

**Universidad de las Ciencias Informáticas**

**Facultad 2**



**Desarrollo del subsistema Sala Situacional del Sistema de  
Gestión Penitenciaria de la República Bolivariana de  
Venezuela para el régimen Extramuros del Sistema  
Penitenciario Venezolano**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Autor:** Yasiel Diaz Mendez

**Tutor(es):** Ing. Ramón E. González Peralta  
Ing. Valentín de León Torres

Ciudad de La Habana, mayo de 2011

## **DECLARACIÓN DE AUTORÍA**

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 firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Yasiel Diaz Mendez**

---

**Ing. Ramón E. González Peralta**

---

**Ing. Valentín de León Torres**

## DATOS DE CONTACTO

Ing. Ramón Enrique González Peralta

- Graduado de Ingeniero en Ciencias Informáticas en la UCI, año 2008.
- Ha trabajado en la universidad desde su graduación, como profesor de la Facultad 8, luego como Especialista de Calidad en Calisoft y actualmente se desempeña como profesor de la Facultad 2.
- Ha trabajado en el tema de la calidad desde el 3er año de su carrera, y en su vida profesional ha obtenido 1 publicación como autor único y 5 como coautor de otros trabajos.
- Formó parte del proyecto CICPC como asesor de calidad del mismo, participando en diversas liberaciones de productos y en la aceptación del SIIPOL v2.0.
- Ha sido tutor en cursos anteriores de tesis relacionadas con el tema de la calidad y participado en diferentes roles durante los tribunales de tesis de este tema.
- Ha participado en eventos como XI Jornadas de Innovación y Calidad de SW de España (JICS) 2009, 9na Semana Tecnológica de FORDES 2009, Fórum Provincial de ciencia y Técnica 2009 (obtuvo mención) y UCIENCIA 2010.

Ing. Valentín de León Torres

- Graduado en el 2010 de la carrera Ingeniería en Ciencias Informáticas de la Universidad de las Ciencias Informáticas.
- Se vinculó a la Universidad como profesor en septiembre del 2010.
- Actualmente recibe cursos de posgrado para su superación como profesional.
- Se ha vinculado a los proyectos productivos en el rol programador de interfaz de usuario y diseñador.

## **AGRADECIMIENTOS**

*A mi mamá Maria por ser la mejor madre del mundo siempre atenta y cariñosa.*

*A mi hermano Yosniel y a mis primos por ayudarme y darme todo su apoyo.*

*A mis tías y tíos por ser tan buenas conmigo y darme tanto cariño todos estos años.*

*A toda mi familia que ha sido la que me ha alumbrado el camino para lograr todos estos resultados y por estar siempre disponible cuando los he necesitado.*

*A todos los amigos y compañeros con los que he compartido estos 5 años.*

*A mis amigos de siempre.*

*A mis tutores Valentin y Ramón, muchas gracias por todo.*

*A todos los profesores del proyecto principalmente a Karina Sánchez y Roberto Granda por su ayuda y dedicación.*

*A todos aquellos que de una forma u otra me ayudaron a llegar aquí.*

*A la Revolución y a nuestro Comandante en jefe por brindarnos la posibilidad de cursar estudios y hacernos mejores personas.*

## **DEDICATORIA**

*Dedico este trabajo de diploma:*

*A mi madre que es lo que más yo quiero en el mundo y por ser mi guía e inspiración.*

*A mi familia por ser los seres humanos más especiales que conozco.*



## **RESUMEN**

En el presente trabajo se hace un estudio de algunas de las principales propuestas a nivel de software que existen actualmente para la gestión de los procesos penitenciarios, enfocándose en el tratamiento de la información para el apoyo a la toma de decisiones, la situación que presenta el régimen Extramuros del Sistema Penitenciario Venezolano y se desarrolla el subsistema Sala Situacional del Sistema de Gestión Penitenciaria (SIGEP) de la República Bolivariana de Venezuela para el régimen Extramuros del Sistema Penitenciario Venezolano.

El subsistema Sala Situacional del SIGEP para las sedes del régimen Extramuros sirve de base para la toma de decisiones estratégicas por parte de los funcionarios y directivos del sistema penitenciario venezolano ya que permite procesar la información y realizar un análisis del comportamiento operativo e histórico del sistema penitenciario en el régimen Extramuros.

Para su desarrollo se siguieron los principios arquitectónicos, de tecnologías y herramientas empleados en la realización del SIGEP como solución informática integral desde su primera versión.

El desarrollo del subsistema permite un mayor aprovechamiento del sistema en las sedes del régimen Extramuros y un mayor control de los procesos a diferentes niveles del Sistema Penitenciario Venezolano.

### **Palabras claves:**

Sistema de Gestión Penitenciaria, Sala Situacional, Toma de decisiones.

## ÍNDICE

INTRODUCCIÓN.....	1
CAPITULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción.....	5
1.2 Introducción al concepto de Sala Situacional.....	5
1.3 Sistemas Informáticos relacionados .....	6
1.3.1 Sistema Penitenciario del Gobierno de Panamá .....	6
1.3.2 Sistema de Gestión Penitenciaria de Ecuador ( <i>e-SIGPEN</i> ).....	6
1.3.3 Sistema Automatizado para el Control del Recluso ( <i>SACORE</i> ) Cuba.....	8
1.3.4 Sistema de Gestión Penitenciaria ( <i>SIGEP</i> ).....	8
1.4 El subsistema Sala Situacional del SIGEP .....	9
1.5 Tecnologías y herramientas utilizadas en el desarrollo .....	10
1.5.1 Metodología de desarrollo.....	10
1.5.2 Patrones de diseño de software .....	12
1.5.3 Java 2 Enterprise Edition (J2EE).....	13
1.5.4 Frameworks.....	13
Spring framework v2.0 .....	14
Hibernate framework v3.2.0 cr4 .....	15
Dojo Toolkit v0.4 .....	16
1.5.5 Entorno Integrado de Desarrollo.....	16
Eclipse v3.4 .....	16
Subclipse v1.0.1 .....	17
SDE v4.4.1 (Smart Development Environment).....	17
Spring IDE v2.0.....	17
Hibernate Tools.....	17
1.5.6 Bibliotecas.....	18
JFreechart v2.0.2 .....	18
JasperReport v1.0.5.....	19
1.5.7 Gestor de Base de Datos.....	19
Oracle 10g Standard Edition.....	19
1.6 Conclusiones.....	20
CAPITULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	21

2.1	Introducción.....	21
2.2	Análisis de la de la arquitectura del SIGEP .....	21
2.2.1	Visión general de la arquitectura.....	21
2.2.1.1	Definición de Subsistemas y Módulos .....	23
2.2.2	Análisis de la arquitectura del subsistema Sala Situacional .....	26
2.3	Características generales del sistema .....	27
2.4	Requerimientos funcionales.....	29
2.5	Requerimientos no funcionales .....	30
2.6	Descripción del sistema propuesto.....	31
2.6.1	Concepción general del sistema .....	31
2.6.2	Modelo de casos de uso del sistema .....	32
2.6.3	Descripción de casos de uso del sistema.....	32
2.6.3.1	Generar reporte individuos por niveles de supervisión .....	32
2.6.3.2	Generar reporte individuos con entradas tardías.....	38
2.6.3.3	Generar reporte individuos vinculados por modalidades .....	43
2.7	Conclusiones.....	49
<b>CAPITULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA .....</b>		<b>50</b>
3.1	Introducción.....	50
3.2	Actividades de diseño e implementación .....	50
3.2.1	Diseño de dominio.....	50
3.2.2	Diseño del modelo de datos.....	51
3.2.3	Diseño de la Capa de Presentación .....	52
3.2.4	Diseño de la Capa Lógica de Negocio .....	54
3.2.5	Diseño de la Capa de Acceso a Datos .....	56
3.2.6	Relaciones entre capas.....	58
3.2.7	Implementación de las entidades del dominio .....	58
3.2.8	Implementación de las interfaces de los managers.....	59
3.2.9	Implementación de la interfaz de la fachada.....	60
3.2.10	Implementación de la capa de acceso a datos .....	61
3.2.11	Implementación de los controladores .....	61
3.2.12	Implementación de las páginas JSP .....	62
3.2.13	Implementación de la lógica en el cliente .....	63



<b>3.3</b>	<b>Modelo de implementación.....</b>	<b>63</b>
3.3.1	Diagrama de componentes .....	63
	Capa de Presentación.....	64
	Capa Lógica de Negocio.....	65
	Capa de Acceso a Datos.....	66
	Relaciones entre capas.....	67
3.3.2	Modelo de despliegue .....	67
<b>3.4</b>	<b>Conclusiones.....</b>	<b>69</b>
	<b>CONCLUSIONES.....</b>	<b>70</b>
	<b>RECOMENDACIONES.....</b>	<b>71</b>
	<b>BIBLIOGRAFÍA.....</b>	<b>72</b>
	<b>ANEXOS.....</b>	<b>74</b>
	<b>Anexo 1. Diseño del CU-Generar reporte individuos por niveles de supervisión.....</b>	<b>74</b>
	Capa de Presentación.....	74
	Capa Lógica de Negocio.....	75
	Capa de Acceso a Datos.....	76
	Relaciones entre capas.....	77
	<b>Anexo 2. Diseño del CU-Generar reporte Individuos con entradas tardías .....</b>	<b>78</b>
	Capa de Presentación.....	78
	Capa Lógica de Negocio.....	79
	Capa de Acceso a Datos.....	80
	Relaciones entre capas.....	81
	<b>Anexo 3. Diagrama de Componentes del CU-Generar reporte individuos por niveles de supervisión .....</b>	<b>82</b>
	Capa de Presentación.....	82
	Capa Lógica de Negocio.....	83
	Capa de Acceso a Datos.....	84
	Relaciones entre capas.....	85
	<b>Anexo 4. Diagrama de Componentes del CU-Generar reporte Individuos con entradas tardías ...</b>	<b>86</b>
	Capa de Presentación.....	86
	Capa Lógica de Negocio.....	87
	Capa de Acceso a Datos.....	88
	Relaciones entre capas.....	89

## **INTRODUCCIÓN**

La nueva Constitución de la República Bolivariana de Venezuela (CRBV) lograda en el año 1998 en medio de una etapa de transformación con el impacto de la Revolución Bolivariana, así como la entrada en vigencia del código Orgánico Procesal, le dieron un vuelco al Sistema Penitenciario Venezolano, en donde la presunción de inocencia y Estado de libertad son sus premisas fundamentales.

En diciembre del año 2004 el presidente Hugo Chávez decreta en la República Bolivariana de Venezuela la Emergencia Penitenciaria, situación heredada de años de administraciones irresponsables que descuidaron muchos sectores de la vida venezolana.

Este esfuerzo del gobierno venezolano tuvo el propósito de crear condiciones para:

- a) Acometer un conjunto de acciones de impacto directo sobre las problemáticas más comprometedoras que presenta el Sistema Penitenciario Venezolano.
- b) La ejecución del Proyecto de Modernización del Sistema Penitenciario. (1)

En el año 2005 se realizó un censo nacional para conocer más detalladamente la situación judicial de la población penitenciaria. Este censo permitió un acercamiento a la situación en la que se encontraba el sistema penitenciario. Para el censo se elaboró una encuesta que recogía información sobre:

- Datos de los internos.
- Datos sobre los delitos.
- Datos sobre el proceso judicial.
- Datos sobre la condición física de la infraestructura penitenciaria.

El censo permitió contar con un acercamiento a la realidad del momento en que fue aplicado; sin embargo, no dio una solución definitiva a las necesidades en materia de gestión, información, comunicación y apoyo a la toma de decisiones de la Dirección Nacional de Servicios Penitenciarios (*DNSP*), además de carecer de mecanismos de actualización. (1)

Este censo arrojó como resultado una serie de limitaciones en el Sistema Penitenciario Venezolano que motivaron un grupo de acciones con el fin de revertir la situación existente. Una de las acciones llevadas a cabo es el proyecto de Humanización del Sistema Penitenciario, que además de integrar aspectos como

la atención a la salud de los individuos y asesoría especializada, incluye una solución tecnológica integral, conocida como Sistema de Gestión Penitenciaria (*SIGEP*).

El SIGEP, desarrollado por especialistas de la Universidad de las Ciencias Informáticas (*UCI*) en el marco del convenio de cooperación Cuba – Venezuela entre los meses de noviembre del año 2006 y julio del año 2008 en su primera fase, tiene como objetivo contribuir al control operativo, administrativo y estratégico del sistema penitenciario y además a garantizar el respeto a los derechos de los internos, su actividad de rehabilitación y reinserción en la sociedad.

En el Sistema Penitenciario Venezolano existen dos tipos de regímenes para clasificar la ubicación de la población penal, el régimen Intramuros que agrupa a los reclusos que permanecen dentro de instalaciones penitenciarias y hacia el que se centraron los esfuerzos de las dos primeras versiones del SIGEP (v1.0, v2.0) y el régimen Extramuros, que es aquella población que se encuentra fuera de los recintos penitenciarios, en formas alternativas de cumplimiento de pena como son los Centros de Residencia Supervisada (CRS), las Unidades Técnicas de Supervisión y Orientación (UTSO) o aquellas personas que están bajo libertad condicional, el cual aborda la última versión del SIGEP (v2.1), actualmente en despliegue por los centros penitenciarios venezolanos.

En la actualidad el proyecto de Humanización Penitenciaria en su quinta fase de desarrollo, aborda el SIGEP desde el punto de vista de procesos que aún no se han informatizado en ambos regímenes, pero con mayor atención en el régimen Extramuros, específicamente en la forma de procesar y visualizar la información con la que trabajan actualmente para así poder tomar decisiones estratégicas en función de la realidad objetiva de cada centro.

De forma general el régimen Extramuros se caracteriza actualmente en cuanto a la toma de decisiones en todos los niveles por carecer de:

- El establecimiento de estrategias reales que beneficien al sistema penitenciario y con ello a los reclusos en el régimen extramuros.
- La posibilidad de realizar una evaluación constante de la progresividad de los privados de libertad a nivel de centro.
- La posibilidad de controlar que se cumplan con los trámites correspondientes con el otorgamiento de los beneficios definidos en la ley penitenciaria venezolana.

Debido a la situación actual en los diferentes niveles: nacional, regional y por centros del Sistema Penitenciario en el régimen Extramuros, surge el siguiente **problema**: El proceso de análisis del comportamiento operativo e histórico del sistema penitenciario en el régimen Extramuros que se realiza actualmente no ayuda a la toma de decisiones estratégicas por parte de los funcionarios y directivos del sistema penitenciario venezolano.

El presente trabajo se propone dar solución al problema existente mediante el desarrollo de un sistema que sirva de soporte al trabajo operativo e histórico de la Sala Situacional<sup>1</sup>, que permita a los funcionarios y directivos del régimen disponer de una herramienta informática de ayuda a la toma de decisiones.

Por tanto se define como **objeto de estudio**: Gestión de la información histórica y operativa del Sistema Penitenciario Venezolano.

Dado el estado en que se encuentra la recepción y el análisis de la información que se genera en el Sistema Penitenciario en el régimen Extramuros, se plantea el siguiente **objetivo general**: Desarrollar un subsistema integrado al SIGEP que procese la información y realice un análisis del comportamiento operativo e histórico del sistema penitenciario en el régimen Extramuros, que sirva de base para la toma de decisiones estratégicas por parte de los funcionarios y directivos del sistema penitenciario venezolano.

Delimitando así el **campo de acción**: Gestión de la información histórica y operativa del régimen Extramuros del Sistema Penitenciario Venezolano.

Del diseño anterior se derivan los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación.
2. Diseñar la solución informática de apoyo a gestión de la información histórica y operativa del régimen Extramuros del Sistema Penitenciario Venezolano.
3. Desarrollar la solución informática de apoyo a gestión de la información histórica y operativa del régimen Extramuros del Sistema Penitenciario Venezolano.

Con el presente trabajo se pretende crear un subsistema que luego de integrado al SIGEP permita contar con información actualizada acerca de diferentes reportes del régimen Extramuros, que genere una serie

---

<sup>1</sup> Sistema Tecno-gerencial bajo un conjunto de normas y procedimientos claramente establecidos, en el marco de un concepto operacional integral, adaptables a situaciones rutinarias ya momentos de crisis, además que se realizan estudios comparativos de eventos pasados. Es considerado como un soporte a la toma de decisiones.

de indicadores, alertas y avisos en formas de tablas y gráficos, lo cual servirá de apoyo para la toma de decisiones estratégicas por parte de los funcionarios y directivos del sistema penitenciario venezolano.

El presente trabajo se encuentra dividido en tres capítulos. En el primero se tratan aquellos temas que constituyen la fundamentación teórica de la investigación a realizar. Se dan a conocer otros sistemas informáticos existentes que de una forma u otra están relacionados con el tema que se aborda y se realiza un estudio de las tendencias y tecnologías definidas para la solución.

En el segundo capítulo se da una descripción de la solución propuesta, definiéndose los requisitos que debe cumplir la misma, además se describe a profundidad la construcción de la propuesta de solución mediante los diversos artefactos que especifica el proceso de software utilizado.

Por último, en el capítulo tres se muestra la solución del diseño y la implementación a través de diagramas y fragmentos de código fuente, teniendo en cuenta las actividades de diseño e implementación definidas para el SIGEP.

## **CAPITULO 1: FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción**

En el presente capítulo se brinda una visión general de los aspectos relacionados con el proceso de análisis de información en las instituciones, y específicamente en los sistemas penitenciarios como organización, para ello se da una descripción de los principales conceptos asociados al dominio del problema. Se describen las herramientas y tecnologías seleccionadas para la solución del problema.

### **1.2 Introducción al concepto de Sala Situacional**

Las organizaciones han reconocido desde hace mucho, la importancia de administrar recursos principales tales como la mano de obra y las materias primas. Pero hay un recurso que se ha colocado en una posición de vanguardia: la información.

Para algunas instituciones la creación de un espacio donde se analicen determinados indicadores relacionados con las entidades que la conforman permite un mejor control de su funcionamiento, contribuye a la toma de decisiones y estrategias correctas según el escenario existente. De vital importancia es contar con una cultura de análisis de la información, puesto que influye directamente en las decisiones a tomar en situaciones normales o críticas.

A estos espacios de análisis se les ha dado entre otros nombres el de Sala Situacional. La Sala Situacional es aplicable a cualquier institución que requiera controlar el cumplimiento de sus objetivos. 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. Se utiliza además, para lograr el control sobre el cumplimiento de los objetivos; pero antes se debe realizar una selección de las informaciones que responderán a los procesos que se quieren controlar.

