Universidad de las Ciencias Informáticas

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Título: Desarrollo de reportes a partir de trazas de la bitácora del Sistema de Información Hospitalaria del Centro de Informática Médica

Autores: Dennis Barrera Pérez

Keilor Martínez González

Tutores: Ing. Angélica de la Caridad Vázquez Rúa

Ing. Yoandris Viquillón Romero

La Habana, junio 2014

"Año 56 de la Revolución"

"La innovación distingue a los líderes de los seguidores"

Steve Jobs

Declaración de autoría

Declaramos que somos los únicos autores de la presente investigación y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 19 días del mes de junio del año 2014.

Keilor Martínez González
Autor

Dennis Barrera Pérez
Autor

Ing. Angélica Vázquez Rúa
Tutor

Ing. Yoandris Viquillón Romero
Tutor

Agradecimientos

Agradecemos a la Revolución por darnos la oportunidad y los medios para convertirnos en ingenieros y en especial a Fidel Castro por crear este centro de altos estudios donde nos formamos como profesionales.

A los tutores Angélica y Yoandris, por darnos su apoyo incondicional, por estar ahí cuando los hemos necesitado y por confiar en nosotros.

Al tribunal y oponente por apoyarnos en todo momento y ayudarnos a graduarnos con resultados satisfactorios.

De Dennis

A mi mamá Miriam, por darme su amor incondicional, dedicación y por estar ahí cuando más la he necesitado.

A mi Abuela Mercedes por apoyarme en todo momento y por tratar de entender lo que es la programación.

A mi papá, Pedro por estar siempre presente y guiarme por el buen camino.

A mi padrastro Raúl por ser mi segundo padre y por confiar en mí en todo momento.

A mi tía Mayiln porque no dejó que la distancia nos separara.

A mis tíos, por comportarse como mis padres y darme fuerzas para seguir adelante

A mis primos, por dejar de ser primos para convertirse en hermanos y especialmente a Leoban por guiarme por el buen camino y aconsejarme en los momentos difíciles.

A toda mi familia y amistades que aunque no los mencione los quiero a todos.

A mi novia, por confiar en mí y por creer que el poder de la atracción existe y que podemos atraer las cosas buenas.

A mi compañero de tesis, Keilor que sin él este trabajo no hubiese sido posible.

Mis profesores de toda la vida Rosa, Toni y Maricel por enseñarme y ayudarme en el plano profesional y personal.

A mis compañeros del apto y aula por estar en las buenas y las malas, Félix, el Liusve, Oscar, Arel, Yairiel, Celio y los que ya no están: Jose, el Niño y demás.

Y todos los que al menos una vez me preguntaron ¿y la tesis?

De Keilor

A mis padres Nancy y Miguel por ser padres ejemplares y por guiarme toda la vida hasta llegar a convertirme en un profesional y prepararme para poder hacer mi propio futuro con mis manos.

A mi mariposita Keila que aunque aún no sabe que ha hecho más por mí que yo por ella cuando sea grande leerá estas letras y comprenderá que ha sido la persona más pequeña pero más importante en los últimos momentos de mi carrera.

A mi compañera de vida y madre de mi mayor tesoro Lianet que se ha mantenido a mi lado bajo cualquier circunstancia dándome su apoyo incondicional.

A mis abuelos Mirta, Lugo, Ortelia, Segundo, Miguel y Vivian que de diferentes formas me han dado todo lo que ha estado a su alcance para que me convierta en Ingeniero.

A mis tíos Isabelita, Lutgardito, Luisita, Nuelvis, Henry y Onel que de muchas maneras me han aportado durante mis años de estudio.

A mi primo-hermano Yuniesky que siempre me ha estado guiando para que me convierta en un hombre de los que necesita estos tiempos.

A mis primos Ailín, Lis Amanda, Christian y Andy que junto a mis sobrinos Melisa y Juan Pablos me han dado alegría para seguir adelante y así poderles ser de ejemplo algún día.

A mis amigos Ernestos y Nioelvis que a pesar de no estar presente estos 5 años en la universidad siempre me alentaron durante la carrera.

A mi amigo Celio que ha estado compartiendo conmigo los bueno y malos momentos y ayudándome en todo lo que ha estado a su alcance, a Lesly que me cuidó como si yo fuera parte de su familia, a Dennis que además realizar la tesis conmigo siempre estaba presente cuando yo no estaba, a Félix y Jorge que compartieron sus dotes conmigo, a Marco que siempre me ha mantenido en sus plegarias y a todo el resto de mis amigos que sería imposible ponerlos por la cantidad de nombres y aun sin estar escritos con el hecho de que yo los llame amigos saben que son personas importante en mi vida.

Dedicatoria

De Dennis

A mi mamá que quiero con la vida, por traerme a este mundo y por depositar su confianza en mí. Te amo mucho.

A mi abuelita por quererme tanto.

A mi hermanito para que pueda seguir mis pasos.

A mi novia que es mi tesoro.

A mi familia y amigos en general.

De Keilor

A mi mamá y papá que son los ángeles que me guían para que me convierta en un hombre de bien.

A mi mariposita Keila que se ha convertido en mi sentido de vivir.

A mi media naranja.

A mis abuelos y demás familiares.

A todos mis amigos.

Datos de contacto

Ing. Angélica Vázquez Rúa

Especialista B, graduada en el año 2011 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Pertenece al departamento Sistemas Especializados en Salud del Centro de Informática Médica, donde se desempeña como Programadora del proyecto Telemedicina.

alvazquez@uci.cu

Ing. Yoandris Viquillón Romero

Recién graduado en adiestramiento, graduado en el año 2012 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Pertenece al departamento Sistemas Especializados en Salud del Centro de Informática Médica, donde se desempeña como Programador del proyecto Telemedicina.

yviquillon@uci.cu

Resumen

Los Sistemas de Información Hospitalaria constituyen una vía para la optimización de los recursos humanos y materiales en las instituciones hospitalarias, que generalmente cuentan con un conjunto de funcionalidades para la generación de reportes en dependencia del área que se requieran. En el Centro de Informática Médica se desarrollan productos de alta calidad como es el Sistema de Información Hospitalaria, el mismo está compuesto por 16 módulos que se encargan de gestionar la información en las diferentes áreas de un hospital; sin embargo dicho sistema no cuenta con funcionalidades que les permitan a los administradores la generación de reportes a partir de las trazas de la bitácora.

La presente investigación tiene como objetivo el desarrollo de funcionalidades en el Sistema de Información Hospitalaria, para la obtención de reportes a partir de las trazas de las acciones que realizan los usuarios en dicho sistema. Para la realización de dichas funcionalidades se utilizó como metodología de desarrollo el Proceso Unificado de Desarrollo. Para el modelado se empleó la herramienta Visual Paradigm 8.0, que utiliza UML como lenguaje de modelado, como entorno de desarrollo integrado Eclipse SDK 3.4.2 que integra Java como lenguaje de programación. Además se empleó como Sistema Gestor de Base de Datos PostgreSQL.

A partir de las herramientas utilizadas, se desarrollaron funcionalidades para la generación de reportes a partir de las trazas de la bitácora. Se les permitirá a los administradores monitorear las acciones realizadas sobre el Sistema de Información Hospitalaria.

Palabras clave: bitácora, módulos, reportes, Sistema de Información Hospitalaria, trazas.

Tabla de contenidos

Índice de tablas

Índice de figuras

Introducción

En un mundo donde la ciencia y la tecnología desempeñan un papel importante, surge y se desarrolla de manera vertiginosa un nuevo concepto conocido como las Tecnologías de la Información y las Comunicaciones (TIC). Las mismas son el conjunto de tecnologías tanto de hardware como de software, que permiten el acceso, estudio, desarrollo, almacenamiento y distribución de la información.

El uso de las TIC se evidencia en todos los sectores de la sociedad, pero uno de los más beneficiados es el de la salud, donde su incorporación en este sector permite el manejo y procesamiento de gran cantidad de información con el objetivo de mejorar la calidad de vida y el bienestar de la sociedad. Su estrecha relación con la medicina da origen a un campo multidisciplinario como es la Informática Médica. Gremy lo define como el conjunto de ciencias, métodos y técnicas que se utilizan para manejar la información médica y completa la definición ampliando que la información médica es aquella que describe el estado de la salud de la población así como el estado del conocimiento en las ciencias de la salud (Gremy, 1998).

El Programa de Informatización de la Sociedad Cubana busca poner las TIC al servicio del desarrollo económico y social del país desde una perspectiva de equidad y participación, donde la salud y la educación son pilares esenciales. Este programa persigue promover el uso masivo de estas tecnologías a escala nacional, a partir de los objetivos estratégicos generales que el país se ha propuesto. Este pretende impulsar de manera coherente a todos los sectores, con una identificación precisa de los actores de la Sociedad de la Información. (1)

Es en dicho contexto que surge la Universidad de Ciencias Informáticas (UCI) con la misión de formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática. Producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación. Servir de soporte a la industria cubana de la informática. (2)

La universidad está conformada por varios centros, entre los que se encuentra el Centro de Informática Médica (CESIM), encargado de desarrollar productos, servicios y soluciones médicas de alta calidad y competitividad para el mejoramiento de la calidad de la atención médica. Dentro de los productos en desarrollo que posee el centro se encuentra el Sistema de Información Hospitalaria (HIS).

HIS es un sistema que permite la gestión médica de hospitales y centros de salud. Incluye la gestión completa de pacientes y sistemas departamentales mediante 16 módulos, brindando información en línea como apoyo a la toma de decisiones, lo cual contribuye a la mejora de la calidad de los procesos,

la asistencia y atención a los pacientes. Es un sistema altamente configurable con soporte para estándares terminológicos y de comunicación internacionales establecidos en materia de salud. (3)

Si bien es cierto que un sistema integral de información sanitaria pone en manos del personal especializado en salud potentes herramientas para gestionar la información, en ocasiones resulta, que dichas herramientas se presentan un tanto abstractas al usuario final, exigiendo de este un esfuerzo adicional para su uso.

La bitácora del HIS, es el lugar que almacena las trazas de las acciones realizadas por cada usuario, agrupadas por año, mes, día, Dirección de Protocolo de Internet (IP, por sus siglas en inglés), usuario, sesión, módulo y acción. En la actualidad resulta poco ilustrativo y complejo de operar, por lo que el administrador tiene que navegar por un árbol para obtener los datos referentes a las acciones de los usuarios en el sistema.

Por lo anteriormente expuesto se plantea como **problema a resolver** de la presente investigación: ¿Cómo viabilizar la obtención de reportes a partir de las trazas de la bitácora del Sistema de Información Hospitalaria del Centro de Informática Médica?

Para el desarrollo de la investigación se define como **objeto de estudio:** El proceso de obtención de reportes para Sistemas de Información Hospitalaria.

Enmarcado en el **campo de acción**: El proceso de obtención de reportes a partir de las trazas de la bitácora del Sistema de Información Hospitalaria del Centro de Informática Médica.

Para resolver el problema identificado se propone el siguiente **objetivo general**: Desarrollar funcionalidades en el Sistema de Información Hospitalaria del Centro de Informática Médica, para la obtención de reportes a partir de las trazas de las acciones que realizan los usuarios en dicho sistema.

Para alcanzar el objetivo planteado se proponen las siguientes **tareas de la investigación**:

1. Analizar los referentes teóricos relacionados con sistemas de información hospitalaria a nivel internacional y nacional que generen reportes a partir de trazas estableciendo comparaciones con la investigación en curso.
2. Analizar los procesos relacionados con las acciones que se realizan actualmente en la bitácora del Sistema de Información Hospitalaria que originaron la situación problemática.
3. Asimilar las tecnologías, metodología y herramientas para el desarrollo de la solución.

4. Realizar los artefactos correspondientes a los flujos de trabajo: Modelado del negocio, Requerimientos, Análisis y Diseño e Implementación propuestos por la metodología de desarrollo Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés).
5. Desarrollar las funcionalidades que respondan a las necesidades del cliente aplicando las pautas de diseño establecidas por el Centro de Informática Médica.

Con el desarrollo de funcionalidades en el Sistema de Información Hospitalaria, para la generación de reportes a partir de las trazas de las acciones que realizan los usuarios en dicho sistema, se esperan obtener los siguientes beneficios:

- ✓ Incremento del monitoreo y control por parte de los administradores, de las acciones de los usuarios en el Sistema de Información Hospitalaria a partir de los reportes generados.
- ✓ Promoción del uso de software libre en el país y reducción de los costos monetarios debido a que las tecnologías empleadas en el desarrollo de la solución son de código abierto.
- ✓ Aumento de la competitividad del Sistema de Información Hospitalaria en el mercado internacional, al contar con funcionalidades para la generación de reporte a partir de las trazas de los usuarios en el sistema.

Para la realización de la investigación en curso se siguió una estrategia descriptiva, donde se reflejaron los aspectos esenciales y más significativos del objeto de investigación. Se emplearon varios métodos científicos clasificados en teóricos y empíricos, así como una técnica para la recopilación de información. Como métodos teóricos se aplicaron los siguientes:

- **Histórico-Lógico:** este método se basa en examinar teóricamente cómo ha evolucionado un fenómeno en un período de tiempo determinado. En la actual investigación se utilizó en la búsqueda de los antecedentes relacionados con la generación de reportes, en sistemas de información hospitalaria a nivel internacional y nacional.
- **Analítico-Sintético:** se utilizó en el análisis de las teorías y documentos relacionados con la generación de reportes en sistemas de información hospitalaria; permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

Como método empírico se utilizó:

- **La observación:** se realizó a sistemas de información hospitalaria con el objetivo de seleccionar aquellos sistemas que contaran con funcionalidades para la generación de reportes a partir de trazas de la bitácora y resultaran de utilidad para la realización de la investigación.

Se utilizó la siguiente técnica para la recopilación de la información:

- **La entrevista:** Se realizó a un especialista del centro CESIM con el objetivo de obtener criterios para comprender los elementos necesarios para el desarrollo de la solución informática.

El documento de la presente investigación está estructurado de la siguiente manera:

Capítulo 1: Fundamentación Teórica:

Contiene un estudio del estado del arte relacionado con la generación de reportes a partir de las trazas que se generan en sistemas de información hospitalaria a nivel mundial. Posee un grupo de características correspondientes a las tecnologías, metodología y herramientas a utilizar para el desarrollo de la solución informática.

Capítulo 2: Características del Sistema:

Como parte de una propuesta de solución informática para resolver la problemática, se describen las principales características del sistema. Se lleva a cabo la realización del Modelo de dominio, la especificación de los requerimientos funcionales y no funcionales, así como la descripción de los casos de uso del sistema.

Capítulo 3: Análisis y Diseño del Sistema:

Está compuesto por los aspectos relacionados con el análisis y diseño de la solución propuesta. Se representan los diagramas de clase de análisis y diagramas de interacción. Se definen los patrones de diseño y estilos arquitectónicos empleados en la solución con una breve descripción de sus características.

Capítulo 4: Implementación:

Se introduce el flujo de trabajo de implementación, partiendo de los resultados obtenidos en el diseño. Se detalla la implementación del sistema propuesto a través del diagrama de despliegue y diagrama de componentes. Se presentan los elementos correspondientes al Modelo de Datos, así como aspectos a tener en cuenta para la seguridad del sistema. Se presentan las estrategias y estilos de codificación a utilizar durante la fase de implementación.

CAPÍTULO 1. Fundamentación teórica

En el presente capítulo se ofrece una introducción a los términos asociados a la investigación para un mejor entendimiento de los temas a tratar. Se exponen los resultados de un estudio del estado del arte relacionados con la generación de reportes a partir de las trazas en sistemas de gestión hospitalaria, en el ámbito nacional e internacional. Se presentan las principales tecnologías, metodología y herramientas a utilizar para el desarrollo de la solución informática.

1.1 Términos asociados a la investigación

Reporte: Un reporte es un informe o una noticia. Este tipo de documento (que puede ser impreso, digital, audiovisual, etc.) pretende transmitir una información, aunque puede tener diversos objetivos. Existen reportes divulgativos, persuasivos y de otros tipos. El reporte puede ser la conclusión de una investigación previa o adoptar una estructura de problema-solución en base a una serie de preguntas. En el caso de los informes impresos, el texto suele ir acompañado por gráficos, diagramas, tablas de contenido y notas al pie de página. (4)

Bitácora: Herramienta que permite registrar, analizar, detectar y notificar eventos que sucedan en cualquier sistema de información utilizado en las organizaciones. Permite guardar las transacciones realizadas sobre una base de datos en específico, de tal manera que estas transacciones puedan ser auditadas y analizadas posteriormente. (5).

Sistema de información: Un sistema de información se puede definir técnicamente como un conjunto de componentes relacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control en una organización. (6)

1.2 Antecedentes en el ámbito nacional e internacional

En el mundo existen diversos sistemas de información hospitalaria, los mismos están orientados a satisfacer las necesidades de generación de información, para almacenar, procesar y reinterpretar datos médico-administrativos de la cualquier institución hospitalaria. Para un estudio a profundidad de los sistemas existentes, se realizó una **guía de observación** con el objetivo de conocer si los mismos pueden

ser tomados como base para las funcionalidades que se desean desarrollar, dicha guía se detalla en el (**Anexo 1**). A continuación se presentan los sistemas estudiados.

1.2.1 Sistema Integral Hospitalario SIHO

El Sistema Integral Hospitalario (SIHO) es una solución modular que tiene como objetivo fundamental, dar soporte a la operación y administración de una institución hospitalaria. SIHO está compuesto por 16 módulos, los cuales se encuentran relacionados entre sí para mantener un flujo de información constante y rápida, que permita disponer de información confiable y oportuna, indispensable para la toma de decisiones. (7) SIHO fue creado para satisfacer las necesidades de hospitales y clínicas, por ende, está diseñado para operar las 24 horas del día sin interrupción, los 365 días del año. (8) A continuación se muestran los tipos de reportes que se generan en el sistema.

Reportes

En el sistema se generan reportes en varios módulos como: Admisión, Servicios auxiliares, Laboratorio Nómina, Compras, Inventario, Cuentas por cobrar, Expediente clínico entre otros, generando en este último los reportes siguientes:

- Concentrado de cirugías, el cual puede ser definido por rango de fechas, tipo de paciente (interno o de cirugía ambulatoria), edad, sexo, cirujano, anestesiólogo, tipo de cirugía (mayor, menor o ambas), cirugía y tipo de anestesia.
- Concentrado de pacientes atendidos en área de urgencias por rango de fechas o médico que atendió.
- Reporte estadístico que contabiliza las defunciones por diagnóstico.
- Reporte estadístico que contabiliza los diagnósticos de pacientes al ingreso.

Además de los reportes propios del expediente según la Norma Oficial Mexicana: NOM-168-SSA1-1998 del expediente clínico.

Sistemas y seguridad

SIHO tiene este módulo para que el departamento de sistemas controle y lleve el monitoreo las transacciones que se realizan en los diferentes departamentos. Cuenta con diferentes niveles de seguridad en el acceso a cada una de las opciones. Es posible configurar diferentes usuarios según su

nivel, funciones y responsabilidades. Cada vez que un usuario hace uso del SIHO, queda registrado, quién realizó dicha transacción, la fecha y la hora, pudiendo de esta forma rastrear cada movimiento tanto clínico como administrativo. Cuenta con una doble seguridad para el expediente clínico, de tal forma que si un médico o una enfermera, inician una captura, debe ser el mismo usuario quien lo termine. Sin embargo, este sistema no cuenta con funcionalidades para la obtención de reportes a partir de las acciones de los usuarios en el mismo.

1.2.2 Sistema integral para la administración hospitalaria xHosp

xHosp es un sistema de información hospitalario desarrollado totalmente en México para el apoyo operacional y el control administrativo integral de un hospital o clínica.

Permite el control administrativo de pacientes hospitalizados durante su internamiento, de pacientes de corta estancia o ambulatorios así como de pacientes externos a una unidad hospitalaria. Controla el acceso del paciente a las diferentes áreas hospitalarias registrando las prestaciones otorgadas al paciente directamente en el servicio donde se otorgaron. Permite la explotación de información médica con fines estadísticos o de la información administrativa con fines de control. Por medio de interfaces puede realizarse la transferencia de información administrativa hacia los sistemas contables de la compañía. (9)

xHosp fue diseñado desde su inicio como un sistema modular y escalable que puede ser instalado en la mayoría de las instituciones hospitalarias, desde clínicas regionales pequeñas hasta los grandes centros médicos institucionales o privados.

Características generales:

- Aprovecha todas las ventajas de una aplicación diseñada y desarrollada para Windows como la interfaz de usuario gráfica amigable y la capacidad de compartir la información con otras aplicaciones de Windows como Word, Excel o Access entre otras.
- Permite la utilización de prácticamente cualquier base de datos para el almacenamiento de la información de pacientes y resultados.

