro caso enlaza JSF y *RichFaces* con *Hibernate* – que veremos más adelante –, así como con cualquier tecnología de *WebServices* la cual es otra de las características que ofrece, con tan sólo una sencilla configuración XML.

Del lado de la vista, *Spring* es el encargado de centralizar las operaciones que serán llamadas desde la vista, abstrayendo a esta, de las tareas realizadas en la lógica de negocios, es decir, la vista hace llamados a la lógica de negocio (a través de *Spring*) de una manera desacoplada, a través de

interfaces, la vista no conoce de la implementación de la lógica de negocios, apenas conoce la firma de los métodos, esto permite incluso que la lógica de negocios sea modificada sin el más mínimo cambio en la vista y viceversa.

Del lado de la interacción con la Base de Datos, *Spring* permite la definición de los niveles de transaccionalidad, bloqueo e instanciación de los *beans* de negocios, de una manera unitaria, y sin acoplamiento entre la vista y el modelado de negocios, lo que permite que estos métodos funcionen de igual forma aun cuando la vista pueda cambiar de manera completa. En pocas palabras, *Spring* desacopla la vista y el modelo de manera que ambas trabajen de manera unitaria, de este modo es posible tener grupos de trabajos especializados en cada capa.

*Spring* es distribuido bajo la licencia *Apache 2.0 License*, la cual es totalmente compatible con licencias de software libre como la GPL (*General Public License* – Licencia pública general).

### **3.8 Hibernate**

Para Crespo (2004), es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

*Hibernate* se encuentra en la capa Modelo del MVC, y entre una de las características más poderosa destaca el hecho de abstraer a la aplicación del motor de base de datos utilizado para obtener los datos e información necesarios para su funcionamiento. *Hibernate* es el encargado de traducir los requerimientos provenientes de la capa de control al lenguaje utilizado por el manejador de base de datos, a través del uso de HQL (*Hibernate Query Language* – Lenguaje de consulta de *Hibernate*) lo cual ofrece la facilidad de utilizar múltiples manejadores de base de datos, según las necesidades del cliente.

Con el uso de *Hibernate* se garantiza que el prototipo es capaz de funcionar con distintos manejadores de base de datos, haciendo cambios mínimos en los archivos de configuración.

*Hibernate* es una herramienta de software libre, distribuido bajo los términos de la licencia GNU LGPL (*Lesser General Public License* – Licencia pública general reducida).

### **3.9 Motor de generación de reportes JasperReport**

Es una librería que permite la creación de informes Java de código abierto, que tiene la habilidad de entregar contenido enriquecido en el monitor, a la impresora o a ficheros PDF (*Portable document format* – Formato de documento portátil), HTML (*HyperText Markup Language* – Lenguaje de marcado de hipertexto), XLS (*Microsoft Excel Spreadsheet*), CSV (*Comma separated values* - Valores separados por coma) y XML (*Extensible markup language* – Lenguaje de marcas extensible). Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones Java, incluyendo JEE o aplicaciones Web, para generar contenido dinámico.

### 3.10 *Spring Security*

Es un marco de trabajo que provee mecanismos de autenticación y autorización además de otros aspectos de seguridad para aplicaciones Web usando el marco de trabajo *Spring*. Este marco (*Spring Security*) llamado inicialmente ACEGI fue incorporado al portafolio de herramientas integrables con Spring y oficial sub-proyecto del mismo.

Para Evans (2008), entre las particularidades de ACEGI tenemos:

- Trabaja con casi cualquier mecanismo de autenticación, ya sea formulario básico de java hasta LDAP (*Lightweight directory Access protocol* – Protocolo ligero de acceso a directorios).
- Responsable de permitir la autorización a los recursos Web dependiendo de los roles y usuarios establecidos en la aplicación Web.
- Fácilmente configurable con archivos XML.
- Totalmente integrado con *Spring*.
- Permite la definición de autorizaciones mediante AOP (*Aspect-oriented programming* – Programación orientada a aspectos).

Además es posible habilitar los protocolos seguros a través de ACEGI con SSL (*Secure sockets layer* – Capa de conexión segura), adicionando parámetros de configuración al Servidor de aplicaciones para indicar los recursos que deben estar bajo el protocolo HTTPS (*Hipertext transfer protocol secure* – Protocolo seguro de transferencia de hipertexto). El certificado de autenticación vendrá dado por el cliente.

Para el desarrollo del prototipo de sistema se utilizó Spring Security (ACEGI) como marco de seguridad para la autenticación y autorización, por su probada confiabilidad.

## **CAPÍTULO 4. MARCO APLICATIVO**

En este capítulo se presenta la descripción del proceso de desarrollo del prototipo de sistema Web, así como la adaptación del método XP (Programación Extrema), donde cabe mencionar algunos principios que no aplican en el presente proyecto, como la Programación en Parejas y el uso de las Tarjetas CRC. Sin embargo, se cumple con los principios restantes, como el uso de las Historias de Usuario; la Metáfora del Sistema con el propósito de describir las funcionalidades de la aplicación a través de una representación gráfica; y otras relevantes que se mencionan a lo largo del presente capítulo.

### **4.1 ADAPTACIÓN DE LA PROGRAMACIÓN EXTREMA (XP) PARA DESARROLLO DEL PROTOTIPO DE SISTEMA**

El método que se aplicó durante el proceso de desarrollo del prototipo de sistema, es una adaptación de la mencionada Programación Extrema (XP), la cual fue seleccionada por las bondades que brinda para este caso particular de desarrollo, como la búsqueda de la simplicidad, las constantes y rápidas respuestas al cliente buscando su mayor satisfacción, la reducción de documentación, el estar orientada a equipos pequeños; entre otras. A continuación se describe cómo se aplicaron las 4 actividades de XP al desarrollo de la aplicación.

#### **4.1.1 PLANIFICACIÓN**

En esta fase del método se buscó en primera instancia, tener una reunión con el gerente del centro de salud, para pedirle que escribiera una lista de Historias de Usuario. Esta primera reunión y las siguientes se trataron directamente con el gerente, ya que es quién conoce todos los procesos de negocio, y fue el responsable de agregar y/o modificar los requerimientos durante el proceso de desarrollo.

Para definir los tiempos de entrega del prototipo se utilizó el mencionado "Plan de Liberación", mediante el cual se pudo obtener la retroalimentación necesaria entre el cliente y el desarrollador, típica al aplicar un método ágil. Para cumplir con el "Plan de Liberación" de esta fase, se definió una bitácora de desarrollo, planificando los requerimientos en base a días. Se trabajó por iteraciones compuestas de historias de usuario, cada una con una duración estimada, una fecha de comienzo y una fecha de fin. En este sentido, el formato de la bitácora que se utilizó es el mostrado en la Tabla 1.



Nombre de tarea	Duración	Comienzo	Fin

Tabla 1: Modelo de tabla para la bitácora de desarrollo

Fuente: Pérez, 2011

Como indica el método, se trabajó en función del tiempo, implementando cada Historia de Usuario en una cantidad de días comprendidos entre 1 y 3 semanas, siguiendo el formato de la Tabla 2.

<b>Número de historia:</b>	<b>Prioridad:</b>	<b>Tiempo estimado:</b>
<b>Nombre de la historia:</b>		
<b>Descripción de la historia:</b>		

Tabla 2: Formato para registrar las Historias de Usuario

Fuente: Pérez, 2011

De acuerdo a los lineamientos establecidos en el método XP anteriormente descritos, se lograron agrupar los requerimientos del cliente en un conjunto de iteraciones. Cabe destacar, que en cada iteración se incluyeron las fases de: planificación, diseño, codificación y pruebas. La sumatoria de todas las iteraciones reúnen todos los requerimientos del cliente.

En el siguiente listado se mencionan las historias de usuario más importantes en base a su prioridad:

- Desarrollo del módulo de acceso al prototipo sistema y menú principal.
- Desarrollo del módulo de seguridad.
- Desarrollo del módulo de gestión de almacenes.
- Desarrollo del módulo de creación de la portada de una historia médica física.
- Desarrollo del módulo de registro de entradas de historias médicas en un almacén.
- Desarrollo del módulo de registro de salidas de historias médicas en el almacén.
- Desarrollo del módulo de recepción de peticiones de historias médicas físicas.
- Desarrollo del módulo de registro de solicitudes de historias médicas físicas.
- Desarrollo del módulo de inventario.
- Desarrollo del módulo de reportes de inventario.
- Desarrollo del módulo de reportes asociado con la salida de historias del almacén, "orden de salida".
- Desarrollo de un módulo de control de historias médicas en préstamo.
- Desarrollo de un módulo de control de historias médicas en tránsito.

### 4.1.2 DISEÑO

En esta etapa del método, se analizó el prototipo de sistema globalmente, pudiendo de esta manera crear una sencilla representación de las partes que la conforman y cómo se comunicarán entre ellas. Esto cumple con el principio de XP, de escribir una metáfora que represente de forma general, el resultado que se busca en el desarrollo del prototipo de sistema.

Para cumplir con el principio del diseño simple, se realizaron reuniones con el cliente en las cuales se propusieron soluciones entre ambas partes y se definieron las funcionalidades de cada módulo del prototipo de sistema, permitiendo de esta manera mantener todo lo más sencillo posible.

Cabe destacar que la programación y el desarrollo de cada uno de los módulos estuvieron a cargo de una sola persona, razón por la cual se descartó el uso de las tarjetas CRC (Clases, Responsabilidades y Colaboración), cuyo objetivo no es más que facilitar la comunicación entre los desarrolladores y documentar los resultados, lo cual en este caso en particular no aplica. Sin embargo, se cumplió con el principio de Refactorización, ya que constantemente se revisó el código y la documentación, refinándolos siempre que fuese necesario.

### 4.1.3 CODIFICACIÓN

En esta fase, se definieron los estándares de Codificación para la programación en JEE, implementación del modelo de datos y la plataforma de desarrollo.

En cuanto a la codificación para la programación JEE, el estándar de codificación empleado fue el descrito por Scott Hommel de *Sun Microsystems Inc.* (Ver Anexo I).

En lo que se refiere a la implementación del modelo de datos, se siguieron los siguientes lineamientos:

- Todas las tablas sin excepción, constaron de 5 columnas adicionales para fines de auditoria: **usuario\_creacion** (usuario que crea el registro en la tabla), **fecha\_creacion** (fecha en la que se crea el registro en la tabla), **usuario\_modificacion** (usuario que actualiza la información del registro), **fecha\_modificacion** (fecha en la que se lleva a cabo la actualización del registro) y **version** (se refiere a la versión del registro, este número se incrementa cada vez que un usuario realiza una actualización).
- Todas las tablas sin excepción, usaron como clave primaria una columna llamada **ID**, la cual consiste de un número entero que se incrementa con cada registro nuevo.

#### 4.1.4 PRUEBAS

En esta fase, se realizaron las pruebas del código de cada Historia de Usuario, asegurando de esta forma, el buen funcionamiento de dicha implementación. Al finalizar cada iteración, se realizaron pruebas unitarias, funcionales y de aceptación, escribiendo los casos de prueba, sus resultados esperados, y los obtenidos. Para ello, se utilizó el siguiente formato (Ver tabla 3).

Nº Caso de prueba	Módulo	Caso de prueba	Resultado esperado	Resultado obtenido	Motivo de la falla

Tabla 3: Formato para el registro de pruebas

Fuente: Pérez, 2011

## 4.2 ANÁLISIS GLOBAL DEL PROTOTIPO DE SISTEMA

Cumpliendo con los principios de XP, se realiza un análisis general del prototipo de sistema, levantando los requerimientos necesarios para el desarrollo del mismo. A partir de ello, se obtienen las Historias de Usuario desde la primera reunión con el gerente del centro de salud y se construye la llamada "metáfora del prototipo de sistema" (Ver imagen 4).

