Universidad de las Ciencias Informáticas

Facultad 2

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas.

Título: Desarrollo de los módulos Solicitudes y Decisiones del SIDEP.

Autor: Elián González Hernández

Tutor: Ing. Dasiel Cordero Morales

Co-tutor: Ing. Alejandro Santana Coterón

Consultante: MSc. Yoelvis Osés Sosa

La Habana 2012

“Año 54 de la Revolución”

"…hagamos el propósito de redoblar nuestro esfuerzo, y juremos ante nosotros mismos que si un día nuestro trabajo nos pareciera bueno, debemos luchar por hacerlo mejor; y si fuera mejor, debemos luchar por hacerlo perfecto, conociendo de antemano que para un Comunista nada será nunca suficientemente bueno y ninguna obra humana será jamás suficientemente perfecta".

Fidel Castro Ruz,

III Congreso del Partido Comunista de Cuba.

Declaración de Autoría

Declaro ser autor de la presente tesis y concedo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2012.

Elián González Hernández

_____

Firma del Autor

Ing. Dasiel Cordero Morales

_____

Firma del Tutor

Ing. Alejandro Santana Coterón

_____

Firma del Cotutor

MSc. Yoelvis Osés Sosa

_____

Firma del Consultante

Agradecimientos

Hoy es un día especial para mí, ya han pasado cinco cursos de sacrificios, sueños, alegrías y ha llegado el momento de enfrentarme al mundo como un profesional, son innumerables las personas a las que tengo que agradecerles por ayudarme a llegar aquí.

Primeramente agradecer a mis padres Ana y Eligio que siempre me han apoyado y dado fuerzas en los momentos difíciles, a ellos que siempre han sabido sacar lo mejor de mí, brindándome todo su amor y cariño, desear mejores padres sería imposible.

Sandra gracias por todo el amor, el cariño que me has dado y sobre todo la paciencia que has tenido.
A mis amigos que siempre han estado ahí para mí y a los que conocí luego en la Universidad.

Al equipo de desarrollo de SIDEP que me ayudaron a que este trabajo pueda mostrar hoy sus resultados.

Muchas gracias a todos aquellos que de una forma u otra ayudaron a la realización de este trabajo.

Dedicatoria

A mis padres, a mi familia y a mi novia que han estado apoyándome en todo momento, que han sido mi guía a lo largo de los años, por su paciencia, su confianza en mí, por cada granito de arena que pusieron en mi vida y en mi futuro para ustedes es este regalo.

Resumen

En el año 2009 se decide crear el proyecto Prisiones Cuba que tiene como objetivos fundamentales unir los tres sistemas existentes en la Dirección de Establecimientos Penitenciarios así como digitalizar toda la información y solucionar las demoras en la comunicación entre los distintos niveles de mando. Como parte del Sistema Informativo de la Dirección de Establecimientos Penitenciarios (SIDEP), se hace necesario gestionar las solicitudes enviadas desde los diferentes módulos, las cuales deben ser analizadas en tiempo para que se tome la decisión correcta y las decisiones que no dependan de una solicitud sean tramitadas en el menor tiempo posible.

El presente trabajo tiene como objetivo el desarrollo de los módulos Solicitudes y Decisiones del subsistema Registro Legal del SIDEP para contribuir a la gestión de la información referente a las solicitudes y decisiones en el sistema penitenciario cubano. Se tomó como punto de partida el análisis de los requerimientos de software establecidos en acuerdo con el cliente, dando paso a la obtención del diseño de la solución, su posterior implementación y comprobación de su funcionamiento a través de las pruebas de software.

Para el desarrollo de la aplicación web fue empleado: RUP como metodología de desarrollo de software, Java como plataforma de desarrollo, Visual Paradigm como herramienta CASE y Spring Source Tool Suite como Entorno de Desarrollo Integrado para la codificación y gestión del código fuente en general. Grails fue definido como framework de desarrollo a utilizar y como lenguajes de programación Groovy y Java.

Palabras Clave: Decisiones, Registro Legal, Interno, Solicitudes, Gestión.

Índice

Índice 7
Introducción 10
Fundamentación teórica 15
1.1 Introducción 15
1.2 Conceptos Fundamentales 15
1.3 Flujo de solicitudes y decisiones en el Sistema Penitenciario Cubano 17
1.4 Soluciones existentes relacionadas con los Sistemas Penitenciarios 20
1.4.1 OFFENDERTRAK Corrections Management System 21
1.4.2 Jail Administration and Management Software (JAMS) 22
1.4.3 Spillman Corrections Management System 23
1.4.4 Sistema de Gestión Penitenciario (SIGEP) 23
1.4.5 Sistema de Gestión Penitenciaria (SIGPEN) 24
1.4.6 Sistema Penitenciario del Gobierno de Panamá 25
1.4.7 Sistemas Informáticos en el Sistema Penitenciario Cubano 25
1.5 Metodología, tecnologías y herramientas 26
1.5.1 Metodología de Desarrollo: Rational Unified Process (RUP) 26
1.5.2 Unified Modeling Language (UML) 27
1.5.3 Herramienta CASE: Visual Paradigm 5.0 28
1.5.4 Integrated Development Environment (IDE) 28
1.5.5 Plataforma Java 29
1.5.6 Java Virtual Machine (JVM) 29
1.5.7 Lenguaje de Programación Groovy 29
1.5.8 Framework de desarrollo: Grails 1.3.7 30
1.5.9 Hyper Text Markup Languaje (HTML) 31
1.5.10 Lenguaje JavaScript 32
1.5.11 Framework de presentación: Dojo 1.5 32
1.5.12 Sistema Gestor de Base de Datos: Oracle 11g 32
1.5.13 Embarcadero ERStudio 8.0 33
1.5.14 Apache Tomcat (6.0.25) 33

1.6 Conclusiones parciales 33
Diseño del sistema 35
2.1 Introducción 35
2.2 Descripción de las funcionalidades del módulo Solicitudes 35
2.3 Descripción de las funcionalidades del módulo Decisiones 37
2.4 Marco de trabajo de desarrollo web 39
2.4.1 Arquitectura n-capas del sistema 40
2.5 Patrones de diseño 40
2.5.1 Modelo – Vista – Controlador (MVC) 41
2.5.2 Experto 44
2.5.3 Controlador 45
2.5.4 Creador 46
2.6 Diagrama de Clases 46
2.6.1 Diagrama de clases del diseño 46
2.6.2 Descripción de las clases 48
2.7 Diagrama de colaboración 53
2.8 Diseño de la Base de datos 54
2.9 Entidades más importantes presentes en el diseño de la base datos de los módulos solicitudes y decisiones 55
2.10 Conclusiones Parciales 56
Implementación y Pruebas 58
3.1 Introducción 58
3.2 Implementación 58
3.3 Diagrama de Componentes 58
3.3.1 La capa Vista 62
3.3.2 La capa Control 62
3.3.3 Componente Servicio 62
3.3.4 Componente Dominio 62
3.3.5 Implementación de las clases del Dominio 62
3.3.6 Validaciones 63

3.3.7 Operaciones sobre el modelo de datos 63
3.3.8 Internacionalización 64
3.4 Diagrama de Despliegue 64
3.5 Pruebas 66
3.6 Estrategia de Prueba 66
3.7 Niveles de Pruebas 67
3.8 Tipo de Prueba 68
3.9 Métodos de Pruebas 68
3.10 Estrategia de prueba seguida 68
3.11 Resultados 69
3.12 Conclusiones Parciales 73
Conclusiones Generales 74
Recomendaciones 75
Referencias Bibliográficas 76
Bibliografía 78
Anexo 1 ..... **¡Error! Marcador no definido.**
Anexo 2 ..... **¡Error! Marcador no definido.**
Anexo 3 ..... **¡Error! Marcador no definido.**
Anexo 4 ..... **¡Error! Marcador no definido.**
Anexo 5 ..... **¡Error! Marcador no definido.**

Introducción

Luego del triunfo de la revolución el 1ro de enero de 1959 el gobierno revolucionario toma un grupo de medidas con el objetivo de revertir la situación de malestar existente en Cuba como resultado del legado de la tiranía. Una de las medidas tomadas en aras de hacer un ordenamiento jurídico y crear nuevas formas para enfrentar los delitos fue la renovación el sistema penitenciario existente en aquel entonces.

Muchas de las antiguas prisiones heredadas del gobierno precedente, donde el abuso y las torturas constituían los métodos y procedimientos que caracterizaban la estancia de los detenidos en la prisión, fueron desactivadas construyéndose un sistema penitenciario basado en el respeto a los valores humanos y en la gestión del control inspirada en la regulación de normas y leyes internacionales en cuanto al tratamiento a los reclusos.

Como parte de esta política de mejoras en el año 1989 comienza el sistema de informatización de los centros penitenciarios, con la automatización de los datos principales del recluso. En el año 2002 culmina el desarrollo del Sistema Automatizado para el Control del Recluso (SACORE), pero no fue hasta enero del 2003 que finaliza su despliegue y se comienza a utilizar en todo el Sistema Penitenciario Cubano.

El SACORE posee tres módulos principales: Control Penal, Reeducación Penal y Orden Interior, recogiendo los principales datos del recluso y los aspectos penales existentes. A pesar de las facilidades brindadas por este sistema aún existen áreas en las cuales no han quedado informatizados los procesos en los que se ven envueltas; como son los equipos multidisciplinarios y servicios médicos.

Junto a este sistema en los centros penitenciarios se utilizan otros dos sistemas tales como el Sistema Automatizado de Capacidades de la Dirección de Establecimientos Penitenciarios (SACDEP) y el Sistema de Automatización de Incidencias de la Dirección de Establecimientos Penitenciarios (SAIDEP), estos dos últimos también insuficientes para lograr una gestión de la información que en dicha institución se maneja.

Con motivo del 50 aniversario del Ministerio del Interior (MININT) surge el plan 20 x 50, que impulsa la modernización tecnológica y la colaboración con las Universidades. En dicho plan se decide por parte de la Jefatura del MININT desarrollar en conjunto con la Universidad de las Ciencias Informáticas un

proyecto productivo con el propósito de informatizar toda la gama de procesos que se controlan en una unidad penitenciaria, de ahí el surgimiento del Sistema Informativo de la Dirección de Establecimientos Penitenciarios (SIDEP).

Esta aplicación tiene como objetivo unir los tres sistemas existentes en la Dirección de Establecimientos Penitenciarios (DEP), el SACORE, el SACDEP y el SAIDEP, además de adicionar un conjunto de funcionalidades necesarias para la gestión de los procesos en los centros penitenciarios, aprovechando el desarrollo de las tecnologías de la informática y las comunicaciones que tiene lugar en el país. La misma cuenta con un subsistema denominado Registro Legal, encargado de gestionar las bases legales del sistema penitenciario cubano en cada centro.

Como parte del funcionamiento del Sistema Penitenciario y específicamente dentro de la gestión de las bases legales de este, se contempla la gestión de las solicitudes, que son peticiones que realizan los centros penitenciarios, los internos y sus familiares a la Dirección de Establecimientos Penitenciarios. Estas pueden ser de otorgamiento de beneficios, de traslado interpenal, conducciones, entre otras. Además, se incluye el registro y control de dichas solicitudes y los documentos que las avalan.

Luego, las decisiones son las acciones ordenadas por las autoridades judiciales competentes, los órganos provinciales o la DEP en función de las solicitudes o de manera independiente. Las decisiones se comunican a las jefaturas provinciales o centros penitenciarios a través de documentos de carácter legal, estas determinan la ejecución obligatoria de procesos en los establecimientos penitenciarios.

En el sistema penitenciario cubano actualmente existen retrasos en la comunicación de los distintos niveles de mando y algunas de las solicitudes son muy delicadas en cuanto a las consecuencias que pueden acarrear, tanto para el interno como para la dirección del Sistema Penitenciario, tales como:

- Si algún familiar del recluso se encuentra en su fase terminal y la solicitud que se realiza para trasladar a dicho recluso hasta el hospital o la decisión para esta solicitud se retrasa y el familiar muere causaría un grave daño psicológico y emocional al recluso además de convertirse en una grave violación de los derechos humanos por parte de la dirección del sistema penitenciario.
- Cuando se solicita el internamiento de un recluso en un área de seguridad incrementada y no se especifica detalladamente el motivo, esto podría influir en el aumento del tiempo de condena del mismo por una mala conducta.

- La solicitud de requisa general requiere de una buena comunicación entre los distintos eslabones de la cadena de mando pues si existe algún retraso para tomar la decisión podría causar algún altercado en el centro penitenciario que debía ser prevenido con esta requisa.
- Cuando se solicita un traslado o una conducción no puede existir ningún problema en los datos, pues podría enviarse al interno a una localización errónea o bien el interno podría fugarse.

Por todo lo anteriormente planteado surge como **problema:** ¿Cómo contribuir a la gestión de las solicitudes y decisiones a los diferentes niveles de mando de la Dirección de Establecimientos Penitenciarios?

Teniendo en cuenta lo anterior, el **objeto de estudio** se enmarca en la gestión de la información en los Sistemas Penitenciarios. El **campo de acción** está centrado en la gestión de solicitudes y decisiones en el Sistema Penitenciario Cubano.

Para dar solución a la problemática anterior se define como **objetivo general:** Desarrollar los módulos Solicitudes y Decisiones para contribuir a la gestión de las solicitudes y decisiones en el subsistema Registro Legal del SIDEP.

La **idea a defender** durante la investigación es: El desarrollo de los módulos Solicitudes y Decisiones del SIDEP contribuye a la gestión de las solicitudes y decisiones a los diferentes niveles de mando de la Dirección de Establecimientos Penitenciarios.

El objetivo anterior, se desglosa en los siguientes **objetivos específicos:**

- Diseñar los módulos Solicitudes y Decisiones del subsistema Registro Legal del SIDEP para facilitar la implementación de los mismos.
- Implementar los módulos Solicitudes y Decisiones para gestionar las solicitudes y decisiones en los diferentes niveles de mando de la DEP.
- Comprobar el funcionamiento de los módulos Solicitudes y Decisiones del subsistema Registro Legal del SIDEP, a través de la realización de pruebas de calidad.

Para dar cumplimiento al objetivo general se plantean las siguientes **tareas de la investigación:**

- Análisis de las soluciones informáticas nacionales e internacionales relacionadas con la gestión de información en sistemas penitenciarios.

- Caracterización de la metodología, tecnologías y herramientas que tributan al desarrollo del sistema.
- Confección del diagrama de clases del diseño correspondiente a los módulos Solicitudes y Decisiones del SIDEP.
- Confección del Modelo de la base de datos correspondiente a los módulos Solicitudes y Decisiones del SIDEP.
- Confección de los diagramas de colaboración pertenecientes al diseño de los módulos Solicitudes y Decisiones del SIDEP.
- Elaboración del diagrama de componentes de los módulos Solicitudes y Decisiones del SIDEP.
- Elaboración del diagrama de despliegue de los módulos Solicitudes y Decisiones del SIDEP.
- Implementación de la gestión de solicitudes y decisiones como parte del SIDEP.
- Confección de los casos de pruebas, para la realización de pruebas de software.
- Realización de las pruebas de Caja Negra para los módulos Solicitudes y Decisiones del SIDEP.

Para desarrollar las tareas anteriormente planteadas se utilizarán los **métodos de investigación** siguientes:

Métodos teóricos

- Análisis - Síntesis: Se empleó durante la revisión de los documentos consultados en la investigación para conocer todo lo relacionado con el proceso de gestión de las solicitudes y decisiones en el Sistema Penitenciario Cubano.
- Histórico-Lógico: A través de este método se identificaron las principales etapas por las que ha transitado el Sistema Penitenciario Cubano y la evolución del mismo con la introducción de las TICs.

El presente documento se estructura en: Resumen, Introducción, Capítulo 1: Fundamentación teórica, Capítulo 2: Diseño, Capítulo 3: Implementación y Pruebas, Conclusiones, Recomendaciones, Referencias Bibliográficas y Anexos. En los capítulos se aborda lo siguiente:

Capítulo 1: Fundamentación teórica

Presenta un estudio referente a la situación actual del Sistema Penitenciario en Cuba y de otras partes del mundo, haciendo énfasis en los procesos de solicitudes y decisiones. Se abordan además las tecnologías utilizadas, metodologías de desarrollo y las herramientas fundamentales para la solución del problema.

Capítulo 2: Diseño

En este capítulo se explica cómo está estructurada la arquitectura de la aplicación que se implementa, los patrones de diseño que se utilizan y se describen las funcionalidades para las cuales responden los módulos Solicitudes y Decisiones. Se expone una representación de los diagramas de clases e interacción, así como el diseño de la base de datos para la persistencia de la información.

Capítulo 3: Implementación y Pruebas

En este capítulo se presentan los artefactos correspondientes al flujo de Implementación, además, la estrategia y los resultados de las pruebas realizadas a los módulos implementados.

Fundamentación teórica

1.1 Introducción

En este capítulo son expuestos conceptos fundamentales correspondientes al negocio que se maneja. Se analizan además sistemas existentes a nivel internacional y nacional respecto a la gestión de información en establecimientos penitenciarios. Se exponen las herramientas, tecnologías y lenguajes definidos para el desarrollo de la solución, así como una descripción general del proceso a automatizar.

