

Universidad de las Ciencias Informáticas



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

Título

Desarrollo de los módulos Administración de biblioteca y Emulación del Sistema Informativo de la Dirección de Establecimientos Penitenciarios (SIDEPE).

Autor

Damián Gazmuriz Adán.

Tutora

Ing. Liset Schery Sánchez.

Co-tutor

Msc. Julio Omar Prieto Entenza.

La Habana, 2011

DECLARACIÓN DE AUTORÍA

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

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

Damián Gazmuriz Adán

Firma del tutor

Ing. Liset Schery Sánchez

Firma de la tutora

Msc. Julio Omar Prieto Entenza

Firma del co-tutor

DEDICATORIA

A mis padres, por ser los máximos responsable de mi existencia, por amarme con tanta pasión y dedicación, por ser los motores impulsadores para la realización de este sueño.
“Verme graduado”

A mi familia y amigos de toda la vida.
A todos los que han confiado y siguen confiando en mí.
A todo aquel que me brindó apoyo en mi vida.
A todos, gracias.

AGRADECIMIENTOS

A mi mamá, por hacer de mí una persona íntegra y capaz de luchar por sus sueños. Por ser la luz de mi vida, por ser mi razón de existir y ser mejor persona.

A mi papá, por estar conmigo en todo momento apoyándome. Por sentirse orgulloso de mí siempre, y así impulsarme a ser mejor cada día.

A mi hermano, por siempre estar ahí en las buenas y malas. Por constituir mi ejemplo de superación y mi razón de orgullo.

A mi tutora Liset Schery, por dedicar tanto al desarrollo de esta tesis y brindándome todo lo necesario para la realización de la misma.

Al profesor Roberto Granda, por su apoyo en el desarrollo de la tesis ayudándome a esclarecer todas las dudas que surgieron en el proceso.

A Noris, por haberme apoyado incondicionalmente durante estos 3 años. Por impulsarme a seguir logrando mis sueños y metas, este resultado es tuyo también.

A todos mis familiares, por apoyarme y estar presentes de una forma u otra durante toda mi vida.

A todas las personas que han estado a lo largo del camino a mi lado, en las buenas y en las malas.

RESUMEN

Con la actual informatización que se lleva adelante en el país se han beneficiado muchos sectores de la sociedad cubana, dentro de los que se incluye el sector penitenciario. Debido a esta iniciativa en la Universidad de las Ciencias Informáticas, en colaboración con el estado mayor de Ministerio del Interior (MININT), se desarrolla un proyecto encargado de implementar un sistema de gestión para los centros penitenciarios en Cuba, el cual pretende propiciar una creciente mejora en el proceso de toma de decisiones, eficiencia en el trabajo y elevar los niveles de conformidad por parte de los penales.

El propósito fundamental de este trabajo es el desarrollo de los módulos Administración de Biblioteca y Emulación, los cuales brindan información referente a los préstamos, devoluciones de libros, así como el control de la puntuación que reciben los internos. Debido a esto se toma como punto de partida el análisis de los requisitos de software establecidos en acuerdo con el cliente, haciendo uso de las tecnologías y herramientas definidas en la arquitectura del proyecto.

Los resultados que se esperan con la implementación de los módulos ya mencionados es que el sistema cumpla con todas las expectativas, brindando una aplicación capaz de gestionar todo el flujo de procesos del sistema penitenciario cubano.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN.....	10
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	14
1.1. Introducción.....	14
1.2. Marco conceptual.....	14
1.3. Valoración del estado del arte.....	15
1.3.1. Sistemas nacionales e internacionales.....	15
1.4. Metodología, herramientas, tecnologías y lenguajes.....	18
1.4.1. Metodología de desarrollo.....	18
1.4.2. Herramientas CASE.....	20
1.4.3. Herramientas de desarrollo.....	21
1.4.4. Gestor de Base de Datos.....	22
1.4.5. Tecnologías.....	22
1.4.6. Plataforma de desarrollo.....	24
1.4.7. Lenguajes.....	24
1.5. Conclusiones parciales.....	25
CAPÍTULO 2. DISEÑO DEL SISTEMA.....	27
2.1. Introducción.....	27
2.2. Reglas del negocio.....	27
2.3. Requisitos funcionales.....	29
2.4. Diagrama de casos de uso del sistema.....	31
2.5. Principales funcionalidades.....	34
2.6. Modelo del diseño.....	38
2.7. Arquitectura del sistema.....	39
2.8. Patrones del diseño.....	41
2.8.1. Patrones GRASP.....	42
2.8.2. Patrones GOF.....	43

2.8.3. Otro patrón de diseño utilizado	43
2.9. Diagramas de clases	44
2.9.1. Descripción de las clases más significativas	44
2.9.2. Diagramas de clases del diseño	50
2.10. Diagramas de colaboración	53
2.11. Diseño de la Base de Datos	56
2.11.1. Descripción de las tablas más significativas	59
2.12. Conclusiones parciales	62
CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS	63
3.1. Introducción	63
3.2. Modelo de implementación	63
3.3. Diagramas de componentes	63
3.4. Seguridad	67
3.5. Internacionalización	68
3.6. Diagrama de despliegue	69
3.7. Pruebas	71
3.7.1. Estrategia de prueba	71
3.7.2. Resultados de las pruebas	75
3.8. Conclusiones parciales	78
CONCLUSIONES GENERALES	79
RECOMENDACIONES	80
REFERENCIAS BIBLIOGRÁFICAS	81

ÍNDICE DE TABLAS

Tabla 1 Reglas del negocio del módulo Administración de Biblioteca	28
Tabla 2 Reglas del negocio del módulo Emulación	29
Tabla 3 Requisitos funcionales módulo Administración de biblioteca	30
Tabla 4 Requisitos funcionales módulo Emulación	31
Tabla 5 Funcionalidades del módulo Administración de biblioteca	36
Tabla 6 Funcionalidades del módulo Emulación	38
Tabla 7 Descripción de entidades del módulo Administración de la biblioteca	44
Tabla 8 Descripción de la clase obraLiteraria.html del módulo Administración de la biblioteca	45
Tabla 9 Descripción de la clase obraLiterariaController del módulo Administración de la biblioteca	46
Tabla 10 Descripción de la clase obraLiterariaService del módulo Administración de la biblioteca	46
Tabla 11 Descripción de entidades del módulo Emulación	47
Tabla 12 Descripción de la clase administrarPuntuacion.html del módulo Emulación	47
Tabla 13 Descripción de la clase administrarPuntuacionController del módulo Emulación	48
Tabla 14 Descripción de la clase parametroService del módulo Emulación	49
Tabla 15 Descripción de la clase aspectoService del módulo Emulación.....	49
Tabla 16 Descripción de la clase puntuacion.js del módulo Emulación	50
Tabla 17 Descripción de la tabla ObraLiteraria del módulo Administración de biblioteca	60
Tabla 18 Descripción de la tabla Prestamo del módulo Administración de biblioteca	60
Tabla 19 Descripción de la tabla Parametro del módulo Emulación	61
Tabla 20 Descripción de la tabla PropuestaEstimulo del módulo Emulación.....	61
Tabla 21 Descripción de la tabla ConfiguracionEmulacionColectiva del módulo Emulación.....	62
Tabla 22 No conformidades del módulo Administración de biblioteca	76
Tabla 23 No conformidades del módulo Emulación	76

ÍNDICE DE FIGURAS

Figura 1 Fases y flujos de trabajo de RUP.....	19
Figura 2 Diagrama de casos de uso del sistema módulo Administración de biblioteca	32
Figura 3 Diagrama de casos de uso del sistema módulo Emulación.....	33
Figura 4 Arquitectura del sistema.....	40
Figura 5 DCD Gestionar obra literaria del módulo Administración de biblioteca.....	51
Figura 6 DCD Administrar puntuación del módulo Emulación	52
Figura 7 DC Registrar obra literaria (Gestionar obra literaria) del módulo Administración de biblioteca	54
Figura 8 DC Administrar puntuación del módulo Emulación.....	55
Figura 9 DER del módulo Administración de biblioteca.....	57
Figura 10 DER del módulo Emulación	58
Figura 11 Diagrama de componentes general del módulo Administración de biblioteca	64
Figura 12 Diagrama de componentes general del módulo Emulación.....	64
Figura 13 Diagrama de componentes específico del módulo Administración de biblioteca	65
Figura 14 Diagrama de componentes específico del módulo Emulación.....	66
Figura 15 Ejemplo de la utilización del Sidep-Admin en el módulo Emulación	67
Figura 16 Ejemplo de la seguridad en el módulo Emulación	68
Figura 17 Fichero para los mensajes en español.....	69
Figura 18 Diagrama de despliegue de los módulos Administración de biblioteca y Emulación.....	70
Figura 19 CP del CU Registrar obra literaria (Gestionar obra literaria) del módulo Administración de biblioteca	73
Figura 20 CP del CU Administrar puntuación del módulo Emulación	74
Figura 21 Resultados obtenidos en las pruebas al módulo Administración de biblioteca	77
Figura 22 Resultados obtenidos en las pruebas al módulo Emulación.....	77

INTRODUCCIÓN

En la actualidad las Tecnologías de la Informática y las Comunicaciones (TIC) se encuentran en constante desarrollo, lo que ha propiciado que se hayan logrado muchos avances con respecto a años anteriores. Como consecuencia de este desarrollo surge una nueva era, donde la información y el conocimiento juegan un papel primordial para el funcionamiento de cualquier organización.

La informatización es, en sentido general, el rasgo más importante de las actuales transformaciones como paso previo e imprescindible para ese desarrollo científico técnico, por lo que se hace necesario producir profundos cambios en los sistemas y estructuras del país.

El sistema penitenciario nacional, acogiéndose a los cambios actuales, lleva adelante la automatización de todos sus procesos. Con la colaboración de la Universidad de las Ciencias Informáticas (UCI) y teniendo como precedente un sistema implementado en 1989, el cual llevaba a cabo la automatización de los datos pertenecientes a los reclusos y diferentes aspectos de control penal, la jefatura del Estado Mayor del MININT autorizó el inicio del proyecto Prisiones Cuba, el cual tendría como objetivo fundamental obtener un software para la centralización de toda la información referente a los establecimientos penitenciarios del país: Sistema Informativo de la Dirección de Establecimientos Penitenciarios.

El SIDEP cuenta con diferentes subsistemas que están constituidos por módulos, los que representan los procesos que se desarrollan en el sistema penitenciario nacional. Entre los módulos que componen este sistema se incluyen Administración de biblioteca y Emulación, los cuales controlan el mantenimiento organizativo desde el punto de vista estadístico, como el desarrollo de actividades que aportan a la enseñanza y aprendizaje relacionado específicamente a la biblioteca; por otra parte, la gestión de la información referente a los puntos que reciben los mismos en la emulación, dicha puntuación es obtenida de la incorporación de los internos a los distintos programas deportivos, culturales, literarios y otros cursos, así como su comportamiento en reclusión.

Los procesos relacionados con los módulos Administración de biblioteca y Emulación presentan determinados problemas que atentan en contra de su correcto funcionamiento, como por ejemplo:

- Son desarrollados manualmente, lo que dificulta el control y seguimiento de los reclusos atendidos diariamente.
- No se cuenta con una Base de Datos (BD) que permita recoger información acerca de los temas referentes al control del interno.

Derivado de estos problemas se hace necesaria la automatización de dichos módulos para la obtención de una solución informática acorde a lo antes planteado.

A partir de la situación problemática expuesta surge como **problema a resolver**:

- ¿Cómo gestionar los parámetros que miden la emulación de los internos y los procesos de préstamos de libros en la biblioteca en el SIDEP?

En correspondencia con lo anterior surge como **objeto de estudio**:

- Los procesos de tratamiento educativo en los sistemas penitenciarios.

Para darle solución se trazó como **objetivo general**:

- Desarrollar los módulos Administración de biblioteca y Emulación del SIDEP a partir del análisis de los requisitos de software establecidos con el cliente y haciendo uso de la arquitectura y las tecnologías definidas para el proyecto.

La investigación se enmarca en el **campo de acción**:

- La gestión de los procesos de administración de la biblioteca y emulación en el sistema penitenciario cubano.

Se plantea la siguiente **idea a defender**:

- Con el desarrollo de los módulos Administración de biblioteca y Emulación del SIDEP se lograrán gestionar los parámetros que miden la emulación de los internos y los procesos de préstamos de libros en la biblioteca en los centros penitenciarios de Cuba.

Obteniendo como **posibles resultados**:

- Modelo de diseño de los módulos Administración de biblioteca y Emulación del SIDEP.
- Modelo de implementación de los módulos Administración de biblioteca y Emulación del SIDEP.
- Validación de los módulos Administración de biblioteca y Emulación del SIDEP.

Como **objetivos específicos** se trazaron los siguientes:

- Elaborar el marco teórico de la investigación.
- Realizar el modelo de diseño de los requisitos planteados para los módulos Administración biblioteca y Emulación del SIDEP.
- Implementar los módulos Administración de biblioteca y Emulación del SIDEP.

- Realizar las pruebas de calidad a los módulos Administración de biblioteca y Emulación del SIDEPE.

Como apoyo a los objetivos específicos se plantearon las siguientes **tareas de investigación**:

- Estudio y análisis de las tendencias actuales de los sistemas penitenciarios en el mundo.
- Descripción de las tecnologías y herramientas a utilizar en el diseño e implementación de los módulos Administración de biblioteca y Emulación.
- Confección del modelo de diseño para los módulos Administración de biblioteca y Emulación.
- Confección del diagrama de componentes para los módulos Administración de biblioteca y Emulación.
- Implementación de los módulos Administración de biblioteca y Emulación.
- Diseñar la estrategia de prueba para los módulos Administración de biblioteca y Emulación.
- Ejecución de pruebas a los módulos Administración de biblioteca y Emulación.

Para la obtención de estos resultados se plantea la necesidad del uso de diferentes **Métodos Científicos**, como estrategia general, que orienta y permite organizar globalmente la actividad hacia el conocimiento y la solución al problema como:

Histórico-Lógico: Se utiliza para el análisis de la trayectoria y evolución de la metodología de desarrollo de software y de las herramientas que se usarán durante el trabajo.

Analítico-Sintético: Se utiliza en el análisis de la bibliografía donde se resumen los elementos más importantes de la gestión del aprendizaje teniendo en cuenta el objeto de la investigación.

Modelación: Se utiliza para la elaboración de diagramas, figuras y otros artefactos importantes en las diferentes disciplinas a desarrollar, se escoge este método ya que se pueden crear abstracciones con el propósito de explicar la realidad.

El trabajo de diploma está constituido por 3 capítulos en los que se encuentra el contenido distribuido de la siguiente forma:

Capítulo 1: Fundamentación Teórica.

Este capítulo muestra la fundamentación teórica de la investigación: el estado del arte de algunas de las soluciones informáticas existentes relacionado con los módulos Administración de biblioteca y Emulación, así como un estudio de los elementos de diseño de software y las metodología, lenguajes y herramientas de modelado utilizadas para el desarrollo en el proyecto.

Capítulo 2: Diseño del Sistema.

En este capítulo se relacionan los aspectos relacionados con la disciplina Diseño aplicado a los módulos Administración de biblioteca y Emulación. Además incluyen los diagramas de clases del diseño con estereotipos web, los diagramas de secuencia y el modelo físico de datos. Se puntualizan las funcionalidades del sistema y la arquitectura que se utiliza en el proyecto para el desarrollo de la aplicación.

Capítulo 3: Implementación y Pruebas.

En este capítulo se enfatiza en la validación del sistema, empleando pruebas de caja negra y pruebas de caja blanca mediante la realización de casos de prueba, con el objetivo de detectar no conformidades existentes en la aplicación y poder solucionarlas para obtener un software que responda a los requisitos definidos por el cliente.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En el presente capítulo se explican elementos relacionados con el sistema penitenciario cubano, enfocados en los procesos de administración de la biblioteca y emulación. Se realiza un estudio de las tendencias actuales en el uso de las aplicaciones informáticas para la gestión de la información, para conocer cómo se han venido desarrollando dentro y fuera del país. También se explican los aspectos más importantes de la metodología, las tecnologías, las herramientas y lenguajes a utilizar para dar solución al problema propuesto.

1.2. Marco conceptual

Sistema penitenciario

Manuel Osorio, creador del diccionario de Ciencias Jurídicas, Políticas y Sociales, asocia el término sistema penitenciario con régimen penitenciario, definiéndolo como: "Conjunto de normas legislativas o administrativas encaminadas a determinar los diferentes sistemas adoptados para que los penados cumplan sus condenas. Se encamina a obtener la mayor eficacia en la custodia o en la readaptación social de los delincuentes. Esos regímenes son múltiples, varían a través del tiempo y van desde el aislamiento absoluto y de tratamiento rígido, hasta el sistema de puerta abierta con libertad vigilada". (1) Se puede concluir que el sistema penitenciario, es el conjunto de normas generales, establecidas y específicas referidas a las penas en sí, el modo de su cumplimiento y el tratamiento de los penados y procesados.

Cada gobierno define la estructura de su sistema penitenciario de acuerdo a la legislación y las condiciones reales que posee. En el caso de la República de Cuba, tal sistema está constituido por la legislación vigente, los métodos que se emplean para lograr su funcionamiento, las diferentes dependencias encargadas de su aplicación, los equipos de trabajo y la infraestructura carcelaria. El Reglamento del Sistema Penitenciario de Cuba establece que, el sistema penitenciario cubano es el encargado de garantizar el proceso de ejecución de la sanción de privación de libertad, de la sanción de trabajo correccional con internamiento, la medida de seguridad reeducativa de internamiento y la medida cautelar de prisión provisional. Este sistema, dirigido por la DEP del MININT, 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 de los internos.

Como iniciativa en el sistema penitenciario cubano se creó un método de emulación para motivar a los internos a participar dinámicamente en las actividades y tareas destinadas a impulsar el proceso de readaptación y reinserción a la sociedad de cada uno de estos en los penales. A través de la emulación se eligen los internos que son destacados en períodos determinados, teniendo derecho a disfrutar de estímulos en dependencia de la puntuación y la reincidencia en esta buena conducta.

