

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



**Título: Diseño e Implementación de los módulos Datos
Generales e Ingreso del Sistema Penitenciario Cubano.**

*Trabajo de diploma para optar por el título de Ingeniero en
Ciencias Informáticas*

Autor: José Manuel Cabrera Gutiérrez

Tutor(a): Ing. Yanet del Risco Batista

Ciudad de la Habana, 17 de junio de 2010

DECLARACIÓN DE AUTORÍA

Declaro ser autor del presente trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos el presente trabajo a los ____ días del mes de _____ del año 2010.

José Manuel Cabrera Gutiérrez
Firma del Autor

Ing. Yanet del Risco Batista
Firma del Tutor

AGRADECIMIENTOS

A mis padres Iliana y Jorge por ser los mejores padres del mundo y darme siempre su apoyo incondicional para poder llegar hasta este momento.

A mi hermana Nene y mi hermanito Jorgito que son lo más lindo que tengo en la vida.

A mi sobrino Joseito por recordarme que soy su tío querido.

A mis abuelas Daris, mamá Lola, Clara y Alba y mis abuelos Papitón, el viejo Rosalía, papá Botica y Juan que han sido mis segundos padres en todo el camino recorrido.

A mis primos y primas que siempre han estado presentes cuando los he necesitado.

A mis tíos y tías que nunca me han defraudado y siempre han sabido darme los mejores consejos del mundo.

A mis amigos que me han acompañado durante toda esta carrera en la que me han brindado las mejores satisfacciones del mundo, Ángel Islán, Danny, Julio, Víctor, Vitali, Leydis, El Chino, Felipe, Yurismara, Irislaydis.

A mis dos compañeros de tesis Pepe (El Yeti) y Yoña (El pinchador) por darme ánimos todos los días y por las madrugadas que pasamos juntos trabajando.

A mi tutora Yanet, que más que una tutora se convirtió en mi amiga y siempre confió en mí y me dio todo su apoyo.

Al tribunal, que somos compañeros de proyecto y más que compañeros amigos.

A todos mis hijos del proyecto, en especial a La Corporación (Leandro) y a la Aldea (Javier) que nunca dijeron que no cuando los necesitaba.

DEDICATORIA

A toda mi familia y amigos, en especial a mis padres Iliana y Jorge, este título es de ustedes por todo el sacrificio que han hecho a lo largo de este camino que no ha sido corto.

Al hombre más grande que he conocido en el mundo, Fidel Castro y a la obra que ha creado, La Revolución.

Resumen

El presente trabajo consiste en el diseño y la implementación de los módulos Datos Generales e Ingreso pertenecientes al subsistema Área de ingreso, observación, evaluación y diagnóstico del Sistema Penitenciario Cubano.

La informatización del Sistema Penitenciario Cubano tiene sus inicios en el año 2002 con la orden 43/99 del Viceministro Primero del Interior que planteaba realizar un sistema para el control de los reclusos. Como cumplimiento a esta orden se diseña, programa e implanta el sistema para el control del recluso (SACORE), utilizando la tecnología existente en ese entonces en el MININT. En sus seis años de explotación se ha ido perfeccionando a partir de los requerimientos y solicitudes de los usuarios.

A partir del Plan 20 x 50 y en colaboración con la Universidad de las Ciencias Informáticas, se decide por parte de la Jefatura del MININT crear el proyecto Prisiones Cuba, el cual tiene como objetivo fundamental unir los tres sistemas que existen en la Dirección de Establecimientos y adicionar un conjunto de funcionalidades necesarias para el funcionamiento óptimo de los procesos en los centros penitenciarios, aprovechando el desarrollo de las tecnologías de la informática y las comunicaciones que se está llevando a cabo en el país.

El presente trabajo de diploma tiene como objetivo realizar el diseño y la implementación de los Módulos Datos Generales e Ingreso del Sistema Penitenciario Cubano (SIGI) y de esta manera exponer todo el proceso de desarrollo y los pasos para obtener los módulos como componentes ejecutables integrados al SIGI. Tomando como punto de partida el análisis de los requerimientos de software establecidos en acuerdo con el cliente y haciendo uso de las tecnologías y herramientas definidas para el proyecto.

Palabras claves: SIGI, Control Penal, Reclusos, SACORE.

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica.....	3
1.1 <i>Introducción</i>	3
1.2 <i>Marco Conceptual</i>	3
1.2.1 Sistema Penitenciario	3
1.2.2 Acusado:	3
1.2.3 Sancionado:.....	3
1.2.4 Asegurado:.....	3
1.3 <i>Sistemas informáticos dentro de los sistemas penitenciarios.</i>	8
1.3.1 OFFENDERTRAK Corrections Management System:	8
1.3.2 Sistema de Gestión Penitenciaria (SIGEP):	8
1.4 <i>SIGI (Sistema de Gestión del Interno)</i>	10
1.5 <i>Tecnologías y herramientas utilizadas en el desarrollo</i>	12
1.5.1 Herramientas de modelado	12
1.5.2 Herramientas de desarrollo	13
1.5.3 Sistemas Gestor de Base de Datos	13
1.5.4 Tecnologías	13
1.5.5 Metodologías de Desarrollo.....	14
1.6 <i>Conclusiones</i>	16
Capítulo 2. Características del Sistema	16
2.1 <i>Introducción</i>	16
2.2 <i>Arquitectura del SIGI</i>	16
2.3 <i>La aplicación estará dividida en capas lógicas fundamentales:</i>	16
2.4 <i>Seguridad</i>	18
2.5 <i>Conclusiones</i>	18
Capítulo 3. Diseño e Implementación del Sistema	25
3.1 <i>Introducción</i>	25
3.2 <i>Patrones de Diseño</i>	25
3.3 <i>Diseño de la solución</i>	26
3.3.1 Descripción del módulo Ingresos	26

3.3.2 Descripción del módulo Datos Generales.....	26
3.3.3 Análisis de las funcionalidades.....	27
3.3.4 Diseño del Dominio	31
3.3.5 Diseño del modelo de datos.	38
3.3.6 Diseño de la Arquitectura en Capas.	54
3.3.7 Implementación de las clases de dominio.....	55
3.3.8 Implementación de las interfaces	50
3.3.9 Implementación de los controladores.....	51
3.3.10 Implementación de la capa de datos.....	52
3.3.11 Implementación de la lógica en el cliente.	51
Conclusiones.....	54
Recomendaciones.....	55
Referencias Bibliográficas	56
Bibliografía	56
Glosario de Términos.....	68
Anexos 1.....	56
Anexo 2.Validaciones.....	63
Anexo 3.Implementaciones.....	65
Anexo 4.Diagrama de clases persistentes.....	70

Introducción

La automatización del Sistema Penitenciario a nivel nacional comienza en el año 1989, con un sistema de control penal, que permitía la gestión de los principales datos del recluso y algunos aspectos del trabajo de la especialidad. En cumplimiento de la orden 43/99 del Viceministro Primero del Interior se diseña, programa e implanta el sistema para el control del recluso (SACORE), utilizando la tecnología existente en ese entonces en el MININT (Computadoras 486, sin conectividad interna ni externa, programación en Developer y Gestor Base Datos ORACLE), tarea que se concluye en el año 2002, comenzando la explotación en enero del 2003.

En el año 2008 el Sistema Penitenciario Nacional se adentra en una transformación organizacional que permite mejorar y optimizar los procesos fundamentales para el trabajo con los privados de libertad, la cual trae como principal resultado una modificación en la documentación Legal (Reglamento Penitenciario y Procedimiento de Trabajo de las Especialidades).

A partir del Plan 20 x 50 y en colaboración con la Universidad de las Ciencias Informáticas, se decide por parte de la Jefatura del MININT crear un proyecto de conjunto con la Universidad de Ciencias Informáticas, el cual tiene como objetivo fundamental unir los tres sistemas que existen en la Dirección de Establecimientos y adicionar un conjunto de funcionalidades necesarias para el funcionamiento óptimo de los procesos en los centros penitenciarios, aprovechando el desarrollo de las tecnologías de la informática y las comunicaciones que se está llevando a cabo en el país.

El sistema penitenciario cubano presenta dificultades en el proceso de ingreso porque no existe la posibilidad de efectuar una búsqueda avanzada lo que trae como consecuencia que a veces se registre el mismo interno dos veces, motivo por el cual se generan demoras en el momento de realizar el ingreso, otro proceso que presenta dificultades es creación del plan de trabajo individual (PTI) del interno por no existir una recogida que abarque la mayor cantidad de los datos personales del interno, también existen deficiencias en los procesos de trabajo con los tribunales por no contar con la digitalización de todos los documentos relacionados con el interno. La recogida de los datos personales del interno no se hace lo más exhaustiva posible, faltándole datos de interés como pueden ser, los otros nombres que pueda poseer el interno, así como las otras direcciones particulares en las que se pueda localizar. También existe la dificultad que no guardan en formato digital las imágenes de las señas particulares y las reseñas decadactilares. Otros de los datos que no se recogen en estos momentos son los oficios que poseen el interno, los idiomas y las profesiones y no existe digitalizado la posibilidad de verificar estos datos.

Tomando como base la situación problemática anterior se plantea que el **problema a resolver** es el siguiente: ¿Cuál es el diseño e implementación de los módulos Datos Generales e Ingreso, a partir de los requisitos de software determinados, teniendo en cuenta la arquitectura y tecnologías definidas para el proyecto SIGI?

El **objetivo general** de este trabajo constituye: Diseñar e implementar los módulos Datos Generales e Ingresos del SIGI a partir del análisis de los requisitos de software establecidos en acuerdo con el cliente, haciendo uso de las tecnologías y herramientas definidas para el proyecto. Teniendo en cuenta todo lo anterior se define como **Objeto de estudio**:

- El proceso de desarrollo de los módulos Datos Generales e Ingresos del Sistema Penitenciario Cubano.

Dentro del objeto de estudio anterior se define como **Campo de Acción**:

- Los modelos de Diseño e Implementación de los módulos Datos Generales e Ingresos del Sistema Penitenciario Cubano.

Para la realización correcta de este trabajo se plantean las siguientes **Tareas Investigativas**:

- Análisis de los requisitos de software y el modelo de negocio correspondientes a los módulos de Datos Generales e Ingreso.
- Estudio de las tecnologías y herramientas definidas por el proyecto.
- Diseño de la solución de software para los requisitos relacionados con los Módulos Datos Generales e Ingreso.
- Implementación de la solución de software para los requisitos relacionados con los Módulos Datos Generales e Ingreso.

Capítulo 1: Fundamentación Teórica.

1.1 Introducción

Este capítulo constituye la fundamentación teórica sobre la que se desarrolla el presente trabajo. Se abordan los conceptos fundamentales del Sistema Penitenciario Cubano que se referencian como parte de la situación problemática y a lo largo del trabajo. Además se analizan las tecnologías, herramientas y tendencias de desarrollo a utilizar en el diseño e implementación de los procesos datos Generales e Ingresos del Sistema Penitenciario cubano, en el marco del desarrollo del Sistema de Control del Interno (SIGI).

1.2 Marco Conceptual

1.2.1 Sistema Penitenciario: Es el encargado de garantizar el proceso de ejecución de la sanción de privación de libertad, de la sanción de trabajo correccional con internamiento, la medida de seguridad reeducativa de internamiento y la medida cautelar de prisión provisional. (15) Este sistema, regido por la Dirección de Establecimientos Penitenciarios del Ministerio del Interior, se sustenta en la integración de principios, conceptos, procedimientos, fuerzas y medios que garantizan el funcionamiento de los centros destinados al internamiento y el tratamiento a los internos.

1.2.2 Acusado: Persona a quien se le haya decretado la medida cautelar de prisión provisional. (15)

1.2.3 Sancionado: Persona ejecutoriamente sancionada a privación de libertad o a trabajo correccional con internamiento. (15)

1.2.4 Asegurado: Persona a quien se le hubiere impuesto una medida de seguridad reeducativa de internamiento. (15)

1.3 Sistemas informáticos dentro de los sistemas penitenciarios.

En la región de América Latina, así como en algunos países de Europa la poca organización y la realización manual de la mayoría de los procesos son características comunes que podemos encontrar dentro de los sistemas penitenciarios. A raíz de estos problemas en algunos países se han realizado esfuerzos por tratar de automatizar la mayor parte de estos procesos y a la vez lograr que estos sistemas sirvan de apoyo en la tomas de decisiones a un nivel centralizado .A continuación se abordan dos de estos sistemas :

1.3.1 OFFENDERTRAK Corrections Management System: Es una solución de Software empresarial proporcionada por Motorola, aplicable en departamentos de policía y centros penitenciarios, la cual permite:

- Registrar los datos personales de los individuos.
- Clasificar a los individuos atendiendo a distintos criterios.
- Mantener identificados biométricamente a los individuos capturando las huellas de los dedos y de la palma de la mano de los individuos y efectuar la verificación de identidad.
- Registrar señas particulares de los individuos con fotos y descripción.
- Registrar rasgos físicos de los individuos.

1.3.2 Sistema de Gestión Penitenciaria (SIGEP): permite a los establecimientos penitenciarios (Internados Judiciales, Centros Penitenciarios, Centros de Tratamiento Comunitario) y otras sedes (Coordinaciones Regionales y UTASP), recopilar y controlar la información operativa que se genera en este tipo de centros. Esta solución controla el tránsito de los privados de libertad por todos los componentes del sistema penitenciario, y en general maneja datos sobre la población penal y auditará los procesos legales para garantizar un cumplimiento justo de la sentencia. El sistema incluye además la gestión de los servicios médicos y alimenticios y el tratamiento ofrecido en estos establecimientos penitenciarios .Dentro de sus servicios, según Arias (3) podemos encontrar los siguientes:

1. Mantener identificado al privado de libertad en su tránsito por el sistema penitenciario.
2. Conocer la cantidad de privados de libertad, su ubicación y situación legal.
3. Conocer el comportamiento de los ingresos y egresos al sistema penitenciario. Pronosticar las libertades y libertades anticipadas.

4. Controlar el estricto cumplimiento de las disposiciones legales: ejecución de la sanción y otorgamiento de fórmulas alternativas de cumplimiento de pena, otorgamiento de beneficios.
5. Controlar la ejecución de la pena y las presentaciones a juicio.
6. Conocer y valorar la incorporación al trabajo, el estudio y las actividades educativas.
7. Registrar las personas vinculadas al interno y los tipos de influencia.
8. Controlar los planes de rehabilitación, entrevistas y diagnósticos psicosociales.
9. Controlar la aplicación de medidas disciplinarias y suspensiones de derechos.
10. Planificar y controlar las requisas y sus resultados.
11. Planificar y controlar las visitas.
12. Controlar las salidas transitorias.

¿Por qué si el SIGEP brinda todas estas facilidades no se utiliza en el sistema penitenciario cubano?

Entre las causas principales por las que se decide construir un nuevo software para el sistema penitenciario cubano está el Reglamento Legal, este reglamento cambia completamente de acuerdo a las leyes de cada país. Otra causa por la cual no se puede utilizar el SIGEP es que aunque Cuba tiene el derecho sobre este software solo puede ser utilizado por Venezuela.