1.2 Conceptos Fundamentales

Se define como Sistema Penitenciario a la organización encargada 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"*. (1)

Cada estado estipula dichas organizaciones basándose en sus concepciones políticas y adecuándose a las condiciones sociales y económicas de su desarrollo. En Cuba este sistema está dirigido por la Dirección de Establecimientos Penitenciarios del Ministerio del Interior y *"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)

Los fundamentos de la política penitenciaria están determinados en la Constitución de la República de Cuba, La Ley 62 del Código Penal, además "*la organización y las condiciones de ejecución de las sanciones privativas de libertad se corresponden con lo establecido en las "Reglas Mínimas Clásicas para el Tratamiento a los Reclusos" aprobadas el 30 de agosto de 1955 por la Organización de las Naciones Unidas (ONU) y la Declaración Universal de los Derechos del Hombre del 10 de diciembre de 1948*". (2)

En todo Sistema Penitenciario constituye un elemento fundamental la gestión de solicitudes y decisiones, lo que debe ser considerado en aras de proporcionar un mecanismo capaz de resolver en alguna medida el envío de las solicitudes así como las decisiones generadas a partir de éstas o como

parte de la cadena de mando de este tipo de establecimientos. En este contexto, se define como solicitudes y decisiones:

Solicitudes: "*Son peticiones que realizan los internos, sus familiares y los centros penitenciarios a la Dirección de Establecimientos Penitenciarios"*. (1)

Decisiones: "*Son las acciones ordenadas por las autoridades judiciales competentes, los órganos provinciales o la Dirección de Establecimientos Penitenciarios (DEP). Las decisiones se comunican a las jefaturas provinciales o centros penitenciarios a través de documentos de carácter legal, estas determinan la ejecución obligatoria de procesos en los establecimientos penitenciarios".* (1)

Para abordar cualquier tema sobre la gestión de la información en el sistema penitenciario, en particular sobre la gestión de las solicitudes y decisiones es necesario conocer algunas definiciones del vocabulario jurídico penitenciario. Estas se emplean en el momento de realizar cualquier búsqueda de información sobre los internos y es necesario conocerlas para emitir una solicitud o decisión. Tal es el caso de la clasificación de un interno, pues solo pueden emitirse solicitudes o decisiones sobre los internos sancionados; pudiéndose consultar esta información desde cualquier nivel de la cadena de mandos del sistema penitenciario cubano mediante un único expediente legal que se le asigna a cada interno.

Ingreso: Se considera ingreso a *"… toda remisión al lugar de internamiento de acusado, sancionado y asegurado por orden de la autoridad judicial competente, mediante la documentación establecida y debidamente identificado"*. (1)

Los internos, al ingreso, se pueden clasificar de varias formas:

- **Acusado:** "*Es la persona a quien se le haya decretado la medida cautelar de prisión provisional"*. (1)
- **Sancionado:** "*Es la persona ejecutoriamente sancionada a privación de libertad o a Trabajo Correccional Con Internamiento (TCCI)".* (1)
- **Asegurado:** "*Es la persona a quien se le hubiere impuesto una medida de seguridad reeducativa de internamiento".* (1)

El Expediente Legal representa un punto de acceso a toda la información de una persona que ha ingresado al Sistema Penitenciario. El mismo está conformado por "*… documentos legales expedidos por las autoridades competentes que amparan su internamiento, en él se recogen los aspectos más sobresalientes de la trayectoria penal y penitenciaria del interno. Contiene los documentos que se remiten con el detenido, así como los que se obtienen y confeccionan a su ingreso, durante su permanencia y egreso, lo cual permite el control del acusado, sancionado o asegurado"*. (1)

Al entrar al Sistema Penitenciario Cubano por primera vez, a los internos se les crea un expediente legal. El cual puede transitar por dos estados, abierto o cerrado. Se dice que el expediente está abierto, cuando el interno haya efectuado un ingreso y esté transitando por el Sistema Penitenciario, cumpliendo sanción dentro de un Centro Penitenciario. Al cumplir la sanción dictada el expediente toma el estado de cerrado y no se vuelve a trabajar con él. Estando abierto, puede tomar otros dos estados: habilitado o deshabilitado. Está habilitado, cuando el interno está cumpliendo la sanción dentro de un Centro Penitenciario y deshabilitado cuando está fuera del centro penitenciario, tal es el caso de la libertad condicional.

Es necesario destacar que las solicitudes y decisiones que se relacionan con un interno solo se pueden emitir sobre un expediente legal que esté en estado abierto y esté habilitado.

1.3 Flujo de solicitudes y decisiones en el Sistema Penitenciario Cubano

Las solicitudes y decisiones que se emiten en el sistema penitenciario cubano pueden transitar por los cuatro niveles de mando de la DEP. El primer nivel es el de colectivo y en él intervienen el oficial educador guía y los internos, en el mismo no se pueden emitir decisiones. El segundo nivel es el de unidad y en él intervienen el jefe de unidad y los oficiales que pertenecen a distintos departamentos fundamentalmente el departamento de registro legal y el de tratamiento educativo. El tercer nivel es el provincial y en él se agrupan todas las jefaturas pertenecientes a éste, existiendo una por cada provincia. En el último nivel -el nacional- solo se encuentra la Dirección Nacional de Establecimientos Penitenciarios única en Cuba, en la misma no se pueden emitir solicitudes.

A continuación se describen los tipos de solicitudes que pueden ser emitidas en el sistema penitenciario cubano:

- **Traslado interprovincial:** El solicitante es el jefe del centro penitenciario y el encargado de evaluarla es el jefe de registro legal de la DEP. Su objetivo es solicitar el traslado de los internos que se seleccionen hacia un centro de otra provincia.
- **Traslado interpenal:** El solicitante es el jefe del centro penitenciario y el encargado de evaluarla es el jefe del órgano provincial de establecimientos penitenciarios. Su objetivo es solicitar el traslado de los internos que se seleccionen hacia otro centro perteneciente a la misma provincia.
- **Conducción:** El solicitante es el educador guía o el oficial de atención a la ciudadanía y el encargado de evaluarla es el jefe de tratamiento penitenciario. Tiene como objetivo solicitar la conducción de los internos que se seleccionen hacia un lugar específico tales como la funeraria o el hospital.
- **Ampliación o reducción de capacidades:** El solicitante es el oficial de aseguramiento multilateral y el encargado de evaluarla es el jefe de la DEP. Tiene como objetivo solicitar una obra de ampliación o reducción de capacidades en un determinado centro penitenciario.

Las decisiones pueden ser el resultado de la evaluación de una solicitud o bien emitirse de forma independiente. Es importante destacar que las decisiones no podrán emitirse desde el nivel de colectivo. A continuación se describen los tipos de decisiones que se pueden gestionar en el sistema penitenciario cubano:

- **Suspensión de permisos:** Este tipo de decisión no depende de una solicitud. El responsable de tomar esta decisión es el jefe de la DEP y la reciben los jefes de centros penitenciarios y tratamiento educativo. Tiene como objetivo ordenar una suspensión de los permisos en la estructura que se indique.
- **Cantidad de internos con Permiso Especial de Salida al Hogar (PESH):** Este tipo de decisión no depende de una solicitud. El responsable de tomar esta decisión es el jefe del

centro penitenciario y la recibe el jefe del sistema educativo. Tiene como objetivo decidir la cantidad de internos que tendrán un permiso especial de salida al hogar.

- **Otorgamiento de licencia extrapenal:** Este tipo de decisión no depende de una solicitud. El responsable de tomar esta decisión es el jefe de la DEP y la reciben el jefe de registro legal y el jefe del centro. Tiene como objetivo otorgar una licencia extrapenal a un interno determinado.
- **Conducción por circular 115:** Este tipo de decisión no depende de una solicitud. El responsable de tomar esta decisión es el jefe de registro legal de la DEP y la reciben el jefe de registro legal y el jefe del centro. Esta decisión es una orden especial para conducir solamente a un interno a juicio indicándose entre otros datos la provincia y el tribunal al que será conducido.
- **Traslado por circular 115:** Este tipo de decisión no depende de una solicitud. El responsable de tomar esta decisión es el jefe de registro legal de la DEP y la reciben el jefe de registro legal y los jefes de centro de las unidades que intervienen en el traslado del interno. Esta decisión es una orden especial para ejecutar un traslado interpenal o interprovincial de solamente un interno.
- **Prohibición de cambio de ubicación:** Este tipo de decisión no depende de una solicitud. Los responsables de tomar esta decisión son el jefe del centro penitenciario, el jefe del órgano provincial de establecimientos penitenciarios o el jefe de la DEP. Siendo los responsables de recibirla el jefe del centro (en caso de que no sea el mismo quien haya creado la decisión), el jefe de tratamiento penitenciario, el jefe de orden interior y el jefe de registro legal. Esta decisión es una orden que prohíbe el cambio de ubicación de un interno en específico.
- **Traslado interpenal:** Este tipo de decisión se puede emitir a partir de la evaluación de una solicitud o de manera independiente. El responsable de tomar esta decisión es el jefe de registro legal de la DEP y la reciben el jefe de registro legal y los jefes de centro de las unidades que intervienen en el traslado del interno. Su objetivo es solicitar el traslado de los internos que se seleccionen hacia otro centro perteneciente de la misma provincia.

- **Traslado interprovincial:** Este tipo de decisión se puede emitir a partir de la evaluación de una solicitud o de manera independiente. El responsable de tomar esta decisión es el jefe de registro legal de la DEP y la reciben el jefe de registro legal y los jefes de centro de las unidades que intervienen en el traslado del interno. Su objetivo es solicitar el traslado de los internos que se seleccionen hacia un centro de otra provincia.

La figura que se muestra a continuación representa el recorrido de las solicitudes y decisiones por los diferentes niveles de mando de la dirección de establecimientos penitenciarios, donde la imagen de la izquierda muestra el flujo para las solicitudes y el de la derecha para las decisiones, como lo describe la leyenda.

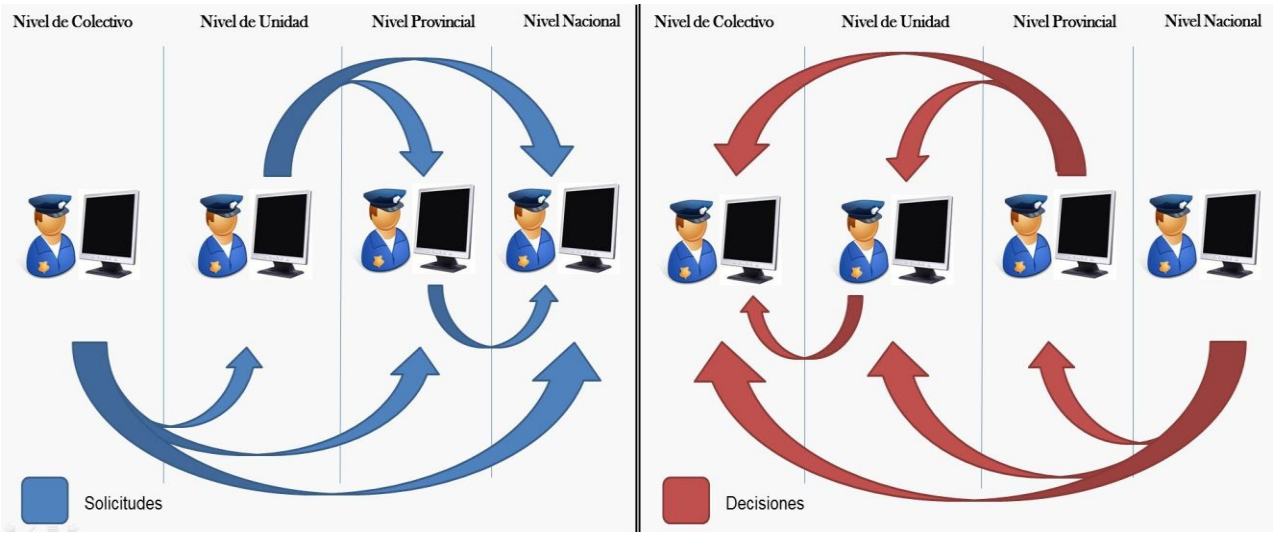

Figura 1: Recorrido de las solicitudes y decisiones por los cuatro niveles de mando de la DEP.

1.4 Soluciones existentes relacionadas con los Sistemas Penitenciarios

La informatización de las sociedades surge para resolver los problemas de almacenamiento y control de datos en los sistemas penitenciarios permitiendo una mayor eficiencia en la realización de los procesos dentro de estos centros. Aspectos como la seguridad, control y evaluación se han visto día a día amenazados por la evolución constante de los sistemas informáticos y el rápido incremento de las tecnologías. Viejos y rigurosos mecanismos de trabajo, así como los arduos procesos de gestión

existentes durante décadas en las prisiones, hoy día han sido sustituidos por soluciones informáticas capaces de gestionar dichos procesos con una mayor seguridad y mínima dificultad.

Los sistemas para la gestión penitenciaria 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 con un sistema de reportes, que les permite saber el estado de ciertos indicadores. A continuación se hace un análisis de algunos sistemas que permiten la gestión de centros penitenciarios. Se utilizaron los datos oficiales que se muestran en sus portales web pues no brindan información suficiente para hacer estudios más rigurosos. Además, la información y procesos que gestionan son sensibles por el negocio en cuestión, lo que se une a una estrategia para evitar la competencia con otras compañías que puedan plagiar su solución.

1.4.1 OFFENDERTRAK Corrections Management System[1]

OFFENDERTRAK Corrections Management System es una solución de software empresarial proporcionada por Motorola, que responde a las necesidades operativas y administrativas de centros penitenciarios. Esta herramienta permite "… *administrar, procesar, archivar, recuperar y compartir la información de los reclusos y sus incidencias así como gestionar gran parte de la información relativa a los procesos de estos centros*" (3). Este software es aplicable en departamentos de policía y centros penitenciarios. Entre las principales características de este software se encuentran:

- Los documentos son interactivos con capacidad de firma electrónica.
- Seguridad basada en roles.
- Reconocimiento facial.
- Identificación positiva utilizando las huellas dactilares.
- Soporte de código de barras.
- Registrar los datos personales de los individuos.
- Registrar proceso judicial de los individuos.
- Archivar el expediente del individuo una vez dado de libertad.

[1] De su traducción al español OFFENDERTRAK Sistema de Gestión Correccional.

En la fuente bibliográfica analizada no se mostró ninguna evidencia que demostrara o explicara la gestión de solicitudes y decisiones en este software, debido a que esto representa información específica sobre el negocio que se maneja en este sistema de gestión penitenciaria.

Motorola es una empresa estadounidense de gran prestigio internacional pero debido al bloqueo económico que se le ha impuesto a Cuba no se puede realizar ninguna transacción económica con esta institución, aunque si fuese posible, extenderla a todas las instalaciones del sistema penitenciario, por su alto precio implicaría un alto costo en licencias y servicios profesionales.

1.4.2 Jail Administration and Management Software (JAMS)[2]

Jail Administration and Management Software (JAMS) es un sistema web creado por la compañía CISCO que "… *se encarga de proveer la información necesaria para gestionar centros penitenciarios"* (4). Se hace referencia a que *"… está diseñado para satisfacer las necesidades de cualquier institución correccional*" (4). A continuación se describen algunos de los módulos que intervienen de alguna manera en la gestión de solicitudes y decisiones:

- **Procesamiento de celdas:** "*Mantiene dos archivos de datos, el primero mantiene un registro de todas las celdas, los bloques de celdas, los pisos dentro de la cárcel y la ocupación actual de cada uno. El segundo es un registro de transacciones que realiza el seguimiento del movimiento de los presos dentro y fuera de las celdas. Este registro actualiza automáticamente los registros de ocupación y se utiliza para generar una decisión de cambio de ubicación de los internos en el centro penitenciario*". (4)
- **Actualización Médica:** *"Se encarga de realizar el seguimiento de las prescripciones asignadas a los presos. Los datos incluyen la identificación del prisionero y el medicamento prescrito, la cantidad, la frecuencia y las fechas de inicio y finalización para las prescripciones. También se puede generar una solicitud a través del módulo de información del interno para enumerar todos los medicamentos que deben darse a los internos en una fecha determinada".*(4)

En estos dos módulos se hace referencia de forma breve a la gestión de solicitudes y decisiones en el sistema JAMS pero no se especifica nada sobre el flujo de las mismas dentro del sistema, ni se revela

[2] De su traducción al español Software de Gestión y Administración de Cárceles.

ningún detalle sobre la existencia de otras solicitudes o decisiones que tengan un comportamiento similar en el Sistema Penitenciario Cubano.

La compañía CISCO es reconocida a nivel mundial por la calidad de sus productos, es por eso que se incluye el sistema JAMS en la realización de esta investigación, aunque también, debido al bloqueo económico que se le ha impuesto a Cuba no se puede realizar ninguna transacción económica con esta empresa.