### 4.2.1 HISTORIAS DE USUARIO

Como parte del análisis global del prototipo de sistema a desarrollar, se crean las Historias de Usuario que se mencionan a continuación:

<b>Número de historia:</b> 1	<b>Prioridad:</b> ALTA	<b>Tiempo estimado:</b> 6 días
<b>Nombre de la historia:</b> Desarrollo del módulo de acceso al prototipo de sistema y menú principal con opciones a los distintos módulos		
<b>Descripción de la historia:</b> Mediante esta pantalla de acceso, el usuario podrá ingresar al prototipo de sistema con un usuario y contraseña previamente definidos, luego de permitir el acceso, se debe mostrar el menú principal con las opciones correspondientes según los privilegios que posea el usuario que está accediendo al prototipo de sistema. Los botones del menú principal deben ser generosos en cuanto a tamaño, ya que se pretende utilizar la aplicación a desarrollar en computadores con pantalla táctil, así como en tabletas y otros dispositivos móviles.		

<b>Número de historia:</b> 2	<b>Prioridad:</b> ALTA	<b>Tiempo estimado:</b> 4 días
<b>Nombre de la historia:</b> Desarrollo del módulo de seguridad		
<b>Descripción de la historia:</b> Módulo que permitirá la configuración del prototipo de sistema, en el cual se podrán definir los usuarios que darán uso a este, así como también los privilegios de cada uno de ellos. Este módulo estará situado en la opción "Configuración" del menú principal del prototipo de sistema.		

<b>Número de historia:</b> 3	<b>Prioridad:</b> ALTA	<b>Tiempo estimado:</b> 2 días
<b>Nombre de la historia:</b> Desarrollo del módulo de gestión de almacenes		
<b>Descripción de la historia:</b> Mediante este módulo se le permitirá al usuario crear, eliminar, listar y modificar almacenes en el prototipo de sistema. Este módulo estará situado en la opción "Configuración" del menú principal del prototipo de sistema.		

<b>Número de historia:</b> 4	<b>Prioridad:</b> ALTA	<b>Tiempo estimado:</b> 4 días
<b>Nombre de la historia:</b> Creación del módulo de creación de la portada de una historia médica física		
<b>Descripción de la historia:</b> Se refiere a una opción en el menú principal que permita la creación de una portada que identifique cada una de las historias de los pacientes atendidos en el centro de salud. Cada historia médica tendrá asociado un código de barras, el cual deberá ser generado por el prototipo de sistema, a partir del número de cédula de identidad del paciente.		

<b>Número de historia:</b> 5	<b>Prioridad:</b> ALTA	<b>Tiempo estimado:</b> 5 días
<b>Nombre de la historia:</b> Desarrollo del módulo de registro de entradas de historias médicas en un almacén		
<b>Descripción de la historia:</b> Opción del menú principal en la cual se permite registrar el ingreso de una(s) historia(s) medica(s) en un almacén previamente definido. Cabe destacar que para registrar la entrada de una historia, se debe tener previamente impresa la portada de la misma. El registro de la entrada de historias debe realizarse usando un escáner de código de barras.		

<b>Número de historia:</b> 6	<b>Prioridad:</b> ALTA	<b>Tiempo estimado:</b> 5 días
<b>Nombre de la historia:</b> Desarrollo del módulo de registro de salida de historias médicas en el almacén		
<b>Descripción de la historia:</b> Opción del menú principal en la cual se permite registrar la salida de una(s) historia(s) medica(s) de un almacén origen hacia un almacén destino. Cada vez que se registre una salida de una historia o grupo de historias, se debe generar un reporte llamado "Orden de salida" en el cual se especificara el detalle de la(s) historia(s) que van en tránsito.		

<b>Número de historia:</b> 7	<b>Prioridad:</b> MEDIA	<b>Tiempo estimado:</b> 10 días
<b>Nombre de la historia:</b> Desarrollo del módulo de recepción de peticiones de historias médicas físicas		
<b>Descripción de la historia:</b> Se refiere a una opción ubicada en el menú principal, la cual tiene como objetivo gestionar las peticiones de historias médicas que llegan al almacén donde estas se encuentran. Es visto como una bandeja donde llegan las solicitudes de historias que hacen los almacenes a un almacén en particular.		

<b>Número de historia:</b> 8	<b>Prioridad:</b> MEDIA	<b>Tiempo estimado:</b> 10 días
<b>Nombre de la historia:</b> Desarrollo del módulo de registro de solicitudes de historias médicas físicas		
<b>Descripción de la historia:</b> Es una opción en el menú principal que le permite a un usuario solicitar una o varias historias que sean requeridas por un determinado almacén. La solicitud realizada por el usuario, se convertirá automáticamente en una petición para el almacén en donde se encuentre(n) ubicada(s) la(s) historia(s) en cuestión.		

<b>Número de historia:</b> 9	<b>Prioridad:</b> MEDIA	<b>Tiempo estimado:</b> 5 días
<b>Nombre de la historia:</b> Desarrollo del módulo de inventario		
<b>Descripción de la historia:</b> Este módulo tendrá ubicación en el menú principal y como su nombre lo indica permitirá realizar el inventario de las historias médicas físicas existentes en un almacén determinado. Básicamente se refiere a una verificación de las historias médicas físicas existentes, cotejándolas con el listado de historias médicas físicas que se tienen registradas en ese mismo almacén en el prototipo de sistema.		

<b>Número de historia:</b> 10	<b>Prioridad:</b> MEDIA	<b>Tiempo estimado:</b> 3 Días
<b>Nombre de la historia:</b> Desarrollo del módulo de reportes de Inventario		
<b>Descripción de la historia:</b> Este módulo, formará parte del módulo de inventario y proporcionará un informe en formato .pdf al finalizar el proceso de inventario en un determinado almacén. En el reporte de inventario debe quedar impresa la información más relevante del proceso de inventario, como lo es: fecha de realización del inventario, datos básicos del almacén, número de historias existentes en el almacén, historias extraviadas, entre otros.		

<b>Número de historia:</b> 11	<b>Prioridad:</b> MEDIA	<b>Tiempo estimado:</b> 2 días
<b>Nombre de la historia:</b> Desarrollo del módulo de reportes asociados con la salida de historias del almacén, "orden de salida".		
<b>Descripción de la historia:</b> Este módulo formará parte del módulo de registro de salud de historias médicas y debe proporcionar un informe en formato .pdf al finalizar el registro de las salidas... El informe, debe reunir información de las historias que están viajando de un lugar a otro, así como también la fecha en la cual se generó el mismo, el responsable del traslado de las historias, etc.		

Las Historias de Usuario previamente descritas son el resultado del levantamiento de requerimientos que se realizó durante el análisis global del prototipo de sistema.

#### 4.2.2 METÁFORA DEL PROTOTIPO DE SISTEMA

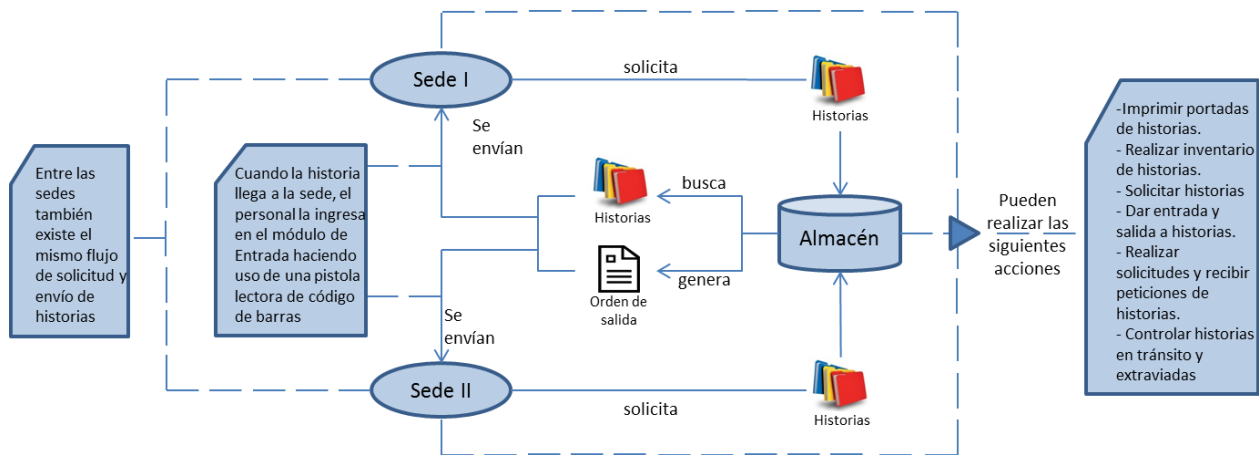


Imagen 4: Metáfora del prototipo de sistema

#### 4.3 ESPECIFICACIONES TÉCNICAS DEL PROTOTIPO DE SISTEMA

Con base en el estudio de las diversas herramientas tecnológicas en el trabajo de seminario, y considerando la necesidad de integración de este prototipo de sistema con un sistema administrativo y de gestión de citas existentes, se utilizaron las siguientes herramientas o programas:

- MVC (Modelo-Vista-Controlador) como patrón de diseño del prototipo.
- Java (JEE) como lenguaje de programación y patrón de diseño.
- Eclipse como entorno de programación.
- JasperReport como motor para la presentación de reportes/informes.
- IReport como entorno para el diseño de reportes/informes.
- RichFaces y JSF en la capa Vista.
- Spring web flow en la capa Control.
- Hibernate para la capa Modelo.
- Tomcat como servidor de la aplicación.
- PostgreSQL como sistema manejador de base de datos.

### 4.3.1 ARQUITECTURA DE DESARROLLO

El objetivo principal de la arquitectura fue separar, de la forma más limpia posible, las distintas capas de desarrollo, con la finalidad de proveer amplias facilidades de mantenimiento y evolución del prototipo de sistema desarrollado. El empleo de esta arquitectura de desarrollo facilitó las labores de despliegue de la aplicación y su funcionamiento se adapta perfectamente al patrón Modelo - Vista - Controlador (MVC) explicado anteriormente.

La arquitectura de desarrollo utilizada para el desarrollo del prototipo de sistema, está basada en las siguientes herramientas que podemos observar en la imagen 5, las cuales están organizadas dentro del patrón de diseño MVC (Modelo - Vista - Controlador):



Imagen 5: Arquitectura propuesta, considerando el patrón MVC

Es importante destacar que uno de los motivos por los cuales se utilizaron estas herramientas de desarrollo, fue porque este prototipo de sistema de inventario y manejo de historias médicas físicas debe integrarse con un sistema administrativo y de gestión de citas ya desarrollados, ambos destinados a satisfacer un requerimiento general del cliente en el centro de salud seleccionado como caso de estudio.

#### **4.4 PLAN DE ITERACIONES**

Debido al método de desarrollo utilizado para el presente Trabajo Especial de Grado, fue necesaria la definición y ejecución de un conjunto de iteraciones, estableciendo fecha de inicio y culminación para cada una de estas. A continuación, se encuentra la descripción de cada una de las iteraciones que fueron necesarias para lograr el desarrollo del prototipo de sistema de inventario de historias médicas físicas.

##### **4.4.1 ITERACIÓN 0**

Esta iteración comprende el diseño y la implantación del modelo de datos que fue utilizado para el desarrollo del prototipo de sistema. Para ello se realizó un análisis exhaustivo de cada uno de los requerimientos del cliente expresados a través de las historias de usuario, de manera tal que se evite la constante modificación del modelo de datos a lo largo del desarrollo del prototipo.

En la imagen 6 se presenta el modelo relacional del prototipo de sistema, en el cual se pueden observar las tablas que fueron necesarias para lograr el desarrollo del prototipo. Debido al tamaño de este modelo, el contenido de cada tabla no fue mostrado en el mismo, sin embargo, el detalle de cada una de las mismas puede ser observado consultando el anexo 2.