Estos sistemas por lo general, generan informes sobre el estado actual y sobre el comportamiento histórico de los indicadores de la entidad. Se generan gráficos y cuadros de análisis, facilitando así la comprensión de los datos mostrados y vigilan la evolución de los indicadores, emitiendo diferentes alertas de acuerdo al comportamiento de los mismos.

Los sistemas para la gestión penitenciaria, son organismos encargados de controlar y manejar el complejo proceso que representa el sistema penitenciario de un país. Generalmente derivan grandes volúmenes de

información que demandan un análisis exhaustivo y del cual depende la toma de decisiones estratégicas debido a la importancia que reviste este sector para la sociedad.

### **1.3 Sistemas Informáticos relacionados**

Existen diversos sistemas para la gestión penitenciaria, muchos países tienen el suyo propio. Estos sistemas tienen como objetivo facilitar el control de los reclusos y gestionar los procesos esenciales en los establecimientos penitenciarios relacionados con la población penal. También cuentan de una manera u otra, con un sistema de reportes, que les permite saber del estado de ciertos indicadores.

A continuación se muestra un análisis de algunos de los sistemas de gestión penitenciaria existentes en países de América Latina desde el punto de vista del tratamiento de la información:

#### **1.3.1 Sistema Penitenciario del Gobierno de Panamá**

Sitio web oficial: <http://www.sistemapenitenciario.gob.pa>

El sistema penitenciario panameño cuenta con una aplicación web destinada a la Dirección General del Sistema Penitenciario. Presenta seis módulos, entre los que se encuentra *Transparencia y Estadística*. El objetivo de este módulo es mostrar datos de interés para la organización, presenta los reportes siguientes:

- Población Extranjera, que es una presentación estadística de la población extranjera, por nacionalidad y centro penitenciario con actualización hasta el año 2009.
- Evadidos y prófugos, que es una presentación de las personas evadidas con actualización hasta el año 2009.
- Estadísticas, que es una presentación de la población penal de todos los Centros Penitenciarios desde el año 1991 al 2007; y una apartado con la población actual de los centros penitenciarios de la República de Panamá, con actualización hasta el año 2010. (2)

Este sistema tiene como objeto social la dirección general del sistema penitenciario panameño, no la gestión de los centros penitenciarios en el país.

Los reportes que se presentan requieren de una actualización manual y no abordan todos los indicadores necesarios en la solución del problema por lo que no se adapta al entorno tratado.

#### **1.3.2 Sistema de Gestión Penitenciaria de Ecuador (e-SIGPEN)**

Sitio web oficial: <http://www.minjusticia.gob.ec:8000/mjdh/admin/login.jsp>



Figura 1.1: Interfaz principal del **e-SIGPEN**

Sistema Nacional de Gestión Penitenciaria "**e-SIGPEN**", permite recopilar y controlar la información operativa que se genera en un Centro de Rehabilitación Social.

Entre sus módulos, cuenta con *Reportes* que contiene los tipos de reportes generales que muestra el sistema:

- Estadísticas
- Ingreso y egreso de internos
- Internos por celda
- Resultados del censo

La opción **Estadísticas** permite la generación de dos tipos de reportes estadísticos que son: por Delitos cometidos y Estado de las sentencias, estos reportes son generados en formato pdf.

**Ingresos y egresos de internos** genera el reporte de los Internos que ingresaron o *egresaron* del centro de Rehabilitación en un período dado.

En el caso de **Internos por celda** permite la generación de varios tipos de reportes en formato pdf del Centro de Rehabilitación.

En la opción **Resultados del Censo** permite descargar los documentos con los resultados del Censo Penitenciario generados por la empresa que realizó el Censo. (3)



Este sistema está dirigido a un Centro de Rehabilitación, no análogo a los Centros de Residencia Supervisada como pudiera parecer por el nombre, más bien a un centro interno, y aunque es posible generar algunos reportes no es aplicable al problema planteado pues no incluye los indicadores necesarios además de su forma de presentación.

### **1.3.3 Sistema Automatizado para el Control del Recluso (SACORE) Cuba**

El SACORE cuenta también con más de 200 reportes impresos, que permite un rápido acceso a la información y sirve de ayuda a la toma de decisiones por parte de los directivos del sistema. El módulo de reportes del SACORE es una aplicación web que permite la actualización constante del estado de los indicadores pero 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.

De forma general se puede decir que resulta muy útil la existencia de estos mecanismos informáticos de generación de reportes en estos sistemas, ya que por la cantidad de información que manejan resulta engorroso realizarlo sin la ayuda de un software, pero estos sistemas no cuentan con una buena organización de la información, ni con un sistema de avisos o alertas, que son muy importantes al prevenir de forma automática a los directivos de posibles irregularidades en el sistema.

Además es importante destacar que todos están ligados a las leyes judiciales y regulaciones propias del país para el cual fueron creados lo que imposibilita aplicarlo en otro.

### **1.3.4 Sistema de Gestión Penitenciaria (SIGEP)**

La República Bolivariana de Venezuela también cuenta con su propio sistema informático penitenciario puesto en práctica en tres tipos de entidades dentro del Sistema Penitenciario Venezolano, sede central, centros penales externos (CRS y UTSO) y centros penales internos. En este último tipo de entidad, el SIGEP cuenta con un subsistema llamado Sala Situacional con una serie de indicadores predeterminados para mostrar la información operativa e histórica.

Este subsistema Sala Situacional de la aplicación SIGEP-Intramuros, implementado para el mismo sistema legal que se necesita será la base del nuevo subsistema Sala Situacional para los centros del régimen Extramuros (CRS y UTSO).

#### **1.4 El subsistema Sala Situacional del SIGEP**

El Sistema de Gestión Penitenciaria (*SIGEP*), solución informática desarrollada por un equipo de desarrollo de la Universidad de las Ciencias Informáticas (UCI) con el objetivo de apoyar la gestión del Sistema Penitenciario Venezolano, surge como resultado de la necesidad urgente de modernizar e informatizar los procesos que ocurren en esta entidad.

El sistema desde su surgimiento ha cursado por una serie de versiones, debido a que su desarrollo ha sido de forma escalonada, añadiéndose funcionalidades en cada versión de acuerdo con las necesidades identificadas y no resueltas. En la versión 1.0 surge la necesidad de integrar al sistema un subsistema de soporte a la Sala Situacional con el mismo nombre, con el objetivo de mejorar la toma de decisiones estratégicas por parte de los directivos y funcionarios del Sistema Penitenciario Venezolano.

Esta Sala Situacional emite una serie de reportes en formato tabular y gráficos sobre un grupo de indicadores predeterminados que fueron identificados en la captura de requisitos realizada en la Dirección Nacional de Servicios Penitenciarios (DNSP). También emite avisos y alertas que son configurables y que previenen de situaciones extremas o de peligro, que de caer en ellas se pondría en riesgo el correcto funcionamiento de la vida de los individuos dentro del sistema.

Esta aplicación, conocida también como SIGEP-Intramuros debido al régimen penal para la cual está dirigida es la primera de cuatro que están actualmente en explotación, SIGEP-Sede Central, SIGEP- CRS y SIGEP-UTSO.

Las dos últimas, recién terminadas y puestas en prácticas, pertenecen al régimen Extramuros y carecen de un sistema de soporte a la Sala Situacional.

Como parte de la fase de desarrollo actual se solicitó por el cliente el desarrollo de la Sala Situacional para el Régimen Extramuros. Para su desarrollo se tendrá en cuenta la información generada por los Centros de Residencia Supervisada (CRS) y las Unidades Técnicas de Supervisión y Orientación (UTSO).

Con el desarrollo de este sistema informático se proveerá una herramienta que complementará el trabajo de la Sala Situacional anterior. Al finalizar el desarrollo del subsistema y su posterior integración al sistema, se contará con un instrumento capaz de brindar información de todo el sistema penitenciario en su conjunto, tanto del Régimen Intramuros, como del Extramuros facilitando el análisis y la toma de

decisiones por parte de los funcionarios de los establecimientos penitenciarios como de la Dirección Nacional de Servicios Penitenciarios Venezolano.

## **1.5 Tecnologías y herramientas utilizadas en el desarrollo**

La selección adecuada de herramientas y tecnologías a utilizar en el desarrollo de un software tiene estrecha relación con el tiempo de desarrollo y la calidad final del producto. Todas las herramientas utilizadas en la realización de SIGEP fueron definidas por el grupo de arquitectura así como las tecnologías y flujos de procesos. Es por ello que en este epígrafe no se definen los instrumentos utilizados sino que se realiza una identificación de cada uno de ellos y las ventajas que proporciona en el trabajo del subsistema a realizar. (4)

### **1.5.1 Metodología de desarrollo**

Metodología de desarrollo de software en ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Para el SIGEP se escogió desde sus inicios como metodología de desarrollo el Proceso Unificado Racional (RUP del inglés Rational Unified Process), por sus probadas ventajas basadas en la experiencia de años de usos por diferentes proyectos de desarrollo de software en el mundo entero.

RUP es un proceso que se caracteriza por ser:

- **Dirigido por casos de uso.** Los casos de uso describen los requisitos funcionales del sistema desde la perspectiva del usuario y se usan para determinar el alcance de cada iteración y el contenido de trabajo de cada persona del equipo de desarrollo.
- **Centrado en la arquitectura.** La arquitectura permite ganar control sobre el proyecto para manejar su complejidad y controlar su integridad. Hace posible la reutilización a gran escala y provee una base para la gestión del proyecto.
- **Iterativo e incremental.** Se divide en 4 fases: Inicio, Elaboración, Construcción y Transición, y cada una de ellas se divide en iteraciones. En cada iteración se trabaja en un número de disciplinas haciendo énfasis en algunas de ellas. Las disciplinas propuestas por RUP son: Modelado del negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas, entre otras. (5)

Cada iteración añade funcionalidades al producto de software o mejora las existentes.

RUP indica que al inicio del proyecto se realice una adecuación de cada flujo de trabajo de manera que se produzcan solo los artefactos y se realicen las actividades que tienen un propósito dentro del proyecto. Para el SIGEP se llevó a cabo desde su primera fase y está debidamente documentado. (4)

### ***Notaciones o lenguajes de modelado de software***

El modelado de sistemas software es una técnica para tratar con la complejidad inherente a estos sistemas. El uso de modelos ayuda al ingeniero de software a "visualizar" el sistema a construir. Además, los modelos de un nivel de abstracción mayor pueden utilizarse para la comunicación con el cliente. (6)

**Lenguaje Unificado de Modelado:** (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se aplica en el desarrollo del SIGEP entregando gran variedad de formas para dar soporte a RUP.

### ***Herramienta de Modelado***

#### **Visual Paradigm v6.4**

**Sitio web oficial:** <http://www.visual-paradigm.com>

Las herramientas de modelado son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

En el SIGEP se define al Visual Paradigm como la herramienta a utilizar para el modelado del software. Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

### 1.5.2 Patrones de diseño de software

Los patrones de diseño proponen una manera de reutilizar la experiencia de los desarrolladores, para ello se clasifican y se describen formas de solucionar problemas que ocurren frecuentemente en el desarrollo del software.

Según la escala o nivel de abstracción:

- **Patrones de arquitectura:** Aquéllos que expresan un esquema organizativo estructural fundamental para sistemas de software.
- **Patrones de diseño:** Aquéllos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de software.
- **Dialectos:** Patrones de bajo nivel específicos para un lenguaje de programación o entorno concreto.

Además de los patrones ya vistos actualmente existen otros patrones como el siguiente:

- **Interacción:** Son patrones que nos permiten el diseño de interfaces web. (7)

En el diseño y la implementación realizada del sistema se utilizaron diferentes patrones de diseño para brindar y ayudar a desarrollar una solución más robusta y escalable. De los patrones básicos pertenecientes a los patrones de asignación de responsabilidades en inglés (General Responsibility Assignment Software Patterns, GRASP), se utilizaron de manera general los patrones:

- Bajo acoplamiento
- Alta cohesión
- Experto

De los patrones propuestos por el grupo de los cuatros, en inglés (Gang of Four), se utilizó el patrón Facade.

- **Facade (Fachada):** Simplifica el acceso a un conjunto de objetos proporcionando uno, llamado *fachada*, que los clientes pueden usar para comunicarse con el conjunto. Reduce el número de objetos con los que tiene que interactuar un cliente proporcionando un menor acoplamiento y facilitando el cambio de componentes sin afectar a sus clientes.

Otros patrones presentes en la propuesta de solución de software son (4):

- **Data Access Object (DAO):** Centraliza todo el acceso a datos en una capa independiente, aislándolo del resto de la aplicación. Sus principales beneficios son que reduce la complejidad de

los objetos de negocio al abstraerlos de la implementación real de la comunicación con la fuente de datos y que permite una migración más fácil de fuente de datos.

- **Controller (Controlador):** Este patrón propone asignar la responsabilidad de controlar el flujo de eventos de un sistema, a clases específicas llamadas *controladores*. Los controladores no ejecutan las tareas sino que las delegan en otras clases, con las que mantiene un modelo de alta cohesión.
- **Model-View-Controller (Modelo-Vista-Controlador, MVC):** Divide una aplicación en tres componentes, el modelo, las vistas y los controladores. El modelo contiene la funcionalidad y los datos del sistema, las vistas muestran información al usuario y los controladores manejan la interacción del usuario. Las vistas y controladores comprenden la interfaz de usuario. MVC permite crear diferentes vistas para el mismo modelo incluso en tiempo de ejecución.

### 1.5.3 Java 2 Enterprise Edition (J2EE)

**Sitio web oficial:** <http://java.sun.com/j2ee/overview.html>

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 del comportamiento de la aplicación automáticamente, evitando una 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. (7)

### 1.5.4 Frameworks

La palabra inglesa "framework" define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. (8)

Un framework es un conjunto de componentes con interfaces bien definidas que interactúan entre sí para cumplir una tarea. No tiene funcionalidades de una aplicación específica, sino que las aplicaciones se construyen sobre ellos. Es una mini arquitectura reusable que provee una estructura y comportamiento genérico. En el desarrollo de aplicaciones empresariales similares al SIGEP, el uso de frameworks se ha convertido en algo imprescindible porque implica ahorro en tiempo de diseño y código puesto que implementa un conjunto de patrones de diseño, lo que permite su reutilización como componentes de

software y a su vez suelen implementar las partes más engorrosas y difíciles del dominio del problema permitiendo que el desarrollador se concentre en implementar las tareas propias de la aplicación. Existen varias clasificaciones para los frameworks de acuerdo a diferentes criterios. A continuación se muestra una de estas clasificaciones:

- **Frameworks de aplicación:** Proveen un amplio rango de funcionalidades típicamente usadas en una aplicación. Interviene en varias capas de la aplicación como Interfaz de Usuario, Acceso a Datos, etcétera.
- **Frameworks de dominio:** Proporciona funcionalidades para un dominio específico de aplicación.
- **Frameworks de soporte:** Se dirigen a dominios muy específicos dentro de una aplicación como manejo de memoria, reportes, etcétera.

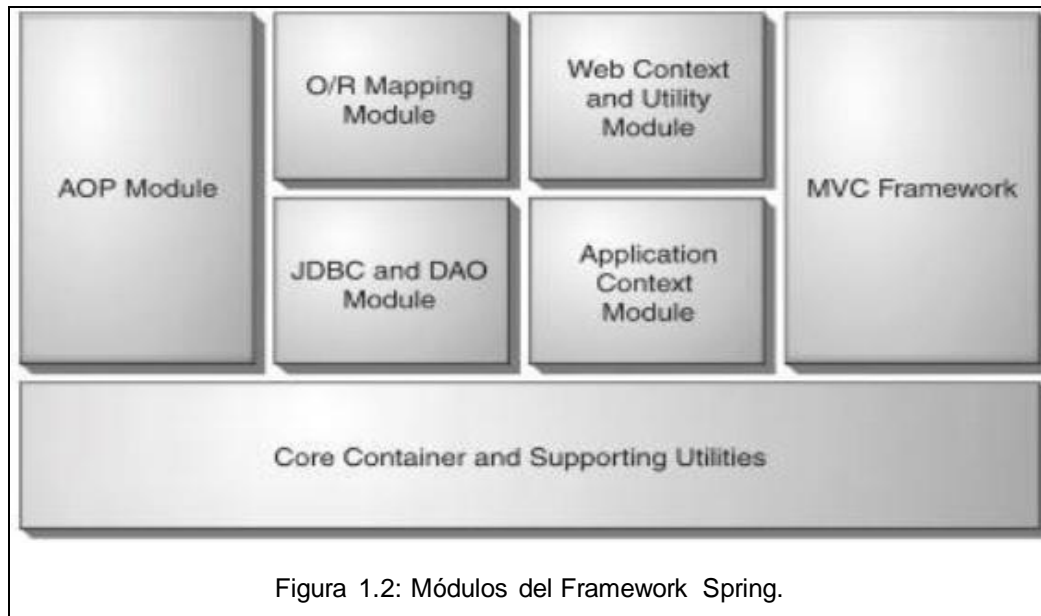
Sobre la plataforma J2EE existen un conjunto amplio de frameworks que utilizan e integran las APIs que esta brinda. La mayoría de estos frameworks constituyen referencias para otras plataformas. En el desarrollo de la arquitectura base del SIGEP se utilizó el framework de aplicación Spring que al igual que el framework de soporte Hibernate son muy populares, por sus múltiples ventajas, en el desarrollo de aplicaciones web dentro de la plataforma J2EE. (7)

### ***Spring framework v2.0***

**Sitio web oficial:** <http://www.springsource.org>

El Spring Framework (también conocido simplemente como Spring) es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java. La primera versión fue escrita por Rod Johnson, quien lo lanzó primero con la publicación de su libro *Expert One-on-One Java EE Design and Development* (Wrox Press, octubre 2002). También hay una versión para la plataforma .NET, Spring .NET. (9)

En el SIGEP los módulos más utilizados son: *Spring Core* ya que como núcleo del framework constituye la base del sistema, *Spring MVC* que provee su propia implementación del modelo MVC, *Spring ORM* que provee a la aplicación de una forma de construir la capa de acceso datos basándose en el patrón DAO e integrándose con el framework Hibernate y *Spring AOP* que provee de un amplio soporte para la programación orientada a aspectos. (7)



***Hibernate framework v3.2.0 cr4***

**Sitio web oficial:** <https://www.hibernate.org/>

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Es software libre, distribuido bajo los términos de la licencia GNU LGPL.