1.4.3 Spillman Corrections Management System[3]

Spillman Corrections Management System es un software de gestión penitenciaria que desarrolló la empresa Spillman y se utiliza en algunos centros penitenciarios de los Estados Unidos de América (EUA). "*Cuenta con más de cuarenta módulos que permiten personalizar un sistema de gestión penitenciaria para satisfacer las necesidades individuales de las instalaciones*. *En la esfera de la comunicación permite compartir información con entidades que no pertenecen al centro penitenciario como los departamentos de policía, los centros de comunicación y departamentos de bomberos para la interoperabilidad total de la seguridad pública".* (5)

En la información que se ofrece en la página oficial de esta compañía referente al producto Spillman Corrections Management System y sus más de cuarenta módulos no se menciona nada que pueda ser relacionado con el manejo de las solicitudes y decisiones dentro del centro penitenciario. Tampoco, debido al bloqueo económico que se le ha impuesto a Cuba se puede realizar ninguna negociación con dicha compañía, siendo imposible la utilización de este software.

1.4.4 Sistema de Gestión Penitenciario (SIGEP)

El SIGEP se crea para dar respuesta a las necesidades de gestión, información y apoyo a la Dirección Nacional de Servicios Penitenciarios (DNSP) de la República Bolivariana de Venezuela. Tiene como objetivo lograr un incremento de la confianza en el sistema penitenciario venezolano, generar estadísticas confiables y actualizadas sobre la situación jurídica de los privados de libertad, condiciones de vida y salud, actividades de rehabilitación y reinserción, la situación operativa y la actividad administrativa en centros penitenciarios. También permite la comunicación en línea con tribunales,

[3] De su traducción al español Sistema de Gestión de Correccional.

sistemas de identificación y antecedentes penales, que complementan la información necesaria para la gestión de los procesos.

Contiene un módulo Solicitudes que permite registrar las solicitudes y los documentos que las avalan. También contiene un módulo donde se gestionan las decisiones que se comunican a través de documentos de carácter legal. Estas implican el desarrollo de procesos en los establecimientos penitenciarios, tales como, traslados, egresos o ingresos.

Un aspecto a considerar es que las solicitudes se envían a entidades fuera del sistema penitenciario como son las fiscalías, tribunales y el Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC); siendo la Dirección Nacional de Servicios Penitenciarios (DNSP), la única entidad que pertenece al sistema penitenciario venezolano que recibe solicitudes. Todas las decisiones son emitidas desde los tribunales y solamente el DNSP está autorizado a emitir una decisión en caso de emergencia.

Estas características del flujo de solicitudes y decisiones que se implementan en el SIGEP son muy parecidas a las que presenta el sistema penitenciario cubano, su principal diferencia radica en que las solicitudes y decisiones del sistema penitenciario cubano solo pueden ser gestionadas por entidades que pertenezcan al mismo, como es el caso de la jefatura nacional y las jefaturas provinciales.

1.4.5 Sistema de Gestión Penitenciaria (SIGPEN)

El SIGPEN, es el sistema que se utiliza en Ecuador para la gestión penitenciaria. El mismo permite "… *obtener datos estadísticos reales de la situación de los sistemas penitenciarios y tiene como objetivos facilitar el seguimiento y manejo de las actividades que realizan los Centros de Rehabilitación Social (CRS) en cada una de sus áreas, logrando de esta manera tener un control adecuado y oportuno de la información de las personas privadas de libertad*" (6). El Ministerio de Justicia de Ecuador conjuntamente con la Dirección Nacional de Rehabilitación Social de Ecuador monitorean las actividades realizadas en los CRS por medio del sistema SIGPEN. "*Está compuesto por diferentes módulos (Interno, Departamento Jurídico, Departamento Médico, entre otros) los cuales brindan toda la información referente al estado de los reclusos*". (6)

En la documentación consultada referente a este sistema que aparece en la página oficial del Ministerio de Justicia de Ecuador no se hace ninguna mención sobre la gestión de las solicitudes y decisiones en

el funcionamiento de este sistema, ni se explica cómo se realiza la comunicación entre los distintos ministerios que utilizan este último.

1.4.6 Sistema Penitenciario del Gobierno de Panamá

El Sistema Penitenciario del Gobierno de Panamá fue creado a principios de año 1997, como resultado de un proyecto financiado por las Naciones Unidas y el gobierno español en coordinación con la Dirección General de Sistemas Penitenciarios de Panamá (DGSP). Este software tiene como objetivo *"… mantener en una base de datos los registros de los internos que están detenidos en los centros penales a nivel nacional. Entre estos registros se encuentran los datos personales, jurídicos, socioeconómicos y antecedentes médicos del interno".* (7)

"La información se registra en una base de datos que radica en la sede central y la cual está disponible para los diferentes departamentos que tiene acceso al sistema. Esto garantiza que se reflejen de forma inmediata los cambios en el expediente del interno tales como traspasos de autoridad, traslados de un centro a otro, libertades, diligencias médicas, jurídicas y la realización del cómputo automático de la sentencia". (7)

En la documentación consultada referente a este sistema se menciona la realización de traslados de los internos, diligencias médicas y jurídicas lo que en el sistema penitenciario cubano se traduce como traslados tanto interprovinciales, interpenales o conducciones a hospital, fiscalía o tribunal, pero no se realiza ninguna descripción sobre el proceso mediante el cual se solicitan o se deciden estas acciones.

1.4.7 Sistemas Informáticos en el Sistema Penitenciario Cubano

En los años 80 la información manejada en el Sistema Penitenciario Cubano se hacía de forma manual, lo que provocaba que el trabajo fuera mucho más lento y engorroso. Para mitigar este problema y junto con el auge de la informatización de la sociedad cubana, en cumplimiento de la orden 43/99 del Viceministro Primero del Ministerio del Interior de Cuba se deciden automatizar los procesos de las prisiones cubanas. De esta forma como producto se obtiene un sistema informático conocido como Sistema Automatizado para el Control del Recluso (SACORE) que en gran medida ayuda a gestionar los procesos de las prisiones.

Luego de implantado, se verificó que este sistema no logró abarcar todos los procesos que se ejecutaban en los establecimientos penitenciarios. Es por ello que fue necesario desarrollar otras dos

aplicaciones, surgiendo así el SAIDEP y el SACDEP que trabajan de manera conjunta con el SACORE. El SAIDEP gestiona el control y seguimiento de las incidencias o hechos extraordinarios que alteran el orden interior y afectan la seguridad de los establecimientos penitenciarios. Por otra parte el SACDEP permite definir la estructura física del área de reclusión penal y gestiona el control sobre las capacidades penitenciarias, los niveles de ocupación, la disponibilidad y las afectaciones de las instalaciones penitenciarias en Cuba.

En cuanto al proceso de gestión de solicitudes y decisiones en estos tres sistemas no se implementa ninguna solución, solamente en el SACORE se está analizando la posibilidad de gestionar solicitudes solamente para los traslados y las decisiones correspondientes a estas.

El análisis de los sistemas internacionales y nacionales anteriormente mencionados, arrojó que de forma general los desarrolladores ofrecen muy poca información sobre dichos sistemas. De estos, solo dos (JAMS y SIGEP) hacen alusión a la gestión de solicitudes y decisiones pero el flujo implementado difiere de las características del sistema penitenciario cubano. Además, en su mayoría son empresas norteamericanas, que por cuestiones del bloqueo se hace imposible la comercialización con las mismas.

1.5 Metodología, tecnologías y herramientas

A continuación se realiza una breve descripción de la metodología, las herramientas y tecnologías definidas, lo cual constituye uno de los pilares fundamentales para elaborar un software con calidad y en el tiempo requerido.

1.5.1 Metodología de Desarrollo: Rational Unified Process (RUP)[4]

Una metodología de desarrollo de software abarca los disímiles procedimientos y técnicas que deben seguirse para desarrollar un software. Concretamente "*define quién está haciendo qué, cuándo hacerlo y cómo alcanzar cierto objetivo*" (8). La metodología de desarrollo de software se encarga de "… *elaborar estrategias que están centradas en las personas o los equipos y orientadas hacia la*

[4] De su traducción al español Proceso Unificado de Desarrollo.

funcionalidad y la entrega; su objetivo es elevar la calidad del software a través de un mayor control sobre el proceso de desarrollo del software". (8)

La metodología de desarrollo de software seleccionada para el desarrollo de la solución fue RUP, que al ser una metodología pesada puede ser utilizada en proyectos de gran envergadura como el SIDEP, en el que se captura una gran cantidad de requisitos y en el cual intervienen un gran número de personas.

El volumen de documentación que genera RUP durante el proceso de desarrollo del software es una característica que puede ser aprovechada por personas con poca experiencia que necesitan de esta para conocer el trabajo realizado en otras fases.

1.5.2 Unified Modeling Language (UML)[5]

UML es un lenguaje para *"… visualizar, especificar, construir y documentar los artefactos de un sistema*. *Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. Además de permitir la modelación de sistemas con tecnología orientada a objetos*". (9)

Los diagramas pueden ser definidos como *"… entes importantes del UML, cuya finalidad es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo*. *Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementarlo. El modelo gráfico de UML tiene un vocabulario en el que se identifican: elementos, relaciones y diagramas*". (9)

En el desarrollo de la aplicación se utilizará UML como lenguaje de modelado debido a la experiencia que se posee en su utilización. Esto permite distribuir una mayor cantidad de tiempo para modelar el diseño de la solución a desarrollar así como obtener y documentar los artefactos derivados del trabajo realizado para el cumplimiento de los objetivos planteados.

[5] De su traducción al español Lenguaje Unificado de Modelado.

1.5.3 Herramienta CASE[6]: Visual Paradigm 5.0

Esta herramienta puede servir de apoyo *"… en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como el proceso de realizar un diseño del proyecto, documentación, detección de errores, entre otras".* (10)

"Visual Paradigm es una herramienta CASE profesional, fácil de utilizar y que soporta el ciclo de vida completo del desarrollo de software; usa al UML como lenguaje de modelado lo cual ayuda a la construcción de aplicaciones con calidad, de manera intuitiva y a menor coste; además de que facilita la comunicación entre los miembros del equipo de desarrollo al garantizar el uso de un lenguaje estándar común". (10)

Se decidió utilizar esta herramienta CASE para el modelado del diseño de la aplicación, ya que entre sus principales características y facilidades se encuentran:

- Fácil de instalar, actualizar y usar.
- Soporta el ciclo de vida completo del desarrollo de software.
- Propicia el modelado con el lenguaje seleccionado, además de proporcionar un apoyo importante en la generación de la documentación referente a los artefactos generados en las fases de desarrollo del sistema que se llevan a cabo en este trabajo.

1.5.4 Integrated Development Environment (IDE)[7]

Un IDE es un programa informático compuesto por *"… un conjunto de herramientas de programación que a su vez puede dedicarse en exclusiva a un sólo lenguaje de programación o utilizarse para varios lenguajes"* (11).

"SpringSource Tools Suite 2.5 (STS) está basado en Eclipse, pero además incorpora muchas herramientas y asistentes cuyo objetivo es el de facilitar y agilizar el desarrollo de aplicaciones, las cuales obviamente utilicen tecnologías como Spring Framework y Spring Web Flow". (11)

[6] Ingeniería de Software Asistida por Computadora. Por sus siglas en inglés Computer Aided Software Engineering (CASE).

[7] De su traducción al español Entorno de Desarrollo Integrado.

La principal característica por lo cual se definió la utilización de STS como IDE para el desarrollo del SIDEP es la experiencia que se posee en la utilización de este software para el desarrollo de aplicaciones, lo que disminuye la curva de aprendizaje y permite que se emplee una mayor cantidad de tiempo en otras tareas.

1.5.5 Plataforma Java

Una plataforma de desarrollo lo constituye "… *el entorno común en el cual se desenvuelve la programación de un grupo definido de aplicaciones, la plataforma Java es un entorno o plataforma de computación creada por la empresa Sun Microsystems, es capaz de ejecutar aplicaciones desarrolladas con el uso del lenguaje de programación Java y un conjunto de herramientas de desarrollo. En este caso, la plataforma no es un hardware específico o un sistema operativo, sino una máquina virtual encargada de la ejecución, y un conjunto de librerías estándares que ofrecen funcionalidades comunes".* (12)

Para el desarrollo de la aplicación se seleccionó la Edición Empresarial: J2EE o Java EE (Java Enterprise Edition), porque es una plataforma madura utilizada para el desarrollo de aplicaciones web, con una amplia gama de proveedores, documentación y principalmente porque se posee experiencia con el lenguaje Java.

1.5.6 Java Virtual Machine (JVM)[8]

El término JVM se refiere a *"… la especificación abstracta de una máquina de software para ejecutar programas Java"* (13). Este es el núcleo de ese lenguaje de programación, por lo que resulta imposible ejecutar cualquier programa Java sin la ejecución de alguna implantación de la JVM, por tanto, "… *el código no se ejecuta directamente sobre un procesador físico, sino sobre un procesador virtual Java*". Es la encargada de traducir los bytecode (Código resultante de la compilación del código fuente) en las instrucciones nativas, permitiendo la portabilidad de las aplicaciones*".* (13)

1.5.7 Lenguaje de Programación Groovy

Groovy es un lenguaje que "… utiliza *una sintaxis muy parecida a Java y comparte el mismo modelo de objetos, de hilos y de seguridad. Desde Groovy se puede acceder directamente a todas las Interfaces*

[8] De su traducción al español Máquina Virtual de Java.

de Programación de Aplicaciones existentes en Java. *La mayor parte del código escrito en Java es totalmente válido en Groovy* y esto permite *que sea un lenguaje fácil para programadores de Java; la curva de aprendizaje se reduce mucho en comparación con otros lenguajes que generan bytecode para la JVM, tales como Jythono y JRuby. Groovy puede usarse también de manera dinámica como un lenguaje de scripting".* (14)

Este lenguaje fue seleccionado principalmente porque está basado en el lenguaje Java en el cual se posee mayor experiencia que con otros lenguajes y además incorpora innovadoras características que permiten mayores facilidades a los desarrolladores como la incorporación de nuevos operadores que permiten la reducción de código.

1.5.8 Framework de desarrollo: Grails 1.3.7

En el desarrollo de software, un marco de trabajo (framework en inglés) es "… *una estructura de soporte definida sobre la cual un proyecto puede ser organizado y desarrollado*". (15)

"El futuro del desarrollo web está bien claro y se basa en el principio de reutilización de código para aumentar la productividad y disminuir los riesgos de desarrollo. Basado en estos principios surge Grails, un entorno para desarrollo de aplicaciones web sobre la plataforma Java Enterprise Edition". (15)

Desde el punto de vista del diseño, Grails se basa en dos principios fundamentales:

Convention Over Configuration (CoC)[9]

En lugar de tener que escribir interminables archivos de configuración en formato XML, Grails se basa *"… en una serie de convenciones para que el desarrollo de la aplicación sea mucho más rápido y productivo. Por ejemplo, todas las clases de la carpeta grails-app/controllers serán tratadas como Controladores, y se mapearán convenientemente a las urls de la aplicación".* (15)

DRY: Don't repeat yourself (DRY)[10]

[9] De su traducción al español Convención sobre Configuración.

[10] De su traducción al español No Te Repitas.

La participación de Spring Container en Grails permite "… *la inyección de dependencias mediante el patrón IoC*[11] *(Inversión of Control), de forma que cada actor en la aplicación deba definirse una única vez, haciéndose visible a todos los demás de forma automática"* (15). Esto permite ahorrar gran tiempo y evitar errores que puedan surgir a la hora de definir actores del sistema.

Grails está compuesto por un conjunto de elementos esenciales, a continuación se describen los más usados:

GORM (Grails Object Relational Mapping)

El modelo de datos en una aplicación Grails está compuesto por las clases del dominio. Grails utiliza GORM, un gestor de persistencia escrito en Groovy, para controlar el ciclo de vida de las entidades.

"GORM está construido sobre Hibernate, una herramienta de mapeo objeto-relacional, que se encarga de relacionar las clases del dominio con tablas de una base de datos, y las propiedades de las clases con campos en las tablas, además garantiza que cada operación que sea realizada sobre los objetos del modelo de datos será traducida por Hibernate a las sentencias SQL necesarias para quedar reflejado en la base de datos". (15)

GORM proporciona los métodos de búsqueda y modificación que permiten manipular las clases del dominio, además agrega un campo id a la clase y se encargará de generar un valor único para cada instancia de la clase logrando una mayor sencillez en la búsqueda de datos específicos.

1.5.9 Hyper Text Markup Languaje (HTML) [12]

Es un lenguaje de marcas orientado a la publicación de documentos en Internet. "… *HTML es un lenguaje extensible, al que se le pueden añadir nuevas características, marcas y funciones. Los documentos HTML están formados por una serie de bloques de texto con una entidad lógica (titulares, párrafos, listas, entre otros). La interpretación de estas entidades se deja al navegador, lo cual da una gran flexibilidad a la presentación del documento, que puede ser mostrado, por ejemplo, en terminales gráficos o de texto*". (16)

[11] De su traducción al español Inversión de Control.

[12] De su traducción al español Lenguaje de Marcado de Hipertexto.

Este lenguaje se utilizará como parte de la codificación de las páginas servidoras (GSPs) y como resultado del procesamiento de las mismas. Permite la creación de los componentes que contendrán las vistas para la interacción con el usuario.