Módulos que lo componen:

- Admisión de pacientes
- Transferencia y egreso de pacientes
- Censo y control de cuartos
- Cargo automático de cuartos
- Áreas de atención a hospitalizados
- Programación y control de quirófanos
- Laboratorio de análisis clínicos
- Farmacias
- Facturación y Cobranzas
- Caja General (Pagos)
- Consulta Externa
- Gabinete de Imagenología
- Banco de Sangre
- Interfaz Contable
- Administración

El módulo Administración se encarga de:

- Seguridad de usuarios a 3 niveles
- Aumento automático de precios
- Actualización de catálogos
- Configuración del sistema
- Reportes administrativos y estadísticas médicas

Este sistema en su módulo de Administración posee una funcionalidad de Reportes administrativos y estadísticas médicas, pero no cuenta con una funcionalidad para la generación de reportes a partir de las acciones de los usuarios en el sistema.

1.2.3 PHPClinica

PHPClinica es un sistema de información hospitalaria desarrollado en lenguajes de programación web como: HTML, Ajax, Css, JavaScript, principalmente PHP y MySQL como motor de base de datos por sus grandes prestaciones de seguridad, rapidez e interacción con el lenguaje PHP.

El sistema está conformado en su primera versión por trece módulos donde ochos de ellos son de registrar los datos de los pacientes, dos que su principal función es mostrar indicadores estadísticos, uno de los medios de diagnósticos y los dos últimos son de configuración y administración del sistema. Además existen cincos niveles de seguridad de usuarios para que estos no tengas los mismos privilegios a la hora de interactuar con el sistema. (10)

Este sistema permite almacenar en su base de datos información de quien se registró como:

- Usuario.
- Número de IP de la máquina
- Fecha actual

El módulo de Administración es el encargado de controlar las trazas que deja el sistema para mantener la seguridad en los datos que son registrados. Sin embargo, este sistema no cuenta con funcionalidades para la obtención de reportes a partir de las trazas de las acciones de los usuarios en el mismo.

1.2.4 Sistema Básico de Gestión Hospitalaria GalenHos

GalenHos ha sido diseñado con el propósito de apoyar a los establecimientos de salud en el correcto registro de información clínica y administrativa y la generación de información gerencial que permita una adecuada toma de decisiones. De esta forma, el uso de GalenHos contribuirá a hacer más eficiente la gestión de los procesos operativos críticos de un hospital, específicamente los de consulta externa, hospitalización, emergencia, archivo clínico y facturación, así como a optimizar el uso de los establecimientos públicos. (11)

Objetivos:

- Proveer información para la toma de decisiones
- Optimizar esfuerzos del personal de salud mediante un sistema articulado de gestión
- Disminuir trámites y tiempos de espera para los pacientes

El sistema GalenHos se ha desarrollado utilizando el lenguaje de programación Visual Basic 6.0, por la facilidad de esta herramienta para el diseño y la implementación de interfaces gráficas intuitivas y amigables. Utiliza una arquitectura cliente-servidor de dos capas, esta arquitectura está compuesta por los siguientes elementos:

- Un servidor de base de datos, de las siguientes características:
  - Motor de base de datos SQL Server 2000 y sistema operativo Windows 2000/2003
- Terminales clientes donde se ejecutará la aplicación GalenHos, de las siguientes características:
  - Sistema Operativo Windows 95, Windows 2000 Professional, Windows XP Professional.

Módulos que lo integran:

- Consulta Externa
- Hospitalización
- Emergencia
- Facturación
- Archivo Clínico
- Farmacia
- Laboratorio

A continuación se muestra una tabla comparativa entre los sistemas estudiados:

Sistemas	Tecnología de desarrollo		Tipo de aplicación		Generan reportes	País
	Libre	Propietaria	Web	Escritorio		
SIHO	x	-	x	-	x	México
XHosp	-	x	-	x	x	México
PHPClinicas	x	-	x	-	-	Cuba
GalenHos	-	x	-	x	x	Perú

Tabla 1. Tabla comparativa de sistemas relacionados con la generación de reportes.

1.2.5 Valoración de los sistemas estudiados

Luego del análisis de los sistemas existentes en el ámbito nacional e internacional relacionados con el objeto de estudio, se ha concluido lo siguiente:

- Algunos de los sistemas como: xHosp y GalenHos están desarrollados sobre la base de software propietario, por lo que su código fuente tiene acceso restringido, limitando al país de poder usarlos, modificarlos o redistribuirlos.
- Los sistemas estudiados no cuentan con la funcionalidad de generación de reportes a partir de las trazas de las acciones que realizan los usuarios en los mismos, lo que dificulta tener un control estricto de dichas acciones por parte de los administradores. Por lo tanto, dichos sistemas no se adaptan o pueden ser empleados como base para las funcionalidades que se desean desarrollar.

Después de valorar los sistemas estudiados y determinar que ninguno de ellos puede ser empleado como base para el desarrollo de las funcionalidades especificadas en el objetivo general, se procede a crear dichas funcionalidades. La creación de dichas funcionalidades debe estar guiada por una metodología de desarrollo. Por tal motivo a continuación se introduce la metodología a emplear en la solución propuesta, la cual acompañará al software durante su ciclo de vida.

1.3 Metodología de desarrollo

Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. (12) La misma se fundamenta sobre tres pilares básicos: qué hacer y en qué orden, cómo deben realizarse las tareas y con qué pueden llevarse a cabo.

La metodología a utilizar en la presente investigación es el Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés). RUP es un modelo en fases que identifica cuatro fases diferentes en el proceso del software. Sin embargo, a diferencia del modelo en cascada donde las fases se equiparan con las actividades del proceso, las fases en el RUP están mucho más relacionadas con asuntos de negocio más técnicos. (Sommerville, 2005) Sus fases son las siguientes:

Inicio: El objetivo de esta fase es establecer un caso de negocio para el sistema. Se deben identificar todas las entidades externas (personas y sistemas) que interactuarán con el sistema y definir estas interacciones.

Elaboración: Los objetivos de esta fase son desarrollar una comprensión del dominio del problema, establecer un marco de trabajo arquitectónico para el sistema, desarrollar el plan el proyecto e identificar los riesgos claves del proyecto.

Construcción: Comprende el diseño del sistema, la programación y las pruebas. Durante esta fase se desarrollan e integran las partes del sistema. Al terminar esta fase se debe tener un software operativo y la documentación correspondiente lista para entregarla a los usuarios.

Transición: Esta fase se ocupa de mover el sistema desde la comunidad de desarrollo a la comunidad de usuario y hacerlo trabajar en un entorno real.

Características:

- Dirigido por casos de uso: los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos.
- Centrado en la arquitectura: la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción.
- Iterativo e Incremental: propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. (Gallego, 2007)

Para un mejoramiento de la calidad de los procesos y optimización de los flujos de trabajo establecidos por la metodología presentada, se adopta el Modelo de Madurez de Capacidades (CMMI, por sus siglas en inglés), el siguiente acápite explica a profundidad cómo se emplea dicho modelo en la solución propuesta.

1.4 Guía de desarrollo de Software

CMMI es un modelo de madurez de mejora de los procesos para el desarrollo de productos y de servicios. Consiste en las mejores prácticas que tratan las actividades de desarrollo y de mantenimiento que cubren el ciclo de vida del producto, desde la concepción a la entrega y el mantenimiento. (13)

CMMI se utiliza como modelo de calidad el cual cuenta con 22 áreas de procesos que se distribuyen dentro de 5 distintos modelos de madurez, tales como: Inicial, Administrado, Definido, Cuantitativamente administrado y Optimizado. El enfoque en el que se definen estos niveles de madurez para las organizaciones se denomina “Representación Escalonada”.

El centro CESIM se encuentra certificado en el nivel dos de CMMI, el cual está conformado por las siguientes áreas de procesos:

- Planeación del Proyecto (PP).
- Monitoreo y Control del Proyecto (PMC).
- Administración de Acuerdos con Proveedores (SAM).
- Medición y Análisis (MA).
- Aseguramiento de la Calidad de Procesos y productos (PPQA).
- Administración de la Configuración (CM).
- Administración de Requisitos (REQM).

La utilización de CMMI se evidencia en la investigación en curso a partir de modificaciones realizadas a los artefactos establecidos por RUP, en correspondencia con las diferentes áreas de procesos establecidas por el nivel 2 de dicho modelo de mejora.

Una vez asimilada la metodología a emplear y estudiado CMMI como plan de mejora, se describen las tecnologías y herramientas definidas por el centro CESIM utilizadas en el desarrollo de sus productos y que se emplean en el transcurso de la investigación.

1.5 Herramientas y tecnologías

El ambiente de desarrollo de software constituye un mecanismo que permiten guiar y simplificar los procesos en la elaboración de cualquier sistema. El mismo está compuesto por herramientas y tecnologías que facilitan el proceso de desarrollo de un sistema.

Java 2

Lenguaje de programación desarrollado por James Gosling y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis proviene de lenguajes como C y C++, pero tiene menos facilidades de bajo nivel que estos.

Algunas de sus características son:

- Es un lenguaje orientado a objetos y basado en clases.
- Permite el desarrollo de aplicaciones basado en la arquitectura Cliente Servidor, lo que posibilita las tareas puedan ser ejecutadas simultáneamente por varios ordenadores.
- Es independiente de la plataforma, por lo que un programa escrito en lenguaje Java puede ejecutarse en cualquier hardware.
- Está distribuido bajo la licencia GNU GPL.

UML 2.1

UML, “es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.” (Dugarte, y otros, 2009). Surge en 1995 como Método Unificado transformándose posteriormente en UML a partir de una reorientación del mismo. Se basa en un conjunto de elementos gráficos que dan lugar a diagramas. Es utilizado para describir lo que hará un sistema pero no explica cómo se debe implementar.

Sus características principales son (14):

- Permite la conexión con lenguajes de programación a través de la ingeniería directa e inversa.
- Permite documentar todos los artefactos de un proceso de desarrollo.
- Es independiente del proceso, aunque para utilizarlo óptimamente, se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.
- El tiempo invertido en el desarrollo de la arquitectura se minimiza.

- La detección y resolución de errores se agiliza siempre y cuando, se haga uso de herramientas adecuadas de diagnóstico y depuración.

Eclipse SDK 3.4.2

Entorno de desarrollo integrado desarrollado originalmente por IBM. Es utilizado principalmente para crear Aplicaciones de Cliente Enriquecido (RCP, por sus siglas en inglés). Eclipse fue utilizado para desarrollar entornos de desarrollo integrado (IDE, por sus siglas en inglés) como es el IDE de Java llamado Java Development Toolkit (JDT, por sus siglas).
Sus características principales son (15):

- Ofrece diversos marcos de trabajo que son utilizados por el programador para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software.
- Emplea módulos o plug-in para proporcionar toda su funcionalidad al frente de la Plataforma de Cliente Enriquecido.
- El SDK de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java.
- Hace uso de un espacio de trabajo permitiendo modificaciones externas a los archivos mientras se refresca el espacio de trabajo correspondiente.

XHTML 1.0

El Lenguaje de Marcado de Hipertexto Extensible (XHTML, por sus siglas en inglés) es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de remplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. Permite mantener una separación entre el contenido y el diseño. Fue diseñado para ser funcional no solo de navegadores sino de otros dispositivos como teléfonos móviles y portátiles. (16)

CSS 2.0

Hojas de Estilo en Cascada (CSS, por sus siglas en inglés), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la

presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. (17)

Enterprise JavaBeans 3.0

Enterprise Java Beans (EJB, por sus siglas en inglés) es una plataforma para construir aplicaciones de negocio portables, reusables y escalables usando el lenguaje de programación Java. Un EJB es una porción de código que se ejecuta en un contenedor EJB, que no es más que un ambiente especializado (runtime) que provee determinados componentes de servicio. Los EJBs pueden ser vistos como componentes, desde el punto de vista que encapsulan comportamiento y permite reutilizar porciones de código, pero también pueden ser vistos como un marco de trabajo, ya que, desplegados en un contenedor, proveen servicios para el desarrollo de aplicaciones enterprise, servicios que son invisibles para el programador y no afectan la lógica de negocio. (18)

Java Persistence API 1.0

Java Persistence API (JPA, por sus siglas en inglés) fue desarrollado por un grupo de expertos de EJB 3.0 como parte de JSR 220. Puede utilizarse directamente en aplicaciones web y aplicaciones clientes; incluso fuera de la plataforma Java EE, por ejemplo, en aplicaciones Java SE. Proporciona un modelo de persistencia para mapear bases de datos relacionales en Java. (19)

Está conformado por tres áreas (20):

- El Java Persistence API
- El lenguaje de query
- El mapeo de los metadatos objeto/relacional

Hibernate 3.3

Herramienta de Mapeo Objeto-Relacional (ORM, por sus siglas en inglés) para la plataforma Java, que proporciona el mapeo de atributos entre una base de datos y el modelo de objetos de una aplicación mediante archivos declarativos (XML), que permiten establecer estas relaciones. (21) Este término conocido como ORM consiste en la técnica de realizar una transformación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa. Sus características principales son:

- Posee una gran flexibilidad en cuanto al esquema de tablas utilizado, lo cual permite que su uso sea adaptable fácilmente en bases de datos existentes.
- Proporciona capacidades para la obtención y almacenamiento de datos de la base de datos que reducen el tiempo de desarrollo.
- Ofrece un lenguaje de consulta de datos llamado HQL (Hibernate Query Language, por sus siglas en inglés).
- Genera las sentencias SQL (Structured Query Language) y libera al desarrollador del manejo manual de los datos que resultan de la ejecución.
- Es una tecnología de software libre distribuida bajo los términos de la licencia LGPL.

Java Server Faces 1.2

Java Server Faces (JSF, por sus siglas en inglés) es un marco de trabajo para crear aplicaciones java J2EE basadas en patrón Modelo-Vista-Controlador de tipo 1.

Sus características principales son:

- Utiliza páginas JSP para generar las vistas, añadiendo una biblioteca de etiquetas propia para crear los elementos de los formularios HTML.
- Forma parte del estándar J2EE. En efecto, hay muchas alternativas para crear la capa de presentación y control de una aplicación web java, como Struts y otros frameworks, pero solo JSP forma parte del estándar.(22)

RichFaces 3.3.1

RichFaces es una librería de componentes visuales para JSF, escrita en su origen por Exadel y adquirida por Jboss. Además, RichFaces posee un marco de trabajo avanzado para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF. Brinda facilidades de validación y conversión de los datos suministrados por el usuario y administración avanzada de recursos como imágenes, código java script y hojas de estilo en cascada (CSS).

Sus características principales son:

- Soporta facelets.
- Se integra completamente en el ciclo de vida de JSF.
- Al pertenecer a un subproyecto de JBoss, se integra perfectamente con Seam.
- Es un proyecto código abierto, activo y con una comunidad también activa. (23)

Ajax

Asynchronous JavaScript and XML, no es un lenguaje de programación, sino un conjunto de tecnologías (HTML-JavaScript-CSS-DHTML-PHP/ASP.NET/JSP-XML) que permiten hacer páginas web más interactivas. Su característica fundamental es permitir actualizar parte de una página con información que se encuentra en el servidor sin tener que refrescar completamente la página. (24)

Ajax4jsf

Es una librería de código abierto que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax sin añadir código JavaScript. Mediante este marco de trabajo se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de recargarla por completo, realizar peticiones al servidor automáticas, control de cualquier evento de usuario. (25)

Facelets 1.1

JavaServer Facelets es un marco de trabajo de código abierto para plantillas basado en la tecnología JSF (JavaServer Faces). Es el sistema de plantillas estándar de facto en la versión 1.2 de JSF y que ahora, en la versión 2, forma parte de la propia especificación. Es de código abierto, distribuido bajo la licencia Apache y constituye una tecnología alternativa de controlador de vista de JSF. Soporta todos los componentes de interfaz de usuario JSF y construye su propio árbol de componentes. (26)

JSP y JSF no se complementan naturalmente, cuando se usan juntos ambos escriben output al response, pero lo hacen de una manera diferente: JSP crea output ni bien encuentra código JSP (es decir procesa los elementos de la página de arriba a abajo), mientras que JSF dicta su propio re-rendering (ya que su ciclo de vida está dividido en fases marcadas). Facelets llena este vacío entre JSP y JSF, siendo una tecnología centrada en crear árboles de componentes y estar relacionado con el complejo ciclo de vida JSF. (27)

JBoss Seam 2.1.1

Marco de trabajo de código abierto que enlaza diferentes tecnologías y estándares java, lo cual garantiza la plena comunicación de los elementos de la capa de presentación, de negocio y de acceso a datos. Con este marco de trabajo basta agregar anotaciones propias de éste a los objetos entidad y sesión de EJB, lo que permite escribir menos código Java y XML. Otra característica relevante es que se pueden hacer validaciones en los POJOs (Plain Old Java Object) y además manejar directamente la lógica de la aplicación y de negocios desde las sessions bean. (28)

JavaEE 5

Java Platform Enterprise Edition es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java que se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. Java EE incluye varias especificaciones de API y define cómo coordinarlas. Se ha convertido en un estándar de la industria de desarrollo de software que permite trabajar aplicaciones empresariales seguras, portables, robustas y escalables, basadas en servidores. (29)

JRE 6

Java Runtime Environment es una máquina virtual de Java y su función es hacer de intermediario entre una aplicación programada en Java y el sistema operativo que se esté usando. De este modo, cualquier aplicación puede funcionar en cualquier sistema operativo que disponga del JRE. (30).

JBoss Application Server 4.2.2

JBoss Application Server es un servidor multiplataforma de aplicaciones J2EE, de código abierto, implementado en Java puro y orientado a la arquitectura de servicios. Por ser una plataforma certificada J2EE, soporta todas las funcionalidades de J2EE 1.4 e incluye servicios adicionales como clustering y persistencia. JBoss es ideal para aplicaciones Java y aplicaciones web. También soporta EJB 3.0, lo que hace el desarrollo de las aplicaciones mucho más simple. Con su utilización se pueden implementar sistemas con facilidad, únicamente es necesario colocar los archivos en el directorio /server/default/deploy**.** (31)

Visual Paradigm 8.0 para UML

Es una herramienta de Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés), que utiliza como lenguaje de modelado UML y que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Se integra con varios IDE como Eclipse, NetBeans, VisualStudio, entre otros. Provee un generador de mapeo de objetos relacionales para los lenguajes de programación Java, PHP y algunos lenguajes del ambiente .Net. (32)

PgAdmin III 1.10.5

Es una aplicación gráfica para gestionar las bases de datos de PostgreSQL, siendo la más completa y popular con licencia de código abierto. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma. (33)

Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas.

PostgreSQL 8.4

Es un Sistema Gestor de Base de Datos (SGBD) relacional. Puede ejecutarse sobre la mayoría de los sistemas operativos actuales. Incluye características orientadas a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, etc. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. (33)

PostgreSQL presenta alta concurrencia, para esto utiliza la tecnología de Control de Concurrencia Multi-Versión (Multiversion concurrency control (MVCC)), con lo que se logra que ningún lector sea bloqueado por un escritor.

A continuación se describen algunas de las herramientas existentes en el mundo para la generación de reportes en sistemas informáticos, con el objetivo de seleccionar la adecuada para el desarrollo de dichos reportes.

Crystal Reports 9

Crystal Reports es un producto de inteligencia empresarial, con alta tecnología para la creación e integración de reportes, con datos provenientes de múltiples SGBD, entre los que se encuentran: PostgreSQL, MySQL, Oracle, DB2, MSSQL, Informix, InterBase, Sybase y Frontbase. Permite crear contenido interactivo con calidad de presentación en la plataforma .NET debido a que es la herramienta de elaboración de informes estándar para Visual Studio .NET, lo que ha supuesto una ventaja fundamental para Crystal Reports durante años. (34)

Características:

- Permite diseñar informes ilimitados para uso en aplicaciones de Visual Studio con el diseñador de Crystal Reports integrado.
- Brinda la oportunidad de generar gráficos interactivos y enviarlos mediante la Web, correo electrónico, Microsoft Office, Adobe PDF o incrustado en una aplicación empresarial.

Sistema Operativo: Windows en todas sus versiones, Linux y Mac OS.

Licencia: Crystal Reports es un sistema propietario cuyas licencias de comercialización cuestan $2495. Se distribuye bajo los términos de la EULA, licencia por la cual el uso de un producto sólo está permitido para un único usuario, el comprador.

Active Reports 8

Active Report es una herramienta de plataforma propietaria que permite diseñar y crear reportes de forma fácil y rápida. Se caracteriza por dar soporte tanto a aplicaciones web como de escritorio siendo capaz de emplear una amplia variedad de orígenes de datos. Además soporta gráficos, imágenes y sub-reportes.

Características principales:

- El diseñador de informes se integra en los entornos de desarrollo Visual Studio .NET.
- Permite la exportación de los reportes a los formatos PDF, Excel, RTF, TIFF, XML y HTML.

Licencia: Se distribuye bajo licencia privativa perteneciente a GrapeCity inc.

JasperReports 4.0.2

Herramienta de generación de reportes capaz de utilizar cualquier tipo de datos y producir a partir de ellos documentos de calidad gráfica que pueden ser vistos y exportados. (35)

Es una librería de clases de Java de código abierto desarrollada por Teodor Danciu en el año 2001. Permite añadir funcionalidades de generación de reportes de manera simple y flexible en las aplicaciones desarrolladas en la plataforma Java. Como JasperReports no tiene ninguna dependencia con las librerías de Java asociadas a aplicaciones web, también se puede utilizar en aplicaciones Java de escritorio.