Además, en los establecimientos penitenciarios se le da seguimiento a los procesos relacionados con la biblioteca, elemento que también se incluye como parámetro medido en la emulación, debido a la importancia que reviste la interacción de los internos con la literatura y las actividades que se desarrollan en este ámbito.

1.3. Valoración del estado del arte

1.3.1. Sistemas nacionales e internacionales

Actualmente el desarrollo de las TIC ha propiciado que las organizaciones basen su desempeño en el uso de aplicaciones informáticas para la gestión de la información. La información que es generada por estas organizaciones se guarda en BD relacionales, donde los datos se acumulan en grandes volúmenes que son humanamente difíciles de manejar en un tiempo razonable.

A continuación se analizarán algunas soluciones informáticas existentes en los sistemas penitenciarios del mundo y de Cuba, los cuales se estudiarán para determinar si pueden constituir una posible solución o parte de la solución del presente trabajo, teniendo en cuenta las funcionalidades que brindan.

Sistema de Gestión Penitenciaria

El Sistema de Gestión Penitenciaria (SIGEP), desarrollado en la UCI y desplegado en la República Bolivariana de Venezuela, es una aplicación informática que tiene como objetivo fundamental controlar la situación operativa de los establecimientos penitenciarios y constituye una base para la toma de decisiones tácticas y estratégicas de la Dirección General de Custodia y Rehabilitación del Recluso, teniendo en cuenta las legislaciones del sistema penitenciario de este país.

Este sistema se ha puesto en práctica en tres tipos de entidades dentro del sistema penitenciario venezolano: la sede central, los centros penales externos y los centros penales internos; (2) logrando mantener centralizada la información relacionada con los elementos más importantes de los internos.

El proceso de biblioteca está incluido en el SIGEP, pero no abarca la totalidad de las funcionalidades que satisfacen las necesidades de información del sistema penitenciario cubano. En el caso de emulación, no constituye un proceso de interés para las autoridades de este país, por lo que el sistema no gestiona ninguna información relacionada con este.

Sistema Penitenciario del Gobierno de Panamá

Este sistema fue creado a principios de año 1997, como resultado de un proyecto financiado por la Programa de las Naciones Unidas para el Desarrollo (PNUD) y el gobierno español en coordinación con la Dirección General de Sistemas Penitenciarios de Panamá (DGSP). La finalidad del software es mantener en una base de datos los registros de los internos que están detenidos en los centros penales a nivel nacional.

Este sistema tiene como objetivo planificar, organizar, administrar, coordinar, supervisar y dirigir el funcionamiento de los diferentes tipos de centros penitenciarios existentes y por crearse en la república de Panamá, dentro del marco del respeto a los derechos humanos, los principios de seguridad, rehabilitación y defensa social, y la aplicación de los avances científicos en su gestión. (3)

El sistema registra la información referente a los datos personales del interno, antecedentes médicos, datos socioeconómicos y jurídicos y algún otro dato de importancia, además muestra datos relacionados con los centros penitenciarios del país. La información capturada se registra automáticamente en una base de datos Oracle que radica en la sede central y la cual está disponible para los diferentes departamentos que tienen acceso al sistema. Esto garantiza que se refleje de forma inmediata los cambios en el expediente del interno tales como trasposos de autoridad, traslados de un centro a otro, libertades, diligencias médicas, jurídicas y la realización del cómputo automático de la sentencia.

Las limitaciones se evidencian en los centros que no poseen enlace aún con la dirección central lo que debe llevar a una transformación de la red externa, reestructuración de la red interna de la sede, la dotación de internet a la institución, la actualización de toda la información de los centros penitenciarios y actualización de los equipos.

Al igual que el SIGEP, este sistema se rige por las leyes jurídicas y penales del gobierno panameño, donde los procesos de administración de la biblioteca y emulación no constituyen aspectos relevantes en la gestión del sistema penitenciario, por tal razón estos procesos no se encuentran incluidos como funcionalidades del sistema.

Sistema Automatizado para el Control del Recluso

El Sistema Automatizado para el Control del Recluso (SACORE) fue desarrollado en Cuba en el año 2002 y se comenzó a utilizar a partir del 2003. Este sistema surge para dar cumplimiento a la Orden 43/99 del Vice-Ministro Primero del Ministerio del Interior de Cuba y con el objetivo de automatizar la gestión de los datos principales del recluso y algunos aspectos de control penal, este es el primer sistema informático que gestiona la información que generan las prisiones cubanas. (4)

El SACORE garantiza respuestas inmediatas a las solicitudes de información de los diferentes centros penitenciarios del país, siendo un sistema que recoge prácticamente la totalidad de la información de los reclusos en todas las especialidades. Entre las principales funcionalidades que brinda están contempladas las siguientes:

- Respuestas inmediatas a las solicitudes de información de los diferentes órganos e instituciones del estado como son: Jefatura del MININT, Ministerio de Justicia, Tribunales, Fiscalías, entre otros.
- Los partes que se emiten son obtenidos de forma automatizada.
- Se recogen datos de medidas disciplinarias, visitas, ocupaciones de objetos no permitidos por lo establecido en el reglamento, requisas, entre otros. (4)

SACORE cuenta con 9 años de explotación y presenta determinados problemas que impide su uso exclusivo en la gestión de los procesos del sistema penitenciario cubano:

- Gran cúmulo de información difícil de procesar.
- Realización de consultas a la BD por personal no especializado.
- Necesidad de generar gráficos para realizar los análisis estadísticos.

Estas dificultades provocan que el análisis estadístico se convierta en un proceso engorroso y en muchas ocasiones lento, propiciando la demora en la entrega a tiempo de los informes, afectando de esta forma la toma de decisiones.

Luego de comprobar que el SACORE no lograba abarcar todos los procesos que se ejecutaban en las prisiones, la DEP se dio a la tarea de desarrollar otras dos aplicaciones, surgiendo así el Sistema Automatizado de Incidencias de la Dirección de Establecimientos Penitenciarios (SAIDEP), que gestiona el control y seguimiento de las incidencias o hechos extraordinarios que alteran el orden interior y afectan la seguridad de los establecimientos penitenciarios y el Sistema Automatizado de Capacidades de la Dirección de Establecimientos Penitenciarios (SACDEP), que 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 el SACORE muchos datos son manejados y conservados en formato duro, o a través de herramientas informáticas, como SAIDEP y SACDEP, que no ofrecen grandes facilidades para su gestión. No se puede recurrir a ninguno de estos 3 sistemas para dar solución a la problemática planteada debido a que no poseen las funcionalidades que dan cumplimiento a la gestión de los procesos de administración de la biblioteca y emulación.

Análisis general

Después de haber analizado algunos de las soluciones informáticas de los sistemas penitenciarios nacionales e internacionales, se puede concluir de que dichos sistemas no cumplen con las características que van acorde a las especificaciones del SIDEPE. Además, algunos de estos sistemas no incluyen los procesos de administración de biblioteca ni de emulación como parte de su solución y en otros casos no incluyen todas las funcionalidades que requiere el sistema penitenciario cubano. Por tal razón, se hace necesario implementar las funcionalidades relacionadas con dichos procesos como parte integradora del SIDEPE.

1.4. Metodología, herramientas, tecnologías y lenguajes

1.4.1. Metodología de desarrollo

El desarrollo de software no es tarea fácil y es difícil de controlar, pero utilizando un proceso o una metodología de desarrollo se obtienen clientes y desarrolladores más satisfechos con el resultado. Una metodología es una guía que permite producir y mantener un producto de software, define una serie de pasos a seguir para obtener un software de calidad. La metodología de desarrollo de software se encarga de elaborar estrategias; están centradas en las personas o los equipos y orientadas hacia la funcionalidad y la entrega. Su objetivo es elevar la calidad del software a través de un mayor control sobre el proceso. (5)

Rational Unified Process

Rational Unified Process (RUP por sus siglas en inglés) es una metodología que fue desarrollada por Rational (IBM) y constituye una enorme base de conocimiento de Ingeniería de Software que guía a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo controlado, mientras se balancean los requisitos del negocio y los riesgos del proyecto.

Lo que define a RUP es que está dirigido por casos de uso, está centrado en la arquitectura y es iterativo e incremental. Es un marco de referencia de procesos que brinda una buena documentación. RUP organiza a los proyectos en términos de fases, las cuales incluyen diferentes flujos de trabajo (disciplinas) y fases.

Las 4 fases el desarrollo del software que define RUP son:

- Inicio: el objetivo es determinar la visión del proyecto y definir lo que se desea realizar.
- Elaboración: etapa en la que se determina la arquitectura óptima del proyecto.
- Construcción: se obtiene la capacidad operacional inicial.
- Transición: obtener el producto acabado. (6)

En la figura 1 se muestran las fases y flujos de trabajo (disciplinas) que propone RUP como ciclo de vida de un proyecto.

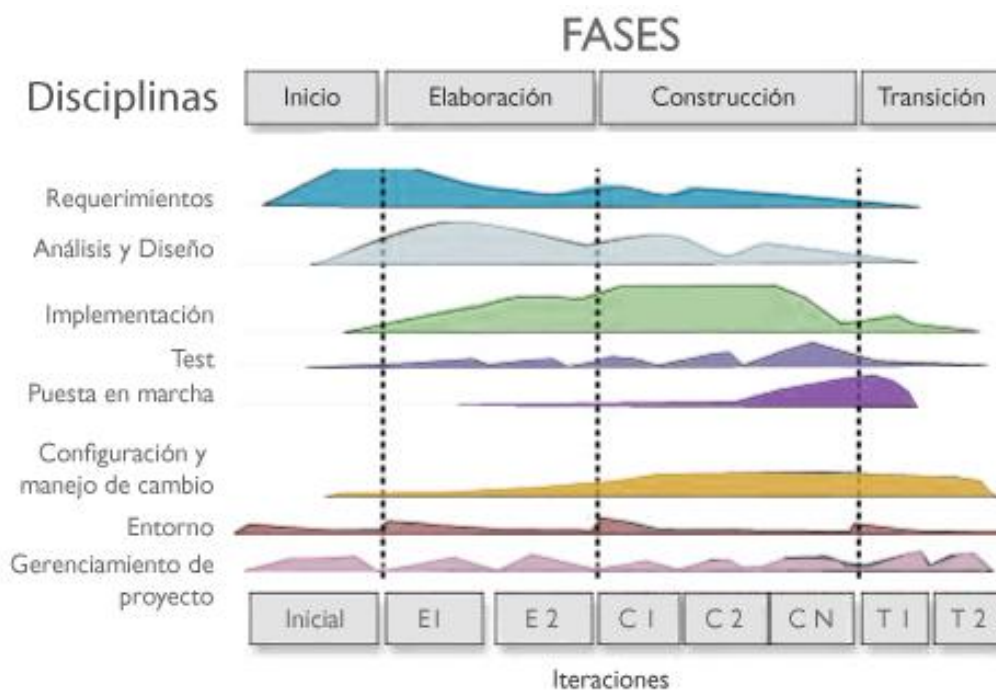


Figura 1 Fases y flujos de trabajo de RUP

SIDEP es una aplicación de gran envergadura para el país, es complejo y para su desarrollo necesita atravesar por todas las fases y los flujos de trabajo que establece RUP como parte del desarrollo de software. El objetivo principal de usar RUP es garantizar el éxito del desarrollo de los módulos haciendo uso del ciclo de vida de un proyecto que esta describe.

1.4.2. Herramientas CASE

Las herramientas de desarrollo de software llamadas Herramientas de Ingeniería de Software Asistida por Computación (*Computer Aided Software Engineering*, CASE por sus siglas en inglés) son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y programadores, durante todos los pasos del ciclo de vida de desarrollo de un software. (7)

Visual Paradigm Suite 5.0

Visual Paradigm es una herramienta CASE multiplataforma diseñada para la ayuda al desarrollo de software, porque soporta el ciclo de vida completo. Su diseño es centrado en casos de uso, y ayuda a una rápida construcción de aplicaciones de calidad a un menor coste. Visual Paradigm permite dibujar diagramas de casos de uso, generar código inverso y código desde diagramas en lenguajes como Java y PHP (*Hypertext Preprocessor*). (7)

Para llevar a cabo las disciplinas del desarrollo de software incluidas en el presente trabajo, Diseño e Implementación, se hace uso de la herramienta Visual Paradigm Suite 5.0, por las funcionalidades que brinda en la modelación de los diferentes diagramas, como por ejemplo: diagrama de clases del diseño, diagrama de colaboración, diagrama de componentes y modelo de despliegue. Además, esta herramienta, a pesar de ser privativa, puede ser utilizada en la universidad ya que se cuenta con la licencia para su uso.

Embarcadero E/R Studio 8.0

Es una poderosa herramienta CASE, para efectuar el modelado lógico y físico de bases de datos Relacionales y Multidimensionales. Entre otras características, ER/Studio permite hacer una separación entre los modelos lógicos y físicos, pero manteniendo una total integración entre ellos, así como con la base de datos. Es posible generar múltiples modelos físicos asociados a un mismo modelo lógico, para diferentes plataformas. (8)

Entre sus ventajas se tienen las siguientes:

- Diseño físico y lógico separados.
- Transformación automática de un diseño lógico a un diseño físico para una plataforma específica de BD.
- Extensa validación de los modelos, lo que hace posible validar ampliamente la calidad e integridad, tanto del diseño lógico como del diseño físico del modelo.

Se decide utilizar esta herramienta CASE para el modelado de las entidades de la BD de los módulos Administración de biblioteca y Emulación, ya que es líder en el diseño y construcción de BD a nivel físico y lógico, además permite comparar el diseño con la estructura real física de las tablas.

1.4.3. Herramientas de desarrollo

SpringSource Tools Suite 2.5

SpringSource Tools Suite 2.5 (STS por sus siglas en inglés) es un Marco de Desarrollo Integrado (*Integrated Development Enterprise*, IDE por sus siglas en inglés) que está basado en Eclipse, pero que también reúne muchas herramientas y asistentes, cuyo objetivo es el de facilitar y agilizar el desarrollo de aplicaciones las cuales utilicen tecnologías como Spring Framework y Spring Web Flow. STS es una herramienta admirable ya que ayuda a manejar de manera muy potente los proyectos Spring, con multitud de asistentes, editores, opciones y configuraciones. STS trae soporte para Groovy y Grails, pues este IDE permite ejecutar aplicaciones Grails en tcServer, lanzar comandos Grails, gestiona distintas versiones de Grails, gestiona dependencias, posee un soporte de depuración mejorado para los proyectos Groovy, mejoras en el tipo de inferencia y soluciones rápidas en el editor de Groovy.

Se decide hacer uso de STS para el desarrollo de la solución ya que es un IDE de código abierto multiplataforma el cual soporta una amplia gama de lenguajes de programación, por lo que constituye una buena opción de trabajo. Finalmente, se puede afirmar que la principal razón para utilizarlo es su característica de integración con el marco de trabajo de aplicaciones web definido por el SIDEPE, Grails.

Apache Tomcat 6.0

Apache Tomcat 6.0 es un servidor web y funciona como un contenedor de servlets. Es la implementación de referencia de las especificaciones de servlets 2.5 y de Java Server Pages (JSP) 2.1, especificaciones para Java Community Process, usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que Apache Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java. (9)

Apache Tomcat es usado en numerosas aplicaciones web críticas y de gran escala en diversas industrias y organizaciones que se referencian en su sitio oficial.

Se decide utilizar Apache Tomcat, además de los argumentos dados anteriormente, debido a que viene embebido en el marco de trabajo de desarrollo web seleccionado por el SIDEPE, Grails.

1.4.4. Gestor de Base de Datos

Oracle Database 11g

Oracle Database 11g es un Sistema de Gestión de Base de Datos Objeto-Relacional (*Object-Relational Data Base Management System*, ORDBMS por sus siglas en inglés), desarrollado por Oracle Corporation y basado en la tecnología cliente/servidor. Es un manejador de BD relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información. Está considerado como uno de los sistemas de BD más completos, destacando entre sus principales fortalezas: el soporte de transacciones, la estabilidad y el soporte multiplataforma.

Oracle Database11g es la primera base de datos del mundo en incluir funcionalidades que permiten hacer pruebas de cambios en aplicaciones simulando las cargas reales generadas por los usuarios en los entornos de producción. (10) Actualmente Oracle proporciona nuevas funcionalidades y facilidades, las cuales van dirigidas a lograr mayor rendimiento, escalabilidad, fiabilidad y seguridad en el manejo y almacenamiento de datos.

Se decide utilizar Oracle 11g como sistema gestor para la BD de los módulos Administración de biblioteca y Emulación ya que ayuda a administrar y almacenar grandes volúmenes de datos. Se puede afirmar que constituye una herramienta de administración gráfica cómoda de utilizar y que permite el análisis de la información. Además, la elección de Oracle 11g se basa en la idea de que es ORDBMS robusto y capaz de garantizar la seguridad e integridad de los datos.

1.4.5. Tecnologías

Grails 1.3.7

Grails es un framework de desarrollo para aplicaciones web, dentro de la plataforma Java, con base en otros proyectos *open-source*¹ como Spring, Hibernate y SiteMesh, que utiliza el lenguaje de programación Groovy. El objetivo de Grails es brindar al usuario un entorno de alta productividad, extensible y fácil de utilizar, ofreciendo el balance adecuado entre consistencia y funcionalidades potentes.

Grails está construido sobre cinco fuertes pilares:

- Groovy: para la creación de propiedades y métodos dinámicos en los objetos de la aplicación.