1.5.10 Lenguaje JavaScript

Es un lenguaje de scripting basado en objetos, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.

JavaScript se caracteriza por ser un lenguaje *"…* basado *en prototipos, con entrada dinámica y con funciones de primera clase. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM. Su principal importancia radica en que tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML"* (17). Se empleará como lenguaje de codificación en el lado del cliente.

1.5.11 Framework de presentación: Dojo 1.5

Dojo es una biblioteca JavaScript de código abierto que "… r*esuelve problemas comunes y engorrosos en el desarrollo de aplicaciones con JavaScript como la disparidad del modelo de eventos de los distintos navegadores. Provee un conjunto de componentes de interfaz gráfica de usuario como calendarios y menús, que se pueden insertar de manera sencilla en páginas HTML*" (18). Estos componentes pueden ser utilizados en las interfaces del SIDEP logrando agilizar el desarrollo al reutilizar código existente con un alto grado de terminación.

Otros de los principales motivos que propiciaron el uso de Dojo es que cuenta con un mecanismo de creación de componentes lo suficientemente flexible para modificar los existentes y proporcionar la creación de nuevos componentes en correspondencia con las necesidades del proyecto.

1.5.12 Sistema Gestor de Base de Datos: Oracle 11g

Oracle es un Sistema Gestor de Bases de Datos con características objeto-relacionales que permite *"… la gestión de grandes bases de datos, un alto rendimiento en transacciones, disponibilidad controlada*

de los datos de las aplicaciones, la gestión de la seguridad y la autogestión de la integridad de los datos" (19). Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando: soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma.

1.5.13 Embarcadero ERStudio 8.0

Es una herramienta de modelado de datos para el diseño y construcción de bases de datos a nivel físico y lógico. "*Está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, construcción automática de bases de datos, documentación y fácil creación de reportes*". (20)

Esta herramienta será utilizada en la fase de diseño para la confección de los diagramas de entidad-relación y para normalizar las tablas de la Base de Datos.

1.5.14 Apache Tomcat (6.0.25)

Apache Tomcat es un contenedor web escrito en Java, por lo que funciona en cualquier sistema operativo que disponga de la Máquina Virtual de Java. *"Tomcat es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software Licence".* (21)

Será empleado como contenedor web, lo que permitirá mantener un constante acceso a la aplicación durante las fases de desarrollo, pruebas y despliegue.

1.6 Conclusiones parciales

Partiendo del análisis de las soluciones informáticas para la gestión de los procesos penitenciarios a nivel internacional y nacional, se arribó a la conclusión de que ninguno de los sistemas analizados ofrece las funcionalidades requeridas para llevar a cabo la gestión de solicitudes y decisiones en el sistema penitenciario cubano.

Durante este capítulo, se realizó una descripción de la metodología, herramientas y tecnologías a emplear en la elaboración de la solución, exponiendo los factores que intervinieron en la selección de RUP como metodología, UML como lenguaje de modelado y Visual Paradigm 5.0 como herramienta CASE.

También se fundamentan las razones por las cuales se seleccionó como parte del entorno de desarrollo: la Plataforma Java J2EE, la Máquina Virtual de Java, el lenguaje de programación Java, el lenguaje de programación Groovy, frameworks como Dojo 1.5 y Grails 1.3.7, SpringSource Tool Suit 2.5 como IDE, Oracle 11g como gestor de base de datos y Apache Tomcat 6.0 como Contenedor Web.

Diseño del sistema

2.1 Introducción

En este capítulo se explica cómo está estructurada la arquitectura de la aplicación que se implementa y los patrones de diseño que se utilizan. Además, se describen las funcionalidades por las cuales están compuestos los módulos Solicitudes y Decisiones. Se expone una representación gráfica de los diagramas de clases e interacción, así como el diseño de la base de datos para la persistencia de la información.

La presente investigación parte del levantamiento del negocio realizado por integrantes del proyecto Prisiones Cuba y los requisitos acordados con el cliente. A partir de los cuales se realiza el diseño de los módulos para su posterior implementación y la realización de pruebas.

Para una mejor comprensión se reflejan las funcionalidades de los módulos Solicitudes y Decisiones del SIDEP.

2.2 Descripción de las funcionalidades del módulo Solicitudes

Nombre	Descripción
CRUD-C (consultar, actualizar y eliminar) solicitud de ampliación o reducción de capacidades.	El caso de uso permite al oficial de aseguramiento multilateral consultar, actualizar o eliminar una solicitud de ampliación o reducción de capacidades.
Evaluar solicitud de ampliación o reducción de las capacidades.	El caso de uso permite al jefe de la DEP evaluar una solicitud de ampliación o reducción de las capacidades.
Registrar solicitud de ampliación o reducción de capacidades.	El caso de uso permite al oficial de aseguramiento multilateral registrar una solicitud de ampliación o reducción de capacidades.
CRUD-C solicitud de conducción.	El caso de uso permite al solicitante de conducción (educador guía, oficial de atención a la ciudadanía) consultar, actualizar o eliminar

	una solicitud de conducción.
Evaluar solicitud de conducción por el jefe de centro.	El caso de uso permite al jefe de centro evaluar una solicitud de conducción.
Evaluar solicitud de conducción a internos de mayor severidad o categoría especial.	El caso de uso permite al jefe de la DEP evaluar una solicitud de conducción a internos de mayor severidad o categoría especial.
Evaluar solicitud de conducción por el jefe de tratamiento penitenciario.	El caso de uso permite al jefe de tratamiento penitenciario evaluar una solicitud de conducción.
Registrar solicitud de conducción.	El caso de uso permite al solicitante de conducción (educador guía, oficial de atención a la ciudadanía) registrar una solicitud de conducción.
CRUD-C solicitud de traslado interpenal.	El caso de uso permite al jefe de centro consultar, actualizar o eliminar una solicitud de traslado interpenal.
Evaluar solicitud de traslado interpenal.	El caso de uso permite al jefe del órgano provincial de establecimientos penitenciarios y el MEIJ evaluar una solicitud de traslado interpenal.
Registrar solicitud de traslado interpenal.	El caso de uso permite al jefe de centro registrar una solicitud de traslado interpenal.
CRUD-C solicitud de traslado interprovincial.	El caso de uso permite al jefe de centro consultar, actualizar o eliminar una solicitud de traslado interprovincial.
Evaluar solicitud de traslado interprovincial.	El caso de uso permite al jefe de registro legal de la DEP evaluar una solicitud de traslado

	interprovincial.
Registrar solicitud de traslado interprovincial.	El caso de uso permite al jefe de centro registrar una solicitud de traslado interprovincial.
Evaluar solicitud de traslado para internos de mayor severidad.	El caso de uso permite al jefe de registro legal de la DEP evaluar una solicitud de traslado para internos de mayor severidad.

2.3 Descripción de las funcionalidades del módulo Decisiones

Nombre	**Descripción**
Consultar decisión de ampliación o reducción de las capacidades.	El caso de uso permite al oficial de aseguramiento multilateral y al jefe de la DEP consultar decisiones de ampliación o reducción de las capacidades.
Consultar decisión de conducción.	El caso de uso permite al solicitante de conducción (educador guía, oficial de atención a la ciudadanía) y al jefe de tratamiento penitenciario consultar decisiones de conducción.
Crear decisión de suspensión de permisos.	El caso de uso permite al jefe de la DEP crear una decisión de suspensión de permisos.
Consultar decisión de suspensión de permisos.	El caso de uso permite al jefe de la DEP, al jefe del centro y al jefe de tratamiento educativo consultar las decisiones de suspensión de permisos.
Crear decisión de cantidad de internos para PESH.	El caso de uso permite al jefe de unidad crear una decisión de cantidad de internos para PESH.
Consultar decisión de cantidad de internos para PESH.	El caso de uso permite al jefe de unidad y al oficial de sistema educativo consultar las

	decisiones de cantidad de internos para PESH.
Crear decisión de otorgamiento de licencia extrapenal.	El caso de uso permite al jefe de la DEP crear una decisión de otorgamiento de licencia extrapenal.
Consultar decisión de otorgamiento de licencia extrapenal.	El caso de uso permite al jefe de la DEP, al jefe de registro legal y al jefe del centro consultar las decisiones de otorgamiento de licencia extrapenal.
Crear decisión de conducción por circular 115.	El caso de uso permite al jefe de registro legal de la DEP crear una decisión de conducción por circular 115.
Consultar decisión de conducción por circular 115.	El caso de uso permite al jefe de registro legal, al jefe del centro y al jefe de registro legal consultar las decisiones de conducción por circular 115.
Crear decisión de traslado interpenal.	El caso de uso permite al jefe de registro legal de la DEP crear una decisión de traslado interpenal.
Consultar decisión de traslado interpenal.	El caso de uso permite al jefe de registro legal de la DEP, al jefe del centro penitenciario y al jefe de registro legal consultar las decisiones de traslado interpenal.
Crear decisión de traslado interprovincial.	El caso de uso permite al jefe de registro legal de la DEP crear una decisión de traslado interprovincial.
Consultar decisión de traslado interprovincial.	El caso de uso permite al jefe de registro legal de la DEP, al jefe del centro penitenciario y al jefe de registro legal consultar las decisiones de traslado interprovincial.

Crear decisión de prohibición de cambio de ubicación.	El caso de uso permite al jefe del centro penitenciario, o al jefe del órgano provincial de establecimientos penitenciario crear una decisión de prohibición de cambio de ubicación.
Consultar decisión de prohibición de cambio de ubicación.	El caso de uso permite al jefe del centro penitenciario, al jefe del órgano provincial de establecimientos penitenciarios, al jefe de tratamiento penitenciario, al jefe de orden interior y al jefe de registro legal consultar las decisiones de prohibición de cambio de ubicación.

Cada una de las funcionalidades listadas anteriormente serán implementadas con el objetivo de lograr una gestión de las solicitudes y decisiones en el sistema penitenciario cubano. El trabajador que esté autenticado en el sistema, según los permisos que tenga asignados podrá crear, eliminar, actualizar o evaluar una solicitud, así como crear una decisión. También podrán ser consultados los datos de las solicitudes y decisiones con los casos de uso Consultar.

El desarrollo de los módulos Solicitudes y Decisiones como parte del SIDEP, por sus propias características requieren un alto nivel de disponibilidad y por tanto fácil acceso por parte de los usuarios del sistema. Además en respuesta a una de las solicitudes planteadas por el cliente, surge la necesidad del desarrollo del sistema bajo un marco de trabajo flexible que pueda ser accedido desde múltiples puntos de acceso así como facilidades de interacción con el usuario.

2.4 Marco de trabajo de desarrollo web

Grails es un marco de trabajo que se utiliza para el desarrollo de aplicaciones web sobre la plataforma Java Enterprise Edition. Utiliza el estilo arquitectónico Llamada y Retorno y dentro de este el patrón Modelo-Vista-Controlador (MVC), el cual propone una organización en tres capas, las cuales están conformadas por el modelo o capa de datos, la vista o capa de presentación, el controlador o capa de control. Grails, define un cuarto componente que son los servicios los cuales se añaden al modelo.

La arquitectura del sistema está basada en el marco de trabajo Grails, que permite vincular de forma adaptable y escalable los componentes para la capa de presentación. Además posee componentes y librerías que permiten y facilitan la interacción entre capas.

2.4.1 Arquitectura n-capas del sistema

Los módulos Solicitudes y Decisiones como componente del SIDEP son desarrollados sobre la Máquina Virtual de Java, siendo esta capaz de interpretar el lenguaje Groovy que se utiliza para la implementación del sistema. Lo que permite aprovechar toda la robustez de Java, y la simpleza de los principios DRY y CoC planteadas por el marco de trabajo.

Grails utiliza Spring para la inyección de dependencias y la seguridad de los datos. GORM (Grails Object Relational Mapping) es la abstracción de mapeo objeto-relacional. Es muy simple de usar y se emplea mayormente para la realización de consultas a la base de datos. La librería Dojo es utilizada en la elaboración de la interfaz visual.

La figura que se muestra a continuación representa la arquitectura n-capas del sistema donde se aplica el patrón de diseño Modelo Vista Controlador (MVC). Además en la capa Modelo se contemplan los servicios que se utilizan en las aplicaciones Grails.

Figura 2: Arquitectura del Sistema.

2.5 Patrones de diseño

Un patrón es *"… un conjunto de información que proporciona una respuesta a un conjunto de problemas similares, es decir, un patrón es una solución a un problema en un contexto".* (22)

2.5.1 Modelo – Vista – Controlador (MVC)

Modelo – Vista – Controlador (MVC) es un patrón que *"… separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos"* (22). A continuación se describe la utilización de este patrón en el diseño de la arquitectura del SIDEP:

Vista

En esta capa se encuentran las Vistas, según la arquitectura que propone Grails estarán enmarcadas sus clases en el paquete View donde se encontrarán las Groovy Server Pages[13] (GSP). Grails implementa el patrón MVC, en el que la lógica del negocio se separa de la presentación de la aplicación. Esto permite cambiar fácilmente el aspecto de la aplicación, sin modificar su comportamiento.

"Grails utiliza para la interacción con el usuario la tecnología Java Server Pages[14] (JSP), pero basada en una implementación mediante GSP. Además los desarrolladores pueden mezclar etiquetas de lenguajes de marcas tradicionales como HTML con código Groovy para producir vistas dinámicas. Las Vistas son los recursos que junto al modelo generado por los controladores le permiten al cliente visualizar la información, estos pueden ser páginas HTML, documentos en formato PDF, hojas de cálculo, entre otras. Las mismas están representadas en el paquete de clase Views". (23)

La capa de presentación agrupa los siguientes componentes:

- Clases de presentación (gsp)
- Taglibs
- Java Script
- Cascading Style Sheets (CSS)[15]

[13] De su traducción al español Páginas Servidoras Groovy.

[14] De su traducción al español Páginas Servidoras Java.

[15] De su traducción al español Hojas de Estilo en Cascada.

En la figura que se muestra a continuación se señalan los paquetes que contienen los componentes de la capa de presentación:

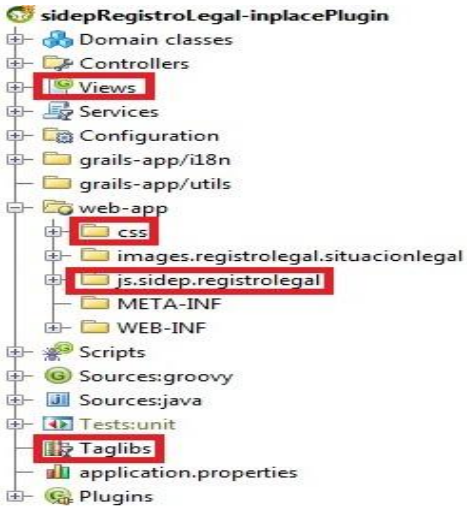

Figura 3: Componentes que intervienen en la Capa de Presentación.

Controlador

En esta capa se encuentra la Lógica de Presentación, en las cuales según la arquitectura que propone Grails estarán enmarcadas las clases controladoras en los paquetes Controllers. *"Un controlador de Grails es una clase responsable por el manejo de los pedidos provenientes de la aplicación. El controlador recibe la petición, realiza un trabajo potencial con la misma y finalmente decide que sucederá a continuación, lo cual puede incluir algunos de los siguientes flujos"*. (23)

- *"Ejecutar otra función de controlador".* (23)
- *"Mostrar una vista".* (23)
- *"Mostrar información directamente con la respuesta de la petición".* (23)

En la figura que se muestra a continuación se señala el paquete que contiene las clases controladoras:

Figura 4: Paquete de Controladores.

Modelo

En esta capa se agrupan las clases de dominio y los servicios, los cuales son descritos a continuación:

Dominio: Son las clases encargadas de *"… persistir la información de la aplicación haciendo uso de Hibernate. Estas clases son utilizadas por los servicios para hacer las consultas y obtener la información que el usuario solicita o el sistema necesita. Las clases del dominio son entidades fundamentales para el sistema, ya que en ellas persisten los datos con los que posteriormente el cliente va a interactuar, modificar o insertar".* (23)

Grails utiliza un gestor de persistencia escrito en Groovy sobre Hibernate conocido por GORM para controlar el ciclo de vida de las entidades.

"Cuando se definen las propiedades de las clases del dominio, GORM se encarga de generar las tablas correspondientes en la base de datos con los campos necesarios para almacenar cada propiedad y proporcionar los métodos de búsqueda y modificación que permitan manipular la entidad. Añade automáticamente un campo id a la clase, genera un valor único secuencial para cada instancia de la misma y restringe los valores que pueden asignarse a los atributos de cada entidad mediante la propiedad constraints". (23)

Servicios: Son componentes que implementan la lógica de negocio de la aplicación. *"Todas las entidades de una aplicación Grails que pertenecen al modelo de datos poseen métodos de instancia que facilitan su manipulación y pueden ser llamados desde los servicios. La utilización del marco de*

trabajo permite el manejo de métodos estáticos para hacer consultas o bien insertar o eliminar datos en la base de datos". (23)

En la figura que se muestra a continuación se señalan los paquetes que contienen las clases de dominio y los servicios:

Figura 5: Paquetes de Dominio y Servicio.