¿En qué estado se encuentra la automatización de los procesos de Ingreso y Datos Generales del SIGI?

Aunque se ha logrado que el Sistema Informativo existente en los tres niveles de mando sea el resultado del sistema automatizado implantado, todavía existen dificultades en los procesos mencionados, entre estas deficiencias se pueden observar las siguientes:

- Dificultades en los procesos de ingresos.
- Demoras en los procesos de trabajo con los tribunales.
- No se cuenta con una búsqueda avanzada en el momento de registrar el ingreso de un interno.

- No se registran otros nombres que pueda tener el interno.
- No se registran otras direcciones que puede poseer el interno.
- No se guardan en formato digital las imágenes de las señas particulares y las huellas dactilares.
- No se registran los oficios del interno.
- Existen registros dobles de un mismo interno.

Teniendo como base la aplicación del SIGEP en Venezuela y la necesidad de dar respuesta a las dificultades existentes dentro de los procesos que se desarrollan en el sistema penitenciario cubano surge la idea de crear un sistema que automatice la mayor cantidad posible de los procesos que hoy se realizan a través del software SACORE (Sistema para el control del recluso) o manualmente. Como respuesta a lo antes mencionado se comienza a desarrollar el SIGI (Sistema de Gestión del Interno).

1.4 SIGI (Sistema de Gestión del Interno)

Aprovechando el desarrollo de la informática y las comunicaciones en el cual el país está inmerso en estos momentos se decide desarrollar un sistema que informatice los procesos que se llevan a cabo dentro del Sistema Penitenciario Nacional entre los cuales se pueden destacar, el control, tratamiento y atención a los acusados, asegurados y sancionados en los centros penitenciarios y los correspondientes a los tres niveles de mando, incluyendo las relaciones con otros Organismos de la Administración Central del Estado. Este sistema tiene como objetivos específicos:

1. Garantizar el proceso de ingreso de todos los acusados, asegurados y sancionados remitidos por los órganos instrucción, fiscalías y tribunales al sistema penitenciario.
2. Garantizar la correcta evaluación y clasificación en régimen de los internos.
3. Lograr el control de los procesos jurídicos, trámites legales, otorgamiento de beneficios y seguridad de los internos en los centros penitenciarios.
4. Informatizar el funcionamiento de los consejos de dirección que se realizan en los centros penitenciarios.

5. Proveer a los especialistas de tratamiento educativo de una herramienta que les permita realizar un trabajo profundo y detallado en la preparación para la reinserción social de los internos: incorporación al estudio, trabajo, actividades culturales, políticas, recreativas, etc.
6. Velar por el cumplimiento del pago de los adeudos de los internos a la caja de resarcimiento para el pago de la responsabilidad civil derivada por la comisión del delito.
7. Lograr una adecuada gestión de los trámites legales de los internos mediante la integración con los diferentes órganos de justicia.
8. Garantizar información operativa y oportuna para la toma de decisiones estratégicas a los tres niveles de mando.
9. Mantener identificado al interno en su tránsito por el sistema penitenciario.
10. Conocer la cantidad de acusados, asegurados y sancionados, así como su ubicación y situación legal.
11. Registrar las personas vinculadas al interno y los tipos de influencia que ejercen sobre el mismo.
12. Controlar la aplicación de medidas disciplinarias y suspensión de beneficios.
13. Controlar las requisas, ocupaciones y hallazgos a los internos en los centros penitenciarios.
14. Realizar la planificación de visitas familiares, institucionales y pabellones, así como realizar el control de la ejecución de las mismas.
15. Administrar las capacidades, conocer las disponibilidades y afectaciones de la infraestructura física de las instalaciones.
16. Controlar las incidencias que ocurren en los establecimientos penitenciarios.
17. Elaborar partes informativos para los tres niveles de mando.
18. Controlar el estado de salud del interno.
19. Controlar las consultas, chequeos médicos, planes de vacunación y análisis clínicos (programados y no programados) y control de enfermedades.

Dentro del SIGI existe el subsistema **Área de ingreso, observación, evaluación y diagnóstico** que es el que abarca todos los procesos relacionados con el ingreso y los datos personales de los internos. Dentro de este subsistema se encuentran los módulos Datos Generales e Ingreso, módulos en los cuales está enmarcado el presente trabajo.

1.5 Tecnologías y herramientas utilizadas en el desarrollo.

La selección correcta de las de las herramientas y tecnologías que se utilizan en el desarrollo de un software se traduce en ahorro de tiempo y de trabajo dentro del proyecto. A continuación se realiza una breve descripción de los instrumentos definidos por parte del equipo de arquitectura del SIGI.

1.5.1 Herramientas de modelado

➤ **Visual Paradigm Suite 3.1:** Herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

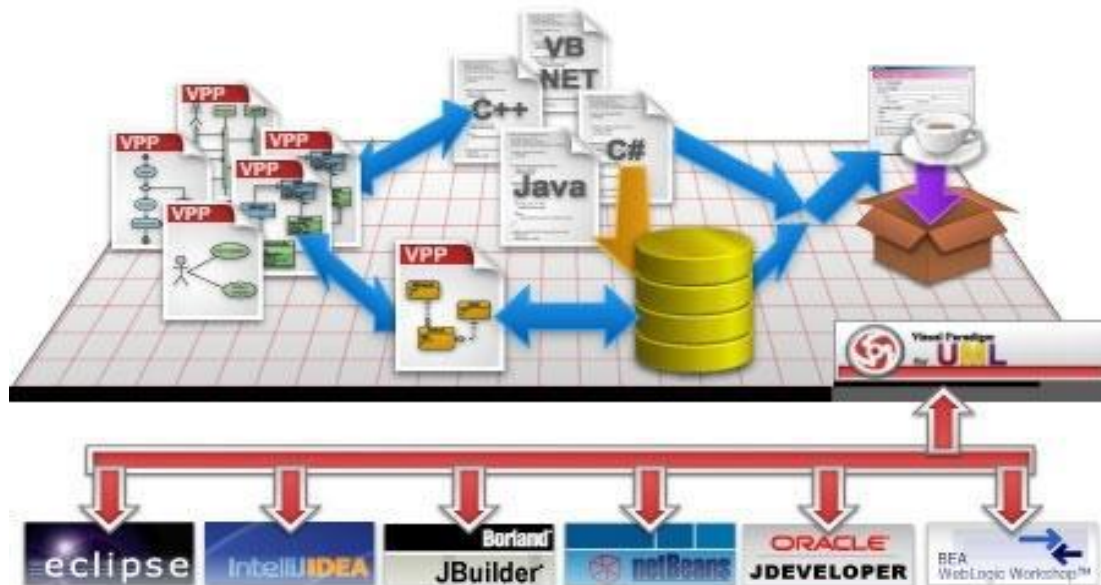


Figura 1. Visual Paradigm Suite

1.5.2 Herramientas de desarrollo

- **NetBeans 6.8 Milestone2:** Es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. Cuenta con un compilador de Java interno y un modelo completo de los archivos fuente de Java. El IDE también hace uso de un espacio de trabajo, en este caso un grupo de datos en un espacio para archivos plano, permitiendo modificaciones externas a los archivos en tanto se refresque el espacio de trabajo correspondiente, emplea módulos para proporcionar toda su funcionalidad. Puede extenderse usando otros lenguajes de programación como son C/C++ y Python y trae incluido entre sus novedades más destacadas el uso de Groovy.
- **Contenedor Web (Apache Tomcat):** Apache Tomcat es un contenedor Web, desarrollado en un ambiente participativo y abierto. Apache Tomcat es usado en numerosas aplicaciones web de gran escala y críticas en diversas industrias y organizaciones que se referencian en su sitio oficial.
- **Subversion:** Subversion es un sistema de control de versiones que administra el acceso a un conjunto de ficheros, y mantiene un historial de cambios realizados. Es software libre bajo una licencia de tipo Apache/BSD y se lo conoce también como svn por ser ese el nombre de la herramienta de línea de comandos.

1.5.3 Sistemas Gestor de Base de Datos

- **Oracle:** Es un sistema gestor de base de datos líder en la industria, utilizable para almacenar todo tipo de datos, incluyendo datos relacionales, documentos, multimedia, XML y datos de localización. Oracle está definido como el gestor de base de datos a utilizar para el desarrollo y despliegue del SIGI debido a que ofrece una excepcional disponibilidad, escalabilidad, fiabilidad y seguridad y por ser el sistema gestor de bases de datos definido por parte de la dirección del MININT. 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.

1.5.4 Tecnologías

Framework de desarrollo Grails:

Es una plataforma para desarrollo de aplicaciones web basada en el lenguaje de tipado dinámico Groovy, se compila y ejecuta en una JVM (Java Virtual Machine). Su desarrollo fue inspirado en

SmallTalk, Python y Ruby (5) .Entre sus ventajas se encuentran que se integra con el lenguaje Java en cualquier nivel, además incorpora componentes que van desde los flujos webs hasta la capa de datos. El lenguaje de programación que se utiliza es Groovy, el mismo está basado en herramientas existentes y poderosas como: Spring e Hibernate, entre otras.

➤ **Spring MVC:** El modelo MVC es muy utilizado en la construcción de aplicaciones web. Spring se integra con Frameworks de soporte MVC como Struts, JSF, WebWork y Tapestry, pero también provee su propia implementación del modelo MVC basada en el patrón Controlador Frontal.

➤ **JSON:** JSON, acrónimo de (JavaScript Object Notation), es un formato ligero para el intercambio de datos, es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

➤ **Dojo Toolkit:** Framework AJAX o librería de clases (*Asynchronous JavaScript + XML* o JavaScript asíncrono + XML) el cuál brinda una variedad de clases y widgets para facilitar el desarrollo de aplicaciones web. Dojo puede ser interpretado por diferentes navegadores web y posee un sistema de empaquetado muy parecido al del JDK de java. La gran variedad de clases componentes y widgets, es un punto muy importante en la elaboración en la capa presentación de la aplicación, posibilita a los programadores de interfaz una serie de funcionalidades y elementos dinámicos que facilita la programación en el cliente.

1.5.5 Metodologías de Desarrollo

Rational Unified Process (RUP)

Su objetivo es asegurar la construcción de sistemas de software de alta calidad que satisfagan las necesidades de los usuarios finales y clientes cumpliendo con los cronogramas y presupuestos previstos. Es adecuada para sistemas con extensos cronogramas y equipos de desarrollo numerosos. RUP es un proceso que se caracteriza por ser según JACOBSON (7) :

➤ **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. Cada iteración añade funcionalidades al producto de software o mejora las existentes. En la Figura 1 se muestran los flujos de trabajo y las iteraciones que define RUP por las cuales debe transitar el desarrollo de un producto software. Además RUP identifica una serie de roles para los trabajadores que realizan las actividades de cada flujo, dentro de los principales roles se encuentran: ingeniero de requisitos, diseñador, arquitecto, implementador, jefe de proyecto y probador.

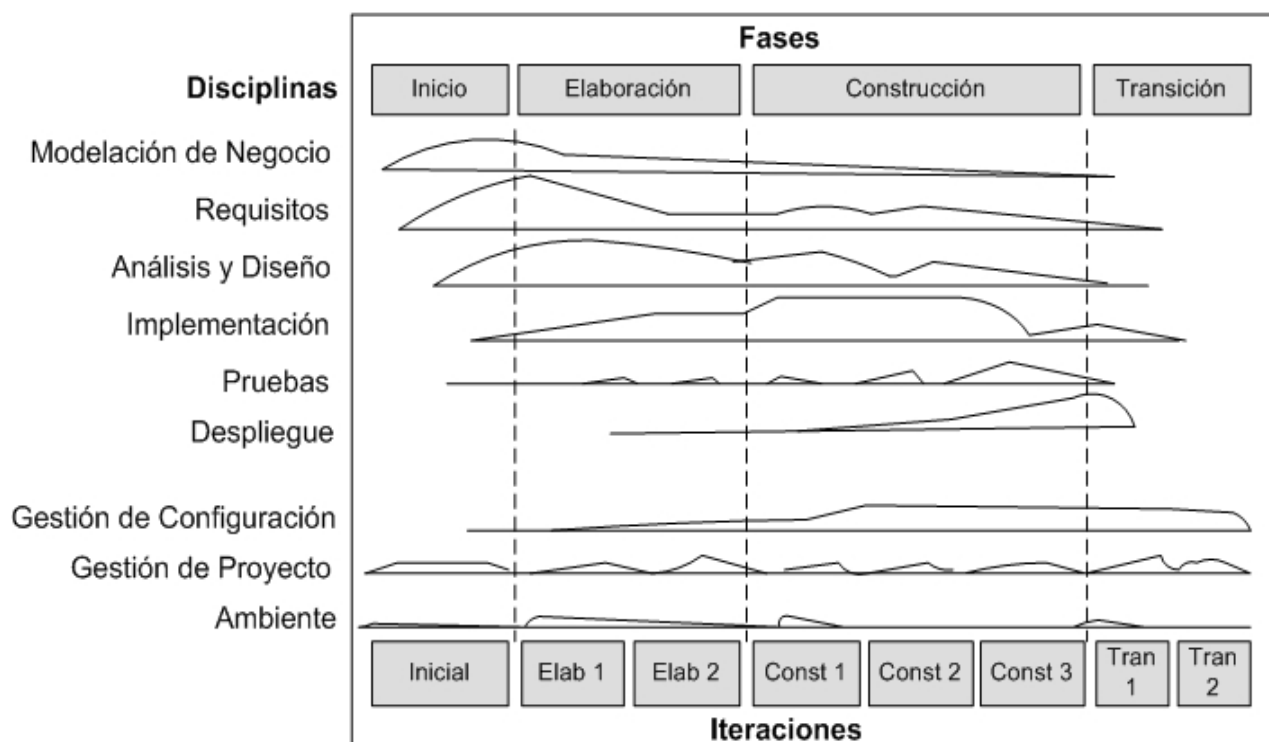


Figura 2. Fases y flujos de trabajo en RUP (RATIONAL SOFTWARE CORPORATION, 2003).

Los flujos de trabajo representados en la Figura 1 proponen la realización de un gran número de actividades y la elaboración de un amplio conjunto de artefactos, que usualmente los proyectos de desarrollo de software se comprometen a desarrollar pero que no realizan en su totalidad debido a la carencia de tiempo o a que descubren que no eran necesarios. 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.

1.6 Conclusiones

En este capítulo se abordaron conceptos relacionados con el sistema penitenciario cubano y se mostro brevemente la solución informática que se desarrolló para la gestión del sistema penitenciario de Venezuela (SIGEP). Además en este capítulo se hace una breve descripción de las herramientas y metodologías definidas por parte del proyecto para la realización de este software.

Capítulo 2. Características del Sistema

2.1 Introducción

En este capítulo se brinda una descripción de la arquitectura sobre la que se desarrollan los módulos Datos Generales e Ingresos pertenecientes al SIGI. Se abordan las capas por las que está compuesta y se brinda un breve análisis de estas. Se explica el flujo de actividades a utilizar en el diseño e implementación de un módulo en el SIGI.