¹ Código abierto

- Spring: para los flujos de trabajo e inyección de dependencia.
- Hibernate: para la persistencia.
- SiteMesh: para la composición de la vista.
- Ant: para la gestión del proceso de desarrollo. (11)

Además, desde el punto de vista del diseño, combina dos principios fundamentales:

- Convención mejor que configuración: Se basa en la idea de eliminar los archivos de configuración, utilizados generalmente para bases de datos y mapeo de peticiones, siendo reemplazados por convenciones pre-establecidas, tomadas directamente desde el código fuente.
- No te repitas (*Don't Repeat Yourself*, DRY por sus siglas en inglés): este principio alienta al desarrollador a implementar estructuras reutilizables, de forma de minimizar la repetición de código. (12)

Se decide utilizar Grails como framework de desarrollo porque permite a los desarrolladores Java trabajar sobre un ambiente confiable, familiar y robusto, sin necesidad de utilizar extensos archivos de configuración y con la posibilidad de integrar sistemas existentes, logrando así un proceso de desarrollo ágil y productivo.

Dojo Toolkit 1.5

Dojo Toolkit es un framework Javascript que permite el desarrollo de aplicaciones web enriquecidas en el cliente y AJAX². Es popular porque está integrada en numerosos IDEs y otros framework para desarrollo de webs. Contiene un sistema de empaquetado inteligente, los efectos de Interfaz de Usuario (*User Interface*, UI por sus siglas en inglés), drag and drop³ APIs⁴, widget⁵ APIs, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX. (13)

Es conocido como "la navaja suiza del ejército de las bibliotecas JavaScript" y proporciona una gama más amplia de opciones en una sola biblioteca JavaScript y es compatible con navegadores antiguos.

² *Javascript Asíncrono And XML*: es una técnica de desarrollo web para crear aplicaciones interactivas.

³ Filosofía "Agarrar y soltar" aplicable a los componentes

⁴ *Application Programming Interface*, Interfaz de Programación de Aplicaciones: es un conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación.

⁵ Identificador de componente

Se define como marco de trabajo de presentación por sus estilos visuales y su implementación de AJAX.

1.4.6. Plataforma de desarrollo

Java 2 Enterprise Edition

Java 2 Enterprise Edition (J2EE por sus siglas en inglés) es una plataforma de programación que forma parte de la Plataforma Java, la cual es utilizada para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Es una plataforma que se utiliza para el desarrollo de aplicaciones empresariales.

J2EE añade el soporte completo para componentes Enterprise Java Beans, el API Java Servlets y la tecnología Java Server Pages. El estándar J2EE incluye todas las especificaciones y pruebas de conformidad que permiten la portabilidad de las aplicaciones a través de la amplia gama de sistemas empresariales compatibles con J2EE

Se decide utilizar J2EE como plataforma de desarrollo ya que constituye una propuesta atractiva, interesante y de vanguardia que responde a la demanda actual para el desarrollo de software bajo el concepto de arquitectura en capas.

1.4.7. Lenguajes

Lenguaje Unificado para la Construcción de Modelos

El Lenguaje Unificado de Modelado (*Unified Model Language*, UML por sus siglas en inglés) se define como un "lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software". Es un sistema notacional (que, entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. (14)

UML cuenta con varios tipos de modelos, los cuales muestran diferentes aspectos de las entidades representadas. Uno de los objetivos principales de la creación de UML fue posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. (15)

Se decide utilizar UML como lenguaje de modelado ya que facilita a los desarrolladores la visualización de los resultados del trabajo realizado en esquemas o diagramas estandarizados y trabaja correctamente con la mayoría de los procesos de desarrollo existentes. Este es el lenguaje que usa el Visual Paradigm, que es la herramienta seleccionada para la modelación del sistema.

Groovy

Groovy es un lenguaje de programación ágil, dinámico, orientado a objetos y de código abierto.

Groovy tiene la posibilidad de compilarse a bytecode y ejecutarse en el entorno de la Máquina Virtual de Java (*Java Virtual Machine*, JVM por sus siglas en inglés), con lo que es posible utilizar prácticamente cualquier API en Java. El hecho de que se pueda utilizar el API de Java, permite a los desarrolladores Java aprender el lenguaje de forma rápida y la curva de aprendizaje se reduce bastante en comparación con otros lenguajes que se pueden ejecutar en la JVM. (16)

Las características que resaltan de Groovy son:

- Cuenta con la potente base de Java y algunas características de lenguajes de programación tan productivos como Python, Ruby o Smalltalk.
- El código generado es fácil de leer y mantener.
- Permite la creación de lenguajes DSL⁶.
- Puede mezclarse con aplicaciones Java al permitir la compilación en *bytecode*.
- Tipado estático y dinámico.
- Existencia de *closures*⁷. (16)

Se decide usar Groovy como lenguaje de programación ya que es el lenguaje que utiliza el marco de trabajo elegido para el desarrollo de los módulos Administración de biblioteca y Emulación, Grails. Este lenguaje tiene una sintaxis muy parecida a Java; la mayor parte del código escrito en Java es totalmente válido en Groovy por lo que este lenguaje es de muy fácil adopción para programadores Java. Además de las características antes dichas, Groovy proporciona otras facilidades que lo convierten en una mejor opción, como por ejemplo: incluye mejoras sintácticas con relación a la facilidad de manejos de objetos mediante nuevas expresiones y sintaxis, distintas formas de declaración de literales, control de flujo avanzado mediante nuevas estructuras, y nuevos tipos de datos, con operaciones y expresiones específicas.

1.5. Conclusiones parciales

En este capítulo se hizo un análisis del término de sistema penitenciario y los elementos descriptivos de los procesos de administración de biblioteca y emulación. Se realizó un estudio de las tendencias nacionales e internacionales de los sistemas penitenciarios. Además, se analizaron los aspectos

⁶ *Domain Specific Language*, Lenguaje Específico de Dominio

⁷ Básicamente es un trozo de código empaquetado como un objeto y definido entre llaves. Actúa como un método, al cual se le pueden pasar parámetros y pueden devolver valores

relacionados con el proceso de desarrollo de software que serán necesarios para incluir las funcionalidades de administración de biblioteca y emulación en el SIDEPE.

Después de tener todos estos elementos se arriba a las siguientes conclusiones:

- El estudio de los elementos característicos de administración de biblioteca y emulación en el sistema penitenciario cubano permitió tener un mejor entendimiento de estos procesos y lograr una solución adecuada al problema propuesta en este trabajo.
- El análisis de las soluciones informáticas nacionales e internacionales permitió determinar que las mismas no satisfacen las necesidades del SIDEPE, ya que no se corresponden con las características adaptables a las necesidades de los procesos administración de biblioteca y emulación del sistema penitenciario cubano, por lo que se hace necesario desarrollar una aplicación para ello.
- La metodología, herramientas, tecnologías y lenguajes que serán utilizados facilitarán el trabajo a realizar en cada una de las disciplinas que serán desarrolladas para darle solución al problema propuesto de la manera más óptima posible de acuerdo con las especificaciones del SIDEPE.

CAPÍTULO 2. DISEÑO DEL SISTEMA

2.1. Introducción

En el presente capítulo se hace referencia a elementos propios de los procesos de administración de biblioteca y emulación, dígame las reglas del negocio y los requisitos funcionales del sistema, para tener una idea del propósito que se quiere lograr con la solución del presente trabajo. Se explica cómo está estructurada la arquitectura de la aplicación que se implementará. Se analizan los aspectos relacionados con el diseño de los módulos Administración de Biblioteca y Emulación. Además, se expone una representación gráfica de los diagramas de clases del diseño y colaboración, así como el diseño de la BD para la persistencia de la información.

2.2. Reglas del negocio

Para lograr el funcionamiento eficiente de una organización deben existir determinadas reglas de negocio. Estas reglas permiten establecer un control en los procesos que se desarrollan, asegurando que las actividades que se llevan a cabo cumplan con las condiciones y políticas que impone la organización. A continuación se muestran las reglas del negocio correspondientes a los procesos de administración de biblioteca y emulación.

Administración de biblioteca

No	Nombre	Descripción
1.	Indisciplinas por las cuales no se puede realizar un préstamo de obra literaria a un interno.	Las indisciplinas pueden ser: ✓ Maltrato de material literario. ✓ Pérdida de material literario.
2.	Categoría del lector	Un interno por defecto tiene como categoría del lector "Sin categoría".
3.	Categoría del lector	Un interno obtiene la categoría de "Lector asiduo" cuando lleva más de 6 meses solicitando libros, al menos uno por mes.

4.	El contenido de las obras literarias mayormente debe abordar temáticas que contribuyan a influir en el proceso educativo.	Las temáticas son: <ul style="list-style-type: none"> ✓ Instrucción general del interno. ✓ La educación en valores. ✓ La formación de oficios.
5.	Tiempo límite para los préstamos	El tiempo máximo para un préstamo es de 15 días.

Tabla 1 Reglas del negocio del módulo Administración de Biblioteca

Emulación

No	Nombre	Descripción
1.	Categoría de Destacado	El interno podrá optar por la condición de Destacado, siempre y cuando haya acumulado más de 90 en el período emulativo.
2.	Resta de puntos por indisciplinas	A los internos que cometan violaciones del régimen y la disciplina, se le restan puntos, según su categoría: <ul style="list-style-type: none"> ✓ Por cada indisciplina grave: -15 puntos ✓ Por cada indisciplina menos grave: -10 puntos ✓ Por cada indisciplina leve: -5 puntos
3.	Sin categoría de destacado	El que haya incurrido en violaciones del régimen y la disciplina de carácter grave, no podrá ser seleccionado Destacado en el período emulativo.
4.	Puntos para el TSU	El que no esté incorporado al TSU, se valora su participación en la limpieza y embellecimiento del dormitorio u otras áreas del interior penal y otras

		actividades.
5.	Período de la emulación	Los cierres trimestrales de la emulación colectiva, a nivel de lugar de internamiento, se realizan en base a los resultados obtenidos en la emulación diaria de los colectivos, la cual tendrá cierre semanal (viernes de cada semana) y mensual (del 25 al 30 de cada mes). La emulación se hace trimestral.
6.	Tipos de estímulos	El Jefe del lugar de internamiento, a propuesta del Jefe de Tratamiento Educativo, puede otorgar al interno destacado, algunos de los siguientes estímulos: ✓ Visita familiar. ✓ Visita conyugal.
7.	Categoría Destacado para el Colectivo	Resultan destacados en la emulación colectiva los colectivos que el 90% de los internos que los integran obtengan más de 90 puntos en la emulación individual.
8.	Evaluación del interno	Se debe mostrar por cada interno si tuvo incidencias leves o menos graves, especificar la incidencia y la cantidad de puntos que se le quitan.

Tabla 2 Reglas del negocio del módulo Emulación

2.3. Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Los requisitos funcionales se mantienen invariables sin importar con que propiedades o cualidades se relacionen. (17) Se puede afirmar que los requisitos funcionales constituyen un elemento fundamental a la hora de desarrollar cualquier sistema.

Teniendo en cuenta los procesos y las reglas del negocio que se identificaron anteriormente, a continuación se definen los requisitos funcionales correspondientes a los módulos Administración de biblioteca y Emulación.

Administración de biblioteca

Numeración	Requisitos funcionales
3.8.1	Registrar obra literaria
3.8.2	Modificar obra literaria
3.8.3	Eliminar obra literaria
3.8.4	Registrar préstamo
3.8.5	Eliminar préstamo
3.8.6	Consultar préstamos vencidos
3.8.7	Consultar lector asiduo
3.8.8	Registrar actividad de la biblioteca
3.8.9	Modificar actividad de la biblioteca
3.8.10	Consultar actividad de la biblioteca
3.8.11	Registrar resultado de actividad de la biblioteca
3.8.12	Registrar asistencia a la biblioteca
3.8.13	Consultar asistencia a la biblioteca
3.8.14	Registrar medida disciplinaria
3.8.15	Modificar medida disciplinaria
3.8.16	Eliminar medida disciplinaria

Tabla 3 Requisitos funcionales módulo Administración de biblioteca

Emulación

Numeración	Requisitos funcionales
3.5.1	Registrar puntuación emulativa
3.5.2	Consultar puntuación individual
3.5.3	Consultar puntuación del colectivo

3.5.4	Registrar propuesta de estímulos
3.5.5	Consultar estímulos otorgados a los internos
3.5.6	Administrar puntuación emulativa
3.5.7	Consultar puntuación colectiva
3.5.8	Configurar emulación colectiva

Tabla 4 Requisitos funcionales módulo Emulación

2.4. Diagrama de casos de uso del sistema

Los casos de uso constituyen una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. (5)

En la figura 2 se muestra el diagrama de casos de uso correspondiente al módulo Administración de biblioteca, el cual muestra 10 casos de uso identificados a partir de los requisitos funcionales mencionados anteriormente.

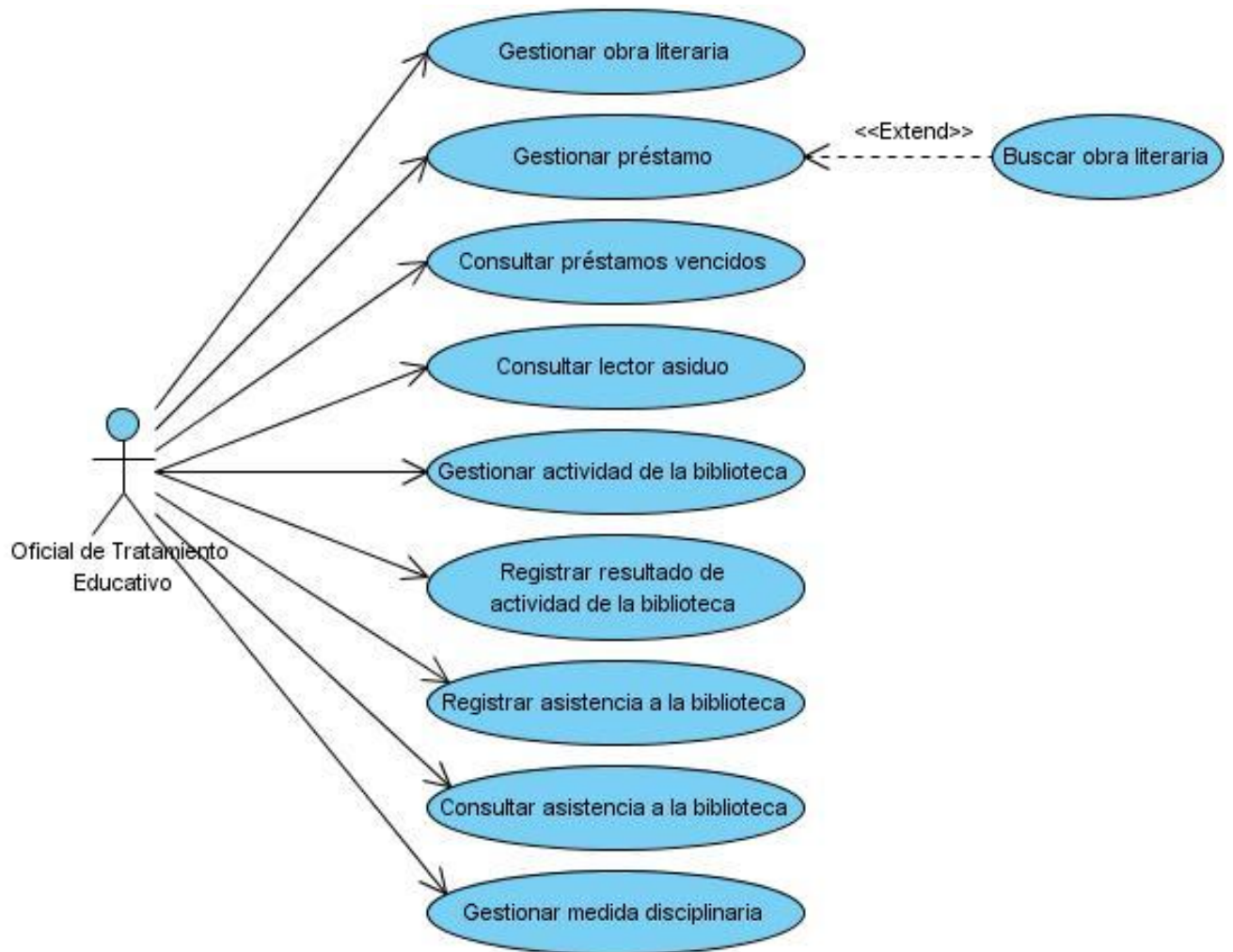


Figura 2 Diagrama de casos de uso del sistema módulo Administración de biblioteca

El Oficial de Tratamiento Educativo (OTE) es el máximo responsable de gestionar y controlar todas las actividades relacionadas con el proceso educativo de los internos para su reinserción a la sociedad, entre ellas las de la biblioteca. Como se puede observar en la diagrama de casos de uso del sistema del módulo Administración de la biblioteca el único actor involucrado es el OTE, el cual es el encargado de inicializar todos los casos de uso de dicho módulo.

A continuación se muestra el diagrama de casos de uso del sistema correspondiente al módulo Emulación (Figura 3), donde están representados 7 casos de uso a partir de los requisitos funcionales.