2.5.2 Experto

El patrón experto es el encargado de *"… asignar una responsabilidad al más competente en información, donde dicha clase cuenta con la información necesaria para cumplir la responsabilidad. Es el principio básico de asignación de responsabilidades que suele utilizarse en el diseño Orientado a Objetos. Es el patrón más usado para asignar responsabilidades".* (24)

- En el siguiente fragmento de código que pertenece al controlador EvaluarSolController se evidencia la utilización del patrón experto. Esta clase se especializa en evaluar todas las solicitudes que se creen en la aplicación para convertirse luego en decisiones.

```
class EvaluarSolController {
    def decisionesService
    def urlbase = "/registrolegal/gestion/solicitudes"
    def index = {
        render view: "${urlbase}/principal"
    }
    def evaluar = {
      Solicitud solicitud = Solicitud.findById(params.solicitud.toLong())
      Decision decision = new Decision(params.decision)
      decision.fechaDecision = new Date()
      decision.visible = true
      decision.tipoDecision = decisionesService.tipoDecision(solicitud)
      decision.de = decisionesService.trabajadorDe()
      decision.addToPara(solicitud.de)
      if (decision.validate()) {
       if (decision.save(flush: true)) {
         if (new DecSoli(decision: decision, solicitud: solicitud).save(flush: true)) {
            redirect(controller: 'gestion', action: 'index')
         } else {
            println("no guardo la relación entre solicitud y decisión")
          }
         }  else {
            println("no salvo la decisión")
         }
       } else {
       decision.errors.each {
       println it
    }
```

Figura 6: Patrón experto utilizado en el controlador EvaluarSolController.

2.5.3 Controlador

El patrón controlador es el encargado de *"… asignar la responsabilidad de recibir o manejar un mensaje de evento del sistema. Dicho controlador utiliza el sistema de enrutamiento para asociar el nombre de una acción con la URL solicitada por el usuario".* (24)

- En el siguiente fragmento de código que pertenece al controlador GestionController se evidencia la utilización del patrón controlador. Esta clase se encarga de redireccionar las peticiones que se reciban hacia el controlador correspondiente.

```
class GestionController {

    def decisionesService
    def solicitudesService

    def index = {
        def dec = decisionesService.mostrarDecisiones()
        def sol = solicitudesService.mostrarSolicitudes()
        solicitudesService.limpiarTablaInternos()
        render view: "/registrolegal/gestion/principal", model: [decisiones: dec, solicitudes: sol]
    }

    def solicitudesEntrantes = {
        def sol = solicitudesService.listaSolEntrantes()
        render view: "/registrolegal/gestion/solicitudes/solEntrantes", model: [solicitudes: sol]
    }
```

Figura 7: Patrón controlador utilizado en el controlador GestionController.

2.5.4 Creador

El patrón creador plantea *"… asignarle a la clase B la responsabilidad de crear una instancia de la clase A. Grails en cada una de las clases controladoras o servicios que crea para cada uno de los módulos o funcionalidades de la aplicación, se encarga de la creación de las instancias de las clases que describen los objetos que en ellos se manejan, ya que estos agregan, contienen, registran o utilizan las instancias de estos objetos".* (24)

En el siguiente fragmento de código que pertenece al controlador SolTrasladoController se evidencia la utilización del patrón creador.

```
class SolTrasladoController {
    def decisionesService
    def urlbase = "/registrolegal/gestion/solTraslado"

    def index = {
        render view: "${urlbase}/principal"
    }
    ...
      decision.tipoDecision = decisionesService.tipoDecision(solicitud)
      decision.de = decisionesService.trabajadorDe()
    ...

}
```

Figura 8: Patrón creador utilizado en el controlador SolTraslado.

2.6 Diagrama de Clases

El diagrama de clases del diseño es *"… una representación concreta donde se representa la parte estática del sistema. En él se representa la vista con su formulario y elementos Java Script, la clase controladora que la relaciona con los servicios necesarios y las entidades que se utilizan para el caso de uso diseñado".* (25)

2.6.1 Diagrama de clases del diseño

A continuación se explica a través de las siguientes tablas el flujo del Caso de Uso Crear Decisión de Cantidad de Internos con PESH. Todos los diagramas de clases del diseño pueden ser consultados en el Anexo 2.

Para crear este tipo de decisión el usuario accederá a la página principal donde seleccionará la decisión que se desea crear, en este caso se selecciona la decisión de Cantidad de Internos con

PESH, se ejecuta un link al controlador **DecisioneController** y el mismo redirecciona para el controlador **DecCantIntPESHController**, este se comunica con **decisionCantidadInternosPESH.gsp** y este construye la página **decisionCantidadInternosPESH.html**.

Esta última tiene una relación de composición con el formulario **decisionCantidadInternosPESHForm** y con la página **DecisionCantidadInternosPESH.js**. Desde la página **decisionCantidadInternosPESH.html** el usuario aceptará la decisión que está creando, se validarán los datos desde el JavaScript y se ejecutará una acción submit desde él, donde se enviarán los datos contenidos dentro del formulario hacia el controlador **DecCantIntPESHController.** Este controlador se comunicará con la clase entidad **DecisionesService** para validar los datos en el servidor y salvar la nueva decisión. Luego de que la nueva decisión esté guardada el controlador **DecCantIntPESHController** se comunica con **Principal.gsp** y esta construye la página **Principal.html** donde culmina el caso de uso.

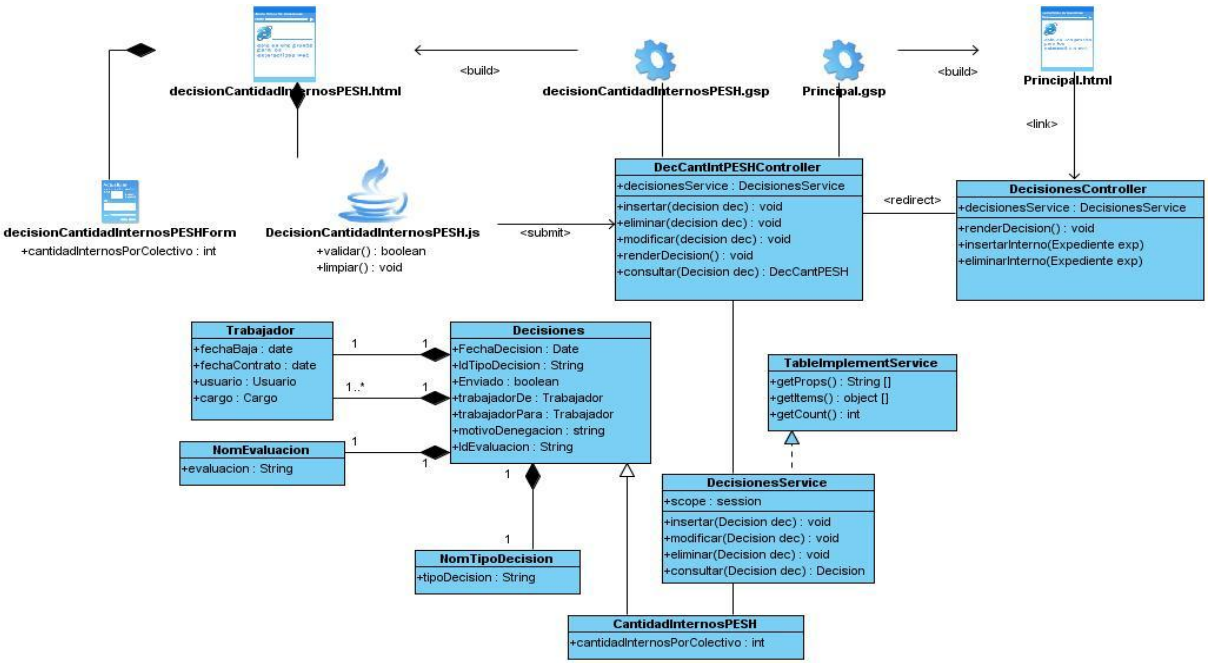

Figura 9: Diagrama de clases del diseño - CU Decisión de cantidad de internos con PESH.

2.6.2 Descripción de las clases

A continuación se describen las clases que intervienen en el caso de Caso de Uso Crear Decisión de Cantidad de Internos con PESH en el cual se pueden ver las principales entidades que son necesarias para la gestión de cualquier decisión.

En el Anexo 1 se puede consultar la descripción del resto de las clases que intervienen en la gestión de las solicitudes y las decisiones.

Nombre de la clase: Principal.gsp
Descripción: Es la página desde donde se accede a todas las solicitudes y decisiones del sistema, en ella se muestran todas las solicitudes o decisiones creadas, enviadas o recibidas.

Tabla 1: Página Principal

Nombre de la clase: DecisionCantidadInternosPESH.gsp		
Descripción: Es la página principal del caso de uso, cuenta con un formulario con los atributos principales de la decisión y un archivo con extensión JavaScript para la validación de los errores en el momento de introducir los datos.		
Atributos	**Tipo**	**Descripción**
cantidadInternosPorColectivo	int	Este campo guarda la cantidad de internos que deberá existir por colectivo.

Tabla 2: Página Decisión de Cantidad de Internos con PESH

Nombre de la clase: DecisionesController.groovy		
Descripción: Es una clase controladora donde se gestionan los tipos de decisiones que podrán tomar los usuarios autenticados en la aplicación y que se encarga de redireccionar hacia el controlador encargado de guardar la decisión que se vaya a crear.		
Atributos	**Tipo**	**Descripción**
decisionesService	DecisionesService	Es un objeto de tipo clase servicio DecisionesService, para llamar los métodos de la

		misma en el controlador.
Acciones	**Tipo**	**Descripción**
render	Spring Model and View	Es una acción que permite mostrar una nueva vista.
insertar(Decisiones dec)	Objeto	Es una acción que permite guardar los datos que se introdujeron en el formulario para la decisión, haciendo uso del método insertar (Decisiones dec) definido en la clase servicio DecisionesService.
modificar(Decisiones dec)	Objeto	Es una acción que permite modificar los datos que se introdujeron en el formulario para la decisión, haciendo uso del método modificar (Decisiones dec) definido en la clase servicio DecisionesService.
consultar(Decisiones dec)	DecCantIntPESH	Es una acción que permite que se puedan consultar las decisiones que hayan sido enviadas.
eliminar(Decisiones dec)	void	Es una acción que permite eliminar una decisión guardada, haciendo uso del método eliminar (Decisiones dec) definido en la clase servicio DecisionesService.

Tabla 3: Clase controladora Decisiones Controller

Nombre de la clase: DecisionesService.groovy		
Descripción: Es una clase de servicio en la cual se implementa la lógica de negocio de todas las decisiones y donde se implementan los métodos que se utilizan en cada controlador encargado de gestionar una decisión.		
Atributos	**Tipo**	**Descripción**
mostrarInternosService	MostrarInternosService	Es un objeto de tipo clase servicio MostrarInternosService, para llamar los métodos

		de la misma en el controlador.
Acciones	**Tipo**	**Descripción**
eliminar (decisiones)	void	Elimina una decisión de la tabla.
modificar(decisiones)	Objeto	Modifica la decisión que se pasa por parámetro.
insertar(decisiones)	Objeto	Guarda la decisión que se pasa por parámetro.
consultar(decisiones)	DecCantIntPESH	Es una acción que permite que se puedan consultar las decisiones que hayan sido enviadas.

Tabla 4: Clase de Servicio Decisiones Service

Nombre de la clase: TableImplementService.groovy		
Descripción: Es una clase, en la cual se implementa la lógica de negocio correspondiente a las tablas que se utilizan en la aplicación. Estas tablas son dinámicas pues se pueden listar elementos de cualquier tipo pasándole solamente el servicio desde el cual deben listar.		
Acciones	**Tipo**	**Descripción**
getProps()	List	Devuelve una lista con el nombre de las columnas de la tabla.
getItems()	List	Devuelve una lista con todos los elementos de la tabla.
getCount()	int	Devuelve el total de elementos mostrados en la tabla.

Tabla 5: Clase de servicio Table Implement Service

Nombre de la clase: Decisiones.groovy
Descripción: Es una clase de dominio que contiene los atributos de una decisión y de ella heredan todas las demás decisiones. En la misma se restringen los valores que pueden tomar los atributos.

Atributos	Tipo	Descripción
fechaDecision	date	Fecha en que se realizó la decisión.
idTipoDecision	String	Tipo de decisión.
Enviado	boolean	Un atributo para saber si la decisión fue enviada.
trabajadorDe	Trabajador	Trabajador que emite la decisión.
trabajadorPara	Trabajador	Trabajador que recibirá la decisión.
idEvaluacion	String	Evaluación de la decisión
motivoDenegación	String	En caso de que la decisión se deniegue se debe llenar este campo con el motivo.

Tabla 6: Clase entidad Decisiones

Nombre de la clase: CantidadInternosPESH.groovy		
Descripción: Es una clase de dominio que contiene los atributos de una decisión de cantidad de internos con PESH. En la misma se restringe el valor que puede tomar su único atributo.		
Atributos	**Tipo**	**Descripción**
cantidadInternosPorColectivo	int	Atributo que representa la cantidad de internos que puede tener un colectivo.

Tabla 7: Clase entidad Cantidad de Internos con PESH

Nombre de la clase: CantidadInternosPESH.js
Descripción: Es una clase JavaScript en la que se valida el único atributo que tiene la clase CantidadInternosPESH.gsp y se limpian los componentes de esta última si se cancela la creación de

la decisión.		
Acciones	**Tipo**	**Descripción**
limpiar	void	Acción que limpia todos los campos del formulario después de enviarse la decisión.
validar	boolean	Acción que valida los datos del formulario antes de enviarse a la clase controladora.

Tabla 8: Clase JavaScript Cantidad de Internos con PESH.js

Nombre de la clase: NomEvaluacion.groovy		
Descripción: Es una clase Nomenclador que se encarga de listar los tipos de evaluaciones que existen que en este caso es aprobado o denegado.		
Atributos	**Tipo**	**Descripción**
evaluacion	String	Atributo que representa la evaluación de una decisión.

Tabla 9: Clase Nomenclador NomEvaluacion

Nombre de la clase: NomTipoDecision.groovy		
Descripción: Es una clase Nomenclador que se encarga de listar los tipos de decisiones que existen.		
Atributos	**Tipo**	**Descripción**
tipoDecision	String	Atributo que representa el tipo de decisión.

Tabla 10: Clase Nomenclador NomTipoDecision

Nombre de la clase: Trabajador.groovy		
Descripción: Es una clase de dominio que contiene los atributos de un trabajador. En la misma se restringen los valores que pueden tomar los atributos.		
Atributos	**Tipo**	**Descripción**

fechaBaja	Date	Atributo que representa la fecha de baja.
fechaContrato	Date	Atributo que representa la fecha de contrato.
cargo	Cargo	Atributo que representa el cargo de una persona.
usuario	Usuario	Atributo que representa el usuario de una persona.

Tabla 11: Clase entidad Trabajador

2.7 Diagrama de colaboración

A continuación se explica a través del siguiente diagrama de colaboración el flujo del Caso de Uso Crear Decisión de Cantidad de Internos PESH. Todos los diagramas de colaboración pueden ser consultados en el Anexo 3.

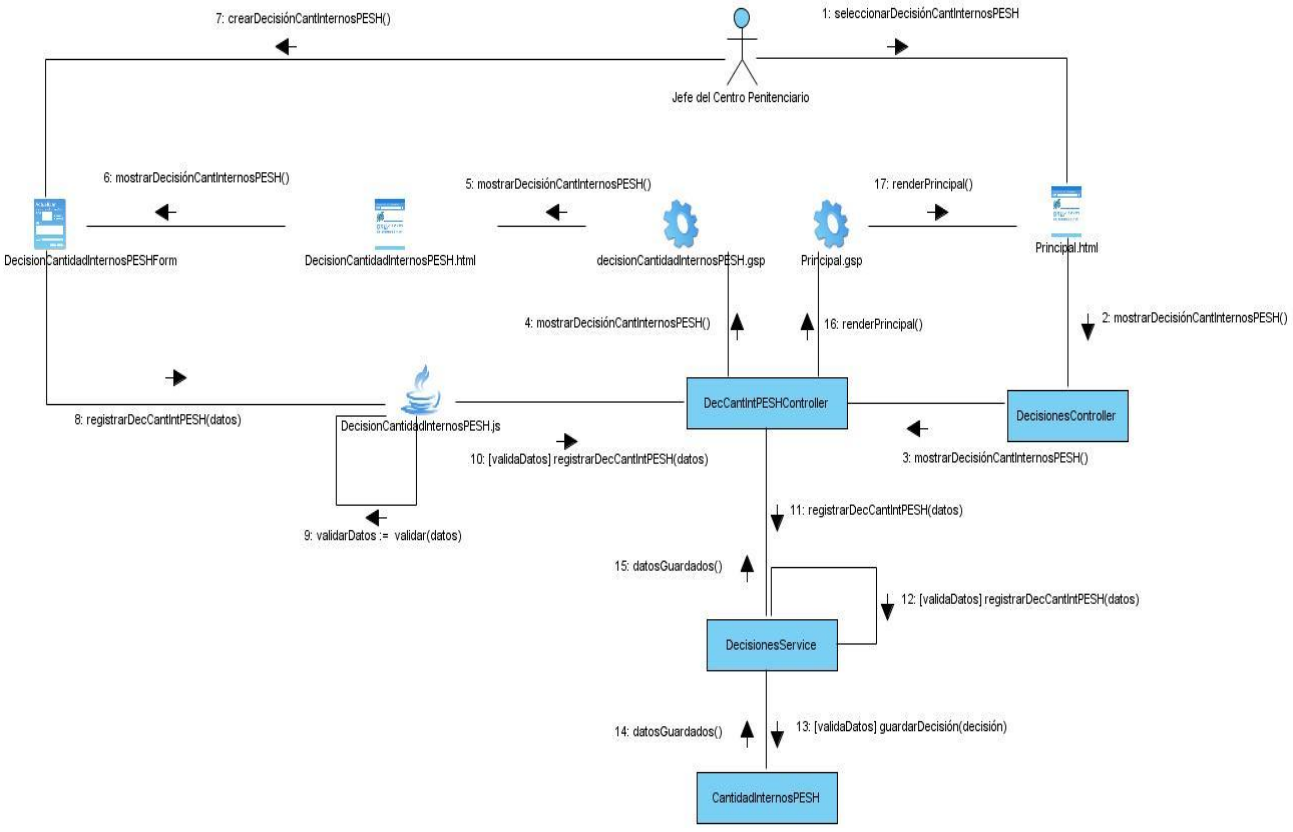

Figura 10: Diagrama de colaboración - CU Crear decisión de cantidad de internos para PESH.