Características principales:

- Permite la inclusión de imágenes y gráficos en los reportes obteniendo una mejor calidad.
- Permite una diagramación flexible de los reportes: Los reportes se pueden dividir en secciones opcionales que son: título del reporte, el encabezado de página, una sección para los detalles del reporte, el pie de página y una sección de resumen que aparece al final del reporte.
- Genera sub-reportes lo que facilita el diseño debido a que se pueden usar estos sub-reportes en otros reportes.
- Los reportes pueden ser exportados a una multitud de formatos como PDF, XLS, RTF, HTML, XML y texto plano.

Sistema Operativo: Multiplataforma

Licencia: El software es libre pues es distribuido mundialmente bajo los términos de GNU (LGPLv3). Está respaldado por una gran comunidad internacional de desarrollo, el proyecto JasperForge.org y la empresa JasperSoft Corporation.

A continuación se muestra una tabla comparativa entre herramientas para la generación de reportes:

Herramientas	SO	Licencia	Formato de Salida
Crystal Reports 9	Windows, Linux y Mac OS.	Propietaria: EULA	XML, PDF, HTML y Microsoft Excel.
Active Reports 8	Windows en todas sus versiones.	Propietaria	PDF, Excel, RTF, TIFF, XML y HTML.
JasperReports 4.0.2	Multiplataforma.	Libre: LGPLv3	PDF, XLS, RTF, HTML, XML y texto plano.

Tabla 2. Comparación entre motores de reportes estudiados.

Valoración de los motores de reportes estudiados

Después de analizadas las características de los diferentes motores de generación de reportes, y establecida la comparación entre ellas, se decide escoger como herramienta a utilizar a JasperReports 4.0.2. El motivo de su elección consiste en que la misma posee un grupo de ventajas como es la gran variedad de formatos de salida que brinda, que es multiplataforma distribuido bajo la licencia LGPL. Para el diseño de informes visuales se decide seleccionar IReport 4.0.2, a continuación se exponen algunas de sus características.

IReport 4.0.2

La herramienta iReport es un constructor/diseñador de informes visuales, poderoso, intuitivo y fácil de usar para JasperReports, está escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, sub-informes, etc. iReport está además integrado con JFreeChart, una de la biblioteca gráficas código abierto más difundida para Java. (37)

Sus características principales son: (37)

- Soporta JavaBeans como orígenes de datos (éstos deben implementar la interface JRDataSource).
- Incluye Wizard's (asistentes) para crear automáticamente informes.

Para un mejor entendimiento en el presente capítulo se presentaron los principales conceptos asociados a la investigación. Se realizó un análisis crítico de algunos sistemas de información hospitalaria existentes en el mundo. El objetivo de dicho estudio fue seleccionar aquellos sistemas que contaran con funcionalidades para la generación de reportes a partir de las trazas, de las acciones realizadas por los usuarios en el sistema. Sin embargo, se concluye que los sistemas estudiados no cuentan con dichas funcionalidades, por lo que se hace necesario el desarrollo de las mismas en una solución informática. Para ello, se presentaron como metodología de desarrollo RUP, como herramienta de modelado Visual Paradigm 8.0, que utiliza UML como lenguaje de modelado, Java 2 como lenguaje de programación, utilizando el entorno integrado de desarrollo Eclipse SDK 3.4.2, todas estas herramientas están definidas por el centro CESIM para el desarrollo de sus aplicaciones y se encuentran descritas en el documento de Arquitectura del Sistema de Información Hospitalaria. Como generador de reporte, se seleccionó Jasper Report 4.0.2 debido a la gran variedad de formatos de salida que presenta.

CAPÍTULO 2. Características del Sistema

En el presente capítulo se describe la modelación del negocio mediante el Modelo de dominio con el objetivo de comprender y describir las clases más importantes dentro del contexto del sistema y sus relaciones. Se definen los requisitos funcionales y no funcionales del sistema. Además, se presenta el Diagrama de Casos de Uso del Sistema con las correspondientes descripciones de los Casos de Uso (CU) para un mejor entendimiento del funcionamiento de la aplicación. Para lograr comprender los procesos relacionados con las acciones que se realizan actualmente en la bitácora del Sistema de Información Hospitalaria que originaron la situación problemática, se presenta a continuación la propuesta solución.

2.1 Propuesta solución

El Sistema de Información Hospitalaria permite la gestión médica de hospitales y centros de salud. Está compuesto por 16 módulos que atienden las diferentes áreas de una institución hospitalaria aumentando la calidad de los servicios prestados. Posee un módulo de Configuración que se encarga de la administración y configuración del sistema. Algunas de sus funcionalidades principales son:

- Configurar módulos
- Seguridad avanzada
- Administrar seguridad
- Usuarios y roles
- Bitácora

En dicho módulo se encuentra la Bitácora del sistema (**Ver Anexo 2**). La misma almacena las trazas de las acciones de los usuarios agrupadas por año, mes, día, IP, usuario, sesión, módulo y acción. Sin embargo dicha bitácora carece de flexibilidad ya que su estructura es rígida en cuanto a la navegación se refiere. En caso de que el administrador desee conocer las acciones que se realizan en el sistema, éste debe navegar por una estructura jerárquica, que en ocasiones omite información relevante como es el módulo accedido y la acción realizada por un usuario en el sistema. Por tal motivo se propone el desarrollo de funcionalidades para la generación de reportes a partir de las trazas de la bitácora, que les permitan a los administradores tener un mayor control de las acciones de los usuarios en el sistema.

Para obtener información más detallada de cómo desarrollar estas funcionalidades, se le realizó una entrevista al Ing. Alain Ramos Medina, especialista "A" en Ciencias Informáticas del centro CESIM (**Ver Anexo 3**).

La realización de un sistema conlleva un grado de complejidad, por lo que para llegar a comprenderlo es necesario modelarlo. (38) En el caso de la presente investigación, se elaboró un modelo de dominio debido a que no se definen procesos de negocio involucrados con la problemática existente.

2.2 Modelo del dominio

Un modelo del dominio es un diccionario visual de términos importantes del dominio y sus acciones, se centra solo en las clases del dominio más importantes dentro del contexto del sistema y las relaciones entre ellas, utiliza la notación UML de diagrama de estructura estática.

Utilizando la notación UML, un modelo de dominio se representa con un conjunto de diagramas de clases en los que no se define ninguna operación. Pueden mostrar (39):

- Objetos del dominio o clases conceptuales.
- Asociaciones entre las clases conceptuales.
- Atributos de las clases conceptuales.

2.2.1 Diagrama de conceptos del dominio

Diagrama de conceptos del dominio

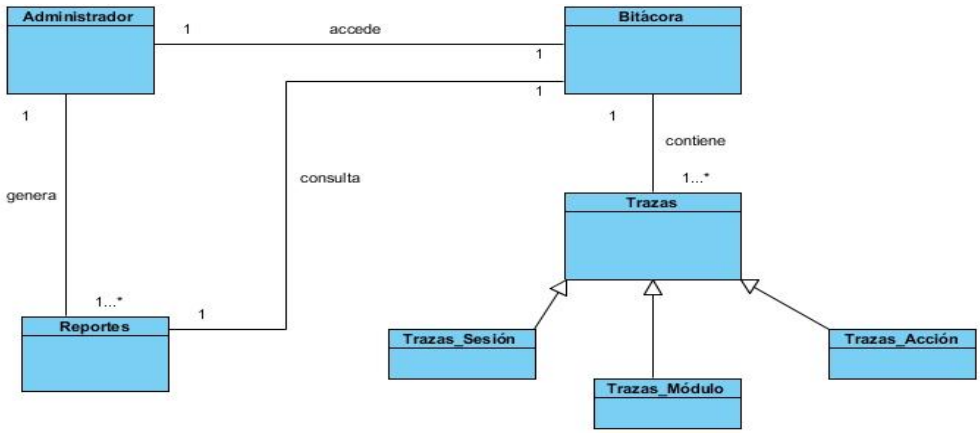


Figura 1. Diagrama de conceptos del dominio.

2.2.2 Descripción de los conceptos del Modelo de dominio

Reportes: Informes que se generan a partir de cualquier combinación de parámetros seleccionados por el usuario (usuario, módulo, IP, fecha, hora, acción y atributo) dando la posibilidad de aplicar filtros a cada uno de ellos.

Trazas: Concepto asociado a la información almacenada en el sistema referente a la secuencia de acciones que realizan los usuarios en el mismo.

Trazas_Sesión: Contiene el identificador de la sesión, además de información referente a la misma como es fecha de inicio, fecha de fin y el número de IP desde el cual se inicia dicha sesión.

Trazas_Módulo: Concepto correspondiente a las trazas de los módulos accedidos en el sistema, las cuales están compuestas por el identificador de cada módulo accedido.

Trazas_Acción: Agrupa el conjunto de acciones realizadas sobre un determinado módulo del sistema, contiene los identificadores de las mismas, así como información correspondiente a la hora de inicio y fin en que se realizaron dichas acciones.

Bitácora: Concepto asociado al lugar donde se almacenan las trazas de las acciones realizadas por cada usuario, agrupadas por año, mes, día, dirección IP, usuario, sesión, módulo y acción.

Administrador: Usuario con los privilegios necesarios para acceder a la bitácora del sistema y generar reportes con el objetivo de tener un control sobre las acciones que se realizan en el sistema.

EL modelado del sistema constituye un área importante dentro de fase de análisis de requisitos para la construcción de cualquier sistema. A continuación se presentan los componentes correspondientes al modelado del sistema de la investigación en curso.

2.3 Modelado del sistema

Según Pressman, el modelado se encarga de crear los modelos del sistema con el objetivo de tener un mejor entendimiento del flujo de datos y control, el tratamiento funcional, el comportamiento operativo y el contenido de la información. Además permite al ingeniero de software visualizar el sistema a construir. (Pressman, 1998) El mismo parte de definir un conjuntos de requisitos ya sean funcionales y no funcionales que sirvan de guía para el desarrollo de cualquier sistema.

2.3.1 Requisitos funcionales

Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. (40) Estos son seleccionados durante la primera etapa de desarrollo de software y constituyen una forma de representar condiciones o capacidades que deben cumplir un sistema, para satisfacer las necesidades de un contrato, estándar o cualquier documento formal.

A continuación se presenta el listado de todas las funcionalidades que el sistema debe permitir, en el caso de los reportes a generar, los mismos serán el resultado de una combinación de los diferentes parámetros (IP, fecha, hora, acción y atributo), además se mantendrán estables los parámetros **usuario y módulo** de forma tal que estén presentes en cada reporte a generar. Por lo tanto el sistema debe permitir la generación de **31 reportes** a partir de los parámetros seleccionados por el usuario.

RF1. Generar reporte.

RF2. Filtrar reporte.

RF3. Exportar reporte.

El requerimiento funcional Generar Reporte engloba un conjunto de funcionalidades que constituyen cada reporte que debe generar el sistema. Para visualizar el listado de los 31 posibles reportes a generar (**Ver Anexo 4**).

Los reportes a generar siguen una estructura por niveles los cuales están establecidos de la siguiente manera. El primer nivel lo compone la tabla sesión, en el segundo nivel se encuentra la tabla módulo, el tercer nivel contiene a la tabla acción y el cuarto nivel está compuesto por la tabla atributo. Para la generación de un reporte se realiza una consulta a la base de datos, por lo tanto, los parámetros de selección se tomarán a partir del último nivel. A continuación se presentan dos ejemplos de reportes que utilizan el mismo parámetro de selección (hora) tomando en cada caso un valor diferente.

- En un reporte de usuario, módulo y hora, los datos como la hora se seleccionan del último nivel que posee este parámetro, en este caso se selecciona la hora de la tabla sesión que es el elemento que contiene dicho parámetro, este reporte indica la hora en que un usuario accede a un módulo.

- En el caso de generar un reporte por usuario, modulo, hora y acción, el parámetro hora se toma de la tabla acción, indicando la hora en que se realiza una acción determinada por un usuario en un módulo.

2.3.2 Requisitos no funcionales

Los requisitos no funcionales, como su nombre sugiere, son aquellos requisitos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. (41) Los mismos son las propiedades o cualidades que un producto debe tener, que se refieren a las características que hacen al producto usable, rápido o confiable. Para el desarrollo de la presente investigación se definen los siguientes requisitos no funcionales:

RNF1. Usabilidad

La usabilidad se refiere a la capacidad de un software de ser comprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso. La misma no depende sólo del producto, sino también del usuario, de la facilidad con que este interactúe con la herramienta, en menos pasos o más naturales a su formación específica.

De acuerdo a estas condiciones: El sistema debe ser diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido, con una previa preparación. La navegabilidad interna no debe ser compleja, lo que hace que el acceso a las diferentes funcionalidades sea rápido. Debe ser sencillo a la vista de los usuarios, lo que favorece el buen entendimiento y aceptación del producto.

RNF2. Fiabilidad

El sistema debe estar disponible de forma permanente y funcionar sin necesidad de intervención del usuario. El sistema debe mantener tiempos de respuestas en un marco razonable de diez segundos, permitiendo que existan al menos 10 usuarios conectados de forma simultánea para lo cual se deben realizar pruebas de carga y estrés corroborando estos resultados. Los reportes generados deberán mostrar toda la información existente en la base de datos correspondiente a dicho reporte.

RNF3. Eficiencia

El sistema debe minimizar el volumen de datos en las peticiones y optimizar el uso de recursos críticos como la memoria. Para ello se debe potenciar como regla, guardar en la memoria caché los datos y

recursos de alta demanda. Se debe respetar las buenas prácticas de programación para incrementar el rendimiento de las operaciones costosas para la máquina virtual como la creación de objetos. Se debe destruir referencias que ya no estén siendo usadas, optimizar el trabajo con cadenas, entre otras buenas prácticas que ayudan a mejorar el rendimiento.

RNF4. Portabilidad

Al sistema se debe poder acceder desde cualquier estación de trabajo de la red local y desde cualquier plataforma para ver los reportes como es el caso de los Sistemas Operativos Linux (Oracle Linux 5.5+, 6.x, Ubuntu 8.04 LTS o superior) y Windows (2000, XP, Server 2003, Server 2008, Server 2012, Vista, 7 y 8).

RNF5. Restricciones de diseño

El sistema debe estar dividido en las siguientes capas:

RNF5.1. Capas físicas

- Cliente: computadora con cualquier tecnología o sistema operativo que cuente con un navegador actualizado y que siga los estándares web (se recomienda IE 7 o superior o Firefox 3.0 o superior).
- Servidor de Aplicaciones: servidor con cualquier tecnología o sistema operativo que soporte el Java Runtime Environment (JRE) 1.6.05 o superior y al JBoss Application Server 4.2.2 o superior.
- Servidor de Base de Datos: servidor con cualquier tecnología o sistema operativo que soporte a PostgreSQL Server 8.4 o superior.

RNF5.2. Capas lógicas

- Presentación: contiene todas las vistas y la lógica de la presentación.
- Negocio: mantiene el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar siendo ejecutados por cada usuario.
- Acceso a Datos: contiene las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar Java Persistence API (JPA) y particularmente en la implementación del motor de persistencia Hibernate.

RNF6. Interfaz de usuario

RNF6.1. La aplicación debe ser sencilla e intuitiva, de fácil navegación por parte del usuario. Estará diseñada para una óptima visualización siendo adaptable a cualquier resolución.

RNF6.2. La entrada de datos incorrecta será detectada claramente e informada al usuario.

RNF7. Software

La solución informática debe poder ser desplegada en sistemas operativos Windows y Linux, utilizando la plataforma Java (Máquina Virtual de Java - Java Enterprise Edition).

- Servidor de Base de datos: Postgres 8.4 o superior.
- Servidor de Aplicaciones: JBoss 4.2.2 o superior.
- Navegador: Mozilla Firefox 3.0 o Internet Explorer 7 o versiones superiores en ambas opciones.

RNF8. Hardware

RNF8.1. Estaciones de trabajo

Para que el sistema se ejecute correctamente en la computadora del cliente, la misma debe contar con un procesador Intel Dual-Core, Intel Core-2 Duo o AMD a 800 MHz o superior, 512 MB RAM o superior y 4 GB de espacio libre en el disco duro como mínimo.

RNF8.2. Servidores

- Servidores de Base de datos: Procesador Intel® Xeon® 5140 Dual-Core 3.0 GHz, 4 GB de memoria RAM, 1199 GB de disco duro y sistema operativo GNU/Linux. PostgreSQL 8.4 o superior.
- Servidores de Aplicaciones: Procesador Intel® Xeon® 5140 Dual-Core3.0 GHz, 4 GB de memoria RAM, 1199 GB de disco duro y sistema operativo GNU/Linux. JRE 1.6 o superior. JBoss AS 4.2.2 o superior.

RNF9. Rendimiento

RNF9.1. Procesamiento de datos y tiempo de respuesta

Se debe contar con un rápido procesamiento de los datos y con un tiempo de respuesta mínimo.

RNF10. Seguridad

RNF10.1. Se mantendrá la seguridad y el control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan.

RNF10.2. Se registrarán todas las acciones que se realizan, llevando el control de las trazas de las acciones de cada usuario en todo momento.

2.3.3 Patrones de Casos de uso

Extensión Concreta: Se utiliza en los casos de uso Exportar reporte y Filtrar reporte. Consiste en la existencia de una relación de extensión entre dos casos de uso. El caso de uso extendido puede ser o no instanciado por el caso de uso base. El caso de uso base puede ser concreto o abstracto. Este patrón se utiliza cuando un flujo puede extender del flujo de otro caso de uso o bien puede ejecutarse dentro de este.

2.4 Modelo de casos de uso del sistema

2.4.1 Actores del sistema

Los actores de un sistema son agentes externos, roles que las personas (usuarios) o dispositivos juegan cuando interactúan con el software. A continuación se muestra el actor del sistema con una descripción textual del mismo.

Actor	Descripción
Administrador	Usuario con privilegios para acceder al módulo de Configuración del sistema y seleccionar el tipo de reporte a generar, así como los filtros para cada tipo de reporte.

Tabla 3. Actores del sistema

2.4.2 Vista global de actores del sistema

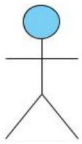
Vista global de los actores del sistema
 Administrador

Figura 2. Vista global de los actores del sistema

2.4.3 Diagrama de casos de uso del sistema

Los diagramas de casos de uso documentan y modelan el comportamiento de un sistema, subsistema o una clase, mostrando las relaciones entre los actores y el caso de uso del sistema.

Diagrama de Casos de Uso del sistema

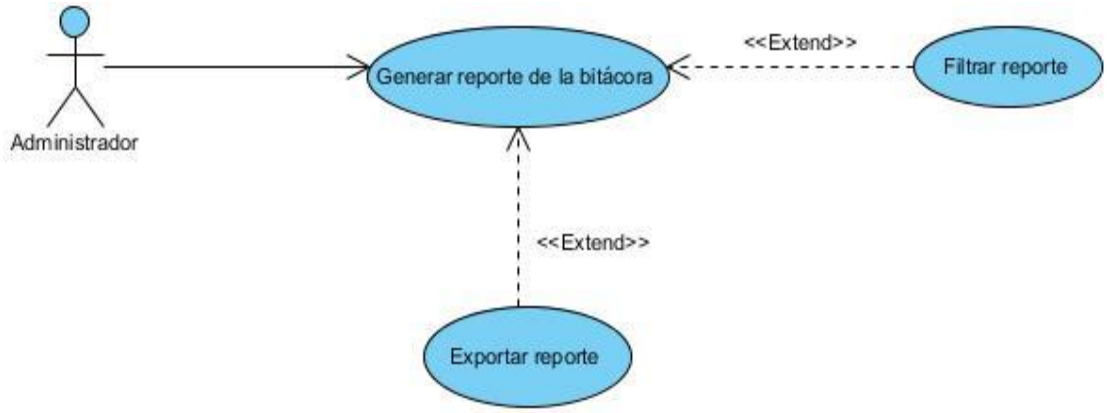


Figura 3. Diagrama de Casos de Uso del sistema

2.4.4 Descripción de los casos de uso del sistema.

CU 1. Generar reporte de la bitácora

Objetivo	Generar reporte de la bitácora.
Actores	Administrador
Resumen	El caso de uso inicia cuando el Administrador accede a la opción Reportes de la Bitácora, el sistema muestra la interfaz correspondiente y brinda la posibilidad de seleccionar uno o varios parámetros para la generación de reportes, el Administrador selecciona los parámetros deseados, el sistema a partir de los criterios seleccionados consulta las entidades correspondientes y genera el reporte de información, el caso de uso termina.
Complejidad	Alta

Prioridad	Crítico	
Precondiciones	No existen	
Postcondiciones	Se generó un reporte de información.	
Flujo de eventos		
Flujo básico Generar reporte de la bitácora.		
	Actor	**Sistema**
1.	Accede a la opción Reporte de la bitácora.	
2.		Muestra la interfaz correspondiente, con los parámetros Usuario y Módulo seleccionados por defecto además de los filtros correspondientes a cada uno de ellos y brinda la posibilidad de seleccionar como mínimo uno de los parámetros siguientes: • IP • Rango de fecha • Rango de hora • Acción • Atributo
3.	Selecciona los parámetros deseados para la generación de un reporte.	