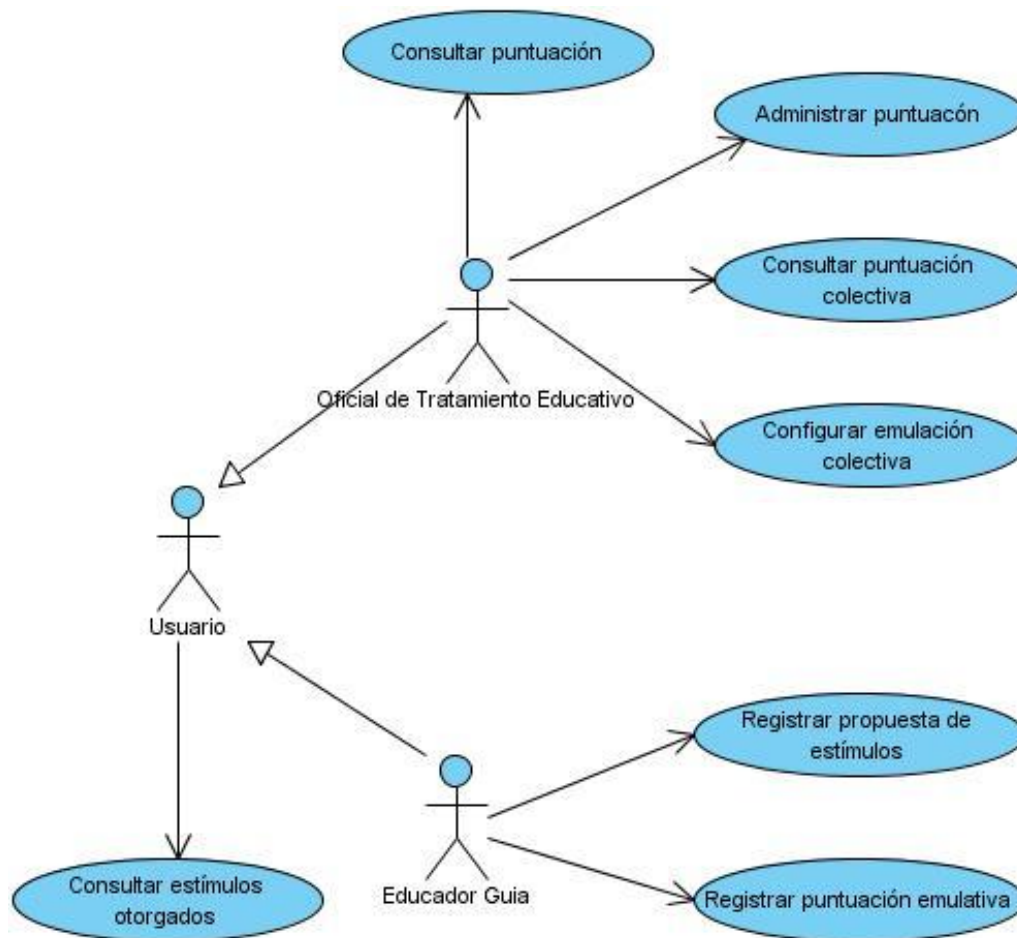


Figura 3 Diagrama de casos de uso del sistema módulo Emulación

El Educador Guía (EG) es el responsable del control de los internos de un colectivo, incluyendo todos los aspectos de convivencia, superación y disciplina. Uno de estos elementos es la emulación, donde el EG es el encargado de registrar la puntuación emulativa de cada interno de su colectivo. Además, es el responsable de registrar la propuesta de estímulos a los internos que se elijan como destacados de la emulación.

El OTE es el encargado de gestionar la emulación en el centro penitenciario, la puntuación que registrará la emulación, así como la configuración del porcentaje que determina elección de los colectivos destacados. Además puede consultar la puntuación obtenida por un interno o un colectivo determinado.

Ambos actores, EG y OTE, podrán consultar los estímulos que hayan sido otorgados en el colectivo y en el centro penitenciario respectivamente.

2.5. Principales funcionalidades

Para tener un mejor entendimiento de las funcionalidades correspondientes a los módulos Administración de biblioteca y Emulación, teniendo en cuenta los diagramas de casos de uso de cada uno de estos módulos representados anteriormente, a continuación se describen las funcionalidades que se implementarán en la solución del presente trabajo.

No.	Nombre	Descripción
1.	Gestionar obra literaria	Tiene como objetivo registrar, actualizar o eliminar una obra literaria de la biblioteca del centro penitenciario. El caso de uso (CU) se inicia cuando el OTE selecciona la opción "Obra Literaria", el sistema brinda la posibilidad de registrar, eliminar o modificar una obra literaria. El OTE introduce los datos de la obra literaria y el sistema registra dicha información.
2.	Buscar obra literaria	Tiene como objetivo la búsqueda de una obra literaria existente en la biblioteca. El CU se inicia cuando el sistema muestra los campos para buscar una obra literaria. El OTE introduce los criterios de búsqueda y el sistema muestra los resultados correspondientes.
3.	Gestionar préstamo	Tiene como objetivo registrar o eliminar un préstamo de obra literaria. El CU se inicia cuando el OTE selecciona la opción "Préstamo", el sistema brinda la posibilidad de registrar un nuevo préstamo o eliminar uno existente. El OE introduce los datos correspondientes y el sistema registra la información.
4.	Consultar préstamos vencidos	Tiene como objetivo consultar los préstamos vencidos en la biblioteca. El CU se inicia cuando el

		OTE selecciona la opción "Préstamos vencidos". El sistema permite mostrar los préstamos que vencen el día de la consulta o los préstamos vencidos en días anteriores. El sistema muestra el listado de los préstamos vencidos.
5.	Consultar lector asiduo	Tiene como objetivo mantener identificados a los internos con categoría de "Lector asiduo" en el centro penitenciario. El CU se inicia cuando el OTE selecciona la opción de "Consultar lector asiduo" y el sistema muestra los internos que son lectores asiduos.
6.	Gestionar actividad de la biblioteca	Tiene como objetivo registrar, actualizar o eliminar una actividad realizada en la biblioteca del centro penitenciario. El CU inicia cuando el OTE selecciona la opción "Gestionar actividad de la biblioteca". El sistema permite registrar, actualizar o consultar los datos de las actividades que se desarrollan en la biblioteca.
7.	Registrar resultado de actividad de la biblioteca	Tiene como objetivo registrar los resultados de las actividades de la biblioteca del centro penitenciario. El CU inicia cuando el OTE selecciona la opción "Resultado de actividad de la biblioteca". El sistema permite registrar los datos de las actividades que se desarrollan en la biblioteca.
8.	Registrar asistencia a la biblioteca	Tiene como objetivo llevar el control de los internos que asisten a la biblioteca. El CU se inicia cuando el OTE selecciona la opción "Asistencia a la biblioteca", el sistema muestra los campos para registrar la asistencia, el OTE introduce la asistencia de internos al registro de la biblioteca, dependiendo

		esta de la presencia física del lector o por alguna de las actividades desarrolladas por la biblioteca. El sistema registra dicha información.
9.	Consultar asistencia a la biblioteca	Tiene como objetivo mantener el control de la asistencia de los internos a la biblioteca. El CU se inicia cuando el OTE selecciona la opción "Consultar asistencia". El sistema permite buscar un interno para visualizar la asistencia que ha tenido a la biblioteca.
10.	Gestionar medida disciplinaria	Tiene como objetivo registrar, actualizar o eliminar las medidas disciplinarias a los internos que tengan indisciplinas en la biblioteca. El CU se inicia cuando el OTE selecciona la opción "Medida disciplinaria en la biblioteca" para registrar, modificar o eliminar una medida disciplinaria a un usuario de la biblioteca que haya incurrido en un indisciplina. El OTE introduce los datos y el sistema registra la información

Tabla 5 Funcionalidades del módulo Administración de biblioteca

No.	Nombre	Descripción
1.	Registrar puntuación emulativa	Tiene como objetivo registrar la puntuación alcanzada por un interno en las distintas actividades según su participación y comportamiento en las mismas, así como las incidencias que tenga y su influencia en dicha puntuación. El CU inicia cuando el EG selecciona la opción "Registrar puntuación" para registrar la puntuación de los internos en los diferentes puntos a evaluar en la emulación. El EG introduce los datos y el sistema registra la información.

2.	Consultar puntuación	Tiene como objetivo mostrar la puntuación que reciben los internos en los distintos parámetros de la emulación en los colectivos del centro penitenciarios. El CU se inicia cuando el OTE selecciona la opción "Puntuación individual" para ver los puntos que han los internos en la emulación en los diferentes colectivos, así como la categoría con la cual se le ha calificado.
3.	Registrar propuesta de estímulos	Tiene como objetivo registrar los estímulos a los internos con categoría de destacados. El CU se inicia cuando el EG selecciona la opción "Registrar propuesta de estímulo" para registrar la propuesta de estímulos a los internos de su colectivo para la posterior evaluación por parte el Jefe de Unidad. El EG introduce los datos y el sistema registra la información.
4.	Consultar estímulos otorgados	Tiene como objetivo mostrar los estímulos que han sido otorgados a los internos destacados y que hayan sido registrados en el sistema. Este CU se inicia cuando el Usuario selecciona la opción "Consultar estímulos otorgados" para ver los estímulos otorgados a los internos destacados. El Usuario selecciona el colectivo y el sistema muestra los estímulos otorgados a los internos de ese colectivo.
5.	Administrar puntuación	Tiene como objetivo definir la puntuación de la emulación desglosada en parámetros según sea el caso. El CU inicia cuando el OTE selecciona la opción "Administrar puntuación" para administrar la puntuación relacionada con la emulación de los

		internos del centro penitenciario. El OTE introduce los datos y el sistema registra la información.
6.	Consultar puntuación colectiva	Tiene como objetivo mostrar la puntuación de los colectivos y la correspondiente categoría en la emulación colectiva. El CU se inicia cuando el OTE selecciona la opción "Puntuación colectiva" para ver los distintos puntos que ha obtenido un colectivo en la emulación, así como la categoría con la cual se le ha calificado.
7.	Configurar emulación	Tiene como objetivo configurar el porcentaje de internos que deben ser destacados en un colectivo para que este pueda resultar Destacado. El CU se inicia cuando el OTE selecciona la opción "Configurar emulación", para cambiar el porcentaje de internos destacados que debe tener el colectivo para obtener la categoría de Destacado. El OTE introduce el dato y el sistema registra la información

Tabla 6 Funcionalidades del módulo Emulación

2.6. Modelo del diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y los requisitos no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. El modelo de diseño sirve de abstracción a la implementación y se utiliza como una entrada fundamental a las actividades de implementación. (14)

De acuerdo al libro de Roger S. Pressman, Ingeniería de Software. Un enfoque Práctico en su quinta edición, a la hora de diseñar es necesario tener en cuenta una serie de principios que garantizan el correcto funcionamiento del sistema cuando se implementa. A continuación se enumeran algunos de los principios más importantes a tener en cuenta de acuerdo al autor.

- El diseño no deberá inventar nada que ya esté inventado.

- El diseño deberá minimizar la distancia intelectual entre el software y el problema como si de la misma vida real se tratara.
- El diseño deberá estructurarse para admitir cambios.
- El diseño no es escribir código y escribir código no es diseñar.
- El diseño deberá revisarse para minimizar los errores conceptuales. (18)

Cuando estos principios se aplican correctamente, entonces en el diseño creado se evidencian los factores de calidad internos y externos. Los factores externos son aquellas propiedades del software que se observan fácilmente: velocidad, fiabilidad, grado de corrección y usabilidad. Los internos revisten gran importancia para el equipo de trabajo encargado de realizar el proyecto, conduciendo a un diseño de alta calidad.

El modelo de diseño incluye todos los artefactos que se generan en el diseño del sistema, y para que el resultado final sea satisfactorio trabajan en conjunto dos trabajadores: el diseñador del sistema y el arquitecto. Con el desarrollo de este modelo se generan tres de las cinco vistas arquitectónicas del sistema, por lo que no pueden verse separados el diseño y la arquitectura, teniendo en cuenta que el modelo de diseño se basa en la arquitectura que se define para el desarrollo de software.

2.7. Arquitectura del sistema

La arquitectura de software establece los fundamentos para que analistas, diseñadores y desarrolladores trabajen en una línea común que permita alcanzar los objetivos del sistema, cubriendo todas las necesidades. Se necesita una arquitectura robusta, que guíe el proceso de desarrollo, y que defina de manera abstracta los componentes que llevan a cabo alguna tarea, sus interfaces y la comunicación entre ellos. (19)

La arquitectura definida para la construcción del SIDEP es una arquitectura n-capas la cual está compuesta de la siguiente manera: la capa de datos o modelo, la capa de presentación o vista y la capa de control o controlador, en estas tres capas se aplica el patrón Modelo-Vista-Controlador (MVC) de Grails, y una capa de servicios, que es donde normalmente se implementa la lógica de negocio de los casos de uso del sistema.

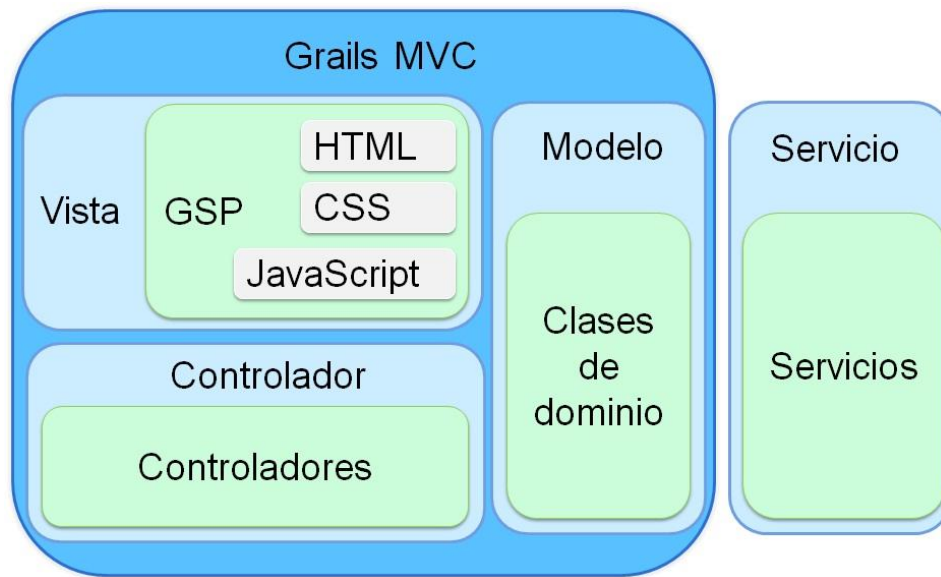


Figura 4 Arquitectura del sistema

A continuación se explican las capas que conforman la arquitectura del sistema:

Capa de datos o modelo

Esta capa está compuesta por las entidades o clases del dominio, en las cuales va a persistir la información de la aplicación. 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, incluyendo las acciones de modificar o registrar. Grails utiliza un gestor de persistencia escrito en Groovy sobre Hibernate conocido por Mapeo Relacional de Objetos Groovy (*Grails Object Relational Mapping*, GORM por sus siglas en inglés) para controlar el ciclo de vida de las entidades.

Una vez definidas las propiedades de las clases del dominio, GORM se encarga de:

- Generar las tablas correspondientes en la BD, incluyendo 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 y genera un valor único secuencial para cada instancia de la misma.
- Restringe los valores que pueden asignarse a los atributos de cada entidad mediante la propiedad *constraints*.

- Grails presenta una técnica llamada *scaffolding*, la cual permite, una vez creada las clases de dominio, que se creen las clases controladoras y las vistas necesarias para realizar las operaciones.

Capa de presentación o vista

Esta capa está formada por plantillas Groovy Server Page (GSP), las cuales son las encargadas de mostrar en el navegador del cliente el contenido que se encuentra en el modelo, mediante código HTML, JavaScript y Hojas de Estilos en Cascada (*Cascading Style Sheets*, CSS por sus siglas en inglés). Se utilizó el framework Dojo Toolkit para la creación de vistas, utilizando la tecnología AJAX.

Capa de control o controlador

Esta capa se comunica con las demás capas. Los controladores son los componentes responsables de recibir órdenes del usuario, gestionar la ejecución de la lógica del negocio y después actualizar la vista para mostrar al usuario el estado en que ha quedado el modelo de datos. Son los responsables de interceptar las peticiones HTTP⁸ del navegador y generar la respuesta correspondiente, ya sea en el propio controlador o delegando el trabajo en una vista.

Capa de servicios

En esta capa se encuentran los servicios, los cuales son componentes que implementan la lógica de negocio de los casos de uso que componen la aplicación. Tienen una estrecha relación con el modelo de datos, porque en él se hacen llamadas a los métodos que brinda GORM para la consulta de los datos. 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. GORM también inyecta una serie de métodos estáticos en las clases persistentes para hacer consultas sobre los datos almacenados en la tabla o tablas correspondientes.

2.8. Patrones del diseño

Los patrones permiten reutilizar la esencia de la solución de un problema al enfrentar nuevos problemas similares. Es decir, los patrones constituyen una especie de mecanismo de registro y concentración de experiencia, haciendo el trabajo con los sistemas más fácil y mucho más simple su mantenimiento. (2) Específicamente los patrones del diseño guían el proceso de construcción del software en la disciplina de Diseño, ya que constituyen una fuente de información descriptiva para la

⁸ *Hyper Text Transfer Protocol*, Protocolo de Transferencia de Hipertexto

resolución de problemas que pueden surgir en la elaboración de los diagramas correspondientes a dicha disciplina.

Los patrones de diseño reconocidos en el desarrollo de software se agrupan de la siguiente manera: Patrones de Software para Asignación de Responsabilidades (*General Responsibility Assignment Software Patterns*, GRASP por sus siglas en inglés) y patrones de la “Banda de los cuatro” (*Band of Four*, GOF por sus siglas en inglés). A continuación se describen y explican los patrones de diseño utilizados en el desarrollo de la solución.

2.8.1. Patrones GRASP

Los patrones GRASP describen un conjunto de principios fundamentales para la asignación de responsabilidades. Dentro de este grupo se encuentran los siguientes patrones:

- Experto.
- Creador.
- Bajo acoplamiento.
- Alta cohesión.
- Controlador.

Experto

Asigna una responsabilidad al más competente en información, donde la clase cuenta con la información necesaria para cumplir la responsabilidad. (14) Es el principio básico de asignación de responsabilidades que suele utilizarse en el diseño Orientado a Objetos, y es el más usado para asignar responsabilidades.

Creador

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. (14)

Bajo acoplamiento

Este patrón asigna una responsabilidad para mantener bajo acoplamiento. (14) Este patrón sugiere que las clases no deben estar tan relacionadas entre sí, de manera que en caso de producirse una modificación en alguna de las clases, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

Alta cohesión

Este patrón explica que la información que almacena una clase debe de ser coherente y debe estar relacionada con la clase. Asigna una responsabilidad de modo que la cohesión siga siendo alta. (14)

Controlador

Un Controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. (14) Todas las peticiones web son manejadas por un solo controlador, que es el punto de entrada único de toda la aplicación en un entorno determinado. Dicho controlador utiliza el sistema de enrutamiento para asociar el nombre de una acción con la URL solicitada por el usuario.