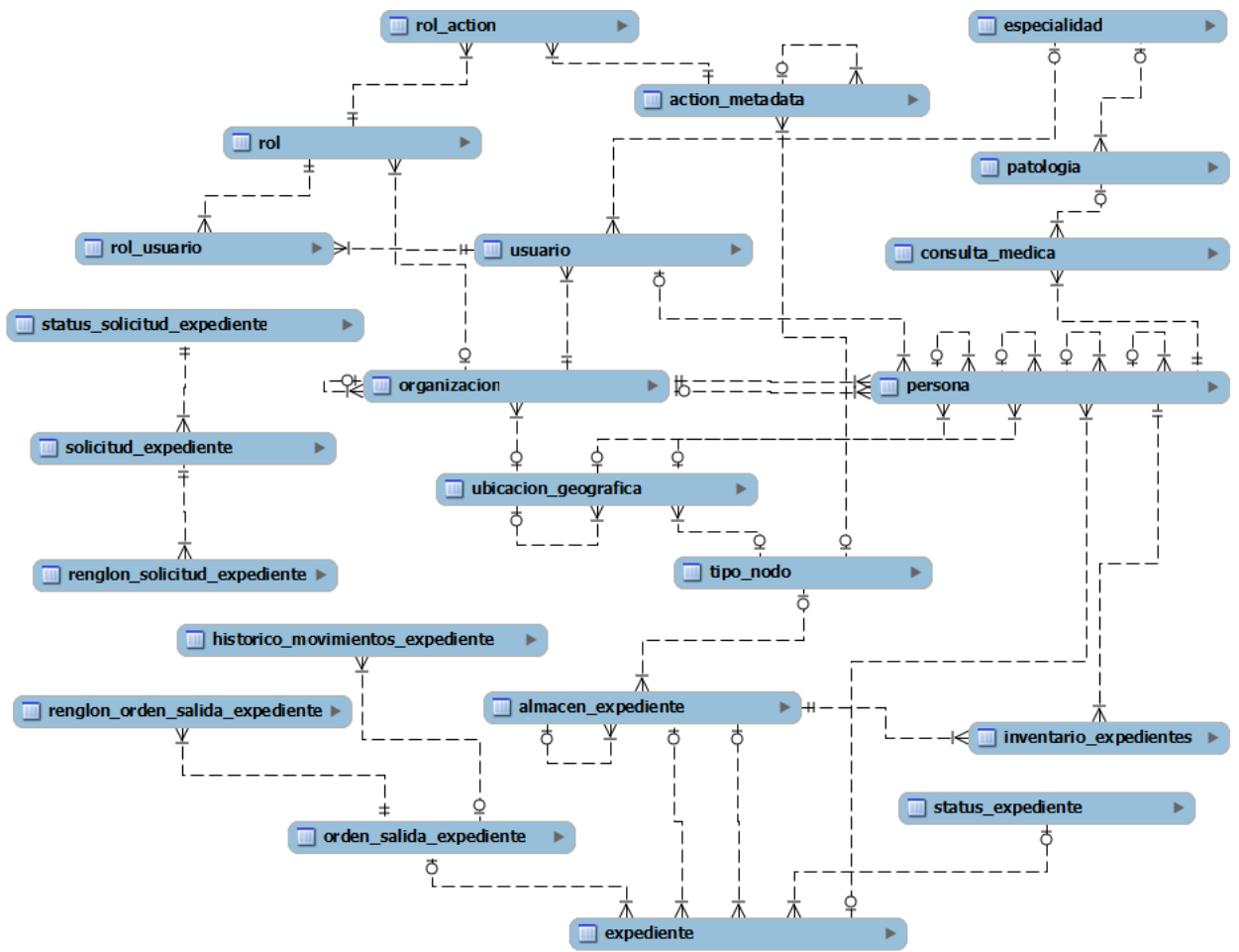


Imagen 6: Modelo relacional del prototipo de sistema

Para lograr integrar este prototipo a un sistema administrativo y de gestión de citas ya existentes, fue necesario la utilización y/o modificación de ciertas tablas ya existentes en el modelo de datos del sistema administrativo y de gestión de citas al cual se pretende integrar este prototipo. Entre algunas de las tablas previamente existentes destacan: persona, action\_metadata y organización.

#### 4.4.2 ITERACIÓN 1

En esta iteración se desarrolló la funcionalidad que permite la definición de usuarios en el prototipo de sistema, así como también los roles que estos usuarios pueden tener dentro del mismo y que definen el acceso a los distintos módulos. Adicionalmente, se contempló el acceso y la autenticación de estos usuarios (según cada uno de sus roles) al prototipo de sistema. Por último, se desarrolló el menú principal del prototipo con cada uno de los botones que permitan realizar las acciones contempladas dentro del diseño del prototipo.

#### 4.4.2.1 Planificación

<b>Nro. de iteración</b>	1
<b>Descripción</b>	Definición de usuarios, roles y asignación de roles a usuarios dentro del prototipo de sistema. Autenticación y acceso al prototipo. Desarrollo del menú principal con las opciones del prototipo de sistema.
<b>Historias de usuario</b>	- Desarrollo de la pantalla de acceso al prototipo de sistema y menú principal con opciones a los distintos módulos. - Desarrollo del módulo de seguridad.
<b>Tiempo estimado</b>	128 horas
<b>Fecha de inicio y fin</b>	12/06/2012 – 03/07/2012

Tabla 4: Planificación de la iteración 1

#### 4.4.2.2 Bitácora de Desarrollo

En esta fase, se incluye la bitácora de desarrollo (ver tabla 5), donde se describe el tiempo estimado y real de las actividades generales que se realizan durante este período.

<b>Nro.</b>	<b>Tarea</b>	<b>Precede</b>	<b>Fecha inicio</b>	<b>Fecha fin</b>	<b>Días estimados</b>
1	Definición de usuarios	-	12/06/2012	14/06/2012	3
2	Definición de roles	-	15/06/2012	19/06/2012	5
3	Asignación de roles a usuario	1,2	20/06/2012	22/06/2012	3
4	Autenticación y acceso de usuarios al prototipo de sistema	1	25/06/2012	27/06/2012	3
5	Diseño del menú principal del prototipo	1	28/06/2012	03/07/2012	4

Tabla 5: Bitácora de desarrollo de la iteración 1

#### 4.4.2.3 Diseño

En esta fase se busca realizar un diseño de interfaz cónsono con la interfaz del sistema administrativo y de gestión de citas existentes y con el cual, se busca integrar este prototipo. Para ello se utilizaron los mismos colores y tipografías, de manera tal que este nuevo prototipo tuviese las mismas características con respecto a la interfaz y la distribución de las opciones que los sistemas existentes, las cuales serán presentadas en dos tipos de menú dependiendo del caso, pudiendo existir

un menú horizontal derecho (generalmente permitirá regresar al menú principal, salir del prototipo o ir al sistema administrativo al cual se pretende integrar) y un menú vertical izquierdo, en el cual se pueden acceder a las opciones específicas que provee cada una de las opciones principales del prototipo.

Dentro de las pantallas que se diseñaron para esta iteración tenemos las siguientes:

- Acceso al prototipo de sistema.
- Definición de usuarios, roles y asignación de roles a los usuarios.
- Menú principal del sistema (Ver Imagen 7).

Como se puede observar en la siguiente imagen (Ver imagen 7), los botones de la aplicación se diseñaron con un tamaño generoso, ya que el cliente sugirió que la aplicación en un futuro sería usada en pantallas táctiles, razón por la cual se debió pensar en la manera de que el usuario final pudiese interactuar correctamente con el prototipo a través de este tipo de dispositivos.

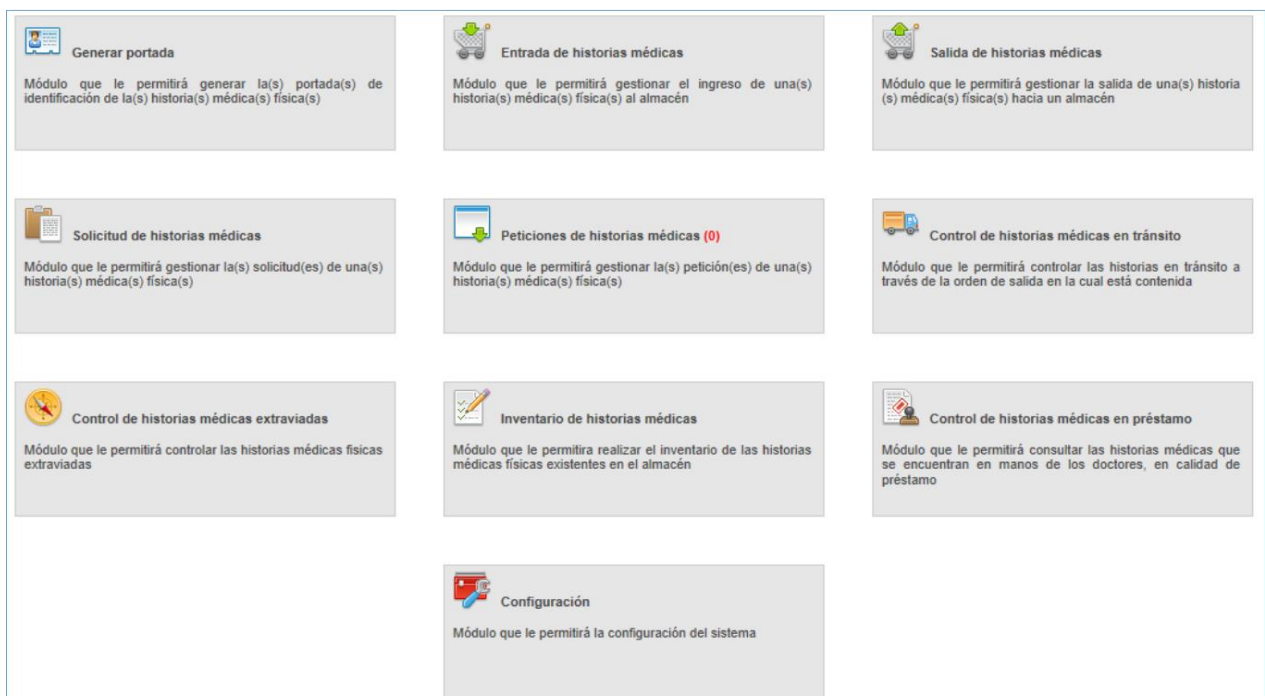


Imagen 7: Menú principal del prototipo de sistema

#### 4.4.2.4 Pruebas

En esta fase de la iteración, se busca probar cada uno de las historias de usuario desarrolladas y contempladas en la misma. Para ello se contó con la presencia de una persona encargada de realizar cada una de las pruebas. Esta persona tenía la profesión de Ingeniera en sistemas del Instituto

Universitario de Tecnología Dr. Federico Rivero Palacio (I.U.T.) y era completamente ajena al desarrollo de este proyecto, todo esto con la finalidad de realizar de manera imparcial y objetiva cada una de las pruebas correspondientes en cada iteración.

En el caso particular de esta iteración, se realizaron pruebas sobre las siguientes funcionalidades: definición de usuarios en el prototipo, creación de roles de usuario y la asignación de estos a usuarios previamente definidos. Una vez probado lo anterior, se prosiguió a probar el acceso al prototipo de sistema de un usuario previamente creado, y por último, se probó el menú principal del prototipo. Las pruebas fueron documentadas en una bitácora, donde se almacenaba información relevante como: número de caso de prueba, caso de prueba, módulo, resultado esperado y resultado obtenido.

Nº Caso de prueba	Módulo	Caso de prueba	Resultado esperado	Resultado obtenido
1	Configuración del prototipo de sistema	Crear un usuario	Creación de un nuevo usuario en el prototipo de sistema con un nombre y clave de acceso al mismo	Se creó correctamente un nuevo usuario en el prototipo de sistema, el cual puede acceder con su nombre y clave de acceso
2	Configuración del prototipo de sistema	Crear un nuevo rol de usuario	Definición de un nuevo rol de usuario dentro del prototipo	Se creó correctamente un nuevo rol dentro del prototipo de sistema
3	Configuración del prototipo de sistema	Asignar de un rol a un usuario	Asignación de un rol definido en el prototipo a un usuario existente en el mismo	Se logró asignar correctamente un rol a un usuario
4	Seguridad	Ingresar al prototipo de sistema con un usuario y clave previamente definidos	Ingresar al prototipo de sistema con un nombre de usuario definidos en el prototipo de sistema	Se logró ingresar al prototipo de sistema con un nombre de usuario y su respectiva clave, los cuales fueron definidos previamente
5	Menú principal del prototipo de sistema	Probar el funcionamiento de cada uno de los botones del menú principal	Acceder a cada una de las opciones mostradas en el menú principal mediante el uso de los botones del prototipo	Se comprobó el funcionamiento de cada uno de los botones del menú principal, excepto el botón "historias médicas en tránsito", al tratar de acceder a esta opción se envía a la página de "historias médicas extraviadas".