4.		Muestra los filtros correspondientes a los parámetros seleccionados. Y permite: • Seleccionar valores en los filtros. **Ver Sección 1:** “Filtrar”. • Generar reporte. • Cancelar. **Ver Sección 2:** “Cancelar”.
5.	Selecciona la opción “Generar” para generar un reporte.	
6.		Valida que los valores de los campos de los filtros sean correctos. Si los valores son incorrectos. **Ver Evento Nº 1:** “Datos incorrectos”.
7.		Busca los datos consultando las entidades: • Módulo Accedido • Sesión • Usuario • Funcionalidad • Acción • Atributo Modificado
8.		Si no se encuentra ninguna información que cumpla con los criterios seleccionados. **Ver Evento Nº 2:** “No se encuentra información que cumpla con los criterios seleccionados.”

9.		Muestra el reporte generado y brinda la posibilidad de exportar el reporte a formato PDF, Excel o Word. **Ver Sección 3:** "Exportar."
10.		Termina el caso de uso.
Flujos alternos		
Nº 1 Datos incorrectos.		
	Actor	**Sistema**
1.		Muestra un indicador sobre el campo incorrecto.
2.		Regresa al paso 6 del **Flujo básico de eventos.**
Nº 2 No se encuentra información que cumpla con los criterios seleccionados.		
	Actor	**Sistema**
1.		Muestra el mensaje "No existe información que cumpla con los criterios seleccionados."
Sección 1: "Filtrar"		
Flujo básico Generar reporte de la bitácora.		
	Actor	**Sistema**
1.		Muestra los filtros correspondientes a los parámetros seleccionados y permite seleccionar los valores deseados en los mismos. Ver caso de uso:

		Filtrar reporte.
Sección 2: "Cancelar"		
Flujo básico Generar reporte de la bitácora.		
	Actor	**Sistema**
1.	Selecciona la opción "Cancelar".	
2.		Elimina los valores seleccionados en los filtros y desmarca los parámetros seleccionados por el Administrador.
Sección 3: "Exportar"		
Flujo básico Generar reporte de la bitácora.		
	Actor	**Sistema**
1.	Selecciona la opción "Exportar".	
2.		Exporta los datos del reporte al formato seleccionado. Ver caso de uso: **Exportar reporte.**

Relaciones	**CU Incluidos**	-
	CU Extendidos	Exportar reporte; ver caso de uso: **Exportar reporte.** Filtrar reporte; ver caso de uso: **Filtrar reporte.**
Requisitos no	Usabilidad, Fiabilidad, Restricciones de diseño, Interfaz.	

funcionales	

Tabla 4. Descripción del caso de uso “Generar reporte de la bitácora”.

Para visualizar la interfaz correspondiente al caso de uso Generar reporte de la bitácora (**Ver Anexo 5**).

CU2. Filtrar reporte

Objetivo	Filtrar reporte	
Actores	Administrador	
Resumen	El caso de uso inicia cuando el Administrador selecciona los valores deseados en los filtros, el sistema da la posibilidad de seleccionar entre un grupo de valores almacenados en los mismos o de ingresar manualmente los criterios de búsqueda, el caso de uso termina.	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones	Se seleccionaron los parámetros para generar un reporte.	
Postcondiciones	Se generó un reporte a partir de los valores seleccionados en los filtros.	
Flujo de eventos		
Flujo básico Filtrar reporte.		
	Actor	**Sistema**
1.	Selecciona los valores deseados en los filtros.	

2.		Termina el caso de uso.
Requisitos no funcionales	Usabilidad, Fiabilidad, Restricciones de diseño, Interfaz.	

Tabla 5. Descripción del caso de uso “Filtrar reporte”.

Para visualizar la interfaz correspondiente al caso de uso “Filtrar reporte” (**Ver Anexo 6**).

CU3. Exportar reporte

Objetivo	Exportar reporte
Actores	Administrador
Resumen	El caso de uso inicia cuando el Administrador accede a la opción Exportar, el sistema muestra la interfaz correspondiente y brinda la posibilidad de seleccionar el formato de salida del reporte a exportar, el Administrador selecciona el formato deseado y exporta el reporte, el caso de uso termina.
Complejidad	Media
Prioridad	Secundario
Precondiciones	Se generó un reporte de información.
Postcondiciones	Se exportó un reporte.
Flujo de eventos	
Flujo básico Exportar reporte.	

	Actor	**Sistema**
1.	Selecciona la opción “Exportar”.	Muestra una interfaz con los formatos de salida del reporte (**Ver Anexo 7**). Y permite: • Aceptar. • Cancelar. **Ver Sección 1:** “Cancelar exportar.” • Salir: **Ver Sección 2:** “Salir.”
2.	Elige el formato de salida del reporte y selecciona la opción “Aceptar”.	
3.		Valida que el campo de selección de formato no esté incompleto. Si el campo está incompleto (**Ver Evento Nº 1**: “Campo de formato incompleto”).
4.		Exporta el reporte en un fichero con el formato seleccionado (PDF, Word o Excel). (**Ver Anexo 8**).
5.		Termina el caso de uso.
Flujos alternos		
Nº 1 Campo de formato incompleto.		
	Actor	**Sistema**
1.		Muestra un indicador sobre el campo incompleto y muestra el mensaje “Existen datos incompletos.”

<table>
<tr><td colspan="5">Sección 1: “Cancelar exportar”</td></tr>
<tr><td colspan="5">Flujo básico Exportar reporte.</td></tr>
<tr><td></td><td colspan="2">Actor</td><td colspan="2">Sistema</td></tr>
<tr><td>1.</td><td colspan="2">Selecciona la opción “Cancelar”.</td><td colspan="2"></td></tr>
<tr><td>2.</td><td colspan="2"></td><td colspan="2">Regresa a la vista anterior.</td></tr>
<tr><td colspan="5">Sección 2: “Salir”</td></tr>
<tr><td colspan="5">Flujo básico Exportar reporte.</td></tr>
<tr><td></td><td colspan="2">Actor</td><td colspan="2">Sistema</td></tr>
<tr><td>1.</td><td colspan="2">Selecciona la opción “Salir”.</td><td colspan="2"></td></tr>
<tr><td>2.</td><td colspan="2"></td><td colspan="2">Regresa a la vista anterior.</td></tr>
<tr><td rowspan="2">Relaciones</td><td>CU Incluidos</td><td colspan="3">-</td></tr>
<tr><td>CU Extendidos</td><td colspan="3">-</td></tr>
<tr><td>Requisitos no funcionales</td><td colspan="4">Usabilidad, Fiabilidad, Restricciones de diseño, Interfaz.</td></tr>
</table>

Tabla 6. Descripción del caso de uso “Exportar reporte”.

En el presente capítulo se realizó el modelo de dominio donde se describieron las clases conceptuales que ayudarán a comprender los conceptos claves para un mejor dominio del problema. Además se identificaron los requisitos funcionales y no funcionales que el sistema debe cumplir, basados en software,

hardware, interfaz, eficiencia, usabilidad y restricciones de diseño e implementación con el objetivo de tener una guía para la creación de un producto que responda a las necesidades del cliente. Se identificaron los actores que intervienen en el sistema así como una vista global de los mismos. También se realizó el diagrama de casos de uso del sistema, para un mejor entendimiento del comportamiento del mismo y con ello dar paso a la creación de los artefactos correspondientes a la fase de análisis y diseño del sistema con la calidad requerida.

CAPÍTULO 3. Análisis y diseño del sistema.

En el presente capítulo se presentan los elementos correspondientes a la fase de diseño de la solución propuesta. Se representan los diagramas de clase del diseño con una descripción de cada uno de ellos y los diagramas de secuencia para los casos de uso más importantes. Se definen los patrones de diseño y estilos arquitectónicos a utilizar, así como una breve descripción de sus características.

Para dar inicio al diseño de la solución propuesta se deben introducir un conjunto de elementos dentro de los que se destacan la arquitectura a emplear, la misma se expone a continuación.

3.1 Descripción de la arquitectura

La arquitectura del software proporciona una visión global de un sistema a construir. Describe la estructura y organización de los componentes de software, sus propiedades y las conexiones entre ellos. Está compuesta por un conjunto de patrones y estilos arquitectónicos capaces de proveer un marco de referencia necesario para guiar la construcción de un software. (42) Por lo anterior expuesto se selecciona para el desarrollo de la solución informática, la arquitectura de tres niveles o Programación en capas como le denominan algunos autores. Dicha arquitectura constituye una especialización de la arquitectura cliente-servidor donde su objetivo es dividir la carga en tres capas o niveles con funciones específicas. El patrón arquitectónico Modelo-Vista-Controlador es un ejemplo claro de ello.

3.1.1 Patrón Arquitectónico

Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. No es una arquitectura como tal, sino un concepto que captura elementos esenciales de una arquitectura de software. (43) El patrón a utilizar será Modelo-Vista-Controlador (MVC), el cual separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes.

Modelo: El modelo administra el comportamiento y los datos del dominio de aplicación, responde requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: Está compuesta por las interfaces encargadas de mostrar la información al usuario y permite la interacción con el sistema, se comunica únicamente con la capa de negocios o controladora.

Controlador: Interpreta las acciones del mouse y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado.

Una vez seleccionada la arquitectura y patrón arquitectónico a utilizar se da paso a definir los componentes del modelo de diseño, presentando inicialmente su definición y posteriormente, los patrones para la asignación de responsabilidades y patrones GoF.

3.2 Modelo de diseño

El diseño es una representación significativa de algo que se va a construir. Se puede hacer el seguimiento basándose en los requisitos del cliente, y al mismo tiempo la calidad se puede evaluar y cotejar con el conjunto de criterios predefinidos para obtener un diseño bueno (44).

3.2.1 Patrones de diseño

Los patrones de diseño ofrecen las soluciones que los expertos en el diseño orientado a objeto se sirven para crear sistemas generalmente se caracterizan porque son soluciones concretas, se utilizan en situaciones frecuentes y favorecen la reutilización del código. Además ofrecen las ventajas de ser una experiencia real, probada y que funciona.

Patrones de diseño GRASP

Los Patrones de Software para la Asignación General de Responsabilidad (GRASP por sus siglas en ingles), representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones (45). A continuación se muestran los patrones GRASP presentes en la solución:

Creador: El propósito fundamental de este patrón es encontrar el responsable de crear un nuevo objeto de alguna clase. Con este patrón se da soporte al bajo acoplamiento. Su utilización en el sistema se evidencia en la clase GenerarReporteBitacora, la cual tiene la información necesaria para la creación de los objetos de los reportes, los cuales son enviados como una lista a la clase ReportManagerConfig, para finalmente construir los reportes.

Bajo acoplamiento: El acoplamiento mide qué tan fuerte está una clase conectada con otras clases. Una clase con bajo acoplamiento no depende de muchas otras, de tal forma que en caso de producirse una modificación en ella, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización y disminuyendo la dependencia entre las clases. En el sistema en cuestión este patrón se pone en evidencia debido a que existe una única clase controladora (GenerarReporteBitacora), la cual posee la mínima dependencia de las demás clases existentes en el sistema como ReportManagerConfig y EntityManager y las clases envolventes (Wrapper classes).

Controlador: El patrón controlador funciona como intermediario entre una interfaz y el algoritmo que la implementa. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, con el objetivo de aumentar la reutilización de código y a la vez tener un mayor control de las funcionalidades. El patrón descrito se pone de manifiesto a través de la clase controladora GenerarReporteBitacora que funciona como intermediaria entre la vista y la capa de acceso a datos.

Patrones de diseño GoF

Los patrones GoF (Gang of Four, en español Pandilla de los Cuatro) se clasifican en 3 categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

El patrón GoF utilizado en la investigación fue el Singleton (instancia única), el cual garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia, evitando la duplicación de información. Se implementó creando en la clase controladora GenerarReporteBitacora una instancia de las clases ReportManagerConfig y EntityManager, esta última encargada de realizar operaciones sobre los datos persistentes en la base de datos.

Para mostrar como los objetos se comunican entre ellos a fin de cumplir con los requerimientos, se hace necesario realizar los diagramas de clases del diseño. Los mismos muestran la definición de las clases e interfaces implementadas en software, a continuación se presentan los empleados en la investigación en curso.

3.2.2 Diagramas de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. A diferencia del modelo conceptual, un diagrama de este tipo contiene las definiciones de las entidades del software en vez de conceptos del mundo real. (46)

Las clases del diseño están agrupadas en:

Páginas Servidoras: Están compuestas por componentes como Facelets, RichFaces, JSF, Seam UI, así como código HTML. Permite ejecutar todo este código en el servidor web, generando páginas clientes que pueden ser interpretadas por los navegadores web.

Páginas Clientes: Contienen código HTML, JavaScript y CSS. Son interpretadas por los navegadores web, presentándole al usuario la interfaz para que pueda interactuar con el sistema.

Formularios: Un formulario HTML es una sección de un documento enmarcado entre etiquetas <form> y que puede contener elementos especiales llamados controles. Permite al servidor obtener la información entrada por el usuario a través de los formularios.

A continuación se muestran los diagramas de clases del diseño para los casos de uso: Generar reporte de la bitácora, Filtrar reporte y Exportar reporte.

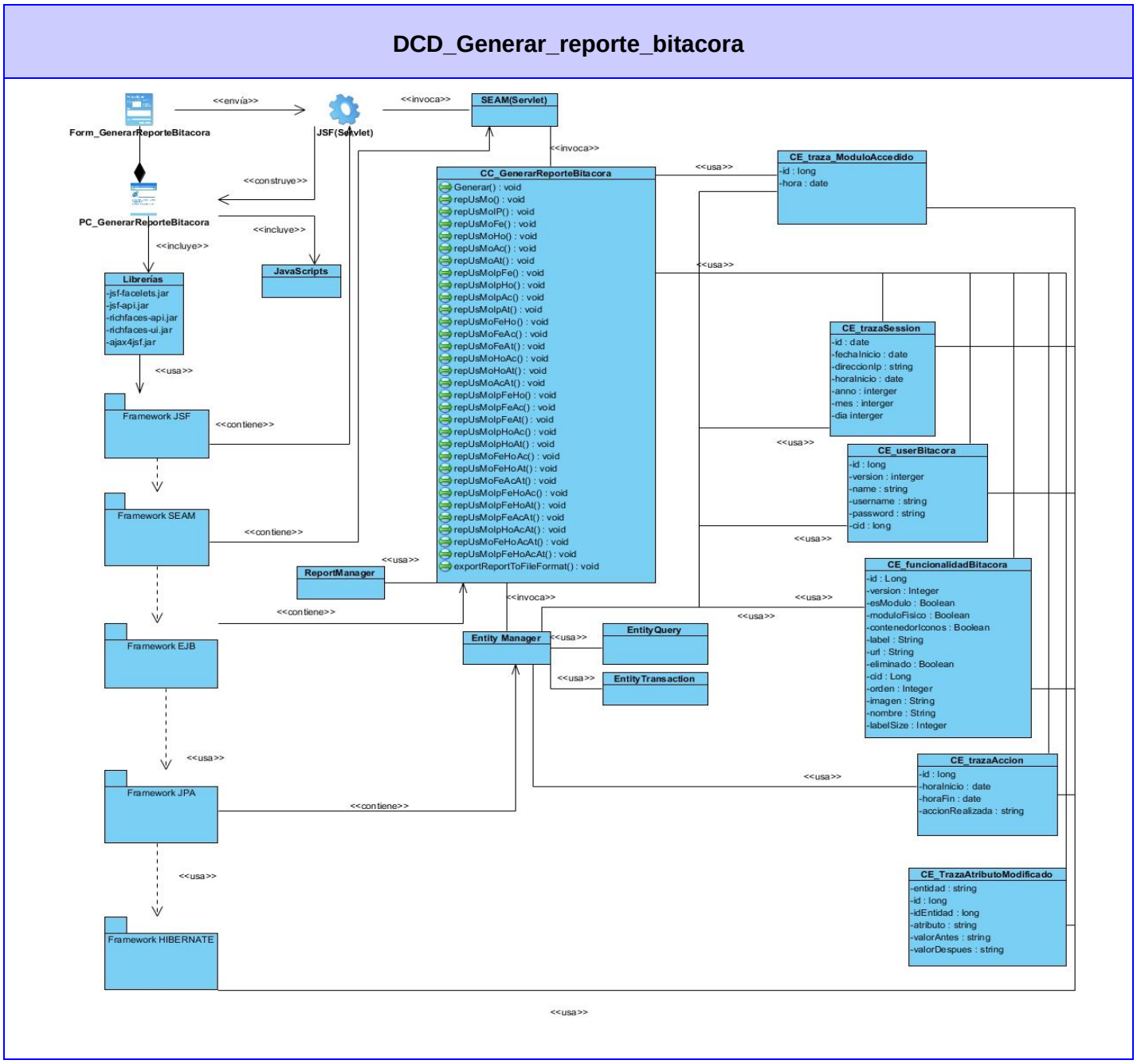


Figura 4. Diagrama de clases del diseño del caso de uso “Generar reporte de la bitácora”

A continuación se muestra una descripción de la página cliente y clase controladora que se presenta en el diagrama anterior correspondiente al caso de uso Generar reporte de la bitácora.

Nombre de la clase	PC_GenerarReporteBitacora
Tipo	Página cliente.
Descripción general	Permite al administrador seleccionar los parámetros deseados para la generación de un reporte.

Tabla 7: Descripción página cliente “PC_ GenerarReporteBitacora”.

Caso de Uso	Generar reporte de la bitácora
Nombre	CC_GenerarReporteBitacora
Tipo	Clase controladora
Descripción	Clase encargada de crear los objetos necesarios para construir los reportes.
Atributos	
Nombre	**Tipo**
usuario	String
ip	String
modulo	String
fecha	Date
fechaad	Date
fechaaa	Date
horai	Date
horaf	Date

accion	String
atributo	String
txthoraini	String
txthorafin	String

Responsabilidades		
Nombre	**Tipo**	**Descripción**
Generar	void	Método que se encarga de procesar las peticiones del usuario para cada combinación de parámetros seleccionados e invoca el método correspondiente a cada una de ellas.
repUsMoIp	void	Genera un reporte que permite conocer todos los IP, desde los cuales los usuarios acceden a todos los módulos del sistema.
repUsMoFe	void	Permite conocer a través de un reporte la fecha en que los usuarios acceden a los módulos del sistema.
repUsMoHo	void	Muestra mediante un reporte todos los usuarios que acceden a los módulos del sistema en un horario determinado
repUsMoAc	void	Genera un reporte correspondiente a la acción que realiza un usuario sobre los módulos del sistema.
repUsMoAt	void	Permite conocer a través de un reporte que modificaciones realizan los usuarios sobre los módulos del sistema.

Tabla 8: Descripción de la clase controladora “GenerarReporteBitacora”.

DCD_Filtrar_Reporte

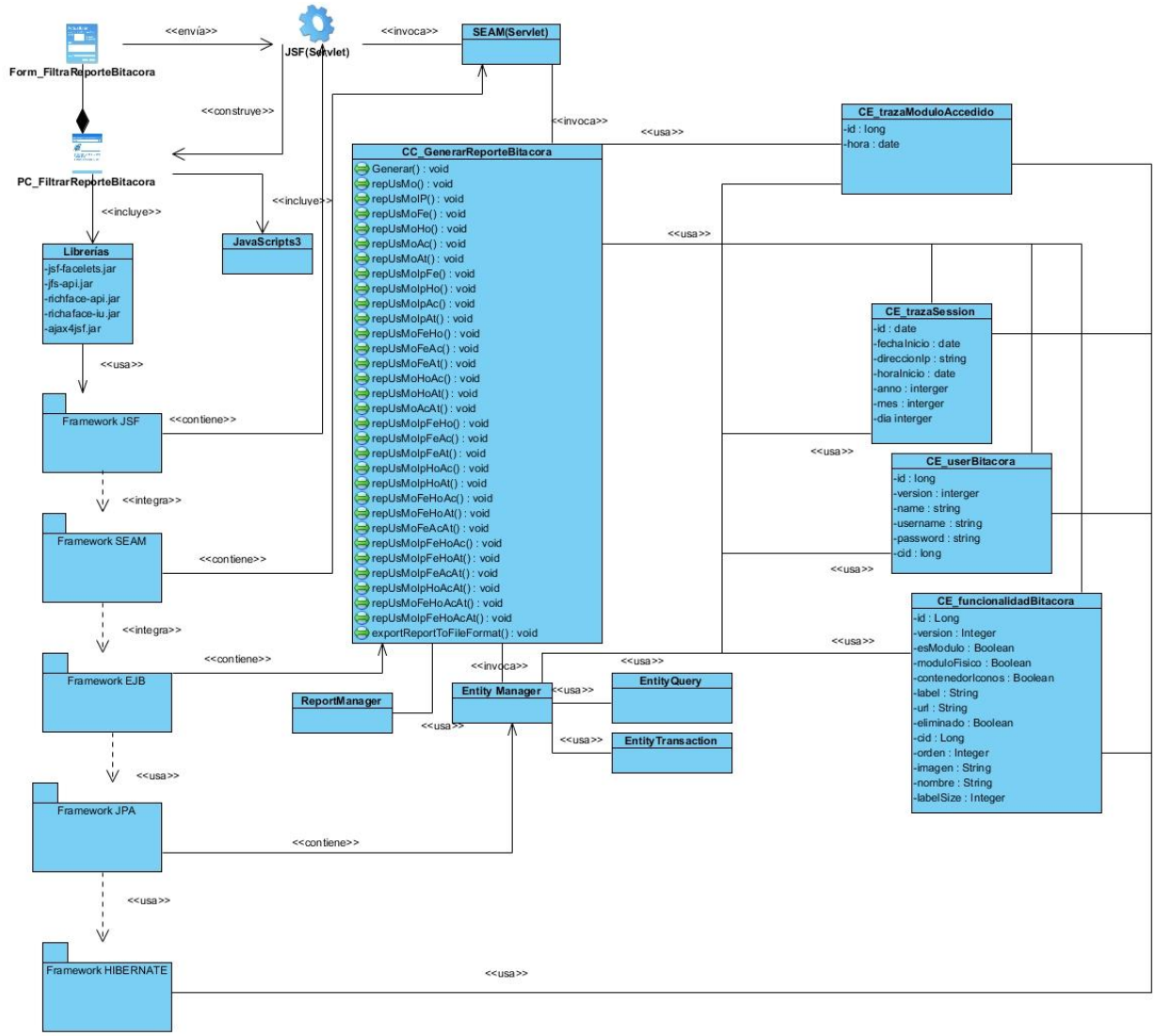


Figura 5. Diagrama de clases del diseño del caso de uso “Filtrar Reporte”

A continuación se muestra una descripción de la página cliente y clase controladora que se presenta en el diagrama anterior correspondiente al caso de uso Filtrar Reporte.