2.8.2. Patrones GOF

“Banda de los cuatro” es el nombre con el que se conoce a los autores del libro “Design Patterns”, ellos son: Erich Gamma, Richard Helm, Ralph Jhonson y Jhon Vissides. Este grupo de patrones se divide en tres subgrupos fundamentales que dan una medida del propósito de los patrones que los conforman:

- Patrones de creación.
- Patrones estructurales.
- Patrones de comportamiento.

Los patrones GOF definen una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular.

Singleton

Este patrón garantiza que una sola clase sólo tenga una instancia y proporciona un punto de acceso global a ella. (20) La utilización de este patrón se evidencia en los servicios, los cuales son instanciados por los controladores para utilizar las funcionalidades que los mismos brindan. El acceso a estos es global, lo que significa que todos los controladores acceden a la misma instancia del servicio, accediendo a la misma información contenida en ellos.

2.8.3. Otro patrón de diseño utilizado

Inversión de Control

Inversión de Control (*Inversion of Control*, IoC según sus siglas en inglés) plantea que las dependencias de un componente no deben gestionarse desde el propio componente para que este

solo contenga la lógica necesaria para hacer su trabajo. (9) Cuando se crea un componente en la aplicación, Grails configura a Spring para controlar su ciclo de vida (cuándo se crea, cuántas instancias mantiene vivas a la vez, cómo se destruyen, etc.) y sus dependencias (qué otros componentes necesita para desarrollar su trabajo y cómo seguirlos). El objetivo de esta técnica es mantener los componentes lo más sencillos posible, incluyendo únicamente código que tenga relación con la lógica de negocio, así la aplicación será más fácil de comprender y mantener.

2.9. Diagramas de clases

2.9.1. Descripción de las clases más significativas

A continuación se describen las clases más significativas de los módulos Administración de biblioteca y Emulación, las cuales serán tomadas en cuenta para la posterior realización de los diagramas de clases del diseño.

Módulo Administración de biblioteca

Entidades	Descripción
Obra literaria	Almacena todas las características identificativas del material literario.
Prestamo	Almacena los datos para el préstamo de un material literario.
Actividad de la biblioteca	Almacena todos los datos referentes a las actividades realizadas o por realizar de la biblioteca.
Medida disciplinaria de biblioteca	Almacena los datos de las medidas disciplinarias tomadas por la biblioteca.

Tabla 7 Descripción de entidades del módulo Administración de la biblioteca

Nombre	obraLiteraria.html
Tipo de clase	Interfaz
Descripción	Muestra los datos de la obra literaria.

Atributo	Tipo de dato	Descripción del atributo
autor	VARCHAR2(255 BYTE)	Contiene el nombre del autor de la obra literaria.
titulo	VARCHAR2(255 BYTE)	Contiene el título de la obra literaria.
genero	NUMBER(19,0)	Contiene el género al cual pertenece la obra literaria.
editorial	NUMBER(19,0)	Contiene la editorial que imprime la obra literaria.
edicion	VARCHAR2(255 BYTE)	Contiene el número de edición de la obra literaria.
cant_Ejemplares	NUMBER(10,0)	Contiene la cantidad de ejemplares existentes de la obra literaria.

Tabla 8 Descripción de la clase obraLiteraria.html del módulo Administración de la biblioteca

Nombre	obraLiterariaController	
Tipo de clase	Controladora	
Descripción	Encargada de gestionar las obras literarias de la biblioteca.	
Atributo	Tipo de dato	Descripción del atributo
Obra Literaria	Obra Literaria	Posee los datos de una obra literaria.
obraLiterariaService	Service	Se utiliza para guardar una obra literaria que tiene un interno.
Para cada responsabilidad:		
Nombre	Descripción	

index()	El método muestra el gsp obra literaria.
registrarObraLiteraria	El método registra una obra literaria a la biblioteca.
eliminarObraLiteraria	El método elimina una obra literaria de la biblioteca.
actualizarObraLiteraria	El método actualiza una obra literaria de la biblioteca.

Tabla 9 Descripción de la clase obraLiterariaController del módulo Administración de la biblioteca

Nombre	obraLiterariaService
Tipo de clase	Service
Descripción	Es donde se realiza la lógica de negocio.
Para cada responsabilidad:	
Nombre	Descripción
obtenerObraLiteraria()	El método obtiene la lista de obras literarias.
getItems	El método obtiene una lista de obras literarias con sus datos.
getProps()	El método muestra los nombres de los campos de una tabla.
add()	El método adiciona una obra literaria si sus datos son válidos.
eliminar()	El método elimina una obra literaria determinada.

Tabla 10 Descripción de la clase obraLiterariaService del módulo Administración de la biblioteca

Módulo Emulación

Entidades	Descripción
Categoria	Almacena los datos correspondientes a la categoría del interno alcanzada en la emulación.
Tipo Estimulo	Almacena los tipos de estímulo que reciben los internos alcanzado en

	la emulación.
Propuesta Estimulo	Almacena los datos correspondientes a los estímulos que se le pueden otorgar al interno.
Parametro	Almacena los datos correspondientes a los parámetros de la emulación en un período.
Configuracion emulacion colectiva	Almacena el porcentaje que internos destacados que debe tener un colectivo para ser destacado.

Tabla 11 Descripción de entidades del módulo Emulación

Nombre	administrarPuntuacion.html	
Tipo de clase	Interfaz	
Descripción	Muestra los datos de los parámetros a medir en la emulación.	
Atributo	Tipo de dato	Descripción del atributo
nombre	VARCHAR2(255 BYTE)	Contiene el nombre del parámetro.
puntos	NUMBER(10,0)	Contiene los puntos que vale el parámetro.

Tabla 12 Descripción de la clase administrarPuntuacion.html del módulo Emulación

Nombre	administrarPuntuacionController	
Tipo de clase	Controladora	
Descripción	Encargada de administrar la puntuación que tomarán los parámetros.	
Atributo	Tipo de dato	Descripción del atributo

Parametro	Parametro	Posee los datos de un parámetro.
aspectoService	Service	Se utiliza para guardar un aspecto que pueda tener un parámetro.
parametroService	Service	Se utiliza para guardar un parámetro.
Para cada responsabilidad:		
Nombre	Descripción	
index()	El método muestra el gsp administrar puntuación.	
registrarPuntuacion	El método registrar la puntuación que va tener cada parámetro.	
eliminarParametro	El método elimina de la lista de parámetros.	
actualizarParametro	El método actualiza de la lista de parámetros.	

Tabla 13 Descripción de la clase administrarPuntuacionController del módulo Emulación

Nombre	parametroService
Tipo de clase	Service
Descripción	Es donde se realiza la lógica de negocio.
Para cada responsabilidad:	
Nombre	Descripción
obtenerParametro()	El método obtiene la lista de parámetros.
getItems	El método obtiene una lista de parámetros con su puntuación.
getProps()	El método muestra los nombres de los campos de una tabla.

add()	El método adiciona un parámetro si sus datos son válidos.
eliminar()	El método elimina un parámetro determinado.

Tabla 14 Descripción de la clase parametroService del módulo Emulación

Nombre	aspectoService
Tipo de clase	Service
Descripción	Es donde se realiza la lógica de negocio.
Para cada responsabilidad:	
Nombre	Descripción
obtenerAspecto()	El método obtiene la lista de aspectos de un parámetro determinado.
getItems	El método obtiene una lista de aspectos con su puntuación.
getProps()	El método muestra los nombres de los campos de una tabla.
add()	El método adiciona un aspecto si sus datos son válidos.
eliminar()	El método elimina aspecto determinado.

Tabla 15 Descripción de la clase aspectoService del módulo Emulación

Nombre	Puntuacion.js
Tipo de clase	JavaScript
Descripción	Se validan los datos introducidos en la clase administrarPuntuacion.html.
Para cada responsabilidad:	
Nombre	Descripción

registrarParametro()	El método registra un parámetro.
registrarAspectos()	El método registra los aspectos pertenecientes al parámetro.
validar()	El método valida los datos registrados.
resetColors()	El método cambia el color del borde de un campo que no sea válido.

Tabla 16 Descripción de la clase puntuacion.js del módulo Emulación

2.9.2. Diagramas de clases del diseño

Los diagramas de clase de diseño describen gráficamente las descripciones de las clases del sistema y sus interfaces, así como las relaciones que existen entre ellas.

Se seleccionaron 2 CU pertenecientes a los módulos Administración de biblioteca y Emulación, a través de los cuales se trabajarán todos los elementos relacionados con las disciplinas del proceso de desarrollo de software incluidas en el alcance del presente trabajo. Los CU de los módulos Administración de biblioteca y Emulación que serán analizados serán los siguientes: Gestionar obra literaria y Administrar puntuación respectivamente.

A continuación se muestran los diagramas de clases del diseño de los CU antes mencionados.

El resto de los diagramas de clases del diseño se encuentran incluidos en los anexos (Anexo 1).

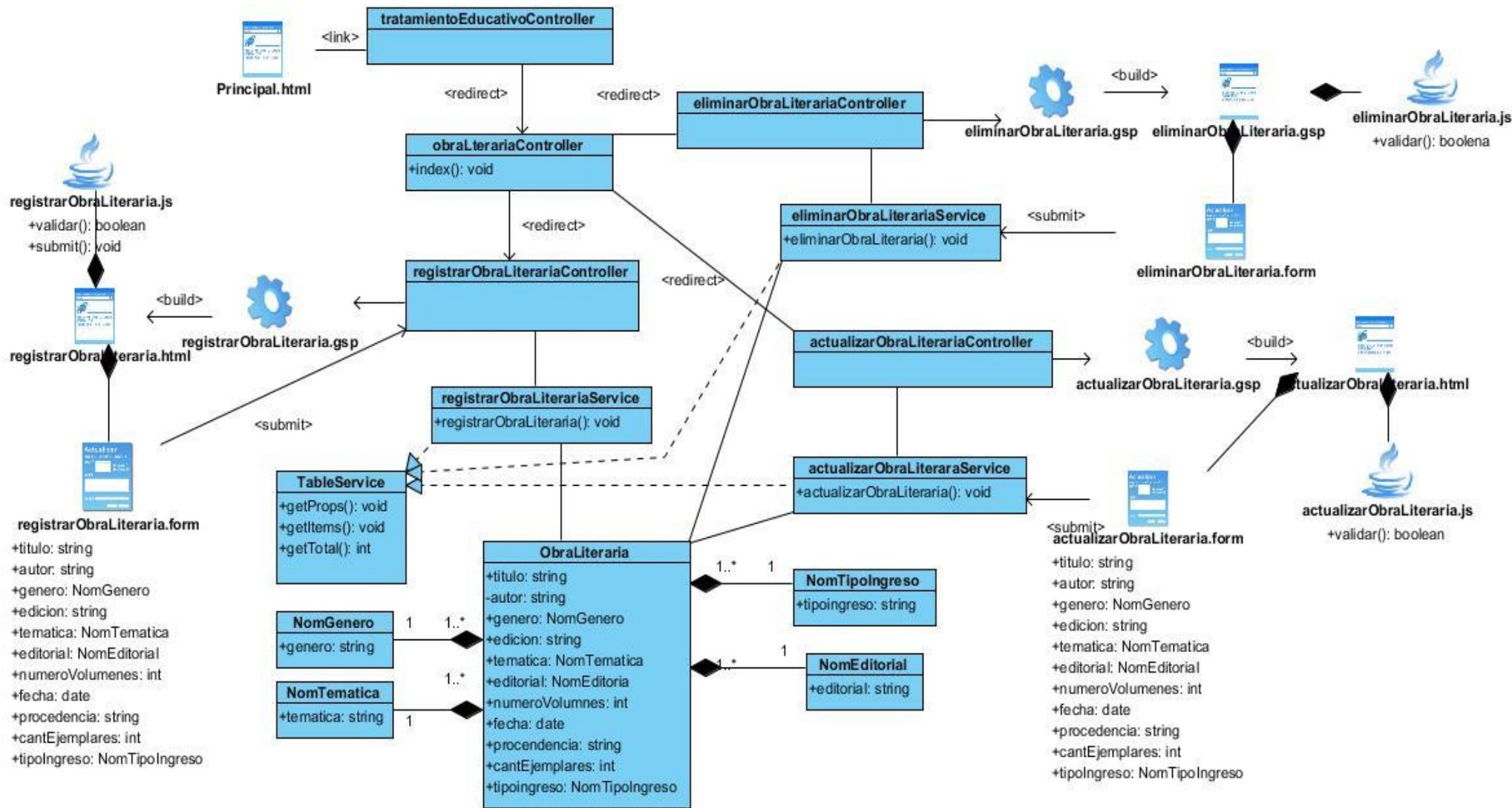


Figura 5 DCD9 Gestionar obra literaria del módulo Administración de biblioteca

⁹ Diagrama de clases del diseño

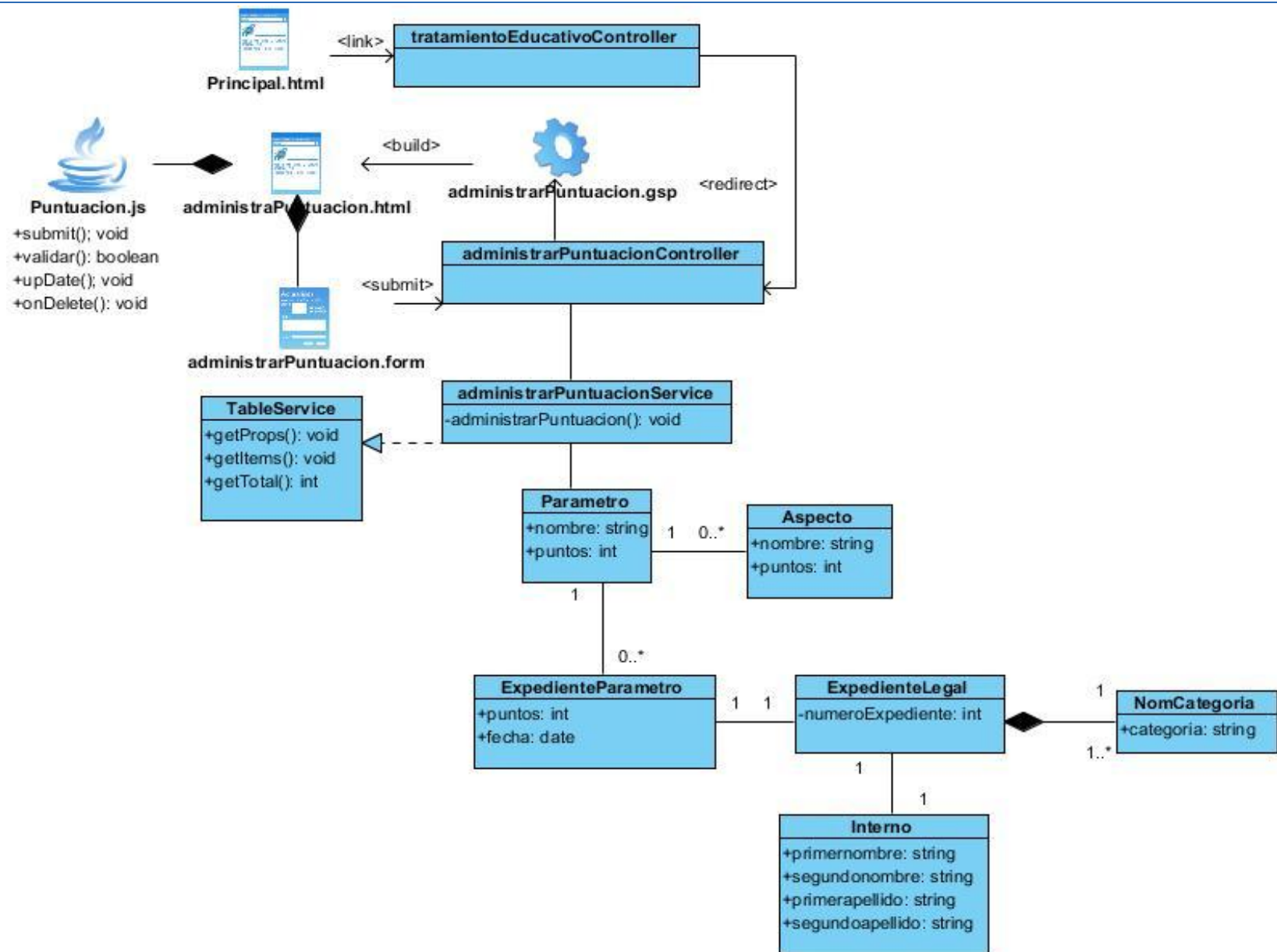


Figura 6 DCD Administrar puntuación del módulo Emulación

2.10. Diagramas de colaboración

Un diagrama de interacción contiene un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos, organizados lógicamente y secuencialmente. Dentro del grupo de los diagramas de interacción se encuentran los diagramas de colaboración, los cuales representan una combinación de información tomada desde el diagrama de clases, diagrama de secuencia y diagrama de casos de uso, describiendo tanto la estructura estática como el comportamiento dinámico de un sistema.

A continuación se muestran los diagramas de colaboración de la funcionalidad "Registrar obra literaria" correspondiente a los CU Gestionar obra literaria y Administrar puntuación de los módulos Administración de biblioteca y Emulación respectivamente.

El resto de los diagramas de colaboración se encuentran incluidos en los anexos (Anexo 2).

