

Universidad de las Ciencias Informáticas

Facultad 2



**Título: Desarrollo del módulo Salud Integral del
subsistema Sala Situacional del SIGEP en el
régimen Intramuros.**

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor(es): Aymee Regla Castillo Pozo

José Antonio Rivera Hernández

Tutor(es): Ing. Yannia Moreira Gamboa

Ciudad de La Habana, Mayo 2011

Declaratoria de autoñía

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Aymee Regla Castillo Pozo

José Antonio Rivera Hernández

Ing. Yannia Moreira Gamboa

Datos de contacto

Graduada en la CUJAE 2004-2005. Tiene 5 años de experiencia.

Fue especialista en la DP5 y lleva 4 años de especialidad.

Agradecimientos

Compartidos

A nuestra tutora Yannia que a pesar de estar lejos siempre hemos contado con su apoyo, en especial a la líder de proyecto, Karina que estuvo apoyándonos en todo momento. También queremos agradecer a Gueorgui, Roberto y René por prestarnos su valioso tiempo y de poder contar con ellos en la recta final.

A todos sinceramente, muchas gracias.

De Aymee Regla

Primeramente agradecer a todos mis compañeros de proyecto por siempre haber compartido las experiencias de trabajo y habernos hecho, los unos a los otros, cada vez mejores profesionales. A todos a mis amigos y amigas que de una forma u otra han aportado su granito de arena a mi formación.

De José Antonio

A mis padres por dedicar su vida a guiarme por el buen camino y estar conscientes de que podía lograrlo.

A mis abuelos que siempre me aconsejaron y estuvieron ahí para mí.

A mis tías y primos que confiaron en mí desde el principio.

A mis compañeros de la infancia y vecinos que aunque estábamos lejos siempre estuvieron al tanto de mí.

A mis compañeros con los que he compartido estos últimos años, esos que estuvieron presente en los buenos y malos momentos, los que me demostraron que en ellos se podía confiar, que son verdaderos amigos.

A todos los que de una forma u otra supieron apoyarme y confiar en mí, a aquellos que han contribuido a mi formación académica y personal.

Dedicatoria

Dedicar esta tesis a mi amiga incondicional, que es mi mamá quien me dio la vida y que sin ella no hubiera sido posible hacer realidad este sueño.

A mi papá que a pesar de su carácter siempre ha estado apoyándome y darle las gracias por nunca dudar de mi.

A mi novio Gabriel, que estos dos años a su lado me han iluminado la vida.

A mi abuela Isabel que siempre ha esperado lo mejor de mí.

A mi tía Ita, la cual extraño muchísimo por no encontrarse físicamente entre nosotros.

A mis tíos Javier, Lázaro y Virgen los cuales de una forma u otra han contribuido con mi formación..

A mis primos Dirzeus y Thalia para los cuales pretendo ser ejemplo.

A mi primo Yordan el cual ha sido como un hermano, a su esposa Elizabeth y a la pequeñita Isabel porque me han apoyado y ayudado a lo largo de mi carrera.

A todas mis hermanas por brindarme su apoyo y en especial a Isabel Companioni.

A todos mis compañeros con los cuales he compartido los 5 años de la carrera y que nunca voy a olvidar.

Aymee

A toda mi familia en general por todo su apoyo, confianza y paciencia.

A mis padres, a quienes se lo debo todo en esta vida.

A mis abuelos que siempre soñaron con verme graduado y me han demostrado todo el cariño del mundo.

A mis tías, que siempre me demostraron que podía contar con ellas, y me quieren como si fuera un hijo más.

A mis primos, que mas que primos son hermanos.

José

Resumen

Para contribuir a la solución de los problemas por los que atraviesa el Sistema Penitenciario Venezolano, se decidió crear el Sistema de Gestión Penitenciaria (SIGEP). Teniendo en cuenta que no se mostraba la información actualizada sobre los indicadores de salud, surge la necesidad de informatizar dichos indicadores lo cual facilita la toma de decisiones.

Debido a la grave situación en el Sistema Penitenciario de Venezuela y dentro del marco del convenio de colaboración Cuba-Venezuela, donde se le da la tarea a la Universidad de las Ciencias Informáticas (UCI) de realizar el Sistema de Gestión Penitenciaria (SIGEP). A raíz de esto es que surge la necesidad de informatizar los indicadores de salud de los reclusos lo cual facilita la toma de decisiones.

El presente trabajo muestra la solución que dieron los autores a partir de las actividades realizadas por ellos como integrantes del SIGEP. En él se muestra el análisis, diseño e implementación de la solución brindada para el módulo Salud Integral para la ampliación de la Sala Situacional, a partir de los requisitos definidos con el cliente y el estudio de la arquitectura definida para el proyecto.

Para darle cumplimiento a las funcionalidades para diseñar el módulo Salud Integral se utilizaron diagramas de clases del diseño de cada una de las capas que componen la arquitectura. En la implementación se utilizaron herramientas y tecnologías de la plataforma Java que exponen tendencias de la programación web y que en su mayoría son libres y de código abierto.

Con el desarrollo del módulo de reportes de los indicadores de salud en los centros penitenciarios en la República Bolivariana de Venezuela en el régimen Intramuros, se obtuvo como resultado la integración de este módulo al subsistema Sala Situacional.

PALABRAS CLAVE

SIGEP, Salud Integral, Sala Situacional, Indicadores, Toma de decisiones, Análisis, Diseño, Implementación.

Índice

AGRADECIMIENTOS	I
DEDICATORIA.....	I
RESUMEN	I
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Conceptos asociados al dominio del problema.	6
Sistema Penitenciario	6
1.2 Reportes basados en la arquitectura del SIGEP	7
1.3 Sistemas penitenciarios y soluciones informáticas.....	8
1.3.1 Sistema Automatizado para el Control del Recluso (SACORE) Cuba.....	9
1.3.2 Sistema de Gestión Penitenciaria de Ecuador (SIGPEN).....	9
1.3.3 i3j Gestión para Centros Penitenciarios.....	10
1.3.4 Sistema de Gestión Penitenciaria (SIGEP)	10
1.4 Limitaciones de las soluciones informáticas	10
1.5 Procesos de ayuda a la toma de decisiones en el sistema penitenciario venezolano	11
1.6 Tecnologías y herramientas utilizadas en el desarrollo	12
1.6.1 Lenguajes de programación.....	12
1.6.2 Metodología de desarrollo	12
1.6.3 Herramienta de modelado	15
1.6.4 Plataforma de desarrollo.....	16
1.6.5 Apache Tomcat	17

1.7 Frameworks	17
1.7.1 Framework Spring v2.0.....	18
1.7.2 Framework Hibernatev3.2.0 cr4	20
1.8 Gestor de Base de Datos.....	21
1.9 Entorno integrado de desarrollo (IDE)	22
1.10 API (Interfaz de Programación de Aplicaciones)	22
1.11 Conclusiones Parciales.....	23
CAPÍTULO 2: ANÁLISIS Y DISEÑO	24
2.1 Indicadores.....	24
2.2 Análisis.....	25
2.3 Análisis de la Arquitectura del SIGEP.....	25
2.3.1 Visión general de la arquitectura.....	25
2.3.2 Definición de Subsistemas y Módulos	29
2.4 Características generales del módulo.....	30
2.4 Descripción del sistema propuesto	32
2.4.1 Requerimientos funcionales	32
2.4.2 Requerimientos no funcionales	35
2.5 Descripción del sistema propuesto	36
2.5.1 Concepción general del sistema	36
2.5.2 Modelo de casos de uso del sistema.....	37
2.5.3 Mapas de navegación	37
2.5.4 Descripción de casos de uso del sistema.....	38
2.6 Diseño.....	47

2.6.1 Patrones de diseño.....	47
2.7 Diagrama de clases.....	50
2.7.1 Descripción de los diagramas de clases del diseño.....	50
2.8 Modelo de Datos	51
2.9 Conclusiones Parciales.....	51
CAPÍTULO 3: IMPLEMENTACIÓN.....	52
3.1 Implementación de las entidades del dominio	52
3.2 Implementación de las interfaces de los managers	52
3.3 Implementación de la interfaz de la fachada.....	52
3.4 Implementación de la capa de acceso a datos.....	53
3.5 Implementación de los controladores.....	53
3.6 Implementación de las páginas JSP.....	53
3.7 Implementación de la lógica en el cliente	54
3.8 Modelo de despliegue.....	54
3.9 Descripción del Modelo de Componentes	55
3.10 Conclusiones Parciales.....	55
CAPÍTULO 4: PRUEBAS	57
4.1 Pruebas de software.....	57
4.2 Herramientas para automatizar las pruebas.....	57
4.3 Niveles de Pruebas.....	58
4.4 Técnicas de pruebas	58
EL SUMMARYREPORT OFRECE EL SIGUIENTE ASPECTO:	¡ERROR! MARCADOR NO DEFINIDO.
4.5 Conclusiones Parciales.....	62

CONCLUSIONES	64
RECOMENDACIONES.....	65
BIBLIOGRAFÍA	66
ANEXO 1.....	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 2.....	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 3.....	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 4.....	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 5.....	¡ERROR! MARCADOR NO DEFINIDO.

Índice de figuras

Figura 1: Subsistema del SIGEP.....	12
Figura 2: Fases y flujos de trabajo de RUP.....	14
Figura 3: Elementos de UML	15
Figura 4: Estructura de Spring.....	19
Figura 5: Arquitectura de Hibernate.....	21
Figura 6: Arquitectura del SIGEP	26
Figura 7: Estructura de capas del sistema	27
Figura 8: Subsistemas del SIGEP.....	29
Figura 9: Estructura de paquetes de los módulos de Información Operativa e Histórica	30
Figura 10: Caso de uso del sistema	37
Figura 11: Mapa de Navegación de Hábitos Psicobiológicos.....	38

Figura 12: Hábitos Psicobiológicos\Información Operativa\Información tabular	44
Figura 13: Hábitos Psicobiológicos\Información Operativa\Información gráfica.....	45
Figura 14: Hábitos Psicobiológicos\Información Operativa>Listados.....	45
Figura 15: Hábitos Psicobiológicos\Información Histórica \ Información tabular.....	46
Figura 16: Hábitos Psicobiológicos\Información Histórica \ Información gráfica	46
Figura 17: Hábitos Psicobiológicos\Información Histórica \ Configuración	47
Figura 18: Modelo Vista Controlador.....	49
Figura 19: Modelo de despliegue para el régimen Intramuros.....	54
Figura 20: Resumen de las no conformidades del módulo Salud Integral del subsistema Sala Situacional	59
Figura 21: Acceso al usuario ssOperativa a la Información Operativa	60
Figura 22: El usuario ssOperativa no tiene acceso a la Información Histórica.....	¡Error! Marcador no definido.
Figura 23: Acceso al usuario ssHistorico a la Información Histórica.....	¡Error! Marcador no definido.
Figura 24: El usuario ssHistorico no tiene acceso a la Información Operativa.....	¡Error! Marcador no definido.
Figura 25: Gráfica de la simulación de 30 usuarios con un período de 30 segundos	¡Error! Marcador no definido.
Figura 26: Informe resumen para 30 usuarios para un período de 30 segundos de subida ...	¡Error! Marcador no definido.
Figura 27: Informe resumen para 60 usuarios para un período de subida de 180 segundos.	¡Error! Marcador no definido.
Figura 28: Mapa de Navegación de Alergias	¡Error! Marcador no definido.
Figura 29: Mapa de Navegación de Evaluación Nutricional	¡Error! Marcador no definido.
Figura 30: Mapa de Navegación de Medicamentos.....	¡Error! Marcador no definido.
Figura 31: Mapa de Navegación de Patologías de Interés.....	¡Error! Marcador no definido.
Figura 32: Mapa de Navegación de Vacunación.....	¡Error! Marcador no definido.

Figura 33: Alergia\Información Operativa\tabular	¡Error! Marcador no definido.
Figura 34: Alergias\Información Histórica\Configuración	¡Error! Marcador no definido.
Figura 35: Evaluación Nutricionales\Información Operativa>Listados	¡Error! Marcador no definido.
Figura 36: Evaluación Nutricionales\Información Histórica\gráfica.....	¡Error! Marcador no definido.
Figura 37: Medicamentos\Información Operativa\tabular	¡Error! Marcador no definido.
Figura 38: Medicamentos\Información Histórica\Configuración.....	¡Error! Marcador no definido.
Figura 39: Patologías de Interés\Información Operativa>Listados	¡Error! Marcador no definido.
Figura 40: Patologías de Interés\Información Histórica\gráfica.....	¡Error! Marcador no definido.
Figura 41: Vacunación\Información Operativa\gráfic	¡Error! Marcador no definido.
Figura 42: Vacunación\Información Histórica\tabular	¡Error! Marcador no definido.
Figura 43: Diagrama de clases del diseño de Alergias	¡Error! Marcador no definido.
Figura 44: Diagrama de clases del diseño de Evaluación Nutricional	¡Error! Marcador no definido.
Figura 45: Diagrama de clases del diseño de Medicamentos.....	¡Error! Marcador no definido.
Figura 46: Diagrama de clases del diseño de Patologías de Interés	¡Error! Marcador no definido.
Figura 47: Diagrama de clases del diseño de Vacunación	¡Error! Marcador no definido.
Figura 48: Diagrama de clases del diseño de Hábitos Psicobiológicos.....	¡Error! Marcador no definido.
Figura 49: Modelo de componentes de Alergias	¡Error! Marcador no definido.
Figura 50: Modelo de componentes de Evaluación Nutricional	¡Error! Marcador no definido.
Figura 51: Modelo de componentes de Medicamentos.....	¡Error! Marcador no definido.
Figura 52: Modelo de componentes de Patologías de Interés	¡Error! Marcador no definido.
Figura 53: Modelo de componentes de Vacunación.....	¡Error! Marcador no definido.
Figura 54: Modelo de componentes de Hábitos Psicobiológicos.....	¡Error! Marcador no definido.

Figura 55: Diagrama entidad relación del módulo Salud Integral para el subsistema Sala situacional..**¡Error! Marcador no definido.**

Índice de tablas

Tabla 1: Actor del sistema	37
Tabla 2: Descripción del caso de uso Hábitos Psicobiológicos	38
Tabla 3: Descripción del casos de uso Generar Reporte de Alergias.....	¡Error! Marcador no definido.
Tabla 4: Descripción del casos de uso Generar Reporte de Evaluación Nutricional.....	¡Error! Marcador no definido.
Tabla 5: Descripción del casos de uso Generar Reporte de Medicamentos	¡Error! Marcador no definido.
Tabla 6: Descripción del casos de uso Generar Reporte de Patologías de Interés	¡Error! Marcador no definido.
Tabla 7: Descripción del casos de uso Generar Reporte de Vacunación	¡Error! Marcador no definido.

Introducción

En diciembre del año 2004, el Presidente de la República Bolivariana de Venezuela decretó la Emergencia Penitenciaria. A raíz de este suceso se celebraron reuniones de coordinación en mayo del 2005 entre la dirección del Ministerio Interior y Justicia (MIJ), la Dirección General de Custodia y Rehabilitación del Recluso (DGCRR) y la parte cubana. También se realizó un Diagnóstico integral al sistema penitenciario; lo cual permitió establecer un convenio de colaboración entre instituciones cubanas y venezolanas, en este contexto surge el Proyecto de Humanización Penitenciaria.

En agosto de 2005 se inicia la colaboración médica odontológica por diferentes centros penitenciarios, tiempo después se les presenta a los directores de penales la propuesta de solución tecnológica y es cuando nace la solución de software Sistema de Gestión Penitenciaria (SIGEP); alternativa para resolver el número elevado de deficiencias en infraestructura y los problemas identificados en el análisis de los datos recogidos en el 2005 en el censo nacional de la situación judicial de la población penitenciaria que presenta el Sistema Penitenciario Venezolano.

Actualmente SIGEP se desarrolla en la Universidad de las Ciencias Informáticas (UCI) creada con el objetivo de producir software a partir de la vinculación estudio-trabajo. La misma cuenta con varios centros de desarrollo de software entre los que se encuentra Informatización de la Seguridad Ciudadana (ISEC), al que pertenece dicho proyecto, dividido en varios subsistemas. Entre estos subsistemas se encuentra el de Salud Integral que gestiona las historias clínicas de los privados de libertad con el objetivo de controlar la información referente a las enfermedades transmisibles y crónicas, estados de salud, patologías, antecedentes quirúrgicos, hábitos psicobiológicos, evaluación nutricional, inmunizaciones, enfermedades de los familiares y las enfermedades de la infancia de la población penitenciaria.

Sin embargo, todas estas funcionalidades no son suficientes. La Dirección Nacional de Servicios Penitenciarios (DNSP) necesita información operativa para analizar situaciones de riesgo o realizar estudios de comportamiento de determinado fenómeno, en este caso específicamente en el área de atención médica. La información operativa se lleva a través de los reportes centralizados que se obtienen vía correo electrónico, teléfono o valija, desde las diferentes dependencias. Esto implica que la

información que se tiene en la Dirección en este sentido no sea la más exacta ni la más completa en muchos casos.