Nombre de la clase	CP_FiltrarReporteBitacora
Tipo	Página cliente.
Descripción general	Permite al administrador seleccionar los filtros deseados para la generación de un reporte.

Tabla 9. Descripción página cliente “CP_FiltarReporteBitacora”.

Caso de Uso	Filtrar Reporte
Nombre	CC_GenerarReporteBitacora
Tipo	Clase controladora
Descripción	Clase encargada de crear los objetos necesarios para construir los reportes.

Atributos	
Nombre	**Tipo**
verfiltro	boolean
verusuario	boolean
verip	boolean
vermodulo	boolean
verfechaa	boolean
verfechad	boolean
verhora	boolean
veraccion	boolean

veratributo	boolean
Responsabilidades	
Nombre	**Tipo**	**Descripción**
Generar	void	Método encargado de invocar el método adecuado en dependencia de la combinación de parametros seleccionados y los filtros aplicados a cada uno de ellos para restringir los resultados.
repUsMoIpFeHo	void	Genera un reporte que permite conocer el IP, la fecha y la hora desde los cuales los usuarios acceden a un módulo determinado del sistema.
repUsMoIpFeAc	void	Permite conocer la acción realizada por un usuario sobre un módulo desde un IP determinado, en un rango de fecha específico.
repUsMoIpHoAc	void	Permite conocer a través de un reporte la hora en que un usuario realiza una acción sobre un módulo determinado desde un IP específico.
repUsMoFeHoAt	void	Genera un reporte con información correspondiente a la las modificaciones que realiza un usuario sobre un módulos específico, además muestra la fecha y la hora en que tuvo lugar dicha modificación.
repUsMoFeHoAcAt	void	Muestra mediante un reporte los usuarios que acceden a los módulos del sistema en un rango de fecha y horario determinado, así como las acciones que realizan y los atributos que modifican.

Tabla 10. Descripción de la clase controladora “CC_GenerarReporteBitacora”.

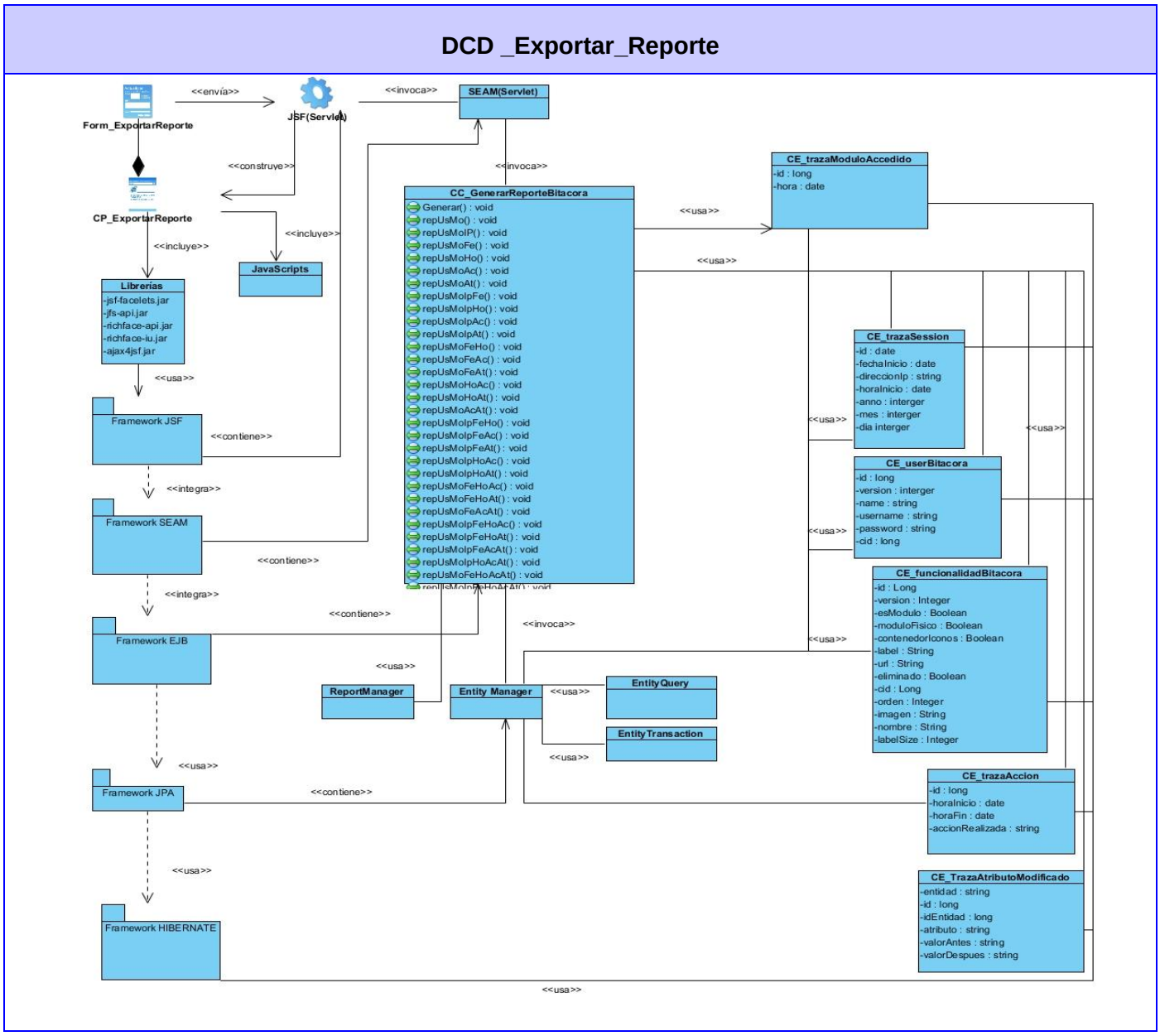


Figura 6. Diagrama de clases del diseño para el caso de uso “Exportar reporte”

A continuación se muestra una descripción de la página cliente y clase controladora con algunas de las responsabilidades más importantes que intervienen en el diagrama anterior.

Nombre de la clase	PC_ExportarReporte
Tipo	Página cliente.
Descripción general	Permite al administrador seleccionar el formato de salida deseado para exportar un reporte.

Tabla 11. Descripción de la página cliente “PC_ExportarReporte”.

Caso de Uso	Generar reporte General	
Nombre	CC_GenerarReporteBitacora	
Tipo	Clase controladora	
Descripción	Clase encargada de crear los objetos necesarios para construir los reportes.	
Atributos		
Nombre	**Tipo**	
fileformatToExport	String	
filesFormatCombo	List<String>	
pathExportedReport	String	
tipoReporte	Integer	
Responsabilidades		
Nombre	**Tipo**	**Descripción**

exportReportToFileFormat	void	Se encarga de exportar cada tipo reporte al formato de salida seleccionado por el usuario.

Tabla 12. Descripción de la clase controladora “GenerarReporteBitacora”.

3.2.3 Diagramas de interacción

Un diagrama de interacción, representa la forma en como un cliente (actor) u objetos (clases) se comunican entre sí en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades claramente. (47) En UML los diagramas de interacción pueden representarse a través de los Diagramas de Colaboración y/o de los Diagramas de Secuencia. En la presente investigación se realiza un diagrama de secuencia con el objetivo de mostrar la comunicación entre los objetos en una secuencia de tiempo, además se presentan los ciclos de vida y los mensajes que se envían entre ellos ordenados secuencialmente. A continuación se muestran los diagramas de secuencia para los casos de uso: Generar reporte de la bitácora, Filtrar reporte y Exportar reporte.

DS_Generar_reporte_bitacora

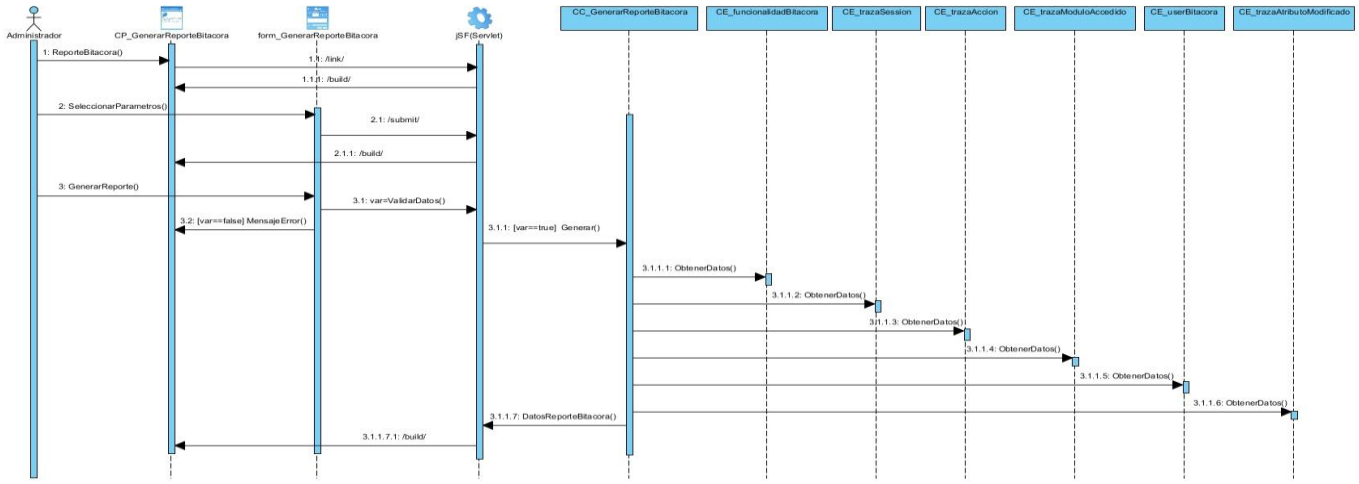


Figura 7. Diagrama de secuencia correspondiente al caso de uso “Generar reporte de la bitácora”.

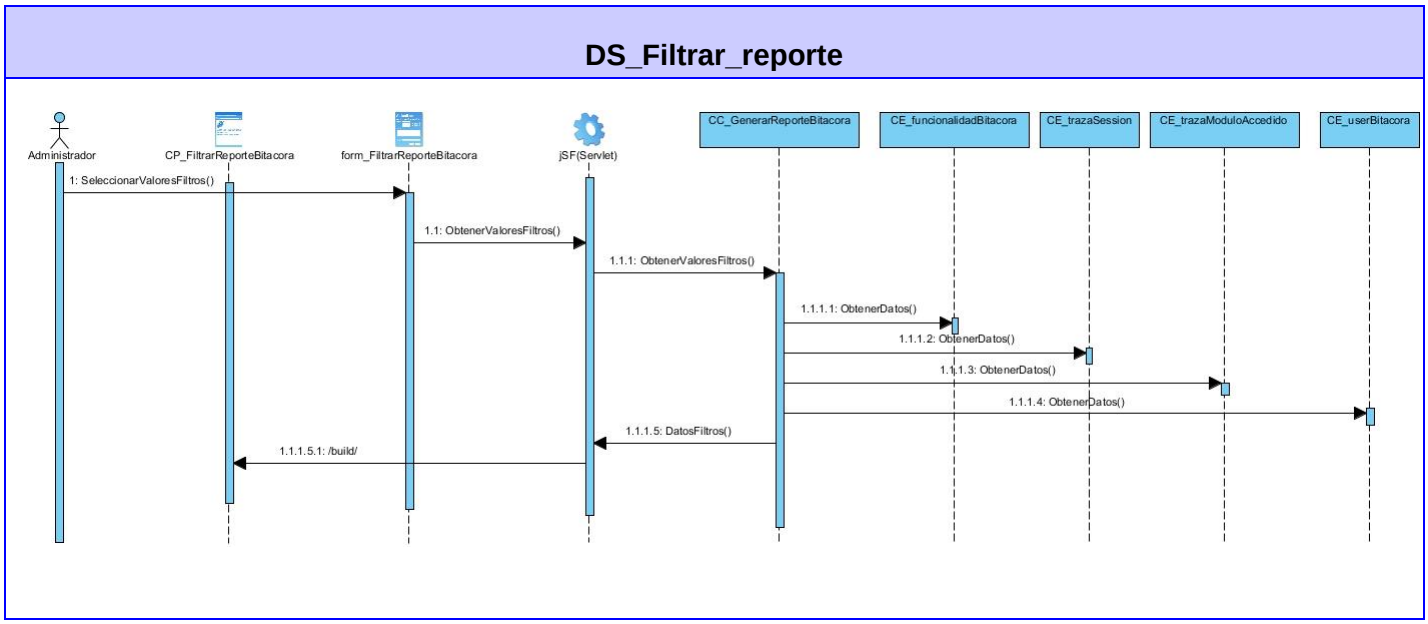


Figura 8. Diagrama de secuencia correspondiente al caso de uso “Filtrar reporte”.

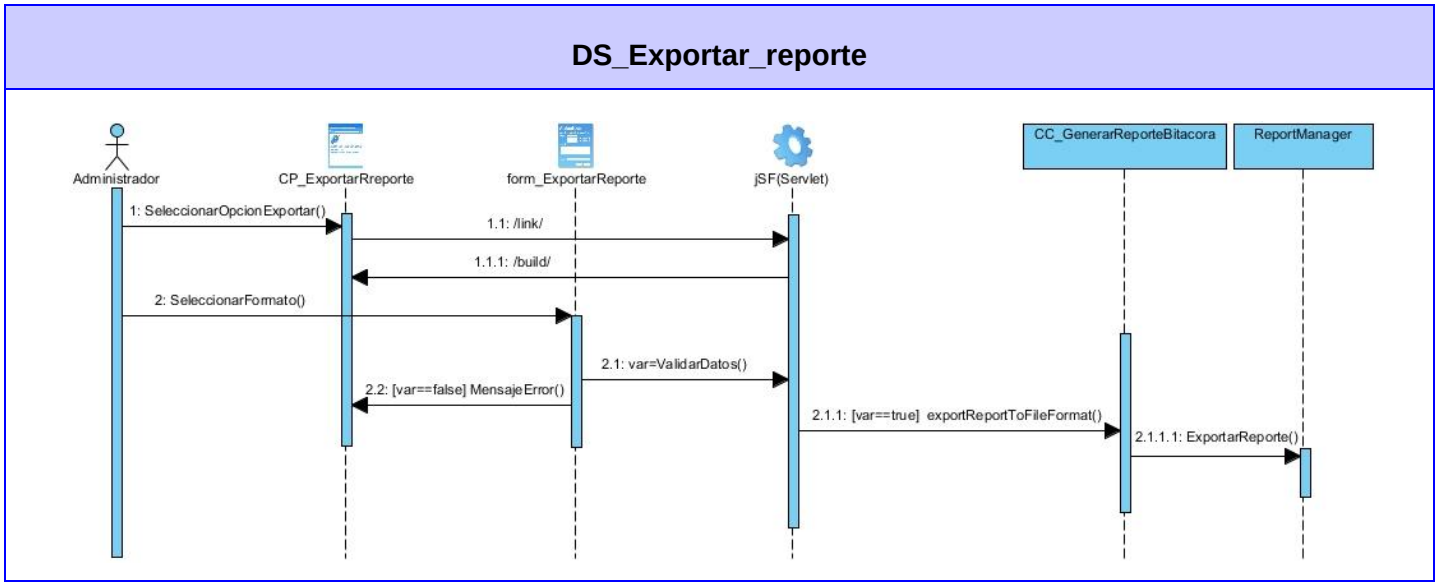


Figura 9. Diagrama de secuencia del caso de uso “Exportar reporte”.

En el presente capítulo se presentaron los elementos correspondientes al modelo de diseño como son los diagramas de clases del diseño y diagramas de secuencia de los casos de uso: Generar reporte de la bitácora, Filtrar reporte y Exportar reporte. Se presentó una descripción de las páginas clientes y clases

controladoras para cada diagrama expuesto, con el objetivo de mostrar la interacción entre las clases existentes. Además, se seleccionó como patrón arquitectónico a utilizar el patrón Modelo-Vista-Controlador y como patrones de diseño GRASP para la asignación de responsabilidades los patrones Creador, Bajo Acoplamiento y Controlador, además se empleó el patrón GoF Singleton. Todos estos elementos permitieron obtener una visión de la estructura global del sistema a desarrollar por lo que son tomados como base para la fase de implementación.

CAPÍTULO 4. Implementación.

En el presente capítulo se presentan los elementos correspondientes a la fase de implementación. Se introduce el flujo de trabajo de implementación, partiendo de los resultados obtenidos en el diseño. Se presenta el Modelo de Implementación a través de los diagramas de despliegue y diagramas de componentes. Además como parte de la fase de implementación se presenta el Modelo de Datos, el cual se detalla a continuación.

4.1 Modelo de Datos

Un modelo de datos es un sistema formal y abstracto que permite describir los datos de acuerdo con reglas y convenios predefinidos. Es formal pues los objetos del sistema se manipulan siguiendo reglas perfectamente definidas y utilizando exclusivamente los operadores definidos en el sistema, independientemente de lo que estos objetos y operadores puedan significar. (48) Para un mejor entendimiento la siguiente figura muestra el Modelo de Datos correspondiente a la solución propuesta.

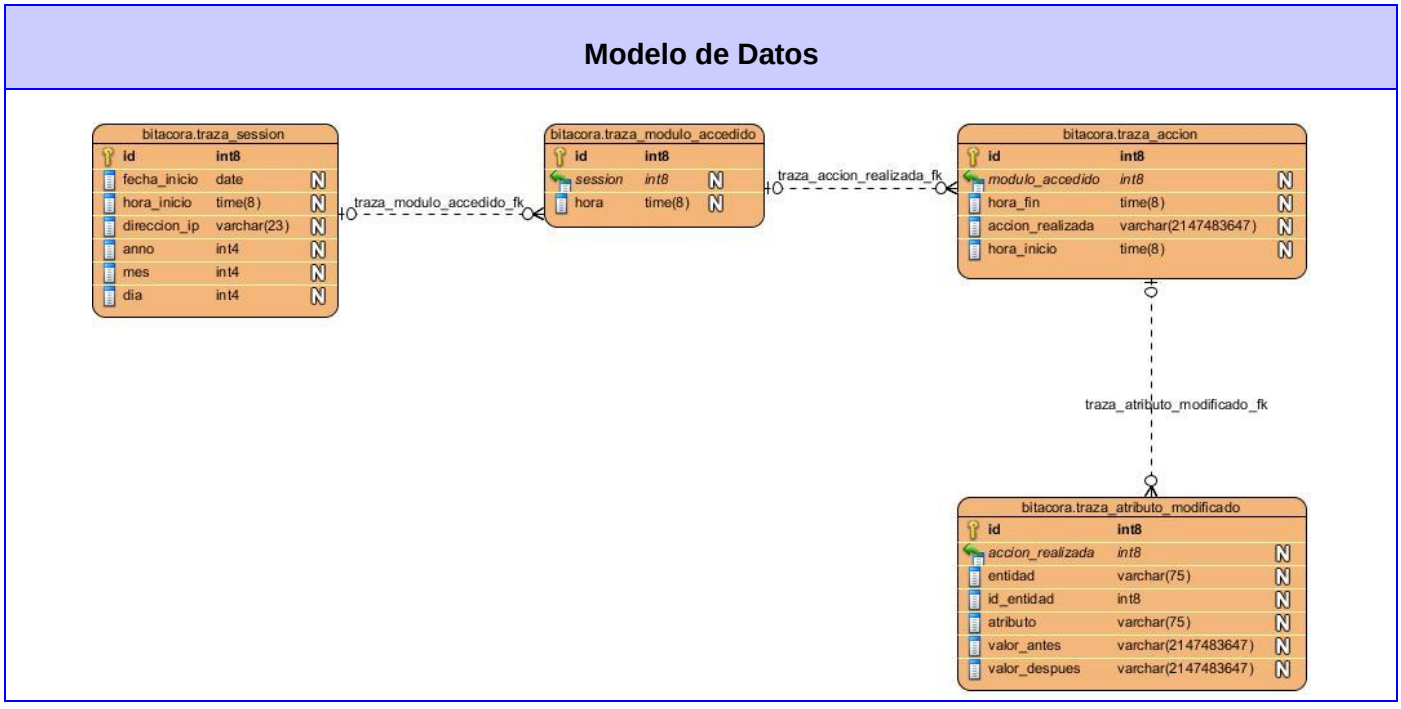


Figura 10. Modelo de Datos

4.1.1 Descripción de las tablas de la base de datos

A continuación se describen las entidades del modelo anterior.