Hibernate es una capa de persistencia objeto/relacional y un generador de sentencias SQL de código abierto. Con este framework, se puede diseñar objetos persistentes que podrán incluir relaciones, colecciones, polimorfismo, y un gran número de tipos de datos. Con Hibernate se logra una abstracción total del gestor de base de datos a utilizar. De una manera muy rápida y optimizada se puede realizar consultas contra cualquiera de los entornos soportados: Oracle, DB2, MySQL, etc. 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. Con HQL se obtienen los resultados de una consulta de forma objetual lo que permite desarrollar la capa de acceso a dato más rápido e intuitivo para el programador. (10)



### **Dojo Toolkit v0.4**

**Sitio web oficial:** <http://www.dojotoolkit.org/>

Dojo Toolkit es un framework que contiene APIs y widgets (controles) para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX. Contiene un sistema de empaquetado inteligente, los efectos de UI, drag and drop APIs, widget APIs, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX. Resuelve asuntos de usabilidad comunes como pueden ser la navegación y detección del navegador, soportar cambios de URL en la barra de URLs para luego regresar a ellas (bookmarking), y la habilidad de degradar cuando AJAX/JavaScript no es completamente soportado en el cliente. Es conocido como "la navaja suiza del ejército de las bibliotecas Javascript". Proporciona una gama más amplia de opciones en una sola biblioteca JavaScript y es compatible con navegadores antiguos.

#### **1.5.5 Entorno Integrado de Desarrollo**

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI, Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

### **Eclipse v3.4**

**Sitio web oficial:** <http://www.eclipse.org/>

Eclipse IDE cuenta con un compilador de Java interno y un modelo completo de los archivos fuente de Java. Esto permite técnicas avanzadas de refactorización y análisis de código. El IDE también hace uso de un espacio de trabajo, en este caso un grupo de metadata en un espacio para archivos plano, permitiendo modificaciones externas a los archivos en tanto se refresque el espacio de trabajo correspondiente, emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Adicionalmente Eclipse puede extenderse usando otros lenguajes de programación como son C/C++ y Python, además de trabajar con lenguajes para procesamiento de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos. La arquitectura plugin permite escribir cualquier extensión deseada en el ambiente. (11)

En el SIGEP los plug-ins más utilizados son los siguientes:

### ***Subclipse v1.0.1***

**Sitio web oficial:** <http://subclipse.tigris.org/>

Subclipse es un plugin para Eclipse que adiciona integración para el control de versiones (Subversion, específicamente), que es el que se utiliza en el proyecto SIGEP permitiendo operaciones de sincronización, actualización, entre otras. Permite bloqueos a recursos para que otros usuarios no puedan modificarlo.

### ***SDE v4.4.1 (Smart Development Environment)***

**Sitio web oficial:** <http://www.visual-paradigm.com/product/sde/ec/>

Este plug-in es una herramienta CASE<sup>2</sup> que permite la integración de la herramienta Visual Paradigm con el IDE Eclipse. Soporta todos los diagramas UML generados a lo largo del desarrollo del software.

También permite hacer ingeniería inversa para obtener diagramas de clases a partir del código Java y viceversa. Fundamentalmente mantiene la sincronización de las clases con los modelos, además de generar documentación. (7)

### ***Spring IDE v2.0***

**Sitio web oficial:** <http://www.springide.org>

Spring IDE es un plugin que sirve como interfaz de usuario gráfica para la configuración de los archivos usados por Spring Framework. Permite el completamiento de etiquetas, valores de atributos y elementos en estos archivos de configuración. (11)

### ***Hibernate Tools***

**Sitio web oficial:** <http://www.tools.hibernate.org/>

Hibernate Tools es una herramienta de mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Es un motor de persistencia, que es el componente software encargado de traducir entre objetos y registros.

---

<sup>2</sup> Aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software.

Facilita el trabajo con el framework Hibernate ya que aporta herramientas de fácil utilización para los programadores: un editor de HQL y un editor de mapeo de XMLs, que posibilita el auto-completamiento y el resaltado de sintaxis. Hibernate Tools es una herramienta de ingeniería inversa de bases de datos que puede generar las clases del modelo de dominio y los archivos de mapeo. También cuenta con varios asistentes para la generación rápida de configuraciones de Hibernate. (7)

### **1.5.6 Bibliotecas**

Una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es 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 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. (11)

#### **JFreechart 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.

JFreeChart incluye:

- Amplio conjunto de funciones
- Bien documentada
- Admite una amplia gama de tipos de gráficos
- Un diseño flexible que es fácil de extender y objetivos tanto del lado del servidor y las aplicaciones del lado del cliente.
- Soporte para los tipos de salida, incluyendo los componentes Swing, archivos de imagen (incluyendo PNG y JPEG) y gráficos vectoriales en formato de archivo (incluyendo PDF, EPS y SVG);

JFreeChart es "fuente abierta" o, más específicamente, el software libre. Se distribuye bajo los términos de la GNU Lesser General Public Licence (LGPL), que permite su uso en aplicaciones propietarias.

### **JasperReport v1.0.5**

JasperReport es la mejor herramienta de código libre, específicamente para generar reportes en Java y para este lenguaje fue creada. Está completamente escrita en Java y se puede utilizar en variedad de aplicaciones Java, incluyendo J2EE o aplicaciones Web, para generar contenido dinámico. Crea ricas presentaciones o diseños en la pantalla, para la impresora o para archivos en formato PDF, HTML, RTF, XLS, CSV y XML.

JasperReports trabaja en forma similar a un compilador y a un intérprete. El usuario diseña el reporte usando XML, define completamente el reporte, describiendo donde colocar texto, imágenes, líneas, rectángulos, cómo adquirir los datos, como realizar ciertos cálculos para mostrar totales, etc.

Este archivo fuente XML debe ser compilado para obtener un reporte real. El diseño del reporte es leído por JasperReport de un archivo XML o de .jasper. Por este problema los desarrolladores crearon una herramienta (iReport) que facilita generación gráfica del diseño del reporte.

Esta librería posee características apreciadas por todos como:

- Generación de archivos como pdf, HTML, Microsoft Excel, CSV, XML, etc. Los informes definidos pueden incluir gráficas tales como empanada, barra, apiló serie de la barra, de la línea, del área, del diagrama de la dispersión, de la burbuja, y de tiempo, todas proporcionadas por la biblioteca JFreeChart que genera gráficos de calidad profesional.
- Los datos pueden ser obtenidos mediante JDBC, JavaBeans, XML, Hibernate, etc. Varias de estas fuentes pueden ser combinadas en caso de ser necesario.

Estas librerías son las utilizadas en las soluciones SIGEP para crear los reportes gráficos de la solución.

### **1.5.7 Gestor de Base de Datos**

#### **Oracle 10g Standard Edition**

**Sitio web oficial:** <http://www.oracle.com>

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado por Oracle Corporation. (12)

Oracle 10g está definido como el gestor de base de datos a utilizar para el desarrollo y despliegue del SIGEP debido a que ofrece una excepcional disponibilidad, escalabilidad, fiabilidad y seguridad.

Además proporciona un adecuado soporte para la réplica de los datos, necesario en el SIGEP, para la comunicación desde los servidores de los centros penitenciarios al servidor del Centro de Datos, que

recoge toda la información del sistema a nivel nacional. La infraestructura del SIGEP en tiempo de despliegue identifica que en el Centro de Datos se utilice Oracle Enterprise Edition v10.2.0.3 y en cada centro penitenciario Oracle Standard Edition 10.2.0.3 que es más ligero y cuya licencia es menos costosa. La principal desventaja de la utilización de Oracle como herramienta son los elevados precios de las licencias de software y del soporte técnico, lo que lo hace un gestor típico de sistemas informáticos de grandes compañías o instituciones gubernamentales. A ello se suma los elevados requisitos de hardware que tiene. (4)

### **1.6 Conclusiones**

Se puede concluir que con el desarrollo de este capítulo:

- Se han introducido conceptos indispensables para la comprensión del proceso de análisis de la información mediante un sistema de soporte a la Sala Situacional.
- Se analizaron sistemas similares relacionados identificando al subsistema Sala Situacional del SIGEP-Intramuros como referencia para la solución a desarrollar.
- Se describieron las herramientas y tecnologías a utilizar en el desarrollo de la solución, las cuales fueron definidas desde la fase I del proyecto por el equipo de desarrollo.

## **CAPITULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

### **2.1 Introducción**

El presente capítulo constituye el resultado del estudio de los procesos que tienen lugar para el análisis de la información en los centros penales del régimen Extramuros. 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.

### **2.2 Análisis de la de la arquitectura del SIGEP**

#### **2.2.1 Visión general de la arquitectura**

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

La arquitectura del SIGEP está regida por una arquitectura de tres capas lógicas fundamentales bien definidas:

#### **Capa de acceso a datos**

La interacción del negocio con la capa de persistencia se realiza mediante el uso de interfaces. El término de Objeto de Acceso a Datos o en inglés, Data Access Object (DAO) es ampliamente usado en el desarrollo de software.

Los DAOs encapsulan la persistencia de los objetos de dominios, proveen la persistencia de los objetos transitorios y las actualizaciones de los objetos existentes en la base de datos. Las implementaciones de los DAOs están disponibles para los objetos (típicamente para los objetos de negocio) haciendo uso de la inyección de dependencias con los objetos de negocio y las instancias de los DAOs, configurada en el contenedor de inversión de control de Spring Framework.

#### **Capa de servicios de negocio**

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.

### **Capa de presentación**

La capa de presentación descansa sobre una capa de servicios de negocio. Esto significa que esta capa será fina y no contendrá lógica de negocio, sino simplemente lo concerniente a aspectos de presentación, por ejemplo, el código para manipular las interacciones web. Esta capa puede variar en complejidad en el transcurso del desarrollo de la aplicación, sin embargo ninguna de las capas inferiores debe ser afectada por estos cambios.

### **Capa Web**

En capa Web se utiliza Spring MVC como framework MVC para manejar el flujo de datos entre el cliente y las capas inferiores.

Esta capa web 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 contienen los datos resultantes de la ejecución de la lógica de negocio, los cuales deberían ser 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, PDF, documentos de Excel, 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.

Esta combinación de estos tres objetos, es la aplicación del patrón arquitectónico Modelo Vista – Controlador o MVC (Model View Controller).

El cliente interactúa con esta capa directamente utilizando el navegador Web a través de peticiones URL. En el cliente se formará dinámicamente una pequeña capa de presentación que estará formada por

código html y JavaScript. Esta capa interactúa directamente mediante peticiones basadas en la tecnología Ajax con los componentes necesarios del servidor creados bajo la tecnología de SpringMVC. (8)

### **2.2.1.1 Definición de Subsistemas y Módulos**

#### **Subsistema**

Un subsistema se refiere a un conjunto de módulos que por razones de similitud o de perseguir objetivos comunes, son agrupados.

#### **Estructura de un Subsistema.**

El SIGEP está conformado por subsistemas que contienen varios módulos que tienen similitudes. Esta estructura de paquetes permite mantener una buena organización dentro de las diferentes capas. En la Figura 2.1 se expone esta estructura.

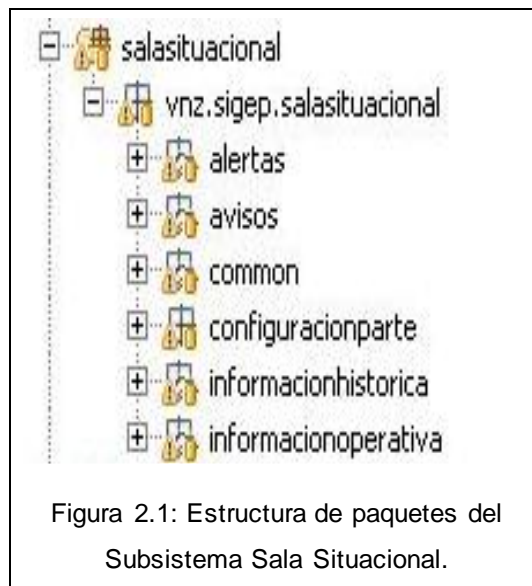


Figura 2.1: Estructura de paquetes del Subsistema Sala Situacional.

Para cada subsistema existe un módulo común donde se gestionan las funcionalidades comunes nombrado “**common**” ver Figura 2.5. Los demás paquetes encapsulan los módulos pertenecientes al subsistema.



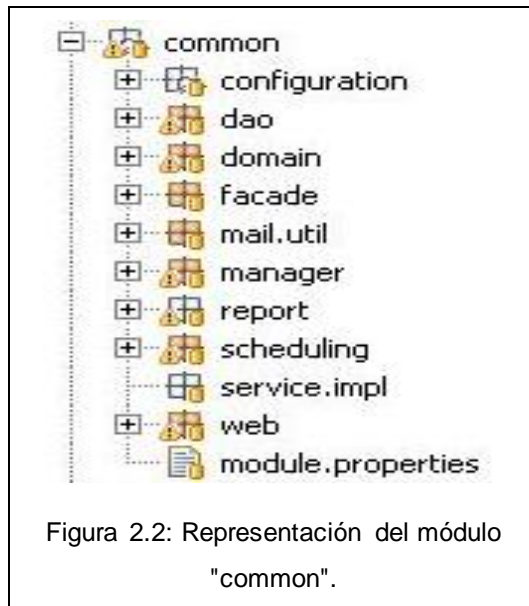


Figura 2.2: Representación del módulo "common".

### **Módulo**

Un módulo encapsula un conjunto de funciones que debe realizar el sistema, las cuales son agrupadas por tener características muy similares y se definen en la etapa de diseño.

### **Estructura de un módulo en el SIGEP**

La estructura interna de cada módulo está compuesta por paquetes (figura 2.3). Algunos de estos paquetes contienen en su interior la implementación correspondiente a la capa de presentación, capa de lógica de negocio y capa de acceso a datos.

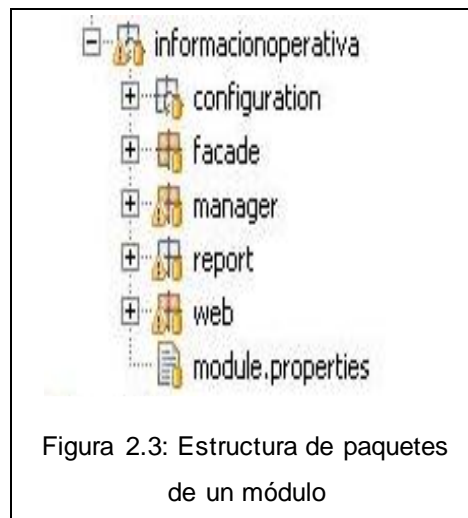


Figura 2.3: Estructura de paquetes de un módulo

Algunos módulos están compuestos por los siguientes paquetes:

**configuration:** Contienen los archivos que tienen que ver con la configuración del módulo, los “Application-Context” de Spring Framework, los archivos utilizados para la internacionalización, ficheros de propiedades “.properties” y cualquier otro destinado a estos fines.

**manager:** Se encuentran las interfaces de los managers de negocio.

**manager.impl:** Se encuentran las implementaciones de los managers de negocio.

**facade:** Se encuentran las interfaces de las fachadas de negocio.

**facade.impl:** Se encuentran las implementaciones de las fachadas de negocio.

**web:** Se encuentran los controladores de la capa de presentación.

En las siguientes figuras se muestran las estructuras de las “**Vistas**” del subsistema Sala Situacional, ubicadas en la capa de presentación.

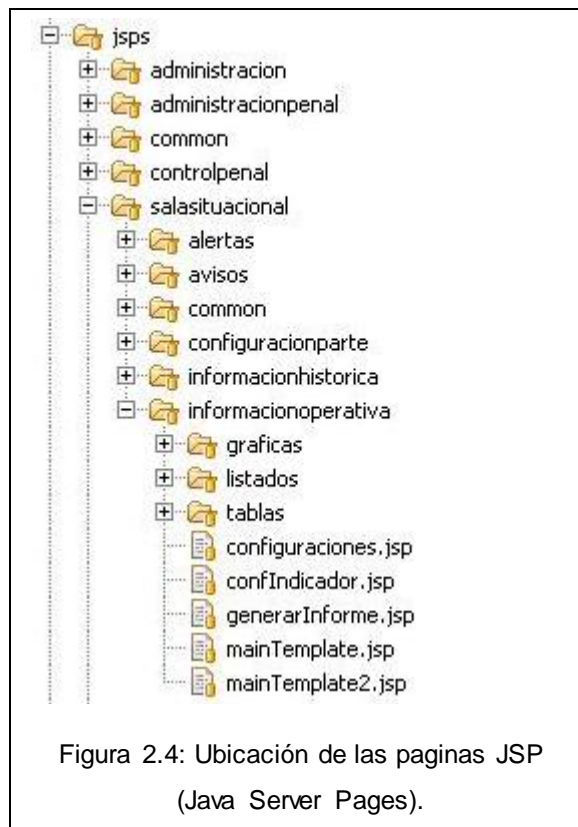


Figura 2.4: Ubicación de las paginas JSP  
(Java Server Pages).

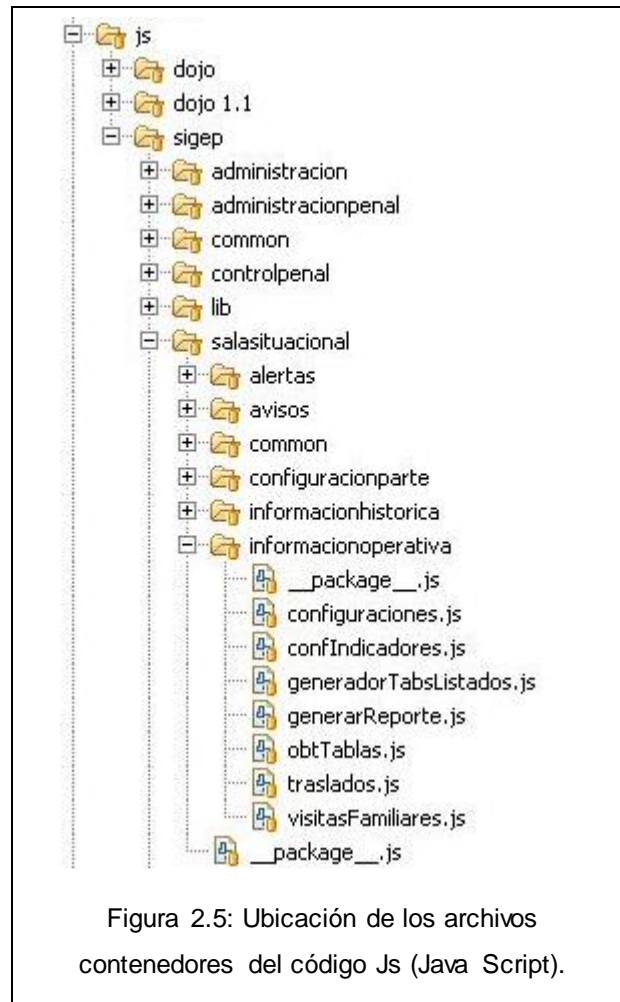


Figura 2.5: Ubicación de los archivos contenedores del código Js (Java Script).