En consecuencia, existe un desconocimiento casi total de la situación operativa de los establecimientos penitenciarios que obstaculiza la estrategia de la DNSP en función de lograr un sistema justo, pues no se cuenta con un mecanismo de actualización de la información que permita establecer estrategias reales, resulta casi imposible realizar una valoración con exactitud del estado de salud de los individuos, imposibilitando tomar medidas y dedicar recursos para mitigar los riesgos de propagación de epidemias y otras enfermedades, no se pueden tomar decisiones en función de la realidad objetiva de cada establecimiento.

Partiendo de lo expuesto anteriormente, el **problema a resolver** queda formulado de la siguiente manera: En el sistema que existe actualmente en los centros penales internos para procesar los indicadores de salud de los reclusos no contiene toda la información necesaria que facilite la toma de decisiones en los establecimientos penitenciarios.

En correspondencia con el problema a resolver se define como **objeto de estudio**: Los indicadores de salud que apoyan la toma de decisiones en los establecimientos penitenciarios.

Dadas las condiciones descritas en el problema a resolver planteado se define como **objetivo general**: Desarrollar el módulo de reportes de los indicadores de salud en los centros penales internos para la ampliación del subsistema Sala Situacional del SIGEP que facilite la toma de decisiones.

Por lo que se especifica el siguiente **campo de acción**: Los indicadores de salud en los centros penales internos.

En función a darle cumplimiento al objetivo general se definen los siguientes **objetivos específicos**:

- Elaborar el marcoteórico de la investigación.
- Analizar los mecanismos que se utilizarán para brindar información operativa e histórica, en específico información vinculada al módulo Salud Integral del subsistema Sala Situacional.

- Diseñar los mecanismos que se utilizarán para brindar información operativa e histórica, en específico información vinculada al módulo Salud Integral del subsistema Sala Situacional.
- Implementar la propuesta de solución de software.
- Realizar las pruebas al módulo Salud Integral del subsistema Sala Situacional.
- Integrar el módulo Salud Integral del subsistema Sala Situacional para el régimen Intramuros del SIGEP.

Se plantea la siguiente **idea a defender**: El desarrollo del módulo de reportes de los indicadores de salud en los centros penales internos para la ampliación del subsistema Sala Situacional del SIGEP facilita la toma de decisiones.

Las **tareas de la investigación** son:

- Estudio de la bibliografía referente al tema.
- Redacción del marco teórico de la investigación.
- Definición de los requisitos funcionales y no funcionales correspondientes al módulo Salud Integral.
- Análisis de los mecanismos que se utilizarán para brindar información operativa e histórica, en específico información vinculada al módulo Salud Integral del subsistema Sala Situacional.
- Diseño de los mecanismos que se utilizarán para brindar información operativa e histórica, en específico información vinculada al módulo Salud Integral del subsistema Sala Situacional.
- Implementación del diseño realizado al módulo Salud Integral del subsistema Sala Situacional.
- Realización de las pruebas al módulo.
- Integración del módulo Salud Integral del subsistema Sala Situacional al SIGEP.

En la presente investigación se utilizan los métodos científicos que permiten valorar el subsistema Salud Integral, con el propósito de ver su funcionamiento y obtener las buenas prácticas que ayuden a desarrollar la investigación.

Métodos Teóricos

El **analítico- sintético**, se utilizó para realizar un análisis de la documentación, para sintetizar y obtener los elementos más importantes de la información contenidos en toda la documentación referente al tema y de esta obtener lo esencial para enriquecer la investigación.

El **inductivo- deductivo**, se utilizó para identificar las herramientas, metodologías y técnicas a utilizar para el desarrollo del trabajo.

También se utiliza el **análisis histórico- lógico**, pues se analizó la información referente al subsistema Salud Integral, desde el punto de vista de su evolución y su funcionamiento.

Unos de los métodos teóricos que se utiliza es la **modelación**, el cual se evidencia en la fase de diseño donde se modelan los diagramas de clases del diseño y el modelo de datos.

Por lo anteriormente planteado la ampliación de la Sala Situacional le proporcionará a la DNSP, información operativa e histórica referente a la salud de los individuos como apoyo ala toma de decisiones.

El presente trabajo está estructurado en 4 capítulos que abarca toda la investigación realizada:

Capítulo 1: “Fundamentación teórica”.

Este capítulo contiene la fundamentación teórica del presente trabajo de diploma. Se definen conceptos básicos y se describen las principales características de las herramientas. Además se especifica las tecnologías empleadas durante el diseño e implementación de los procesos relacionados con el subsistema Salud Integral del Sistema de Gestión Penitenciaria de la República Bolivariana de Venezuela.

Capítulo 2: “Análisis y diseño”.

El capítulo Análisis y diseño constituye el resultado del estudio de los procesos que tienen lugar para el análisis y diseño de la información en los centros penales del régimen Intramuros. A partir de este estudio se definen los requisitos funcionales y no funcionales que deberá satisfacer la propuesta de solución. Por otra parte la descripción de los elementos principales del análisis y diseño a través de artefactos obtenidos como parte del proceso de desarrollo del subsistema Salud Integral del Sistema de Gestión Penitenciaria de la República Bolivariana de Venezuela.

Capítulo 3: “Implementación”.

El capítulo muestra la solución de la implementación para el subsistema Salud Integral para la ampliación de la Sala Situacional del Sistema Penitenciario de la República Bolivariana de Venezuela a través fragmentos de código y sus respectivas descripciones y el diagrama de componente, teniendo en cuenta las actividades de implementación definidas para el SIGEP.

Capítulo 4: “Pruebas”.

El presente capítulo se describe como se prueban los componentes bajo unas condiciones o requerimientos especificados. Se realizan pruebas para validar y verificar la calidad de la solución propuesta.

Capítulo 1: Fundamentación Teórica

Introducción

En el presente capítulo se realiza un estudio valorativo sobre los sistemas informáticos existentes que soportan los procesos de ayuda a la toma de decisiones en los centros penales, tanto en el ámbito nacional como internacional. Se estudian las tendencias, tecnologías y metodologías usadas para el desarrollo del SIGEP como parte de la propuesta de solución. Se definen conceptos asociados al dominio del problema que serán de utilidad en la comprensión del mismo.

1.1 Conceptos asociados al dominio del problema.

Sistema Penitenciario

Un sistema penitenciario se define como el conjunto de normas, procedimientos y dependencias dispuestas por el Estado para la ejecución del régimen penitenciario entre los que se encuentran además los principios, programas, recursos humanos, dependencias e infraestructura que se encuentran relacionadas y destinadas a este régimen.(1)

Cada gobierno define la estructura de su sistema penitenciario de acuerdo a la legislación y las condiciones reales que posee. En el caso específico de la República Bolivariana de Venezuela, tal sistema está constituido por la legislación vigente, los métodos que se emplearán para lograr su funcionamiento, las diferentes dependencias encargadas de su aplicación, los equipos de trabajo y la infraestructura carcelaria.(1)

Toma de decisiones

Se conoce como toma de decisiones al proceso que consiste en realizar una elección entre diversas alternativas, después de analizar un conjunto de indicadores lo cual contribuirá a la toma de medidas en caso de ocurrir una alteración en algún indicador. En el módulo Salud Integral del subsistema Sala Situacional, está basada en la forma en que los reportes muestran la información. Estos datos que se obtienen le permite al funcionario de la Sala Situacional hacer un análisis del comportamiento de cada uno

Capítulo 1: *Fundamentación Teórica*

de los parámetros de salud, que van a tener en cuenta para a través de su estado dedicar recursos y tomar medidas para mitigar los riesgos de expansión de epidemias y demás enfermedades que puedan surgir.

Indicadores

Los indicadores son variables declaradas de las cuales se tendrá un constante monitoreo para conocer cómo se comporta su evolución en el tiempo. Son elementales para evaluar, dar seguimiento y predecir tendencias de la situación de un país, región o entidad. Permite evaluar, en términos cuantitativos, la importancia de un objeto o fenómeno comparándolo con otro de la misma especie.

En la salud los indicadores son aquellos que van a tener una evaluación del comportamiento en el tiempo, para así tener conocimiento del estado de la población penal en este campo. En la Sala Situacional van a estar dado por dos tipos de información definidas por un intervalo de tiempo, dándole a conocer al funcionario la información necesaria para la ayuda a la toma de decisiones.

Sala situacional

Una Sala Situacional es un Sistema Tecno-gerencial bajo un conjunto de normas y procedimientos claramente establecidos, en el marco de un concepto operacional integral, adaptables a situaciones rutinarias y a momentos de crisis, además que se realizan estudios comparativos de eventos pasados. Es considerado como un soporte a la toma de decisiones.

En el módulo Salud Integral esta va a ser la encargada de mostrar una visión global de la situación de salud de los reclusos en los centros penitenciarios. Muestra un conjunto de reportes basado en indicadores que le facilitaran a la DNSP dedicar recursos para mitigar los riesgos en situaciones excepcionales como epidemias y demás enfermedades.

1.2 Reportes basados en la arquitectura del SIGEP

Los reportes tienen como propósito informar el estado o condiciones, también sirven para recomendar, dar sugerencias o conclusiones. En el caso del módulo Salud Integral del subsistema Sala Situacional los reportes se generan con el objetivo de obtener información actualizada de cada uno de los indicadores.

Capítulo 1: *Fundamentación Teórica*

Existen distintos tipos de reportes como el técnico, informal, financieros, de tráfico, básicos, tablero de comando, entre otros. En la primera versión del SIGEP no se tuvo en cuenta un generador de reportes dinámico porque la solución inicial que se le brindó al cliente solo tenía que mostrar un número determinado de indicadores. Cumpliendo con la arquitectura propuesta por el proyecto se comienza a hacer uso de la herramienta JasperReport para la generación de dichos reportes.

Después de firmado el contrato con el cliente, el equipo de desarrollo se da cuenta de la necesidad de registrar nuevos indicadores en el sistema, lo que imposibilitará la búsqueda de una solución fácil a raíz de la arquitectura planteada. Los reportes que se tenían hasta el momento solo brindaban la posibilidad de consultarlos y no se podían gestionar dinámicamente. Para aplicarlos sería necesario emplear una tecnología que permitiera realizar reportes dinámicos, lo cual implicaría un enorme costo al proyecto. Se propuso como posible solución la utilización del Pentaho, que es un conjunto de herramientas de código abierto, con mayor cantidad de funcionalidades, de las más fáciles de trabajar, además de que está escrita en Java y tiene un ambiente de implementación basado en Java. Añadirle esto al proyecto traería como consecuencia volver a negociar el costo con el cliente, lo que no se consideró una vía factible. Por esta razón los reportes que se dan en la ampliación de la Sala Situacional se muestran a partir de lo que está desarrollado anteriormente, brindando solo información.

En la Universidad de las Ciencias Informáticas existe actualmente un proyecto que se dedica a realizar un sistema web para gestionar los reportes dinámicos, pero están desarrollándolo en PHP y PostgreSQL lo que significa que aún no es compatible con el SIGEP. Hace un tiempo comenzaron a migrar a Java con Gestor de Base Datos Oracle, cuando esto se logre se propondrá para futuras versiones del proyecto.

1.3 Sistemas penitenciarios y soluciones informáticas

Con el objetivo de mejorar los centros penitenciarios, varios países han desarrollado campañas de modernización con el propósito de promover el uso de soluciones informáticas para la gestión de los procesos y el apoyo a la toma de decisiones en estos centros. Una de las características de los sistemas penitenciarios de América Latina es el precario funcionamiento de las instituciones y la estandarización de los procesos penitenciarios.

Capítulo 1: Fundamentación Teórica

A continuación se realiza un estudio de las principales características de algunos sistemas informáticos implantados en centros penitenciarios como punto de partida para el desarrollo de este trabajo.

1.3.1 Sistema Automatizado para el Control del Recluso (SACORE) Cuba

El SACORE surge para dar cumplimiento a la Orden 43/99 del Vice Ministro Primero del Ministerio del Interior de Cuba. Es una aplicación web que se caracteriza por: garantizar respuestas inmediatas a las solicitudes de información de los diferentes órganos e instituciones del estado, recoger prácticamente la totalidad de la información de los reclusos en todas las especialidades, tener más de 200 reportes impresos, permitir la recuperación dinámica a partir de una solicitud de búsqueda, emitir partes de forma automatizada y permitir el traslado automático de todos los datos del recluso al nivel nacional.

El sistema contiene un módulo de reportes que permite la actualización constante del estado de los indicadores. La forma de mostrar la información es tabular y de listado, no se puede tener una vista gráfica de los indicadores y por tanto una vista rápida de su estado para tomar una decisión correcta en un momento de excepción.

1.3.2 Sistema de Gestión Penitenciaria de Ecuador (SIGPEN)

El SIGPEN es un sistema web centralizado perteneciente a Ecuador, por lo que la información que se maneja sobre los reclusos está constantemente actualizada. Este gestiona la información referente al interno, así como los datos personales, la situación jurídica, el expediente médico y laboral, los movimientos y la peligrosidad.

Está compuesto por diferentes secciones (Interno, Departamento Jurídico, Departamento Médico, entre otras) las cuales brindan toda la información referente al estado de los reclusos en la sección seleccionada.

SIGPEN permite recopilar y controlar la información operativa que se genera en un Centro de Rehabilitación Social. Cuenta con los siguientes tipos de reportes:

- Estadísticas
- Ingreso y egreso de internos

Capítulo 1: Fundamentación Teórica

- Internos por celda
- Resultados del censo

1.3.3 i3j Gestión para Centros Penitenciarios

El proyecto i3j fue creado por una empresa de consultoría y desarrollo de software nombrada NEOTEC de Bolivia, es una solución de gestión de centros penitenciarios. La información de los penados está disponible a las instituciones judiciales a las cuales es posible otorgar acceso. Dicha información incluye el registro de los reclusos, movimientos, visitas, compañeros de celda, entre otros. El sistema provee servicios que gestionan el tratamiento de los internos facilitando el control del estudio y el trabajo de los reclusos.

1.3.4 Sistema de Gestión Penitenciaria (SIGEP)

El sistema nació de un convenio de amistad entre los países hermanos de Venezuela y Cuba con el objetivo de lograr un sistema informático integral que contribuya al control operativo, administrativo y estratégico del sistema penitenciario. Además garantiza el respeto a los derechos de los individuos, su actividad de rehabilitación y reinserción a la sociedad. Por la situación existente surge la necesidad de integrar al sistema un subsistema Sala Situacional, la cual tiene como objetivo mejorar la toma de decisiones estratégicas por parte de los directivos y funcionarios del Sistema Penitenciario Venezolano. Actualmente en el SIGEP se maneja toda la información referente a los internos, así como los datos personales, presentación ante tribunales, el expediente médico y laboral, los movimientos (salidas, traslados y visitas), medidas disciplinarias, entre otras.

1.4 Limitaciones de las soluciones informáticas

Los sistemas informáticos del gobierno de Cuba, Ecuador y el producto Boliviano i3j tienen como principal limitante la falta de conectividad entre los centros penitenciarios que imposibilita la instalación de sistemas centralizados que garanticen la integridad e inmediatez de la información. En el caso del SACORE no cuenta con una Sala Situacional, por lo que dificulta la toma de decisiones, pero contiene más de 200 salidas a nivel de tabla y no tiene gráficos. Sin embargo el SIGPEN contiene un Departamento Médico

Capítulo 1: Fundamentación Teórica

pero no tiene centralizado los registros de salud de cada uno de los reclusos. Por otra parte el SIGEP carece de reportes que indiquen el estado de salud de los internos ya sea de carácter operativo o histórico. Estos sistemas informáticos no son portables a otros sistemas penitenciarios debido a la legislación vigente de los países para los cuales fueron definidos.

1.5 Procesos de ayuda a la toma de decisiones en el sistema penitenciario venezolano

En los sistemas penitenciarios contar con una sala situacional es fundamental, pues esta constituye la fuente de los procesos que apoyan a la toma de decisiones. Es un espacio destinado a recibir continuamente las principales informaciones para luego realizar los análisis correspondientes y dictar las decisiones por las cuales se regirá la entidad. Esta debe contar con un sistema informativo que apoya la recepción de la información y en consecuencia a la tomar medidas oportunas y prácticas.

Para monitorear y controlar la toma de decisiones, funcionan en Venezuela diferentes Salas Situacionales. Con el objetivo de recibir informaciones diarias y semanales de los equipos regionales en materia comunal, se estableció una sala en el puesto de mando de los consejos comunales. Para la evaluación de la salud integral de la población en diferentes regiones también funcionan las salas situacionales, brindando información constante sobre el comportamiento de los principales indicadores de salud.