2.2 Arquitectura del SIGI

La arquitectura del proyecto SIGI está basada principalmente en las funcionalidades y facilidades que propone el framework de desarrollo Grails ([ver](#)). Se definirá un sistema con arquitectura en capas, Capa de Presentación, Capa de Servicios y Capa de Datos ([ver fig. 3](#)). Además está instalada sobre la tecnología Cliente-Servidor. ([ver fig.4](#))

- **Cliente:** La aplicación se ejecuta a través de un navegador de Internet instalado sobre cualquier sistema operativo, aunque se recomienda que para las PC clientes que tengan conectados algún dispositivo (cámara digital o escáner de huellas) tengan instalado el sistema operativo Windows XP o superior.
- **Servidor Web:** En este servidor radica la lógica de negocio de la aplicación. Sistema operativo Linux SuSE Enterprise Server 10.0, para los servidores del centro de datos y alguna distribución de Linux para los servidores locales de los centros penitenciarios. Servidor Web Apache Tomcat versión 6.0 Java Runtime Environment (JRE), versión 1.6.
- **Servidor de Base de Datos:** Sistema operativo Linux SuSE Enterprise Server 10.0, para los servidores del centro de datos y alguna distribución de Linux para los servidores locales de los establecimientos penitenciarios. Servidor de base de datos Oracle 11g Enterprise Edition.

2.3 La aplicación estará dividida en capas lógicas fundamentales:

- **Capa Web:** En esta capa se encuentran las Vistas y la Lógica de Presentación. En la Lógica de Presentación se maneja todo el flujo web utilizando la implementación que brinda Grails. Las Vistas son los recursos que junto al modelo generado por los controladores le permiten al cliente visualizar la información.

- **Capa de Servicios:** Encapsula toda la lógica de la aplicación del negocio y que son utilizadas por los controladores en la capa de presentación y se exponen algunos procesos de negocio a través de interfaces de servicios.
- **Capa de Datos:** Maneja los objetos de acceso a datos abstrayéndolos del mecanismo de persistencia usado; a través de interfaces que exponen las operaciones de persistencia definidas para cada uno de las implementaciones utilizando Hibernate, que es un framework de mapeo a objeto-relacional (ORM).

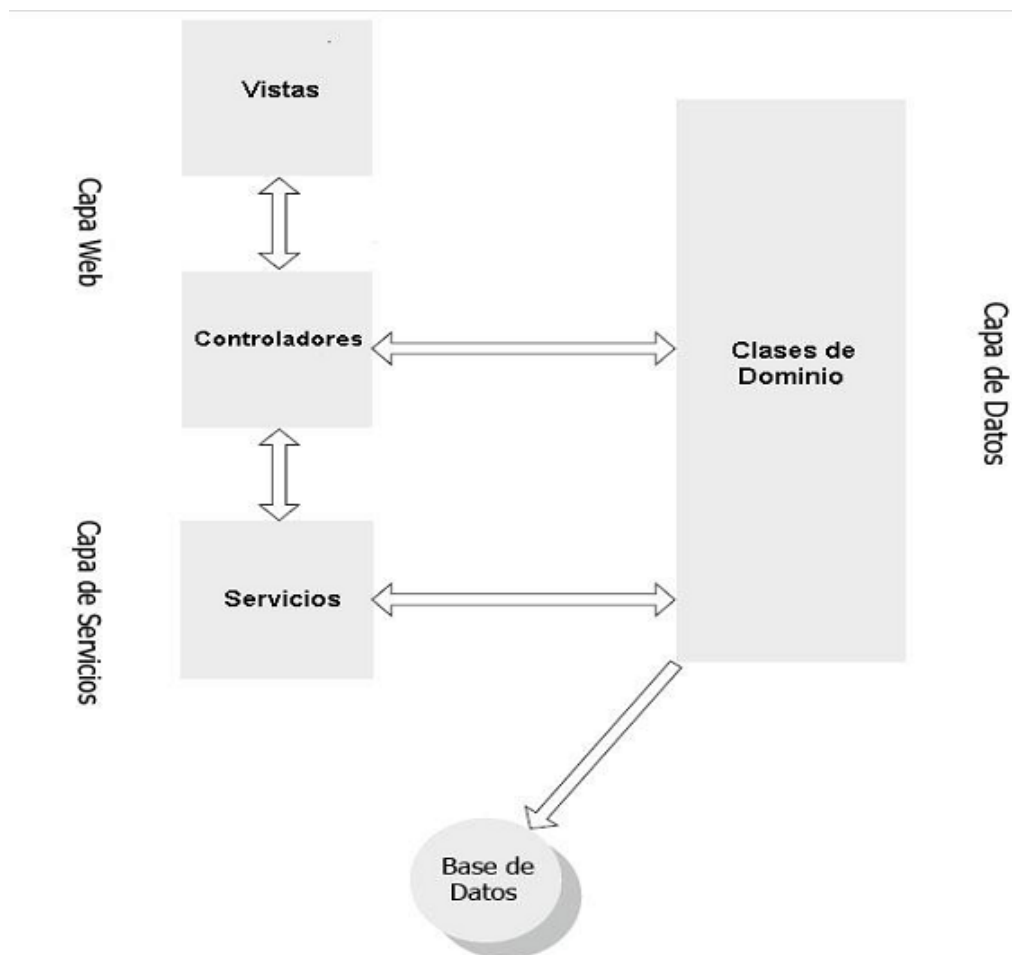


Figura 3. Arquitectura de una aplicación en Grails.

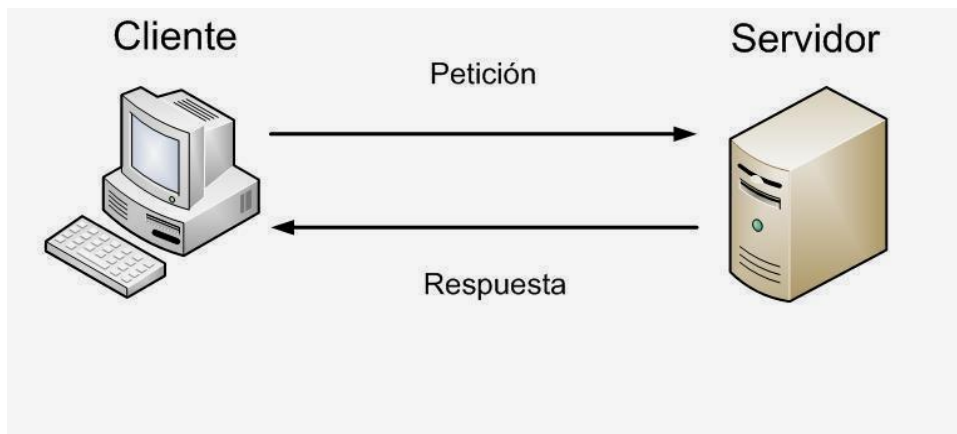


Figura 4. Modelo Cliente- Servidor

2.4 Seguridad

La seguridad es uno de los temas que se prioriza en todo sistema informático, pero aún más en los proyectos de atención especial como es el caso del MININT. Este asunto se maneja a través del framework Spring Security incorporado en Grails y para la transmisión de datos el canal seguro (HTTPS), además de ello se agrega la seguridad por Roles. También se controlará el acceso a los servidores del Ministerio, que estarán protegidos por sistemas cortafuegos (firewalls) y servidores de autenticación, que sólo permitan la entrada a los usuarios y máquinas autorizadas.

2.5 Conclusiones

En este capítulo se aborda de manera general la arquitectura propuesta para el desarrollo de los módulos Ingreso y Datos Generales haciendo una breve descripción del Frameworks Grails, así como una síntesis de cómo se puede observar la seguridad dentro del sistema.

Capítulo 3. Diseño e Implementación del Sistema

3.1 Introducción

En este capítulo se muestra la solución dada a los módulos de Datos Generales e Ingresos de acuerdo a su Diseño e Implementación. Se utiliza como lógica de presentación en el capítulo la secuencia de actividades realizadas, en correspondencia con las definidas para la realización de un módulo dentro del SIGI. En este capítulo no se exponen todos los diagramas e implementaciones hechas sino solo una muestra representativa de las funcionalidades distintivas de ambos módulos.

3.2 Patrones de Diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Estos se caracterizan por estar conformados por un conjunto de elementos, como: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (8). En el diseño y la implementación realizada en los módulos Ingreso y Datos Generales se utilizaron algunos patrones para brindar y ayudar a desarrollar una solución más robusta y escalable. De los patrones Grasp se utilizaron de forma general los patrones Alta Cohesión, Bajo Acoplamiento, Controller y Front-Controller. El patrón Controller 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.

Cada clase de Grails promueve el uso de este patrón como parte de los convenios del Frameworks. A su vez el patrón Front-Controllers propone utilizar un controlador como el punto inicial de contacto para manejar las peticiones del usuario en una aplicación. El controlador maneja el control de peticiones, incluyendo la invocación de los servicios de seguridad como la autenticación y autorización, la elección de una vista apropiada, el manejo de errores, y el control de la selección de estrategias de creación de contenido. Grails, como Frameworks de aplicaciones web y con la visión de convertirse en un marco de trabajo altamente productivo no puede estar ajeno al uso de patrones por lo que utiliza como principales paradigmas en este sector dos patrones fundamentales: Convención sobre Configuración. Este patrón no es más que un paradigma de programación de software que busca disminuir el número de decisiones que un desarrollador necesita hacer, ganando así en simplicidad pero no perdiendo flexibilidad por ello y DRY es una filosofía de definición de procesos que promueve la reducción de la duplicación. Ambos patrones en general proporcionan un entorno de desarrollo

estandarizado y ocultan en gran parte detalles de configuración. Con la utilización del Frameworks Spring como componente de Grails se utilizan diferentes patrones de dicho Frameworks que aportan un mejor diseño e implementación al sistema.

3.3 Diseño de la solución

Los módulos Ingreso y Datos Generales forman parte del subsistema Área de ingreso, observación, evaluación y diagnóstico y además tienen relación con el módulo Situación Legal del subsistema Registro legal.

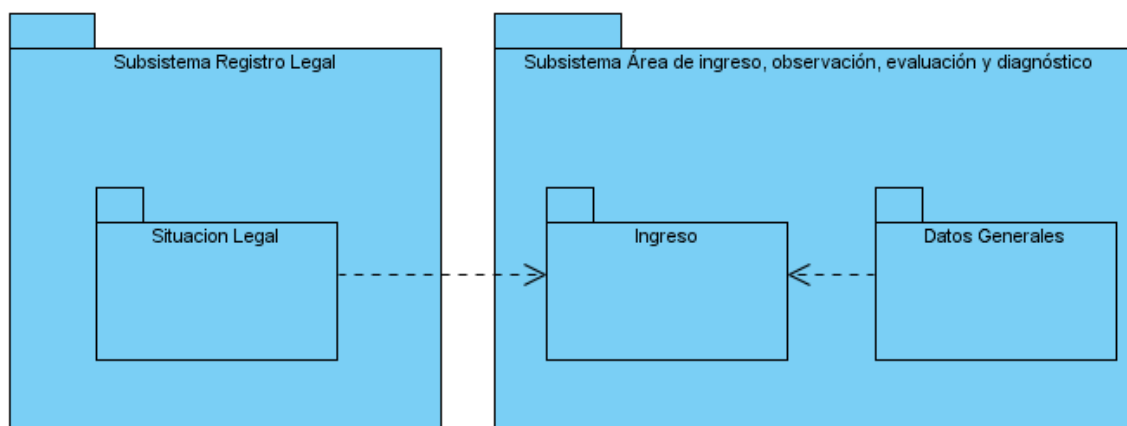


Figura 5.Relacion entre subsistemas.

3.3.1 Descripción del módulo Ingresos

Este módulo permite registrar los datos del ingreso de un acusado, sancionado o asegurado. El ingreso se realiza en las prisiones receptoras y puede tratarse de un ingreso o de un reingreso. Para realizar el ingreso se debe contar con la documentación legal del acusado, sancionado o asegurado proveniente de los órganos de justicia. El módulo debe permitir ver la ficha de los internos con expedientes creados en el sistema penitenciario así como buscar el expediente que se necesite.

3.3.2 Descripción del módulo Datos Generales

El módulo Datos Generales permite gestionar los datos personales de los acusados, asegurados sancionados: datos familiares, dirección particular, señas particulares, rasgos físicos, reseña fotográfica y reseña decadaclares. La reseña fotográfica puede ser actualizada y se mantiene un registro histórico de las fotografías de los cuatro perfiles (frente, perfil derecho, perfil izquierdo y cuerpo

entero) que se le han tomado al interno durante toda su estancia en el centro penitenciario. Las señas particulares se refieren a cicatrices, amputaciones, deformaciones, tatuajes, lunares u otras deformaciones o marcas que su ubicación en el cuerpo y su descripción. De las señas particulares se pueden registrar imágenes fotográficas.

3.3.3 Análisis de las funcionalidades

A partir de las funcionalidades descritas para los módulos Datos Generales e Ingresos en la documentación generada por la captura de requisitos resultó que las funcionalidades a implementar para ambos módulos son:

Módulo Datos Generales:

- Insertar datos personales.
- Insertar dirección.
- Actualizar datos personales.
- Actualizar dirección.
- Insertar datos de los familiares.
- Actualizar datos de los familiares.
- Tomar decadactilar.
- Insertar señas particulares.
- Eliminar señas particulares.
- Registrar reseña fotográfica.
- Mostrar histórico de fotos.
- Insertar rasgos físicos.
- Actualizar rasgos físicos.
- Tomar foto mediante dispositivo.

- Mostrar señas particulares.
- Actualizar señas particulares.
- Insertar idiomas.
- Insertar ciudadanías.
- Insertar oficios.
- Insertar profesiones.

Módulo Ingreso:

- Buscar expediente.
- Registrar datos del ingreso.
- Registrar datos del reingreso.
- Mostrar ficha del interno.

Módulo Ingreso

Nº	Funcionalidad	Descripción
RF1.1	Buscar expediente	El Oficial de Control Penal debe tener la posibilidad de buscar expediente del acusado, sancionado o asegurado que arribe al centro.
RF1.2	Registrar datos del ingreso	El Oficial de Control Penal debe ingresar los datos del ingreso de manera que el sistema cree un expediente legal nuevo al interno.
RF1.3	Registrar datos del reingreso	El Oficial de Control Penal debe ingresar los datos del reingreso actualizando el expediente legal abierto que tenga el interno.