Nombre: bitácora.traza_session		
Descripción: Clase encargada de almacenar los datos de cada sesión de un usuario.		
Atributo	**Tipo**	**Descripción**
id	bigint	Identificador de la sesión.
fecha_inicio	date	Fecha de inicio de la sesión.
hora_inicio	time	Hora de inicio de la sesión.
direccion_ip	varchar	Dirección IP desde la cual se inicia la sesión.
anno	int	Año en el cual se inicia la sesión.
mes	int	Mes en el cual se inicia la sesión.
dia	int	Día en el cual se inicia la sesión.

Tabla 13. Descripción de la tabla: bitácora.traza_session

Nombre: bitácora.traza_modulo_accedido		
Descripción: Clase encargada de almacenar los datos de los módulos accedidos por los usuarios del sistema.		
Atributo	**Tipo**	**Descripción**
id	bigint	Identificador del módulo.

session	bigint	ID de sesión.
hora	time	Hora de acceso a un módulo.

Tabla 14. Descripción de la tabla: bitácora.traza_modulo_accedido

Nombre: bitácora.traza_accion		
Descripción: Clase encargada de almacenar los datos de las acciones de los usuarios en el sistema.		
Atributo	**Tipo**	**Descripción**
id	bigint	Identificador de la acción realizada.
modulo_accedido	bigint	Identificador del módulo.
hora_inicio	time	Hora de inicio de una acción.
hora_fin	time	Hora de fin de una acción.
acción_realizada	varchar	Acción realizada sobre un módulo.

Tabla 15. Descripción de la tabla: bitácora.traza_accion

Nombre: bitácora.traza_atributo_modificado		
Descripción: Clase encargada de almacenar los datos de los atributos modificados en los módulos del sistema.		
Atributo	**Tipo**	**Descripción**
id	bigint	Identificador del atributo.

accion_realizada	bigint	Identificador de la acción realizada.
entidad	varchar	Entidad a la que pertenece el atributo.
id_entidad	bigint	Identificador de la entidad.
atributo	varchar	Atributo modificado.
valor_antes	varchar	Valor de un atributo antes de modificado.
valor_despues	varchar	Valor de un atributo después de modificado.

Tabla 16. Descripción de la tabla: bitácora.traza_atributo_modificado.

Una vez presentado el Modelo de datos con la correspondiente descripción de sus atributos, se procede a detallar el Modelo de implementación el cual describe la relación que existe entre los componentes y subsistemas.

4.2 Modelo de implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. (49) Uno de los artefactos correspondientes a este modelo es el Diagrama de Despliegue el cual se presenta seguidamente.

4.2.1 Diagrama de Despliegue

El Diagrama de Despliegue es el encargado de representar la distribución de los componentes de hardware en un sistema, así como las relaciones entre ellos. Los elementos utilizados por este tipo de diagrama son los nodos, componentes y asociaciones. El mismo se representa con un grafo de nodos unidos por asociaciones de comunicación. Estas asociaciones indican algún tipo de ruta de comunicación entre los nodos, donde estos últimos intercambian objetos o envían mensajes a través de esta ruta. Su objetivo es mostrar cómo y dónde se desplegará el sistema final. La figura siguiente muestra el diagrama correspondiente a la presente investigación especificando los requisitos mínimos de software y hardware.

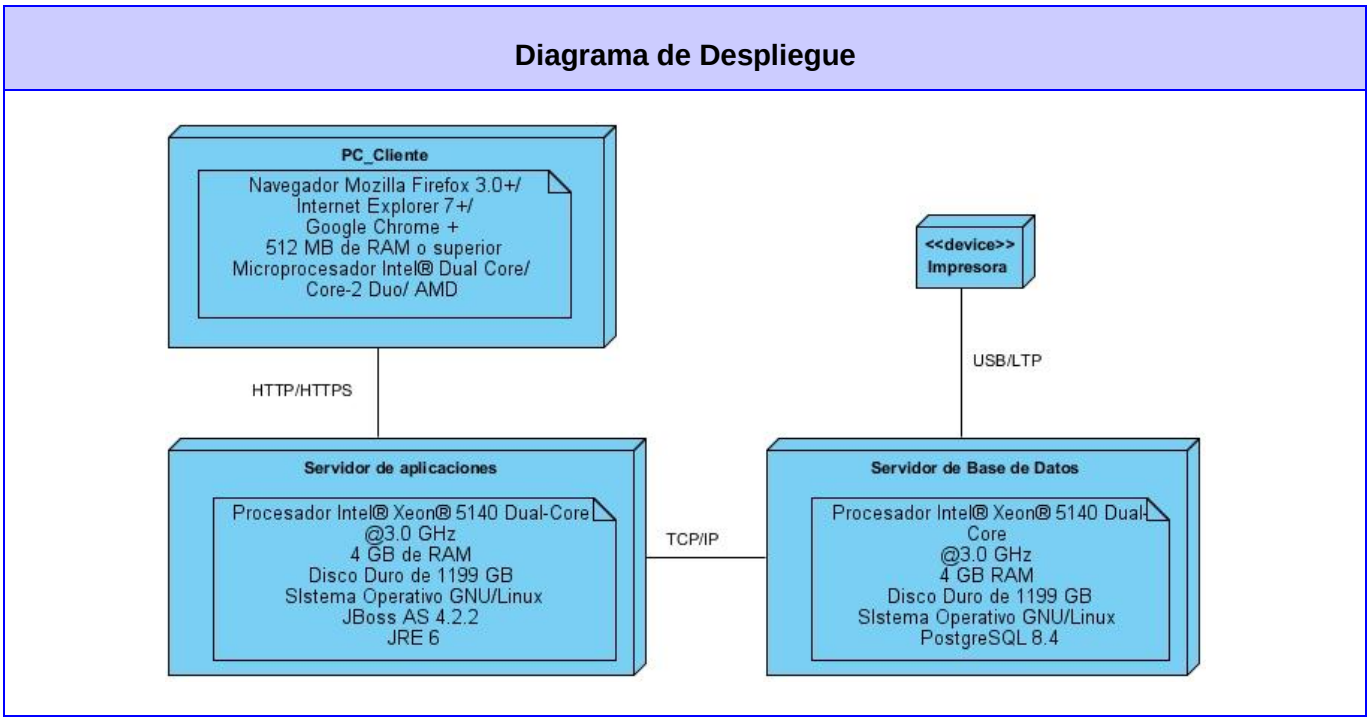


Figura 11. Diagrama de Despliegue

A diferencia del Diagrama de Despliegue presentado anteriormente, el Diagrama de Componentes se representa mediante componentes de un modelo (no de hardware), estos están unidos mediante relaciones de dependencia, seguidamente se profundiza sobre el mismo y se muestra el diagrama correspondiente a la solución propuesta.

4.2.2 Diagrama de Componentes

Un Diagrama de Componente es un esquema o diagrama que muestra las interacciones y relaciones de los componentes de un modelo. Entendiéndose como componente a una clase de uso específico, que puede ser implementada desde un entorno de desarrollo, ya sea de código binario, fuente o ejecutable; dichos componentes poseen tipo, que indican si pueden ser útiles en tiempo de compilación, enlace y ejecución. (50) La siguiente figura muestra el Diagrama de Componentes representado con tres componentes esenciales: Modelo, Vista y Controlador.

Diagrama de Componentes

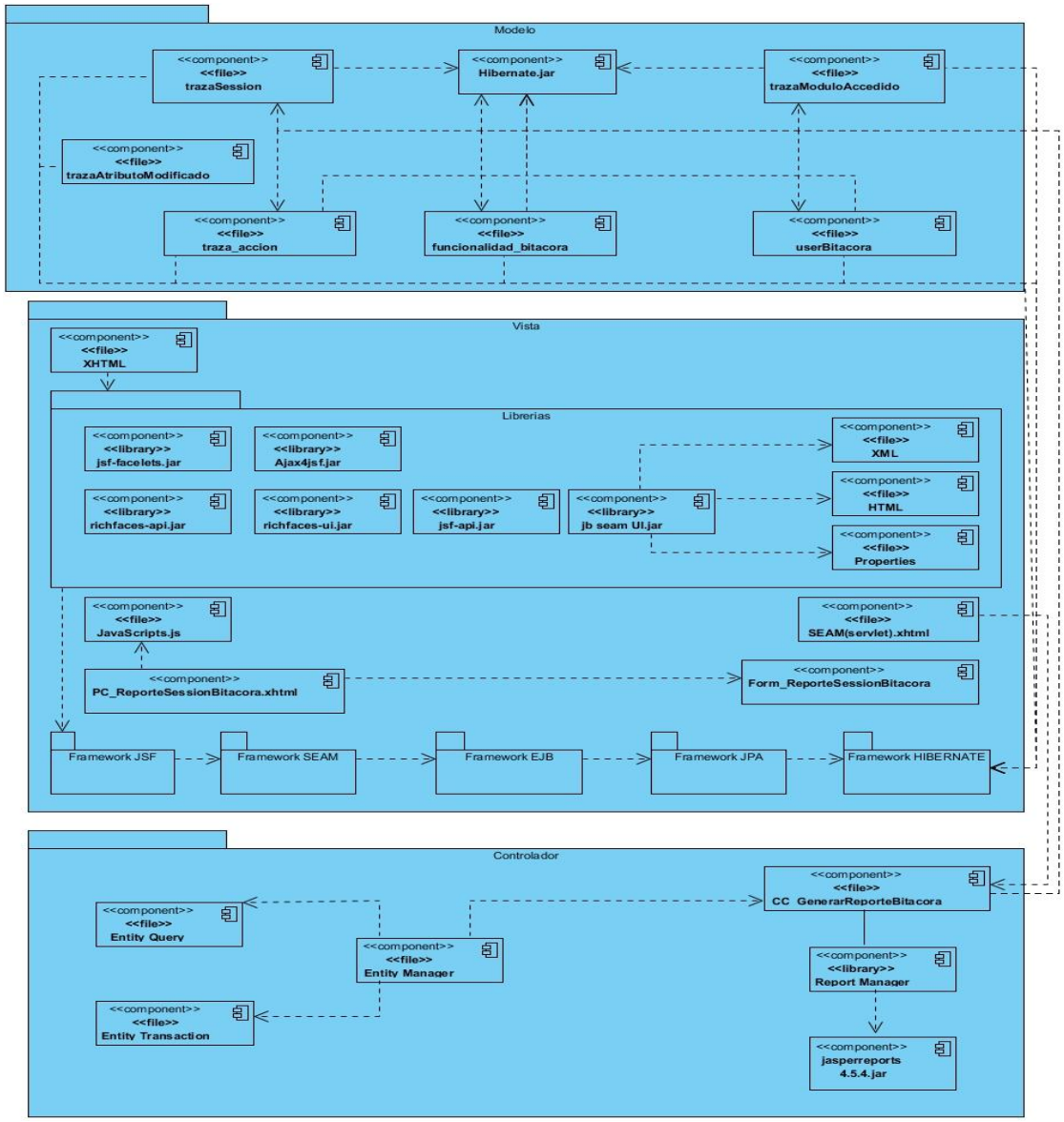


Figura 11. Diagrama de componentes

La existencia de excepciones no contraladas puede dañar el funcionamiento de cualquier sistema de software, por tal motivo a continuación se definen un conjunto de mecanismos para el control de las mismas.

4.3 Tratamiento de errores

Una excepción es un evento que ocurre durante la ejecución del programa que interrumpe el flujo normal de las sentencias. (51) En la solución propuesta se hace un tratamiento de excepciones a través de las sentencias try/catch/finally con el objetivo de evitar fallas en el sistema, en la figura siguiente se muestra su estructura básica.

```
try {
    //Código inconsistente
} catch (Exception e) {
    //En caso de una excepción
}
finally{
    //Código que siempre se ejecutará
}
```

Figura 12. Estructura básica de una excepción.

Por otra parte, para la validación de los datos entrados por el usuario se hace uso de la clase Validaciones.java, la misma está compuesta por un conjunto de expresiones regulares para la validación de los diferentes tipos de datos introducidos por el usuario. Para el control de las excepciones es utilizado además el componente del framework Seam: FacesMessages, el mismo se encarga de mostrar los diferentes mensajes de advertencia al usuario en caso de entradas incorrectas de datos, dichos mensajes son manejados a través del objeto facesMessages creado en la clase controladora.

4.4 Seguridad

La seguridad de un sistema de información está dada por tres elementos fundamentales como son la confidencialidad, la cual implica el acceso a la información por parte únicamente de quienes están autorizados, la integridad, que conlleva el mantenimiento de la exactitud y completitud de la información y la disponibilidad, que se encarga de establecer el acceso a la información por parte de los usuarios

autorizados en el momento que lo requieran. En la solución propuesta se ponen en evidencia estos aspectos de la manera siguiente:

- Existe un control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan.
- Para el acceso al sistema se definió un usuario Administrador, con el máximo privilegio en el sistema, capaz de otorgar a cada usuario permisos estrictamente necesarios para efectuar solo las acciones que se requieran. Con ello se cumple con el principio básico de Seguridad Informática: Mínimo privilegio.
- El sistema permite registrar todas las acciones que se realizan, llevando el control de las trazas de las acciones de cada usuario en todo momento.

4.5 Estrategias de codificación. Estándares y estilos a utilizar.

Las estrategias y estándares de codificación permiten al programador tener una mejor visualización del código escrito. Según los aspectos establecidos en las Convenciones de Código para el lenguaje de programación Java, existen diversas razones por las cuales los programadores deben regirse por estándares y estilos de codificación, algunas de ellas son (52):

- El 80% del costo del código de un programa va a su mantenimiento.
- Casi ningún software lo mantiene toda su vida el auto original.

Indentación

El indentado que se utilizó fue de cuatro espacios (no tabulaciones) para cada declaración de variable. En el caso de las sentencias IF se empleó la norma de ocho espacios debido a que la indentación convencional (4 espacios) hace difícil ver el cuerpo de la sentencia.

```
if (this.usuario != null && this.usuario != ""){
        parametrosQuery.put("usuario", this.usuario);
}
```

Figura 13. Ejemplo de indentado para una sentencia IF.

Para las expresiones largas que ocupan más de una línea, se dividió la misma después de una coma o antes de un operador.

```
+ "where :para1=:para1 "
+ ((this.usuario != null && this.usuario != "") ? "and use.name=:usuario " : "")
+ ((this.modulo != null && this.modulo != "") ? "and mod.modulo.label=:modulo " : "")
+ ((this.ip != null && this.ip != "") ? "and ses.direccionIp=:ip " : "");
```

Figura 14. Ejemplo de expresiones largas divididas antes de un operador.

Se evitaron el uso de las líneas de más de 80 caracteres, ya que no son manejadas bien por muchas terminales y herramientas.

Variables y constantes

La inicialización de las variables locales se realizó donde estas se declaran. Las mismas poseen nombres de fácil entendimiento que con solo leerlo se conoce su propósito fundamental. La nomenclatura empleada fue la Camel-Case específicamente la lowerCamelCase donde la primera letra de cada palabra se escribe en mayúscula, excepto para la primera palabra.

Comentarios

Se emplearon comentarios de documentación y de implementación para un mejor entendimiento del código escrito para futuros desarrolladores.

```
/*
 * Formar cadena de internacionalizaciÃ³n del applet para imprimir.
 *
 */
```

Figura 15. Ejemplo de comentario de bloque.

Espacios en blanco

Para los espacios en blanco se siguieron las normas establecidas por las Convenciones de Código para el lenguaje Java, las cuales plantean que debe haber un espacio en blanco después de cada coma o punto y

coma y que no se debe usar un espacio en blanco entre el nombre de un método y su paréntesis de apertura o luego del paréntesis abierto y antes del cerrado.

Líneas en blanco

En cuanto a las líneas en blanco se dejó una línea en blanco entre métodos, así como entre las variables locales de un método y su primera sentencia.

Clases y objetos

Para los nombres de las clases se utilizó la notación Pascal empleando mayúscula en la primera letra de cada palabra que la forma, ejemplo: **GenerarReporteBitacora.java**. Además no se utilizó en sus nombres ningún prefijo o abreviaturas.

Para los nombres de los atributos de las clases se comenzó con la primera letra en minúscula estando en correspondencia con el tipo de dato al que se refiere, en caso de los nombres compuestos, se utilizó el estilo lowerCamelCase, ejemplo: **listaIpsString**.

En los métodos compuestos, para la primera letra se comenzó con minúscula, y la primera letra de las siguientes palabras que lo forma con mayúscula. Ejemplo: **repUsMoIp().**

En este capítulo se presentaron los elementos correspondientes a la fase de implementación con el objetivo de dar respuesta a los requisitos funcionales y no funcionales especificados durante la investigación. Se presentaron los elementos que componen el Modelo de Datos, para un mejor entendimiento de la estructura de la Base de Datos. Se mostraron los diagramas pertenecientes al Modelo de Implementación para una asimilación de la relación existente entre los componentes tanto de hardware como de software. Se expusieron aspectos a tener en cuenta para el tratamiento de excepciones y de esta forma evitar fallas no controladas en el sistema. En cuanto a la seguridad se presentaron un grupo de procedimientos realizados con el objetivo de preservar la confidencialidad, integridad y disponibilidad a tener en cuenta en el sistema. Se presentaron estrategias y estilos de codificación utilizados durante la codificación para lograr limpieza y claridad en el código, lo cual ayuda a su legibilidad.

Conclusiones

Con el desarrollo de la presente investigación se da solución al problema inicial, a partir de las tareas planteadas y cumpliendo con el objetivo propuesto, por lo que se puede arribar a las siguientes conclusiones:

- El estudio de los sistemas desarrollados relacionados con el campo de acción sirvió de base para comprender el alcance de la presente investigación, sin embargo se pudo apreciar que dichos sistemas no satisfacen las necesidades determinadas en el objetivo general.
- El análisis de los procesos relacionados con las acciones que se realizan actualmente en la bitácora del Sistema de Información Hospitalaria, permitió comprender la necesidad de desarrollar funcionalidades para la generación reportes a partir de trazas de la bitácora del sistema.
- Las tecnologías y herramientas empleadas en el desarrollo de la solución propuesta, basándose en la arquitectura y el patrón de diseño descrito, facilitaron la obtención de los artefactos necesarios para su desarrollo.
- Los artefactos que guiaron el proceso de implementación como respuesta a las necesidades del cliente, permitieron la obtención de un sistema capaz de generar reportes a partir de las trazas de la bitácora del Sistema de Información Hospitalaria del Centro de Informática Médica con el objetivo de permitirle a los administradores tener un mayor control de las acciones que realizan los usuarios en el sistema.

Recomendaciones

Para dar continuidad a la presente investigación se propone la siguiente recomendación:

- Utilizar técnicas de Inteligencia Artificial para la optimización de la generación de reportes en la solución propuesta.