El sistema penitenciario venezolano actual comparte las características básicas de los sistemas penitenciarios latinoamericanos. El SIGEP se encuentra en su versión 2.1 (Ver la figura 1), el cual tiene integrado una Sala Situacional que se convierte en una herramienta para el monitoreo de los distintos indicadores como son: Ingreso-Egresos, Cultura, Deporte, entre otros; pero aún no se cuenta con toda la información de los indicadores de salud. Por lo que se desconoce el estado actual y los factores más importantes que influyen en el deterioro de la salud de la población penal, imposibilitando tomar medidas en caso de epidemias o situaciones de riesgo.



Figura 1: Subsistema del SIGEP

1.6 Tecnologías y herramientas utilizadas en el desarrollo

1.6.1 Lenguajes de programación

Los lenguajes de programación son un conjunto de programas que controlan el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Están formados por un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura, el significado de sus elementos y expresiones.

Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.(3)

El lenguaje de programación Java presenta características muy favorables para aplicaciones de gran envergadura, por lo que fue seleccionado para el desarrollo de SIGEP.

1.6.2 Metodología de desarrollo

Una metodología de desarrollo es una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. La finalidad de una

Capítulo 1: *Fundamentación Teórica*

metodología de desarrollo es garantizar la eficacia (cumplir los requisitos iniciales) y la eficiencia (Ejemplo: minimizar las pérdidas de tiempo) en el proceso de generación de software. (8)

Proceso Unificado de Desarrollo (RUP)

Como metodología de desarrollo fue seleccionada RUP, ya que es una de las más generales que existen en la actualidad, debido a que está pensada para adaptarse a cualquier proyecto, principalmente con grandes grupos de producción. En el proyecto SIGEP el grupo de trabajo está conformado por estudiantes y profesores en su mayoría, los cuales podrían pasar a cumplir otras funciones en cualquier momento de su vida, esto trae como consecuencia que el equipo de desarrollo sea inestable, por esto se necesita una buena organización y abundante documentación para que el avance del proyecto no se vea afectado y se asegure la continuidad del mismo, por otra parte el cliente final es extranjero y no forma parte del equipo de desarrollo. Los casos de uso reflejan lo que los usuarios requieren y desean, por lo que guían todo el proceso de desarrollo. La arquitectura del software juega un papel fundamental y muestra una visión común del sistema, en la que el equipo de proyecto y los usuarios deben estar de acuerdo.

En la figura 2 se muestran las diferentes fases, flujos de trabajos e iteraciones definidos por RUP. Los primeros seis son flujos de ingeniería y los restantes apoyan al desarrollo del proyecto. Su objetivo es obtener resultados de alta calidad y que cumpla con las necesidades de los clientes y los usuarios finales.

Capítulo 1: Fundamentación Teórica

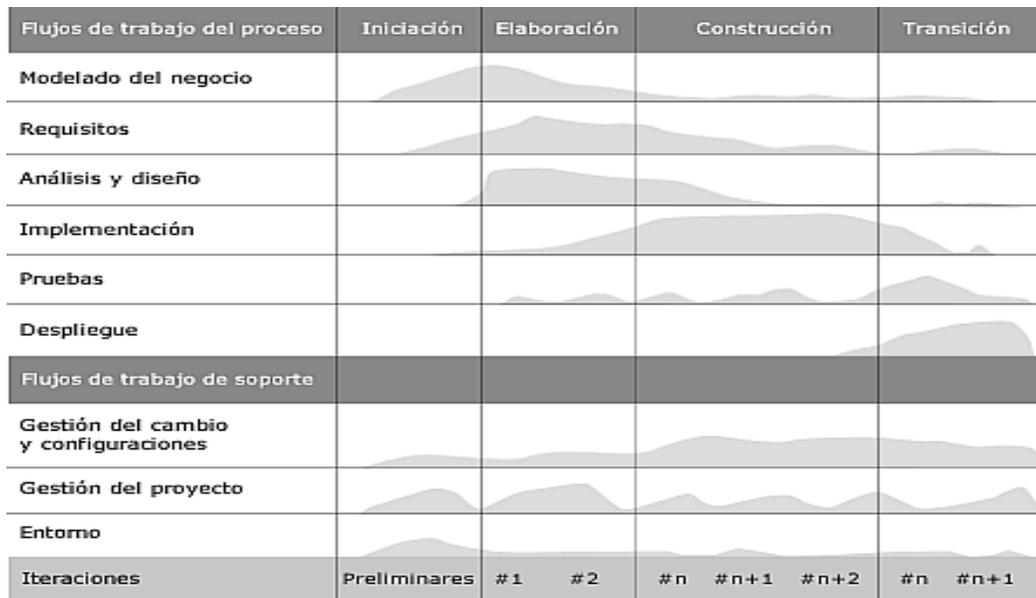


Figura 2: Fases y flujos de trabajo de RUP

Lenguaje de modelado

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es uno de los lenguajes para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, es útil categorizarlos jerárquicamente. A continuación en la figura 3 se muestran los elementos de UML.

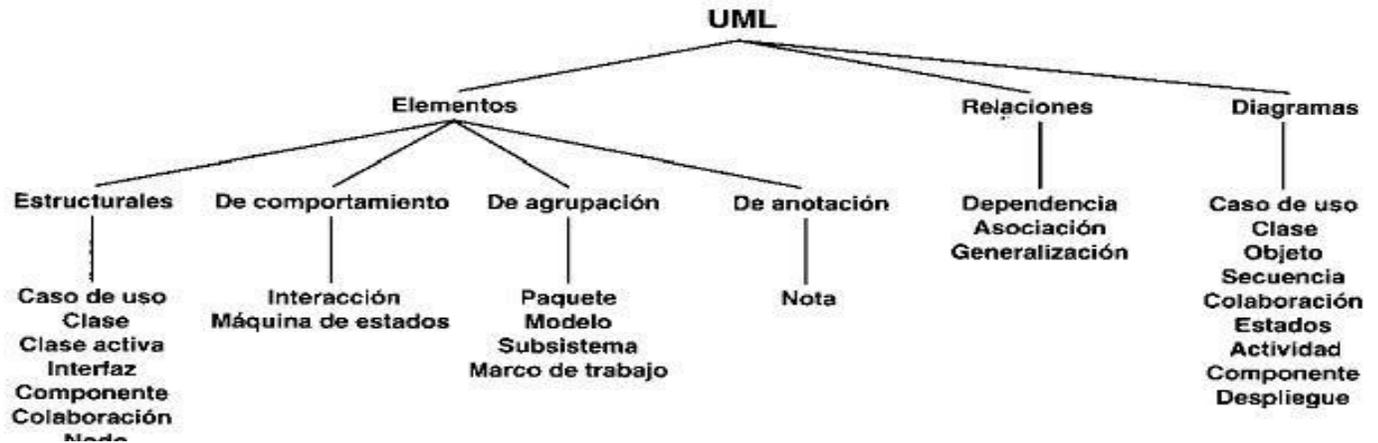


Figura 3: Elementos de UML

1.6.3 Herramienta de modelado

Visual Paradigm:

Visual Paradigm en su versión 3.4, es una herramienta CASE profesional que soporta todo el ciclo de vida del software: análisis y diseño, construcción, prueba y despliegue. Dicha herramienta ayuda a una rápida construcción de la aplicación con alta calidad y a un menor costo. Permite modelar todos los tipos de diagrama de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta también proporciona abundantes tutoriales, demostraciones y proyectos UML.

Como característica atractiva para los desarrolladores, presenta una interfaz de uso intuitiva y con muchas facilidades a la hora de modelar los diagramas que soportan la Ingeniería de Requerimientos.

La herramienta está diseñada para una amplia gama de usuarios, incluidos los ingenieros de software, analistas de sistemas, analistas de negocio o para cualquiera que esté interesado en la construcción de forma fiable a gran escala de sistemas de software, utilizando un enfoque orientado a objetos. Por estas ventajas es seleccionado Visual Paradigm como herramienta CASE para la ampliación de la Sala Situacional.

1.6.4 Plataforma de desarrollo

J2EE

La plataforma J2EE (Java 2 Enterprise Edition) es un estándar para el desarrollo de aplicaciones empresariales utilizando el lenguaje de programación Java. Facilita el desarrollo de aplicaciones basándose en pruebas estandarizadas y componentes modulares. Ofrece un conjunto completo de servicios a esos componentes y maneja muchos detalles de comportamiento de la aplicación automáticamente, sin programación compleja. Dentro de las varias especificaciones de APIs que incluye, las que más aportan a la solución SIGEP son: Servlets y Java Server Pages. (3)

Se decidió seleccionar la plataforma Java 2 Platform Enterprise Edition (J2EE) con el fin de garantizar una solución informática multiplataforma, de alto rendimiento, escalabilidad y seguridad.

Java-servlets:

Un servlet es una clase del lenguaje de programación Java utilizada para ampliar las capacidades de los servidores que alojan aplicaciones accedidas mediante el modelo de programación petición-respuesta. Realizan la función de capa intermedia entre una petición proveniente de un navegador Web u otro cliente HTTP4, y las aplicaciones del servidor.

Pueden utilizar toda la familia de APIs Java y recibir todos los beneficios del lenguaje, como la portabilidad, el rendimiento y la reutilización. Su función principal es proveer páginas web dinámicas y personalizadas, utilizando para este objetivo el acceso a bases de datos, flujos de trabajo y otros recursos.(3)

Java Server Pages:

Como parte de la familia de la tecnología Java, la tecnología JSP permite el desarrollo rápido de aplicaciones basadas en Web que son independientes de la plataforma. La tecnología JSP separa la interfaz de usuario de la generación de contenidos, permitiendo a los diseñadores cambiar el diseño de la página en general, sin alterar el contenido dinámico subyacente. (3)

1.6.5 Apache Tomcat

Es un contenedor de Servlet usado en la implementación de referencia oficial para las tecnologías Java Servlet y Java Server Pages (JSP). Es desarrollado en un ambiente participativo y abierto. La versión 5.5.17 es implementada a partir de las especificaciones Servlet 2.4 y JSP 2.0 y cuenta con un mecanismo de recolección de basura perfeccionado, basado en su reducción. Posee una capa envolvente nativa para Windows y Unix para la integración de las plataformas. Es usado en numerosas aplicaciones web de gran escala y críticas en una amplia gama de industrias y organizaciones.

Ventajas del servidor de aplicaciones Web Tomcat:

- Es gratis.
- Fácil de instalar.
- Ocupa muy poco espacio.
- Es bastante rápido.
- Es fiable.
- Su solidez se basa en que miles de desarrolladores contribuyen con código.

En el caso del SIGEP se selecciona como servidor de aplicaciones debido precisamente a que la plataforma de desarrollo que se utiliza es Java, y este es el que más se ajusta a las necesidades del proyecto.

1.7 Frameworks

Un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Capítulo 1: *Fundamentación Teórica*

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo, como el uso de patrones. Los frameworks son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional.

En el desarrollo de software, se ha convertido en práctica común la utilización de alguna clase de framework que facilite la integración de las tecnologías a utilizar con el software en construcción. Estos frameworks por lo general son un patrón arquitectónico que garantiza la reusabilidad de la arquitectura empleada para el diseño del proyecto. Permitiendo el ahorro de tiempo en función del diseño de la aplicación, y por lo general implementan las partes más trabajosas del problema a tratar, de este modo se disminuye el costo de tiempo de producción de la aplicación.

Sobre la plataforma J2EE existen un conjunto amplio de frameworks que utilizan e integran las APIs que esta brinda. En el desarrollo de la arquitectura base del SIGEP se utilizó el framework de aplicación Spring al igual que el framework de soporte Hibernate, por sus múltiples ventajas, en el desarrollo de aplicaciones web dentro de la plataforma J2EE.

1.7.1 Framework Spring v2.0

Es un framework de código abierto que implementa el patrón Factory y que proporciona un marco de trabajo para el desarrollo de aplicaciones Java EE. Su principal objetivo es ser una alternativa sencilla y fácil de usar frente a Enterprise JavaBean. Está basado en un conjunto de módulos que proporcionan todo lo necesario para desarrollar una aplicación empresarial aunque no es necesario basar la aplicación al completo en este framework, su modularidad hace que sea posible hacer uso de los módulos que requiera la aplicación e ignorar el resto.

Con Spring no se crea una instancia de un objeto para cada usuario del sistema, sino que se instancian una única vez al poner en marcha la aplicación, se guardan en la factoría y luego se comparten por todos

Capítulo 1: Fundamentación Teórica

los usuarios inyectándolos en los objetos que los necesiten mediante el uso de un método 'set' definido en el propio objeto.

Mientras que las características fundamentales pueden emplearse en cualquier aplicación desarrollada en Java, existen muchas extensiones y mejoras para construir aplicaciones basadas en web por encima de la plataforma empresarial de Java. En la figura 4 se muestra la estructura por módulos del framework spring.

Dentro de las ventajas que ofrece *Spring*, se encuentran:

- Facilita la manipulación de los objetos.
- Reduce la proliferación de *Singletons*.
- Elimina la necesidad de usar distintos y variados tipos de ficheros de configuración.
- Mejora la práctica de programación.

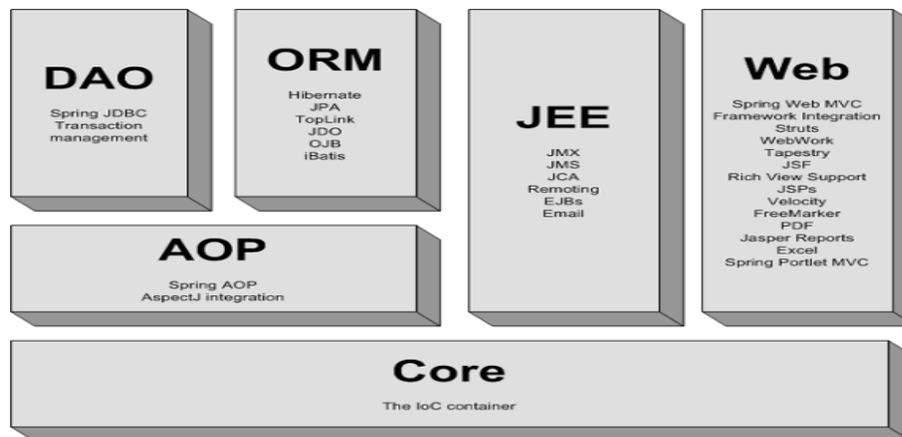


Figura 4: Estructura de Spring

Algunas de las principales características de Spring, que fuesen decisivas para su utilización en SIGEP, es el amplio soporte que este provee para la AOP (Programación orientada a aspectos). Lo que permite un desarrollo cohesivo de la lógica del negocio y los servicios del sistema, a la vez que las separa para lograr un bajo acoplamiento de las mismas. Esto permite que los objetos de la aplicación se centren en

Capítulo 1: Fundamentación Teórica

manejar solamente la lógica del negocio. Al ser un framework, hace posible componer aplicaciones complejas a partir del uso de componentes relativamente sencillos. Esto se logra en Spring, por lo general, a través del uso de XML para declarar los objetos de la aplicación, e inyectarles las dependencias, para que no tengan ellos que gestionarlas (4)

1.7.2 Framework Hibernatev3.2.0 cr4

Hibernate es un framework gratuito, de código abierto, el cual es una solución de mapeo de los objetos de negocio con la información almacenada en bases de datos relacionales. Se encarga del mapeo de clases de Java a tablas de la base de datos y la generación de consultas a la base de datos. Con este framework, se puede diseñar objetos persistentes que podrán incluir relaciones, colecciones, polimorfismo, y un gran número de tipos de datos.

De una manera muy rápida y optimizada se puede realizar consultas contra cualquiera de los entornos soportados: Oracle, DB2, MySql, entre otras. Otra ventaja que existe en la utilización de este frameworks, es el propio dialecto que propone Hibernate, Hibernate Query Lenguaje (HQL), para desarrollar las consultas. Hibernate elimina la necesidad de parte del código de acceso a los datos, permitiendo que el desarrollador se concentre más en la lógica de negocio. Esta reducción de código representa un aumento de la productividad y permite que la capa de acceso a datos sea más entendible y fácil de mantener. A continuación se muestra la figura 5 que representa la arquitectura de Hibernate.(5)

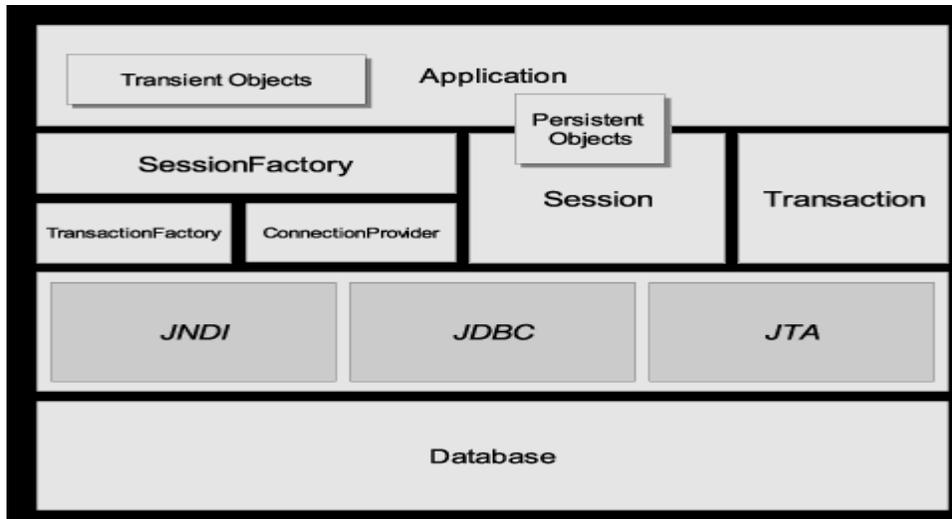


Figura 5: Arquitectura de Hibernate

1.8 Gestor de Base de Datos

Se utilizará como Sistema Gestor de Base Datos Oracle en su versión 10g, para el desarrollo del proyecto SIGEP. Oracle es considerado como uno de los sistemas de bases de datos más completos, destacando: soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma.