Tabla 6: Casos de prueba. Iteración 1

#### 4.4.3 ITERACIÓN 2

Para esta iteración se contempló el desarrollo de la funcionalidad que permite la definición de almacenes y/o sucursales en el prototipo de sistema, así como también la posibilidad de eliminar, modificar o listar cualquiera de los almacenes definidos. También en esta iteración se desarrolló la opción que le permite al usuario la creación e impresión de la portada de una historia médica física. Esta portada cuenta con la incorporación de un código de barras que permite una mejor manipulación de la historia en cada una de las funcionalidades que proveerá el prototipo (a través del uso de un escáner de código de barras).

#### 4.4.3.1 Planificación

<b>Nro. de iteración</b>	2
<b>Descripción</b>	Definición de los almacenes y/o sucursales en el prototipo de sistema, opción que adicionalmente permitirá la eliminación, modificación y el listado de los almacenes/sucursales previamente definidos en el prototipo. Desarrollo de la opción que le permitirá al usuario la creación de las portadas para cada una de las historias médicas físicas
<b>Historias de usuario</b>	- Desarrollo del módulo de gestión de almacenes. - Desarrollo del módulo de Creación de la portada de una historia médica física.
<b>Tiempo estimado</b>	48 horas
<b>Fecha de inicio y fin</b>	04/07/2012 – 11/07/2012

Tabla 7: Planificación de la iteración 2

#### 4.4.3.2 Bitácora de Desarrollo

La siguiente tabla (ver tabla 8), resume la bitácora de desarrollo para esta iteración, en ella se describe el tiempo estimado y real de las actividades generales que se realizan durante este período.

Nro.	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados
6	Definición de almacenes	1	04/07/2012	05/07/2012	2
7	Creación de portadas para historias médicas físicas	1	06/07/2012	11/07/2012	4

Tabla 8: Bitácora de desarrollo de la iteración 2

#### 4.4.3.3 Diseño

En esta fase de la iteración 2, se diseñan las pantallas correspondientes a la gestión de almacenes/sucursales (creación, eliminación, modificación y listado). Por otro lado se realizó el diseño de la portada que llevan todas las historias médicas físicas. Para el diseño de esta portada, se consultó con el cliente la información y la disposición de la misma en la portada, acordando un diseño como se observa en la siguiente imagen (Ver imagen 8).



Imagen 8: Elementos de la portada de la historia médica física

A continuación, se presentan ejemplos de pantallas que permiten la definición y gestión de almacenes en el prototipo de sistema (ver imagen 9 e imagen 10):

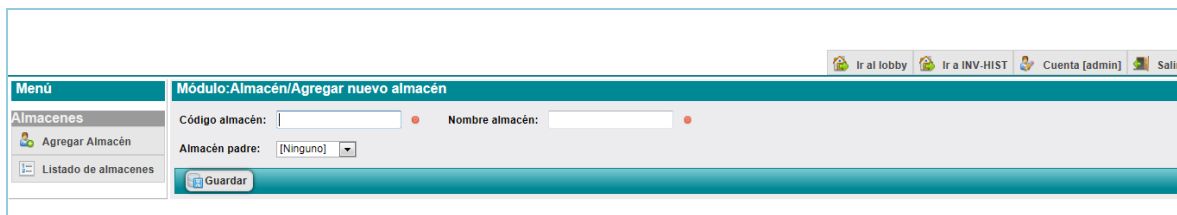


Imagen 9: Pantalla para definir un nuevo almacén en el prototipo



Imagen 10: Listado de almacenes definidos en el prototipo

#### 4.4.3.4 Pruebas

Para la iteración 2, se procedió a probar la correcta definición de un nuevo almacén/sucursal en el prototipo de sistema, así como también cada una de las opciones presentes cuando ya existía un almacén creado, las cuales eran: eliminar un almacén, editar la información de un almacén/sucursal (código o nombre) y por último listar los almacenes/sucursales definidos en el prototipo de sistema.

En cuanto a la creación de la portada, se probó que efectivamente se logra crear una portada para una historia médica específica, comprobando que la información contenida en la misma era correcta y asociada a la historia médica deseada, para de esta manera poder imprimirla satisfactoriamente.

La siguiente tabla (ver tabla 9), resume las pruebas realizadas y los resultados obtenidos:

Nº Caso de prueba	Módulo	Caso de prueba	Resultado esperado	Resultado obtenido
6	Gestión de almacenes	Definir un nuevo almacén/sucursal	Creación de un nuevo almacén/sucursal en el prototipo de sistema	Se creó un nuevo almacén/sucursal para realizar la gestión de las historias médicas físicas
7	Gestión de almacenes	Editar un almacén	Modificación de la información de un almacén existente en el prototipo de sistema	Se realizó correctamente la modificación de la información de los almacenes/sucursales definidos en el prototipo
8	Gestión de almacenes	Eliminar un almacén	Eliminación de un almacén/sucursal existente en el prototipo	Al intentar eliminar un almacén definido en el prototipo ocurría error 404

9	Gestión de almacenes	Listar almacenes /sucursales	Generar un listado con los almacenes existentes en el prototipo	Se pudo obtener un listado de todos los almacenes/sucursales definidos en el prototipo de sistema
10	Creación de la portada de una historia médica física	Crear e imprimir la portada de una historia médica física	Generar la portada de una historia médica física con los datos básicos del paciente y código de barras asociado a la misma y finalmente imprimirla	Se generó e imprimió exitosamente la portada de una historia médica física, con el respectivo código de barras e información asociada al paciente seleccionado

Tabla 9: Casos de prueba. Iteración 2

#### 4.4.4 ITERACIÓN 3

Para esta iteración se desarrollaron las funcionalidades correspondientes al registro de entradas y salidas de historias médicas físicas en los distintos almacenes. Estas dos funcionalidades, son unas de las más importantes dentro del prototipo de sistema, pues son las que permiten el control de entrada y salida de una historia médica o grupo de historias entre las diferentes entidades (almacenes/sucursales) previamente definidas en el prototipo de sistema.

##### 4.4.4.1 Planificación

<b>Nro. de iteración</b>	3
<b>Descripción</b>	Registrar la entrada y salida de una historia médica física o grupo de historias médicas, entre los distintos almacenes y sucursales definidos en el prototipo de sistema
<b>Historias de usuario</b>	- Desarrollo del módulo de registro de entradas de historias médicas en un almacén. - Desarrollo del módulo de registro de salidas de historias médicas de un almacén
<b>Tiempo estimado</b>	80 horas
<b>Fecha de inicio y fin</b>	12/07/2012 – 25/07/2012

Tabla 10: Planificación de la iteración 3

##### 4.4.4.2 Bitácora de Desarrollo

La bitácora de desarrollo para esta iteración puede ser observada en la siguiente tabla (tabla 11) en esta se describe el tiempo estimado y real de las actividades generales que se realizan durante este período.



Nro.	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados
8	Desarrollo del módulo de registro de entrada de historias al almacén/sucursal	1,4,6,7	12/07/2012	18/07/2012	5
9	Desarrollo del módulo de registro de salida de historias del almacén/sucursal	1,4,6,7,8	19/07/2012	25/07/2012	5

Tabla 11: Bitácora de desarrollo de la iteración 3

#### 4.4.4.3 Diseño

En esta fase de la iteración 3, se diseñaron las pantallas correspondientes al registro de entradas y salidas de historias médicas de un almacén/sucursal a otro. Para el registro de salidas de historias médicas, se contemplaron los campos correspondientes al destino de la historia médica, el responsable de la salida y observaciones en caso de ser necesarias; luego de completar los campos y seleccionar las historias deseadas, el usuario puede registrar efectivamente la salida de las historias seleccionadas del almacén/sucursal, generándose automáticamente una orden de salida en la cual se muestra información relevante sobre las historias que se acaban de registrar como salientes del almacén.

Para registrar la entrada de una historia o un grupo de historias médicas, es necesario incluir el número de orden de salida en el formulario de la pantalla de entrada, este número está incluido en la orden de salida impresa al momento de registrar la salida de las historias médicas. Adicionalmente, el personal encargado de registrar la entrada, deberá escanear el código de barras de la portada de cada una de las historias físicas que acompaña esa orden de salida (a través del uso de un escáner de código de barras), de esta forma estas historias se incorporan automáticamente al listado de historias a las que se les da entrada en el almacén; una vez finalizada esta labor, se puede proceder a registrar la entrada de las historias médicas en el almacén/sucursal.

Algunas de las pantallas más importantes elaboradas en esta fase, se muestran en las siguientes imágenes (Ver imagen 11 e imagen 12):

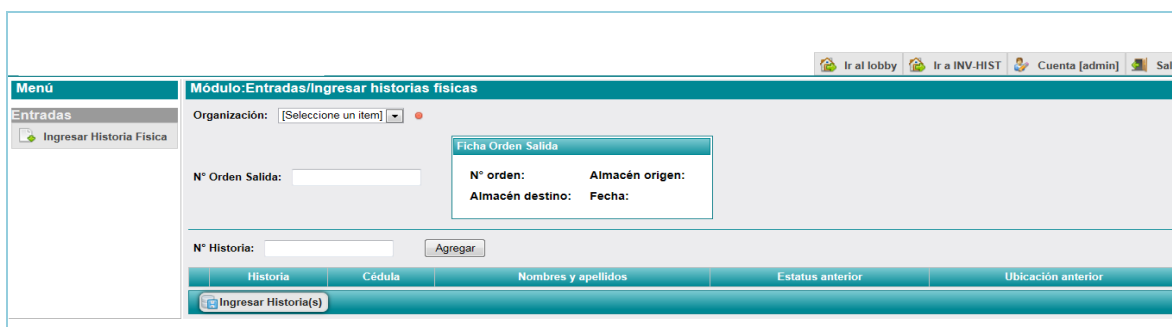


Imagen 11: Pantalla para registrar la entrada de historias médicas al almacén/sucursal



Imagen 12: Pantalla para registrar la salida de historias médicas del almacén/sucursal

#### 4.4.4.4 Pruebas

En esta fase de la iteración 3, se realizaron pruebas exhaustivas para garantizar el correcto funcionamiento de las funcionalidades de registro de entrada y salida de historias médicas físicas dentro del prototipo de sistema. Cada una de las pruebas realizadas pueden observarse con detalle en la bitácora de pruebas que se muestra a continuación (tabla 12):

Nº Caso de prueba	Módulo	Caso de prueba	Resultado esperado	Resultado obtenido
11	Registro de entradas de historias médicas de un almacén	Registrar la entrada de una historia médica	Ingresar una historia médica al almacén/sucursal donde está registrado el usuario mediante la lectura del código de barras de la misma	Se logró ingresar la historia médica en el almacén/sucursal deseado, mediante la lectura del código de barras de la misma
12	Registro de entradas de historias médicas de un almacén	Registrar la entrada de un grupo de historias médicas	Ingresar un grupo de historias médicas al almacén/sucursal donde está registrado el usuario mediante el número de orden de salida y el código de barras de las mismas	Se logró ingresar el grupo de historias médicas en el almacén/sucursal deseado, pudiendo utilizar el número de orden de salida para realizar dicho ingreso

13	Registro de salidas de historias médicas de un almacén	Registrar la salida de una historia médica	Realizar la salida de una historia médica de un almacén/sucursal donde está registrado el usuario a otro, mediante la lectura del código de barras de la misma	Se logró realizar la salida de la historia médica al almacén/sucursal deseado, mediante la lectura del código de barras de la misma
14	Registro de salidas de historias médicas de un almacén	Registrar la salida de un grupo de historias médicas	Realizar la salida de un grupo historias médicas de un almacén/sucursal donde está registrado el usuario a otro	Se pudo realizar la salida del grupo de historias médicas seleccionadas del almacén/sucursal en el cual se inició sesión, a otro diferente
16	Registro de salidas de historias médicas de un almacén	Obtener el reporte de orden de salida	Generar el reporte de orden de salida luego de efectuar la salida de historias médicas del almacén/sucursal donde está registrado el usuario a otro	Al generar el reporte de orden de salida el prototipo se quedaba procesando la solicitud y finalmente no mostraba el reporte

Tabla 12: Casos de prueba. Iteración 3

#### 4.4.5 ITERACIÓN 4

En la iteración 4 estuvieron contemplados el desarrollo de las funcionalidades correspondientes al registro de solicitudes de historias médicas físicas, así como su visualización y despacho por parte del personal encargado (peticiones). Al igual que en el caso de las opciones de registro de entrada y salida, el registro de solicitudes y la visualización y despacho de las mismas (peticiones) juegan un papel fundamental en el funcionamiento del prototipo, ya que mediante las mismas, el personal que labora en cualquier almacén/sucursal del centro de salud seleccionado como caso de estudio, podrá conocer cuáles son las historias requeridas, para su posterior despacho, y de igual manera podrá solicitar las historias médicas que sean requeridas por determinada razón.