### **2.2.2 Análisis de la arquitectura del subsistema Sala Situacional**

El subsistema Sala Situacional para el Régimen Extramuros cuenta con una arquitectura de tres capas, asumida en el SIGEP, dividida de la forma siguiente:

#### **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.

- Información tabular: en este tab es donde se llama a la página en la cual se muestra la información del reporte seleccionado en forma de tabla.
- Información gráfica: en este tab es donde se llama a la página en la cual se muestra la información del reporte seleccionado en forma de gráfica siendo esta lineal.
- Configuración: en este tab es donde se llama a la página en la cual se configuran los reportes de información histórica, teniendo en cuenta intervalos de fecha, frecuencia del intervalo, como tipo de intervalo, dígame anual, mensual, semanal, diario.

Creación de componentes para la reutilización de código como son widget tanto para las tablas como para las gráficas. Desarrollo del mapeo entre controladores y vistas usando Ajax.

### **Capa de 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.

- Definición de fachada para interactuar con la capa de negocio.
- Creación de una estructura para la generación de reportes
- Implementación de los manager

### **Capa de Datos**

Es la encargada del manejo de datos persistentes, forma parte de la misma el servidor de base de datos como encargado de almacenamiento de datos, se comunica con la capa de lógica de negocio desde donde recibe las solicitudes de almacenar o recuperar información. (12)

## **2.3 Características generales del sistema**

El Sistema de Gestión Penitenciaria (SIGEP), dará respuesta a las necesidades de gestión, información y apoyo a la toma de decisiones tomando como soporte el subsistema de Sala Situacional de la Dirección General de Servicios Penitenciarios (DNSP), 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.

### **Sala Situacional**

El subsistema concentra las salidas informativas que se generan en el nivel operativo mostrándolas a través de tablas, gráficos, listados y partes configurables, sirviendo de apoyo al trabajo de la Sala Situacional. De igual manera a través de otros tipos de salidas como avisos, alertas y el compendio de la información histórica, se convierte 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. Dentro de este subsistema se ubican los siguientes módulos:

- **Información Operativa:** Mostrará un conjunto organizado de datos, que reflejarán de modo general el estado del sistema penitenciario venezolano. Estos datos mostrarán el estado de la población penal agrupados por diferentes criterios. También se mostrará el comportamiento de los principales movimientos fuera de los establecimientos (traslados y salidas transitorias), los ingresos y egresos, las visitas familiares e institucionales y sobre las presentaciones a tribunales.
- **Información Histórica:** Observará el comportamiento en el tiempo sobre las informaciones que se obtienen del sistema penitenciario mediante 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.
- **Avisos:** Mostrará informaciones en forma de avisos. Estos avisos tributarán a la dirección penitenciaria a prevenir posibles irregularidades en el sistema penitenciario venezolano. Se expresarán estos avisos a través de un conjunto de parámetros que se hayan cumplido. De no ser atendidos estos avisos, la situación sobre la que previenen puede convertirse en una alerta.
- **Alertas:** Permitirá conocer y controlar irregularidades que pueden ocurrir en el sistema penitenciario a través de reportes. Estos reportes están clasificados como alertas en el sistema informático y van a contener el número de internos afectados en la irregularidad mostrada. Las alertas se dividen en niveles para diferenciar los niveles de riesgo que representan. Para mostrar estos niveles se utilizarán colores, que reflejarán la gravedad de las alertas (Amarillo, Naranja y Rojo).

Dentro del subsistema se brindarán diferentes opciones para configurar los períodos correspondientes a la información operativa, las alertas, los avisos y los reportes para evaluar el comportamiento histórico. De igual manera se podrán configurar los estados de las alertas y de los avisos.

Este trabajo se enfoca en la integración del subsistema Sala Situacional y en las necesidades de los indicadores de los establecimientos del régimen Extramuros (CRS y UTSO) en el nivel operativo. Se describirán funcionalidades específicas del régimen abordado.

### **2.4 Requerimientos funcionales**

El sistema que se propone construir, que brinde de soporte al trabajo operativo e histórico de la Sala Situacional en el régimen Extramuros, deberá permitir:

#### *R1. Generar reporte individuos por niveles de supervisión*

- 1.1 Mostrar la cantidad de individuos por niveles de supervisión de forma tabular para información operativa.
- 1.2 Mostrar la cantidad de individuos por niveles de supervisión de forma gráfica para información operativa.
- 1.3 Mostrar los individuos por niveles de supervisión en forma de listados para información operativa.
- 1.4 Mostrar la cantidad de individuos por niveles de supervisión de forma tabular para información histórica.
- 1.5 Mostrar la cantidad de individuos por niveles de supervisión de forma gráfica para información histórica.
- 1.6 Mostrar los individuos por niveles de supervisión de acuerdo a la configuración especificada para información histórica.

#### *R2. Generar reporte individuos con entradas tardías*

- 2.1 Mostrar la cantidad de individuos con entradas tardías de forma tabular para información operativa.
- 2.2 Mostrar la cantidad de individuos con entradas tardías de forma gráfica para información operativa.
- 2.3 Mostrar los individuos con entradas tardías en forma de listados para información operativa.
- 2.4 Mostrar la cantidad de individuos con entradas tardías de forma tabular para información histórica.

- 2.5 Mostrar la cantidad de individuos con entradas tardías de forma gráfica para información histórica.
- 2.6 Mostrar los individuos con entradas tardías de acuerdo a la configuración especificada para información histórica.

### *R3. Generar reporte individuos vinculados por modalidad*

- 3.1 Mostrar la cantidad de individuos vinculados por modalidad de forma tabular para información operativa.
- 3.2 Mostrar la cantidad de individuos vinculados por modalidad de forma gráfica para información operativa.
- 3.3 Mostrar los individuos vinculados por modalidad en forma de listados para información operativa.
- 3.4 Mostrar la cantidad de individuos vinculados por modalidad de forma tabular para información histórica.
- 3.5 Mostrar la cantidad de individuos vinculados por modalidad de forma gráfica para información histórica.
- 3.6 Mostrar los individuos vinculados de acuerdo a la configuración especificada para información histórica.

## **2.5 Requerimientos no funcionales**

- *Apariencia o interfaz externa.*

El sistema deberá poseer una interfaz web sencilla, en correspondencia con el SIGEP al cual se integra, amigable y sugerente, que permita a los usuarios interactuar con facilidad con el mismo.

- *Usabilidad.*

El sistema está concebido para un número limitado de usuarios, los funcionarios responsables del soporte a la Sala Situacional del régimen Extramuros en los distintos centros. Debe ser fácil de usar aún para personas que no sean expertas en el uso de computadoras.

- *Rendimiento*

Debido a que se trata de una aplicación cliente/servidor debe poseer un rápido nivel de respuesta.

- *Portabilidad*

El sistema debe ser multiplataforma.

- *Seguridad*

Para poder acceder al sistema el usuario deberá estar registrado en el SIGEP y debe tener privilegios de acceso al mismo.

Cada usuario en el SIGEP tiene un rol específico de acuerdo a su trabajo.

Se debe garantizar que los usuarios sólo posean acceso a la información con derecho a ver o a manipular.

- *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 debe integrarse al SIGEP como un subsistema por lo que deberá ser implementada utilizando su arquitectura, bajo las mismas tecnologías y diseño.

## **2.6 Descripción del sistema propuesto**

### **2.6.1 Concepción general del sistema**

El sistema a desarrollar formará parte del SIGEP en su versión para el régimen Extramuros del Sistema Penitenciario Venezolano. Este subsistema del SIGEP llamado Sala Situacional debido a que sirve de soporte a esta herramienta de toma de decisiones será utilizada por el funcionario encargado de esta actividad dentro de cada una de las sedes penitenciarias del régimen (CRS y UTSO).

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:

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.

Tabla 2.1 Actor del sistema



**2.6.2 Modelo de casos de uso del sistema**

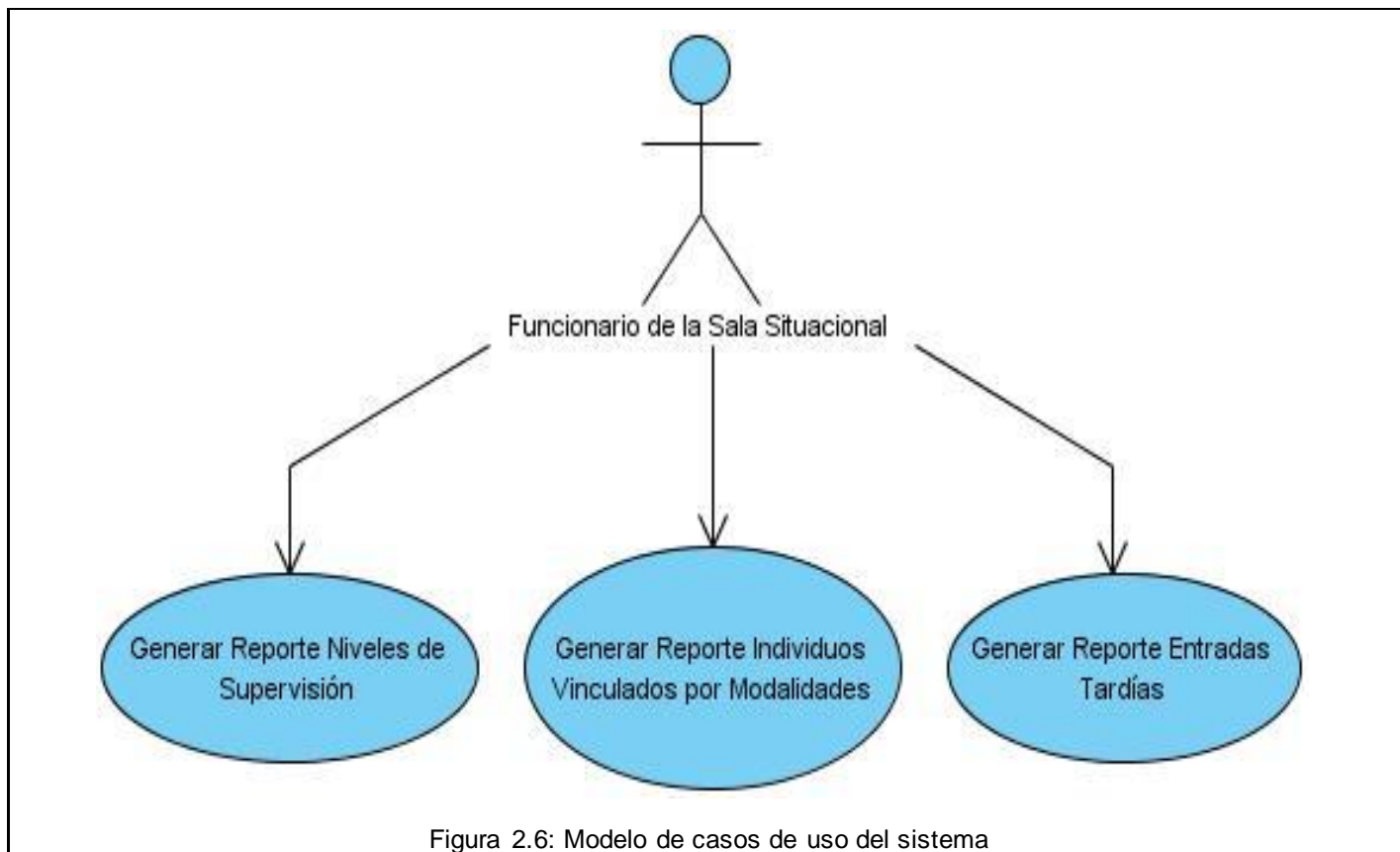


Figura 2.6: Modelo de casos de uso del sistema

**2.6.3 Descripción de casos de uso del sistema**

La descripción de los casos de uso del sistema a construir partirá de un mapa de navegación que permita visualizar la navegación del sistema entre las funcionalidades desarrolladas para cada caso de uso, luego se mostrará al detalle su futuro funcionamiento y un prototipo de interfaz para ubicar visualmente al lector.

**2.6.3.1 Generar reporte individuos por niveles de supervisión**

## **CAPITULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

Nombre del caso de uso:	Generar reporte individuos por niveles de supervisión
Objetivo:	Generar reporte individuos por niveles de supervisión, muestra la composición de la población penal según el nivel de supervisión de cada individuo: <b>máximo, medio, mínimo</b> y a los individuos que no les ha registrado el nivel ( <b>sin nivel</b> ).
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 la Población Penal, en el subsistema de Sala Situacional. Luego de seleccionar la opción Niveles de Supervisión, el sistema buscará las configuraciones correspondientes y muestra la Información Operativa o Histórica según el formato deseado.
Referencias	R1
Precondiciones:	El Funcionario de la Sala Situacional debe estar autenticado.
<b>Garantías (Poscondiciones)</b>	
Mínimas:	Se debe 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 Niveles de Supervisión.
<b>Relaciones</b>	
Caso de uso Incluido:	-
Caso de uso Extendido:	-
<b>Flujo Básico: Generar reporte individuos por niveles de supervisión</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El Funcionario de la Sala Situacional selecciona el Reporte de Niveles de Supervisión.	2. El sistema muestra directamente de la Información Operativa, la Información tabular. (Ver Sección 1), además tiene la opción además de mostrar información Histórica.

## **CAPITULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

<p>3. El Funcionario de la Sala Situacional selecciona el tipo de información que desea consultar.</p>	<p>4. El sistema muestra las posibles acciones según el tipo de información seleccionada:</p> <p>4.1 En el caso de la Información Operativa:</p> <p style="margin-left: 20px;">a) Información gráfico. Ver Sección 2</p> <p style="margin-left: 20px;">b) Listados. Ver Sección 3</p> <p>4.2 En el caso de la Información Histórica:</p> <p style="margin-left: 20px;">a) Información tabular. Ver Sección 1</p> <p style="margin-left: 20px;">b) Información gráfico. Ver Sección 2</p> <p style="margin-left: 20px;">c) Configuración. Ver Sección 4</p>
<b>Sección 1: “Información tabular”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
<p>1. El Funcionario de la Sala Situacional selecciona el tab de Información tabular.</p>	<p>2. El sistema muestra la información correspondiente en formato tabular.</p>
<b>Sección 2: “Información gráfica”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
<p>1. El Funcionario de la Sala Situacional selecciona el tab de Información gráfica.</p>	<p>2. El sistema muestra la información correspondiente en forma de gráficas.</p>
<b>Sección 3: “Listados”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
<p>1. El Funcionario de la Sala Situacional selecciona el tab de Listados.</p>	<p>2. El sistema muestra la información correspondiente en forma de listado. Donde el funcionario puede ver los listados por las siguientes categorías:</p> <ul style="list-style-type: none"> <li>• Máximo.</li> <li>• Medio.</li> <li>• Mínimo.</li> <li>• Sin Nivel.</li> </ul>
<p>3. El funcionario de la Sala Situacional selecciona cualquiera de los tab.</p>	<p>4. Se muestra el listado correspondiente ordenado alfabéticamente.</p>
<b>Sección 4: “Configuración”</b>	

**CAPITULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

Acción del actor	Respuesta del Sistema
1. El Funcionario de la Sala Situacional selecciona el tab de Configuración.	2. El sistema muestra los parámetros a configurar: <ul style="list-style-type: none"> <li>• Unidad de medida.</li> <li>• Fecha de inicio.</li> <li>• Fecha final.</li> <li>• Frecuencia.</li> </ul>
3. El Funcionario de la Sala Situacional selecciona la unidad de medida: <ul style="list-style-type: none"> <li>• Año.</li> <li>• Mes.</li> <li>• Semana.</li> <li>• Día.</li> </ul>	
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 valida que el valor entrado en la frecuencia sea correcto.
	8. El sistema muestra el mensaje de información “Datos guardados correctamente”.
<b>Flujos Alternos</b>	
<b>Flujo Alterno Flujo Básico 2*a Cerrar</b>	
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 Niveles de Supervisión.
	2*a.3 Muestra la interfaz de inicio de la Sala Situacional.
<b>Flujo alternativo Sección 1: 2a No existe información tabular que mostrar.</b>	
Acción del actor	Respuesta del sistema

	2a.1 El sistema muestra el mensaje de información: “No existen datos para mostrar”.
<b>Flujo alternativo Sección 2: 2a No existe información gráfica que mostrar.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	2a.1 El sistema muestra el mensaje de información: “No existen datos para mostrar”.
<b>Flujo alternativo Sección 3: 4a No existe listados que mostrar.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	4a.1 El sistema muestra la interfaz vacía.
<b>Flujo alternativo Sección 4: 2a Cancelar la configuración.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
2a.1 El funcionario de la Sala Situacional selecciona la opción “Cancelar”.	2a.2 El sistema retorna a la última configuración guardada.
<b>Flujo alternativo Sección 4: 7a Datos Incorrectos.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	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.