Oracle fue desarrollado por Oracle Corporation la cual es una de las mayores compañías de software del mundo. Sus productos van desde bases de datos hasta sistemas de gestión. Cuenta además, con herramientas propias de desarrollo para realizar potentes aplicaciones, como Oracle Designer.

Es una herramienta cliente/servidor para la gestión de bases de datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales. Posee un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa muy fuerte. Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas. El software del servidor puede ejecutarse en multitud de sistemas operativos.

1.9 Entorno integrado de desarrollo (IDE)

Un entorno de desarrollo integrado, en inglés Integrated Development Environment (IDE) es un entorno de programación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Dicha aplicación puede formar parte a su vez de otra existente, para brindar facilidad para editar código en casi todos los lenguajes de programación.

Eclipse

Es un entorno de desarrollo integrado (IDE) en su versión 3.4 de código abierto multiplataforma. Para los desarrolladores de software que buscan una herramienta lo más integral posible, Eclipse IDE se muestra como una opción muy viable, ya que además del amplio abanico de extensiones que ya existen, que le permiten implementar variadas tecnologías, permite también desarrollar mediante el mismo las propias extensiones que más tarde se le adicionarán. Gracias a su modularidad, y a que por estar hecho en Java en teoría funciona sobre cualquier plataforma, en el podemos programar además de en Java, en otros lenguajes, como C++, C#, etc.

A pesar de que el Eclipse es gratuito y de código abierto, algunas de sus extensiones no lo son, debido a que son desarrolladas por compañías o individuos con el objetivo de comercializarlas, esto es posible ya que la licencia del IDE no se hace extensiva a sus complementos.

Eclipse dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS (Sistema de Control de Versiones), integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests, etc., y refactorización. Asimismo, a través de "plugins" libremente disponibles es posible añadir control de versiones con Subversion e integración con Hibernate. (6)

1.10 API (Interfaz de Programación de Aplicaciones)

Una API, no es más que un conjunto de especificaciones de comunicación entre componentes software. Se trata del conjunto de llamadas al sistema que ofrecen acceso a los servicios del sistema desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque

Capítulo 1: Fundamentación Teórica

no necesariamente) entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla. De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio.

➤ **JFreeChart API v2.0.2**

JFreeChart es un API libre para Java que facilita la creación de gráficas de distintos tipos, que una vez creadas pueden ser utilizadas perfectamente en una página web o formando parte de un reporte PDF, XLS, etc.

➤ **JasperReport API v1.0.5**

JasperReports es una poderosa API libre para Java para la creación de reportes en variados formatos como PDF, HTML, XLS, CSV y XML de forma relativamente sencilla.

1.11 Conclusiones Parciales

En este capítulo se describieron algunos sistemas penitenciarios y soluciones informáticas de diferentes países que han desarrollado campañas de modernización con el propósito de promover el uso de soluciones informáticas para la gestión de los procesos y el apoyo a la toma de decisiones en estos centros. Estos sistemas penitenciarios carecen de la conectividad entre los centros penitenciarios, lo cual dificulta la inmediatez de la información y la toma de decisiones. Por otra parte también se describieron las principales características de las herramientas, la metodología de desarrollo de software, el lenguaje de programación, los frameworks, entre otros.

Capítulo 2: Análisis y Diseño

Introducción

El modelo de análisis es el resultado de un conjunto de artefactos que propone RUP en el flujo de trabajo Análisis y Diseño. Este modelo permite analizar los aspectos internos del sistema y también ayudan a refinar las funcionalidades del sistema, incluidos sus recursos compartidos. El lenguaje que se utiliza en el análisis, se basa en un modelo de objetos conceptual. Además se desarrollan los diagramas de interacción los cuales se utilizan para describir los aspectos dinámicos del sistema.

El diseño juega un papel importante dentro del ciclo de desarrollo del software debido a que es este el encargado de modelar el sistema, desarrollar la arquitectura y lograr que el software cumpla con los requisitos funcionales y no funcionales además de sustentar la implementación.

2.1 Indicadores

En el SIGEP existe un subsistema Salud Integral que permite llevar el control de las actividades de la atención médico-odontológica dentro del Sistema Penitenciarios. A través del mismo se puede acceder a la Historia Clínica de los individuos, siempre que haya registrado su ingreso al establecimiento penitenciario. Dentro de dicho subsistema se gestionan varios indicadores que van a registrar los datos de salud de los individuos a lo largo de su vida en el sistema penitenciario.

Para seleccionar los indicadores que serán mostrados en los reportes de salud en la Sala Situacional, se analizó con personal de conocimiento médico, cuáles serían los posibles indicadores que más importancia aportarían a la toma de decisiones. Los indicadores que se muestran a continuación fueron escogidos porque brindan elementos para conocer el estado de salud de la población penal. Además son los que más tendencia tienen a provocar situaciones de riesgo, por lo que necesitan un seguimiento de su comportamiento.

Alergias: Las alergias se clasifican según el elemento que la produce, pueden ser alimentaria, medicamentosa u otro tipo de alergia. De las alergias se dará a conocer la cantidad de reclusos que sufren de ella, además del el nombre de reclusos por los distintos tipos de alergias.

Medicamentos: Se mostrará la cantidad de reclusos que consumen medicamentos por cada tipo. También se mostrará un listado con los nombres de cada recluso por cada tipo de medicamento.

Patologías de Interés: Se mostrará la cantidad total de patologías de interés que ha padecido el recluso a lo largo de su vida, a su vez también se dará a conocer el nombre del recluso con la patología asociada.

Inmunizaciones: Se registran las vacunas y la cantidad de pacientes que han sido inmunizados con dicha vacuna. Por otra parte también mostrará un listado de los reclusos por cada dosis de vacuna.

Evaluación Nutricional: Brindará información sobre los reclusos que tienen algún tratamiento dietético por cada tipo de dieta y también se mostrará un listado de los reclusos por cada dieta.

Hábitos Psicobiológicos: Brinda la posibilidad de conocer la cantidad de reclusos que consume algún tipo de hábito psicobiológicos. También el nombre de cada recluso por cada hábito.

2.2 Análisis

El siguiente epígrafe constituye el resultado del estudio de los procesos que tienen lugar para el análisis de la información en los centros penales internos. A partir de este estudio se definen los requisitos funcionales y no funcionales que deberá satisfacer la propuesta de solución, dándose además una descripción detallada de la misma.

2.3 Análisis de la Arquitectura del SIGEP

2.3.1 Visión general de la arquitectura

El Sistema Gestión Penitenciaria se ha desarrollado bajo la tecnología Java Enterprise Edition utilizando como framework arquitectónico base a Spring. Como base de datos Oracle 10g por petición del cliente del software. Como framework de persistencia objetual se utiliza Hibernate. En la capa de presentación como tecnología Spring MVC bajo las adaptaciones realizadas al framework para el proyecto.

Para el desarrollo del SIGEP se definió e implementó como arquitectura base ArBaWeb (Arquitectura Base sobre la Web), un framework que se desarrolló para la arquitectura del proyecto. Esta arquitectura

está organizada teniendo en cuenta dos características fundamentales, la definición de subsistemas y módulos, y el modelo de capas, basado en la arquitectura Cliente-Servidor y la arquitectura de tres capas lógicas bien definidas. (Ver Figura 6)

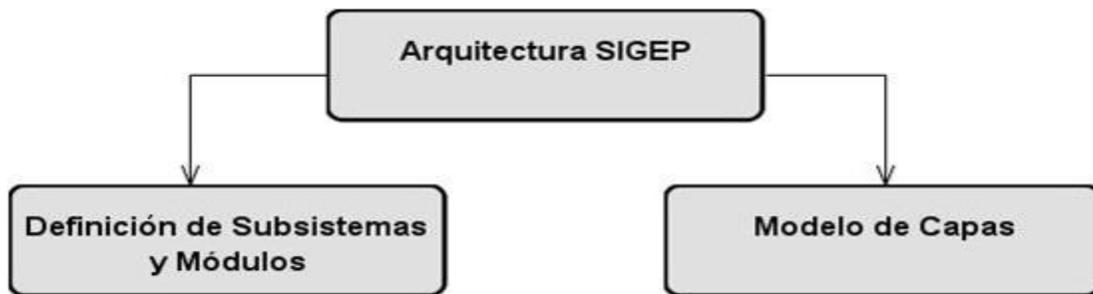


Figura 6: Arquitectura del SIGEP

La arquitectura base definida en ArBaWeb sirve para organizar la forma de codificar según estándares de códigos o recursos predefinidos, brindar una estructura física para soportar el código, creando así una base para el desarrollo, orientar el desarrollo en capas del diseño, y proponer mecanismos de colaboración entre los componentes que la integran. Para ello se basa principalmente en los estilos de arquitectura cliente-servidor, y arquitectura en capas. Cada una de estas capas puede ser modificada tanto como sea posible, sin provocar un impacto importante en las restantes. De este modo se logra a través del uso de interfaces un bajo acoplamiento entre los elementos del sistema. Además, permite que las diferentes capas se encuentren inclusive en distintos nodos físicos. (12)

ArBaWeb propone que la aplicación se divida en tres capas: acceso a datos, lógica de negocio y presentación, donde cada una interactúa con la inmediata inferior mediante interfaces logrando un bajo acoplamiento (12). Las entidades del dominio (persistentes) son utilizadas en cada una de estas capas, por lo que tienen una relación directa con ellas (Ver Figura 7).

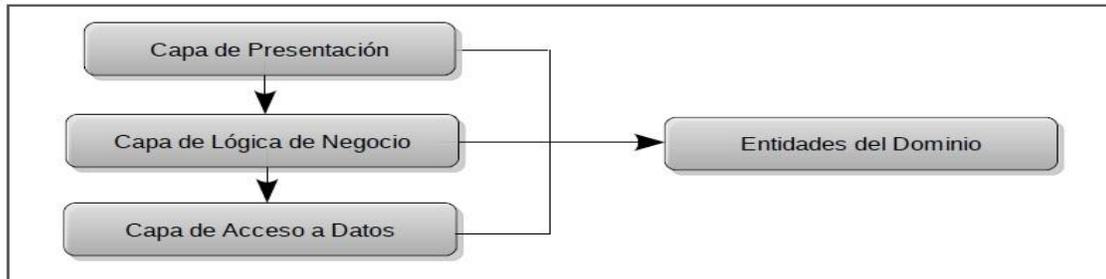


Figura 7: Estructura de capas del sistema

A continuación se describen en detalle cada una de las capas que aparecen en la Figura 7, comenzando de abajo hacia arriba.

Capa de Acceso a Datos:

Capa responsable de persistir y recuperar información hacia y desde la base de datos, se encarga también de la comunicación con el gestor de base de datos utilizando el framework de persistencia objetual Hibernate. Esta capa contiene los objetos que encapsulan la lógica de acceso a datos (DAO) y brinda interfaces para que sea accedida desde la capa de negocio. Las implementaciones de los DAOs estarán disponibles para los objetos (típicamente para los objetos de negocio) haciendo uso de la inyección de dependencias con los objetos de negocio y las instancia de los DAOs, configurada en el contenedor de inversión de control de Spring Framework. (12)

Para el desarrollo del módulo se implementaron interfaces y clases de acceso a datos por cada tipo de indicador, estas tendría los métodos que realizaran las consultas a la base de datos, para obtener la información que se quiere para confeccionar el reporte.

Capa de negocio:

Capa responsable de definir e implementar las funcionalidades que responden directamente a los requisitos. En esta capa radican los objetos de negocio o Business Objects. Los objetos de negocio separan los datos y la lógica de negocio usando un modelo de objetos. Para cada módulo están definidas una o más fachadas en caso de que se requieran las que agrupan los métodos de negocio implementados en los manejadores o managers. La fachada de un módulo está basada en el patrón

Capítulo 2: Análisis y Diseño

Facade para permitir una clara división entre las capas arquitectónicas. Los managers son las clases que se especializan en un conjunto de funcionalidades que representan el negocio sobre una o varias entidades. Estas son las únicas clases en la aplicación que presentan lógica de negocio mientras que las fachadas se limitan solamente a agrupar las funcionalidades de los managers para ser expuestas a capas superiores. (12)

Esta capa va a estar definida por dos paquetes, el fachada donde se encuentran las interfaces e implementación de cada una de las fachadas. El otro paquete que encontramos aquí es el manager donde se va a encontrar un manager por cada indicador, aquí se implementaran cada uno de los métodos que va a resolver en reporte, como por ejemplo, los listados, tablas y contadores.

Capa de Presentación:

En esta capa se utiliza Spring para manejar el flujo de datos entre el cliente y las capas inferiores.

Es la responsable de tratar con las interacciones del usuario y obtener los datos que pueden ser mostrados en un formato determinado. Está compuesta por tres tipos de objetos:

Controladores: Estos objetos son responsables de procesar las entradas del usuario en forma de peticiones HTTP, invocando las funcionalidades necesarias expuestas por la capa de servicios de negocio y devolviendo un modelo requerido para ser mostrado.

Modelo: Estos objetos (command) contienen los datos resultantes de la ejecución de la lógica de negocio los cuales son mostrados en la respuesta.

Vistas: Son responsables de mostrar el modelo resultante en la respuesta de la petición. La forma de mostrar el modelo podrá ser de diferentes tipos de vistas, por ejemplo, archivos JSP, HTML, etcétera. Las vistas no son responsables de modificar los datos o incluso de obtener los datos; estas simplemente sirven para mostrar los datos del modelo que han sido suministrados por un controlador.

La aplicación del patrón arquitectónico llamado MVC (Model View Controller) es la combinación de estos tres objetos.

En esta capa MVC se crearon las vistas y se realizaron los mapeos correspondientes en el servlet, los cuales resuelven las peticiones de cada una de las vistas que le corresponden a los distintos indicadores de salud. Los controladores por su parte implementan los métodos en que se va a mostrar el formato del reporte con los datos.

2.3.2 Definición de Subsistemas y Módulos

Subsistema

Un subsistema es el conjunto de módulos que al ser agrupados por razones de similitud o por objetivos comunes.



Figura 8: Subsistemas del SIGEP.

Módulo

Un módulo encapsula un conjunto de funciones que debe realizar el sistema.

Estructura de un módulo en SIGEP

La estructura interna de cada módulo está compuesta por paquetes, los cuales contienen en su interior la implementación correspondiente a la capa de presentación, capa de lógica de negocio y capa de acceso a datos.

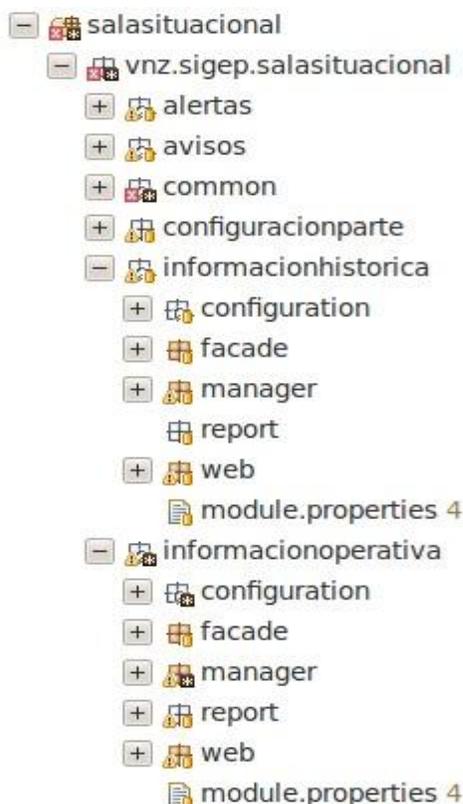


Figura 9: Estructura de paquetes de los módulos de Información Operativa e Histórica

2.4 Características generales del módulo

La Sala Situacional dentro del sistema penitenciario es un espacio para el análisis constante de la información que se genera en Venezuela sobre asuntos penitenciarios, tiene la misión de: Convertirse en una herramienta para el monitoreo de los indicadores del sistema penitenciario permitiéndole al personal especializado analizar situaciones de riesgo o realizar estudios de comportamiento de determinado fenómeno.