##### 4.4.5.1 Planificación

<b>Nro. de iteración</b>	4
<b>Descripción</b>	Desarrollo de las opciones correspondientes a las solicitudes y peticiones de historias médicas físicas
<b>Historias de usuario</b>	<ul style="list-style-type: none"> <li>- Desarrollo del módulo de recepción de peticiones de historias médicas físicas.</li> <li>- Desarrollo del módulo de registro de solicitudes de historias médicas</li> </ul>

	físicas.
<b>Tiempo estimado</b>	160 horas
<b>Fecha de inicio y fin</b>	26/07/2012 – 22/08/2012

Tabla 13: Planificación de la iteración 4

#### 4.4.5.2 Bitácora de Desarrollo

La bitácora de desarrollo para esta iteración puede ser observada en la siguiente tabla (tabla 14), en ella se describe el tiempo estimado y real de las actividades generales que se realizan durante este período.

Nro.	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados
10	Solicitar historias a un almacén/sucursal	1,4,6,7,8	26/07/2012	08/08/2012	5
11	Recibir peticiones de un almacén/sucursal	1,4,6,7,8	09/08/2012	22/08/2012	5

Tabla 14: Bitácora de desarrollo de la iteración 4

#### 4.4.5.3 Diseño

Para esta fase de la iteración, se diseñaron las pantallas correspondientes al registro de solicitudes y préstamos de historias médicas de un almacén/sucursal a otro. Para solicitar una historia médica se diseñó una página en la cual se requiere completar los campos correspondientes a: la historia médica del paciente, para lo cual se colocó un filtro de búsqueda que permite llevar a cabo rápidamente esta operación; la fecha de la próxima cita del paciente (esto a fines de determinar la fecha en la que la historia debe estar en la sede solicitante), campo que será suministrado por el sistema de gestión de citas al cual pretende ser integrado el prototipo planteado; la fecha de entrega requerida, que se refiere a la fecha en la cual se necesita tener físicamente la historia para su uso; y por último un campo de observaciones que puede ser llenado en caso que sea necesario. Cabe destacar que entre la fecha de la próxima cita y la fecha de entrega requerida, el prototipo tomara la fecha más próxima, para procurar que la historia médica esté disponible en el almacén/sucursal donde sea requerida. Luego de completar cada uno de los campos, es posible registrar la solicitud de la historia o grupo de historias médicas físicas que son requeridas para su uso.

Por otro lado, estas solicitudes llegan al almacén/sucursal que dispone de la historia solicitada, a través de una especie de bandeja de entrada, la cual se denominará peticiones. Dentro de esta pantalla que presenta el listado de peticiones, se ofrecerá al usuario facilidades para listar todas

aquellas peticiones que provienen de un determinado almacén/sucursal, organizar las peticiones por fecha de solicitud (se presentará en los primeros lugares de la lista, aquellas historias que se requieran con mayor prioridad). Así mismo, y una vez seleccionadas las historias a despachar, se deben incluir funcionalidades para el registro de la salida de las historias hacia el almacén solicitante, también será posible el registro de una historia como "extraviada", esto en el caso de que alguna de las historias solicitadas no se encuentre físicamente ubicada en el almacén.

Algunas de las pantallas más importantes elaboradas en esta fase, se pueden observar en las siguientes imágenes (Ver imagen 13 e imagen 14):



Imagen 13: Pantalla para administrar las peticiones (solicitudes) que llegan a un almacén/sucursal

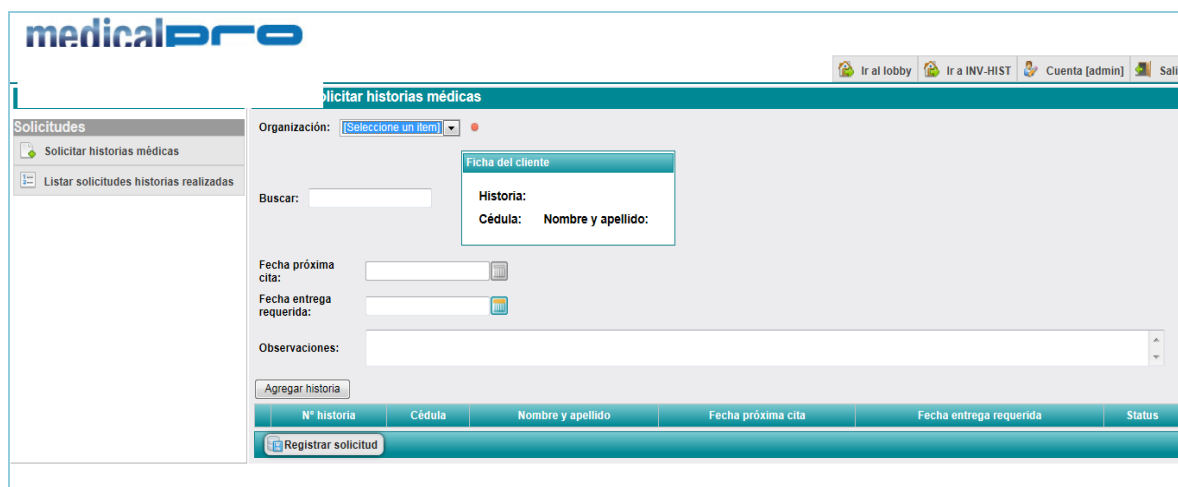


Imagen 14: Pantalla para registrar las solicitudes de historias médicas físicas a otros almacenes/sucursales

#### 4.4.5.4 Pruebas

Para esta fase de la iteración, se realizaron las pruebas que se describen en la siguiente tabla (Ver tabla 15):

Nº Caso de prueba	Módulo	Caso de prueba	Resultado esperado	Resultado obtenido
17	Registro de solicitudes de historias médicas físicas	Realizar la solicitud de historias médicas físicas	Solicitar historias médicas del almacén/sucursal donde está registrado el usuario a otro almacén/sucursal	Se logró solicitar historias médicas mediante el módulo de solicitud de historias médicas al almacén/sucursal seleccionado
18	Registro de solicitudes de historias médicas físicas	Listar solicitudes de historias médicas realizadas	Generar un listado de las solicitudes de historias médicas realizadas en el almacén/sucursal donde está registrado el usuario	Se logró obtener el listado de todas las solicitudes de historias médicas que se realizaron en el almacén/sucursal que se estaba utilizando como caso de prueba
19	Recepción de peticiones de historias médicas físicas	Listar peticiones de historias médicas	Generar un listado de las peticiones de historias médicas que se realizaron al	El listado mostrado por el prototipo no estaba acorde con las peticiones que deberían

		realizadas	almacén/sucursal donde está registrado el usuario	ser listadas según las pruebas
20	Recepción de peticiones de historias médicas físicas	Registrar la salida historias médicas solicitadas desde el módulo de peticiones	Realizar la salida de historias médicas que fueron solicitadas al almacén/sucursal donde está registrado el usuario mediante el botón "Registrar salida" del módulo de peticiones	Se logró realizar la salida de las historias seleccionadas como caso de prueba, pertenecientes al listado de historias médicas solicitadas. El botón "Registrar salida" funciona de manera correcta
21	Recepción de peticiones de historias médicas físicas	Marcar historias médicas como extraviada	Seleccionar historias médicas del listado de peticiones y marcarlas como extraviadas haciendo uso del botón "Registrar historia(s) extraviada(s)" del módulo de peticiones	Se pudo marcar como extraviadas las historias seleccionadas del listado de peticiones mediante el uso del botón "Registrar historia(s) extraviada(s)"

Tabla 15: Casos de prueba. Iteración 4

#### 4.4.6 ITERACIÓN 5

Para esta iteración se desarrollaron las funcionalidades correspondientes al control de historias médicas en préstamo y al inventario de historias médicas físicas. Ambas opciones permitirán al personal encargado de la gestión y almacenamiento de historias médicas físicas del centro de salud seleccionado como caso de estudio, evitar que muchas historias médicas que han sido prestadas a doctores para realizar trabajos, estudios, etc., sean luego catalogadas como extraviadas debido a la falta de un control adecuado y el desconocimiento de su verdadera ubicación.

##### 4.4.6.1 Planificación

<b>Nro. de iteración</b>	5
<b>Descripción</b>	Desarrollo de las opciones correspondientes al control de historias médicas en préstamo e inventario de historias médicas.
<b>Historias de usuario</b>	- Desarrollo de un módulo de control de historias médicas en préstamo - Desarrollo del módulo de inventario. - Desarrollo del módulo de creación del reporte de inventario.
<b>Tiempo estimado</b>	80 horas
<b>Fecha de inicio y fin</b>	23/08/2012 – 05/09/2012

Tabla 16: Planificación de la iteración 5

#### 4.4.6.2 Bitácora de Desarrollo

En la presente fase se incluye la bitácora de desarrollo (Ver tabla 17), describiendo el tiempo estimado y real de las actividades generales que se realizan durante este período.

Nro.	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados
12	Controlar historias médicas en préstamo	1,4,6,7,8	23/08/2012	29/08/2012	5
13	Realizar inventario de historias médicas	1,4,6,7,8	30/08/2012	05/09/2012	5

Tabla 17: Bitácora de desarrollo de la iteración 5

#### 4.4.6.3 Diseño

En esta iteración se diseñaron las pantallas referentes a los módulos de inventario y control de historias médicas en préstamo. Para el módulo de inventario, se diseñó una pantalla sencilla, la cual permite escanear el código de barras de cada una de las historias médicas físicas de un almacén/sucursal y de esta manera ir agregándolas a un listado. Una vez finalizado el proceso de escaneo de todas las historias, se le proporcionan dos opciones al usuario final:

- La opción de finalizar el inventario, a través de la cual, el prototipo de sistema automáticamente compara el listado que acaba de generar el usuario a través del escaneo del código de barras de las portadas de las historias médicas físicas, con el listado de historias, que de acuerdo a la Base de Datos del prototipo, deben estar en el almacén en cuestión. Como producto de esta comparación o cotejamiento se produce un reporte en el cual se listan las historias médicas que, de acuerdo a la base de datos del prototipo de sistema, están en el almacén, pero no fue escaneado su código de barra (faltantes), y también las historias que, según la base de datos del prototipo de sistema, no se encontraban en el almacén en cuestión, pero su código de barras fue escaneado en el proceso de inventario (sobrantes). Cabe destacar que aquellas historias médicas faltantes, se declaran automáticamente como extraviadas (esto por solicitud explícita del cliente).
- La opción de guardar el listado de historias escaneadas y continuar posteriormente con el proceso de inventario. Esta opción es de gran utilidad, cuando el volumen de historias del almacén es considerablemente elevado, impidiendo que la realización del inventario se lleve a cabo en un solo día.



Con respecto al control de historias en préstamo, se diseñó una interfaz bastante sencilla, en la cual, partiendo de uno de los almacenes, el prototipo muestra un listado de las historias médicas físicas que este tiene en calidad de préstamo.