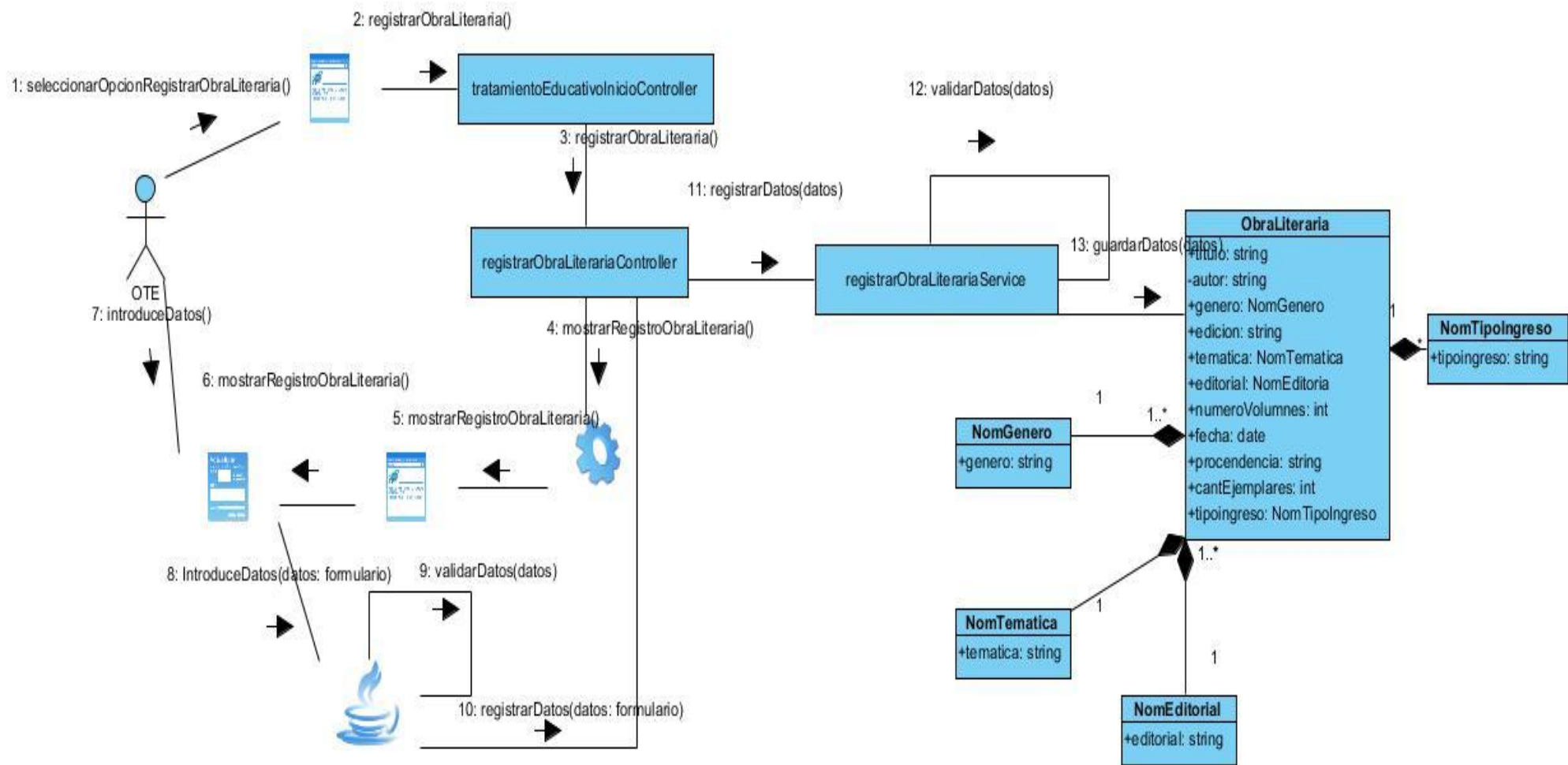


Figura 7 DC¹⁰ Registrar obra literaria (Gestionar obra literaria) del módulo Administración de biblioteca

¹⁰ Diagrama de colaboración

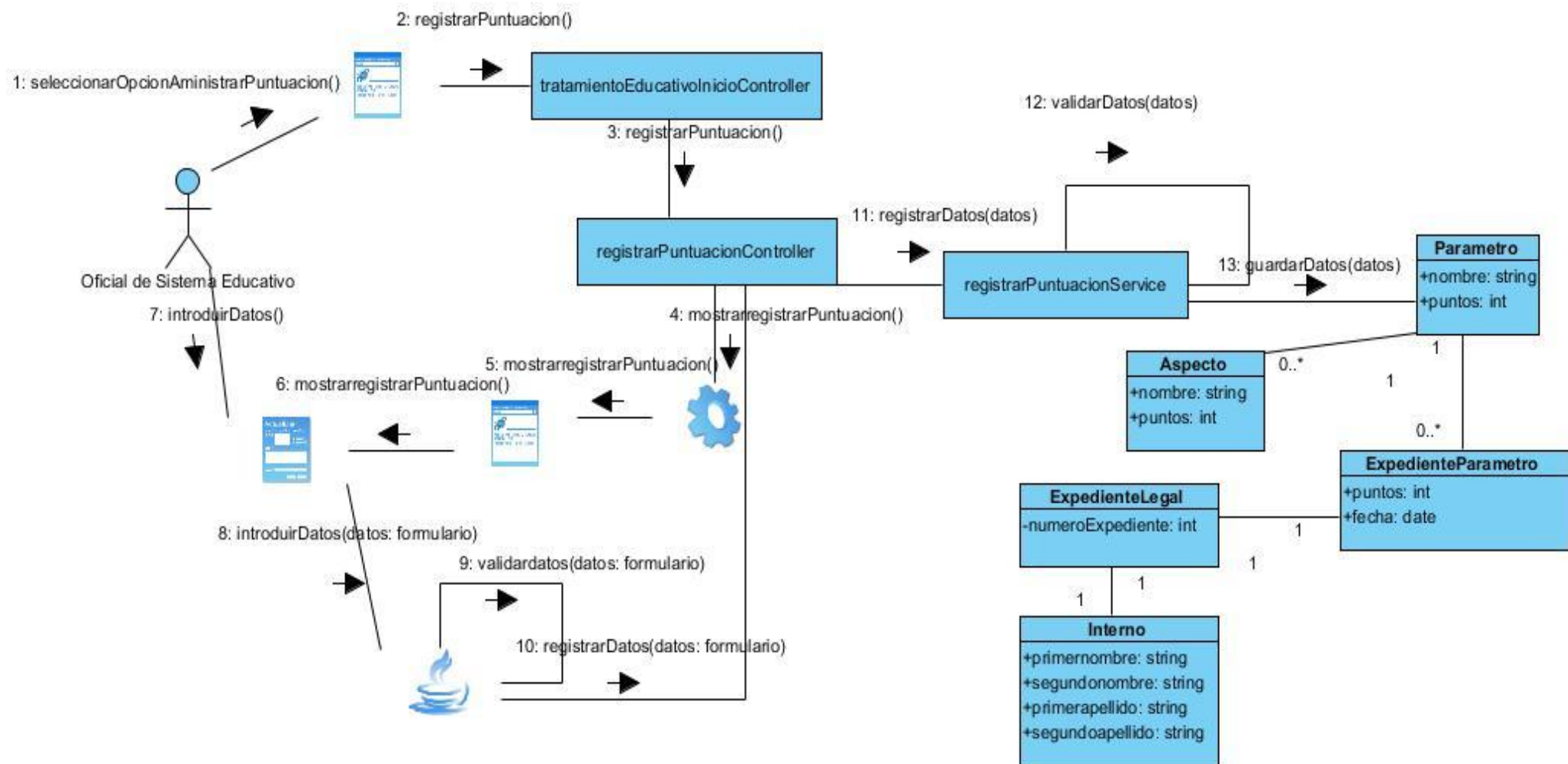


Figura 8 DC Administrar puntuación del módulo Emulación

2.11. Diseño de la Base de Datos

Un Diagrama de Entidad-Relación (DER) es una herramienta utilizada para el modelado de datos de un sistema de información, ya que se representan las entidades relevantes para el sistema, así como sus interrelaciones y propiedades. Teniendo en cuenta los datos de las diferentes tablas contenidas en los DER correspondientes a los módulos Administración de biblioteca y Emulación, se puede afirmar que los mismos se encuentran en Segunda Forma Normal (2FN), ya que están en la Primera Forma Normal (1FN) y, además, no existen dependencias parciales.

A continuación se muestran los DER de los módulos Administración de biblioteca y Emulación. En el caso del DER correspondiente al módulo Administración de biblioteca se ven reflejadas 22 tablas y sus respectivas relaciones y el DER del módulo Emulación consta de 11 tablas. Posteriormente se dará una descripción de las tablas más significativas contenidas en estos DER.

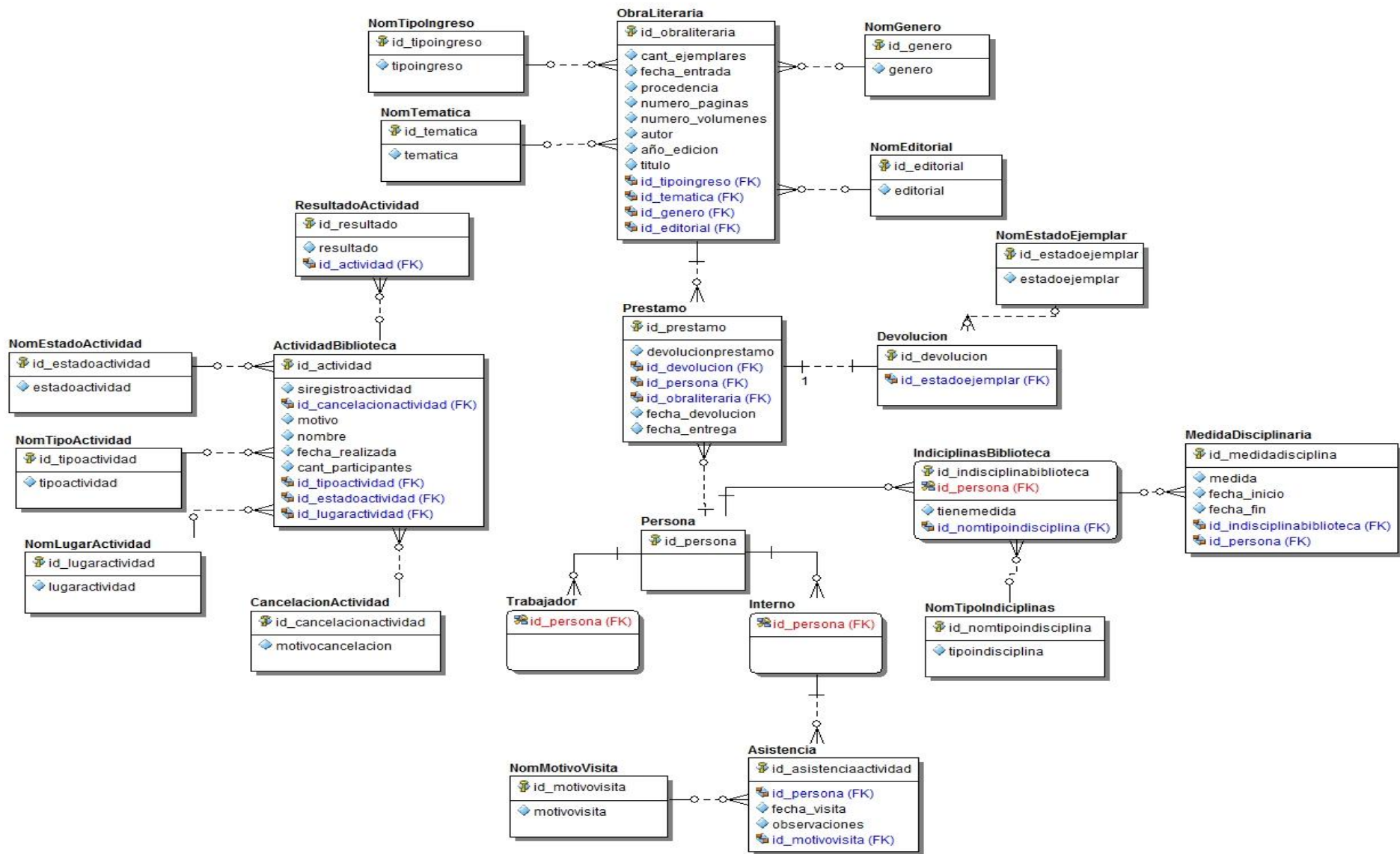


Figura 9 DER del módulo Administración de biblioteca

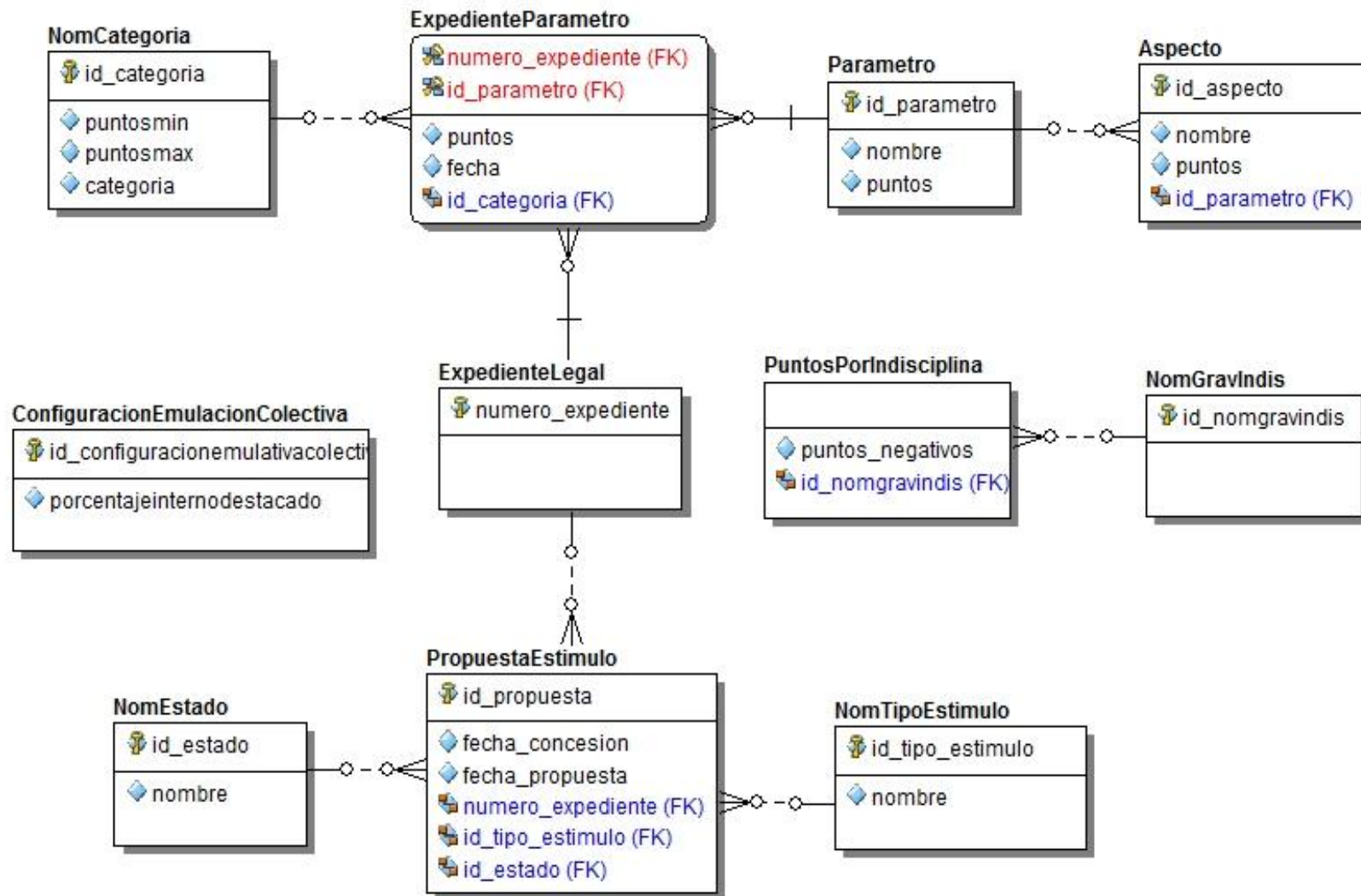


Figura 10 DER del módulo Emulación

2.11.1. Descripción de las tablas más significativas

A continuación se describen las tablas más significativas contenidas en los DER correspondientes a los módulos Administración de biblioteca y Emulación.

Administración de biblioteca

Nombre	ObraLiteraria	
Descripción	Almacena los datos de una obra literaria.	
Atributo	Tipo de dato	Descripción del atributo
OBRALITERARIA_ID	NUMBER(19,0)	Almacena el ID de las obras literarias que pueden recibir los internos, no puede ser nulo.
TITULO	VARCHAR2(255 BYTE)	Almacena el nombre de la obra literaria.
AUTOR	VARCHAR2(255 BYTE)	Almacena el nombre del autor de la obra literaria.
GENERO	VARCHAR2(255 BYTE)	Almacena el género que posee la obra literaria.
ANNO_EDICION	NUMBER(19,0)	Almacena el año en que fue escrita la obra literaria.
EDITORIAL	VARCHAR2(255 BYTE)	Almacena la editorial de la obra literaria.
NUMERO_VOLUMENES	NUMBER(19,0)	Almacena los volúmenes que puede tener la obra literaria.
FECHA_ENTRADA	DATE	Almacena la fecha en la que fue recibida la obra literaria por primera vez en la biblioteca.
PROCEDENCIA	VARCHAR2(255 BYTE)	Almacena el nombre de la entidad que ofreció la obra literaria, según sea el caso.

NUMERO_PAGINAS	NUMBER(19,0)	Almacena el total de páginas de la obra literaria.
CANTIDAD_EJEMPLARES	NUMBER(19,0)	Almacena la cantidad de libros de una misma obra literaria existente en la biblioteca.
TIPO_INGRESO	VARCHAR2(255 BYTE)	Almacena el modo por el cual ingresó la obra literaria.

Tabla 17 Descripción de la tabla ObraLiteraria del módulo Administración de biblioteca

Nombre	Prestamo	
Descripción	Almacena los datos de un préstamo de obra literaria.	
Atributo	Tipo de dato	Descripción del atributo
PARAMETRO_ID	NUMBER(19,0)	Almacena el identificador (ID) del parámetro, no puede ser nulo.
NOMBRE	VARCHAR2(255 BYTE)	Almacena el nombre del parámetro que recibe el interno.
PUNTOS	NUMBER(10,0)	Almacena la puntuación que recibe el interno por cada parámetro.