El Sistema mostrará dos tipos de información para el análisis ya sea actual como histórico de acuerdo a lo que el funcionario de la sala situacional desee consultar. Para ello contamos con:

Información operativa:

Capítulo 2: Análisis y Diseño

Es un conjunto organizado de datos, que constituyen una información sobre determinado ente o fenómeno reflejando de modo general el estado del mismo. Dichos datos se mostrarán en forma de reportes los cuales darán una vista sobre el estado de la población penal agrupados por diferentes criterios.

Información histórica:

Acumulación de la propia información operativa que luego permitirá establecer evaluaciones entre intervalos de fechas. En este tipo de información se observará el comportamiento en el tiempo de disímiles aspectos del sistema penitenciario mediante diferentes tipos de reportes. Dichos reportes reflejarán la evolución de la información tanto a nivel nacional, regional como a nivel de centro, en intervalos de tiempo según se especifique, ya sea, por años, por mes, por semanas o por días. Las informaciones que brindarán estos reportes serán configuradas por el usuario antes de que sean generados. Asociado a estos reportes se van a generar diferentes tipos de gráficas, las cuales facilitarán la comprensión y el análisis de los datos mostrados.

El Sistema de Gestión Penitenciaria (SIGEP), dará respuesta a las necesidades de gestión, información y apoyo a la toma de decisiones de la Dirección General de Servicios Penitenciarios, en tres niveles:

Operativo: integrado por los Establecimientos penitenciarios (Internados Judiciales y Centros Penitenciarios), Centros de Residencia Supervisada (CRS) y Unidades Técnicas de Supervisión y Orientación (UTSO).

Táctico: integrado por las Coordinaciones Regionales.

Estratégico: integrado por la Dirección General de Custodia y Rehabilitación del recluso.

La información que se genera en estas áreas se mostrará de diversas formas ya sea como información operativa, alertas, avisos, o como comportamiento histórico, esta información se dará a conocer en distintos formatos como en tablas y gráficas, permitiendo a los especialistas que laboren en la Sala Situacional, contar con la información necesaria para tomar decisiones oportunas y tácticas, y para establecer estrategias según corresponda.

2.4 Descripción del sistema propuesto

2.4.1 Requerimientos funcionales

Los requerimientos funcionales son los que se encargan de definir lo que la herramienta de software debe hacer. Definen los alcances del sistema en cuanto a las acciones que debe realizar, y en cuanto a la transferencia de datos entre las diferentes funciones del sistema.

R1. Generar reporte de Alergias

- 1.1 Mostrar la cantidad de individuos que padecen alguna alergia de forma tabular para la información operativa.
- 1.2 Mostrar la cantidad de individuos que padecen alguna alergia de forma gráfica para la información operativa.
- 1.3 Mostrar los nombres de los individuos que padecen alguna alergia en forma de listados para la información operativa.
- 1.4 Mostrar la cantidad de individuos que padecen alguna alergia de forma tabular para la información histórica.
- 1.5 Mostrar la cantidad de individuos que padecen alguna alergia de forma gráfica para la información histórica.
- 1.6 Configurar el intervalo de tiempo de la Información Histórica para mostrar los individuos por cada tipo de alergia.

R2. Generar reporte de Hábitos Psicobiológicos

- 2.1 Mostrar la cantidad de individuos que consumen algún tipo de hábitos psicobiológicos de forma tabular para la información operativa.
- 2.2 Mostrar la cantidad de individuos que consumen algún tipo de hábitos psicobiológicos de forma gráfica para la información operativa.
- 2.3 Mostrar los nombres de los individuos que consumen algún tipo de hábitos psicobiológicos en forma listados para la información operativa.

- 2.4 Mostrar la cantidad de individuos que consumen algún tipo de hábitos psicobiológicos de forma tabular para la información histórica.
- 2.5 Mostrar la cantidad de individuos que consumen algún tipo de hábitos psicobiológicos de forma gráfica para la información histórica.
- 1.7 Configurar el intervalo de tiempo de la Información Histórica para mostrar los individuos por cada tipo de hábitos psicobiológicos.

R3. Generar reporte de Medicamentos

- 3.1 Mostrar la cantidad de individuos que consumen algún medicamento de forma tabular para la información operativa.
- 3.2 Mostrar la cantidad de individuos que consumen algún medicamento de forma gráfica para la información operativa.
- 3.3 Mostrar los nombres de los individuos que consumen algún medicamento de forma de listados para la información operativa.
- 3.4 Mostrar la cantidad de individuos que consumen algún medicamento de forma tabular para la información histórica.
- 3.5 Mostrar la cantidad de individuos que consumen algún medicamento de forma gráfica para la información histórica.
- 3.6 Configurar el intervalo de tiempo de la Información Histórica para mostrar los individuos por cada tipo de medicamento.

R4. Generar reporte de Inmunizaciones

- 4.1 Mostrar las vacunas y la cantidad de individuos que han sido inmunizados, de forma tabular para la información operativa.

- 4.2 Mostrar las vacunas y la cantidad de individuos que han sido inmunizados, de forma gráfica para la información operativa.
- 4.3 Mostrar los individuos que han sido inmunizados en forma de listados para la información operativa.
- 4.4 Mostrar las vacunas y la cantidad de individuos que han sido inmunizados, de forma tabular para la información histórica.
- 4.5 Mostrar las vacunas y la cantidad de individuos que han sido inmunizados, de forma gráfica para la información histórica.
- 4.6 Configurar el intervalo de tiempo de la Información Histórica para mostrar los individuos por cada inmunización.

R5. Generar reporte de Evaluación Nutricional

- 5.1 Mostrar la cantidad de individuos que han recibido algún tratamiento dietético de forma tabular para la información operativa.
- 5.2 Mostrar la cantidad de individuos que han recibido algún tratamiento dietético de forma gráfica para la información operativa.
- 5.3 Mostrar los nombres de los individuos que han recibido algún tratamiento dietético en forma de listados para la información operativa.
- 5.4 Mostrar la cantidad de individuos que han recibido algún tratamiento dietético de forma tabular para la información histórica.
- 5.5 Mostrar la cantidad de individuos que han recibido algún tratamiento dietético de forma gráfica para la información histórica.
- 5.6 Configurar el intervalo de tiempo de la Información Histórica para mostrar los individuos por cada tipo de dieta.

R6. Generar reporte de Patologías de Interés

- 6.1 Mostrar la cantidad de individuos que han padecido alguna patología de interés de forma tabular para la información operativa.
- 6.2 Mostrar la cantidad de individuos que han padecido alguna patología de interés de forma gráfica para la información operativa.
- 6.3 Mostrar los nombres de los individuos que han padecido alguna patología de interés en forma de listados para la información operativa.
- 6.4 Mostrar la cantidad de individuos que han padecido alguna patología de interés de forma tabular para la información histórica.
- 6.5 Mostrar la cantidad de individuos que han padecido alguna patología de interés de forma gráfica para la información histórica.
- 6.6 Configurar el intervalo de tiempo de la Información Histórica para mostrar los individuos por cada patología de interés.

2.4.2 Requerimientos no funcionales

Los requerimientos no funcionales hacen relación a las características del sistema que se aplican de manera general como un todo, más que a rasgos particulares del mismo.

- *Apariencia o interfaz externa*

El sistema deberá contener una interfaz sencilla, amigable e interesante, la cual permita a los usuarios interactuar con el mismo.

- *Usabilidad*

El sistema debe ser de fácil navegación, los funcionarios de la sala situacional son los responsables de la misma en el régimen Intramuros para los distintos centros penales.

- *Portabilidad*

El sistema debe ser multiplataforma.

- *Seguridad*

Para acceder al sistema el funcionario de la sala situacional tendrá que estar registrado en el SIGEP, por lo cual debe tener privilegios de acceso y tener un rol específico de acuerdo a su trabajo.

- *Software*

En el lado del Cliente debe existir el navegador Mozilla Firefox 2.0.

En el lado del servidor deben estar instalados: Servidor de aplicaciones (Tomcat 5.5.17), Servidor de base de datos (Oracle 10g Standard Edition), Jre 6.0, SIGEP y Reko.

- *Restricciones en el diseño y la implementación*

La aplicación deberá estar implementada con el mismo lenguaje, las mismas tecnologías y diseño, al integrarse con el SIGEP como subsistema.

2.5 Descripción del sistema propuesto

2.5.1 Concepción general del sistema

El módulo a desarrollar se integra al SIGEP en su versión 2.1 para el régimen Intramuros del Sistema Penitenciario Venezolano. La Sala Situacional es un subsistema del SIGEP el cual sirve para dar soporte y ayuda a la toma de decisiones, la cual será utilizada por el funcionario de la sala situacional que es el encargado en cada una de las sedes penitenciarias del régimen Intramuros.

El subsistema será capaz de mostrar la información de forma operativa e histórica de los indicadores definidos anteriormente.

El actor del sistema se limita al funcionario de la Sala Situacional.

Tabla1: Actor del sistema

Actores del sistema	Justificación
Funcionario de la Sala Situacional	Es el encargado de llevar el seguimiento de la Sala Situacional a través de indicadores definidos de forma operativa e histórica.

2.5.2 Modelo de casos de uso del sistema

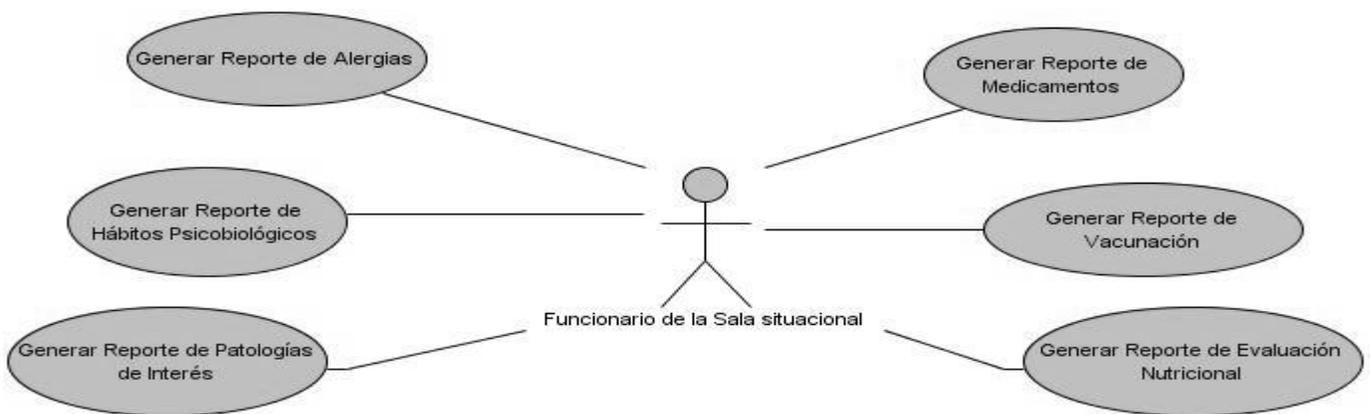


Figura 10: Caso de uso del sistema

Los diagramas de caso de uso describen una interacción típica entre usuarios (actores) y un sistema de cómputo, también es una técnica para capturar información de cómo un sistema o negocio trabaja actualmente o cómo se desea que trabaje. Estos sirven para capturar el comportamiento deseado del sistema sin tener que especificar cómo se implementa ese comportamiento, por otra parte también sirve como medio de comprensión del sistema para desarrolladores, usuarios finales y expertos del dominio y ayudan a validar la arquitectura y a verificar el sistema en el transcurso del desarrollo de este.

2.5.3 Mapas de navegación

Un mapa de navegación es la representación gráfica de la información de una estructura web. Expresa todas las relaciones de jerarquía, secuencia y permite elaborar escenarios de comportamiento de los usuarios. El principal valor de un mapa de navegación es que permite anticipar errores de organización

de la información y el modo de corregirlos. Un mapa es una herramienta de gran utilidad para navegar por un hipertexto, orienta al usuario durante el recorrido y le facilita el acceso directo al lugar que necesite a través de enlaces. A continuación se muestra un ejemplo de un mapa de navegación de la funcionalidad Hábitos Psicobiológicos y los restantes se encuentran en el [ANEXO 1](#).

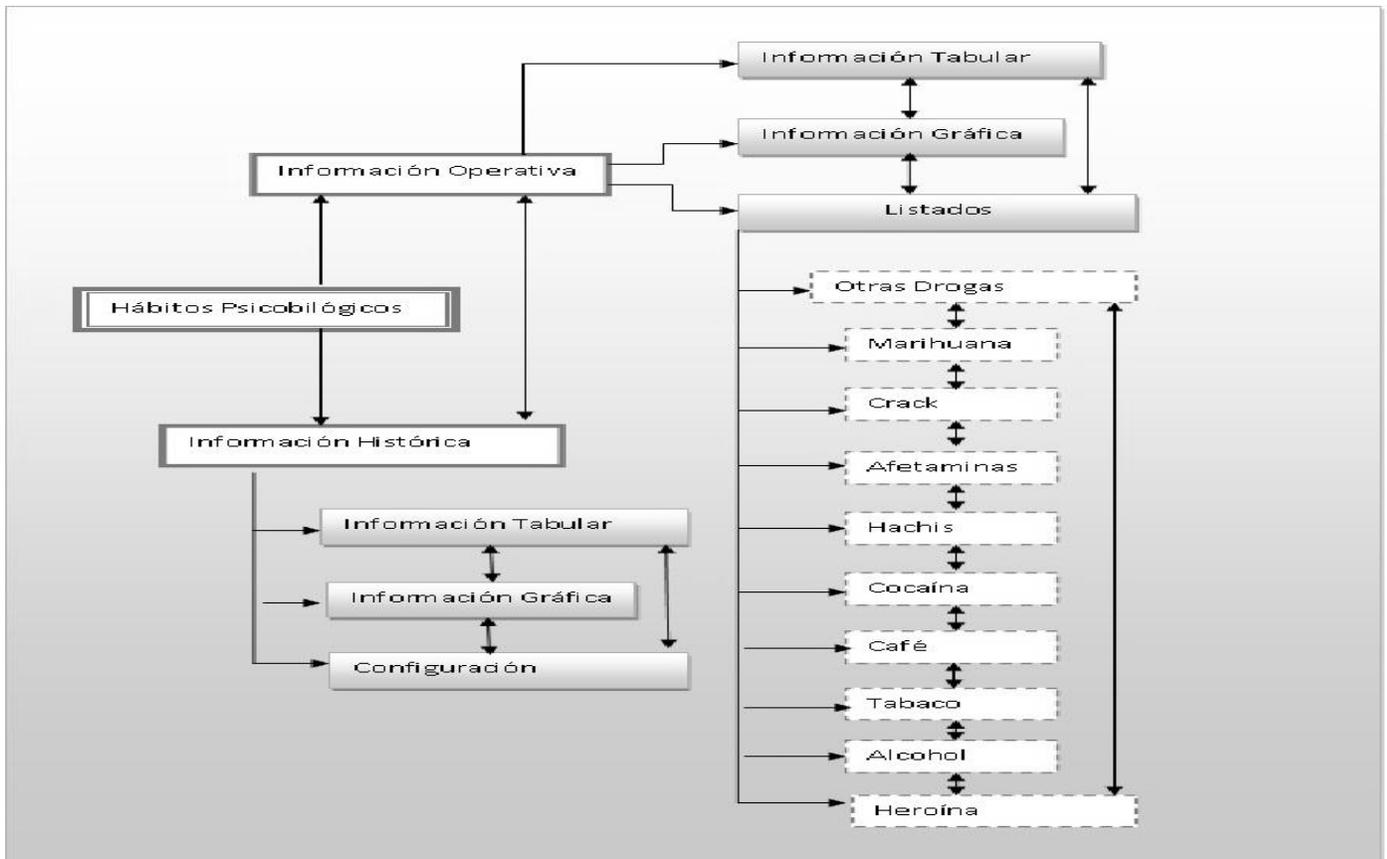


Figura 11: Mapa de Navegación de Hábitos Psicobiológicos

2.5.4 Descripción de casos de uso del sistema

Tabla 2: Descripción del caso de uso Hábitos Psicobiológicos

Descripción del Casos de Uso Generar Reporte de Hábitos Psicobiológicos.