Algunas de las pantallas más importantes elaboradas en esta fase, se observan en las siguientes imágenes (Ver imagen 15 e imagen 16):



Imagen 15: Interfaz para el control de historias médicas en préstamo



Imagen 16: Interfaz para realizar el inventario de historias médicas

#### 4.4.6.4 Pruebas

Para esta fase de la iteración, se realizaron las siguientes pruebas (tabla 18):

Nº Caso de prueba	Módulo	Caso de prueba	Resultado esperado	Resultado obtenido
22	Control de historias médicas en préstamo	Listar las historias médicas en préstamo	Realizar un listado en el cual se muestren las historias médicas que están en calidad de préstamo	Se pudo obtener un listado de las historias médicas que se encontraban en calidad de préstamo con solo completar el campo "almacén" mediante el cual es filtrada la información
23	Inventario	Inventariar historias médicas	Realizar un inventario de historias médicas en el cual se pueda comparar las historias	Se logró realizar el inventario del almacén/sucursal que se tomó como caso de

			médicas físicas, con las historias médicas que contiene el almacén definido en el prototipo de sistema. El resultante de este proceso será un reporte con los datos calculados por el prototipo	prueba. Se debió realizar numerosos intentos para comprobar los cálculos realizados en el reporte (número total de historias, historias sobrantes, historias faltantes, etc.). También se pudo comprobar el correcto funcionamiento de la opción "actualizar inventario", la cual permite realizar el inventario por partes, es decir, en distintos instantes de tiempo y sin necesidad de mantenerse en la misma pantalla
24	Inventario	Marcar historias médicas como extraviada	Seleccionar historias médicas del listado de inventario y marcarlas como extraviadas haciendo uso del botón "Registrar historia(s) extraviada(s)" del módulo de inventario	Al marcar una historia médica e intentar colocarla como extraviada, el prototipo no hacía ninguna acción al presionar el botón "Registrar historia(s) extraviada(s)"

Tabla 18: Casos de prueba. Iteración 5

#### 4.4.7 ITERACIÓN 6

Por último, y como requerimiento adicional del cliente, se desarrollaron en esta iteración, dos funcionalidades adicionales para el control de las historias médicas extraviadas y en tránsito. Estas nuevas funcionalidades, mejoraran la gestión de las historias médicas físicas, ya que mediante éstas se podrá conocer información sobre el tránsito de las historias médicas, así como también de las historias médicas que se han extraviado. Permitiendo la ejecución de acciones en consecuencia, por ejemplo, en el caso de las historias que permanezcan en tránsito por un tiempo mayor al prudente para que las historias viajen del almacén origen al destino.

##### 4.4.7.1 Planificación

<b>Nro. de iteración</b>	6
<b>Descripción</b>	Desarrollo de los módulos de control de historias médicas extraviadas y en

	tránsito
<b>Historias de usuario</b>	- Desarrollo de un módulo de control de historias médicas extraviadas - Desarrollo de un módulo de control de historias médicas en tránsito
<b>Tiempo estimado</b>	80 horas
<b>Fecha de inicio y fin</b>	06/09/2012 – 19/09/2012

Tabla 19: Planificación de la iteración 6

#### 4.4.7.2 Bitácora de Desarrollo

En la presente fase se incluye la bitácora de desarrollo (Ver tabla 20), describiendo el tiempo estimado y real de las actividades generales que se realizan durante este período.

Nro.	Tarea	Precede	Fecha inicio	Fecha fin	Días estimados
14	Controlar historias médicas extraviadas	1,4,6,7,8	06/09/2012	12/09/2012	5
15	Controlar historias médicas en tránsito	1,4,6,7,8	13/09/2012	19/09/2012	5

Tabla 20: Bitácora de desarrollo de la iteración 6

#### 4.4.7.3 Diseño

El diseño de la interfaz para el control de historias médicas en tránsito y control de historias médicas extraviadas, fue el mismo. Se le presenta al usuario un sencillo listado donde se muestran las historias extraviadas o en tránsito según sea el caso. En este listado puede observarse información importante acerca de cada una de las historias médicas, como lo son: el número de historia médica, datos básicos del paciente, número de orden de salida a la cual pertenecía la historia, su origen y su respectivo destino. A partir de esta información el usuario puede determinar ciertas acciones que mejoren la gestión de las historias médicas.

Algunas de las interfaces más importantes elaboradas en esta fase, se resumen en las siguientes imágenes (Ver imagen 17 e imagen 18):



Imagen 17: Interfaz para el control de historias médicas en tránsito

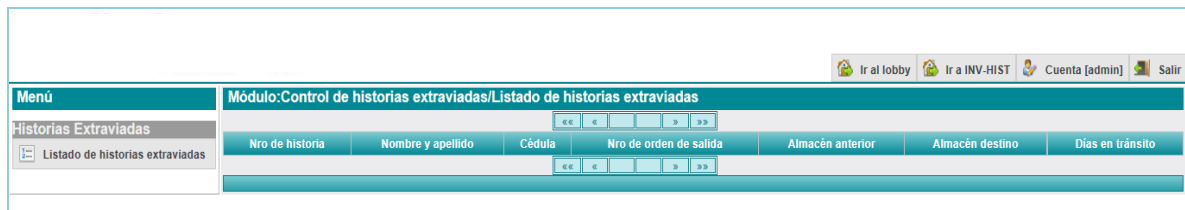


Imagen 18: Interfaz para el control de historias médicas extraviadas

#### 4.4.7.4 Pruebas

Para esta fase de la iteración, se realizaron las siguientes pruebas (Ver tabla 21):

Nº Caso de prueba	Módulo	Caso de prueba	Resultado esperado	Resultado obtenido
25	Control de historias médicas extraviadas	Listar las historias médicas extraviadas	Realizar un listado en el cual se muestren las historias médicas que están extraviadas	Se pudo obtener un listado de las historias médicas que se encontraban extraviadas mediante la opción "Listado de historias extraviadas" del módulo "Control de historias médicas extraviadas"
26	Control de historias médicas en	Listar las historias médicas en	Realizar un listado en el cual se muestren las historias médicas que	Se pudo obtener un listado de las historias médicas que se

	tránsito	tránsito	están en tránsito	encontraban en tránsito mediante la opción "Listado de historias en tránsito" del módulo "Control de historias médicas en tránsito"
--	----------	----------	-------------------	---

Tabla 21: Casos de prueba. Iteración 6

## **RESULTADOS**

La utilización de una adaptación de la Programación Extrema, como método ágil de desarrollo, se logró:

- Identificar objetivamente los requerimientos del cliente y poder contar con su apoyo a lo largo del desarrollo del proyecto, obteniendo una solución bastante cercana a lo que el cliente deseaba y en consecuencia directa la satisfacción del mismo.
- Elaborar la metáfora del sistema, la cual describe el funcionamiento del prototipo de sistema de manera simple y sencilla.
- Traducir los requerimientos en historias de usuarios, en las cuales se estableció su prioridad, los días estimados para su desarrollo y una descripción del requerimiento a desarrollar.
- Definir, planificar y ejecutar 6 iteraciones, estableciendo fecha de inicio y culminación para cada una de estas.

Para dar solución a cada uno de los requerimientos solicitados, fue necesario desarrollar los siguientes módulos:

- Módulo de acceso al prototipo de sistema y menú principal, mediante el cual un usuario puede ingresar al prototipo y sus funcionalidades.
- Módulo de gestión de almacenes, el cual permite crear, eliminar y editar almacenes dentro del prototipo de sistema.
- Módulo de creación de la portada de una historia médica física, en el cual se puede realizar la impresión de una portada que permite la identificación de una historia y la manipulación de la misma por los distintos módulos del prototipo de sistema mediante el uso de un código de barras.
- Módulo de registro de entradas de historias médicas en un almacén, el cual permite ingresar historias médicas en un almacén específico.
- Módulo de registro de salida de historias médicas en el almacén, en el que se puede registrar la salida de historias médicas a un almacén específico. Provee la opción de imprimir un reporte de orden de salida en el cual se detalla las historias que están saliendo, destino, fecha, etc.
- Módulo de recepción de peticiones de historias médicas físicas, el cual permite conocer las historias médicas que son solicitadas en un almacén específico.
- Módulo de registro de solicitudes de historias médicas físicas, en el cual se pueden realizar solicitudes de historias médicas al almacén donde se encuentren y cuando sean

requeridas. Permite de igual manera realizar un listado de las solicitudes que se han realizado.

- Módulo de inventario, mediante el cual se permite la realización de un inventario de las historias médicas que se encuentran físicamente en un almacén. El resultado de este inventario es un reporte detallado en el cual se muestran las historias inventariadas, las historias faltantes (historias que deberían estar físicamente en el almacén donde se realizó el inventario y no se encuentran), historias sobrantes (historias que por algún motivo se encuentran en el almacén donde se realizó el inventario y no deberían estar allí).
- Módulo de control de historias médicas en préstamo, el cual permite consultar aquellas historias que se encuentran en calidad de préstamo a un doctor y por esta razón no se encuentran en los almacenes definidos en el prototipo
- Módulo de control de historias médicas en tránsito, donde se puede consultar aquellas historias que se encuentran en tránsito de un almacén a otro.
- Módulo de historias médicas extraviadas, donde se puede consultar información acerca de todas aquellas historias médicas que no se encuentran en ningún almacén y por consecuencia tengan el estatus "extraviada".

Con el desarrollo de este trabajo especial de grado pudo resolverse la problemática existente en el centro de salud seleccionado como caso de estudio, y esto puede reafirmarse con la satisfacción del cliente al hacer uso del prototipo y emitir sus comentarios positivos respecto al mismo. Esto una vez más ratifica que la computación es una ciencia que puede apoyar y ayudar en la gestión de diferentes áreas de conocimiento.

## **CONCLUSIONES Y RECOMENDACIONES**

El objetivo principal del presente Trabajo Especial de Grado fue alcanzado exitosamente, mediante el desarrollo del prototipo de un sistema web que permite gestionar de manera eficaz el uso y almacenamiento de las historias médicas físicas en el centro de salud que sirvió como caso de estudio. En este sentido se redujo la carga de trabajo del personal encargado de la gestión y almacenamiento de historias físicas, al contar con la rapidez y simplicidad que ofrece el prototipo desarrollado.

Por otra parte, se puede garantizar a los pacientes que acuden al centro de salud, que sus historias siempre estarán disponibles para cuando ellos requieran consultarse en cualquiera de las sedes donde se presta atención médica, ya que el prototipo está diseñado para resolver los problemas que existían anteriormente en el movimiento de historias médicas entre las sedes y el almacén principal.

Adicionalmente el prototipo cuenta con la opción de realizar inventario de historias médicas físicas en una sede o almacén, con lo cual se puede conocer la cantidad de historias médicas físicas dentro de la sede o almacén donde se esté realizando el inventario, pudiendo mediante esta opción conocer si existen historias médicas extraviadas y de esta manera poder tomar las acciones que sean más convenientes en cada caso.

El prototipo de sistema, fue desarrollado utilizando tecnologías actuales entre las cuales podemos mencionar: el lenguaje de programación Java JEE, el entorno de desarrollo Eclipse, el sistema manejador de base de datos PostgreSQL, IReport y JasperReport como entorno y motor para la elaboración de reportes, entre otras; lo que significó un reto personal en vista de que desconocía el manejo y utilización de algunas de ellas, motivándome a documentarme para aprender a utilizarlas y poder de esta manera llevar a cabo con éxito el desarrollo del prototipo.

Otro aspecto importante a mencionar, es la adaptación que se llevó a cabo de la Programación Extrema como método ágil de desarrollo, el cual facilitó el trabajo de forma organizada, agrupando los requerimientos en un conjunto de iteraciones que se fueron desarrollando progresivamente. Además, la constante comunicación con el cliente permitió realizar frecuentes revisiones de los avances de la aplicación. Con la aplicación de XP, se obtuvo un prototipo de sistema acoplado a las necesidades del usuario, por lo cual puedo concluir que este método de desarrollo puede resultarme útil para el desarrollo de futuros proyectos.

Gracias a la elaboración de este Trabajo Especial de Grado se logró un aporte significativo al sector salud, ya que hasta el momento no se contaba con un sistema que facilitara los procesos de gestión y manejo de historias médicas físicas, pudiendo ser este prototipo, el primero de una serie de



subsiguientes adaptaciones que sean incluidas en el resto de los centros de salud que presenten necesidades similares al que se estudió en este trabajo.