Tabla 18 Descripción de la tabla Prestamo del módulo Administración de biblioteca

Emulación

Nombre	Parametro	
Descripción	Almacena los datos correspondientes a los parámetros que se tienen en cuenta en la emulación.	
Atributo	Tipo de dato	Descripción del atributo
PARAMETRO_ID	NUMBER(19,0)	Almacena el identificador (ID) del parámetro, no puede ser nulo.

NOMBRE	VARCHAR2(255 BYTE)	Almacena el nombre del parámetro.
PUNTOS	NUMBER(10,0)	Almacena la puntuación que recibe el interno por cada parámetro.

Tabla 19 Descripción de la tabla Parametro del módulo Emulación

Nombre	PropuestaEstimulo	
Descripción	Almacena los datos correspondientes a la propuesta de estímulos que se le pueden otorgar al interno.	
Atributo	Tipo de dato	Descripción del atributo
PROPUESTAESTIMULO_ID	NUMBER(19,0)	Almacena el ID de la propuesta de estímulo, no puede ser nulo.
EXPEDIENTE_ID	NUMBER(19,0)	Almacena el ID del Expediente Legal del interno.
FECHA_CONCESION	DATE	Almacena la fecha de concesión del estímulo.
FECHA_PROPUUESTA	DATE	Almacena la fecha de la propuesta del estímulo.
TIPOESTIMULO_ID	NUMBER(19,0)	Almacena el tipo de estímulo que se le propone al interno.
ESTADO_ID	NUMBER(19,0)	Almacena el estado de la propuesta de estímulo.

Tabla 20 Descripción de la tabla PropuestaEstimulo del módulo Emulación

Nombre	ConfiguracionEmulacionColectiva
Descripción	Almacena los datos correspondientes al porcentaje de internos

	destacados que debe tener un colectivo para tener dicha categoría.	
Atributo	Tipo de dato	Descripción del atributo
CONFIGURACION_ID	NUMBER(19,0)	Almacena el ID de la configuración de la emulación colectiva, no puede ser nulo.
PORCENTAJEINTERNODESTACADO	NUMBER(10,0)	Almacena el porciento de internos destacados que debe tener un colectivo destacado.

Tabla 21 Descripción de la tabla ConfiguracionEmulacionColectiva del módulo Emulación

2.12. Conclusiones parciales

En este capítulo se detallaron los aspectos fundamentales del diseño, teniendo en cuenta los procesos fundamentales que conforman a los módulos Administración de Biblioteca y Emulación, así como sus funcionalidades y los patrones de diseño que se utilizaron para la solución del problema propuesto.

Se generaron los siguientes artefactos para los módulos Administración de Biblioteca y Emulación:

- Diagramas de clases del diseño.
- Diagramas de colaboración.
- Diagramas de Entidad-Relación.

Después de tener todos estos elementos se arriba a las siguientes conclusiones:

- Los artefactos generados propiciaron un mejor entendimiento y una descripción más detallada del problema, por lo que se garantizó el avance a la posterior implementación de la solución.
- El uso de patrones en el diseño del sistema dio una mejor organización y coherencia a los diagramas correspondientes a la solución del problema resuelto, garantizando su reutilización en otro momento.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS

3.1. Introducción

Luego de obtenidos los artefactos en el flujo de trabajo de diseño, en el presente capítulo se inicia la implementación del sistema. Se presentan los diagramas de despliegue y de componentes correspondientes a los módulos Administración de biblioteca y Emulación. También se expone el funcionamiento de Grails con sus capas para la implementación y se muestran las pruebas realizadas como parte del proceso de comprobación de la solución.

3.2. Modelo de implementación

El modelo de implementación es una correspondencia directa de los modelos de diseño y de despliegue. Este define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación. (21) En sentido general se puede afirmar que el propósito fundamental de la implementación es desarrollar la arquitectura y el sistema como un todo

Como parte de la implementación de los módulos Administración de biblioteca y Emulación, se realizan los diagramas de componentes y el modelo de despliegue correspondientes a la solución que se propone.

3.3. Diagramas de componentes

Un diagrama de componentes “muestra un conjunto de componentes y sus relaciones; los diagramas de componentes muestran los componentes de un sistema desde un punto de vista estático”. (15)

Una de las facilidades que brindan los diagramas de componentes es visualizan cómo los componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema. La distribución de componentes se realizó por paquetes, coincidiendo con la arquitectura de la aplicación.

A continuación se muestran los diagramas de componentes generales de los módulos Administración de biblioteca y Emulación, los cuales representan la relación de estos módulos con el resto de los subsistemas del SIDEPA.

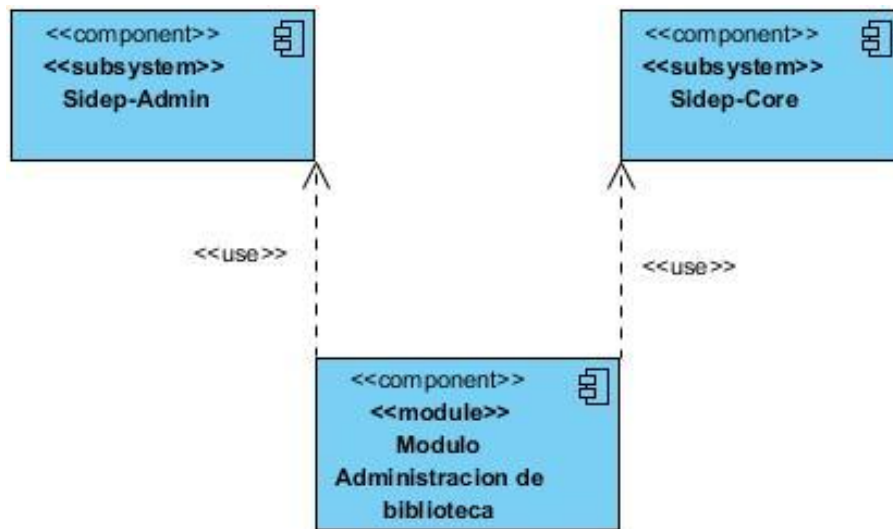


Figura 11 Diagrama de componentes general del módulo Administración de biblioteca

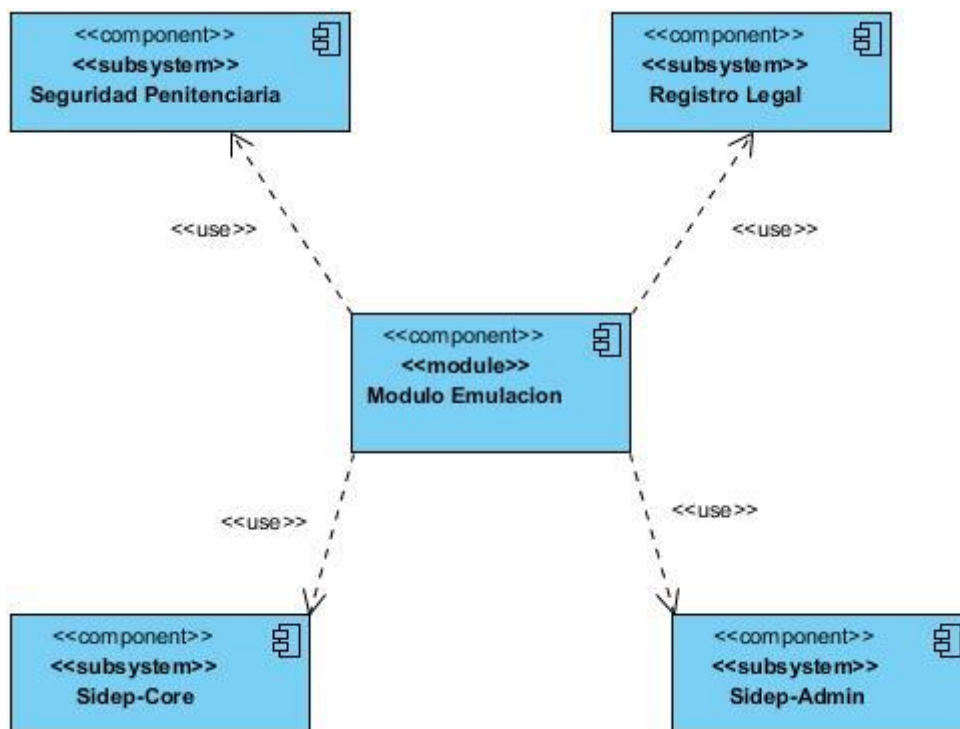


Figura 12 Diagrama de componentes general del módulo Emulación

Posteriormente se muestran detalladamente las relaciones internas de los componentes de los módulos Administración de biblioteca y Emulación, incluyendo además las que tienen estos con los componentes de otros subsistemas.

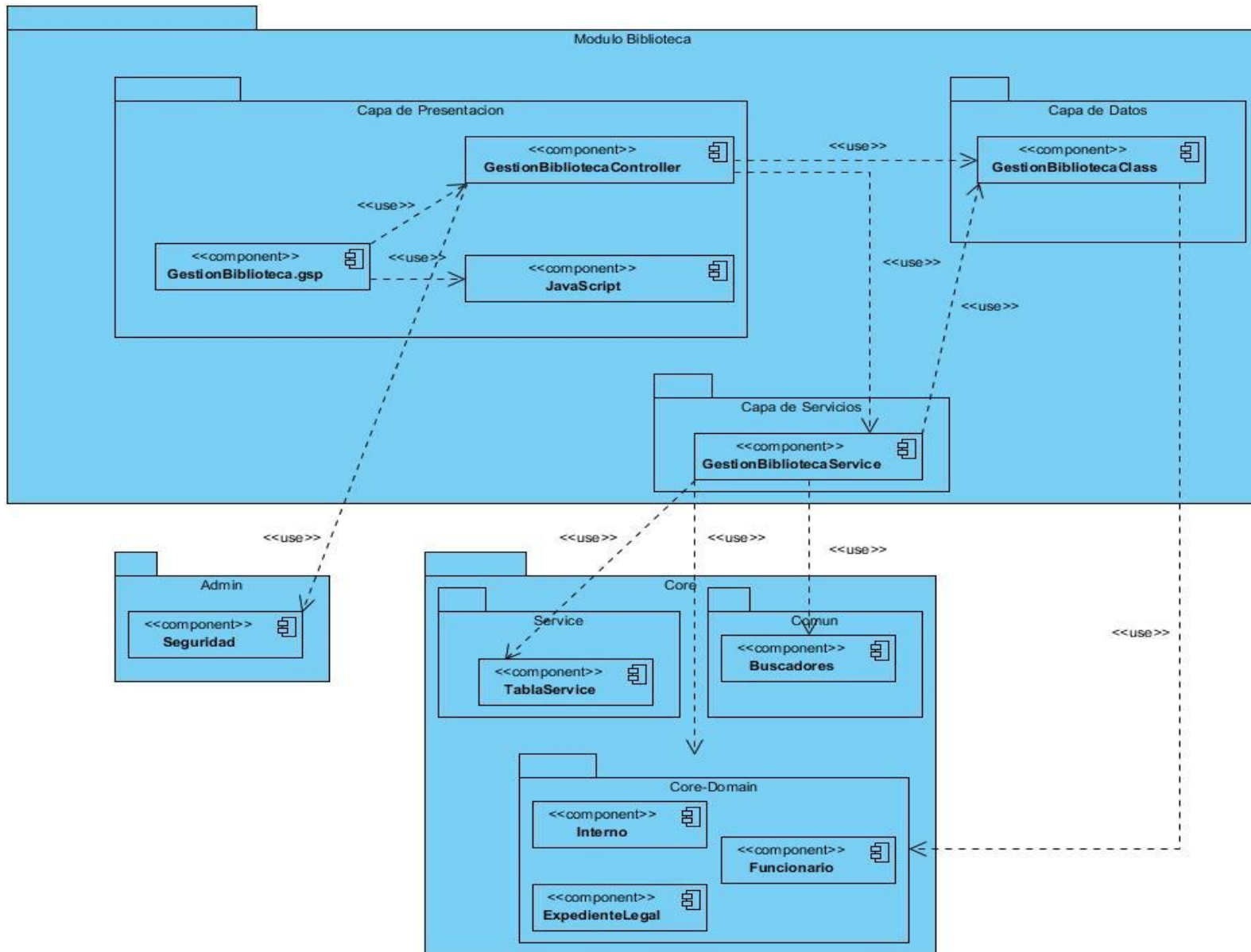


Figura 13 Diagrama de componentes específico del módulo Administración de biblioteca

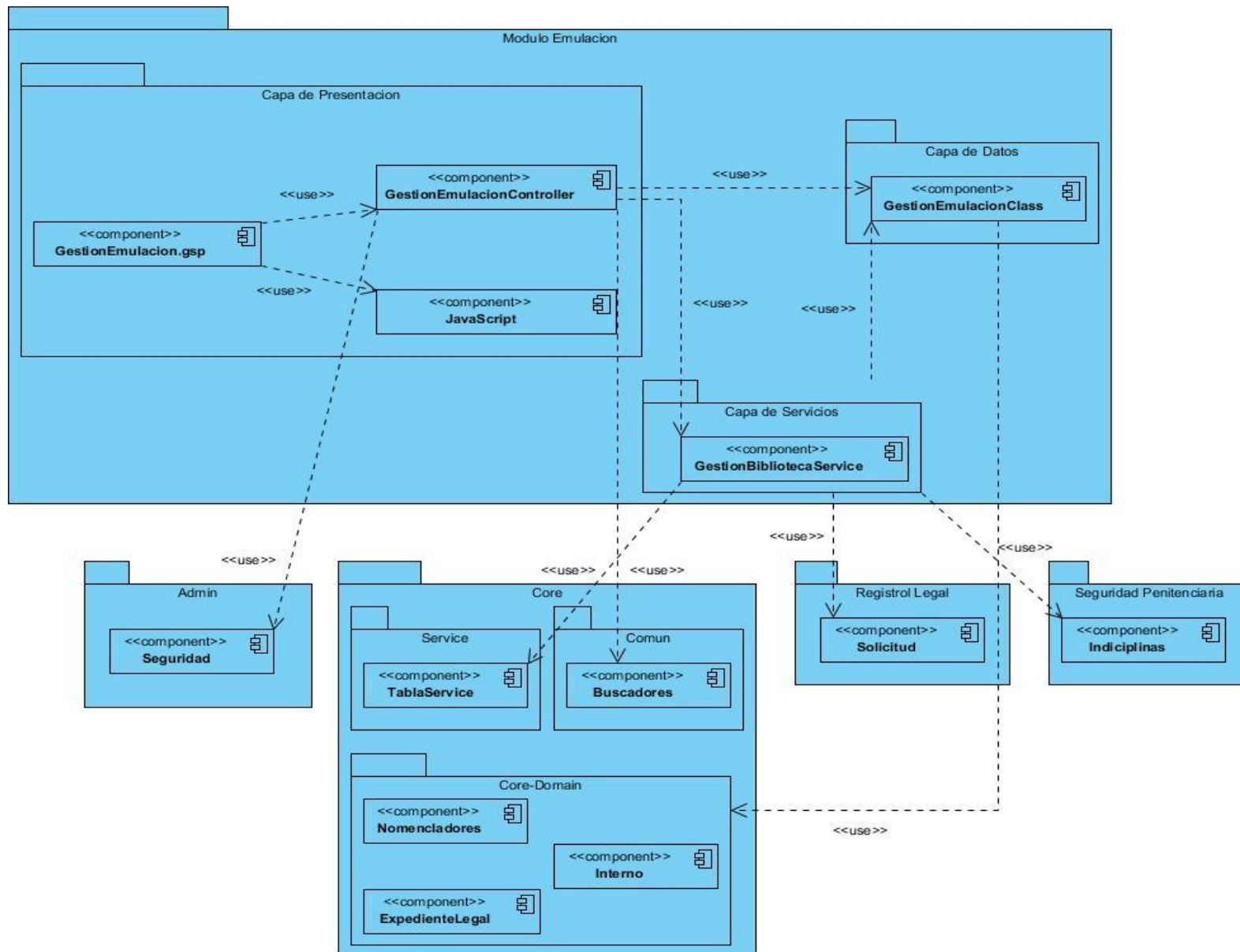


Figura 14 Diagrama de componentes específico del módulo Emulación

En los diagramas de componentes generales de los módulos Administración de biblioteca y Emulación (Figura 11 y Figura 12), su integración con los subsistemas Sidep-Admin y Sidep-Core, los que facilitan el cumplimiento en gran medida de los procesos a desarrollar; en el caso de Sidep-Admin se utiliza para la seguridad en los módulos (Figura 15) y el Sidep-Core para interactuar con clases comunes. En el caso específico del módulo Emulación, tiene además relación con los subsistemas Seguridad Penitenciaria y Registro Legal, para el manejo de indisciplinas cometidas por los internos y el envío de solicitudes a instancias superiores respectivamente.

```
package sidep.tratamientoeducativo.controller.emulacion

import grails.converters.JSON
import sidep.tratamientoeducativo.domain.emulacion.Parametro;
import sidep.tratamientoeducativo.service.emulacion.ParametroService;
import sidep.tratamientoeducativo.service.emulacion.AspectoService;
import sidep.admin.seguridad.Security

@Security("adminPunt")
class AdministrarPuntuacionController {

    def parametroService
    def aspectoService
```

Figura 15 Ejemplo de la utilización del Sidep-Admin en el módulo Emulación

En los diagramas de componentes más detallados de los módulos Administración de biblioteca y Emulación (Figura 13 y Figura 14) se puede observar la interacción de las capas que integran cada uno de estos módulos (capa de datos, capa de presentación y capa de servicios) donde cada uno de estos está compuesto por sus correspondientes componentes. Además, se refleja la relación de estos módulos con los subsistemas Admin y Core y, en el caso específico de Emulación, con los subsistemas Seguridad Penitenciaria y Registro Legal.