Capítulo 2: Análisis y Diseño

Nombre del Caso de Uso:	Generar Reporte de Hábitos Psicobiológicos.
Objetivo:	Generar reportes de Hábitos Psicobiológicos. Este reporte será mostrado en formato HTML.
Actores:	Funcionario de la Sala Situacional. (Inicia)
Resumen:	Este caso de uso se inicia cuando el Funcionario de la Sala Situacional desee generar reportes de Salud Integral, entrando en el subsistema de Sala Situacional. Luego de seleccionar la opción Hábitos Psicobiológicos, el sistema buscará las configuraciones correspondientes y muestra la Información Operativa o Histórica según el formato deseado.
Precondiciones:	El Funcionario de la Sala Situacional debe estar autenticado.
Garantías (Poscondiciones)	
Mínimas:	Se notificó un mensaje en caso de no poder generar el reporte.
De éxito:	Se generó el reporte Tabular y Gráfico. Si la información es operativa se mostró además la información en forma de listado.
Acción que inicia el caso de uso:	El Funcionario de la Sala Situacional desea Generar el Reporte de Hábitos Psicobiológicos.
Relaciones	
Caso de uso Incluido:	

Capítulo 2: Análisis y Diseño

Caso de uso Extendido:	
Flujo Básico 1: Generar reportes de Hábitos Psicobiológicos	
Acción del actor	Respuesta del Sistema
1. El Funcionario de la Sala Situacional selecciona el reporte de Hábitos Psicobiológicos.	2. El sistema muestra directamente la información Operativa/ Tabular, tiene la opción además de mostrar información Histórica. a. Información tabular. Ver Sección 1
3. El Funcionario de la Sala Situacional selecciona el tipo de información que desea consultar.	4. El sistema muestra las posibles acciones según el tipo de información: 4.1 En el caso de la Información Operativa: a) Información gráfico. Ver Sección 2 b) Listados. Ver Sección 3 4.2 En el caso de la Información Histórica: a) Información tabular. Ver Sección 1 b) Información gráfico. Ver Sección 2 c) Configuración. Ver Sección 4
Sección 1: "Información tabular"	
Acción del actor	Respuesta del Sistema
1. El Funcionario de la Sala Situacional selecciona la opción de Información tabular.	2. El sistema muestra la posibilidad de seleccionar el tipo de hábitos psicobiológicos. Ver sección 5
Sección 2: "Información gráfica"	
Acción del actor	Respuesta del Sistema

Capítulo 2: Análisis y Diseño

1. El Funcionario de la Sala Situacional selecciona la opción de Información gráfica.	2. El sistema muestra la posibilidad de seleccionar el tipo de hábitos psicobiológicos. Ver sección 5
Sección 3: "Listados"	
Acción del actor	Respuesta del Sistema
1. El Funcionario de la Sala Situacional selecciona la opción de Listados.	2. El sistema muestra la posibilidad de seleccionar el tipo de hábitos psicobiológicos. Ver sección 5
3. El funcionario de la Sala Situacional selecciona cualquiera de las opciones.	4. Se muestra el listado correspondiente ordenado alfabéticamente.
Sección 4: "Configuración"	
Acción del actor	Respuesta del Sistema
1. El Funcionario de la Sala Situacional selecciona la opción de Configuración.	2. El sistema muestra los parámetros a configurar: <ul style="list-style-type: none">• Unidad de medida.• Fecha de inicio.• Fecha final.• Frecuencia.
3. El Funcionario de la Sala Situacional selecciona la unidad de medida: <ul style="list-style-type: none">• Año.• Mes.• Semana.	

Capítulo 2: Análisis y Diseño

• Día.	
4. El Funcionario de la Sala Situacional selecciona el intervalo de tiempo para la información histórica.	
5. El Funcionario de la Sala Situacional introduce el valor correspondiente a la frecuencia.	
6. El Funcionario de la Sala Situacional guarda los datos.	7. El sistema verifica que el valor entrado en la frecuencia sea correcto.
	8. El sistema muestra el mensaje de información “Datos guardados correctamente”.
Sección 5: “Seleccionar tipo de Hábitos Psicobiológicos”	
Acción del actor	Respuesta del Sistema
1. El Funcionario de la Sala Situacional selecciona la opción	2. El sistema muestra la información correspondiente a la opción seleccionada.
Flujos Alternos	
Flujo Alterno Flujo Básico 2*a Cerrar	
Acción del actor	Respuesta del sistema
2*a.1 El funcionario de la Sala Situacional selecciona la opción “Cerrar”.	2*a.2 Cierra la interfaz Generar Reporte de Hábitos Psicobiológicos.
	2*a.3 Muestra la interfaz de inicio de la Sala

Capítulo 2: Análisis y Diseño

	situacional.
Flujo alternativo Sección 1: 2a No existe información tabular que mostrar.	
Acción del actor	Respuesta del sistema
	2a.1 El sistema muestra el mensaje de información: "No existen datos para mostrar".
Flujo alternativo Sección 2: 2a No existe información gráfica que mostrar.	
Acción del actor	Respuesta del sistema
	2a.1 El sistema muestra el mensaje de información: "No existen datos para mostrar".
Flujo alternativo Sección 3: 4a No existe listados que mostrar.	
Acción del actor	Respuesta del sistema
	4a.1 El sistema muestra la interfaz vacía.
Flujo alternativo Sección 4: 2a Cancelar la configuración.	
Acción del actor	Respuesta del sistema
2a.1 El funcionario de la Sala Situacional selecciona la opción "Cancelar".	2a.2 El sistema retorna a la última configuración guardada.
Flujo alternativo Sección 4: 7a Datos Incorrectos.	
Acción del actor	Respuesta del sistema
	7a.1 Se comprueba que los datos son incorrectos.

	7a.2 Se muestra un mensaje de error “El valor de la frecuencia no es correcto”.
	7a.3 Se muestra de nuevo la interfaz de Configuración para que vuelva a introducir un nuevo valor de Frecuencia.

En el [ANEXO 2](#) se encuentran las restantes descripciones de casos de uso, con su respectivo prototipo de interfaz gráfica de usuario tanto de la información operativa como de la histórica.

2.5.4.1 Prototipo de interfaz gráfica de usuario de la información operativa

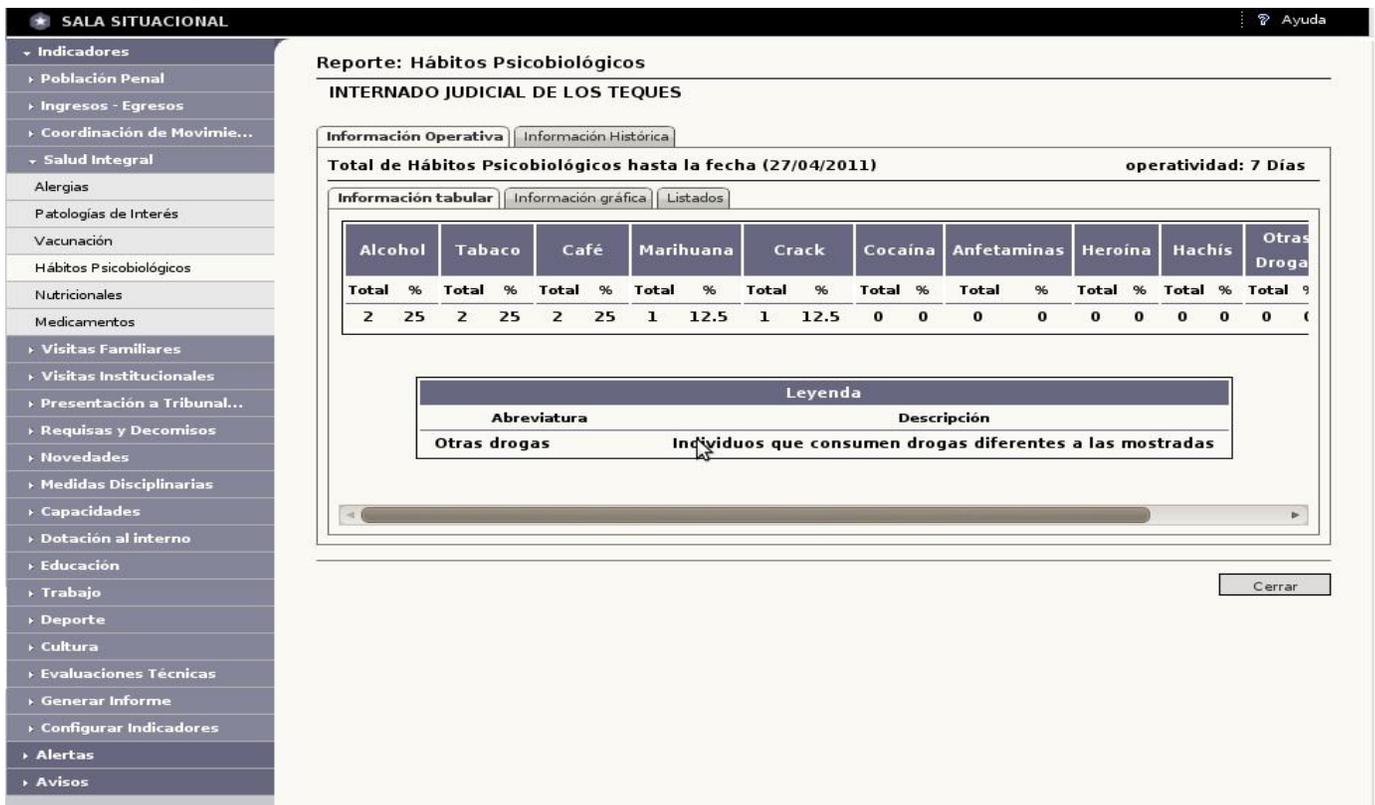


Figura 12: Hábitos Psicobiológicos\Información Operativa\Información tabular

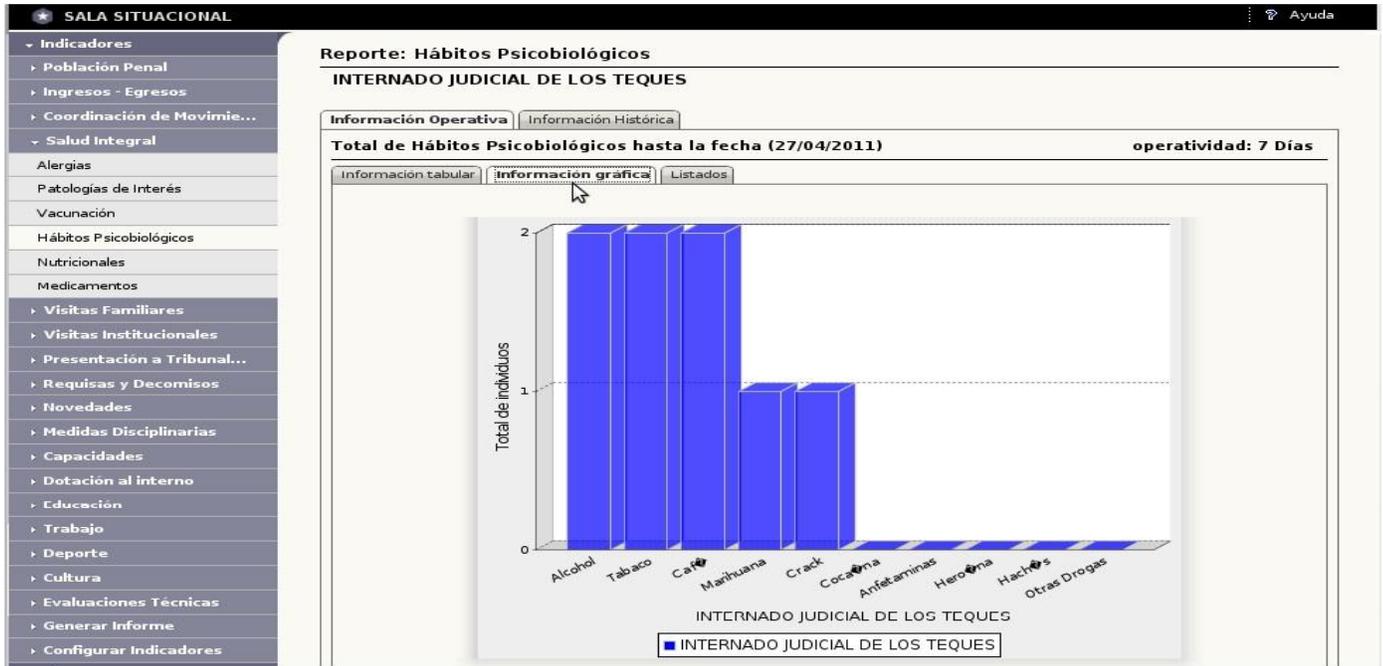


Figura 13: Hábitos Psicobiológicos\Información Operativa\Información gráfica



Figura 14: Hábitos Psicobiológicos\Información Operativa>Listados

2.5.4.2 Prototipo de interfaz gráfica de usuario de la información histórica

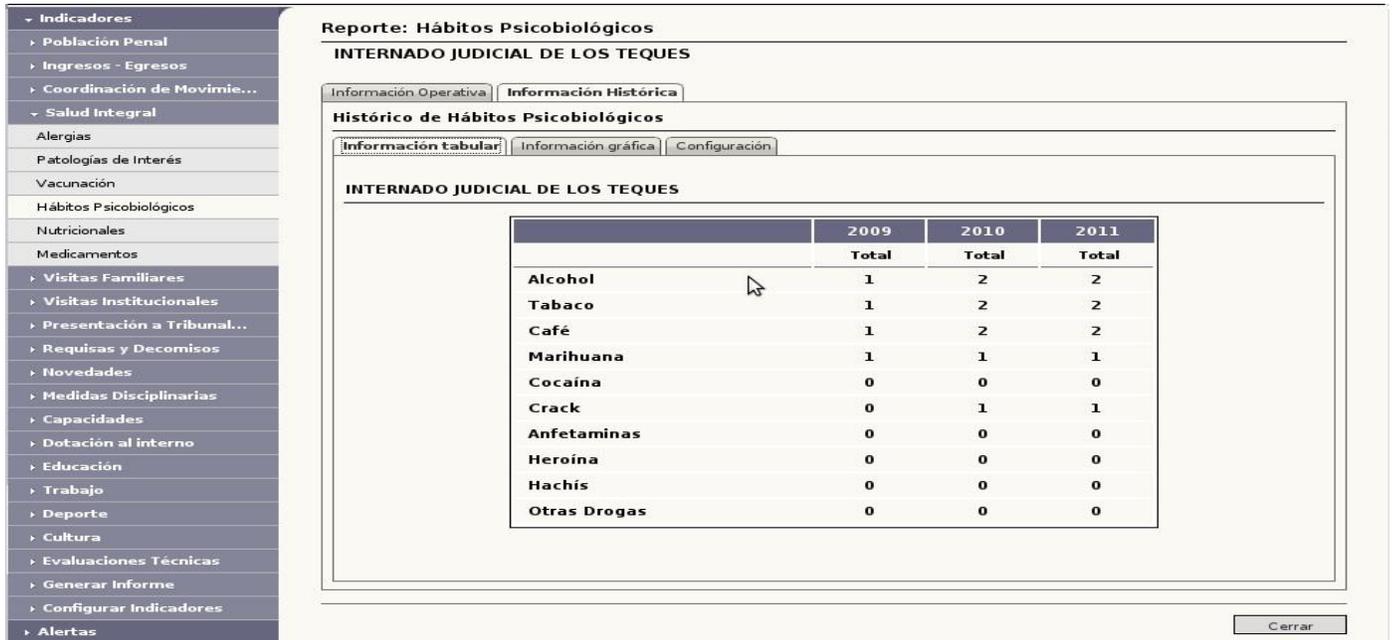


Figura 15: Hábitos Psicobiológicos\Información Histórica \ Información tabular

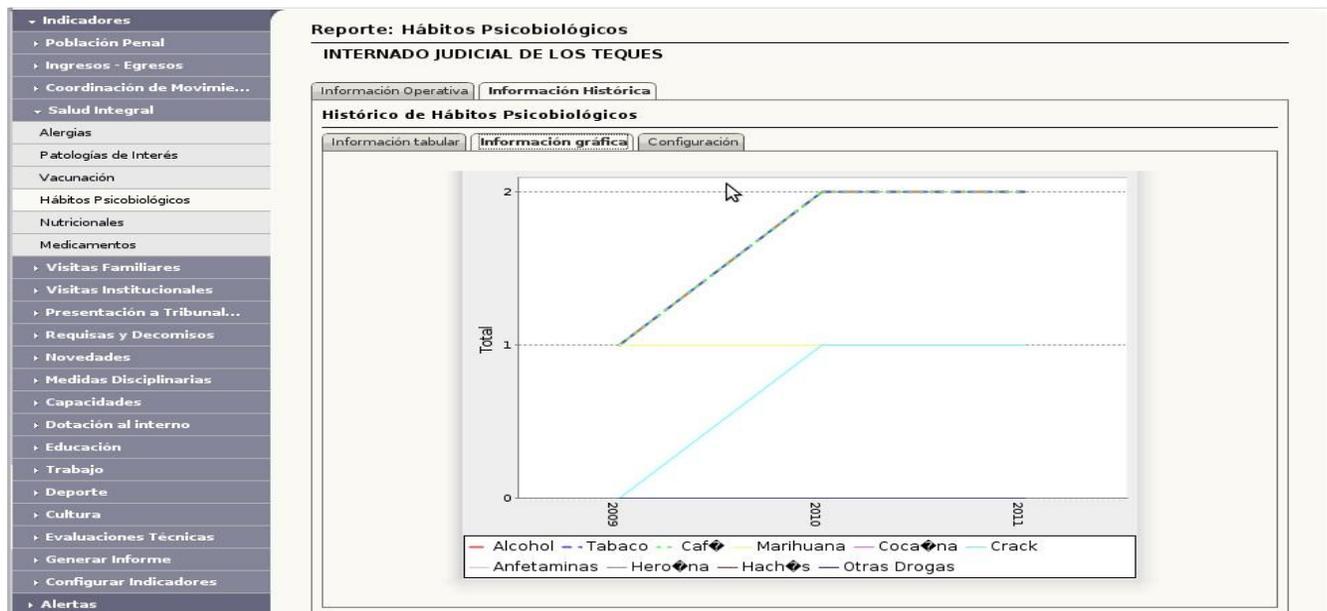


Figura 16: Hábitos Psicobiológicos\Información Histórica \ Información gráfica

Reporte: Hábitos Psicobiológicos

INTERNADO JUDICIAL DE LOS TEQUES

Información Operativa | Información Histórica

Histórico de Hábitos Psicobiológicos

Información tabular | Información gráfica | Configuración

Unidad de medida: Año

Fecha de Inicio: 2009

Fecha Final: 2011

Frecuencia: 1

Guardar Cancelar

Cerrar

Figura 17: Hábitos Psicobiológicos\Información Histórica \ Configuración

2.6 Diseño

La etapa del diseño es importante porque es el proceso sobre el que se asienta la calidad del software y también es un proceso iterativo a través del cual se traducen los requisitos en una representación del software. El diseño debería proporcionar una completa idea de lo que es el software, enfocando los dominios de datos, funcional y de comportamiento desde la perspectiva de la implementación. (7)

2.6.1 Patrones de diseño

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Es una solución estándar para un problema común de programación, una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios y es una manera más práctica de describir ciertos aspectos de la organización de un programa.