Tabla 2.2 Descripción del CU Generar reporte individuos por niveles de supervisión

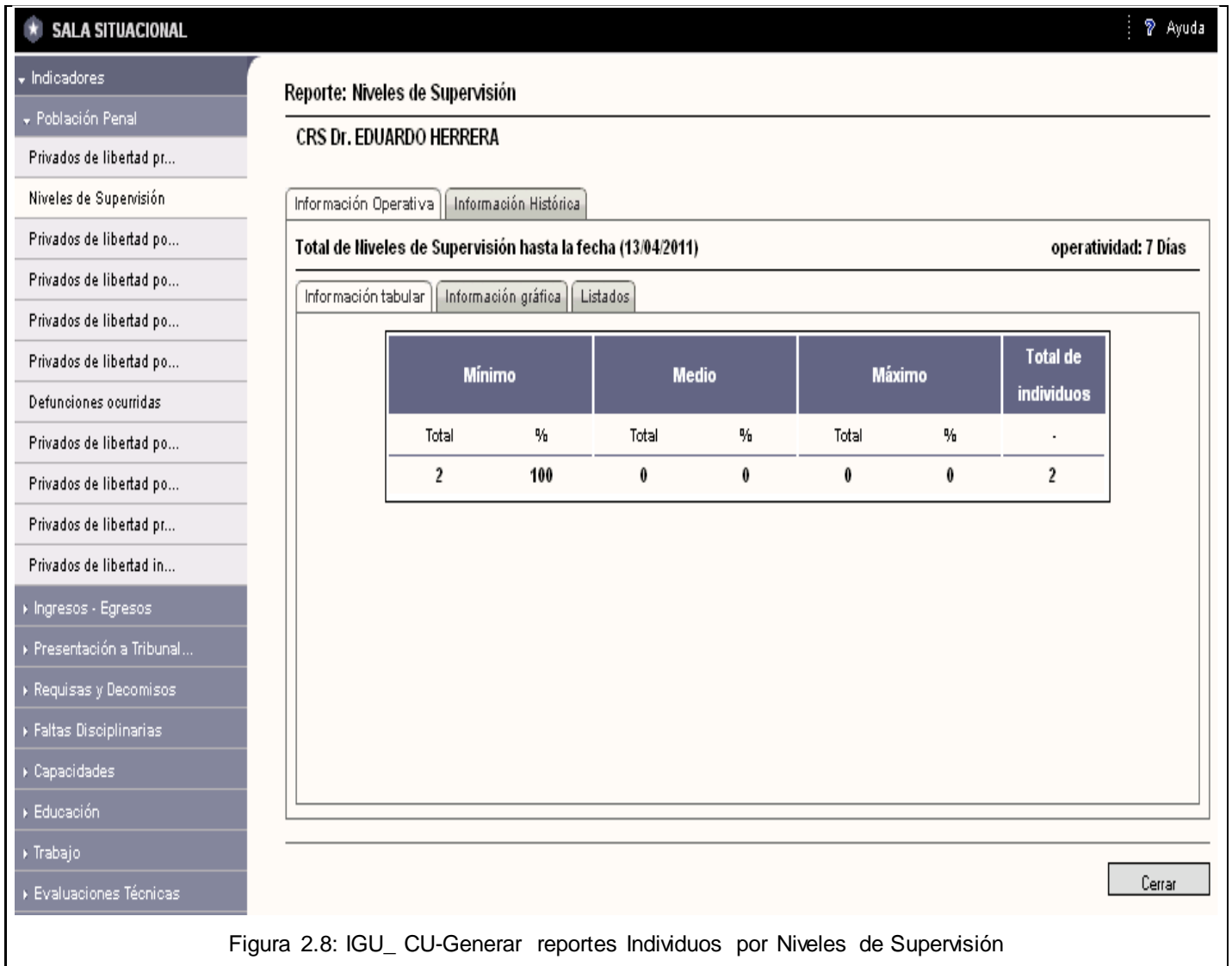


Figura 2.8: IGU\_ CU-Generar reportes Individuos por Niveles de Supervisión

**2.6.3.2 Generar reporte individuos con entradas tardías**

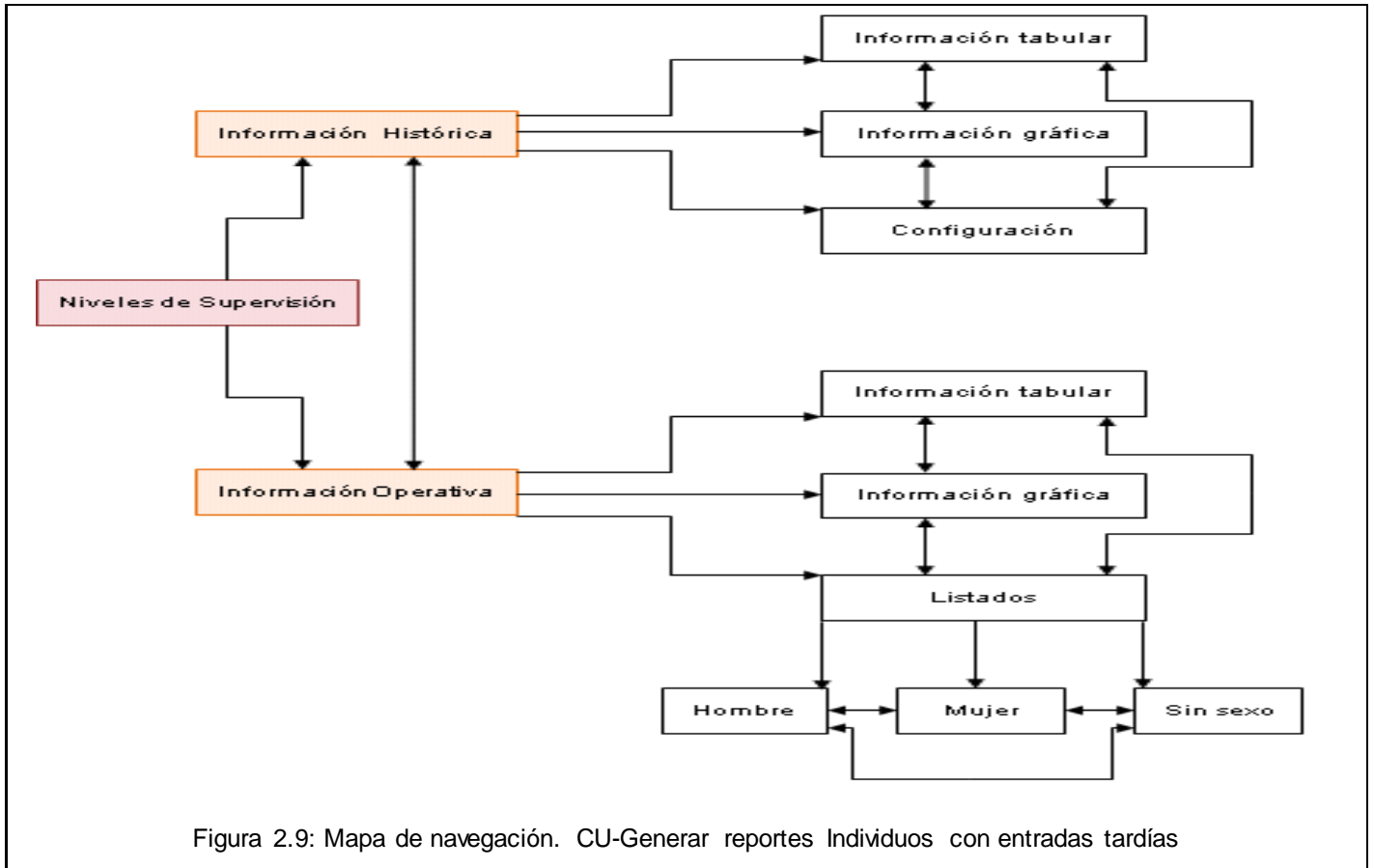


Figura 2.9: Mapa de navegación. CU-Generar reportes Individuos con entradas tardías

Nombre del Caso de Uso:	Generar reporte individuos con entradas tardías
Objetivo:	Generar reporte individuos con entradas tardías, muestra la cantidad de entradas tardías al centro en un periodo dado en 3 categorías <b>hombre, mujer y sin sexo</b> . Este Reporte será mostrado en el formato que seleccione el Funcionario de la Sala Situacional, tanto en Adobe Reader (fichero pdf), Excel o en 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 del Control de Entradas y Salidas, en el subsistema de Sala Situacional. Luego de seleccionar la opción Entradas Tardías, el sistema buscará las configuraciones correspondientes y

## **CAPITULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

	muestra la Información Operativa o Histórica según el formato deseado.
Referencias:	R2
Precondiciones:	El Funcionario de la Sala Situacional debe estar autenticado.
<b>Garantías (Poscondiciones)</b>	
Mínimas:	Se debe 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 Entradas Tardías.
<b>Relaciones</b>	
Caso de uso Incluido:	
Caso de uso Extendido:	
<b>Flujo Básico: Generar reporte individuos con entradas tardías</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El Funcionario de la Sala Situacional selecciona el Reporte de Entradas Tardías.	2. El sistema muestra directamente de la Información Operativa, la Información tabular. (Ver Sección 1), además tiene la opción además de mostrar información Histórica.
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 seleccionada: 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
<b>Sección 1: "Información tabular"</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>



## **CAPITULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

1. El Funcionario de la Sala Situacional selecciona el tab de Información tabular.	2. El sistema muestra la información correspondiente en formato tabular.
<b>Sección 2: “Información gráfica”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El Funcionario de la Sala Situacional selecciona el tab de Información gráfica.	2. El sistema muestra la información correspondiente en forma de gráficas.
<b>Sección 3: “Listados”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El Funcionario de la Sala Situacional selecciona el tab de Listados.	2. El sistema muestra la información correspondiente en forma de listado. Donde el funcionario puede ver los listados por las siguientes categorías: <ul style="list-style-type: none"> <li>• Hombre.</li> <li>• Mujer.</li> <li>• Sin sexo.</li> </ul>
3. El funcionario de la Sala Situacional selecciona cualquiera de los tab.	4. Se muestra el listado correspondiente ordenado alfabéticamente.
<b>Sección 4: “Configuración”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El Funcionario de la Sala Situacional selecciona el tab de Configuración.	2. El sistema muestra los parámetros a configurar: <ul style="list-style-type: none"> <li>• Unidad de medida.</li> <li>• Fecha de inicio.</li> <li>• Fecha final.</li> <li>• Frecuencia.</li> </ul>
3. El Funcionario de la Sala Situacional selecciona la unidad de medida: <ul style="list-style-type: none"> <li>• Año.</li> <li>• Mes.</li> <li>• Semana.</li> <li>• Día.</li> </ul>	

## **CAPITULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

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 valida que el valor entrado en la frecuencia sea correcto.
	8. El sistema muestra el mensaje de información "Datos guardados correctamente".
<b>Flujos Alternos</b>	
<b>Flujo Alterno Flujo Básico 2*a Cerrar</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
2*a.1 El funcionario de la Sala Situacional selecciona la opción "Cerrar".	2*a.2 Cierra la interfaz Generar Reporte Entradas Tardías.
	2*a.3 Muestra la interfaz de inicio de la Sala Situacional.
<b>Flujo alternativo Sección 1: 2a No existe información tabular que mostrar.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	2a.1 El sistema muestra el mensaje de información: "No existen datos para mostrar".
<b>Flujo alternativo Sección 2: 2a No existe información gráfica que mostrar.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	2a.1 El sistema muestra el mensaje de información: "No existen datos para mostrar".
<b>Flujo alternativo Sección 3: 2a No existe listados que mostrar.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	4a.1 El sistema muestra la interfaz vacía.
<b>Flujo alternativo Sección 4: 2a Cancelar la configuración.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

2a.1 El funcionario de la Sala Situacional selecciona la opción "Cancelar".	2a.2 El sistema retorna a la última configuración guardada.
<b>Flujo alternativo Sección 4: 7a Datos Incorrectos.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	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.

Tabla 2.3 Descripción del CU Generar reporte individuos con entradas tardías

**Reporte: Entradas Tardías**  
CRS Dr. EDUARDO HERRERA

Información Operativa | Información Histórica

**Total de Entradas Tardías entre (08/04/2011 y 15/04/2011)** operatividad: 7 Dias

Información tabular | Información gráfica | Listados

Femeninos		Masculinos		No definido		Total de individuos
Total	%	Total	%	Total	%	-
1	50	0	0	1	50	2

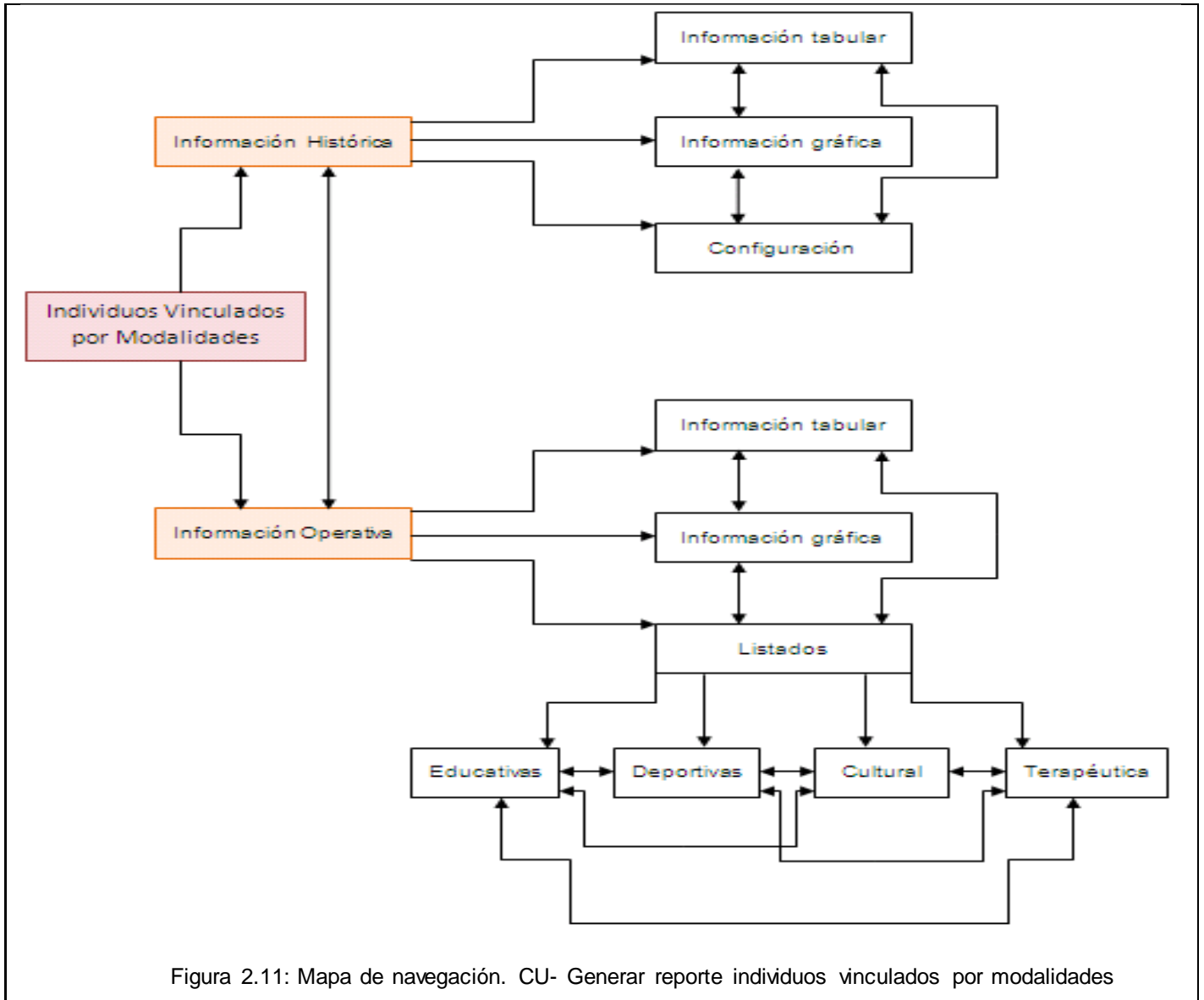
**Leyenda**

Abreviatura	Descripción
<b>No definido</b>	<b>Individuos sin sexo registrado</b>

Cerrar

Figura 2.10: IGU\_CU-Generar reporte individuos con entradas tardías

### 2.6.3.3 Generar reporte individuos vinculados por modalidades



Nombre del Caso de Uso:	Generar reporte individuos vinculados por modalidades
Objetivo:	Generar Reportes Individuos Vinculados por Modalidades, muestra las cantidades de individuos que han sido incorporados en las diferentes actividades: <b>educativas, deportivas, cultural o terapéuticas.</b>

## **CAPITULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

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 Actividades de Atención, en el subsistema de Sala Situacional. Luego de seleccionar la opción Individuos Vinculados por Modalidades, el sistema buscará las configuraciones correspondientes y muestra la Información Operativa o Histórica según el formato deseado.
Referencias:	R3
Precondiciones:	El Funcionario de la Sala Situacional debe estar autenticado.
<b>Garantías (Poscondiciones)</b>	
Mínimas:	Se debe 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 Individuos Vinculados por Modalidades.
<b>Relaciones</b>	
Caso de uso Incluido:	
Caso de uso Extendido:	
<b>Flujo Básico: Generar reporte individuos vinculados por modalidades</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El Funcionario de la Sala Situacional selecciona el Reporte de Niveles de Supervisión.	2. El sistema muestra directamente de la Información Operativa, la Información tabular. (Ver Sección 1), además tiene la opción además de mostrar información Histórica.
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 seleccionada: 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:

## **CAPITULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

	<p>a) Información tabular. Ver Sección 1</p> <p>b) Información gráfico. Ver Sección 2</p> <p>c) Configuración. Ver Sección 4</p>
<b>Sección 1: “Información tabular”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El Funcionario de la Sala Situacional selecciona el tab de Información tabular.	2. El sistema muestra la información correspondiente en formato tabular.
<b>Sección 2: “Información gráfica”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El Funcionario de la Sala Situacional selecciona el tab de Información gráfica.	2. El sistema muestra la información correspondiente en forma de gráficas.
<b>Sección 3: “Listados”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El Funcionario de la Sala Situacional selecciona el tab de Listados.	2. El sistema muestra la información correspondiente en forma de listado. Donde el funcionario puede ver los listados por las siguientes categorías: <ul style="list-style-type: none"> <li>• Educativas</li> <li>• Deportivas</li> <li>• Cultural</li> <li>• Terapéuticas</li> </ul>
3. El funcionario de la Sala Situacional selecciona cualquiera de los tab.	4. Se muestra el listado correspondiente ordenado alfabéticamente.
<b>Sección 4: “Configuración”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
1. El Funcionario de la Sala Situacional selecciona el tab de Configuración.	2. El sistema muestra los parámetros a configurar: <ul style="list-style-type: none"> <li>• Unidad de medida.</li> <li>• Fecha de inicio.</li> <li>• Fecha final.</li> <li>• Frecuencia.</li> </ul>

## **CAPITULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN**

<p>3. El Funcionario de la Sala Situacional selecciona la unidad de medida:</p> <ul style="list-style-type: none"> <li>• Año.</li> <li>• Mes.</li> <li>• Semana.</li> <li>• Día.</li> </ul>	
<p>4. El Funcionario de la Sala Situacional selecciona el intervalo de tiempo para la información histórica.</p>	
<p>5. El Funcionario de la Sala Situacional introduce el valor correspondiente a la frecuencia.</p>	
<p>6. El Funcionario de la Sala Situacional guarda los datos.</p>	<p>7. El sistema valida que el valor entrado en la frecuencia sea correcto.</p>
	<p>8. El sistema muestra el mensaje de información “Datos guardados correctamente”.</p>
<b>Flujos Alternos</b>	
<b>Flujo Alterno Flujo Básico 2*a Cerrar</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>2*a.1 El funcionario de la Sala Situacional selecciona la opción “Cerrar”.</p>	<p>2*a.2 Cierra la interfaz Generar Reporte Individuos Vinculados por Modalidades.</p>
	<p>2*a.3 Muestra la interfaz de inicio de la Sala Situacional.</p>
<b>Flujo alternativo Sección 1: 2a No existe información tabular que mostrar.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>2a.1 El sistema muestra el mensaje de información: “No existen datos para mostrar”.</p>
<b>Flujo alternativo Sección 2: 2a No existe información gráfica que mostrar.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p>2a.1 El sistema muestra el mensaje de información: “No existen datos para mostrar”.</p>

<b>Flujo alternativo Sección 3: 4a No existe listados que mostrar.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	4a.1 El sistema muestra la interfaz vacía.
<b>Flujo alternativo Sección 4: 2a Cancelar la configuración.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
2a.1 El funcionario de la Sala Situacional selecciona la opción "Cancelar".	2a.2 El sistema retorna a la última configuración guardada.
<b>Flujo alternativo Sección 4: 7a Datos Incorrectos.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	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.