2.8 Diseño de la Base de datos

Las Bases de datos necesitan una definición de su estructura que le permitan almacenar, reconocer el contenido y recuperar datos. El artefacto resultante de esta actividad será el Modelo de Datos, que describe la representación lógica y física de los datos persistentes.

Con el objetivo de lograr un mayor entendimiento y visibilidad sobre el modelo de datos se muestra a continuación una imagen que contiene solamente un fragmento del diseño de la Base de Datos. Este fragmento se refiere a las clases que intervienen en el Caso de Uso Gestionar Decisión de Cantidad de Internos con PESH, donde se representan las entidades necesarias para crear este tipo de decisión. De la clase entidad **Decisión** heredan todas las decisiones en el sistema, tiene una relación de uno a uno con la clase entidad **Trabajador** y otra de uno a muchos con esa misma clase **Trabajador**, presenta una relación de uno a muchos con los nomencladores **NOM_TIPO_DECISION** y **NOM_EVALUACION**, posee una relación de uno a uno con la clase entidad **Decision_Solicitud** la cual construye un id compuesto a partir de una solicitud que se genere y la decisión resultado de la evaluación de esa solicitud. En el Anexo 4 se ilustra el diseño del Modelo de la Base de Datos construida para las solicitudes y las decisiones.

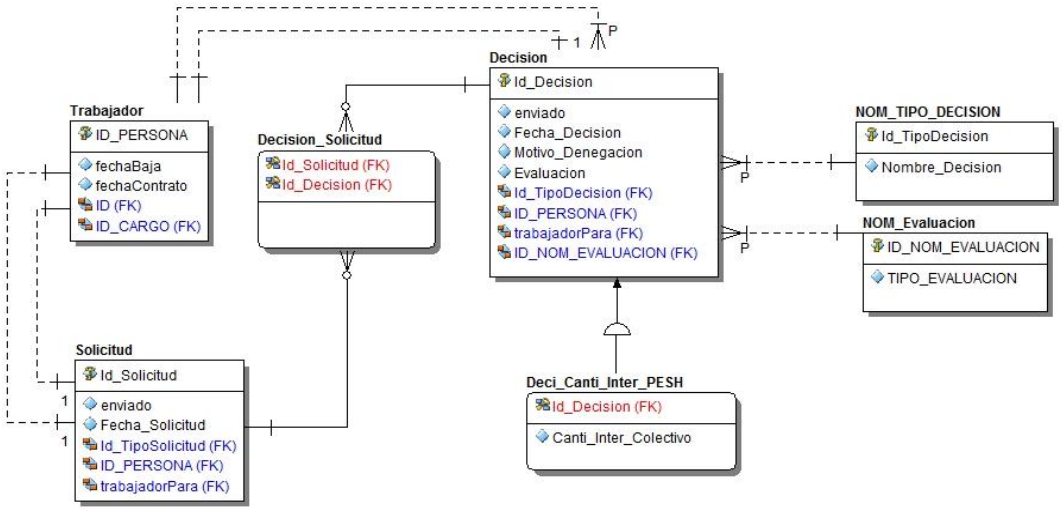

Figura 11: Diseño de la Base de Datos para el Caso de Uso Gestionar Decisión de Cantidad de Internos con PESH.

2.9 Entidades más importantes presentes en el diseño de la base datos de los módulos solicitudes y decisiones.

Para crear cualquier tipo de solicitud y decisión en el sistema se utilizan las tablas que se representan a continuación. Todas las operaciones que se realicen con las solicitudes y las decisiones dependen de estas tablas ya sea crear, actualizar, eliminar o consultar una solicitud o decisión en específico. Sin estas clases no se podría hablar de solicitudes y decisiones ya que de ellas heredan todas las demás.

Solicitudes			
Descripción: Es la clase solicitud que contiene los atributos que serán heredados por todas las demás solicitudes que existen en el sistema.			
Atributos	**Tipo**	**Descripción**	**Puede ser nulo**
enviado	BIT	Este campo guarda si la solicitud fue enviada.	No
fechaSolicitud	DATE	Este campo guarda la fecha de creación de la solicitud.	No
tipoSolicitud	CHAR(10)	Este campo guarda el tipo de solicitud.	No
trabajadorDe	NUMERIC(19.0)	Este campo guarda el trabajador que crea la solicitud.	No
trabajadorPara	NUMERIC(19.0)	Este campo guarda el trabajador que recibirá la solicitud.	No
IdSolicitud	VARCHAR	Este campo guarda el id que recibirá la solicitud.	No

Tabla 12: Solicitudes

Decisión
Descripción: Es la clase decisión que contiene los atributos que serán heredados por todas las demás decisiones que existen en el sistema.

Atributos	Tipo	Descripción	Puede ser nulo
enviado	BIT	Este campo guarda si la solicitud fue enviada.	No
fechaDecision	DATE	Este campo guarda la fecha de creación de la decisión.	No
motivoDenegacion	VARCHAR(10)	Este campo guarda el motivo por el cual se deniega una decisión.	Si
evaluación	NUMERIC(10.0)	Este campo guarda la evaluación de una decisión.	No
tipoDecision	CHAR(10)	Este campo guarda el tipo de decisión.	No
trabajadorDe	NUMERIC(19.0)	Este campo guarda el trabajador que crea la decisión.	No
trabajadorPara	NUMERIC(19.0)	Este campo guarda el trabajador que recibirá la decisión.	No
IdDecision	VARCHAR	Este campo guarda el id que recibirá la decisión.	No

Tabla 13: Decisión

2.10 Conclusiones Parciales

En el presente capítulo se generaron los siguientes artefactos para los módulos Solicitudes y Decisiones:

- Diagramas de clases del diseño.
- Diagramas de secuencia.

- Modelo de Datos.

También se hizo una síntesis del propósito de este capítulo, se describieron las funcionalidades de los módulos Solicitudes y Decisiones, se describió el marco de desarrollo web con el que se trabaja y los patrones de diseño utilizados en la aplicación.

Implementación y Pruebas

3.1 Introducción

A partir de los resultados del diseño, en este capítulo se abordará todo lo referente a la implementación de los módulos Solicitudes y Decisiones, lo cual denota el estado actual del sistema en términos de componentes y subsistemas de implementación. Además, se definen las pruebas del software como elemento crítico para la garantía de la calidad del sistema y revisión final del cumplimiento de las especificaciones de diseño y codificación.

3.2 Implementación

En esta disciplina los elementos del diseño, fundamentalmente las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, librerías, entre otros. También se describe como dependen los componentes unos de otros. Para la implementación de los Módulos Solicitudes y Decisiones se realizan los diagramas de despliegue y de componentes.

3.3 Diagrama de Componentes

Los diagramas de componentes *"… describen los elementos físicos del sistema y las relaciones entre ellos. Permiten observar con más facilidad la estructura general de un sistema de software y el comportamiento de los servicios que estos componentes proporcionan. Además, permiten especificar componentes con interfaces bien definidas y en él se muestran los elementos de diseño del sistema. Proveen una vista arquitectónica de alto nivel y ayudan a los desarrolladores a visualizar el camino de la implementación, incluso a tomar decisiones sobre ella".* (25)

A continuación se explica a través del siguiente diagrama de componentes la relación que existe entre los módulos Solicitudes y Decisiones con los demás componentes del sistema.

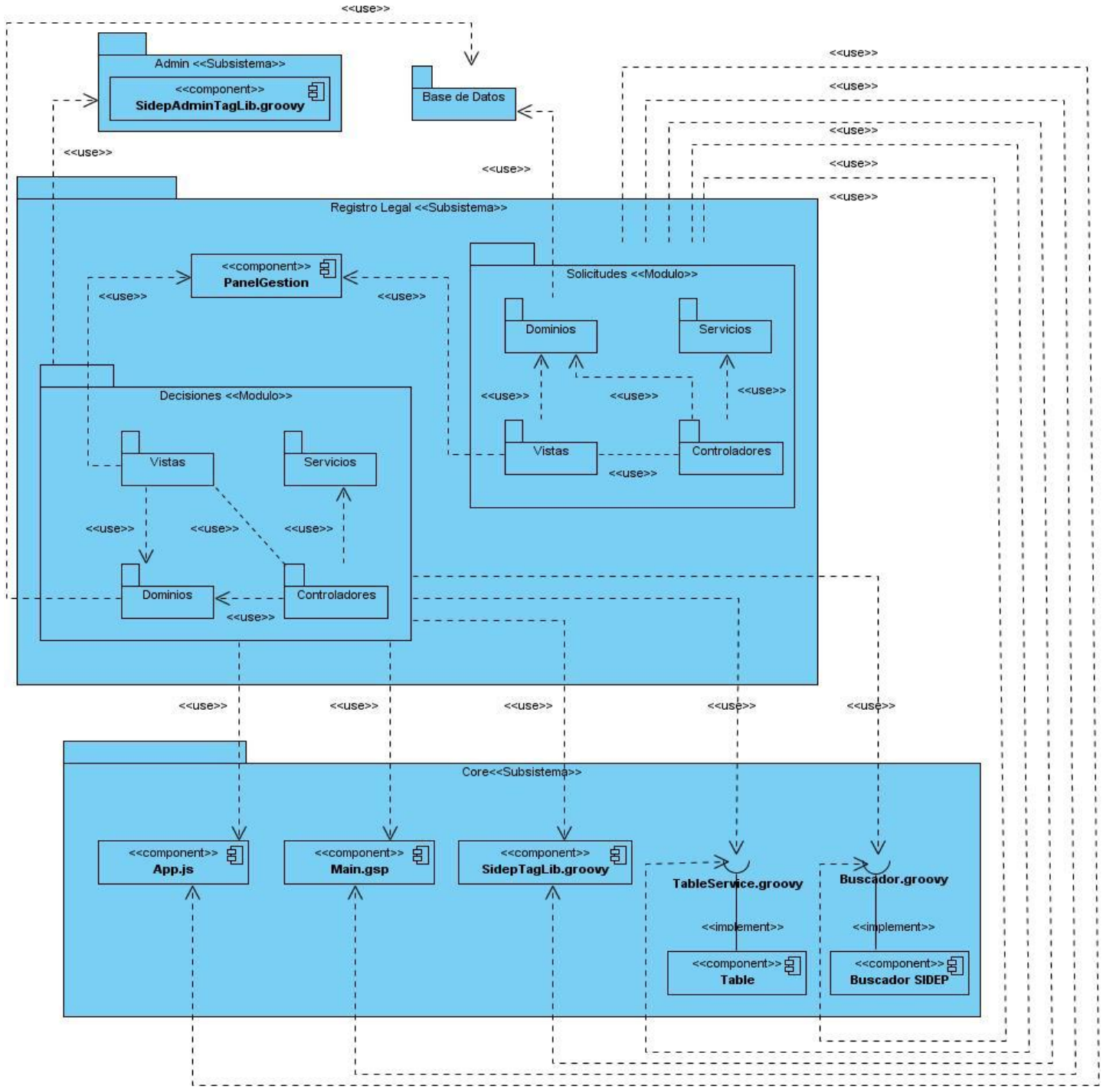

Figura 12: Diagrama de componentes general.

El diagrama de la figura muestra los módulos Solicitudes y Decisiones pertenecientes al Subsistema Registro Legal. Internamente, los módulos Solicitudes y Decisiones están representados por cuatro paquetes de componentes que agrupan a las clases servicios, controladores, vistas, dominios y la relación que existe entre ellas.

El paquete de componentes **Vistas** de ambos módulos interactúa internamente en el subsistema Registro Legal con el componente **PanelGestion** donde se realizan todas las operaciones relacionadas con las solicitudes o las decisiones. El paquete de componentes **Dominios** de ambos módulos interactúa con el paquete de componentes **Base de Datos** que como su nombre lo indica representa la base de datos y los componentes que agrupa lo constituyen las tablas. Ambos módulos utilizan el componente **SidepAdminTagLib.groovy** que se encarga de la seguridad del sistema y se encuentra dentro del subsistema **Admin**.

El subsistema **Core** contiene la mayor cantidad de componentes utilizados por los módulos Solicitudes y Decisiones, entre ellos se encuentra el componente **App.js.** Este componente es un archivo con funcionalidades comunes para todos los módulos del SIDEP y se apoya en las facilidades brindadas por Dojo para la validación y definición del comportamiento de los componentes que integran las vistas.

El componente **Main.gsp** se encarga de referenciar los archivos que sirven como plantilla para el SIDEP y la interfaz **TableService.groovy** le permite a todos los módulos generar tablas capaces de soportar cualquier tipo de datos, es por ello que estos módulos la implementan, al igual que la interfaz **Buscador.groovy** capaz de buscar personas, internos, oficiales, trabajadores entre otros objetos.

A continuación se explica a través del siguiente diagrama de componentes el flujo del Caso de Uso Crear Decisión de Cantidad de Internos con PESH para lograr un mayor entendimiento de cómo se relacionan los componentes dentro de los módulos Solicitudes y Decisiones.

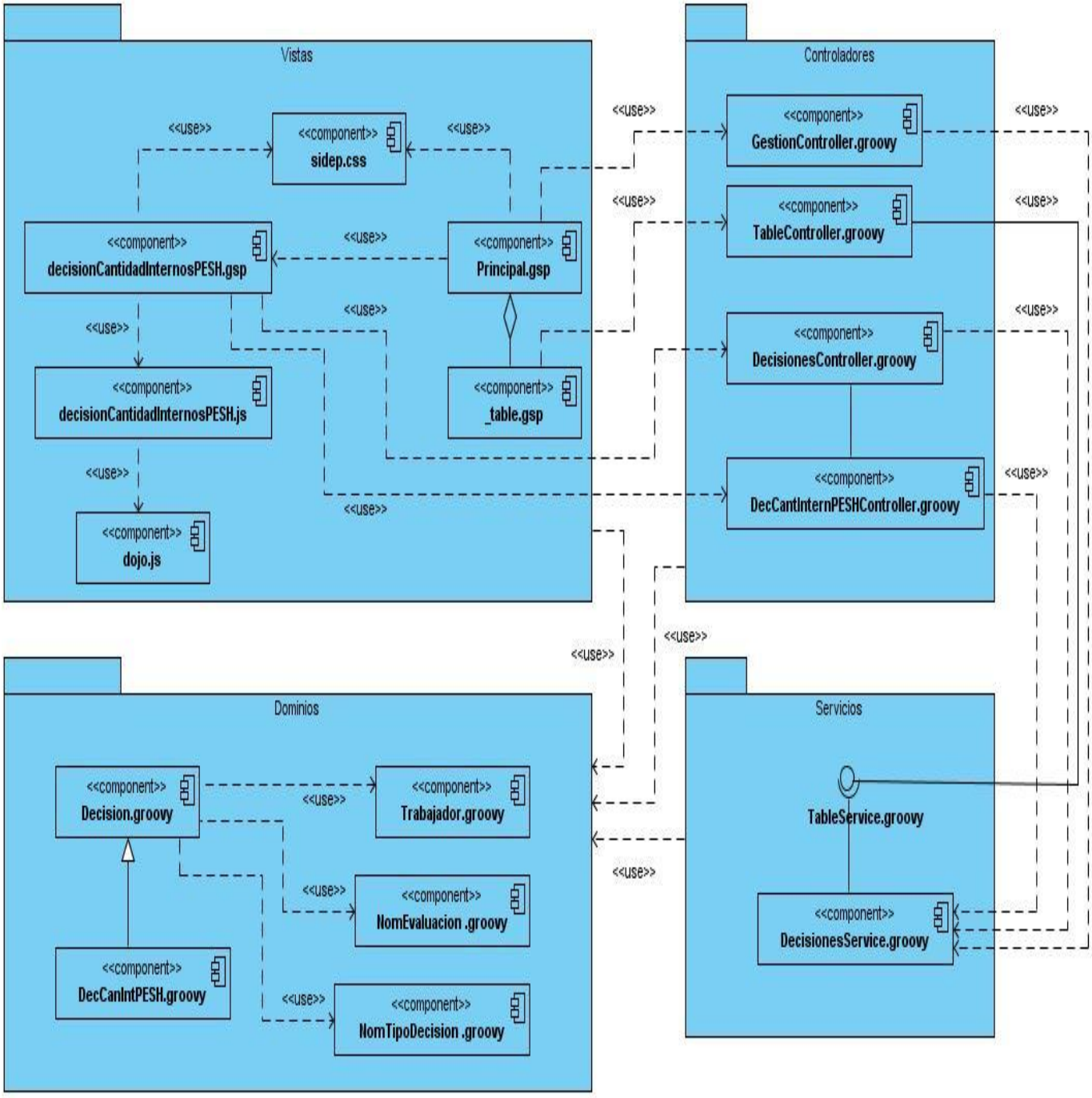

Figura 13: Diagrama de componentes para el Caso de Uso Crear Decisión de Cantidad de Internos con PESH.