Capítulo 2: Análisis y Diseño

Un patrón de diseño es una abstracción de una solución en un nivel alto. Los patrones solucionan problemas que existen en muchos niveles de abstracción. Hay patrones que abarcan las distintas etapas del desarrollo; desde el análisis hasta el diseño y desde la arquitectura hasta la implementación. (19)

Algunos de los patrones de diseño que se utilizaron a lo largo del proceso de diseño para el desarrollo del proyecto SIGEP fueron:

Objeto de Acceso a Datos (DAO):

Este patrón gestiona las distintas fuentes de datos, además encapsula la fuente de datos sirve también para ocultar la forma para acceder a ellos. Por otra parte se logra aislar la capa de acceso a datos del resto de la aplicación, logrando con ello un bajo acoplamiento de la misma con las capas superiores. Estos nos permiten contar con diferentes fuentes de datos, como base de datos, servicios externos, archivos, etc.

No es imprescindible, pero en proyectos de cierta complejidad resulta útil que el DAO implemente un interfaz. De esta forma los objetos cliente tienen una forma unificada de acceder a los DAO.

El DAO accede a la fuente de datos y la encapsula para los objetos clientes. Entendiendo que oculta tanto la fuente como el modo (JDBC) de acceder a ella.

Modelo Vista Controlador (Model View Controller):

Para el diseño de aplicaciones con sofisticados interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador. La lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. En caso de utilizar un diseño que mezcle todos los componentes de la aplicación, al tener que realizar algún cambio en la interfaz, esto afectaría directamente a los demás componentes, dificultando el cambio y aumentando la probabilidad de aparición de errores. Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos. Ver figura 16.

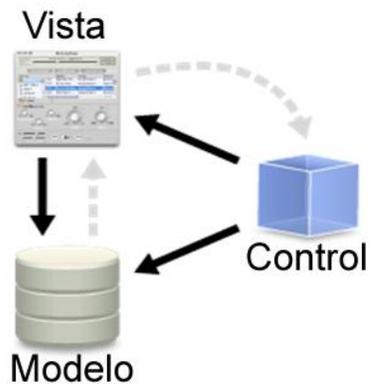


Figura 18: Modelo Vista Controlador

El **modelo** es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio.
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo.

El **controlador** es responsable de:

- Recibe los eventos de entrada.
- Contiene reglas de gestión de eventos. Estas acciones pueden suponer peticiones al modelo o a las vistas.

Las **vistas** son responsables de:

- Recibir datos del modelo y las muestras al usuario.
- Tienen un registro de su controlador asociado.

Fachada (Facade):

Fachada es un patrón de diseño que sirve para proveer de una interfaz unificada sencilla que haga de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas. Facilita cualquier cambio que se realice en las diferentes capas de una aplicación afectando lo menos posible a las restantes capas. El uso de un bean de sesión como una fachada para encapsular la complejidad de las interacciones entre los objetos de negocio y participantes en un flujo de trabajo. El SessionFacade maneja los objetos de negocio y proporciona un servicio de acceso uniforme a los clientes.

2.7 Diagrama de clases

Describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Normalmente los diagramas contienen clases, asociaciones, atributos, interfaces con sus operaciones y constantes, métodos, navegabilidad y dependencias. En el [ANEXO 3](#) se pueden encontrar los diagramas de clases del diseño.

2.7.1 Descripción de los diagramas de clases del diseño

Los diagramas de clases del diseño están divididos en tres capas: Capa de Presentación, Capa Lógica del Negocio y Capa de Acceso a Datos. La funcionalidad a la cual se le hará referencia es a Hábitos Psicobiológicos.

➤ Capa de Presentación

Es la capa encargada de la interacción con los diversos tipos de usuarios, modela la forma en que se muestran y se recogen los datos entrados por los usuarios, también controla la apariencia visual de la aplicación. Se comunica con la capa de lógica de negocio a la cual envía todas las solicitudes de los usuarios y recibe la respuesta después de que hayan sido procesadas cada una de estas solicitudes.

Los controladores son los encargados de las solicitudes y respuestas ejemplo de ello es el CommonTableController que gestiona toda la información referente a las tablas, por otra parte se encuentra GestorReporteInformacionOperativaController y GestorReporteInformacionHistoricaController los cuales tienen la encomienda de gestionar la información histórica y operativa.

➤ Capa Lógica del Negocio

Es la encargada de modelar la lógica de negocio que dará solución a cada uno de los casos de usos de la aplicación, es en esta donde se establecen las reglas o restricciones que debe cumplir la aplicación. Se comunica con la capa de presentación para recibir las solicitudes de los usuarios y enviar las respuestas después del procesamiento.

En esta capa se encuentra la fachada tanto de carácter histórico como operativo, la cual es la encargada de interactuar con la capa de presentación. También están los managers y sus implementaciones que a través de estos se relaciona con la capa de acceso a datos.

➤ Capa de Acceso a Datos

Es la autorizada del manejo de los datos persistentes, forma parte de la misma el servidor de base de datos como encargado de almacenamiento de los mismos, se comunica con la capa de lógica de negocio por la interfaz IndividuoPorCentroHabitosDAO, desde donde recibe las solicitudes de almacenar o recuperar información.

2.8 Modelo de Datos

El modelo de datos es refinado en la medida en que se avanza en las actividades de diseño porque se pueden persistir nuevos elementos o se utilicen procedimientos almacenados, vistas, restricciones o funciones definidas en el gestor de base de datos. La utilización de estos objetos de la base de datos es analizada detenidamente y se decide siempre que implique mayor rendimiento en el acceso a los datos y un acoplamiento mínimo al gestor de base de datos. El modelo de datos generados para el subsistema Salud Integral para la ampliación de la Sala Situacional se encuentra en el [ANEXO 5](#).

2.9 Conclusiones Parciales

En el capítulo se comenzó a desarrollar la propuesta de solución a partir del estudio realizado, donde se definieron los requisitos tanto funcionales como los no funcionales, así como los actores del sistema. La utilización de los patrones de diseño como el MVC, FACADE y DAO, ayudó al desarrollo de la solución propuesta, formalizó un vocabulario común entre el equipo de desarrollo y proporcionó elementos reusables en el diseño del sistema de software.

Capítulo 3: Implementación

Introducción

En el presente capítulo se presentan los elementos fundamentales de la fase de implementación del módulo Salud Integral del subsistema Sala Situacional. De forma representativa a los artefactos que se generan en esta fase, se describen y se muestran fragmentos de códigos de la implementación de funcionalidades características del módulo.

3.1 Implementación de las entidades del dominio

La implementación de las entidades del dominio en general es sencilla debido que su comportamiento no es de gran complejidad. Además de los métodos comunes a todas las entidades, no suele hacerse énfasis en la programación de funcionalidades específicas que serán características de las capas a las que pertenezca cada entidad. Los métodos `equals ()`, `hashCode ()` y `toString ()` deben ser implementados en la mayoría de los casos, ya que son frecuentemente utilizados.

3.2 Implementación de las interfaces de los managers

Estos serán los encargados de gestionar la información que fluya desde y hacia la capa de interfaz de usuario, así como de lanzar y capturar eventos del sistema cuando sea necesario.

Los managers implementados se configuran en el fichero de configuración del contexto de Spring correspondiente a la capa de negocio del módulo. Este fichero se encuentra en el paquete `configuration` y tiene por nombre: `sigep-salasisituacional-informacionoperativa-business-context.xml`.

3.3 Implementación de la interfaz de la fachada

Para la implementación de la interfaz de la fachada solo tiene que conocerse en cuales managers se encuentran implementadas las funcionalidades necesarias para comunicarlás a la capa de interfaz; de esta forma, la fachada no posee ninguna lógica de negocio, solo es experta del lugar donde radica la información que necesita y así la utiliza. El uso de fachadas disminuye el acceso de la capa de presentación a la capa de negocio, reduciendo el número de objetos con los que tiene que interactuar esta.

3.4 Implementación de la capa de acceso a datos

La implementación de la capa de acceso a datos comprende fundamentalmente la creación de los ficheros de mapeo de Hibernate (hbm.xml) y la implementación de los objetos de acceso a datos. Los ficheros hbm se colocan en el paquete dao.impl.map.

La implementación de los DAOs se centra en la utilización de la API Criteria del Framework Hibernate. Esta API es una poderosa herramienta para la creación de complejas consultas de forma relativamente fácil. La utilización de la API mencionada y del HQL es recomendada porque permite escribir código más sencillo, legible y tolerante al cambio.

La implementación de los DAOs está acoplada de forma mínima al gestor de base de datos. Los DAOs implementados se configuran en el fichero sigep-salasisituacional-common-dataaccess-context.xml. En la configuración de estos objetos se inyecta el sessionFactory de Hibernate.

3.5 Implementación de los controladores

A partir del diseño realizado de la capa de presentación se programan los controladores que manejan el flujo web de la aplicación. Los controladores implementan directa o indirectamente la interfaz, org.springframework.web.servlet.mvc.Controller para insertarse en el flujo de Spring-MVC. Estos componentes son los encargados de la comunicación con la capa de negocio a través de su fachada. Son responsables de validar los datos de entrada de la aplicación, formatear los datos de salida y gestionar el flujo web mostrando las vistas correspondientes. Los controladores se configuran en el fichero del contexto de Spring correspondiente a la capa de presentación: sigep-salasisituacional-informacionoperativa-servlet.xml.

3.6 Implementación de las páginas JSP

Las páginas JSP construyen documentos HTML con los datos que el controlador le envía. En la programación de las JSP se usan fundamentalmente la biblioteca de etiquetas de Spring. El diseño gráfico de las JSP se realiza a través del uso de estilos que radican en el fichero CSS definido para el SIGEP, donde se encuentran todos los estilos de la aplicación.

3.7 Implementación de la lógica en el cliente

La programación de la lógica de la aplicación se realizó con la utilización de JavaScript, para validaciones en el cliente además de componentes gráficos de las librerías de DojoToolkit. Desde el lado del cliente se interactúa con la aplicación de manera asíncrona aprovechando las posibilidades que brindan AJAX y JSON para lanzar peticiones al servidor.

3.8 Modelo de despliegue

Es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es un mapa específico de la instalación física del sistema. Un diagrama de despliegue ilustra el despliegue físico del sistema en un ambiente de producción (o prueba). Muestra dónde se ubicarán los componentes, en qué servidores, máquinas o hardware. Puede ilustrar vínculos de red, ancho de banda de LAN, etc.

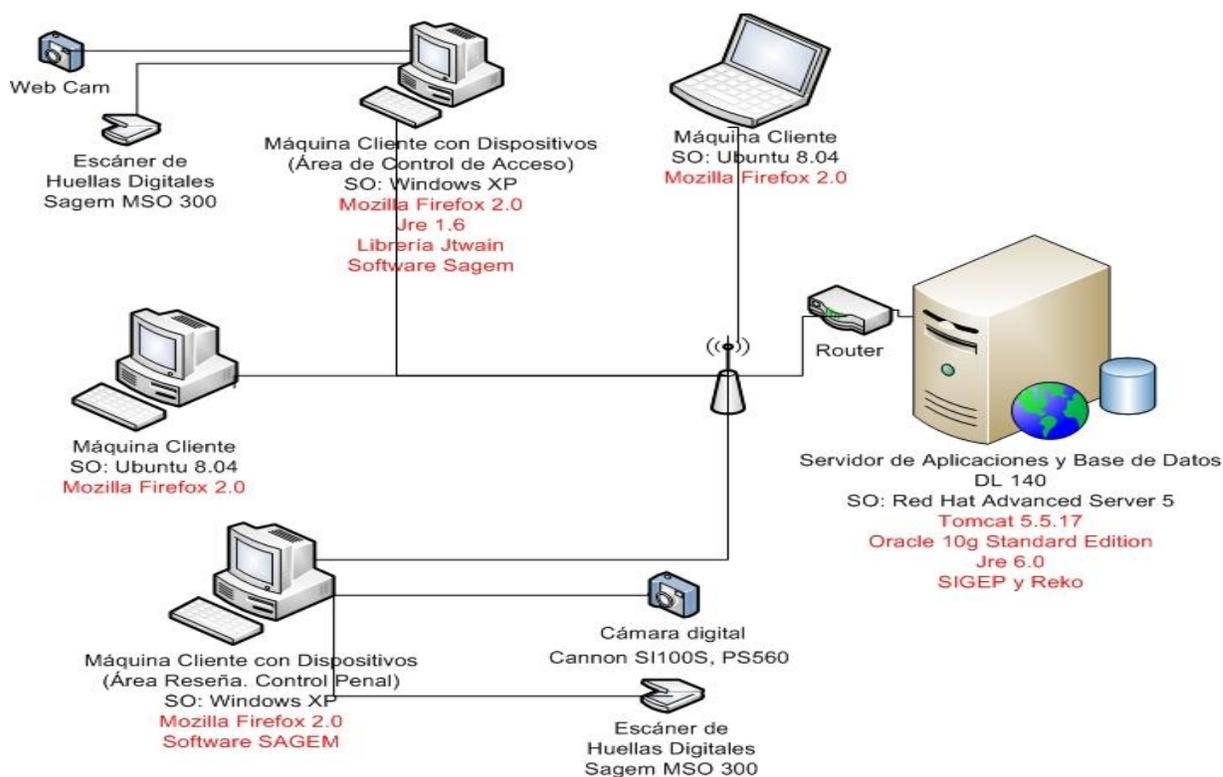


Figura 19: Modelo de despliegue para el régimen Intranuros

3.9 Descripción del Modelo de Componentes

El modelo de componentes ilustra los componentes de software que se usarán para construir el sistema. El diagrama de componentes muestra la relación entre componentes de software, sus dependencias, su comunicación su ubicación y otras condiciones. En el [ANEXO 4](#) se encuentran los modelos de componentes las funcionalidades.

Los diagramas están divididos en tres capas y en cada una de ellas se encuentran sus respectivos componentes.

En la **Capa de Presentación** se encuentra el fichero commonArgTemplate.jsp, el cual se usa los ficheros que existen en los paquetes JS y Dojo para mostrar la página cliente. Esta JSP a su vez se relaciona con el paquete MVC (Modelo Vista Controlador) de Spring para darle respuesta a sus peticiones, en el cual se encontrarán los servlets que mapean todas las peticiones y las respuestas mediante los controladores.

La **Capa de Negocio** se localiza el paquete fachada el cual va a contener el fichero interfaz e implementación de la fachada. Los controladores de la capa de presentación se relacionan con estas interfaces de fachada. También se encuentra el paquete que contiene los managers donde se hallan los managers de cada uno de los tipos de información y lo que da solución a cada uno de los indicadores. Las interfaces de los managers de los distintos tipos de información van a estar relacionadas con las fachadas correspondientes.

En la **Capa de Acceso a Datos** se encuentra el paquete DAO que va a contar con una interfaz de la cual se heredan las demás interfaces de cada uno de los indicadores de salud, de estos también se localiza su implementación. Se encontrará también una clase abstracta que implementará métodos de acceso a datos que serán útiles en la implementación de cada uno de los ficheros de los indicadores. La clase abstracta mencionada anteriormente y la interfaz de la cual heredan las interfaces, van a heredar de los ficheros de acceso a datos del paquete ArBaWeb y los mismos se relacionan con el paquete donde se encuentran los mapeos de Hibernate lo que relaciona directamente a cada una de las entidades del dominio de los indicadores.