Tabla 2.4 Descripción del CU Generar reporte individuos vinculados por modalidades

**SALA SITUACIONAL** Ayuda

**Reporte: Individuos Vinculados por Modalidades**  
**CRS Dr. EDUARDO HERRERA**

Información Operativa | Información Histórica

**Total de Individuos Vinculados por Modalidades entre (08/04/2011 y 15/04/2011)** **operatividad: 7 Días**

Información tabular | Información gráfica | Listados

Deportiva		Cultural		Educativa		Terapéutica		Total de individuos
Total	%	Total	%	Total	%	Total	%	-
1	33.3	1	33.3	1	33.3	0	0	1

Cerrar

Figura 2.12: IGU\_CU- Generar reporte individuos vinculados por modalidades



## **2.7 Conclusiones**

Se puede concluir que en este capítulo:

- Se realizó un estudio de la arquitectura del SIGEP y del futuro subsistema Sala Situacional a desarrollar siguiendo sus especificidades.
- Se profundizó en los elementos que componen el subsistema de Sala Situacional del SIGEP para el régimen Intramuros y hacia dónde estaba dirigido el alcance del trabajo.
- Se definieron los requisitos que debe cumplir el sistema en toda su totalidad, así como los actores del mismo, incluyendo la descripción de cada una de sus funcionalidades.

## CAPITULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

### 3.1 Introducción

En este capítulo se define como se va a desarrollar la aplicación, tanto la lógica a través de los artefactos web como física mediante sus componentes, así como la interrelación entre ambos, aplicando los principios de diseño y codificación establecidos.

### 3.2 Actividades de diseño e implementación

#### 3.2.1 Diseño de dominio

En el diseño del dominio se identifican las entidades a utilizar en las distintas capas de la aplicación .Se identifican las entidades que serán gestionadas por la capa de negocio, persistentes o recuperadas por la capa de acceso a datos y mostradas por la capa de presentación. Se identifican los nomencladores necesarios y las clases del dominio a utilizar definidas en el paquete domain del módulo. La definición del dominio, en especial de las entidades persistentes sirve como una primera aproximación al diseño definitivo del modelo de datos. En la Figura 3.1 se muestra el diagrama de clases persistentes:

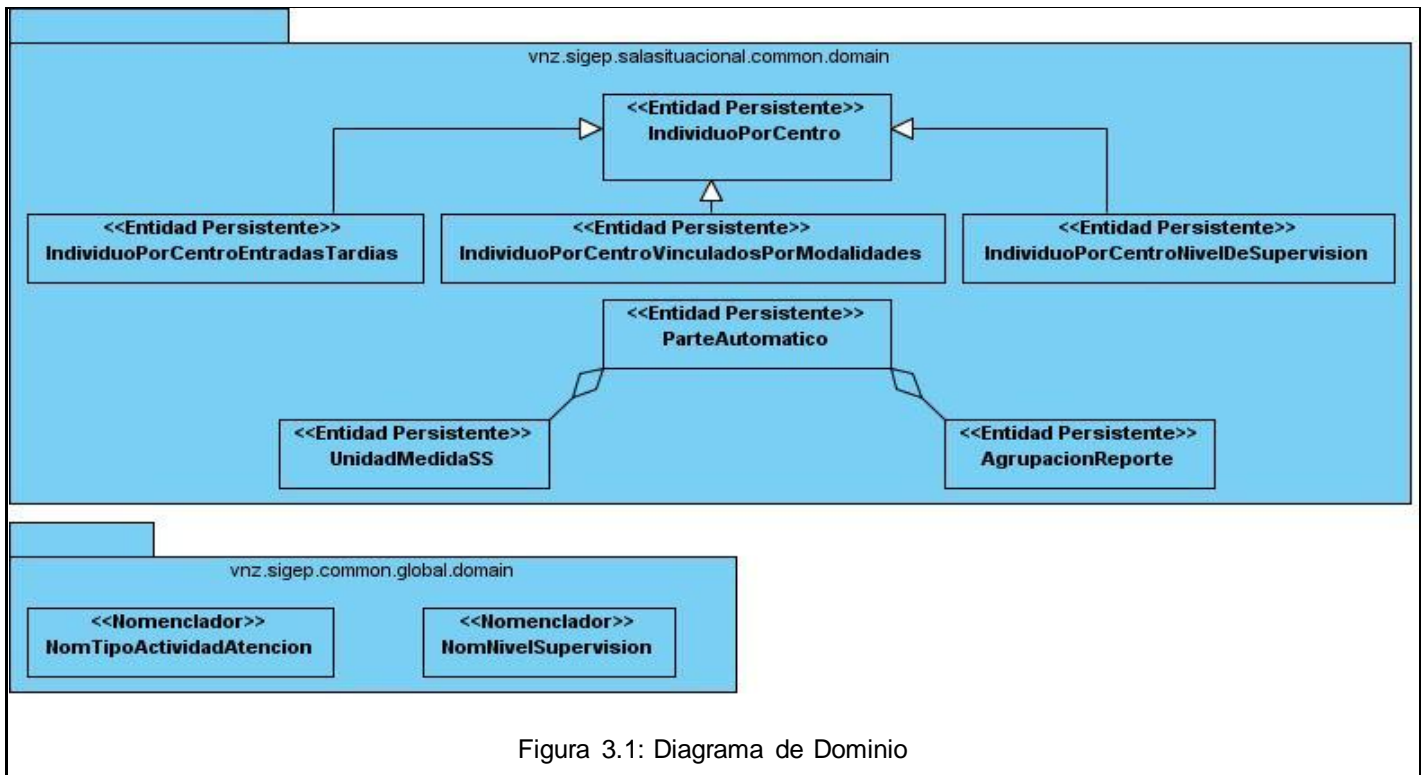


Figura 3.1: Diagrama de Dominio

**3.2.2 Diseño del modelo de datos**

El diagrama de clases del modelo de datos es el diagrama de clases persistentes del sistema visto desde el punto de vista físico de almacenamiento.

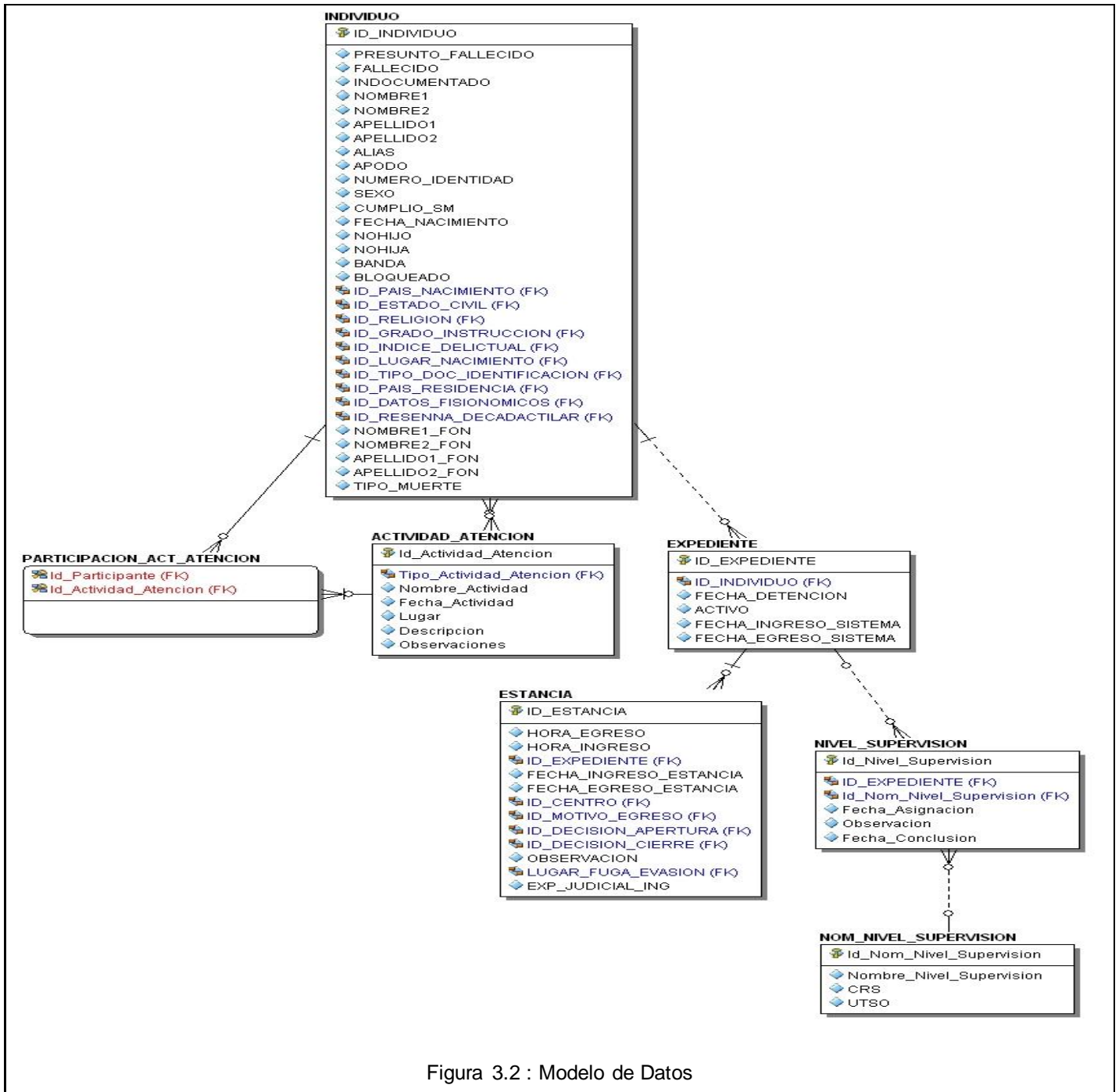


Figura 3.2 : Modelo de Datos

### **3.2.3 Diseño de la Capa de Presentación**

En esta actividad se definen los componentes con los que interactuará directamente el usuario (las vistas) de la aplicación. También se definen los controladores que van a intervenir en este proceso, manejando en el servidor las peticiones del usuario, y la comunicación con las interfaces de la capa de negocio. El diseño de estos componentes está estrechamente relacionado con otros componentes definidos para la arquitectura que cumplen funciones dentro del flujo de la aplicación, tanto del lado del cliente como del servidor. El uso dentro de la arquitectura del sistema de frameworks destinados especialmente a aplicaciones basadas en la web como Spring-MVC proporciona controladores genéricos que realizan funciones comunes para esta clase de aplicaciones. Al definir controladores que extienden de estas clases genéricas y agregarles las funcionalidades específicas deseadas, se gana en productividad y robustez en la solución objetivo.

Se tomará como referencia el caso de uso *Generar reporte individuos vinculados por modalidades* para la presentación en el documento. En la Figura 3.3 se puede observar el resultado de este diseño, los diseños de la capa de presentación de los restantes casos de uso se pueden observar en los Anexos correspondientes.

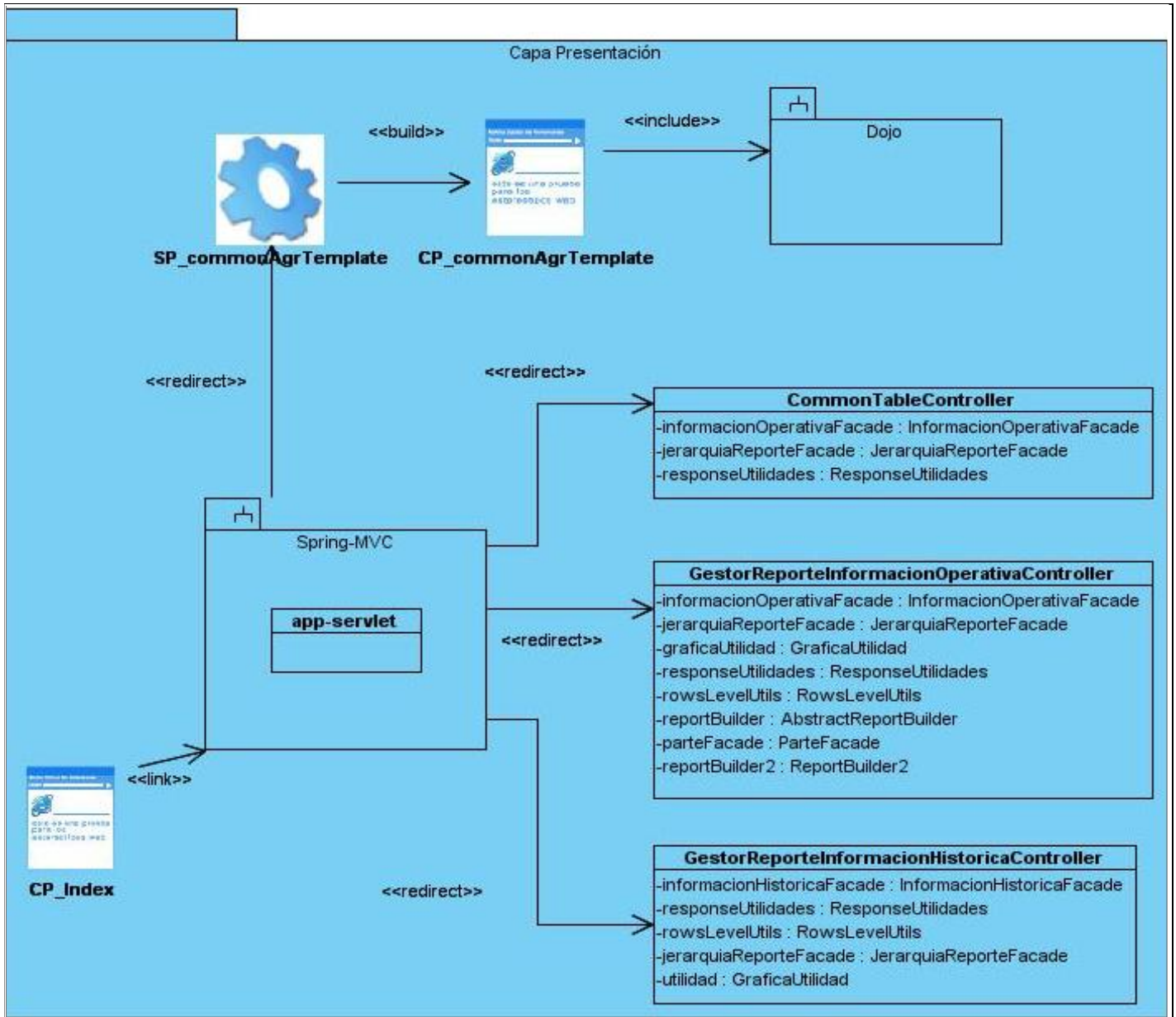


Figura 3.3: Diseño de la Capa de Presentación CU- Generar reporte individuos vinculados por modalidades

### **3.2.4 Diseño de la Capa Lógica de Negocio**

El diseño de la capa de negocio contiene las clases necesarias para cubrir las funcionalidades definidas para el módulo y expone, a través de una fachada, sus funcionalidades a la capa de presentación. Si se necesita la comunicación directa de otros módulos o subsistemas con el que se diseña, se define una fachada para ello y se exponen los servicios necesarios. En la Figura 3.4 se puede observar el resultado de este diseño para el CU de referencia, para ver los restantes puede remitirse a los Anexos correspondientes.

## CAPITULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

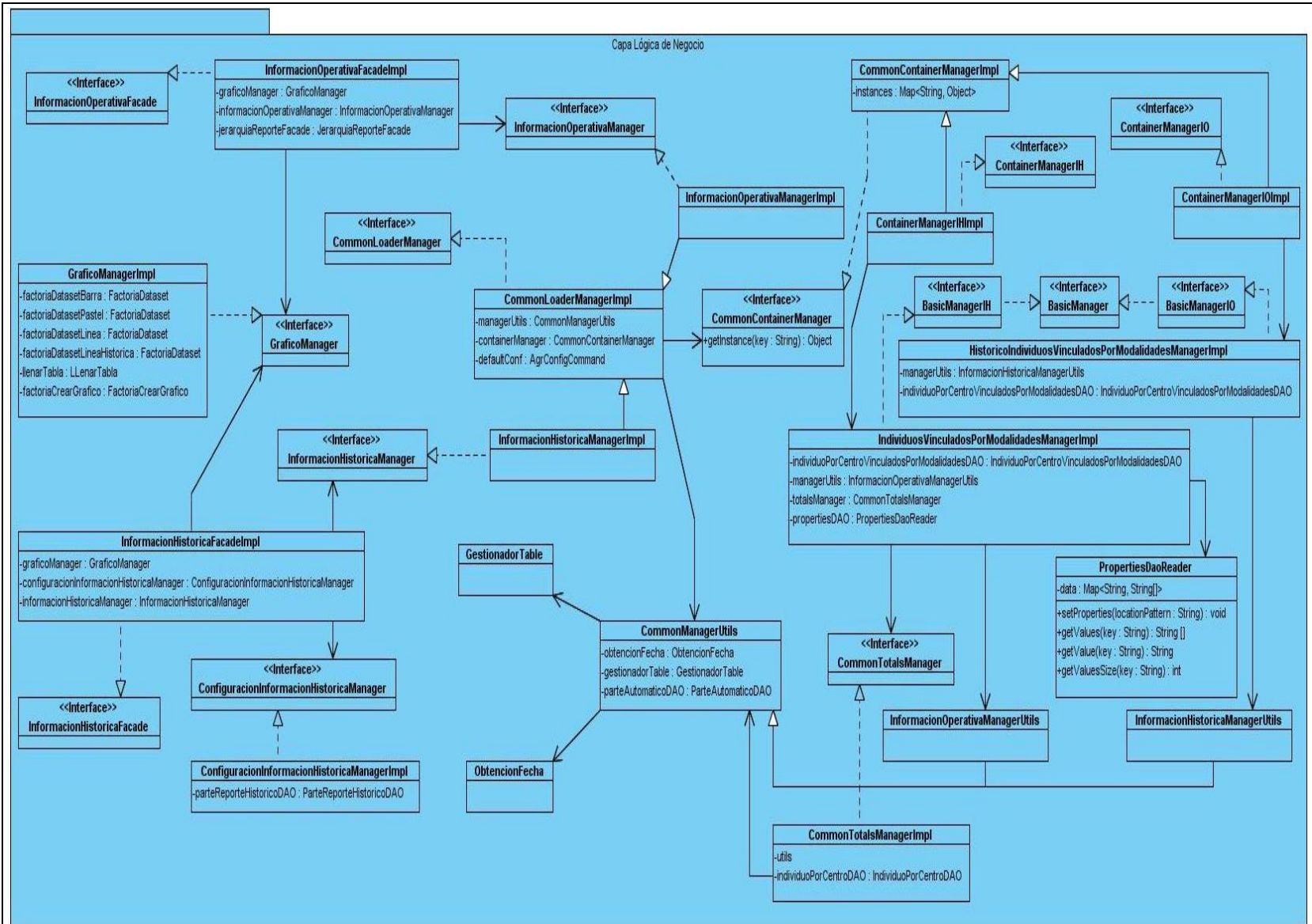


Figura 3.4: Diseño de la Capa de Negocio CU-Generar reporte individuos vinculados por modalidades

### **3.2.5 Diseño de la Capa de Acceso a Datos**

Las funcionalidades de la capa de acceso a datos son una consecuencia directa de las funcionalidades de la capa de negocio. El diseño de esta capa esta por tanto ligado muy estrechamente a la capa superior, pero de modo que exista el acoplamiento mínimo con esta última. En el caso del gestor de base de datos, se busca también un acoplamiento mínimo que garantice la independencia de características específicas del gestor, pero al mismo tiempo se aprovechan al máximo las ventajas que brinda Oracle en los casos que sea conveniente. Como parte de la arquitectura base empleada, se utiliza el framework Hibernate para alcanzar el bajo acoplamiento deseado entre la aplicación y el gestor de base de datos. Dentro de ARBAWEB se especifica también un objeto de acceso a datos genérico, que define e implementa métodos comunes para todos los DAOs, como métodos de persistencia, de recuperación de datos. Las interfaces definidas en esta capa permiten la comunicación hacia y desde la capa de negocio, brindándole a esta ultima de manera transparente las funcionalidades propias del acceso a datos.

En la Figura 3.5 se puede observar el resultado de este diseño para el CU de referencia, para ver los restantes puede remitirse a los Anexos correspondientes.





### 3.2.6 Relaciones entre capas

En este diagrama se muestra la relación que existen entre cada una de las capas.

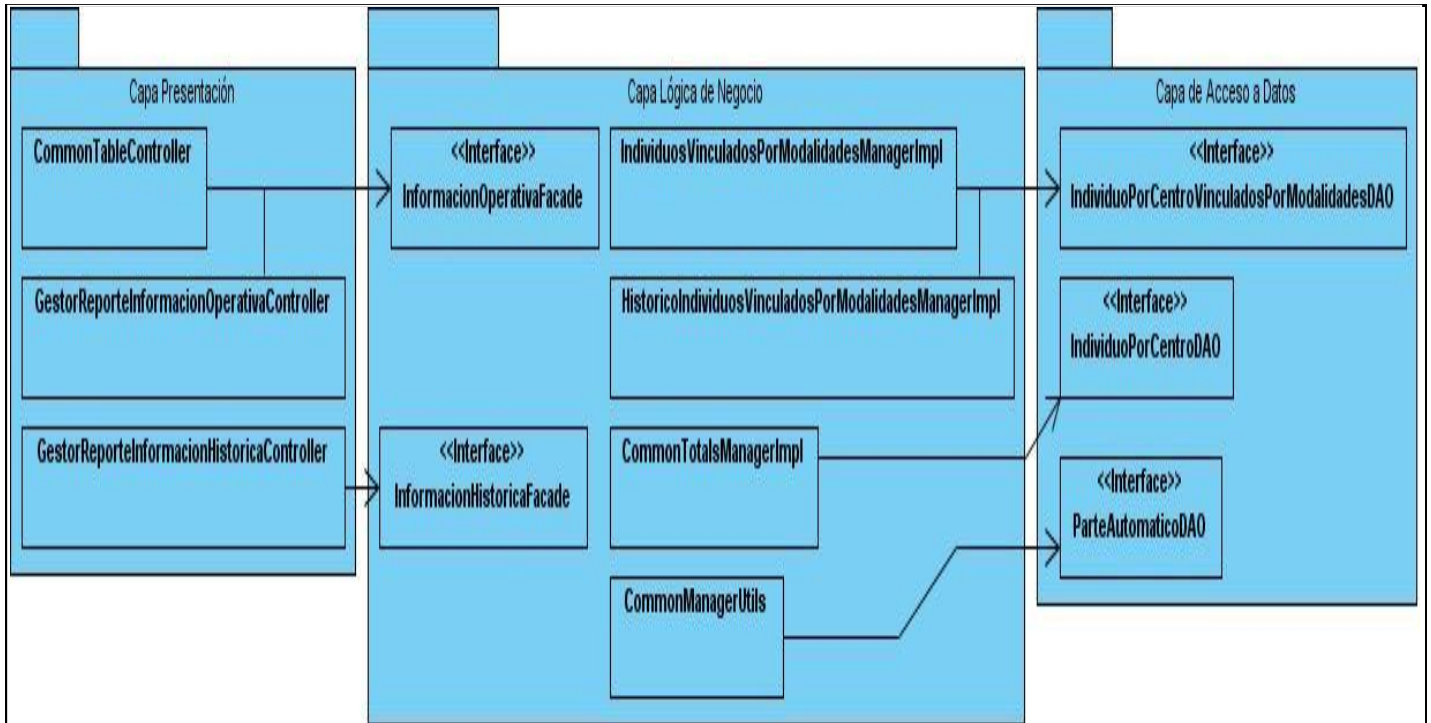


Figura 3.6: Relaciones entre capas CU- Generar reporte individuos vinculados por modalidades

### 3.2.7 Implementación de las entidades del dominio

Las entidades del dominio suelen tener muy poco comportamiento. Los métodos *equals()*, *hashCode()* y *toString()* deben ser implementados en la mayoría de los casos, ya que son frecuentemente utilizados. En la implementación de las entidades del dominio se deben definir los atributos de cada entidad de manera que queden listas en un buen por ciento para implementaciones reusables. A continuación se muestra la implementación de los métodos mencionados anteriormente en la clase del dominio *IndividuoPorCentro*.

```

@Override
public boolean equals(Object o) {
    if (!(o instanceof IndividuoPorCentro))
        return false;

    IndividuoPorCentro ind = (IndividuoPorCentro) o;
    return new EqualsBuilder().append(id_individuo, ind.id_individuo)

```

```
        .append(nombre_centro, ind.nombre_centro)
        .append(id_estancia, ind.id_estancia)
        .append(nombre, ind.nombre)
        .append(sexo, ind.sexo)
        .append(nombre_nacionalidad, ind.nombre_nacionalidad)
        .append(nombre_tipo_centro, ind.nombre_tipo_centro)
        .append(color_piel, ind.color_piel)
        .isEqual();
    }

    @Override
    public int hashCode() {
        return new HashCodeBuilder(13, 17)
            .append(id_individuo)
            .append(nombre_centro)
            .append(id_estancia)
            .append(nombre)
            .append(sexo)
            .append(nombre_nacionalidad)
            .append(nombre_tipo_centro)
            .append(color_piel)
            .toHashCode();
    }

    @Override
    public String toString() {
        return new ToStringBuilder(this, ToStringStyle.SHORT_PREFIX_STYLE)
            .append("id_individuo", id_individuo)
            .append("nombre_centro", nombre_centro)
            .append("id_estancia", id_estancia)
            .append("nombre", nombre)
            .append("sexo", sexo)
            .append("nombre_nacionalidad", nombre_nacionalidad)
            .append("nombre_tipo_centro", nombre_tipo_centro)
            .append("color_piel", color_piel)
            .toString();
    }
}
```

### 3.2.8 Implementación de las interfaces de los managers

En esta actividad se implementan los métodos definidos para cada una de las interfaces de los managers, ajustándose a las funcionalidades previstas. Cada método implementado tiene que verificar la integridad de los datos e informar a la capa de interfaz cualquier eventualidad a través de las excepciones definidas. 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-[subsistema]-[módulo]-business-context.xml**. A continuación se muestra un

ejemplo de la implementación del método `buildTable` perteneciente al manager `IndividuosVinculadosPorModalidadesManagerImpl`.

```
@Override
public Table buildTable(Map<String, Object> params) throws Exception {
    List<String> nivelCentro = (List<String>) params.get("nivelCentro");

    boolean nivelNacion = (Boolean) params.get("nivelNacion");
    Date fechaInicial = (Date) params.get("fechaInicial");
    Date fechaFinal = (Date) params.get("fechaFinal");
    Table table = (Table) params.get("tabla");

    ResultSS result = null;
    ResultSS totalResult = null;

    if (!nivelCentro.isEmpty()) {
        result =
individuoPorCentroVinculadosPorModalidadesDAO.internosVinculadosPorModalidades(nivelCe
ntro, fechaInicial, fechaFinal);
        totalResult =
individuoPorCentroVinculadosPorModalidadesDAO.totalInternosVinculadosPorModalidades(ni
velCentro, fechaInicial, fechaFinal);
        table = managerUtils.llenarTablaFromResult(table, result, "Centro",
false, null);
    }
    if (nivelNacion) {
        result =
individuoPorCentroVinculadosPorModalidadesDAO.internosVinculadosPorModalidades(nivelCe
ntro, fechaInicial, fechaFinal);
        totalResult =
individuoPorCentroVinculadosPorModalidadesDAO.totalInternosVinculadosPorModalidades(ni
velCentro, fechaInicial, fechaFinal);
        table = managerUtils.llenarTablaFromResult(table, result, "Centro",
false, null);
    }

    table = totalsManager.addTotalDeInternosPorIndicador(nivelCentro,
nivelNacion, totalResult, table, true, null);
    table = managerUtils.getGestionadorTable().porCiento(table);

    return table;
}
```

### 3.2.9 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. A continuación se muestra un fragmento del método `buildTable` perteneciente a la implementación de la interfaz de la fachada `InformacionOperativaFacadeImpl`.

```
public Table buildTable(ParteReporte parteReporte, Map<String, Object> params) throws
Exception {
    return informacionOperativaManager.buildTableIO(parteReporte, params);
}
```

### 3.2.10 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 pero, en caso de que no satisfagan las necesidades del implementador, se usa el HQL (Lenguaje de consultas de Hibernate), que permite mayor flexibilidad y, en último caso, SQL. 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-[subsistema]-[módulo]-dataaccesscontext.xml`.

```
<bean id="individuoPorCentroVinculadosPorModalidadesDAO"
    class="vnz.sigep.salasituacional.common.dao.impl.IndividuoPorCentroVinculadosPor
    ModalidadesDAOimpl">
    <property name="sessionFactory">
        <ref bean="sessionFactory" />
    </property>
    <property name="nomTipoActividadAtencionDAO">
        <ref bean="nomTipoActividadAtencionDAO" />
    </property>
</bean>
```

### 3.2.11 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. Se configuran en el fichero del contexto de Spring correspondiente a la capa de presentación: **sigep-[subsistema]-[módulo]-servlet.xml**.

### 3.2.12 Implementación de las páginas JSP

Las paginas servidoras JSP son las encargadas de construir las vistas HTML para el uso del lado del cliente a partir de la información recibida del controlador correspondiente. El diseño gráfico de las paginas JSP en el módulo se hereda del diseño definido para el SIGEP mediante el uso de ficheros CSS (Hojas de Estilo en Cascada, del inglés Cascading Style Sheets). En la programación de las JSP se utilizan fundamentalmente las Etiquetas del framework Spring y las etiquetas estándar de Java.

```
<%@ include file="/WEB-INF/jsp/include.jsp"%>
<div dojoType="sigep:TabContainerSS" doLayout="false"
    selectedChild="deportivas">

<div dojoType="ContentPane" label="<spring:message
code="ss.individuosVinculados.1.label"/>"

href="/verVistaListadoReporte.htm?_idParteReporte=${idParteReporte}&_idAgrupacion=${idAgrupacion}&_actividad=ss.individuosVinculados.1&viewkey=1" executeScripts="true"
scriptSeparation="false" cacheContent="false" loadingMessage="Cargando..."
id="deportivas" widgetId="deportivas" align="center" style="overflow: auto; min-height: 250px"></div>

<div dojoType="ContentPane" label="<spring:message
code="ss.individuosVinculados.2.label" />"

href="/verVistaListadoReporte.htm?_idParteReporte=${idParteReporte}&_idAgrupacion=${idAgrupacion}&_actividad=ss.individuosVinculados.2&viewkey=1" executeScripts="true"
scriptSeparation="false" cacheContent="false" loadingMessage="Cargando..."
id="cultural" widgetId="cultural" align="center" style="overflow: auto; min-height: 250px"></div>

<div dojoType="ContentPane" label="<spring:message
code="ss.individuosVinculados.3.label" />"

href="/verVistaListadoReporte.htm?_idParteReporte=${idParteReporte}&_idAgrupacion=${idAgrupacion}&_actividad=ss.individuosVinculados.3&viewkey=1" executeScripts="true"
scriptSeparation="false" cacheContent="false" loadingMessage="Cargando..."
id="terapeuticas" widgetId="terapeuticas" align="center" style="overflow: auto; min-height: 250px"></div>

<div dojoType="ContentPane" label="<spring:message
```

```
code="ss.individuosVinculados.4.label" />"  
  
href="/verVistaListadoReporte.htm?_idParteReporte=${idParteReporte}&_idAgrupacion=${idAgrupacion}&_actividad=ss.individuosVinculados.4&viewkey=1" executeScripts="true"  
scriptSeparation="false" cacheContent="false" loadingMessage="Cargando..."  
id="educativas" widgetId="educativas" align="center" style="overflow: auto; min-height: 250px"></div>  
  
</div>
```

### 3.2.13 Implementación de la lógica en el cliente

En esta actividad se programa, utilizando el lenguaje JavaScript, validaciones en el cliente, y comportamiento de componentes gráficos como calendarios, tablas, botones y pantallas emergentes de error o información al usuario. También desde el cliente se lanzan peticiones al servidor usando AJAX y JSON-RPC. En el SIGEP se utilizan los componentes de la biblioteca JavaScript de componentes gráficos DojoToolkit (Dojo 0.42) los cuales se localizan en el módulo widget de Dojo.

## 3.3 Modelo de implementación

Se representa como un grafo de componentes de software unidos por medio de relaciones de dependencia, pudiendo mostrarse las interfaces que estos soporten. Es el conjunto de ficheros interrelacionados entre sí para lograr la completa funcionalidad del sistema.

### 3.3.1 Diagrama de componentes

Los diagramas de componentes son utilizados para modelar la vista estática y dinámica de un sistema. Estos muestran la organización y las dependencias entre un conjunto de componentes, ya sean éstos componentes de código fuente, librerías, binarios o ejecutables. El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación.