Referencias bibliográficas

1. Cuba. Ministerio de la Informática y las Comunicaciones. Informatización de la Sociedad. [En línea] [Accedido el: 4 de diciembre de 2013]. Disponible en: http://www.mic.gov.cu/sitiomic/hinfosoc.asp
2. Misión. Portal de la Universidad de las Ciencias Informáticas [En línea]. [Accedido el: 5 de diciembre de 2013]. Disponible en: http://www.uci.cu/?q=mision
3. CATALOGO. Portal de la Universidad de las Ciencias Informáticas [En línea]. [Accedido el: 5 de diciembre de 2013]. Disponible en: http://www.uci.cu/sites//default/files/files/CATALOGO.pdf
4. Definición. De, Definición de reporte - Qué es, Significado y Concepto. [En línea]. [Accedido el: 13 de diciembre de 2013]. Disponible en: http://definicion.de/reporte/
5. Bitácora. GENAP. [En línea]. [Accedido el: 13 de diciembre de 2013]. Disponible en: http://www.genap.com.mx
6. ITSON, Introducción a los Sistemas de Información, Concepto de Sistema. [En línea]. [Accedido el: 13 de diciembre de 2013]. Disponible en: http://biblioteca.itson.mx/oa/dip_ago/introduccion_sistemas/p3.htm
7. Sistema Integral Hospitalario (SIHO). [En línea]. [Accedido el: 15 de diciembre de 2013]. Disponible en: http://www.sihosys.com/Sistema_Integral_Hospitalario_%28SiHO%29.pdf
8. Sistema Integral Hospitalario (SIHO). [En línea]. [Accedido el: 15 de diciembre de 2013]. Disponible en: http://dsp.mx/sistemas-de-informacion-hospitalaria
9. Virtus Group - xHosp - Sistema integral para la administración hospitalaria. [En línea]. [Accedido el: 16 de diciembre de 2013]. Disponible en: http://www.virtus.com.mx/xhosp/

10. Instrucciones para la preparación de Ponencias para Informática 2009 – 2, [En línea]. [Accedido el: 17 de diciembre de 2013]. Disponible en: http://www.informatica2013.sld.cu/index.php/informaticasalud/2013/paper/viewFile/48/2
11. Galenhos : Sistema Integrado de Gestión Hospitalaria , Módulos de GalenHos. [En línea]. [Accedido el: 18 de diciembre de 2013]. Disponible en: http://www.politicasensalud.org/blogpolsalud/?p=106
12. Marble Station: Web personal de Sergi Blanco Cuaresma. [En línea]. [Accedido el: 20 de diciembre de 2013]. Disponible en: http://www.marblestation.com/
13. CMMI LARMAN, C. UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. 2ed. [Accedido el: 21 de diciembre de 2013]. Disponible en: http://publidisa.com/PREVIEW-LIBRO-9788483229279.pdf
14. Hernández Orallo, Enrique. [En línea]. [Accedido el: 21 de diciembre de 2013.]. Disponible en: www.disca.upv.es/enheror/pdf/ActaUML.PDF
15. The Eclipse Foundation. The Eclipse Foundation. [En línea]. [Accedido el: 25 de diciembre de 2013.]. Disponible en: http://www.eclipse.org
16. w3c. [En línea]. febrero de 2008. [Accedido el: 14 de enero de 2014.] Disponible en: http://www.w3c.es/Divulgacion/GuiasBreves/XHTML
17. w3c. [En línea]. enero de 2008. [Accedido el: 14 de enero de 2014.] Disponible en: http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo
18. Sun Developer Network. [En línea]. [Accedido el: 15 de enero de 2014.] Disponible en: http://java.sun.com/products/ejb/
19. Java Persistence API. [En línea]. [Accedido el: 16 de enero de 2014.] Disponible en: http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html

20. juntadeandalucia. [En línea]. [Accedido el: 16 de enero de 2014.] Disponible en: http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/96
21. Mundo Geek. [En línea]. . [Accedido el: 16 de enero de 2014.] Disponible en: http://mundogeek.net/archivos/2007/01/27/hibernate/
22. Almirón, Cristóbal González. Adictos al Trabajo. [En línea] [Accedido el: 21 de enero de 2014.] Disponible en: http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava
23. Suárez, Jose Manuel Sánchez. Adictos al Trabajo. Introducción a RichFaces. Febrero de 2010. [En línea]. [Accedido el: 21 de enero de 2014.] Disponible en: http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsfIntro.
24. ajaxya. [En línea]. [Accedido el: 22 de enero de 2014.] http://www.ajaxya.com.ar/temarios/descripcion.php?cod=8&punto=1.
25. Ramos, Juan Alonso. adictosaltrabajo. [En línea] de abril de 2007. [Accedido el: 23 de enero de 2013.] http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf
26. Suárez Sánchez, Jose Manuel. Adictos al Trabajo. Facelets en JSF 2: sistema de plantillas y componentes por composición. abril de 2010. [En línea]. [Accedido el: 25 de enero de 2014.] http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jsf2FaceletsTemplatesAndCompositeComponents
27. Sánchez Rosas, Juan Eladio. antartec. [En línea] 2008. [Accedido el: 25 de enero de 2014.] Disponible en: http://blogs.antartec.com/desarrolloweb/2008/12/facelets-y-jsf-uso-de-templates/
28. Seamframework.org. [En línea]. [Accedido el: 26 de enero de 2014]. Disponible en: http://www.seamframework.org/
29. Desarrollo JAVA EE. [En línea] [Accedido el: 26 de enero de 2014.] Disponible en: http://www.neuronet.cl/downloads/J2EE.pdf

30. Ayuda BitTorrent. [En línea] [Accedido el: 26 de enero de 2014.] Disponible en: http://www.ayudabittorrent.com/jre
31. Jaramillo M, Wilmer. Fedora People. JBoss Application Server. [En línea] 2006. [Accedido el: 27 de enero de 2014.] Disponible en: http://wilmer.fedorapeople.org/files/presentations/JBoss.pdf
32. Free Download Manager. Visual Paradigm para UML. [En línea] [Accedido el: 27 de enero de 2014.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/
33. PostgreSQL-Investigacion. [En línea] [Accedido el: 27 de enero de 2014.] http://es.scribd.com/doc/36570462/postgreSQL-investigacion. aseqw-7854
34. SAP Business Objects. Chile SAP Business Objects l Crystal Reports l Crystal Reports Visual Advantage. 2008a. [En línea]. [Accedido el: 28 de enero 2014]. Disponible en: http://www.sap.com/chile/solutions/sapbusinessobjects/sme/reporting/crystalreports/index.epx
35. JasperForge.org. JasperForge.org. [En línea] Jaspersoft Corporation, 2009. [Accedido el: 3 de febrero de 2014.] http://jasperforge.org/website/jasperreportswebsite/trunk/highlights.html?group_id=252
36. Herrera, Cristian. Informes en Java con iReports. De abril de 2005 [En línea]. [Accedido el: 4 de febrero de 2014.] Disponible en: http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ireport
37. iReport Designer | Jaspersoft Community. [En línea]. [Accedido el: 4 de febrero de 2014.] Disponible en: http://community.jaspersoft.com/project/ireport-designer
38. EDUCATION. Modelo de dominio. p 10.
39. Modelo Dominio. [En línea]. [Accedido el: 11 de marzo de 2014]. Disponible en: http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf

40. Requerimientos funcionales y no funcionales. [En línea]. [Accedido el: 12 de marzo de 2014]. Disponible en: http://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales.
41. Sommerville, Ian. Ingeniería del software.Séptima edición. Madrid: Pearson Educación.p.111
42. Pressman, Roger S. 1998. Ingeniería del software. Un enfoque práctico. Mc Graw Hill: s.n., 1998. p.256
43. Pressman, Roger S. 1998. Ingeniería del software. Un enfoque práctico. Mc Graw Hill: s.n., 1998. P.237.
44. Pressman, Roger S. 1998. Ingeniería del software. Un enfoque práctico. Mc Graw Hill: s.n., 1998. p.219
45. Larman, craig. Uml y patrones. Introducción al análisis y diseño orientado a objetos. . PRENTICE HAL: 17, 1999. 970-17-0261-1.p.191
46. Larman, craig. Uml y patrones. Introducción al análisis y diseño orientado a objetos. . PRENTICE HAL: 17, 1999. 970-17-0261-1.p.256
47. Diagrama de Interacción. [En línea]. [Accedido el: 15 de marzo de 2014]. [En línea]. [Accedido el: 23 de abril de 2014]. Disponible en: http://labredes.itcolima.edu.mx/fundamentosbd/sd_u2_1.htm.
48. Modelo de Datos 1ª parte». [En línea]. Disponible en: http://blogs.ua.es/mu171m3d14/2011/04/17/modelo-de-datos-1%C2%AA-parte/. [Accedido el: 23 de abril de 2014].
49. MeRinde - Modelo de Implementación. In: [En línea]. [Accedido el: 28 de abril de 2014]. Disponible en:http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=29
50. Diagrama de Componentes. In: [En línea]. [Accedido el: 28 de abril de 2014]. Disponible en: http://diagramacomponente.blogspot.com/.

51. Manejo de Errores Usando Excepciones Java. Programación en Castellano. In: [En línea]. [Accedido el: 1 de mayo de 2014]. Disponible en:http://www.programacion.com/articulo/manejo_de_errores_usando_excepciones_java_98/2.
52. King.P , Naughton.P. , Convenciones de código para el lenguaje de programación JavaTM. 2001. [En línea]. [Accedido el: 23 de abril de 2014]. Disponible en: http://www.um.es/docencia/vjimenez/ficheros/practicas/ConvencionesCodigoJava.pdf
53. Licencia Pública GNU. [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en: http://gugs.sindominio.net/gnu-gpl/gples.html
54. Riehle, Dirk (2000), Framework Design: A Role Modeling Approach, Swiss Federal Institute of Technology [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en:http://www.riehle.org/computer-science/research/dissertation/diss-a4.pdf
55. Entorno de Desarrollo Integrado (IDE). | fergarciac. [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en: http://fergarciac.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/
56. PDF Definición y explicación. [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en: http://www.masadelante.com/faqs/pdf
57. PHP: ¿Qué es PHP? - Manual. [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en: http://www.php.net/manual/es/intro-whatis.php
58. Definición de MySQL - Significado y definición de MySQL. [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en: http://www.mastermagazine.info/termino/6051.php
59. Definición de html - Qué es, Significado y Concepto. [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en: http://definicion.de/html/

Bibliografía

- Almirón, Cristóbal González. Adictos al Trabajo. [En línea] Marzo de 2009. [Accedido el: 21 de enero de 2014.] Disponible en: http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava
- Areba, Jesus Barranco de. Metodología Del Análisis Estructurado de Sistemas. s.l: Univ. Pontificia Comillas, 2002. 8484680436, 9788484680437Gallego, Juan Pablo Gómez. Fundamentos de la metodología RUP. 2007.
- Ayuda BitTorrent. [En línea] [Accedido el: 26 de enero de 2014.] Disponible en: http://www.ayudabittorrent.com/jre
- Bases de datos: Modelos de datos. [En línea]. [Accedido el: 23 de abril de 2014]. Disponible en: http://elies.rediris.es/elies9/4-2.htm
- Bitácora. GENAP. [En línea]. 2008 [Accedido el: 13 de diciembre de 2013]. Disponible en: http://www.genap.com.mx
- CATALOGO. Portal de la Universidad de las Ciencias Informáticas [En línea]. [Accedido el: 5 de diciembre de 2013]. Disponible en: http://www.uci.cu/sites//default/files/files/CATALOGO.pdf
- CMMI LARMAN, C. UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. 2ed. [Accedido el: 21 de diciembre de 2013]. Disponible en: http://publidisa.com/PREVIEW-LIBRO-9788483229279.pdf
- Codificación | Recursos web. [En línea]. [Accedido el: 1 de mayo de 2014]. Disponible en: http://recursosweb.unam.mx/recursos-web/creacion-de-paginas-web/estandares-de-codificacion/
- ConvecionesCodigoJava - ConvencionesCodigoJava.pdf [En línea]. [Accedido el: 1 de mayo de 2014]. Disponible en: http://www.um.es/docencia/vjimenez/ficheros/practicas/ConvencionesCodigoJava.pdf
- Cristóbal González Almirón «Introducción a JSF Java». 2009. [En línea]. [Accedido el: 19 de diciembre de 20013]. Disponible en: http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava

- Cuba. Ministerio de la Informática y las Comunicaciones. Informatización de la Sociedad. [En línea] [Accedido el: 4 de diciembre de 2013]. Disponible en: http://www.mic.gov.cu/sitiomic/hinfosoc.asp
- Definición. De, Definición de reporte - Qué es, Significado y Concepto. [En línea]. [Accedido el: 13 de diciembre de 2013]. Disponible en: http://definicion.de/reporte/
- Definición de html - Qué es, Significado y Concepto. [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en: http://definicion.de/html/
- Definicion_Estandares_2.doc [En línea]. [Accedido el: 1 de mayo de 2014]. Disponible en: http://www.inf.utfsm.cl/~visconti/xp/Definicion_Estandares_2.doc
- Definición de modelo de datos - Qué es, Significado y Concepto. [En línea]. [Accedido el: 23 De abril de 2014]. Disponible en: http://definicion.de/modelo-de-datos/
- Definición de MySQL - Significado y definición de MySQL. [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en: http://www.mastermagazine.info/termino/6051.php
- Desarrollo JAVA EE. [En línea] [Accedido el: 26 de enero de 2014.] Disponible en: http://www.neuronet.cl/downloads/J2EE.pdf
- Diagrama de Componentes. [En línea]. [Accedido el: 28 De abril de 2014]. Disponible en: http://diagramacomponente.blogspot.com/
- Diagrama de Interacción. [En línea]. [Accedido el: 15 de marzo de 2014]. Disponible en: http://uxmcc1.iimas.unam.mx/~cursos/Objetos/Cap18/cap18.html
- Dugarte, Ana, y otros. 2009. Lenguaje Unificado de Modelado. [En línea] 2009. [Accedido el: 19 de diciembre de 20013]. Disponible en: http://www.oocities.org/es/annadugarte/ads1/PRINCIPAL.htm
- Estándares de codificación y buenas prácticas | inside software quality - Un blog de Optimyth en español. [En línea]. [Accedido el: 1 de mayo de 2014]. Disponible en: http://blog.optimyth.com/es/2013/11/estandares-de-codificacion-y-buenas-practicas
- EDUCATION. Modelo de dominio. p 10.
- Estándares de codificación definición - Buscar con Google. [En línea]. [Accedido el: 1 de mayo de 2014]. Disponible en: https://www.google.com.cu/#q=estandares+de+codificacion+definición

- FranVarVil: Excepciones con Try/Catch/Finally (Programación Java). [En línea]. [Accedido el: 1 de mayo de 2014]. Disponible en: http://franvarvil.blogspot.com/2013/04/excepciones-con-trycatchfinally.html
- Free Download Manager. Visual Paradigm para UML. [En línea] marzo de 2007. [Accedido el: 27 de enero de 2014.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/
- Galenhos : Sistema Integrado de Gestión Hospitalaria , Módulos de GalenHos. [En línea]. [Accedido el: 18 de diciembre de 2013]. Disponible en: http://www.politicasensalud.org/blogpolsalud/?p=106
- Gremy F. Medical informatics in developing countries, balance sheet and prospective. Journal of Educational Computing Research 1998 ;(2):235-252.
- Hernández Orallo, Enrique. [En línea]. [Accedido el: 21 de diciembre de 2013.]. Disponible en: www.disca.upv.es/enheror/pdf/ActaUML.PDF
- Herrera, Cristian. Informes en Java con iReports. De abril de 2005 [Accedido el: 4 de febrero de 2014.] [En línea]. Disponible en: http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ireport
- Instrucciones para la preparación de Ponencias para Informática 2009 – 2, [En línea]. [Accedido el: 17 de diciembre de 2013]. Disponible en: http://www.informatica2013.sld.cu/index.php/informaticasalud/2013/paper/viewFile/48/2
- ITSON, Introducción a los Sistemas de Información, Concepto de Sistema. [En línea]. [Accedido el: 13 de diciembre de 2013]. Disponible en: http://biblioteca.itson.mx/oa/dip_ago/introduccion_sistemas/p3.htm
- Jaramillo M, Wilmer. Fedora People. JBoss Application Server. [En línea] 2006. [Accedido el: 27 de enero de 2014.] Disponible en: http://wilmer.fedorapeople.org/files/presentations/JBoss.pdf
- JasperForge.org. JasperForge.org. [En línea] Jaspersoft Corporation, 2009. [Accedido el: 3 de febrero de 2014.] http://jasperforge.org/website/jasperreportswebsite/trunk/highlights.html?group_id=252

- Java Persistence API. [En línea]. [Accedido el: 16 de enero de 2014.] Disponible en: http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html
- juntadeandalucia. [En línea]. . [Accedido el: 16 de enero de 2014.] Disponible en: http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/96
- King.P, Naughton.P. , Convenciones de código para el lenguaje de programación JavaTM. 2001. [En línea]. [Accedido el: 23-abr-2014].Disponible en: http://www.um.es/docencia/vjimenez/ficheros/practicas/ConvencionesCodigoJava.pdf
- Larman, craig. Uml y patrones. Introducción al análisis y diseño orientado a objetos. . PRENTICE HAL: 17, 1999. 970-17-0261-1.p.191
- Licencia Pública GNU. [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en: http://gugs.sindominio.net/gnu-gpl/gples.html
- Manejo de Errores Usando Excepciones Java. Programación en Castellano. [En línea]. [Accedido el: 1 de mayo de 2014]. Disponible en: http://www.programacion.com/articulo/manejo_de_errores_usando_excepciones_java_98/
- Manejo de Errores Usando Excepciones Java. Programación en Castellano. [En línea]. [Accedido el: 1 de mayo de 2014]. Disponible en: http://www.programacion.com/articulo/manejo_de_errores_usando_excepciones_java_98/2
- Manejo de Errores Usando Excepciones Java. Programación en Castellano. [En línea]. [Accedido el: 1 de mayo de 2014]. Disponible en: http://www.programacion.com/articulo/manejo_de_errores_usando_excepciones_java_9
- Manejo de Errores Usando Excepciones Java. Programación en Castellano. In: [En línea]. [Accedido el: 1 mayo 2014]. Disponible en:http://www.programacion.com/articulo/manejo_de_errores_usando_excepciones_java_98/2.
- Marble Station: Web personal de Sergi Blanco Cuaresma». [En línea]. [Accedido el: 20 de diciembre de 2013]. Disponible en: http://www.marblestation.com/
- MeRinde - Modelo de Implementación. [En línea]. [Accedido el: 28 De abril de 2014]. Disponible en: http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=291

- MeRinde - Modelo de Implementación. In: [En línea]. [Accedido el: 28 de abril2014]. Disponible en:http://merinde.net/index.php?option=com_content&task=view&id=495&Itemid=29
- Microsoft PowerPoint - BDa-t2 - BDa-t2.pdf [En línea]. [Accedido el: 23 De abril de 2014]. Disponible en: http://alarcos.inf-cr.uclm.es/doc/bda/doc/teo/2/BDa-t2.pdf
- Microsoft PowerPoint - 02 - Modelos de datos ER-UML-relacional.ppt - 02 - Modelos de datos ER-UML-relacional.pdf [En línea]. [Accedido el: 23 de abril de 2014]. Disponible en: http://personales.unican.es/zorrillm/BasesDatos/02%20-%20Modelos%20de%20datos%20ER-UML-relacional.pdf
- Misión. Portal de la Universidad de las Ciencias Informáticas [En línea]. [Accedido el: 5 de diciembre de 2013]. Disponible en: http://www.uci.cu/?q=mision.
- Modelado de Datos. [En línea]. [Accedido el: 23 de abril de 2014]. Disponible en: http://ict.udlap.mx/people/carlos/is341/bases02.html
- Modelo de Datos 1ª parte». [En línea]. Disponible en: http://blogs.ua.es/mu171m3d14/2011/04/17/modelo-de-datos-1%C2%AA-parte/. [Accedido el: 23-abr-2014].
- Modelo de Datos 1ª parte. [En línea]. [Accedido el: 23 De abril de 2014]. Disponible en: http://blogs.ua.es/mu171m3d14/2011/04/17/modelo-de-datos-1%C2%AA-parte/
- Patrones de diseño. [En línea]. [Accedido el: 12 de marzo de 2014]. Disponible en: http://sourcemaking.com/patr%C3%B3n-de-dise%C3%B1o
- PDF Definición y explicación. [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en: http://www.masadelante.com/faqs/pdf
- PHP: ¿Qué es PHP? - Manual. [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en: http://www.php.net/manual/es/intro-whatis.php
- Pressman, Roger S. 1998. Ingeniería del software. Un enfoque práctico. Mc Graw Hill: s.n., 1998. p.256

- PostgreSQL-Investigacion. [En línea] [Accedido el: 27 de enero de 2014.] http://es.scribd.com/doc/36570462/postgreSQL-investigacion. aseqw-7854
- Ramos, Juan Alonso. adictosaltrabajo. [En línea] de abril de 2007. [Citado el: 23 de enero de 2013.] http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=Ajax4Jsf
- Report Designer | Jaspersoft Community. [En línea]. [Accedido el: 4 de febrero de 2014.] Disponible en: http://community.jaspersoft.com/project/ireport-designer
- Requerimientos funcionales y no funcionales. [En línea]. [Accedido el: 12 de marzo de *2014]. Disponible en:* http://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales.
- Riehle, Dirk (2000), Framework Design: A Role Modeling Approach, Swiss Federal Institute of Technology [En línea]. [Accedido el: 10 de mayo de 2014]. Disponible en:http://www.riehle.org/computer-science/research/dissertation/diss-a4.pdf
- Sánchez Rosas, Juan Eladio. antartec. [En línea]. 17 de 12 de 2008. [Accedido el: 25 de enero de 2014.] Disponible en: http://blogs.antartec.com/desarrolloweb/2008/12/facelets-y-jsf-uso-de-templates/
- SAP Business Objects. Chile SAP Business Objects l Crystal Reports l Crystal Reports Visual Advantage. 2008a. [En línea]. [Accedido el: 28 de enero 2014.] Disponible en: http://www.sap.com/chile/solutions/sapbusinessobjects/sme/reporting/crystalreports/index.epx
- Seamframework.org. [En línea]. [Accedido el: 26 de enero de 2014.] Disponible en: http://www.seamframework.org/
- Seguridad de sistemas. [En línea]. [Accedido el: 1 mayo de 2014]. Disponible en: http://www.duiops.net/hacking/seguridad-sistemas.htm
- Sistema Integral Hospitalario (SIHO). [En línea]. [Accedido el: 15 de diciembre de 2013]. Disponible en: http://www.sihosys.com/Sistema_Integral_Hospitalario_%28SiHO%29.pdf
- Sistema Integral Hospitalario (SIHO). [En línea]. [Accedido el: 15 de diciembre de 2013]. Disponible en: http://dsp.mx/sistemas-de-informacion-hospitalaria