3.10 Conclusiones Parciales

En este capítulo se presentaron los elementos más significativos en la implementación del módulo Salud Integral. Se siguieron los lineamientos de la arquitectura del sistema y el diseño definido para el cumplimiento de las funcionalidades. La utilización de librerías y facilidades de los frameworks durante el

Capítulo 3: Implementación

proceso de implementación agilizó el proceso ahorrando código y tiempo del cronograma. La implementación de los componentes genéricos definidos, utilizando los estándares de código establecidos para el proyecto y la documentación generada en el código fuente eleva las facilidades y posibilidades de reutilización, mantenibilidad y extensión del módulo.

Capítulo 4: Pruebas

Introducción

Este capítulo resumirá el conjunto de pruebas realizadas al módulo Salud Integral del subsistema Sala Situacional para comprobar la satisfacción de los requisitos funcionales y no funcionales asociados al mismo. Las pruebas de software son el elemento crítico para la garantía de la calidad del software, lo cual representa una revisión final de las especificaciones del diseño y de la codificación.

4.1 Pruebas de software

Las pruebas de software implican ejecutar una implementación del software con datos de prueba. Se examinan las salidas del software y su entorno operacional para comprobar que funcionan tal y como se requiere. (22)

Las pruebas de software permiten verificar y revelar la calidad del producto de software. Por otra parte son utilizadas para identificar posibles fallos. Las mismas tienen como principios fundamentales:

- La prueba puede ser usada para mostrar la presencia de errores, pero nunca de su ausencia.
- Las pruebas deberían empezar por "lo pequeño" y progresar hacia "lo grande".

4.2 Herramientas para automatizar las pruebas

JUnit 3. 8.1

Para la realización de las pruebas unitarias se utilizó JUnit, integrado al IDE Eclipse. Gracias a esta herramienta se crearon plantillas para la generación de las pruebas de una clase Java, de forma automática, facilitando al programador, enfocarse en la prueba y el resultado esperado.

JMeter 2.3.4

Se utiliza en aplicaciones web donde se comprueban los recursos del sistema, posee una mayor efectividad en el proceso y en la fiabilidad de los resultados. Es una herramienta de software libre que ofrece la posibilidad de realizar pruebas de carga y dispone de varios componentes que facilitan la elaboración de los escenarios de prueba. Tiene como ventaja fundamental, simular para cada uno de esos escenarios miles de usuarios. Los resultados pueden ser visualizados en diferentes formatos lo cual

facilita las labores de análisis para el tester, por lo que se verifica el rendimiento del sistema mediante las pruebas de Carga y Estrés.

4.3 Niveles de Pruebas

Existen varios niveles de pruebas y cada una es aplicada para disímiles objetivos, en diferentes escenarios o niveles de trabajo. Los niveles de pruebas son los siguientes:

- Prueba de Desarrollador
- Prueba Independiente
- Prueba de Unidad
- Prueba de Integración
- Prueba de Sistema
- Prueba de Aceptación

Para realizar las pruebas del módulo Salud Integral para la ampliación del subsistema Sala Situacional se desarrollaron las siguientes pruebas:

Prueba de Desarrollador

Es la diseñada e implementada por el equipo de desarrollo. Estas pruebas han sido consideradas solo para las pruebas de unidad aunque también se pueden ejecutar en las pruebas de integración.

Prueba de Unidad

Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera. El objetivo es aislar cada parte del programa y mostrar que las partes individuales son correctas. En esta prueba se utilizó la herramienta JUnit.

Prueba de Sistema

Es la prueba que se le aplica al sistema cuando el software funciona como un todo. Después que se hayan integrado todos los componentes de hardware y software, es la que verifica el producto final.

4.4 Técnicas de pruebas

Las técnicas de pruebas se clasifican en pruebas funcionales, de usabilidad, fiabilidad, rendimiento y soportabilidad.

Pruebas de funcionalidad

Dentro de las pruebas de funcionalidad se encuentran:

➤ Prueba Funcional

Objetivo: Centrarse en cualquier requisito que pueda ser trazado directamente hacia los casos de uso o funciones del negocio y reglas de negocio.

Metas: Es verificar la aceptación apropiada de datos, de proceso, y de recuperación, así como la implementación apropiada de las reglas de negocio.

Método de prueba: Este tipo de prueba se basa sobre técnicas de la caja negra; es decir verifica la aplicación y los procesos internos interactuando con la aplicación a través de las interfaces gráficas de usuario (GUI) y analizando la salida o los resultados.

A continuación se muestran de forma detallada los resultados encontrados en las pruebas, obtenidas para contabilizar a modo de resumen el total de No Conformidades del módulo Salud Integral del subsistema Sala Situacional.

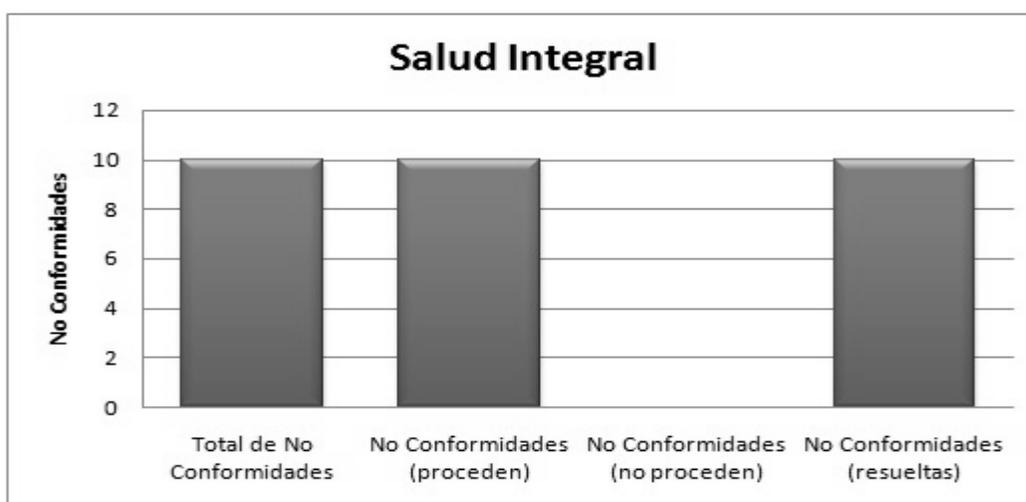


Figura 20: Resumen de las no conformidades del módulo Salud Integral del subsistema Sala Situacional

➤ Prueba de Seguridad

Estas pruebas se centran en dos niveles de seguridad:

- Nivel de seguridad de la aplicación: Incluye el acceso a los datos o funciones de negocio.

- Nivel de seguridad del sistema: Incluye la entrada dentro o accesos remotos al sistema.

Asegura:

- Que a los usuarios se les restrinjan las funciones específicas o casos de uso, o que se limiten los datos disponibles para ellos.
- Que solamente esos usuarios con privilegios para acceder al sistema sean capaces de tener acceso a la aplicación y solamente a través de las entradas apropiadas.

Para la realización de estas pruebas se crearon dos usuarios donde cada uno de ellos iba a tener acceso a información diferente. El usuario ssOperativa solo tendría permisos para el acceso a la Información Operativa y el ssHistorico solo iba a acceder a los datos Históricos.

Dentro del módulo de salud del subsistema Sala Situacional con el usuario ssOperativa autenticado:

- En cada uno de los indicadores se pueden tener los reportes operativos con sus diferentes tipos de formato de información.

Reporte: Alergias
INTERNADO JUDICIAL DE LOS TEQUES

operatividad: 7 Días

Total de Alergias entre (13/05/2011 y 20/05/2011)

Pescados frescos		Leguminosas (arvejas)		Frutas Cítricas (naranja)		Quesos frescos		Frutas		Otros Alimentos		Total de Individuos
Total	%	Total	%	Total	%	Total	%	Total	%	Total	%	-
1	25	1	25	0	0	0	0	0	0	2	50	2

Leyenda

Abreviatura	Descripción
Otras Alimentos	Individuos que padecen de alergias diferentes a las mostradas

Figura 21: Acceso al usuario ssOperativa a la Información Operativa

- El usuario no tiene acceso a la Información Histórica de los indicadores de salud.

Dentro del módulo de Salud Integral del subsistema Sala Situacional con el usuario ssHistorico autenticado:

- En cada uno de los indicadores se pueden tener los reportes históricos con sus diferentes tipos de formato de información y con la posibilidad de configurar el intervalo de tiempo de dicho reporte.
- El usuario no se tiene acceso a la Información Operativa de los indicadores de salud.

Prueba de Confiabilidad

La confiabilidad del software se realiza para evaluar el rendimiento y está relacionado también con el control de la detección de errores y de la recuperación para evitar que se produzcan errores.

➤ **Prueba de Stress**

Objetivo: Se centran en evaluar cómo responderá el sistema bajo condiciones anormales. Las tensiones sobre el sistema pueden incluir cargas extremas, insuficiente memoria, servicios y hardware no disponibles, o recursos compartidos limitados.

Prueba de Rendimiento

Se realiza cuando se puede determinar cuán rápido ejecuta una tarea un sistema en condiciones particulares de trabajo. Entre ellas se pueden encontrar las pruebas de carga.

➤ **Prueba de carga**

Es una prueba del funcionamiento que sujeta al elemento de prueba a cargas de trabajo que varían para medir y evaluar los comportamientos de desempeño y las capacidades del elemento de prueba para continuar funcionando correctamente bajo estas diversas cargas de trabajo.

El propósito de la prueba es determinar y asegurar que el sistema funciona correctamente más allá de la carga de trabajo máxima prevista. Además, la prueba evalúa las características de funcionamiento, tales como tiempos de reacción, porcentajes de transacción, y otros elementos sensibles al tiempo de la carga.

Para estas pruebas se utilizó la herramienta JMeter y tres PC (dos clientes y una servidora).

➤ **Simulación de 30 usuarios con un período de subida de 30 segundos**

Se realizó la simulación con 30 usuarios considerando un período de subida de 30 segundos (con 1 segundo para lanzamiento de cada hilo) los resultados serán los siguientes, teniendo en cuenta que dichos se irán solapando a los ya obtenidos en la simulación anterior.

En la figura anterior podemos observar que las pruebas se han realizado con errores de un 0,91%. El rendimiento nos muestra que para una simulación de 30 usuarios junto a un período de subida de 30 segundos el servidor es capaz de aceptar una media de 28,0 peticiones por segundos. La latencia (entendida como el tiempo de espera para la renderización de la página, el tiempo en obtener respuesta del servidor) para cada conjunto de pruebas no supera el valor de 465 milisegundos (representado por el eje “y” de la gráfica).

➤ **Simulación de 60 usuarios con un período de subida de 180 segundos**

Se realizó la simulación con 60 usuarios considerando un período de subida de 180 segundos (con 3 segundo entre lanzamiento de cada hilo) los resultados serán los siguientes.

Se observa que las pruebas se han realizado sin errores. Esto se deduce de la columna representativa del tanto por ciento de errores para cada una de las peticiones asociadas a cada conjunto de muestras. El rendimiento nos muestra que para una simulación de 60 usuarios junto a un período de subida de 120 segundos el servidor es capaz de aceptar una media de 18,3 peticiones por segundo. La latencia para cada conjunto de pruebas no supera el valor de 3679 milisegundos.

➤ **Simulación de 120 usuarios con un período de subida de 360 segundos**

En la simulación no se pudo terminar las pruebas debido a que la herramienta que se encontraba ejecutándose en las dos PC clientes no soportaban la conexión de más de 120 usuarios conectados simultáneamente con un período de subida de 360 segundos, por lo que colapsó el sistema y no se pudo comprobar el límite en que colapsa el sistema.

4.5 Conclusiones Parciales

Una correcta estrategia de validación y verificación ayuda a mejorar y asegurar la calidad de su desarrollo y permite reducir los costos de corrección de errores. Las ventajas son claras: un mayor control sobre el proceso, una identificación temprana de errores y una reducción drástica de los costos para subsanar dichos errores, por eso la atención esmerada que se le presta a esta actividad. Con la realización de las

pruebas dieron paso a las 10 No Conformidades la cuales fueron resueltas en su totalidad, se comprobó que los usuarios solo tengan acceso a consultar información tanto de carácter histórico como operativo.

Conclusiones

En la confección del presente trabajo de diploma se le dio cumplimiento al objetivo general planteado contribuyendo al mejoramiento y ampliación del subsistema Sala Situacional del SIGEP.

Se le realizó el análisis, diseño, implementación y pruebas al módulo de reportes de salud para la Sala Situacional utilizando las herramientas y convenciones definidas por la arquitectura del SIGEP.

Se garantizó la integración del módulo Salud Integral para la ampliación del subsistema Sala Situacional al SIGEP.

Contar con este módulo de reportes en la Sala Situacional conseguirá tener un control estricto y actualizado del estado de salud de los reclusos en los Sistemas Penitenciarios Venezolanos, lo que le facilitará a la Dirección Nacional de Servicios Penitenciarios a través de los reportes de información operativa e información histórica realizar análisis del comportamiento del indicador de salud en la Sala Situacional para la toma de decisiones en caso de que sea necesario.

Recomendaciones

- Debido a la amplitud de las funcionalidades de la Sala Situacional se recomienda hacer extensiva la experiencia a otros proyectos productivos que generan reportes en sus aplicaciones.
- Un generador de reportes se recomienda para la confección de los reportes de forma rápida, interactiva y con una amplia gama de alternativas para los usuarios.
- Se recomienda mantener la retroalimentación de la información mostrada en la Información Operativa e Información Histórica.
- Para futuras versiones del sistema se propone que la implementación de la lógica del negocio sea más sencilla y flexible.

Bibliografía

1. **Gómez, Juan Carlos Benavides y Yadira.** *Diseño e implementación de los módulos Desiciones y Egresos del Sistema Gestión Penitenciaria de la República Bolivariana de Venezuela.* Ciudad de la Habana : UCI, 2008.
2. **Vision.** *G.C.Sala Situacional.* 2005.
3. Sitio Oficial JAVA. *Sitio Oficial JAVA.* [En línea] [Citado el: 12 de enero de 2010.] <http://java.com/es>.
4. **WALLS, CRAIG Y BREINDENBACH, RYAN.** *Spring in Action, Second Edition.* s.l. : Manning Publications Co, 2008.
5. Hibernate. *Relational Persistence for Java & .NET.* [En línea] [Citado el: 22 de marzo de 2011.] <http://www.hibernate.org/>.
6. Eclipse. [En línea] <http://www.eclipse.org/>.
7. Diseño. [En línea] [Citado el: 20 de abril de 2011.] <http://www.info-ab.uclm.es/asignaturas/42579/pdf/01-Capitulo1.pdf>.
8. **Somerville, Ian.** *Software Engineering.* 2006.
9. i3j Gestión para Centros Penitenciarios. *i3j Gestión para Centros Penitenciarios.* [En línea] [Citado el: 22 de noviembre de 2010.] http://www.neotec.cc/solutions/i3j/index_es.html.
10. Sistema de Gestión Penitenciaria de Ecuador. *Sistema de Gestión Penitenciaria de Ecuador.* [En línea] Ministerio de Justicia, Derechos Humanos y Cultos. [Citado el: 20 de noviembre de 2010.] <http://www.minjusticia-ddhh.gov.ec>.
11. **Presman, Roger S.** *Ingeniería de Software. Un enfoque práctico.*
12. **PÉREZ RIVERO, IÓSEV y PIMENTEL GONZALEZ, LUIS ALBERTO.** *ArBaWeB: Arquitectura Base sobre la web.* Ciudad de La Habana : s.n., 2007.
13. **González, Rolando Alfredo Hernández León y Sayda Coello.** *El proceso de investigación científica.* Ciudad de la Habana : Universitaria, 2011.
14. **Basco, Sociedad Informatica del Gobierno.** *JMeter. Manual de usuario.*
15. *Visual Paradigm for UML.* [En línea] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/
16. JUnit.org Recursos para la Prueba de Desarrollo Impulsado. [En línea] 2009. <http://www.junit.org/>.

17. **Oracle Corporation.** Java EE at a Glance. [En línea] 2010. <http://java.sun.com/javaee/index.jsp>.
18. **JACOBSON, IVAR,BOOCH,GRADY Y RUMBAUGH,JAMES.** *El Proceso Unificado de Desarrollo.* Madrid : PEARSON EDUCACION, 2000.
19. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* 2011.
20. **GILBERT, D.** *The JFreechart Class Library.* 2004.
21. **JasperReport.** [En línea] 2007. <http://jasperforge.org/sf/projects/jasperreports>.
22. *Conferencia #5 Pruebas. Ingeniería de Software II.*
23. Sitio Web Java EE at a Glance. *Java EE at a Glance.* [En línea] 2010. <http://java.sun.com/javaee/index.jsp>.