**Capa de Acceso a Datos**

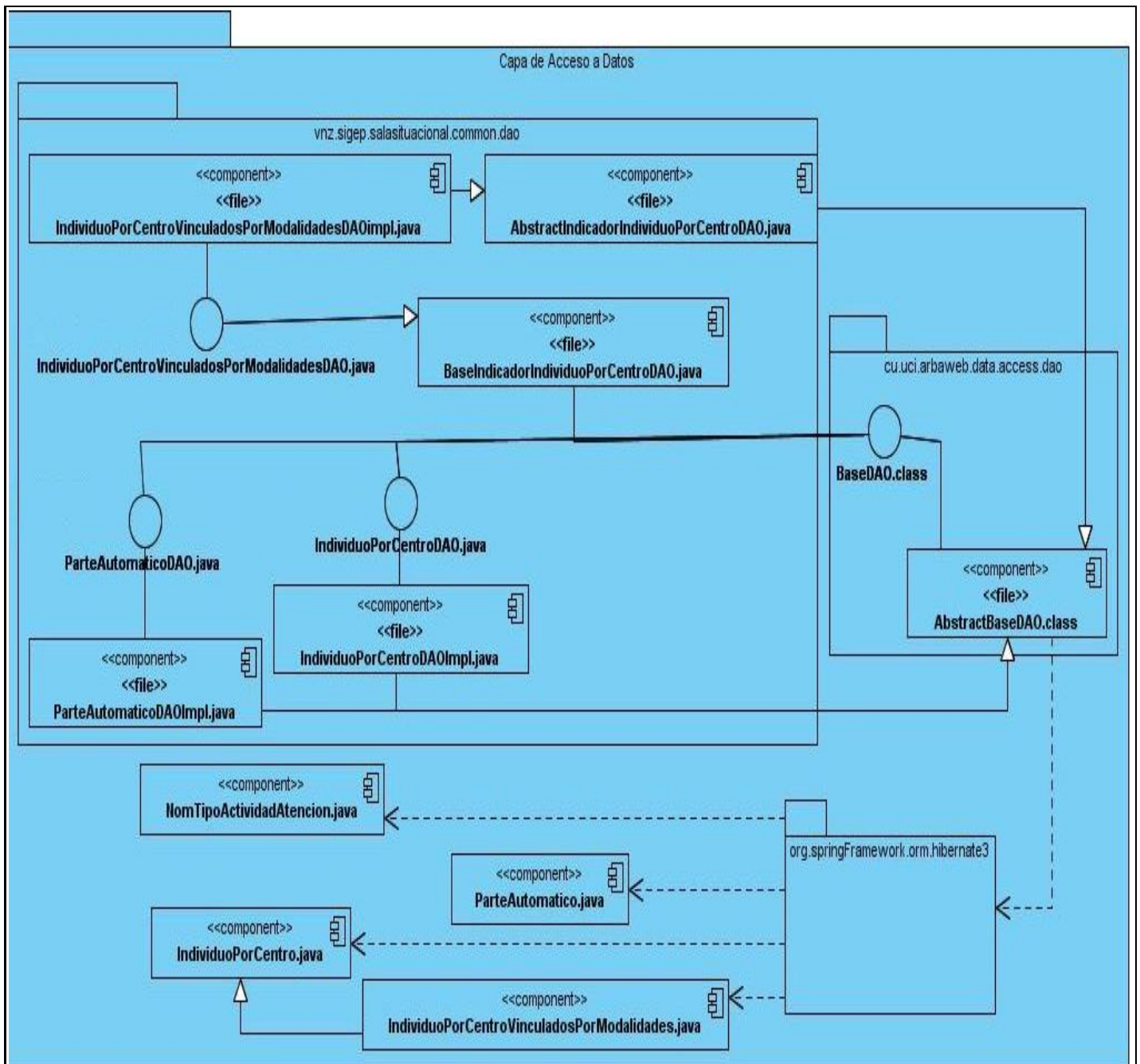


Figura 3.9: CU- Generar reporte individuos vinculados por modalidades

**Relaciones entre capas**

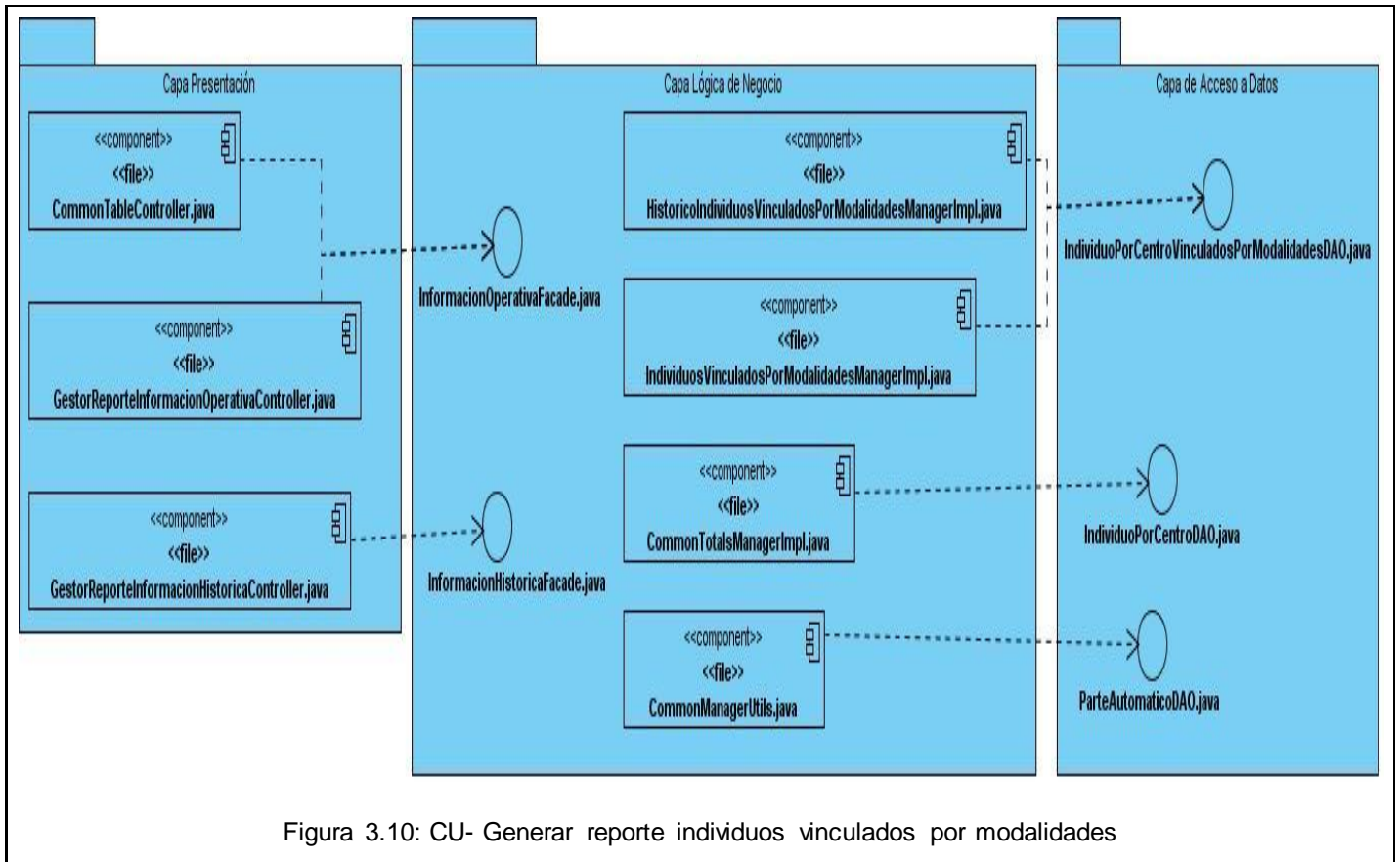
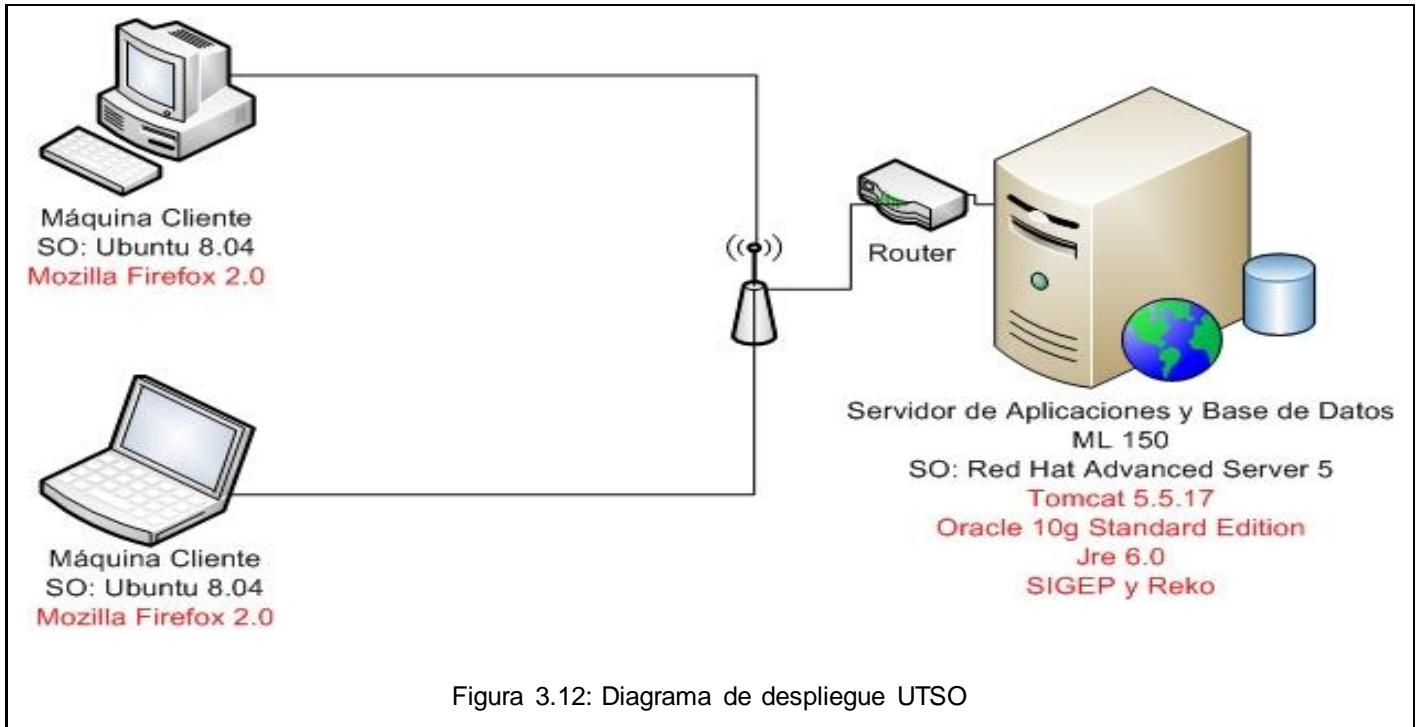
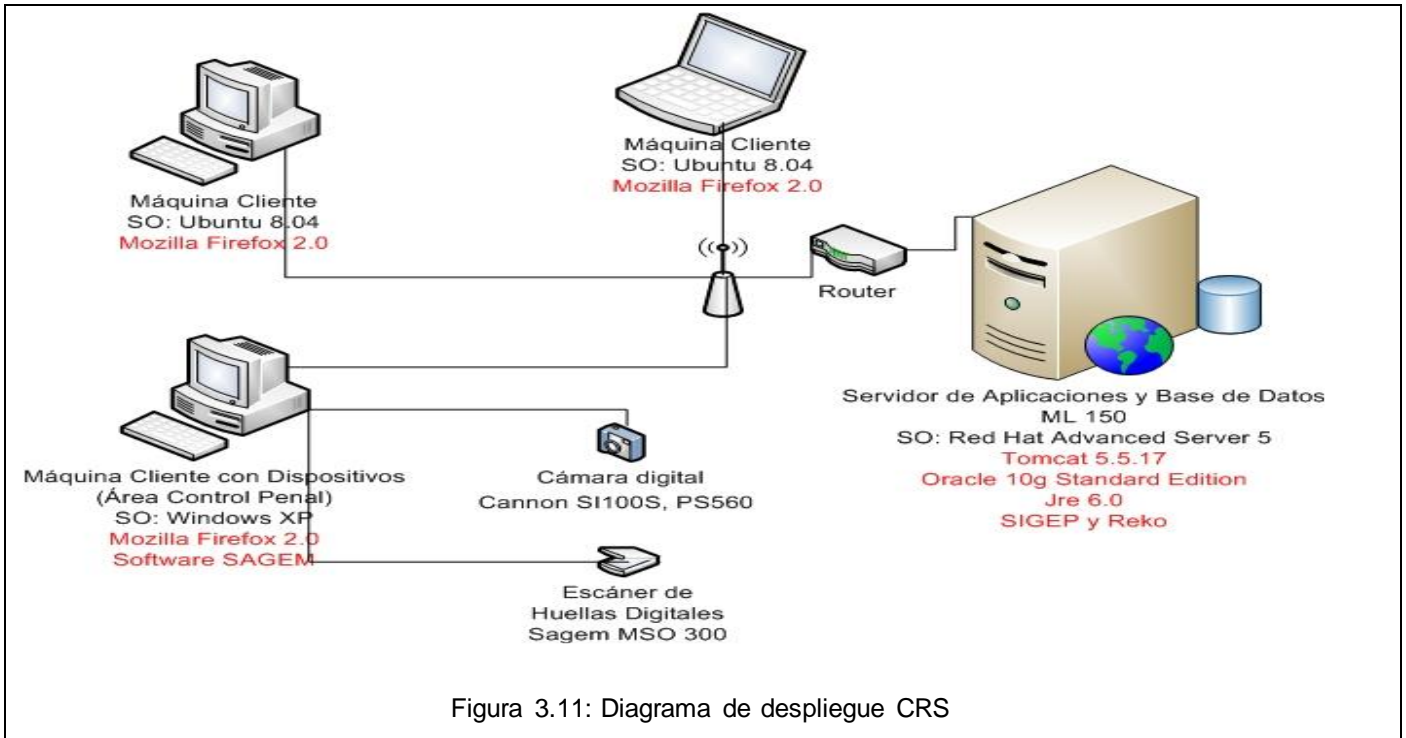


Figura 3.10: CU- Generar reporte individuos vinculados por modalidades

**3.3.2 Modelo de despliegue**

Muestra las relaciones entre el hardware y el software en el sistema final. Se representa como un grafo de nodos unidos por conexiones de comunicación.

Es la manera mediante la cual el sistema interactúa con el usuario, partiendo de una estación de trabajo cliente, la cual le hace la petición necesaria en cada caso al servidor.



### **3.4 Conclusiones**

En el presente capítulo:

- Se ha descrito la solución propuesta a través de los distintos artefactos: diseño de dominio, modelo de datos, diagrama de clases del diseño y sus relaciones.
- Ha quedado especificada además la estructura física del mismo, así como la forma en que puede intercambiar la información con los usuarios.
- Se implementaron todas las funcionalidades definidas para el subsistema.
- Se utilizaron las facilidades de los frameworks y se interactuó con componentes ya existentes.
- Se ha integrado el subsistema Sala Situacional al SIGEP.

## **CONCLUSIONES**

Después de terminado el trabajo, se llegaron a las conclusiones siguientes:

- Ninguno de los sistemas relacionados estudiados se ajusta a los requerimientos del Sistema Penitenciario Venezolano ni del SIGEP.
- Los sistemas desarrollados (SIGEP-CRS, SIGEP-UTSO) para el régimen Extramuros del Sistema Penitenciario Venezolano no cuentan con ninguna herramienta de apoyo a la toma de decisiones a partir del análisis de la información que gestionan.
- El subsistema Sala Situacional del SIGEP de la República Bolivariana de Venezuela para el régimen Extramuros del Sistema Penitenciario Venezolano fue diseñado y desarrollado de acuerdo a los requerimientos del sistema penal para el cual se realizó y al sistema informático del que forma parte.
- El subsistema Sala Situacional fue integrado al SIGEP en la aplicación de Extramuros SIGEP-CRS con resultados satisfactorios.

## **RECOMENDACIONES**

El despliegue del sistema con las nuevas funcionalidades para el régimen Extramuros.

Debido a la amplitud de las funcionalidades de Sala Situacional se recomienda hacer extensiva la experiencia a otros proyectos productivos que generan reportes en sus aplicaciones.

## BIBLIOGRAFÍA

1. **Iglesias, Adolfo M. y Pérez, Maikel.** *Sistema informático de apoyo a la gestión de la Sala Situacional del Sistema Penitenciario Venezolano.* Universidad de las Ciencias Informáticas. Ciudad Habana, Cuba : s.n., 2007. Tesis de grado.
2. **Panamá, República de.** Dirección General del Sistema Penitenciario Panameño. *Dirección General del Sistema Penitenciario Panameño.* [En línea] 2004. [Citado el: 10 de Diciembre de 2010.] <http://www.sistemapenitenciario.gob.pa>.
3. **Ecuador, Sistema Nacional de Gestión Penitenciaria.** Ministerio de Justicia, Derechos Humanos y Cultos. *Ministerio de Justicia, Derechos Humanos y Cultos.* [En línea] 2011. [Citado el: 18 de Enero de 2011.] <http://www.minjusticia.gob.ec:8000/mjdh/admin/login.jsp>.
4. **Zaila, Yadira Benavides y Correa, Juan Carlos Gómez.** *Diseño e implementación de los módulos Decisiones y Egresos del Sistema de Gestión Penitenciaria de la República Bolivariana de Venezuela.* Universidad de las Ciencias Informáticas. Ciudad de la Habana, Cuba : s.n., 2008. Tesis de grado.
5. **Jacobson, Ivar, Booch, Grady y otros.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Empresa Poligráfica de Holguín, 2004. pág. 435.
6. Lenguaje Unificado de Modelado. *Lenguaje Unificado de Modelado.* [En línea] 2010. [Citado el: 15 de Diciembre de 2010.] [http://es.wikipedia.org/wiki/Lenguaje\\_Unificado\\_de\\_Modelado](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado).
7. Patrón de diseño. *Patrón de diseño.* [En línea] 2010. [http://es.wikipedia.org/wiki/Patr%C3%B3n\\_de\\_dise%C3%B1o](http://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o).
8. **Ruiz, Roberto Granda y González, Humberto García.** *Diseño e Implementación del módulo Requisas y Decomisos del Sistema de Gestión Penitenciaria de la República Bolivariana de Venezuela.* Universidad de las Ciencias Informáticas. Ciudad Habana, Cuba : s.n., 2010. Tesis de grado.
9. Framework. *Framework.* [En línea] 2010. [Citado el: 15 de Diciembre de 2010.] <http://es.wikipedia.org/wiki/Framework>.
10. Spring Framework. *Spring Framework.* [En línea] 2010. [Citado el: 15 de Diciembre de 2010.] [http://es.wikipedia.org/wiki/Spring\\_Framework](http://es.wikipedia.org/wiki/Spring_Framework).

11. **King, Gavin, Bauer, Christian y otros.** HIBERNATE - Relational Persistence for Idiomatic Java. *Hibernate Reference Documentation*. [En línea] 15 de Septiembre de 2010. [Citado el: 15 de Diciembre de 2010.] <http://docs.jboss.org/hibernate/core/3.5/reference/en-US/html/>.
12. **González, Liesky Díaz.** *Diseño e Implementación del Módulo de Información Histórica de la Sala Situacional del SIGEP*. Universidad de las Ciencias Informáticas. Ciudad de la Habana, Cuba : s.n., 2008. Tesis de grado.
13. Oracle. *Oracle*. [En línea] 2010. [Citado el: 15 de Diciembre de 2010.] <http://es.wikipedia.org/wiki/Oracle>.
14. **Walls, Craig y Breindenbach, Ryan.** *Spring in Action*. 2da Edición. 2007.
15. **Rivero, Iosev Pérez y Pimentel, Luis A.** *ArBaWeb: Arquitectura Base sobre la Web*. Universidad de las Ciencias Informáticas. Ciudad de la Habana, Cuba : s.n., 2007. Tesis de grado.
16. Modelado del software. *Modelado del software*. [En línea] 24 de Enero de 2010. [Citado el: 15 de Diciembre de 2010.] [http://es.wikipedia.org/wiki/Modelado\\_del\\_software](http://es.wikipedia.org/wiki/Modelado_del_software).
17. **King, Gavin y otros, Christian Bauer y.** *Hibernate Reference Documentation*. 2006.