RF1.4	Mostrar ficha del interno	El sistema debe permitir ver la ficha de un interno cuando es buscado.
-------	---------------------------	--

Módulo Datos Generales:

Nº	Funcionalidad	Descripción
RF1	Insertar datos personales.	Permite insertar todos los datos personales del acusado, sancionado o asegurado que no hayan sido introducidos durante el ingreso.
RF2	Insertar dirección.	Permite insertar la dirección particular del acusado, sancionado o asegurado, la que aparece en el CI.
RF3	Actualizar datos personales.	Permite actualizar los datos personales del interno, ya sea durante el reingreso o en cualquier otro momento.
RF4	Actualizar dirección.	Permite actualizar la dirección del interno.
RF5	Insertar datos de los familiares	Permite insertar los datos de los familiares del acusado, sancionado o asegurado.
RF6	Actualizar datos de los familiares	Permite actualizar los datos de los familiares en caso que lo requiera.
RF7	Tomar decadactilar.	Permite tomar decadactilar del acusado, sancionado o asegurado.
RF8	Insertar señas particulares.	Permite insertar las señas particulares que tenga el acusado, sancionado o asegurado, ya sean tatuajes, cicatrices, etc.

Capítulo III. Diseño e Implementación del Sistema

RF9	Eliminar señas particulares.	Permite eliminar las señas particulares del interno.
RF10	Registrar reseña fotográfica.	Permite registrar las fotos de un sancionado, asegurado o acusado (frente, lado derecho, lado izquierdo, mejor ángulo).
RF11	Mostrar histórico de fotos.	Permite mostrar el histórico de las fotos de las cuatro formas tomadas al interno.
RF12	Insertar rasgos físicos.	Permite insertar los rasgos físicos del interno atendiendo a las 23 características.
RF13	Actualizar rasgos físicos.	Permite actualizar los rasgos físicos del interno.
RF14	Tomar foto mediante dispositivo.	Permite tomar fotos del acusado, sancionado o asegurado, ya sea las fotos cuatro fotos (cuerpo entero, frente, lado izquierdo, lado derecho) como las fotos de las señas particulares del interno.
RF15	Mostrar señas particulares.	El sistema debe posibilitar ver todas las señas particulares identificadas en el interno.
RF16	Actualizar señas particulares.	Permite actualizar los datos de una seña particular registrada en un interno.
RF17	Insertar idiomas.	Permite insertar los idiomas que domina el interno.
RF18	Insertar ciudadanías	Permite insertar las ciudadanías que tenga el interno.
RF19	Insertar oficios	Permite insertar los oficios que tenga en interno.
RF20	Insertar profesiones	Permite insertar las profesiones que tenga el interno.

A continuación se muestra el diagrama de clases de diseño para el caso de uso Registrar Datos Personales del módulo Datos Generales. Para ver otros diagramas ir a ([Anexos 1](#))

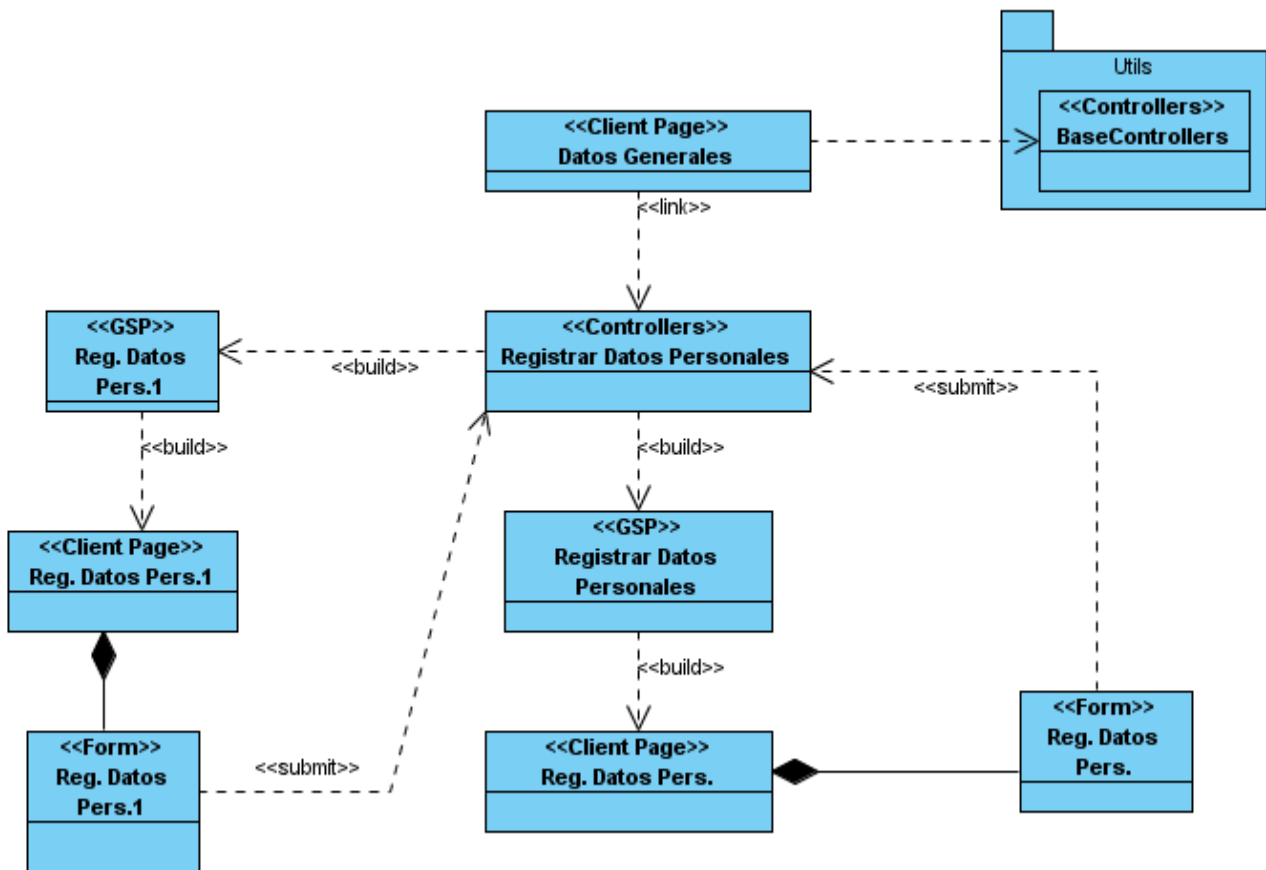


Figura 5. Diagrama de diseño del caso de uso registrar datos personales.

3.3.4 Diseño del Dominio

En el dominio del sistema se van a encontrar el conjunto de entidades persistentes y no persistentes definidas entre las que se puede observar la entidad Interno, Direcciones, Galería de Foto, que van ser manejadas por todas las capas y que constituyen la representación estática de la institución en el sistema informático.

Los objetos de dominio no presentan lógica de negocio, sino que esta responsabilidad recae sobre los objetos de negocio los cuales a partir de ahora se tratarán como manager. Esto permite utilizar a los

objetos de dominio como contenedores de información que se mueven entre las capas arquitectónicas de la aplicación.

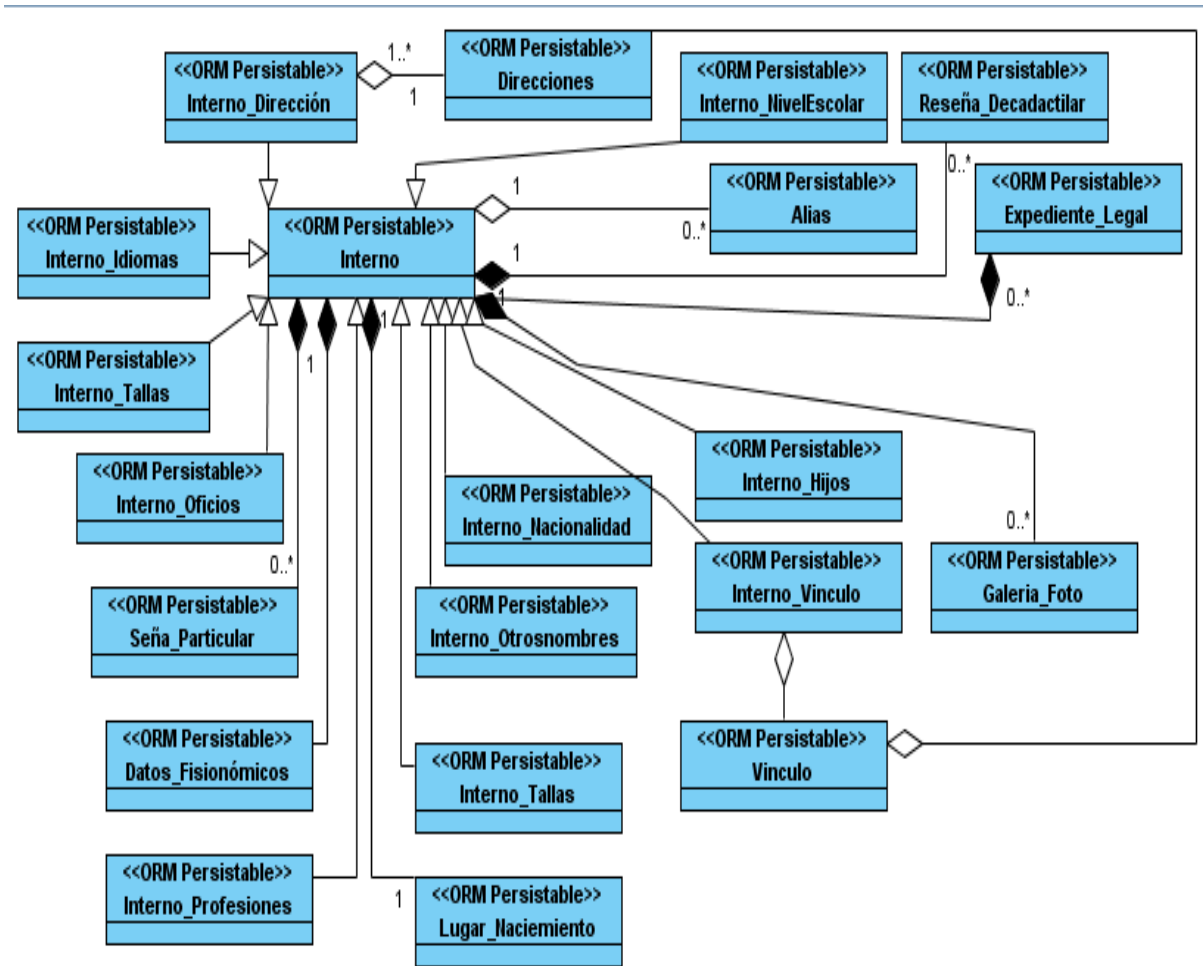


Figura 6. Diagrama de clases persistentes del módulo Datos Generales.

Estereotipos	<<ORM Persistable>>
Interno	Registra los datos personales de cada Interno.
Tabla correspondiente	Interno

Atributos	Visibilidad	Tipo
edad	Private	Integer
personasAbrigo	Private	String
sexo	Private	String
fechaNacimiento	Private	Date
primerNombre	Private	String
segundoNombre	Private	String
primerApellido	Private	String
segundoApellido	Private	String
numeroid	Private	String
paisNacimiento	Private	NomPais
estadoCivil	Private	NomEstadocivil
documentoid	Private	NomDocumentoid

Capítulo III. Diseño e Implementación del Sistema

provinciaNacimiento	Private	NomProvincia
---------------------	---------	--------------

Estereotipos	<<ORM Persistable>>
Direcciones	Registra la dirección particular del interno.
Tabla correspondiente	Direcciones

Atributos	Visibilidad	Tipo
residenciaPermanente	Private	String
numero	Private	String
apto	Private	String
entre	Private	String
localidad	Private	String
provincia	Private	NomProvincia
municipio	Private	NomMunicipio

Estereotipos	<<ORM Persistable>>
Reseña decadactilares	Registra las huellas decadactilares de el interno.

Tabla correspondiente	Resenna_Decadactilar
-----------------------	----------------------

Atributos	Visibilidad	Tipo
vectoriald1	Private	java.sql.Blob
vectoriald2	Private	java.sql.Blob
vectoriald3	Private	java.sql.Blob
vectoriald4	Private	java.sql.Blob
vectoriald5	Private	java.sql.Blob
vectorializ1	Private	java.sql.Blob
vectoriald2	Private	java.sql.Blob
vectorializ3	Private	java.sql.Blob
vectorializ4	Private	java.sql.Blob
vectorializ5	Private	java.sql.Blob
fecha	Private	Date
huellad1	Private	java.sql.Blob
huellad2	Private	java.sql.Blob
huellad3	Private	java.sql.Blob

huellad4	Private	java.sql.Blob
huellad5	Private	java.sql.Blob
pkd1	Private	boolean
pkd2	Private	boolean
pkd3	Private	boolean
pkd4	Private	boolean
pkd5	Private	boolean
pkiz1	Private	boolean
pkiz2	Private	boolean
pkiz3	Private	boolean
pkiz4	Private	boolean
pkiz5	Private	boolean
idinterno	Private	Interno

Estereotipos	<<ORM Persistable>>
Vínculo	Registra los vínculos familiares de cada Interno.
Tabla correspondiente	Vínculo

Capítulo III. Diseño e Implementación del Sistema

Atributos	Visibilidad	Tipo
primerNombre	Private	String
segundoNombre	Private	String
primerApellido	Private	String
segundoApellido	Private	String
interno	Private	boolean
teléfono	Private	NomProvincia
numeroid	Private	NomMunicipio
nacionalidad	Private	NomNacionalidad
documentoid	Private	NomDocumentoid
dirección	Private	Dirección

Esteriotipos	<<ORM Persistable>>
Seña Particular	Registra las señas particulares del interno.
Tabla correspondiente	Seña Particular

Atributos	Visibilidad	Tipo
-----------	-------------	------

descripción	Private	String
foto	Private	java.sql.Blob
ubicación	Private	NomUbicacion
tipoSenna	Private	tipoSenna
interno	Private	Interno

En el Módulo Datos Generales se almacenan todos los datos personales que tienen relación con el Interno por lo tanto todas las entidades están directa o indirectamente relacionadas con el mismo. En este módulo se parte del registro de la mayoría de los datos de un interno o la actualización de los mismos si ya está registrado en la base de datos. La principal entidad que existe es el Interno, porque todas las funcionalidades giran en su entorno.

3.3.5 Diseño del modelo de datos.

De esta actividad se obtiene como resultado el modelo de datos de los módulos Ingreso y Datos Generales. La definición de las entidades persistentes aporta al modelo de datos una buena aproximación de la estructura estática de la base de datos. Esta estructura tiene que garantizar que el almacenamiento y recuperación de la información ocurra de manera adecuada y que se cumplan las restricciones identificadas. En la figura 6 se muestra el diagrama lógico de la base de datos correspondiente al módulo Datos Generales, generado a partir de la definición de las entidades persistentes. El modelo de datos es refinado en la medida en que se avanza en las actividades de diseño porque sea necesario persistir nuevos elementos o se decida la utilización de procedimientos almacenados, vistas, restricciones o funciones definidas en el gestor de base de datos. La utilización de estos objetos de la base de datos es analizada detenidamente y se decide siempre que implique mayor rendimiento en el acceso a los datos y un acoplamiento mínimo al gestor de base de datos.