La distribución de componentes se realizó por paquetes coincidiendo con la arquitectura de la aplicación. A continuación se detallan los componentes por capas utilizados en el caso de uso Crear Decisión de Cantidad de Internos con PESH. La descripción del mismo puede ser consultada en el Anexo 2.

3.3.1 La capa Vista

Los componentes de esta capa se encuentran estrechamente relacionados entre sí. **DecisionCantidadInternosPESH.js** y **dojo.js** tienen una fuerte dependencia debido a que en la primera se crean, se instancian y se usan objetos definidos en el componente **dojo.js**; mientras que en **DecisionCantidadInternosPESH.gsp** se usan las funciones y los eventos creados en **DecisionCantidadInternosPESH.js**, así como es utilizada además la hoja de estilos **sidep.css** para embellecer la interfaz de usuario. Por último, para la manipulación de listas de elementos tales como tipos de solicitudes y decisiones llevada a cabo por el componte **_table.gsp**, agregados en **Principal.gsp**.

3.3.2 La capa Control

Los controladores no tienen una relación directa entre sí debido a que cada vista conoce cuál es su controlador. Ellos son los responsables de gestionar y responder las peticiones del cliente, así como mostrar las vistas.

3.3.3 Componente Servicio

Los servicios envían la información solicitada por los controladores, para de esta forma poder ser visualizada en los componentes de interfaz de usuario.

3.3.4 Componente Dominio

En este paquete se encuentran las entidades que se utilizan en el caso de uso Gestionar Cantidad de Internos con PESH para guardar la información respecto a las mismas.

3.3.5 Implementación de las clases del Dominio

Las clases de dominio se implementan en el paquete dominio disponible como parte de la estructura arquitectónica del framework. Grails utiliza GORM para controlar el ciclo de vida de las entidades y proporcionar una serie de métodos dinámicos que facilitan las búsquedas. GORM está construido

sobre Hibernate, una herramienta de mapeo objeto-relacional que se encarga de relacionar las entidades de las clases con tablas de la base de datos y las propiedades de las entidades con tablas de las base de datos. Grails crea una sesión de Hibernate para cada petición atendida por la aplicación y la cierra cuando la petición ha terminado de atenderse, justo antes de enviar la respuesta al navegador.

3.3.6 Validaciones

GORM permite restringir los valores que pueden asignarse a las propiedades a través de la palabra reservada constraints, la cual define las reglas de validación para cada atributo. Para generar los respectivos mensajes de error utiliza los archivos y recursos de la carpeta grails-app\i18n que se encuentra en el proyecto.

En la imagen que se presenta a continuación se aplican validaciones en la clase de dominio DecTraCir115 para permitir que los valores horaJuicio y sala sean nulos:

```
class DecTraCir115 extends DecTraIntProv

    int causa
    Date fechaJuicio
    String horaJuicio
    int annoCausa

    NomProvincia provincia
    NomMunicipio municipio
    NomSala sala
    NomTribunal tribunal

    static constraints = {
        horaJuicio nullable: true
        sala nullable: true
    }
```

Figura 14: Validaciones realizadas en la clase de dominio DecTraCir115.

3.3.7 Operaciones sobre el modelo de datos

- **Actualizaciones:** Todas las clases de dominio en la aplicación tienen métodos de instancia que facilitan su manipulación:
  - **save():** Se utiliza para insertar el registro en la base de datos o para actualizarlo si ya existía. Usando el argumento flush: true los datos se envían a la base de datos en el mismo momento en que se invoca el método save ().

- **delete():** Se utiliza para eliminar un registro.

- **Dynamic Finders ("localizadores dinámicos"):** En algunas clases servicios se usaron los métodos **findAll()** y **findAllBy()**. La diferencia es que el primero solo devuelve la primera instancia que cumpla la primera condición de búsqueda, mientras que el segundo devolverá una lista con todos los resultados que correspondan. Estos localizadores son muy potentes para realizar consultas básicas sobre el modelo de datos, pero no permiten hacer búsquedas avanzadas que incluyen muchos campos y comparaciones complejas. Para esto se recomienda construir consultas propias mediante criteria.

3.3.8 Internacionalización

Internacionalizar una aplicación significa diseñarla de tal forma que la interfaz de usuario soporte distintos idiomas sin que sea necesario modificar el código fuente.

Con Grails existe soporte para la internacionalización por medio de archivos de recursos almacenados en la carpeta grails-app/i18n. Los archivos de recursos son ficheros de java en los que se guardan las distintas versiones de cada mensaje en todos los idiomas soportados por la aplicación. Cada idioma está definido en un fichero distinto que incluye en su nombre el local del idioma, por ejemplo:

Messages_es_AR.properties para los mensajes en español de Argentina.

Messages_en.properties para los mensajes en inglés.

Messages_es_es.properties para los mensajes en español de España.

Desde el momento en que se crea el proyecto Grails, este coloca en la carpeta correspondiente archivos de mensajes para 11 idiomas distintos. Con escribir en el navegador el comando ?lang=es se cambia el idioma de los componentes de la aplicación a español.

3.4 Diagrama de Despliegue

Los diagramas de despliegue *"… muestran a los nodos procesadores, la distribución de los procesos y de los componentes. Define la arquitectura física del sistema por medio de nodos interconectados. Estos nodos son elementos de hardware sobre los cuales pueden ejecutarse los elementos de software".* (25)

A continuación se explica a través del siguiente diagrama de despliegue el flujo para los módulos Solicitudes y Decisiones. El cliente accede a la aplicación utilizando el navegador Mozilla Firefox este se comunica con el servidor web que contiene a su vez al servidor de aplicaciones donde se capturan los datos que el cliente insertó y se envían al servidor de base de datos donde se obtiene una respuesta que es enviada nuevamente al servidor de aplicaciones donde es procesada y queda lista para ser consultada por el cliente.

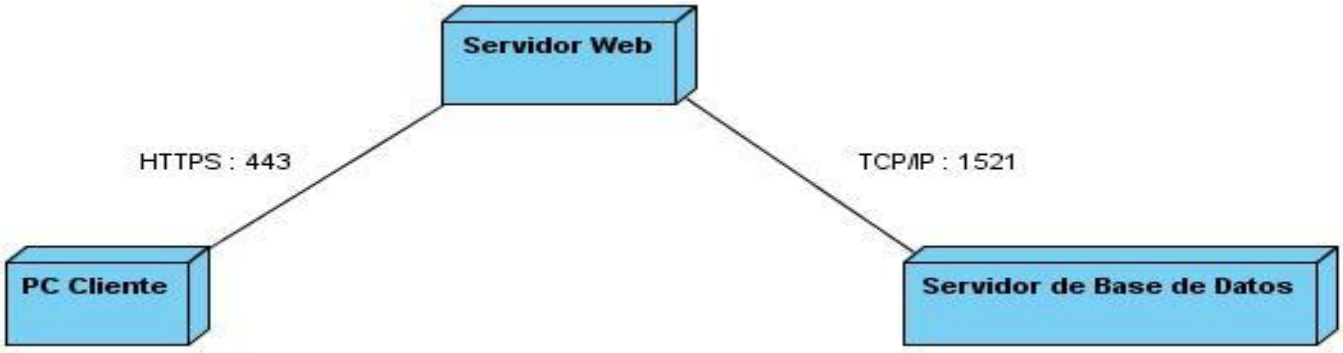

Figura 15: Diagrama de despliegue para los módulos Solicitudes y Decisiones.

Para el correcto despliegue de los módulos se necesita una estructura tecnológica, donde radican el Servidor de Web y el Servidor de Base Datos. A continuación se describen los requisitos no funcionales necesarios para un correcto despliegue de la aplicación:

Requisitos no funcionales de software.

Servidor Web:

- Debe contener la distribución de Linux Suse Enterprise Server 10.0 SP2 y el servidor web Apache Tomcat 6.0.3.
- Necesita tener instalado el JDK 1.6

Servidor de Bases de datos:

- Necesita tener instalado el JDK 1.6
- Se utilizará Oracle 11G Enterprise Edition como sistema gestor de base de datos.

El cliente hará uso de la aplicación a través de clientes ligeros con sistema operativo Windows XP SP3 y navegador Mozilla Firefox 3.6.

Requisitos no funcionales de hardware.

El servidor Web debe disponer de:

- Microprocesador: 4 núcleos, con velocidad igual o superior a los 3.0 GHz
- RAM: 4 GB
- Espacio necesario para la instalación: 250 MB
- Espacio libre requerido: 250 GB

Servidor de Bases de Datos:

- Microprocesador: 4 núcleos, con velocidad igual o superior a los 3.0 GHz
- RAM: 4 GB
- Espacio necesario para la instalación: 2 GB
- Espacio libre requerido: 1 TB

3.5 Pruebas

Las pruebas son una actividad en la cual "… *un sistema o componente es ejecutado bajo unas condiciones específicas. Los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. La prueba está enfocada principalmente en la evaluación y determinación de la calidad del producto. Estas están definidas en estrategias, niveles, tipos, métodos y técnicas de prueba*". (26)

3.6 Estrategia de Prueba

La estrategia de prueba describe *"el enfoque y los objetivos generales de las actividades de prueba. Incluye los niveles de pruebas (unidad, integración, etc.) a ser diseccionados y los tipos de pruebas a ser ejecutadas (funcional, stress, entre otras)".* (26)

La estrategia de prueba define:

- Técnicas de pruebas (manual o automática) y herramientas a ser usadas.
- Qué criterios de éxitos y culminación de la prueba serán usados.
- Consideraciones especiales afectadas por requerimientos de recursos o que tengan implicaciones en la planificación.

3.7 Niveles de Pruebas

Se distinguen los siguientes niveles de pruebas:

- Prueba de desarrollador.
- Prueba independiente.
- Prueba de unidad.
- Prueba de integración.
- Prueba de sistema.
- Prueba de aceptación.

A continuación se describen los tipos de pruebas seleccionados para comprobar el correcto funcionamiento del SIDEP:

Prueba de unidad: Esta prueba está enfocada *"… en los elementos más pequeños del software que pueden ser probados. Aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos y que funcionen como se espera. La prueba de unidad siempre está orientada a caja blanca*". (26)

Prueba de sistema: Esta prueba se realiza cuando el software está funcionando como un todo. "*Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados. En un ciclo iterativo estas pruebas ocurren más temprano, tan pronto como subconjuntos bien formados de comportamiento de casos de uso son implementados*". (26)

3.8 Tipo de Prueba

Se definió que se utilizara el tipo de prueba de funcionalidad. A continuación se describen los aspectos a evaluar en este tipo de prueba:

Funcionalidad

- **Función:** Pruebas fijando su atención en la validación de las funciones, métodos, servicios, casos de uso.
- **Seguridad:** Asegurar que los datos o el sistema solamente sean accedidos por los actores deseados.
- **Volumen:** Enfocada a verificar las habilidades de los programas para manejar grandes cantidades de datos.

3.9 Métodos de Pruebas

Los métodos de prueba son utilizados con el propósito de descubrir fallos y no para demostrar que el software funciona. En las pruebas realizadas a la aplicación se hizo uso del método de Caja Negra.

- **La prueba de caja negra:** Se refiere a las pruebas que "… *se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene*". (26)

3.10 Estrategia de prueba seguida

La estrategia seguida para la realización de las pruebas al SIDEP contempla dos niveles: el nivel de pruebas de Unidad y el nivel de prueba de Sistema.

Pruebas de Unidad:

En este nivel, se realizaron pruebas automáticas haciendo uso del sistema de Testing o Prueba que brinda el propio marco de trabajo utilizado, donde se verifica que un método arroja el resultado que se espera sin tener en cuenta su entorno. Cuando se aplican las pruebas unitarias de un método, Grails no inyectará ninguno de los métodos dinámicos con que cuenta en el momento de ejecutarse la aplicación y es responsabilidad de la persona que está ejecutando las pruebas de crear y gestionar

todos los objetos. Utilizando el sistema Testing fueron probadas todas las clases y sus funcionalidades para un total de tres iteraciones.

Pruebas de Sistema:

Para las pruebas de sistema se utilizó la técnica manual incluyendo el tipo de prueba de funcionalidad con el uso de diseño de casos de prueba, utilizando el método de caja negra. En este se realizaron dos iteraciones de pruebas.

3.11 Resultados

Pruebas de Unidad:

De las tres iteraciones de pruebas unitarias que fueron realizadas se encontraron cinco no conformidades en la primera iteración, una en la segunda iteración y cero en la tercera iteración, las mismas pueden ser consultadas en la tabla de no conformidades que aparece en el Anexo 5.

A continuación se muestra un fragmento del informe generado por el framework sobre las pruebas unitarias realizadas en la tercera iteración:

All Tests

Class	Name	Status	Type	Time(s)
DecCamLuqTests	testProbandoAtributosRelaciones	Success		0.703
DecCanIntPESHTests	testProbandoAtributosRelaciones	Success		0.000
DecCantInternPESHControllerTests	testGestionarDecCantInternPESH	Success		0.594
DecCantInternPESHControllerTests	testValidarDecCantInternPESH	Success		0.000
DecConCir115Tests	testProbandoAtributosRelaciones	Success		0.000
DecConducCircular115ControllerTests	testGestionarDecConducCircular115	Success		0.078
DecConducCircular115ControllerTests	testValidarDecConducCircular115	Success		0.016
DecExpLeqControllerTests	testGestionarDecExpLeq	Success		0.219
DecExpLeqControllerTests	testValidarDecExpLeq	Success		0.000
DecExpLeqTests	testProbandoAtributosRelaciones	Success		0.000
DecOtoLicExtTests	testProbandoAtributosRelaciones	Success		0.016
DecOtorgaLicenExtrapenalControllerTests	testGestionarDecOtorgaLicenExtrapenal	Success		0.047
DecOtorgaLicenExtrapenalControllerTests	testValidarDecOtorgaLicenExtrapenal	Success		0.016

Figura 16: Resultados de las pruebas de unidad efectuadas en la tercera iteración.

Pruebas de Sistema:

Como parte de la ejecución del Método de Caja Negra se encontraron siete no conformidades en la primera iteración y cero en la segunda iteración. Los casos de prueba pueden ser consultados en el Anexo 5.

A continuación se muestra el caso de prueba para el caso de uso CRUD-C solicitud de Conducción:

Descripción de las variables.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
[1]	*Motivo del Conduce*	*Lista desplegable*	*No*	*Se puede modificar el motivo de traslado.*
[2]	*Lugar de Conducción*	*Lista desplegable*	*No*	*Se puede modificar el lugar de conducción.*
[3]	*Origen*	*Campo de texto*	*No*	*Se puede modificar el origen.*

Condiciones de ejecución

El Solicitante de conduce debe estar autenticado en el sistema.Permite actualizar y eliminar las solicitudes que no han sido evaluadas.

SC1 Solicitudes Enviadas

Escenario	Consultar	Eliminar	Actualizar	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Consultar solicitud de "Conducción" .	*V*	*NA*	*NA*	*Se selecciona la solicitud "Conducción" para consultarlo.*	*Muestra los datos de la solicitud de "Conducción" seleccionada.*	*1. Selecciona la opción "Solicitudes Enviadas". 2. Selecciona una solicitud de tipo "Conducción".*
EC 1.2 Eliminar solicitud de "Conducción" .	*NA*	*V*	*NA*	*Se selecciona la opción eliminar para eliminar la solicitud de "Conducción".*	*Eliminala solicitud.*	*1. Selecciona la opción "Solicitudes Enviadas". 2. Selecciona la opción eliminar.*
EC 1.2 Actualizar solicitud de "Conducción" .	*NA*	*NA*	*V*	*Se selecciona la opción actualizar para actualizar la solicitud de "Conducción".*	*Actualiza la solicitud.*	*1. Selecciona la opción "Solicitudes Enviadas". 2. Selecciona la opción actualizar.*

SC2 Consultar solicitud de "Conducción".