Finalmente se presentan a continuación una serie de recomendaciones a ser tomadas en cuenta para realizar mejoras, agregar nuevas funcionalidades e incluso adaptar este prototipo para resolver otro tipo de problemáticas, permitiendo de esta manera el crecimiento constante del prototipo de sistema realizado en este Trabajo Especial de Grado:

- Permitir la impresión de portadas por lotes, esto facilitaría el trabajo al momento de identificar las carpetas, ya que un centro de salud grande como el que se estudió en este trabajo, puede llegar a manejar un volumen alto de historias médicas físicas, resultando tedioso la impresión de portadas de manera individual.
- Incorporar un reporte donde se muestre la información histórica de los movimientos que ha tenido una historia en un rango de tiempo seleccionado por el usuario, pudiendo ser usado como punto de partida para determinar la ubicación de historias médicas de las cuales se desconoce su ubicación.
- Incorporar al prototipo estadísticas que permitan determinar el número de pacientes atendidos por cada sede, con el objetivo de mejorar la atención mediante la distribución de los mismos en cada una de las sedes según las capacidades y personal médico disponible en cada una de las mismas.

### REFERENCIAS BIBLIOGRÁFICAS Y DÍGITALES

- Arias, F. (2006). *El Proyecto de Investigación. Introducción a la metodología científica* (5ta ed.). Caracas: Episteme. Pág. 45-69.
- Asociación Venezolana de derecho Farmacéutico. Historia clínica electrónica. [Página web en línea]. Disponible en: <http://avedefar.blogspot.com/2011/05/historia-clinica-electronica.html>.
- Aulló Chaves, M. y Pelayo Pardos, S. (1997). *Responsabilidad legal profesional: La historia clínica*. Madrid. Pág. 73-115.
- Carballo Maestre, A. (2009). *Arquitectura JEE – Patrón MVC*. [Página web en línea]. Disponible en: <http://carballomaestre.blogspot.com/2009/02/arquitectura-JEE-patron-mvc.html>.
- Consultores y Asesores Gestión en Salud. Proyecto sobre historia clínica virtual gana premio a la calidad en el ámbito del sistema nacional de salud en España 2009. [Página web en línea]. Disponible en: <http://maiquiflores.over-blog.es/article-proyecto-sobre-historia-clinica-virtual-gana-premio-a-la-calidad-en-el-ambito-del-sistema-nacional-de-salud-en-espa-a-2009-65185714.html>.
- Crespo, C. (2004). Introducción a Hibernate. [Página web en línea]. Disponible en: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=hibernate>.
- Diario Jaen. Fotodenuncia. [Página web en línea]. Disponible en: <http://www.diariojaen.es/index.php/fotodenuncia/>
- Díaz, C. (2009). *Historia médica*. [Página web en línea]. Disponible en: <http://historiaclinicaessc1.blogspot.com/2009/09/historia-clinica.html>.
- Exforsys Inc. (2005). *JEE Overview*. [Página web en línea]. Disponible en: <http://www.solotuweb.com/fs~id~1654.html>.
- Hommel, S. (2006). *Convecciones de código para el lenguaje de programación Java*. Estados Unidos. Pág. 5-26.

- Hospital Universitario de Guadalajara. El archivo del Hospital gestiona cerca de 5.000 historias clínicas cada día. [Página web en línea]. Disponible en: <http://hugu.sescam.jccm.es/>
- Intramed. ¿Qué es una historia clínica orientada a problemas?. [Página web en línea]. Disponible en: <http://www.intramed.net/contenidover.asp?contenidoID=42838>
- Java Sun Microsystems. The Java EE 5 tutorial. [Página web en línea]. Disponible en: <http://docs.oracle.com/javaee/5/tutorial/doc/>
- Lago, R. (2007). *Arquitectura JEE*. [Página web en línea]. Disponible en: <http://www.proactiva-calidad.com/java/arquitectura/index.html>.
- Muniagurria, A. y Libman, J. (1998). *Semiología Clínica*. Argentina: Universidad Nacional del Rosario. Pág. 26-53.
- Pérez, L. (2011). XP – Extreme Programming. [Página web en línea]. Disponible en: [http://analystperez.blogspot.com/2011\\_04\\_17\\_archive.html](http://analystperez.blogspot.com/2011_04_17_archive.html)
- Robles, G. (2002). Programación extrema y software libre. [Página web en línea]. Disponible en: <http://es.tldp.org/Presentaciones/200211hispalinux/ferrer/robles-ferrer-ponencia-hispalinux-2002.html>.
- Risberg, T. y Evans R. (2008). Desarrollando una aplicación Spring framework MVC. [Página web en línea]. Disponible en: <http://www.davidmarco.es/tutoriales/spring-mvc-sbs/>.
- Wikipedia (2011). *Java EE*. [Página web en línea]. Disponible en: [http://es.wikipedia.org/wiki/Java\\_EE](http://es.wikipedia.org/wiki/Java_EE).
- Wikipedia (2011). *Modelo Vista Controlador*. [Página web en línea]. Disponible en: [http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador).

## Anexos

### Anexo 1: Convenciones de código para el lenguaje de programación JAVA

A continuación se presentan las convenciones más importantes a ser tomadas en cuenta para el desarrollo de la aplicación propuesta en el lenguaje JEE (Hommel, 2006)

#### Declaraciones de clases e interfaces

	Partes de la declaración de una clase o interface	Notas
1	Comentario de documentación de la clase o interface ( <i>/**...*/</i> )	Ver " <a href="#">Comentarios de documentación</a> " en la página 14 para más información sobre lo que debe aparecer en este comentario.
2	Sentencia <code>class</code> o <code>interface</code>	
3	Comentario de implementación de la clase o interface si fuera necesario ( <i>/*...*/</i> )	Este comentario debe contener cualquier información aplicable a toda la clase o interface que no era apropiada para estar en los comentarios de documentación de la clase o interface.
4	Variables de clase ( <code>static</code> )	Primero las variables de clase <code>public</code> , después las <code>protected</code> , después las de nivel de paquete (sin modificador de acceso), y después las <code>private</code> .
5	Variables de instancia	Primero las <code>public</code> , después las <code>protected</code> , después las de nivel de paquete (sin modificador de acceso), y después las <code>private</code> .
6	Constructores	
7	Métodos	Estos métodos se deben agrupar por funcionalidad más que por visión o accesibilidad. Por ejemplo, un método de clase privado puede estar entre dos métodos públicos de instancia. El objetivo es hacer el código mas legible y comprensible.

### Convenciones de nombres

Tipos de identificadores	Reglas para nombres	Ejemplos
Paquetes	<p>El prefijo del nombre de un paquete se escribe siempre con letras ASCII en minúsculas, y debe ser uno de los nombres de dominio de alto nivel, actualmente com, edu, gov, mil, net, org, o uno de los códigos ingleses de dos letras que identifican cada país como se especifica en el ISO Standard 3166, 1981.</p> <p>Los subsecuentes componentes del nombre del paquete variarán de acuerdo a las convenciones de nombres internas de cada organización. Dichas convenciones pueden especificar que algunos nombres de los directorios correspondan a divisiones, departamentos, proyectos o máquinas.</p>	<pre>com.sun.eng com.apple.quicktime.v2 edu.cmu.cs.bovik.cheese</pre>
Clases	<p>Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Intentar mantener los nombres de las clases simples y descriptivos. Usar palabras completas, evitar acrónimos y abreviaturas (a no ser que la abreviatura sea mucho más conocida que el nombre completo, como URL or HTML).</p>	<pre>class Cliente; class ImagenAnimada;</pre>
Interfaces	<p>Los nombres de las interfaces siguen la misma regla que las clases.</p>	<pre>interface ObjetoPersistente; interface Almacen;</pre>
Métodos	<p>Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula, y la primera letra de las siguientes palabras que lo forma en mayúscula.</p>	<pre>ejecutar(); ejecutarRapido(); cogerFondo();</pre>
Variables	<p>Excepto las constantes, todas las instancias y variables de clase o método empezarán con minúscula. Las palabras internas que lo forman (si son compuestas) empiezan con su primera letra en mayúsculas. Los nombres de variables no deben empezar con los caracteres subguión "_" o signo del dolar "\$", aunque ambos están permitidos por el lenguaje.</p> <p>Los nombres de las variables deben ser cortos pero con significado. La elección del nombre de una variable debe ser un mnemónico, designado para indicar a un observador casual su función. Los nombres de variables de un solo carácter se deben evitar, excepto para variables índices temporales. Nombres comunes para variables temporales son i, j, k, m, y n para enteros; c, d, y e para caracteres.</p>	<pre>int          i; char        c; float miAnchura;</pre>
Constantes	<p>Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas separando las palabras con un subguión ("_"). (Las constantes ANSI se deben evitar, para facilitar su depuración.)</p>	<pre>static final int ANCHURA_MINIMA = 4; static final int ANCHURA_MAXIMA = 999; static final int COGER_LA_CPU = 1;</pre>

## Sentencias

### 1 Sentencias simples

Cada línea debe contener como mucho una sentencia. Ejemplo:

```
argv++;           // Correcto
argc--;          // Correcto
argv++; argc--;  // EVITAR!
```

### 2 Sentencias compuestas

Las sentencias compuestas son sentencias que contienen listas de sentencias encerradas entre llaves "{ sentencias }". Ver la siguientes secciones para ejemplos.

- Las sentencias encerradas deben indentarse un nivel más que la sentencia compuesta.
- La llave de apertura se debe poner al final de la línea que comienza la sentencia compuesta; la llave de cierre debe empezar una nueva línea y ser indentada al mismo nivel que el principio de la sentencia compuesta.
- Las llaves se usan en todas las sentencias, incluso las simples, cuando forman parte de una estructura de control, como en las sentencias *if-else* o *for*. Esto hace más sencillo añadir sentencias sin incluir bugs accidentalmente por olvidar las llaves.

### 3 Sentencias de retorno

Una sentencia *return* con un valor no debe usar paréntesis a menos que hagan el valor de retorno más obvio de alguna manera. Ejemplo:

```
return;

return miDiscoDuro.size();

return (tamanyo ? tamanyo : tamanyoPorDefecto);
```

### 4 Sentencias if, if-else, if else-if else

La clase de sentencias *if-else* debe tener la siguiente forma:

```
if (condicion) {
    sentencias;
}

if (condicion) {
    sentencias;
} else {
    sentencias;
}

if (condicion) {
    sentencias;
} else if (condicion) {
    sentencias;
} else{
    sentencias;
}
```

```
if (condicion) //EVITAR! ESTO OMITI LAS LLAVES {}!  
    sentencias;
```

### 5 Sentencias for

Una sentencia `for` debe tener la siguiente forma:

```
for (inicializacion; condicion; actualizacion) {  
    sentencias;  
}
```

Una sentencia `for` vacía (una en la que todo el trabajo se hace en las cláusulas de inicialización, condición, y actualización) debe tener la siguiente forma:

```
for (inicializacion; condicion; actualizacion);
```

Al usar el operador coma en la cláusula de inicialización o actualización de una sentencia `for`, evitar la complejidad de usar más de tres variables. Si se necesita, usar sentencias separadas antes de bucle `for` (para la cláusula de inicialización) o al final del bucle (para la cláusula de actualización).

### 6 Sentencias while

Una sentencia `while` debe tener la siguiente forma:

```
while (condicion) {  
    sentencias;  
}
```

Una sentencia `while` vacía debe tener la siguiente forma:

```
while (condicion);
```

### 7 Sentencias do-while

Una sentencia `do-while` debe tener la siguiente forma:

```
do {  
    sentencias;  
} while (condicion);
```

### 8 Sentencias switch

Una sentencia `switch` debe tener la siguiente forma:

```
switch (condicion) {  
    case ABC:  
        sentencias;  
        /* este caso se propaga */  
  
    case DEF:  
        sentencias;  
        break;  
  
    case XYZ:  
        sentencias;  
        break;  
  
    default:  
        sentencias;  
        break;  
}
```

Cada vez que un caso se propaga (no incluye la sentencia `break`), añadir un comentario donde la sentencia `break` se encontraría normalmente. Esto se muestra en el ejemplo anterior con el comentario `/* este caso se propaga */`.

Cada sentencia `switch` debe incluir un caso por defecto. El `break` en el caso por defecto es redundante, pero previene que se propague por error si luego se añade otro caso.