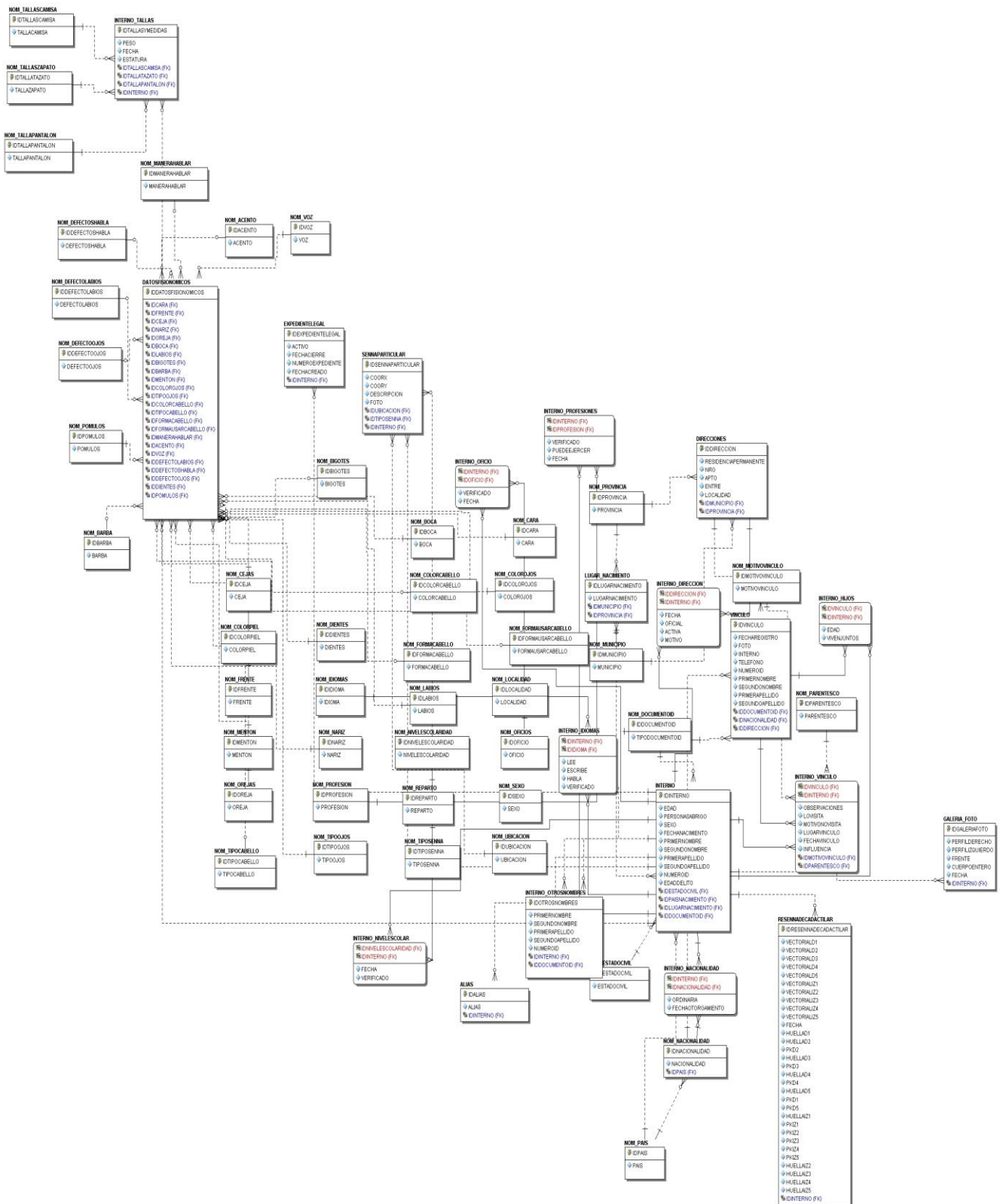


Figura 7. Modelo de datos del módulo Datos Generales.

Capítulo III. Diseño e Implementación del Sistema

A continuación una descripción de las principales entidades persistentes en la base de datos del módulo Datos Generales:

Entidad	Descripción
Interno	Almacena gran parte de los datos personales de cada Interno.

Atributos	Tipo	Puede ser nulo
IDINTERNO	VARCHAR (18)	NO
EDAD	NUMERIC(3.0)	SI
PERSONASABRIGO	NUMERIC(2.0)	SI
SEXO	BIT	NO
FECHANACIMIENTO	DATE	SI
PRIMERNOMBRE	VARCHAR(30)	NO
SEGUNDONOMBRE	VARCHAR(30)	SI
PRIMERAPELLIDO	VARCHAR(30)	NO
SEGUNDOAPELLIDO	VARCHAR(30)	SI
NUMEROID	VARCHAR(30)	NO
EDADDELITO	NUMERIC(3.0)	SI

Capítulo III. Diseño e Implementación del Sistema

Entidad	Descripción
Direcciones	Almacena la dirección de un Interno.

Atributos	Tipo	Puede ser nulo
IDDIRECCION	VARCHAR	NO
RESIDENCIAPERMANENTE	VARCHAR	SI
NRO	INTEGER	SI
APTO	VARCHAR	SI
ENTRE	VARCHAR	SI
LOCALIDAD	VARCHAR	SI
IDMUNICIPIO	INTEGER	NO
IDPROVINCIA	INTEGER	NO

Entidad	Descripción
Vinculo	Almacena los vínculos familiares del Interno

Atributos	Tipo	Puede ser nulo
-----------	------	----------------

Capítulo III. Diseño e Implementación del Sistema

IDVINCULO	VARCHAR (18)	NO
FECHAREGISTRO	DATE	SI
FOTO	PICTURE	SI
INTERNO	BIT	SI
TELEFONO	VARCHAR(18)	SI
NUMEROID	VARCHAR(20)	SI
PRIMERNOMBRE	VARCHAR(20)	NO
SEGUNDONOMBRE	VARCHAR(20)	SI
PRIMERAPELLIDO	VARCHAR(20)	NO
SEGUNDOAPELLIDO	VARCHAR(20)	SI
IDDOCUMENTOID	VARCHAR(20)	NO
IDNACIONALIDAD	VARCHAR(20)	NO
IDDIRECCION	VARCHAR(20)	NO

Entidad	Descripción
Galería de Fotos	Almacena las fotos de los internos.

Capítulo III. Diseño e Implementación del Sistema

Atributos	Tipo	Puede ser nulo
IDGALERIAFOTO	VARCHAR (18)	NO
PERFILDERECHO	PICTURE	NO
PERFILIZQUIERDO	PICTURE	SI
FRENTE	PICTURE	SI
CUERPOENTERO	PICTURE	SI
FECHA	DATE	NO
IDINTERNO	VARCHAR(18)	NO

Entidad	Descripción
Seña Particular	Registra las señas particulares de cada Interno

Atributos	Tipo	Puede ser nulo
IDSENNAPARTICULAR	VARCHAR (18)	NO
COORX	NUMERIC(10.0)	SI
COORY	NUMERIC(10.0)	SI
DESCRIPCION	VARCHAR(30)	SI

Capítulo III. Diseño e Implementación del Sistema

IDUBICACION	VARCHAR(18)	NO
IDTIPOSENNA	VARCHAR(18)	NO
IDINTERNO	VARCHAR(18)	NO

Entidad	Descripción
Datos Fisionómicos	Almacena los datos fisionómicos del Interno

Atributos	Tipo	Puede ser nulo
IDCARA	VARCHAR (18)	NO
IDFRENTE	NUMERIC(10.0)	NO
IDCEJA	NUMERIC(10.0)	NO
IDNARIZ	VARCHAR(30)	NO
IDOREJA	VARCHAR(18)	NO
IDBOCA	VARCHAR(18)	NO
IDLABIOS	VARCHAR(18)	NO
IDBIGOTES	VARCHAR(18)	NO
IDBARBA	VARCHAR(18)	SI

Capítulo III. Diseño e Implementación del Sistema

IDMENTON	VARCHAR(18)	SI
IDCOLOROJOS	VARCHAR(18)	NO
IDTIPOOJOS	VARCHAR(18)	NO
IDCOLORCABELLO	VARCHAR(18)	NO
IDTIPOCABELLO	VARCHAR(18)	SI
IDFORMACABELLO	VARCHAR(18)	SI
IDFORMAUSARCABELLO	VARCHAR(18)	SI
IDMANERAHABLAR	VARCHAR(18)	SI
IDACENTO	VARCHAR(18)	SI
IDVOZ	VARCHAR(18)	SI
IDDEFECTOSLABIOS	VARCHAR(18)	NO
IDDEFECTOSHABLA	VARCHAR(18)	SI
IDDEFECTOOJOS	VARCHAR(18)	SI
IDDIENTES	VARCHAR(18)	NO
IDPOMULOS	VARCHAR(18)	NO
IDCOLORPIEL	VARCHAR(18)	NO
IDINTERNO	VARCHAR(18)	NO

Capítulo III. Diseño e Implementación del Sistema

Entidad	Descripción
Reseña decadaactilares	Recoge las impresiones de las huellas decadaactilares del interno

Atributos	Tipo	Puede ser nulo
IDRESENADECACTILAR	VARCHAR (18)	NO
VECTORIALD1	PICTURE	SI
VECTORIALD2	PICTURE	SI
VECTORIALD3	PICTURE	SI
VECTORIALD4	PICTURE	SI
VECTORIALD5	PICTURE	SI
VECTORIALIZ1	PICTURE	SI
VECTORIALIZ2	PICTURE	SI
VECTORIALIZ3	PICTURE	SI
VECTORIALIZ4	PICTURE	SI
VECTORIALIZ5	PICTURE	SI
FECHA	DATE	NO
HUELLAD1	PICTURE	SI

Capítulo III. Diseño e Implementación del Sistema

HUELLAD2	PICTURE	SI
PKD2	BINARY(20)	SI
HUELLAD3	PICTURE	SI
PKD3	BINARY(20)	SI
HUELLAD4	PICTURE	SI
PKD4	BINARY(20)	SI
HUELLAD4	PICTURE	SI
PKD4	BINARY(20)	SI
HUELLAD4	PICTURE	SI
PKD5	BINARY(20)	SI
PKD1	BINARY(20)	SI
HUELLAIZ1	PICTURE	SI
HUELLAIZ2	PICTURE	SI
PKIZ2	BINARY(20)	SI
HUELLAIZ3	PICTURE	SI
PKIZ3	BINARY(20)	SI
HUELLAIZ4	PICTURE	SI

PKIZ4	BINARY(20)	SI
HUELLAIZ4	PICTURE	SI
PKIZ4	BINARY(20)	SI
HUELLAIZ4	PICTURE	SI
PKIZ5	BINARY(20)	SI
PKIZ1	BINARY(20)	SI

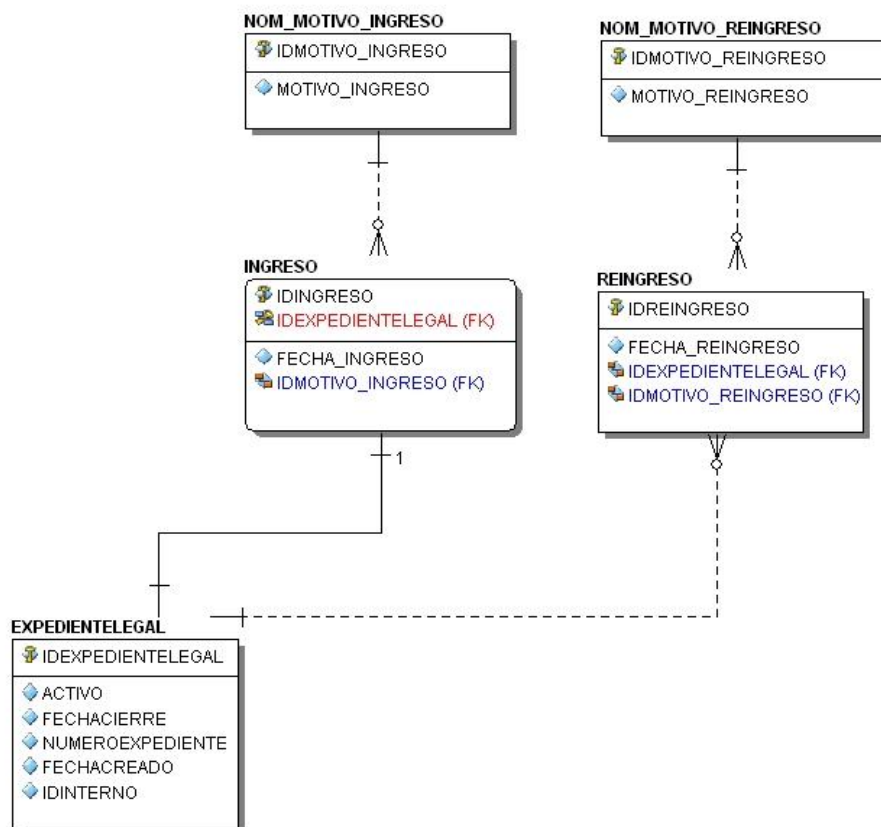


Figura 8. Modelo de datos del módulo Ingreso.

Entidad	Descripción
Ingreso	Registra la fecha del ingreso y el motivo del mismo

Atributos	Tipo	Visibilidad
IDINGRESO	VARCHAR	PRIVATE
IDEXPEDIENTELELEGAL	VARCHAR	PRIVATE
FECHA_INGRESO	DATE	PRIVATE
IDMOTIVO_INGRESO	NUMERIC	PRIVATE

3.3.6 Diseño de la Arquitectura en Capas.

Diseño de la capa web: En esta actividad se definen las vistas y el flujo de navegación, y a partir de esto, las posibles peticiones del usuario y los controladores que las van a atender. Al definir los controladores se extienden estas clases agregando las funcionalidades específicas del formulario o asistente particular, con lo que se gana en productividad.

Diseño de la capa de servicios: El diseño de la capa de servicios contiene las clases necesarias para cubrir las funcionalidades definidas para el módulo y expone sus funcionalidades a la capa web. Los servicios se definen de acuerdo al agrupamiento lógico de funcionalidades que respondan a procesos de negocio. Se definen también los eventos que pueden ser lanzados o escuchados, las excepciones a lanzar y la manera en que van a ser tratadas y mostradas al usuario.

Diseño de la capa de datos: El diseño de la capa de datos se encuentra muy ligado al de la capa de servicios, dado que las funcionalidades de esta capa surgen como necesidades de la capa de servicios. Su diseño se hace de manera que exista el mínimo acople posible al gestor de base de datos, tratando de aprovechar al máximo las potenciales que ofrece el gestor Oracle. El uso del

framework ORM Hibernate posibilita el bajo acoplamiento de la aplicación y la base de datos. Las interfaces de la capa de datos definen las funcionalidades necesarias relacionadas con la persistencia y recuperación de datos del medio de almacenamiento.