- Sommerville, Ian. Ingeniería de Software 7ma Edición.2005. Madrid: Pearson Educación. ISBN: 84-7829-074-5
- Sun Developer Network. [En línea]. [Accedido el: 15 de enero de 2014.] Disponible en: http://java.sun.com/products/ejb/
- Suárez, Jose Manuel Sánchez. Adictos al Trabajo. Introducción a RichFaces. Febrero de 2010. [En línea]. [Accedido el: 21 de enero de 2014.] Disponible en: http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsfIntro
- Suárez Sánchez, Jose Manuel. Adictos al Trabajo. «Facelets en JSF 2: sistema de plantillas y componentes por composición. de abril de 2010. [En línea]. [Accedido el: 25 de enero de 2014.] http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jsf2FaceletsTemplatesAndCompositeComponents
- Try and Catch. [En línea]. [Accedido el: 1 mayo de 2014]. Disponible en: http://www.webtutoriales.com/articulos/try-and-catch
- Virtus Group - xHosp - Sistema integral para la administración hospitalaria. [En línea]. [Accedido el: 16 de diciembre de 2013]. Disponible en: http://www.virtus.com.mx/xhosp/
- w3c. [En línea]. febrero de 2008. [Accedido el: 14 de enero de 2014.] Disponible en: http://www.w3c.es/Divulgacion/GuiasBreves/XHTML
- w3c. [En línea]. enero de 2008. [Accedido el: 14 de enero de 2014.] Disponible en: http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo

Glosario de términos

GNU GPL: Es la conocida GNU Public License (GPL), versión 2 (de junio de 1.991), que cubre la mayor parte del software de la Free Software Foundation. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. Esta licencia fue creada originalmente por Richard Stallman fundador de la Free Software Foundation (FSF) para el proyecto GNU. (53)

Framework: La palabra inglesa "framework" (marco de trabajo) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. (54)

IDE: Un entorno de desarrollo integrado, es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic. (55)

PDF: PDF, de Portable Document Format (formato de documento portable) es el formato de archivos desarrollado por Adobe Systems y creado con los programas Adobe Acrobat Reader, Acrobat Capture, Adobe Distiller, Adobe Exchange, y el plugin Amber de Adobe Acrobat. (56)

PHP: PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. (57)

MySQL: El software MySQL proporciona un servidor de base de datos SQL (Structured Query Language) veloz, multi-hilo, multiusuario y robusto. El servidor está proyectado tanto para sistemas críticos en producción soportando intensas cargas de trabajo como para empotrarse en sistemas de desarrollo masivo de software. (58)

HTML: HTML es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto. (59)

Anexos

Anexo 1. Guía de Observación. Aspectos a tener en cuenta.

Objetivo: Seleccionar aquellos sistemas que cuenten con funcionalidades para la generación de reportes a partir de trazas de la bitácora y resulten de utilidad para la realización de la investigación.

Generación de reportes

- Tipos de reportes que se generan en ellos (dinámicos o estáticos).
- Sistemas que generan reportes a partir de trazas de la bitácora.
- Módulos en los cuales se generan reportes.

Seguridad

- Mecanismos de seguridad establecidos.
- Roles definidos para el acceso al sistema.

Estructura

- Módulos que poseen.
- Composición de la base de datos.
- Estructura del módulo Configuración.

Elementos generales

- Principales funcionalidades.
- Tecnologías de desarrollo (libres o propietarias).
- Tipo de aplicación (web o de escritorio).
- País donde se desarrollaron.

Anexo 2. Interfaz de usuario correspondiente a la bitácora actual del HIS.

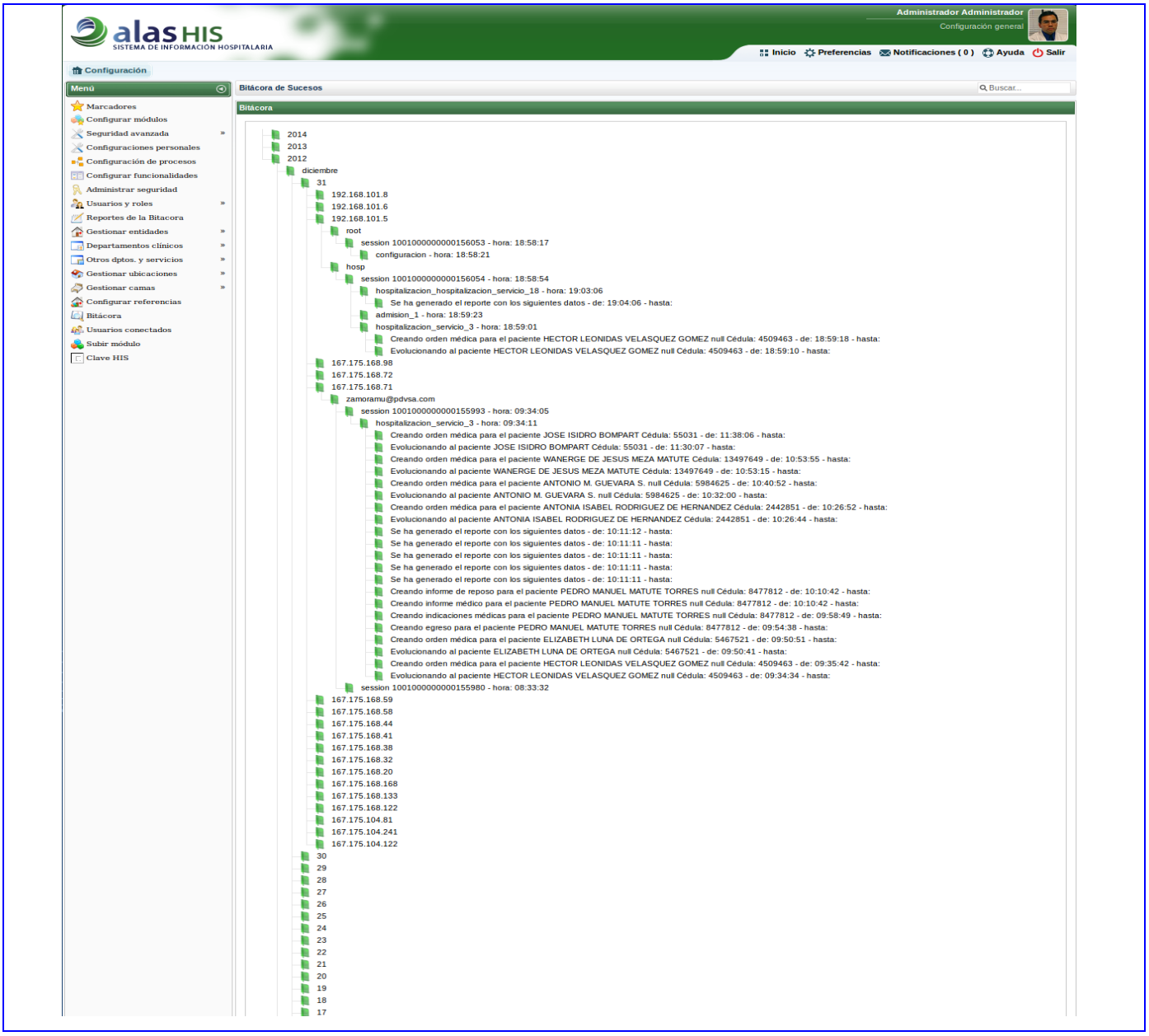


Figura 16. Interfaz de usuario correspondiente a la bitácora actual del HIS.

Anexo 3. Guía con aspectos a considerar en la entrevista.

Objetivo: Obtener criterios para comprender los elementos necesarios para el desarrollo de la solución informática.

1. Procesos que se realizan actualmente en el HIS relacionados con la bitácora.
2. Estructura de la bitácora actual del HIS.
3. Tipo de aplicación a desarrollar (sistema independiente o funcionalidad integrada al sistema HIS).
4. Roles que van a interactuar con la solución informática a desarrollar.
5. Funcionalidades que el sistema debe permitir.
6. Requisitos no funcionales a tener en cuenta para la creación de las funcionalidades definidas por el cliente.
7. Tipos de reportes a generar (dinámicos o estáticos).
8. Estructura de las tablas de la base de datos que almacenan la información referente a las trazas de las acciones de los usuarios en el sistema.
9. Tiempo de almacenamiento de la informacion en la base de datos del HIS de las trazas de los usuarios.

Anexo 4. Listado de reportes de la bitácora generados en el HIS.

Listado de Reportes	
1. Reporte por usuario, módulo e IP.	17. Reporte por usuario, módulo, IP, fecha y acción.
2. Reporte por usuario, módulo y fecha.	18. Reporte por usuario, módulo, IP, fecha y atributo
3. Reporte por usuario, módulo y acción.	19. Reporte por usuario, módulo, IP, hora y acción.
4. Reporte por usuario, módulo y hora.	20. Reporte por usuario, módulo, IP, hora y atributo
5. Reporte por usuario, módulo y atributo.	21. Reporte por usuario, módulo, IP, acción y atributo
6. Reporte por usuario, módulo, IP y fecha.	22. Reporte por usuario, módulo, fecha, hora y

	acción.
7. Reporte por usuario, módulo, IP y hora.	23. Reporte por usuario, módulo, fecha, hora y atributo.
8. Reporte por usuario, módulo, IP y acción.	24. Reporte por usuario, módulo, fecha, acción y atributo.
9. Reporte por usuario, módulo, IP y atributo.	25. Reporte por usuario, módulo, hora, acción y atributo.
10. Reporte por usuario, módulo, fecha y hora.	26. Reporte por usuario, módulo, IP, fecha, hora y acción.
11. Reporte por usuario, módulo, fecha y acción.	27. Reporte por usuario, módulo, IP, fecha, hora y atributo.
12. Reporte por usuario, módulo, fecha y atributo.	28. Reporte por usuario, módulo, fecha, hora, acción y atributo.
13. Reporte por usuario, módulo, acción y hora.	29. Reporte por usuario, módulo, IP, hora, acción y atributo.
14. Reporte por usuario, módulo, acción y atributo.	30. Reporte por usuario, módulo, IP, fecha, acción y atributo.
15. Reporte por usuario, módulo, hora y atributo.	31. Reporte por usuario, módulo, IP, fecha, hora, acción y atributo.
16. Reporte por usuario, módulo, IP, fecha y hora.	

Tabla 17. Listado de reportes de la bitácora que se generan en el HIS.

Anexo 5. Interfaz de usuario para la selección de parámetros.

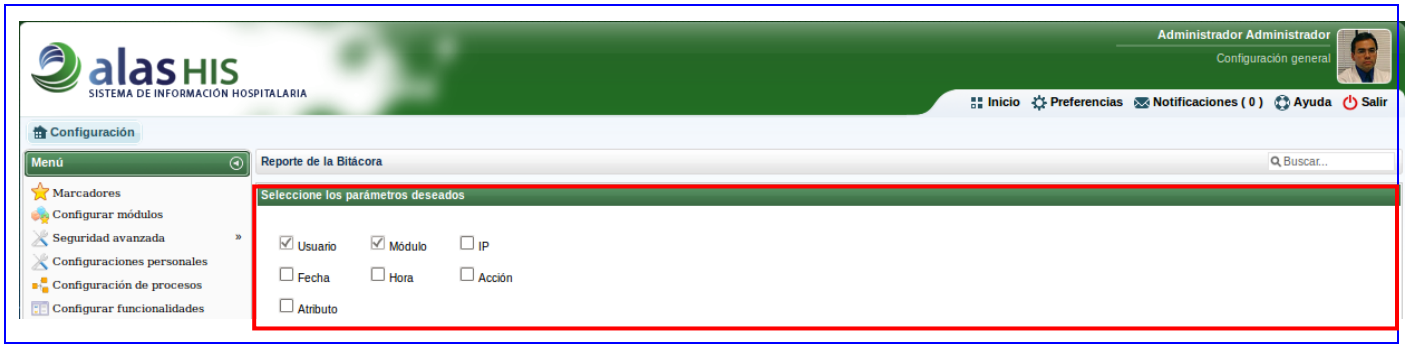


Figura 17. Parametros a seleccionar para la generación de reportes.

Anexo 6. Interfaz de usuario para la selección de los filtros.

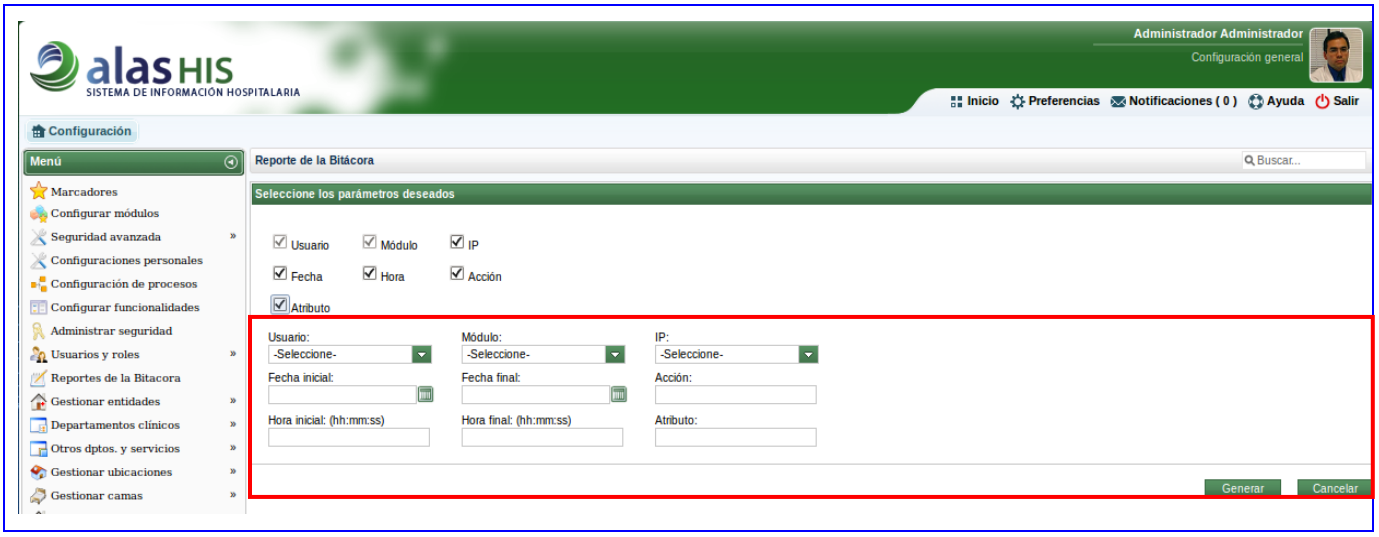


Figura 18. Ejemplo de filtros para la generación de un reporte.

Anexo 7. Interfaz de usuario que permite seleccionar el formato de salida de un reporte.

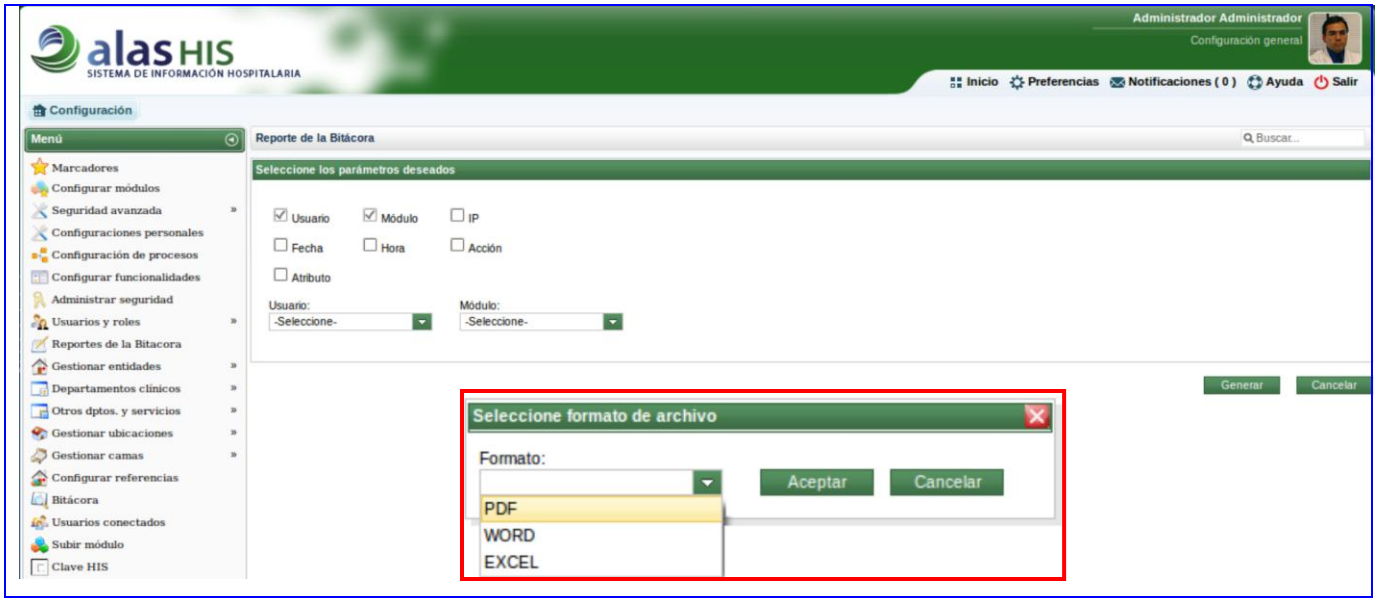
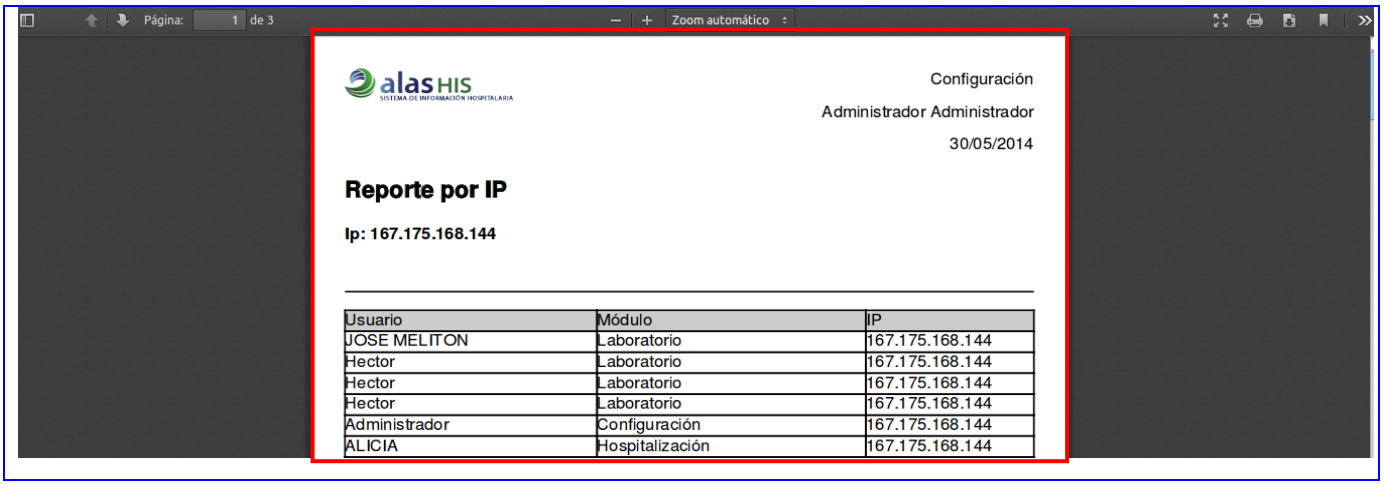


Figura 19. Selección de formatos de salida para la generación de un reporte.

Anexo 8. Ejemplo de un reporte exportado a PDF.



Usuario	Módulo	IP
JOSE MELITON	Laboratorio	167.175.168.144
Hector	Laboratorio	167.175.168.144
Hector	Laboratorio	167.175.168.144
Hector	Laboratorio	167.175.168.144
Administrador	Configuración	167.175.168.144
ALICIA	Hospitalización	167.175.168.144

Figura 20. Ejemplo de un reporte exportado a formato PDF.