## 9 Sentencias try-catch

Una sentencia `try-catch` debe tener la siguiente forma:

```
try {
    sentencias;
} catch (ExceptionClass e) {
    sentencias;
}
```

Una sentencia `try-catch` puede ir seguida de un `finally`, cuya ejecución se ejecutará independientemente de que el bloque `try` se halla completado con éxito o no.

```
try {
    sentencias;
} catch (ExceptionClass e) {
    sentencias;
} finally {
    sentencias;
}
```

## Declaraciones de class e interfaces

Al codificar clases e interfaces de Java, se siguen las siguientes reglas de formato:

- Ningún espacio en blanco entre el nombre de un método y el paréntesis "(" que abre su lista de parámetros
- La llave de apertura "{" aparece al final de la misma línea de la sentencia de declaración
- La llave de cierre "}" empieza una nueva línea indentada para ajustarse a su sentencia de apertura correspondiente, excepto cuando no existen sentencias entre ambas, que debe aparecer inmediatamente después de la de apertura "{"

```
class Ejemplo extends Object {
    int ivar1;
    int ivar2;

    Ejemplo(int i, int j) {
        ivar1 = i;
        ivar2 = j;
    }

    int metodoVacio() {}

    ...
}
```

- Los métodos se separan con una línea en blanco



## Anexo 2: Tablas del Modelo Relacional

orden_salida_expediente	
id	BIGINT(20)
numero_orden_salida	VARCHAR(16)
codigo_almacen_origen	VARCHAR(16)
nombre_almacen_origen	VARCHAR(64)
codigo_almacen_destino	VARCHAR(16)
nombre_almacen_destino	VARCHAR(64)
responsable	VARCHAR(255)
fecha	TIMESTAMP
observaciones	VARCHAR(255)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_numero_orden_salida	

ubicacion_geografica	
id	BIGINT(20)
codigo_ubicacion_geografica	VARCHAR(16)
nombre_ubicacion_geografica	VARCHAR(64)
id_ubicacion_geografica_superior	BIGINT(20)
id_tipo_nodo	BIGINT(20)
nivel_jerarquico	INT(11)
secuencia_nivel	INT(11)
clave_concatenada	VARCHAR(32)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_codigo_ubicacion_geografica	
fk_id_tipo_nodo_ug	
fk_id_ubicacion_geografica_superior	

rol_action	
id	BIGINT(20)
id_rol	BIGINT(20)
id_action_metadata	BIGINT(20)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_rol_action	
fk_id_action_metadata_ra	

rol	
id	BIGINT(20)
codigo_rol	VARCHAR(16)
nombre_rol	VARCHAR(64)
id_organizacion	BIGINT(20)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_codigo_rol_id_organizacion	
fk_id_organizacion_r	

inventario_expedientes	
id	BIGINT(20)
fecha_inventario	TIMESTAMP
es_inventario_finalizado	BIGINT(20)
es_sobranante	BIGINT(20)
es_faltante	BIGINT(20)
id_persona	BIGINT(20)
id_almacen_inventario	BIGINT(20)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
fk_id_almacen_inventario_expedientes_ie	
fk_id_persona_inventario_expedientes_ie	

usuario	
id	BIGINT(20)
codigo_usuario	VARCHAR(16)
password_usuario	VARCHAR(32)
password_hint_usuario	VARCHAR(64)
cedula_usuario	VARCHAR(16)
nombre_usuario	VARCHAR(64)
apellido_usuario	VARCHAR(64)
e_mail_usuario	VARCHAR(32)
es_activa	BIGINT(20)
es_cuenta_expirada	BIGINT(20)
es_cuenta_bloqueada	BIGINT(20)
es_credenciales_expiradas	BIGINT(20)
id_especialidad	BIGINT(20)
id_organizacion	BIGINT(20)
es_pin_habilitado	BIGINT(20)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_codigo_usuario	
fk_id_especialidad	
fk_id_organizacion	

renglon_solicitud_expediente	
id	BIGINT(20)
16 more...	
Indexes	
PRIMARY	
fk_id_solicitud_expediente_res	

status_expediente	
id	BIGINT(20)
codigo_status_expediente	VARCHAR(16)
nombre_status_expediente	VARCHAR(64)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_codigo_status_expediente	

almacen_expediente	
id	BIGINT(20)
codigo_almacen_expediente	VARCHAR(16)
nombre_almacen_expediente	VARCHAR(64)
es_prestamo	BIGINT(20)
id_almacen_expediente_superior	BIGINT(20)
id_tipo_nodo	BIGINT(20)
nivel_jerarquico	INT(11)
secuencia_nivel	INT(11)
clave_concatenada	VARCHAR(32)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_codigo_almacen_expediente	
fk_id_almacen_expediente_superior	
fk_id_tipo_nodo	

status_solicitud_expediente	
id	BIGINT(20)
codigo_status_solicitud_expediente	VARCHAR(16)
nombre_status_solicitud_expediente	VARCHAR(64)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_codigo_status_solicitud_expediente	

consulta_medica	
id	BIGINT(20)
fecha_consulta_medica	TIMESTAMP
motivo_consulta	VARCHAR(255)
enfermedad_actual	VARCHAR(255)
fecha_ultima_regla	TIMESTAMP
edad_gestacional	DOUBLE
fecha_posible_parto	TIMESTAMP
fecha_ultima_menstruacion	TIMESTAMP
tension_arterial	VARCHAR(16)
frecuencia_cardiaca	VARCHAR(16)
temperatura	VARCHAR(16)
frecuencia_respiratoria	VARCHAR(16)
peso	VARCHAR(16)
talla	VARCHAR(16)
estatura	VARCHAR(16)
exámenes_paradigmaticos	VARCHAR(255)
diagnostico	VARCHAR(255)
tratamiento	VARCHAR(255)
evolucion	VARCHAR(255)
codigo_doctor	VARCHAR(16)
cedula_doctor	VARCHAR(16)
nombre_doctor	VARCHAR(64)
apellido_doctor	VARCHAR(64)
url1	VARCHAR(255)
co_peso	DECIMAL(22,2)
co_incremento_ponderal_peso	VARCHAR(16)
co_ta	VARCHAR(16)
co_media_tension_arterial	VARCHAR(16)
co_au	DECIMAL(22,2)
co_ff	DECIMAL(22,2)
23 more...	
Indexes	
PRIMARY	
uq_consulta_dia	
fk_id_patologia	

organizacion	
id	BIGINT(20)
rif_organizacion	VARCHAR(16)
nit_organizacion	VARCHAR(16)
120 more...	
Indexes	
PRIMARY	
uq_rif_organizacion	
fk_id_organizacion_padre	
fk_id_ubicacion_geografica	

expediente	
id	BIGINT(20)
numero_impresiones_portada	BIGINT(20)
codigo_almacen_anterior	VARCHAR(16)
nombre_almacen_anterior	VARCHAR(64)
codigo_status_almacen_anterior	VARCHAR(16)
nombre_status_almacen_anterior	VARCHAR(64)
fecha_status	TIMESTAMP
fecha_solicitud_expediente	TIMESTAMP
es_solicitado	BIGINT(20)
es_portadaimpresa	BIGINT(20)
es_en_inventario	BIGINT(20)
id_almacen	BIGINT(20)
id_almacen_solicitante	BIGINT(20)
id_status_expediente	BIGINT(20)
id_ultima_orden_salida	BIGINT(20)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
fk_id_almacen_expediente	
fk_id_almacen_solicitante	
fk_id_status_expediente	
fk_id_ultima_orden_salida	

solicitud_expediente	
id	BIGINT(20)
numero_solicitud	VARCHAR(16)
codigo_almacen_solicitante	VARCHAR(16)
nombre_almacen_solicitante	VARCHAR(64)
fecha_solicitud	TIMESTAMP
observaciones	VARCHAR(255)
es_de_baja	BIGINT(20)
id_status_solicitud_expediente	BIGINT(20)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_numero_solicitud	
fk_id_status_solicitud_expediente	

persona	
id	BIGINT(20)
codigo_persona	VARCHAR(16)
cedula_persona	VARCHAR(16)
nombre_persona	VARCHAR(64)
rif_persona	VARCHAR(16)
nit_persona	VARCHAR(16)
apellido_persona	VARCHAR(64)
nombre_comercial	VARCHAR(64)
fecha_inscripcion_persona	TIMESTAMP
fecha_nacimiento_persona	TIMESTAMP
fecha_defuncion_persona	TIMESTAMP
fecha_ultima_regla	TIMESTAMP
fecha_vencimiento_rif	TIMESTAMP
telefono_movil_persona	VARCHAR(32)
telefono_fijo_persona	VARCHAR(32)
telefono_empresa_persona	VARCHAR(32)
telefono_referencia_persona	VARCHAR(32)
e_mail_persona	VARCHAR(32)
direccion_persona	VARCHAR(255)
nombre_empresa	VARCHAR(32)
direccion_empresa	VARCHAR(255)
nombre_contacto_uno	VARCHAR(64)
telefono_contacto_uno	VARCHAR(32)
telefono_movil_contacto_uno	VARCHAR(32)
email_contacto_uno	VARCHAR(32)
parentesco_contacto_uno	VARCHAR(255)
nombre_empresa	VARCHAR(32)
direccion_empresa	VARCHAR(255)
nombre_contacto_uno	VARCHAR(64)
telefono_contacto_uno	VARCHAR(32)
telefono_movil_contacto_uno	VARCHAR(32)
email_contacto_uno	VARCHAR(32)
parentesco_contacto_uno	VARCHAR(255)
nombre_contacto_dos	VARCHAR(64)
telefono_contacto_dos	VARCHAR(32)
telefono_movil_contacto_dos	VARCHAR(32)
email_contacto_dos	VARCHAR(32)
62 more...	
Indexes	
PRIMARY	
uq_codigo_persona_id_organizacion	
uq_cedula_id_categoria_persona_id_organizacion	
fk_id_ubicacion_geografica_nacimiento_p	
fk_id_ubicacion_geografica_residencia_p	
fk_id_usuario_p	
fk_id_doctor_p	
fk_id_organizacion_p	
fk_id_madre_p	
fk_id_padre_p	
fk_id_representante_p	
fk_id_expediente_p	
fk_id_organizacion_paciente_p	

action_metadata	
id	BIGINT(20)
codigo_action_metadata	VARCHAR(64)
nombre_action_metadata	VARCHAR(128)
es_activa	SMALLINT(6)
id_tipo_nodo	BIGINT(20)
id_action_metadata_superior	BIGINT(20)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_codigo_action_metadata	
fk_id_tipo_nodo_am	
fk_id_action_metadata_superior	

historico_movimientos_expediente	
id	BIGINT(20)
numero_historia	VARCHAR(16)
nombre_almacen_origen	VARCHAR(255)
nombre_almacen_destino	VARCHAR(255)
status_expediente	VARCHAR(255)
nombre_paciente	VARCHAR(255)
apellido_paciente	VARCHAR(255)
cedula_paciente	VARCHAR(255)
nombre_medico	VARCHAR(255)
responsable	VARCHAR(255)
observaciones	VARCHAR(255)
nombre_tipo_transaccion_expediente	VARCHAR(255)
id_orden_salida_expediente	BIGINT(20)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
fk_id_orden_salida_expediente	

especialidad	
id	BIGINT(20)
codigo_especialidad	VARCHAR(16)
nombre_especialidad	VARCHAR(64)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_codigo_especialidad	

tipo_nodo	
id	BIGINT(20)
codigo_tipo_nodo	VARCHAR(16)
nombre_tipo_nodo	VARCHAR(64)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_codigo_tipo_nodo	

rol_usuario	
id	BIGINT(20)
id_usuario	BIGINT(20)
id_rol	BIGINT(20)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_rol_usuario	
fk_id_rol_ru	

patologia	
id	BIGINT(20)
codigo_patologia	VARCHAR(16)
nombre_patologia	VARCHAR(64)
id_especialidad	BIGINT(20)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
uq_codigo_patologia	
fk_especialidad	

renglon_orden_salida_expediente	
id	BIGINT(20)
status_expediente	VARCHAR(255)
numero_historia	VARCHAR(16)
nombre_paciente	VARCHAR(255)
apellido_paciente	VARCHAR(255)
cedula_paciente	VARCHAR(255)
numero_solicitud	VARCHAR(16)
id_orden_salida_expediente	BIGINT(20)
usuario_creacion	VARCHAR(16)
fecha_creacion	TIMESTAMP
usuario_modificacion	VARCHAR(16)
fecha_modificacion	TIMESTAMP
version	BIGINT(20)
Indexes	
PRIMARY	
fk_id_orden_salida_expediente_rose	