3.3.7 Implementación de las clases de dominio

Las entidades del dominio suelen tener muy poco comportamiento. En este paso se debe refinar la definición de todos los atributos de las entidades de manera que queden listas en un buen por ciento para implementaciones reusables. Grails brinda la posibilidad de validar los atributos en el lado del cliente, el siguiente ejemplo muestra dicha validación:

```
class Interno {
    String idinterno
    java.math.BigDecimal edad
    java.math.BigDecimal personasabrigo
    Date fechanacimiento
    String primernombre
    String segundonombre
    String primerapellido
    String segundoapellido
    String numeroid
    Date edadelito

    static constraints = {
        idinterno(size: 1..18, blank: false)
        edad(nullable: true)
        personasabrigo(nullable: true)
        fechanacimiento(nullable: true)
        primernombre(size: 1..30, blank: false)
        segundonombre(size: 0..30)
        primerapellido(size: 1..30, blank: false)
        segundoapellido(size: 0..30)
        numeroid(size: 1..11, blank: false)
        edadelito(nullable: true)
    }

    String toString() {
        return "${idinterno}"
    }
}
```

Figura 8. Implementación de la clase de dominio Interno


```
package DatosGenerales
import Comun.nomencladores.*
class Direcciones {

    String residenciaPermanente
    String numero
    String apto
    String entre
    String localidad
    Comun.nomencladores.NomProvincia provincia
    Comun.nomencladores.NomMunicipio municipio

    static constraints = {

        residenciaPermanente(nullable: true, size: 0..100)
        numero(nullable: true, size: 0..10)
        apto(nullable: true, size: 0..10)
        entre(nullable: true, size: 0..100)
        localidad(nullable: true, size: 0..100)
        provincia(nullable: true)
        municipio(nullable: true)

    }
}
```

Figura 9. Implementación de la clase de dominio Direcciones

```
class Vinculo

    Date fechaRegistro
    java.sql.Blob foto
    boolean interno
    String telefono
    String numeroid
    String primerNombre
    String segundoNombre
    String primerApellido
    String segundoApellido
    Comun.nomencladores.NomNacionalidad nacionalidad
    nomencladores.NomDocumentoid documentoid
    Direccion direccion

    static constraints = {

        fechaRegistro(nullable: true)
        foto(nullable: true)
        interno(nullable: true)
        telefono(nullable: true)
        numeroid(size: 0..18, nullable: true)
        primerNombre(size: 1..20, blank: false)
        segundoNombre(size: 0..20, nullable: true)
        primerApellido(size: 1..20, blank: false)
        segundoApellido(size: 0..20, nullable: true)
        nacionalidad(nullable: true)
        documentoid(nullable: true)
        direccion(nullable: true)
    }
}
```

Figura 10. Implementación de la clase de dominio Vínculo

```
package DatosGenerales

class SennaParticular {

    String sennaParticular
    java.math.BigDecimal coorx
    java.math.BigDecimal coory
    String descripcion
    java.sql.Blob foto
    nomencladores.NomUbicacion ubicacion
    nomencladores.NomTiposenna tipoSenna
    Interno interno

    static constraints = {
        sennaParticular(blank: false)
        coorx(nullable: true)
        coory(nullable: true)
        descripcion(size: 0..100)
        foto(nullable: true)
        ubicacion()
        tipoSenna()
        interno()
    }
}
```

Figura 11. Implementación de la clase de dominio Señas Particulares

Las clases de dominio mantendrán las relaciones existentes entre ellas siguiendo la estructura del modelo de relaciones que propone Grails, estas relaciones pueden ser de uno-uno, uno-muchos, muchos-uno, muchos-muchos. Para la relación que se establece entre las clases de dominio con las vistas y controladores Grails presenta una técnica llamada Scaffolding que una vez creada las clases de dominio le solicita a Grails que se creen las clases controladoras y las vistas necesarias para realizar las operaciones.

3.3.8 Implementación de las interfaces

Se implementan los métodos definidos para cada una de las interfaces, 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. (ver anexos 2)

```

<html>

<form name="primerSave" action="save" method="post" dojoType="dijit.form.Form" id="formularioSave">

  <table style="position:absolute;top:15px;">
    <tr>
      <td>Documento de identidad:</td>
      <td width="28px">Número</td>
      <td>Expediente legal</td>
      <td>Sexo</td>
      <td width="28px"></td>
    </tr>
    <tr>
      <td>
        <div dojoType="dojo.data.ItemFileReadStore" jsId="docIdent" id="idlolo"
          url="<g:createlink controller='nomDocumentoid' action='selectDoc' />">
        </div>

        <select dojoType="dijit.form.FilteringSelect" store="docIdent"
          searchAttr="nombre"
          id="documento12"
          autocomplete="true" name="documentoid" from="{nomencladores.NomDocumentoid.list()}" optionKey="id" value=
            "${internoInstance?.documentoid?.id}"

        </select>
      </td>
      <td width="28px"><input dojoType="dijit.form.TextBox" class="input" name="numeroid" maxLength="11" value="${internoInstance?.numeroid}"></td>
      <td width="28px"><input dojoType="dijit.form.TextBox" class="input" name="expedienteLegal" value=""></td>
      <td><select dojoType="dijit.form.FilteringSelect" name="sexo" id="sexo" hasDownArrow="true" style="width: 150px">
        <option value="Seleccione...">Seleccione...</option>
        <option value="masculino">Masculino</option>
        <option value="femenino">Femenino</option>
      </select></td>
    </tr>
    <tr height="10px"></tr>
    <tr>
      <td>Primer nombre:</td>
      <td>Segundo nombre:</td>
      <td>Primer apellido:</td>
      <td>Segundo apellido:</td>
    </tr>
    <tr>
      <td><input dojoType="dijit.form.TextBox" class="input" name="primerNombre" maxLength="30" value="${internoInstance?.primerNombre}"></td>
      <td><input dojoType="dijit.form.TextBox" class="input" name="segundoNombre" maxLength="30" value="${internoInstance?.segundoNombre}" ></td>
    </tr>
  </table>

```

Figura 12. Fragmento del código de la vista de Datos Personales.

3.3.9 Implementación de los controladores.

Las clases controladoras se encargan de coordinar y gestionar la interacción entre las vistas y las clases de dominio. Grails se encarga de convertir esta relación en algo sencillo pues el controlador

sólo consta de una línea de código que se encarga manejar las relaciones entre las clases, entre las que se pueden encontrar, redireccionar al usuario a una página diferente, delegar una acción a un actor diferente (puede ser otro controlador o una clase de servicio), y preparar y enviar la respuesta de nuevo a la vista. Además Grails define el directorio `grails-app/controllers` para ubicar las clases controladoras.

3.3.10 Implementación de la capa de datos.

A la hora de realizar las operaciones en la capa de datos Grails utiliza GORM (objeto-relacional), pero este a su vez utiliza las facilidades que brinda Hibernate para la persistencia de los datos en la aplicación. Además GORM tiene incluido algunos métodos dinámicos entre los que se encuentran “`save()`”, “`delete()`”, “`refresh()`” e “`ident()`” así como métodos estáticos como “`count()`”, “`exist()`”, “`find()`”, “`findAll()`”, “`findBy*()`” y “`findWhere()`”, esto proporciona que el acceso a la base de datos se torne más sencillo. Grails brinda la posibilidad de trabajar con una base de datos interna (HSQLDB), la cual no será utilizada en el desarrollo de los módulos Datos Generales e Ingreso porque se utilizará una base de datos externa donde el archivo configurable de la misma para Grails está localizado en:

```
dataSource {
    pooled = true
    driverClassName = "oracle.jdbc.driver.OracleDriver"
    dialect = org.hibernate.dialect.OracleDialect
    username = "proyecto"
    password = "proyecto"
}
hibernate {

    cache.use_second_level_cache=true
    cache.use_query_cache=true
    cache.provider_class='net.sf.ehcache.hibernate.EhCacheProvider'
    //config.location = "file:/hibernate.cfg.xml"
}
// environment specific settings
environments {
    development {
        dataSource {
            dbCreate = "update" // one of 'create', 'create-drop', 'update'
            url = "jdbc:oracle:thin:@10.36.13.154:1521:SIGIDB"
            //jdbc:oracle:thin:@<host>:<port>:<sid>
        }
    }
    test {
        dataSource {
            dbCreate = "update"
            url = "jdbc:oracle:thin:@10.36.13.154:1521:SIGIDB"
        }
    }
}
```

Figura 13. Implementación de la capa de acceso a datos.

3.3.11 Implementación de la lógica en el cliente.

Para esta actividad se implementan en los archivos JS contenidos en **Web Application/Js**. Las validaciones existentes del lado del cliente, así como el comportamiento de cada componente Dojo para los diálogos de información o error. El lanzamiento de peticiones desde el cliente al servidor también está incluido en esta actividad mediante el uso de tecnología AJAX, JSON y para el uso de componentes de bibliotecas Javascript Dojo Toolkit. Algunos ([ver anexos](#))

A continuación se muestran algunas de las interfaces de usuario de los módulos Ingreso y Datos Generales:

The image shows a web application interface for the 'Ingreso' module. At the top, there is a horizontal menu with tabs: 'Ingreso', 'Datos Generales', 'Decomisos y Ocupaciones', 'Situación Legal', 'Entrevistas', and 'Desig. y Ubica'. The 'Ingreso' tab is active. Below the menu, the main content area is titled 'Criterios de Búsqueda Simple:'. It contains a grid of search criteria: 'Documento de Identidad' (with a dropdown arrow), 'Número', 'Expediente Legal', 'Sexo' (with a dropdown arrow), 'Primer Nombre', 'Segundo Nombre', 'Primer Apellido', and 'Segundo Apellido'. Below this grid is a link '<Búsqueda Avanzada>'. At the bottom right of the form area, there are two buttons: 'Buscar' and 'Cancelar'.

Figura 14. Interfaz de Usuario de la búsqueda simple que se realiza en Ingreso

Ingreso | Datos Generales | Decomisos y Ocupaciones | Situación Legal | Entrevistas | Desig. y Ubicar

Criterios de Búsqueda Simple:

Documento de Identidad: <input type="text"/>	Número: <input type="text"/>	Expediente Legal: <input type="text"/>	Sexo: <input type="text"/>
Primer Nombre: <input type="text"/>	Segundo Nombre: <input type="text"/>	Primer Apellido: <input type="text"/>	Segundo Apellido: <input type="text"/>
Alias: <input type="text"/>	EFP: <input type="text"/>	Año de EFP: <input type="text"/>	Causa: <input type="text"/>
Año de la causa: <input type="text"/>	Seña Particular: <input type="text"/>	Ubicación de la seña particular: <input type="text"/>	

<Búsqueda Simple>

Buscar Cancelar

Figura 15. Interfaz de Usuario de la búsqueda avanzada que se realiza en Ingreso

Ingreso | Datos Generales | Decomisos y Ocupaciones | Situación Legal | Entrevistas | Desig. y Ubicar

Personales | Familiares | Dirección | Señas Part. | Rasgos Fisicos | Reseña Fotograf. | Reseña Deca.

Documento de identidad: Número Expediente legal Sexo

Primer nombre: Segundo nombre: Primer apellido: Segundo apellido:

Edad Personas Abrigo

Otros nombres

Religión: Estado civil Fecha de nacimiento Pais de nacimiento:

Provincia de nacimiento Municipio Nacionalidad Organizadores de masas CDR CTC FMC

Organizaciones politicas UJC PCC

Alias Adicionar

Alias Eliminar

1 de 2 **Siguiete**

Cancelar Siguiete

Figura 16. Interfaz de Usuario Registrar Datos Personales

The screenshot shows a software window with a menu bar at the top containing 'Ingreso', 'Datos Generales', 'Decomisos y Ocupaciones', 'Situación Legal', 'Entrevistas', and 'Desig. y Ubica'. Below the menu bar is a sub-menu with 'Personales', 'Familiars', 'Dirección', 'Señas Part.', 'Rasgos Fisicos', 'Reseña Fotog.', and 'Reseña Deca.'. The 'Familiars' tab is active. The main content area is titled 'Datos de la madre:' and contains four input fields for 'Primer nombre:', 'Segundo nombre:', 'Primer apellido:', and 'Segundo apellido:'. Below this is the 'Datos del padre:' section with similar four input fields. The 'Datos del esposo(a) o concubino(a):' section has four input fields for name and surname, plus a '¿Conviven juntos?' checkbox. The 'Datos de los hijos(as):' section has four input fields for name and surname, a 'CIRP:' field, an 'Edad:' field, and another '¿Conviven juntos?' checkbox. There is also an 'Otros hijos(as)' label. At the bottom right of the form are 'Aceptar' and 'Cancelar' buttons.

Figura 17. Interfaz de Usuario Registrar Datos Familiares

The screenshot shows the same software window as Figure 17, but with the 'Dirección' tab selected in the sub-menu. The main content area is titled 'Dirección registrada en el CI:' and contains several input fields: 'Calle:', 'Edificio:', 'No:', 'Apto:', 'Entre:', 'Y:', 'Reparto:', 'Localidad:', 'Carretera:', 'Km:', 'Finca:', 'Cooperativa:', 'Municipio:', 'Provincia:', 'Pais:', and 'Teléfono:'. The 'Municipio:', 'Provincia:', and 'Pais:' fields are dropdown menus. There is also a 'Residencia:' dropdown menu. At the bottom right of the form are 'Aceptar' and 'Cancelar' buttons.

Figura 18. Interfaz de Usuario Registrar Direcciones

Conclusiones

Como resultado del presente trabajo se realizó el diseño e implementación los Módulos Ingreso y Datos Generales a partir de los requerimientos de software establecidos con el cliente y haciendo uso de la arquitectura y las tecnologías definidas por el proyecto. Se realizó un estudio de las tecnologías y herramientas a utilizar en el diseño e implementación del módulo así como del análisis de los requisitos de software y del modelo de negocio correspondientes. Las actividades de diseño e implementación fueron ejemplificadas a través de diagramas de clase y fragmentos de código fuente cumpliendo así con los objetivos propuestos para este trabajo.

Recomendaciones.

Comprobar el rendimiento de los módulos Ingreso y Datos Generales frente a bases de datos de grandes volúmenes de información para validar que el diseño propuesto satisface tales condiciones, puesto que actualmente el sistema está probado para una cantidad limitada de datos que no ponen al límite la solución propuesta. Revisar las funcionalidades de los módulos con el objetivo de agregar nuevas funcionalidades.