Escenario	*Descripción*	*Respuesta del sistema*	*Flujo central*
EC 1.1 Consultar solicitud de "Conducción".	*Consulta la solicitud de "Conducción".*	*El sistema muestra los siguientes campos:* *• Tipo de solicitud* *• Fecha de solicitud* *• De* *• Para* *• Provincia* *• Interno* *• Lugar del conduce* *• Motivo del conduce* *• Origen* *Datos del De:* *• Cargo* *• Grado* *• Primer Nombre* *• Primer Apellido* *Datos del Para:* *• Primer nombre* *• Primer apellido*	*1. Selecciona una solicitud de tipo "Conducción".*

SC3 Eliminar solicitud de "Conducción".

Escenario	*Descripción*	*Respuesta del sistema*	Flujo central	
EC 1.1 Eliminar solicitud de "Conducción".	*Eliminar solicitud de "Conducción".*	*El sistema muestra el mensaje: "¿Está seguro(a) que desea eliminar el elemento seleccionado?".El sistema elimina la solicitud.*	*1. Oprime el botón "Aceptar".*	
EC 1.2 No eliminar	*Se cancela la operación.*	*Se cancela la operación eliminar.*	1. Selecciona el botón "Cancelar".	

SC4 Actualizar solicitud de "Conducción"

Escenario	*Descripción*	*Motivo*	*Origen*	*Lugar*	*Respuesta del sistema*	Flujo central
EC 1.1 Actualizar solicitud de "Conducción"	*Se actualizan los datos de la solicitud de "Conducción"*	*V*	*V*	*V*	*El sistema muestra el mensaje: "¿Está seguro(a) que desea modificar el elemento seleccionado?".El sistema muestra los campos :* *• Origen* *• Lugar* *• Motivo de conduce* *Permite eliminar y adiciona más internos.* *El sistema valida los datos introducidos.El sistema actualiza los datos.*	*1. Inserta los datos que desea modificar.Actualiza los campos para modificar la solicitud.* *Si oprime el botón "Adicionar Interno", ver el flujo alterno. "Adicionar interno".* *Si oprime el botón "Eliminar" para eliminar un interno, ver el flujo alterno . "Eliminar interno".* *2. Oprime el botón "Enviar".*
EC 1.2 No actualizar	*Se cancela la operación.*	*NA*	*NA*	*NA*	*El sistema no actualiza la solicitud.*	1. Selecciona el botón "Cancelar".
EC 1.3 Faltan datos obligatorios.	*Indica que faltan datos que son obligatorios para el traslado.*	*I*	*V*	*V*	*Muestra el mensaje "Los campos en rojo son necesarios" señalando en rojo los campos que se necesitan.*	*1. Introduce los datos obligatorios que faltan en el sistema y oprime el botón "Enviar".*
		V	*I*	*V*	*Muestra el mensaje "Los campos en rojo son necesarios" señalando en rojo los campos que se necesitan.*	*1. Introduce los datos obligatorios que faltan en el sistema y oprime el botón "Enviar".*

		V	V	I	Muestra el mensaje "Los campos en rojo son necesarios" señalando en rojo los campos que se necesitan.	1. Introduce los datos obligatorios que faltan en el sistema y oprime el botón "Enviar".
EC 1.4 Adicionar interno	Se adiciona un interno a la tabla	V	V	V	El sistema ejecuta el CU "Buscar Interno".Muestra el nombre del interno en el campo interno.	1. Selecciona el interno y oprime el botón "Siguiente".Oprime el botón "Aceptar".
EC 1.5 Eliminar Interno	Se elimina un interno de la tabla	V	V	V	El sistema muestra un mensaje "¿Está seguro que desea eliminar al interno?".El sistema elimina el interno del listado.	Oprime el botón "Aceptar". Oprime el botón "Cancelar", ver el flujo alterno "No elimina interno".
EC 1.6 . No elimina interno	Se cancela la operación.	V	V	V	El sistema no elimina al interno.	Oprime el botón "Cancelar".

Resultados generales

En la siguiente figura se muestra un gráfico donde se reflejan las iteraciones que se realizaron en los distintos niveles y la cantidad de no conformidades encontradas cuando se realizaron los tipos de pruebas de unidad y funcionalidad.

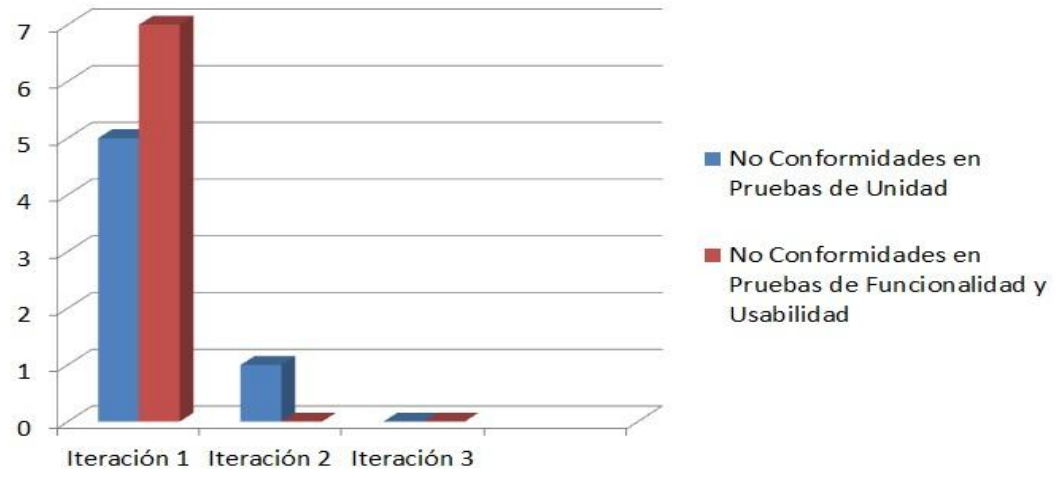

Figura 17: Cantidad de no conformidades encontradas en las distintas iteraciones.

3.12 Conclusiones Parciales

En este capítulo se mostraron y explicaron los artefactos generados durante la implementación, entre ellos el diagrama de despliegue y el diagrama de componentes, además, se abordaron otros aspectos del flujo de implementación tales como la implementación de las clases de dominio, las validaciones, las operaciones sobre el modelo de datos y la internacionalización. Para comprobar el correcto funcionamiento de la aplicación se realizaron pruebas de unidad y caja negra, logrando así un sistema seguro y acorde a las necesidades del cliente. Se detectaron un total de trece no conformidades las cuales fueron resueltas.

Conclusiones Generales

Como parte de la realización de la investigación y en virtud de dar cumplimiento a los objetivos trazados, se arribaron a las siguientes conclusiones:

- Los sistemas analizados no hacen a alusión a la gestión de solicitudes y decisiones o implementan un flujo que difiere de las características del sistema penitenciario cubano.
- Fueron seleccionadas como tecnologías y herramientas para el desarrollo de la solución: Java como plataforma de desarrollo, Grails en su versión 1.3.7 como framework para la implementación de la solución, Oracle 11g como gestor de Base de Datos, Dojo en su versión 1.5 como framework de presentación y Apache Tomcat en su versión 6.0 como contenedor web.
- Fueron obtenidos los artefactos correspondientes al flujo de Análisis y Diseño, específicamente el diseño, a partir de los requisitos acordados con el cliente y el análisis realizado con anterioridad.
- Fueron obtenidos los artefactos correspondientes a los flujos Implementación y Pruebas, realizándose los diagramas de clases, base de datos, diagrama de componentes, de despliegue y casos de pruebas.
- Se aplicaron pruebas de unidad y de caja negra, solucionando las no conformidades detectadas.

Recomendaciones

Luego de haber concluido la investigación, en base a los resultados obtenidos y su alcance, se recomienda:

- Desarrollar los manuales de usuarios como apoyo al personal que hará uso de los módulos.
- Implantar los módulos en un ambiente real para su explotación y la retroalimentación hacia el Centro de Informatización de la Seguridad Ciudadana (ISEC).
- Darle seguimiento a nuevos requisitos que surjan a partir de cambios en las legislaciones vigentes o en el flujo de solicitudes y decisiones en el Sistema Penitenciario Cubano.

Referencias Bibliográficas

1. **Zequeira Peña, Dr. Tte. Cnel. Alfonso y Céspedes Quesada, Lic.1er Tte. Aimeé.** *Vocabulario Jurídico Penitenciario.* Ciudad de la Habana : MININT, 2005.

2. ***Código Penal Cubano**, Ley No. 62.* República de Cuba : s.n.

3. **Corrections.com.** Offendertrak Corrections Management System. [En línea] [Citado el: 4 de junio de 2012.] www.corrections.com/articles/7279.

4. **CISCO**. *CISCO Software.* [En línea] [Citado el: 4 de junio de 2012.] http://www.cisco-ps.com/jams/.

5. **Spillman.com**. *Spillman Corrections Management System.* [En línea] [Citado el: 4 de junio de 2012.] http://www.spillman.com/corrections/.

6. **Ministerio de Justicia de Ecuador**. *Ministerio de Justicia de Ecuador.* [En línea] [Citado el: 5 de junio de 2012.] https://www.minjusticia.gob.ec.

7. **Ministerio de Gobierno y Justicia.** Dirección general del sistema penitenciario. *Dirección general del sistema penitenciario.* [En línea] [Citado el: 6 de junio de 2011.] http://www.sistemapenitenciario.gob.pa.

8. **Ivar JACOBSON, Grady RUMBAUGH, James BOOCH.** *El Proceso Unificado de Desarrollo de Software.* Ciudad Habana : s.n. 2002.

9. **J.R, Cupertino.** *El Lenguaje Unificado de Modelado.* California : s.n., 1998.

10. **Visual Paradigm International.** Visual Paradigm. [En línea] [Citado el: 10 de diciembre de 2011.] http://www.visual-paradigm.com.

11.**SpringSource.** SpringSource. [En línea] [Citado el: 11 de diciembre de 2011.] http://www.springsource.com/developer/sts.

12. **Abián, Miguel Ángel.** *Java2EE vs Net.* Madrid : s.n., 2004.

13. **Moreno, Mariano.** *Java la programación del futuro.* Montevideo : Rosgal, 1997. 987-9131-38-X.

14. **SpringSource.** Groovy. [En línea] [Citado el: 14 de diciembre de 2011.] http://groovy.codehaus.org.

15. **SpringSource.**Grails. [En línea] [Citado el: 26 de marzo de 2012.] http://www.grails.org.

16. **Chuck Musciano, Bill Kennedy.** *HTML La Guía Completa.* México : McGRAW-HILL, 1999. 970-10-2141-X.

17. **Camy, Lázaro Issi.** *JavaScript.* Madrid : Anaya, 2002. 84-415-1384-8.

18. **Dojo Foundation.** DojoTolkit. [En línea] [Citado el: 26 de marzo de 2012.] http://dojotoolkit.org/.

19. **López, Javier.** *Oracle Fundamentos para el desarrollo de aplicaciones web.* Buenos Aires : Artes Gráficas , 2001. 987-526-071-1.

20.**Embarcadero Technologies Inc.** Embarcadero. [En línea] http://www.embarcadero.com/products/er-estudio.

21. **Apache Software Foundation.** Apache Tomcat. [En línea] [Citado el: 11 de diciembre de 2011.] http://tomcat.apache.org/.

22. **Larman, Craig.** *UML Y Patrones. Introducción al análisis y diseño orientado a objeto.* Mexico : Prentice Hall Hispanoamericana, 1999. 970-17-0261-1.

23. **Glen Smith, Peter Ledbrook.** *Grails in Action.* New York : Manning, 2009. 978-1-933988-93-1.

24. **Sommerville, Ian.** *Ingeniería del Software Séptima Edición.* Madrid : Pearson Addison Wesley, 2004. 84-7829-074-5.

25. **Rational software**. *Rational software.* [En línea] [Citado el: 20 de mayo de 2012.] http://www.rational.com.

26. **Pressman, Roger S.** *Ingeniería del software .* Madrid : McGraw-Hill, 1998. 84-205-3093-X.

Bibliografía

1. **Zequeira Peña, Dr. Tte. Cnel. Alfonso y Céspedes Quesada, Lic.1er Tte. Aimeé.** *Vocabulario Jurídico Penitenciario.* Ciudad de la Habana : MININT, 2005.

2. ***Código Penal Cubano****, Ley No. 62.* República de Cuba : s.n.

3. **Corrections.com.** Offendertrak Corrections Management System. [En línea] [Citado el: 4 de junio de 2012.] www.corrections.com/articles/7279.

4. **CISCO**. *CISCO Software.* [En línea] [Citado el: 4 de junio de 2012.] http://www.cisco-ps.com/jams/.

5. **Spillman.com**. *Spillman Corrections Management System.* [En línea] [Citado el: 4 de junio de 2012.] http://www.spillman.com/corrections/.

6. **Ministerio de Justicia de Ecuador**. *Ministerio de Justicia de Ecuador.* [En línea] [Citado el: 5 de junio de 2012.] https://www.minjusticia.gob.ec.

7. **Ministerio de Gobierno y Justicia.** Dirección general del sistema penitenciario. *Dirección general del sistema penitenciario.* [En línea] [Citado el: 6 de junio de 2011.] http://www.sistemapenitenciario.gob.pa.

8. **Ivar JACOBSON, Grady RUMBAUGH, James BOOCH.** *El Proceso Unificado de Desarrollo de Software.* Ciudad Habana : s.n. 2002.

9. **J.R, Cupertino.** *El Lenguaje Unificado de Modelado.* California : s.n., 1998.

10. **Visual Paradigm International.** Visual Paradigm. [En línea] [Citado el: 10 de diciembre de 2011.] http://www.visual-paradigm.com.

11. **SpringSource.** SpringSource. [En línea] [Citado el: 11 de diciembre de 2011.] http://www.springsource.com/developer/sts.

12. **Abián, Miguel Ángel.** *Java2EE vs Net.* Madrid : s.n., 2004.

13. **Moreno, Mariano.** *Java la programación del futuro.* Montevideo : Rosgal, 1997. 987-9131-38-X.

14. **SpringSource.** Groovy. [En línea] [Citado el: 14 de diciembre de 2011.] http://groovy.codehaus.org.

15. **SpringSource.**Grails. [En línea] [Citado el: 26 de marzo de 2012.] http://www.grails.org.

16. **Chuck Musciano, Bill Kennedy.** *HTML La Guía Completa.* México : McGRAW-HILL, 1999. 970-10-2141-X.

17. **Camy, Lázaro Issi.** *JavaScript.* Madrid : Anaya, 2002. 84-415-1384-8.

18. **Dojo Foundation.** DojoTolkit. [En línea] [Citado el: 26 de marzo de 2012.] http://dojotoolkit.org/.

19. **López, Javier.** *Oracle Fundamentos para el desarrollo de aplicaciones web.* Buenos Aires : Artes Gráficas , 2001. 987-526-071-1.

20. **Embarcadero Technologies Inc.** Embarcadero. [En línea] http://www.embarcadero.com/products/er-estudio.

21. **Apache Software Foundation.** Apache Tomcat. [En línea] [Citado el: 11 de diciembre de 2011.] http://tomcat.apache.org/.

22. **Larman, Craig.** *UML Y Patrones. Introducción al análisis y diseño orientado a objeto.* Mexico : Prentice Hall Hispanoamericana, 1999. 970-17-0261-1.

23. **Glen Smith, Peter Ledbrook.** *Grails in Action.* New York : Manning, 2009. 978-1-933988-93-1.

24. **Sommerville, Ian.** *Ingeniería del Software Séptima Edición.* Madrid : Pearson Addison Wesley, 2004. 84-7829-074-5.

25. **Rational software**. *Rational software.* [En línea] [Citado el: 20 de mayo de 2012.] http://www.rational.com.

26. **Pressman, Roger S.** *Ingeniería del software .* Madrid : McGraw-Hill, 1998. 84-205-3093-X.

27.**MINREX**. Sitio del Ministerio de Relaciones Exteriores de Cuba. [Online] http://www.cubaminrex.cu/Ministerio/ministerio.html.

28. **Oracle**. Oracle. [Online] http://www.oracle.com/es/index.html.

29. **Rolando Alfredo Hernández León, Sayda Coello González.** *El Proceso de Investigación Científica.* Ciudad de La Habana : Editorial Universitaria, 2011. 978-959-16-1307-3.

30. **Ricardo Grau, Cecilia Correa, Mauricio Rojas.** *Metodología de la Investigación.* Bogotá : Editorial Coruniversitaria, 2004. 958-8028-10-8.

31. **Yadira Benavides Zaila, Juan Carlos Gómez Correa.** *Diseño e implementación de los módulos Decisiones y Egresos del Sistema de Gestión Penitenciaria de la República Bolivariana de Venezuela.* Ciudad de La Habana : Editorial Universitaria, 2008.

32. **José A. Gómez Llano, Yohairo B. Consuegra Peña.** *Diseño e Implementación del módulo Situación Legal del Sistema Penitenciario Nacional.* Ciudad de La Habana : Editorial Universitaria, 2010.