Referencias Bibliográficas

1. **Abdul-Jawad, Bashar. 2009.** *Groovy and Grails Recipes*. United States of America : Sun Microsystems Inc, 2009. ISBN-13 (pbk): 978-1-4302-1600-1.
2. **Albiol, Francesc Rosés. 2003.** *Introducción a Hibernate*. 2003.
3. **Arias, Arturo. 2006.** *Proyecto Técnico*. . 2006.
4. **Benavides Zaila, Yadira y Gómez Correa, Juan Carlos. 2008.** *Diseño e implementación de los módulos Decisiones y Egresos del Sistema de Gestión Penitenciaria de la República Bolivariana de Venezuela*. Ciudad de la Habana : s.n., 2008.
5. **Glen Smith, Peter Ledbrook. 3.** *Grails in Action*. United States of America : Manning Publications Co., 3. ISBN 978-1-933988-93-1.
6. **Gómez., Joaquina Naranjo. 2006.** *El Sistema Penitenciario Cubano*. . 2006.
7. **Jacobson, Ivar, Booch, Grady y Rumbauch, James. 2000. 2000..** *El Proceso Unificado de Desarrollo del Software*. . Madrid : s.n., 2000.
8. **Larman., Craig.** *UML Y Patrones. Introducción al análisis y diseño orientado a objeto*. . México : s.n.
9. **León, Fernando y Cano García, Welicki. 2007.** *Metodologías de la Ingeniería en Software*. s.l. : Electrocast Team, 2007.
10. **ORACLE, CORPORATION. 2005.** *Oracle Database Documentation Library*. 2005.
11. **Peiso Hernández, TC. Filiberto, Curbelo Mederos, Tte. Coronel Raúl y Fernández Borrego, 1-Tte. Jaen. 2009.** *INFORME DE FACTIBILIDAD E IDEAS PRELIMINARES SOBRE LA INFORMAIZACION DEL SISTEMA PENITENCIARIO CUBANO*. Ciudad de la Habana : s.n., 2009.
12. **Pimentel., Luis Alberto. 2008.** *Herramientas de desarrollo*. . 2008.
13. **Pressman., Roger. 2002..** *Ingeniería de Software. Un enfoque práctico*. . 2002.
14. **Walls, Craig y Breidenbach, Ryan. 2005.** *Spring in Action*. Estados Unidos de América : Manning Publications Co., 2005.

15. Zequeira Peña, Dr. Tte. Cnel. Alfonso, Céspedes Quesada, Lic.1er Tte. Aimeé. *Vocabulario Jurídico Penitenciario*. Ciudad de la Habana : MININT, 2006.

Bibliografía

1. **Williamson, Alan, y otros.** *Programacion Java Server con J2EE Edicion.*
2. **Walls, Craig y Breidenbach, Ryan.** *Spring in Action.* Estados Unidos de América : Manning Publications Co., 2005.
3. **Pressman., Roger.** *Ingeniería de Software. Un enfoque práctico. .* 2002.
4. **Pimentel., Luis Alberto.** *Herramientas de desarrollo. .* 2008.
5. **Peiso Hernández, TC. Filiberto, Curbelo Mederos, Tte. Coronel Raúl y Fernández Borrego, 1-Tte. Jaen.** *INFORME DE FACTIBILIDAD E IDEAS PRELIMINARES SOBRE LA INFORMAIZACION DEL SISTEMA PENITENCIARIO CUBANO.* Ciudad de la Habana : s.n., 2009.
6. **ORACLE, CORPORATION.** *Oracle Database Documentation Library.* 2005.
7. **León, Fernando y Cano García, Welicki.** *Metodologías de la Ingeniería en Software.* s.l. : Electrocast Team, 2007.
8. **Larman., Craig.** *UML Y Patrones. Introducción al análisis y diseño orientado a objeto. .* Mexico : s.n.
9. **Larman, Craig.** *UML y Patrones. Mexico .* 1999. 970-17-0261-1..
10. **LADD., SETH.** *Expert Spring MVC and Web Flow. .* 2006. .
11. **King, Gavin y Bauer, Christian.** *Hibernate in Action.* 2004.
12. **Jacobson, Ivar, Booch, Grady y Rumbauch, James. 2000.** *El Proceso Unificado de Desarrollo del Software. .* Madrid : s.n., 2000.
13. **Interior, Ministerio del.** *Informe de Factibilidad del proyectode desarrollo estratégico del Sistema Penitenciario Nacional.* Ciudad de la Habana : s.n., 2008-2009.
14. **Gómez., Joaquina Naranjo.** *El Sistema Penitenciario Cubano. .* 2006.
15. **Glen Smith, Peter Ledbrook.** *Grails in Action.* United States of America : Manning Publications Co., 3. ISBN 978-1-933988-93-1.
16. **Gaceta Oficial de la República de Cuba.** *Resolución No-76/2000.* Ciudad de La Habana : s.n., 2000.
17. **Estado, Consejo de.** *Constitución de la Republica de Cuba.* Ciudad Habana : s.n., 1976.
18. **Bodoff, Stephanie.** *The J2EE Tutorial.* s.l. : Addison-Wesley., 2002.
19. **Benavides Zaila, Yadira y Gómez Correa, Juan Carlos.** *Diseño e implementación de los módulos Decisiones y Egresos del Sistema de Gestión Penitenciaria de la República Bolivariana de Venezuela.* Ciudad de la Habana : s.n., 2008.

20. **Arias, Arturo.** *Proyecto Técnico*. 2006. 2.
21. **Albiol, Francesc Rosés.** *Introducción a Hibernate*. 2003.
22. **Abdul-Jawad, Bashar.** *Groovy and Grails Recipes*. United States of America : Sun Microsystems Inc, 2009. ISBN-13 (pbk): 978-1-4302-1600-1.

Glosario de Términos

SIGI: Sistema de Gestión del Interno(a).

SACORE: Sistema Automatizado para el Control del Recluso.

Interno: En Derecho Penitenciario y con las diferencias de cada país, se utiliza para referirse a las personas que cumplen sanción y medida de seguridad reeducativa, en un centro penitenciario o especializado de trabajo.

Acusado: Persona a quien se le haya decretado la medida cautelar de prisión provisional.

Sancionado: Persona ejecutoriamente sancionada a privación de libertad o a trabajo correccional con internamiento.

Asegurado: Persona a quien se le hubiere impuesto una medida de seguridad reeducativa de internamiento.

Clasificación: Es la división y agrupación de la población carcelaria en grupos de más o menos homogéneos bajo diferentes criterios jurídicos-criminológicos.

Expediente de Fase Preparatoria: Sumario en que obran todas las acciones policiales, tanto favorables como adversas al acusado.

Generales: En Derecho Penitenciario, son los datos que identifican al recluso y su Situación Legal (nombres y apellidos, padres, fecha y lugar de nacimiento, dirección del domicilio, número del carné de identidad, ocupación, oficio, causa, tribunal, delito, sanción, fecha de ingreso y de extinción).

Delito: Acción u omisión socialmente peligrosa prohibida por la Ley bajo conminación de una sanción.

Responsabilidad civil: Es la obligación del sancionado de resarcir y reparar el daño civil proveniente de la comisión del delito.

Régimen Penitenciario: Conjunto de normas que establecen el Reglamento y la Ley que deben observar los reclusos durante el cumplimiento de la sanción o medida.

Framework: Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

API: Interfaz de Programación de Aplicaciones.

Plugins: Un plugin o plug-in, es una aplicación que interactúa con otra para agregarle una funcionalidad específica y es ejecutada por la aplicación principal. En el caso particular de Eclipse no son más que un conjunto de clases que permiten hacerlo más extensible.

IDE: Integrated Development Environment (Ambiente Integrado de Desarrollo).

JVM: Java Virtual Machine (Maquina Virtual de Java).

CU: Caso de Uso.

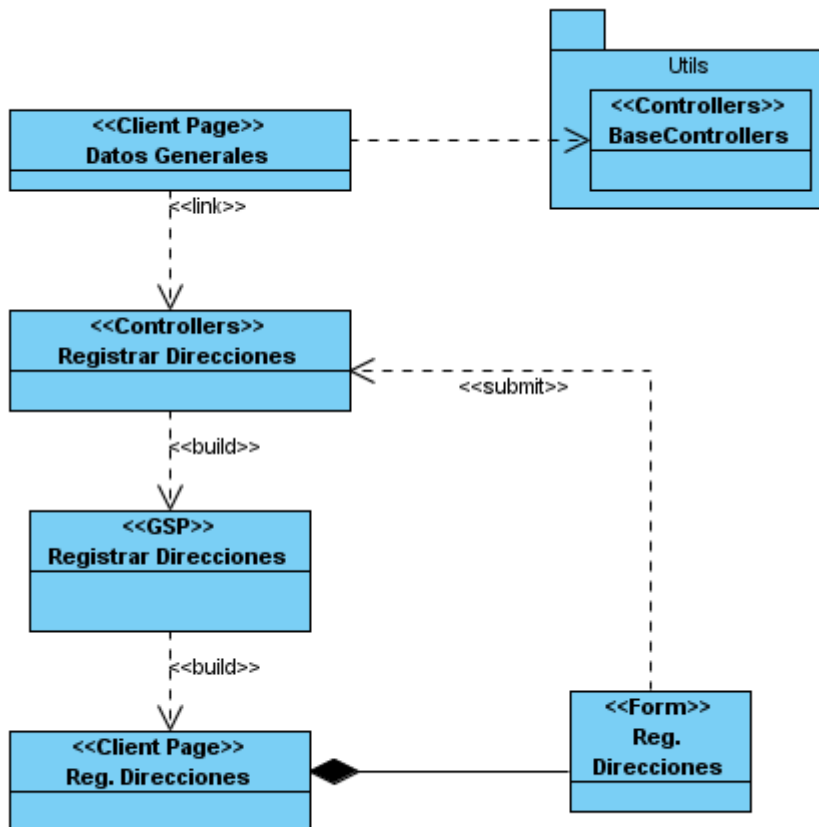
Módulo: Encapsula un conjunto de funciones que debe realizar el sistema, las cuales son agrupadas por tener características muy similares y se definen en la etapa de diseño.

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

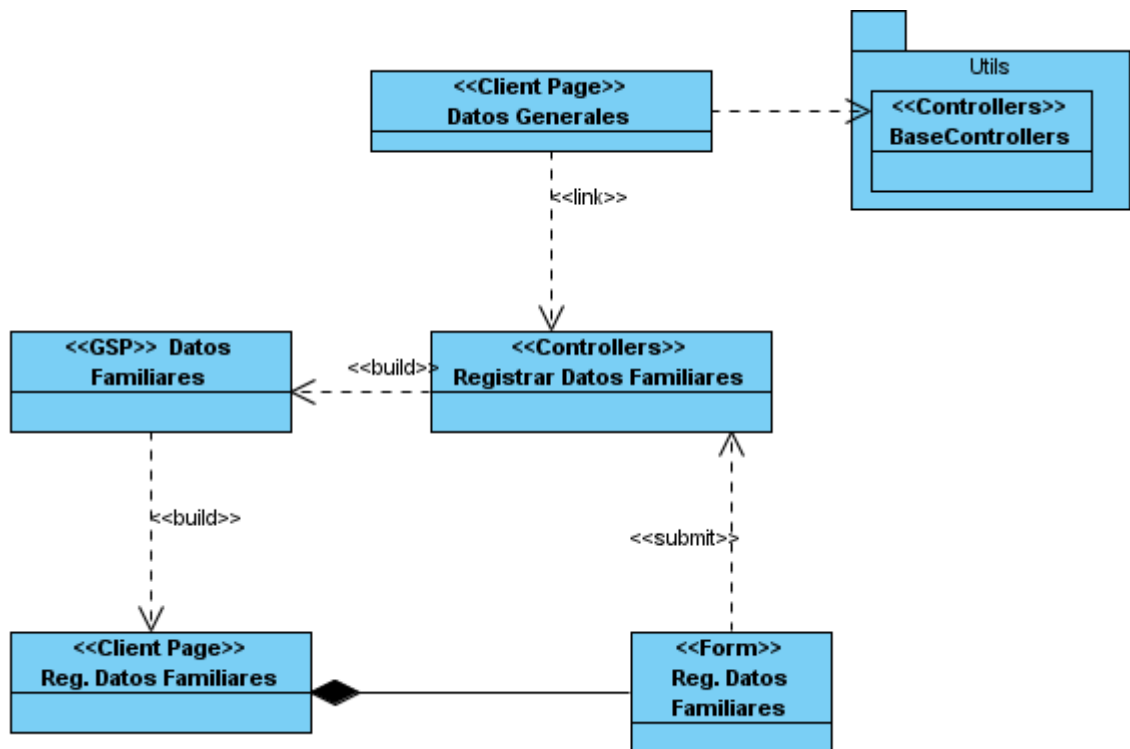
Código o código fuente: Conjunto de instrucciones que componen un programa, escrito en cualquier lenguaje.

Anexos 1

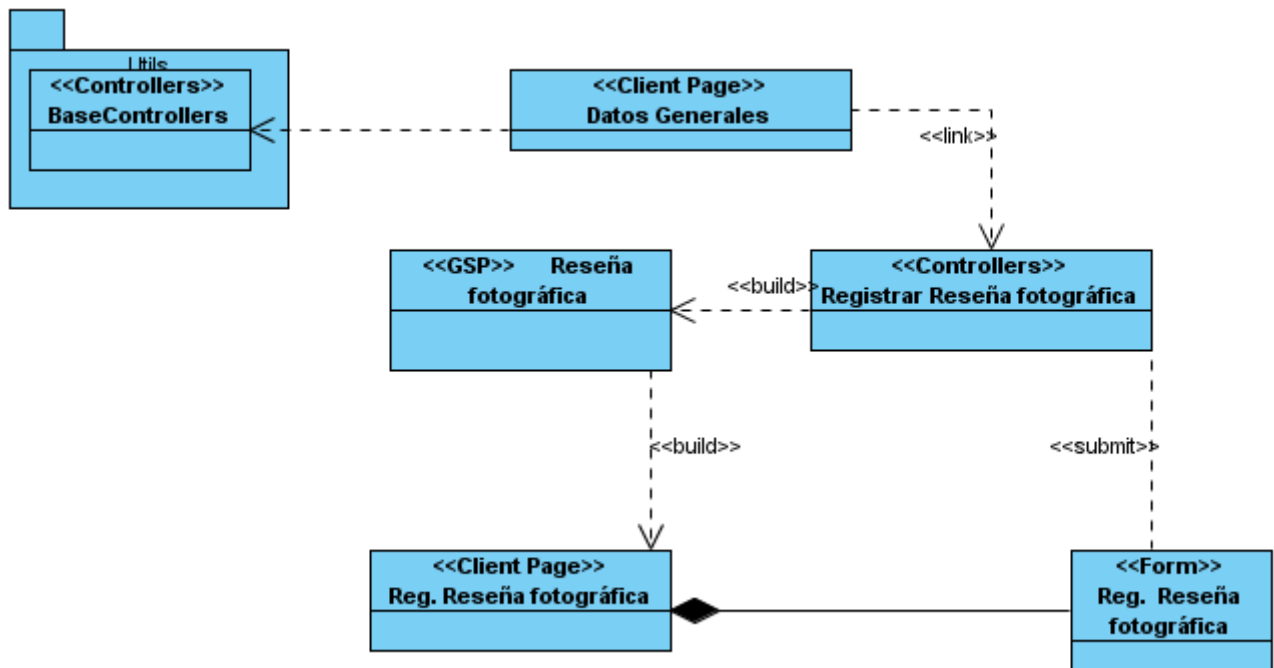
Diagramas de clases de diseño



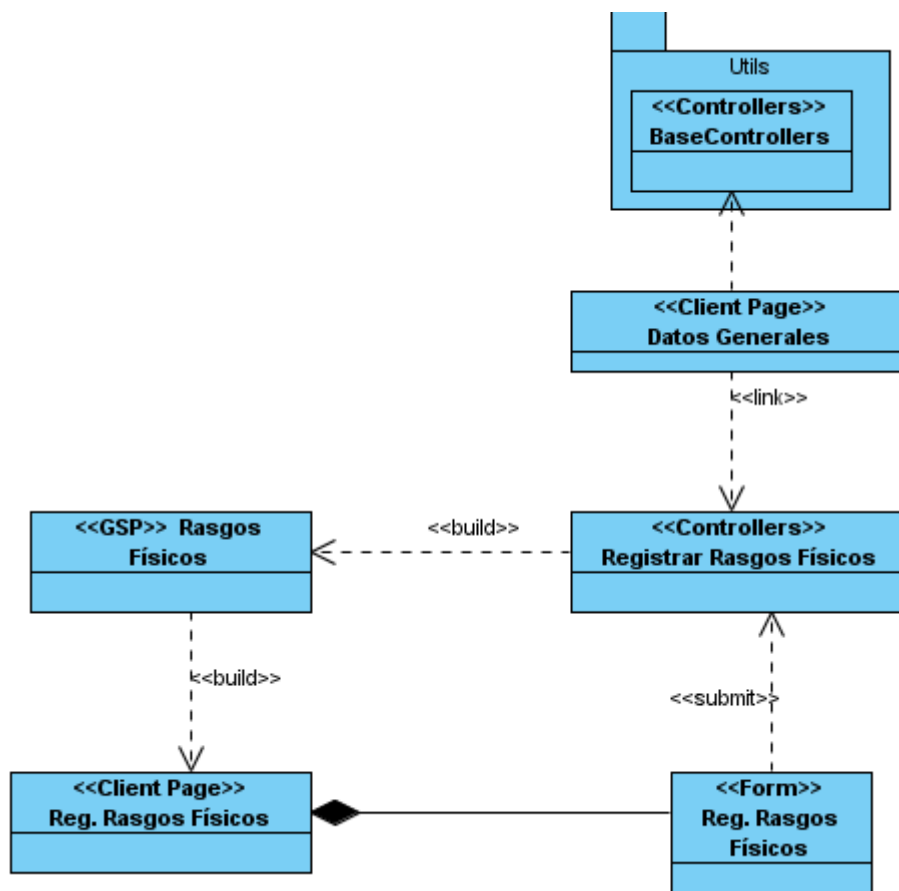
Anexo 1.1. Diagrama de clases del diseño de el caso de uso Registrar Direcciones.



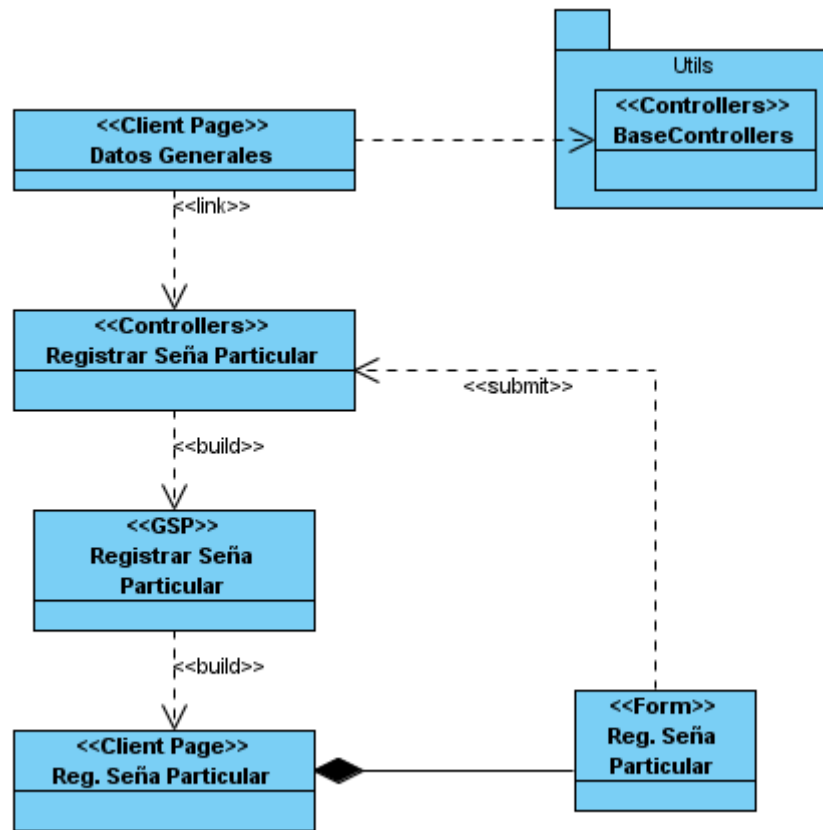
Anexo1.2. Diagrama de clases del diseño de el caso de uso Registrar Datos Familiares.



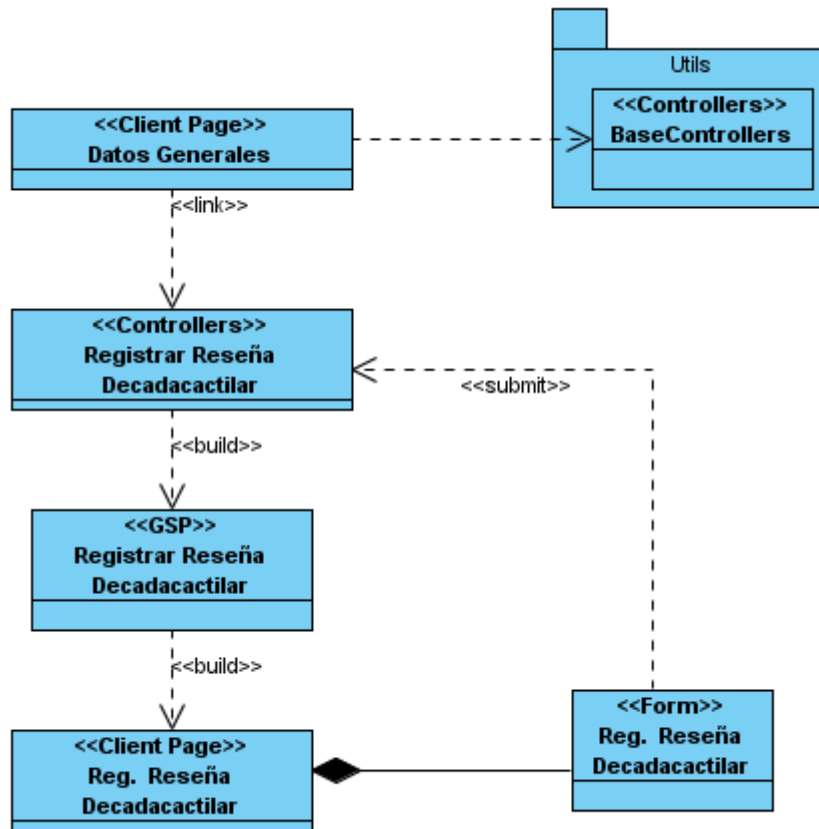
Anexo 1.3. Diagrama de clases del diseño de el caso de uso Registrar Reseña Fotográfica



Anexo 1.4. Diagrama de clases del diseño de el caso de uso Registrar Rasgos Físicos



Anexo 1.5. Diagrama de clases del diseño de el caso de uso Registrar Señal Particular



Anexo 1.6. Diagrama de clases del diseño de el caso de uso Registrar Reseña Decadactilar


Anexo 2. Validaciones

The image shows a web application interface with a tabbed menu at the top. The active tab is 'Datos Generales'. Below the menu, there are several input fields and dropdown menus. A modal error dialog is displayed in the center, with a red 'X' icon and the text: 'ERROR', 'Ha ocurrido un error', and 'Por favor introduzca el numero de identidad del interno'. The error dialog is positioned over the 'Número' field. The form includes fields for 'Documento de identidad', 'Número', 'Expediente legal', 'Sexo', 'Primer nombre', 'Segundo nombre', 'Primer apellido', 'Segundo apellido', 'Edad', 'Otros nombres', 'Religión', 'Provincia de nacimiento', 'País de nacimiento', 'Organizadores de masas' (with checkboxes for CDR, CTC, FMC), 'Organizaciones políticas' (with checkboxes for UJC, PCC), and 'Alias' (with 'Adicionar' and 'Eliminar' buttons). At the bottom right, there are buttons for '1 de 2 Siguiente' and 'Cancelar'.

Anexo 2.1 Error que ocurre al no seleccionar el número de identidad el interno.

Documentos de identidad: Número Expediente legal Sexo

Primer nombre: Segundo nombre: Primer apellido: Segundo apellido:

Edad **Personas Abrigo**  El valor especificado no es válido.

Otros nombres

Religión: Estado civil Fecha de nacimiento País de nacimiento:

Provincia de nacimiento Municipio Nacionalidad Organizadores de masas CDR CTC FMC

Organizaciones políticas UJC PCC

Alias

Alias

1 de 2 **Siguiente**

Anexo 2.2 Advertencia que ocurre al pasarle valores que no están soportados por ese campo.

The image shows a software application window with a tabbed interface. The active tab is 'Personales'. The form contains the following fields and values:

- Documento de identidad: Carnet de Indentidad
- Número: 85122321723
- Expediente legal: 2312
- Sexo: Masculino
- Primer nombre: Jose
- Segundo nombre: Manuel
- Primer apellido: Cabrera
- Segundo apellido: Gutierrez
- Edad: 24
- Religión: Budu
- Provincia de nacimiento: Holguin
- Organizaciones politicas: UJC PCC
- Pais de nacimiento: Cuba
- Organizadores de masas: CDR CTC FMC

A modal dialog box titled 'OK' is displayed in the center, containing a green checkmark icon and the text: 'Operación satisfactoria' and 'Se ha guardado un Interno en la base de datos'. Below the dialog, there is a list of aliases with 'Joseito' and 'Joseito' listed, and buttons for 'Adicionar' and 'Eliminar'. At the bottom right of the main form, there are navigation buttons: '1 de 2 Siguiente' and 'Cancelar'.

Anexo 2.3 Mensaje que se muestra al insertar un interno.

Anexo 3.Implementaciones

```
package DatosGenerales

class InternoService implements IGestionarInernoServices {

    void insertarInterno(Interno nuevoInterno)
    {

        if (nuevoInterno.save(flush: true)){
            request.getSession().setAttribute("Id_del_Interno", nuevoInterno.id)
            def nacionalidad = new InternoNacionalidad()
            nacionalidad.nacionalidad = Comun.nomencladores.NomNacionalidad.get(params.nacionalidad)
            nacionalidad.interno = nuevoInterno
            if (nacionalidad.save(flush: true)){
                println "se ha insertado la nacionalidad del interno"
            }
            request.getSession().setAttribute("CDR", params.cdr)
            request.getSession().setAttribute("CTC", params.ctc)
            request.getSession().setAttribute("FMC", params.fmc)
            request.getSession().setAttribute("UJC", params.ujc)
            request.getSession().setAttribute("PCC", params.pcc)

        }

    }

}
```

Anexo 3.1 Implementación del servicio que se utiliza para crear un Interno.

```
package DatosGenerales

import Comun.nomencladores.*
import nomencladores.*

class InternoController {

  def IGestionarInternoServices
  def insertar = {
    def nuevoInterno = new Interno()

    nuevoInterno.edad = Integer.parseInt(params.edad)

    def sex = NomSexo.get(params.sexo)
    nuevoInterno.sexo = sex.sexo
    nuevoInterno.personasAbrigo = params.personasAbrigo
    nuevoInterno.fechaNacimiento = new Date(Long.parseLong(request.getParameter("fecha")));
    println new Date(Long.parseLong(request.getParameter("fecha")));
    nuevoInterno.primerNombre = params.primerNombre
    nuevoInterno.segundoNombre = params.segundoNombre
    nuevoInterno.primerApellido = params.primerApellido
    nuevoInterno.segundoApellido = params.segundoApellido
    nuevoInterno.numeroid = params.numeroid
    nuevoInterno.paisNacimiento = NomPais.findById(params.paisNacimiento)
    nuevoInterno.estadoCivil = NomEstadocivil.findById(params.estadoCivil)
    nuevoInterno.documentoid = NomDocumentoid.findById(params.documentoid)
    nuevoInterno.provinciaNacimiento = NomProvincia.findById(params.provinciaNacimiento)

    internoService.insertarInterno(nuevoInterno)
  }
}
```

Anexo 3.2 Implementación del controlador que se utiliza para crear un Interno.

```

3 def salvarDatosFamiliares = {
    def Interno = Interno.get(request.getSession().getAttribute("Id_del_Interno"))

    //Aqui se establece la relacion interno - madre
    if (!params.primerNombreMadre.isEmpty() && !params.primerApellidoMadre.isEmpty()){
        def madre = new Vinculo()

        madre.primerNombre = params.primerNombreMadre
        madre.segundoNombre = params.segundoNombreMadre
        madre.primerApellido = params.primerApellidoMadre
        madre.segundoApellido = params.segundoApellidoMadre
        madre.fechaRegistro = new Date()

        //Estos datos los recojo estaticos porque seran utilizados mas adelante.

        madre.interno = false
        madre.telefono = "032 851306"
        madre.numeroid = "618719"
        madre.nacionalidad = NomNacionalidad.list().get(0)
        madre.documentoid = NomDocumentoid.list().get(0)
        madre.direccion = Direccion.list().get(0)
        // rellenadno los datos

        if (madre.save(flush: true)){
            def intVin = new InternoVinculo()

            intVin.parentesco = NomParentesco.findByParentesco("madre")
            intVin.interno = Interno
            intVin.vinculo = madre
            intVin.fechaVinculo = new Date()

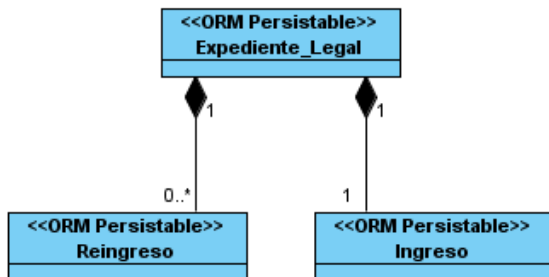
            //Estos datos los recojo estaticos porque seran utilizados mas adelante.
            intVin.observaciones = "dsa"
            intVin.loVisita = false
            intVin.motivoNovisita = "as"
            intVin.lugarVinculo = "asd"
            intVin.influencia = "asd"
            intVin.motivoVinculo = nomencldores.NomMotivovinculo.list().get(0)

            if (intVin.save(flush: true)){

```

Anexo 3.2 Parte de código del método que se utiliza para guardar los vínculos familiares.

Anexo 4. Diagrama de clases persistentes.



Anexo 4.1. Diagrama de clases persistentes del módulo Ingreso