3.4. Seguridad

La seguridad en los módulos Administración de biblioteca y Emulación se logró utilizando Spring Security, el cual proporciona servicios integrales de seguridad para J2EE, teniendo en cuenta las principales áreas de seguridad: la autenticación y la autorización. La autenticación es el proceso en el

que la aplicación pueda o no dar acceso para realizar una acción en la aplicación. La autorización es el proceso que permite decidir si un usuario tiene permiso para realizar una acción dentro de la aplicación.

Para lograr la autorización se le dio un código a cada funcionalidad, luego se crean los roles y a estos se les asignan las funcionalidades que podrá realizar dicho rol y, por último, en los controladores, mediante una notación de seguridad, se aseguran las acciones que se pueden realizar sobre ese controlador. Con este mecanismo se logra que el usuario, para poder acceder a una determinada acción, tenga asignado el rol que corresponda con la acción a realizar sobre la aplicación.

A continuación se muestra un ejemplo de la utilización de la seguridad en la aplicación, específicamente en el CU Administrar puntuación del módulo Emulación.

```
package sidep.tratamientoeducativo.controller.emulacion

import grails.converters.JSON
import sidep.tratamientoeducativo.domain.emulacion.Parametro;
import sidep.tratamientoeducativo.service.emulacion.ParametroService;
import sidep.tratamientoeducativo.service.emulacion.AspectoService;
import sidep.admin.seguridad.Security

@Security("adminPunt")
class AdministrarPuntuacionController {

    def parametroService
    def aspectoService
```

Figura 16 Ejemplo de la seguridad en el módulo Emulación

3.5. Internacionalización

La internacionalización posibilita a una aplicación que la interfaz de usuario soporte distintos idiomas. Al emplear como framework Grails, se creará un fichero para cada idioma que soporte la aplicación en la carpeta `grails-app/i18n` y en estos ficheros se escriben los mensajes para cada componente. Los nombres de estos archivos empiezan con el término "Messages" y luego el "Locale" del idioma; por ejemplo:

- Messages_es.properties: para los mensajes en español.
- Messages_en.properties: para los mensajes en inglés.

A continuación se muestra un ejemplo de los ficheros correspondientes al idioma Español.

```

messages_es.properties x
domain.calcular.label = Calcular

#Radio
domain.metodo.label=M\u00E9todo
domain.dedicado.label=Dedicado
domain.multiproposito.label=Multiprop\u00F3sito
domain.medio.label=Medio
domain.estado_Tecnico.label=Estado T\u00E9cnico
domain.real.label=Real
domain.veintidos.label=22
domain.doscientos.label=212
domain.doscientosA.label=260
domain.dosSetentidos.label=272
domain.kenwood.label=Kenwood
domain.kiritsum.label=Kiritsum
domain.plantilla.label=Plantilla
domain.movil.label=Movil
domain.ensayo.label=Ensayo
domain.uSeiscientos.label=U-600
domain.uSetecientos.label=U-700
domain.dosMil.label=2640
domain.bien.label=Bien
domain.mal.label=Mal
domain.regular.label=Regular
domain.tel.label=Tel
domain.lugarA.label=Lugar

```

Figura 17 Fichero para los mensajes en español

3.6. Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. (22)

Teniendo en cuenta que se despliegan los módulos Administración de biblioteca y Emulación, los componentes físicos que intervienen en los procesos de gestión de la información se distribuyen de la siguiente manera: las vistas estarán incluidas en el nodo de la PC cliente, los controladores y los servicios estarán en el nodo del servidor web y las tablas estarán en el servidor de BD.

A continuación se muestra el modelo de despliegue correspondiente a dichos módulos.



Figura 18 Diagrama de despliegue de los módulos Administración de biblioteca y Emulación

Para el despliegue de los módulos es necesario que se cumplan las siguientes características:

PC cliente:

- Navegador web Mozilla Firefox 3.6.*
- RAM: 512 MB o superior.

Servidor de la aplicación (Servidor Web):

- JDK 1.6
- Apache Tomcat 6.0.*
- Linux Suse Enterprise Server 10.0 SP2
- Microprocesador: 4 núcleos, 3 GHz
- RAM: 4 GB
- Espacio necesario para la instalación: 250 MB
- Espacio libre requerido: 250 GB

Servidor de BD:

- JDK 1.6
- Oracle 11g
- Microprocesador: 4 núcleos, 3 GHz
- RAM: 4 GB
- Espacio necesario para la instalación: 2 GB
- Espacio libre requerido: 500 GB

3.7. Pruebas

Al desarrollar sistemas informáticos se corre un alto riesgo de que se produzcan errores producto al fallo humano. Estos pueden ocurrir desde el comienzo del proceso, ya sea en la definición de los objetivos, el diseño, la implementación o en otras fases. Por tal razón, se hace necesario hacer uso de una actividad que garantice la calidad, con el objetivo de medir el grado en que el software cumple con los requisitos.

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto de software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un sistema informático. (18)

3.7.1. Estrategia de prueba

La estrategia de prueba tiene como finalidad fundamental describir el enfoque y los objetivos generales de las actividades de prueba. Con el objetivo de guiar el proceso de pruebas y garantizar al máximo la eliminación de las no conformidades detectadas por el probador se hace necesario definir la estrategia de prueba, este proceso consiste en la integración de diferentes factores y pasos que dan como resultado una correcta construcción del software.

A continuación se especifica y explica la estrategia seleccionada para las pruebas de los módulos Administración de biblioteca y Emulación:

- Método de prueba: método de caja negra.
- Tipo de prueba: funcionalidad.
- Nivel de prueba: prueba de desarrollador.
- Tipo de técnica de prueba: casos de prueba.

Método de caja negra

Se realizan pruebas que verifican que cada función es operativa y que la entrada y salida se producen de forma correcta. Estas pruebas se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. (18)

Las pruebas de caja negra se centran principalmente en los requisitos funcionales del software y pretenden encontrar los siguientes tipos de errores:

- Funciones incorrectas o ausentes.
- Errores de interfaz.

- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento. (18)

Funcionalidad

Tipo de prueba centrada en validar las funciones que son objeto de prueba como lo que deben ser, ofreciendo los servicios, métodos o casos de usos requeridos. (18)

Este tipo de prueba se basa específicamente en la ejecución y revisión de las funcionalidades previamente diseñadas para el sistema, estas pruebas se hacen mediante el diseño de modelos de pruebas que buscan evaluar cada una de las opciones con las que cuenta el software. El factor fundamental que se tiene en cuenta es la capacidad de la aplicación para manejar grandes volúmenes de datos.

Prueba de desarrollador

Este nivel de prueba indica los aspectos más adecuados de diseño e implementación de la prueba que se debe llevar a cabo por parte del equipo de desarrolladores del proyecto. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas de desarrollador que la diseñó e implementó; aunque es recomendable que los desarrolladores creen las pruebas de forma que estén disponibles para que las ejecuten grupos de pruebas independientes.

Es recomendable que la prueba de desarrollador no cubra únicamente unidades independientes de prueba aisladas.

Casos de prueba

La generación de cada caso de prueba debe ir acompañada del resultado que ha de producir el software al ejecutar dicho caso para detectar un posible fallo en el programa. Los casos de prueba determinan un conjunto de entradas, condiciones de ejecución y resultados esperados para un objetivo particular. Cada técnica de pruebas proporciona unos criterios distintos para generar estos casos o datos de prueba.

Con los diseños de casos de prueba realizados, se pretende demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto. A continuación se muestra los casos de prueba de los CU Registrar obra literaria (Gestionar obra literaria) y Administrar puntuación de los módulos Administración de biblioteca y Emulación respectivamente.

Escenario	Descripción	Título	Autor	Género	Año de Edición	Temática	Editorial	Número de Volúmenes	Fecha de Entrada	Procedencia	Cantidad de ejemplares	Tipo de Ingreso	Respuesta del sistema	Flujo central
EC 2.1 Registrar Obra Literaria	Se registra exitosamente la obra literaria.	V	V	V	V	V	V	V	V	V	V	V	El sistema registra la información referente a las obras literarias.	<ol style="list-style-type: none"> 1. Seleccionar la opción "Obra Literaria". 2. El sistema muestra los campos para introducir los datos de la nueva obra literaria. 3. Introduce los datos y oprime el botón "Aceptar". 4. El sistema valida los datos introducidos, si los datos son incorrectos muestra el mensaje "Corrija los datos erróneos introducidos", si hay campos vacíos muestra mensaje. "Los campos en rojo son requeridos".
EC 2.2 Los datos son incorrectos	No se registra una nueva obra literaria porque el sistema comprueba que los datos son incorrectos	V	V	V	V	V	V	V	V	V	V	V	El sistema muestra el mensaje de error "Corrija los datos erróneos introducidos"	<ol style="list-style-type: none"> 1. Seleccionar la opción "Obra Literaria". 2. El sistema muestra los campos para introducir los datos de la nueva obra literaria. 3. Introduce los datos y oprime el botón "Aceptar". 4. El sistema valida los datos introducidos, si los datos son incorrectos muestra el mensaje "Corrija los datos erróneos introducidos", si hay campos vacíos muestra mensaje. "Los campos en rojo son requeridos".
EC 2.3 Campos vacíos	No se registran los datos de la obra literaria porque el sistema comprueba que faltan datos obligatorios.	V	V	V	V	V	V	V	V	V	V	V	El sistema muestra un mensaje de error "Los campos en rojo son requeridos"	<ol style="list-style-type: none"> 1. Seleccionar la opción "Obra Literaria". 2. El sistema muestra los campos para introducir los datos de la nueva obra literaria. 3. Introduce los datos y oprime el botón "Aceptar". 4. El sistema valida los datos introducidos, si los datos son incorrectos muestra el mensaje "Corrija los datos erróneos introducidos", si hay campos vacíos muestra mensaje. "Los campos en rojo son requeridos".

Figura 19 CP¹¹ del CU Registrar obra literaria (Gestionar obra literaria) del módulo Administración de biblioteca

¹¹ Caso de prueba

Escenario	Descripción	Puntos	Respuesta del sistema	Flujo central
EC 1.1 Administrar puntuación	Se registra la puntuación.	V	Registra la puntuación	<ol style="list-style-type: none"> 1. Seleccionar la opción "Administrar puntuación individual". 2. El sistema muestra la lista de los parámetros que podrán ser configurados y le permite al usuario introducir la puntuación. 3. Registrar la puntuación y oprime el botón "Cerrar".
EC 1.2 Cancelar Administrar puntuación	Se cancela la operación	NA	Cancela la operación y permite al usuario registrar la puntuación nuevamente.	<ol style="list-style-type: none"> 1. Seleccionar la opción "Administrar puntuación individual". 2. El sistema muestra la lista de los parámetros que podrán ser configurados y le permite al usuario introducir la puntuación. 3. Registrar la puntuación y oprime el botón "Cancelar".
EC 1.3 Faltan datos obligatorios	No se registran los datos de la puntuación porque el sistema comprueba que faltan datos obligatorios	I	El sistema muestra el mensaje de error "Los campos en rojo son requeridos" y señala los datos obligatorios que no fueron introducidos.	<ol style="list-style-type: none"> 1. Seleccionar la opción "Administrar puntuación individual". 2. El sistema muestra la lista de los parámetros que podrán ser configurados y le permite al usuario introducir la puntuación. 3. El sistema verifica los datos y comprueba que faltan datos obligatorios. 4. El sistema muestra el mensaje de error "Los campos en rojo son requeridos".
EC 1.4 Limite de puntuación	No se registran los datos de la puntuación porque el sistema comprueba que la puntuación excede el limite.	I	El sistema muestra el mensaje de error "El total de la puntuación excede los 100 puntos".	<ol style="list-style-type: none"> 1. Seleccionar la opción "Administrar puntuación individual". 2. El sistema muestra la lista de los parámetros que podrán ser configurados y le permite al usuario introducir la puntuación. 3. El sistema verifica los datos y comprueba que la puntuación excede el limite. 4. El sistema muestra el mensaje de error "El total de la puntuación excede los 100 puntos".

Figura 20 CP del CU Administrar puntuación del módulo Emulación

3.7.2. Resultados de las pruebas

Luego de aplicar los casos de prueba a los módulos, se detectaron un total de 13 no conformidades en una primera iteración, 7 no conformidades en el módulo Administración de biblioteca y 6 en el módulo Emulación. En una segunda iteración no se detectaron nuevos errores en ninguno de los módulos.

A continuación se describe las no conformidades detectadas en las pruebas de caja negra realizadas a las funcionalidades de los módulos Administración de biblioteca y Emulación.

Administración de biblioteca

No.	Ubicación	Descripción de la no conformidad
1.	Administración de Biblioteca/Gestionar Obra Literaria.	La palabra "título" está sin tilde en nombre del campo para introducir el título de la obra.
2.	Administración de Biblioteca/Gestionar Obra Literaria.	La palabra "género" está sin tilde en nombre del campo para introducir el género de la obra.
3.	Administración de Biblioteca/Gestionar Préstamo.	La palabra "préstamos" está sin tilde en el título para registrar los préstamos.
4.	Administración de Biblioteca/Gestionar Préstamo.	La palabra "préstamos" está sin tilde en el título la tabla de los préstamos.
5.	Administración de Biblioteca/Gestionar Préstamo.	El buscador de internos y funcionarios no funciona.
6.	Administración de Biblioteca/Buscar Obra Literaria.	La palabra "búsqueda" está sin tilde en checkbox de búsqueda avanzada.

7.	Administración de Biblioteca/Gestionar Actividad.	El botón consultar no funciona.
----	---------------------------------------------------	---------------------------------

Tabla 22 No conformidades del módulo Administración de biblioteca

Emulación

No.	Ubicación	Descripción de la no conformidad
1.	Emulación/Administrar Puntuación.	La palabra “puntuación” está sin tilde en el checkbox de aspectos.
2.	Emulación/Administrar Puntuación.	La palabra “parámetro” está sin tilde en el título de las tablas de los parámetros.
3.	Emulación/Administrar Puntuación.	Se adiciona un aspecto que tiene un mismo nombre.
4.	Emulación/Registrar Puntuación.	La palabra “guía” está sin tilde en el título de la tabla de los internos del colectivo del Educador Guía.
5.	Emulación/Registrar Puntuación.	El botón evaluar no funciona.
6.	Emulación/Consultar Estímulos Otorgados.	La palabra “estímulo” está sin tilde en el título de las tablas de los internos con estímulos.

Tabla 23 No conformidades del módulo Emulación

A continuación se muestra un gráfico donde se evidencian los resultados obtenidos en las pruebas a los módulos Administración de biblioteca y Emulación.

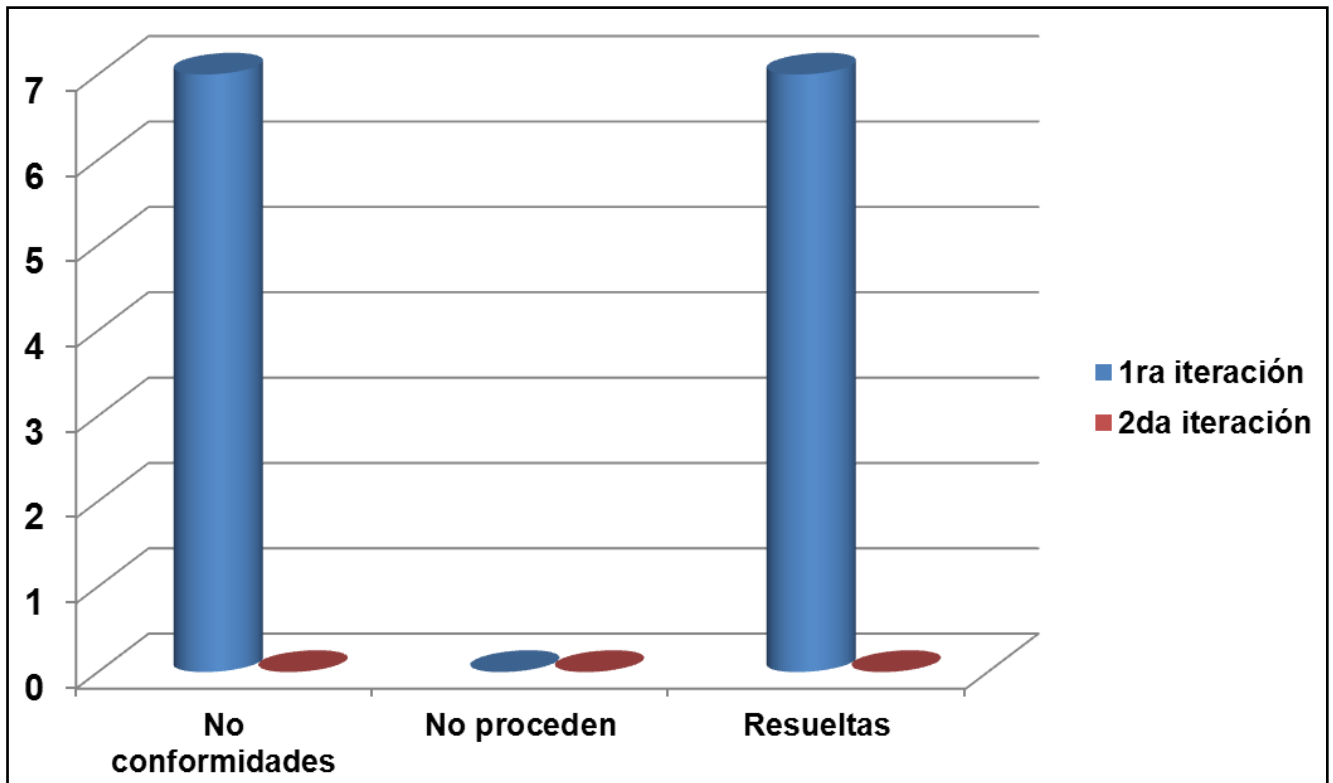


Figura 21 Resultados obtenidos en las pruebas al módulo Administración de biblioteca

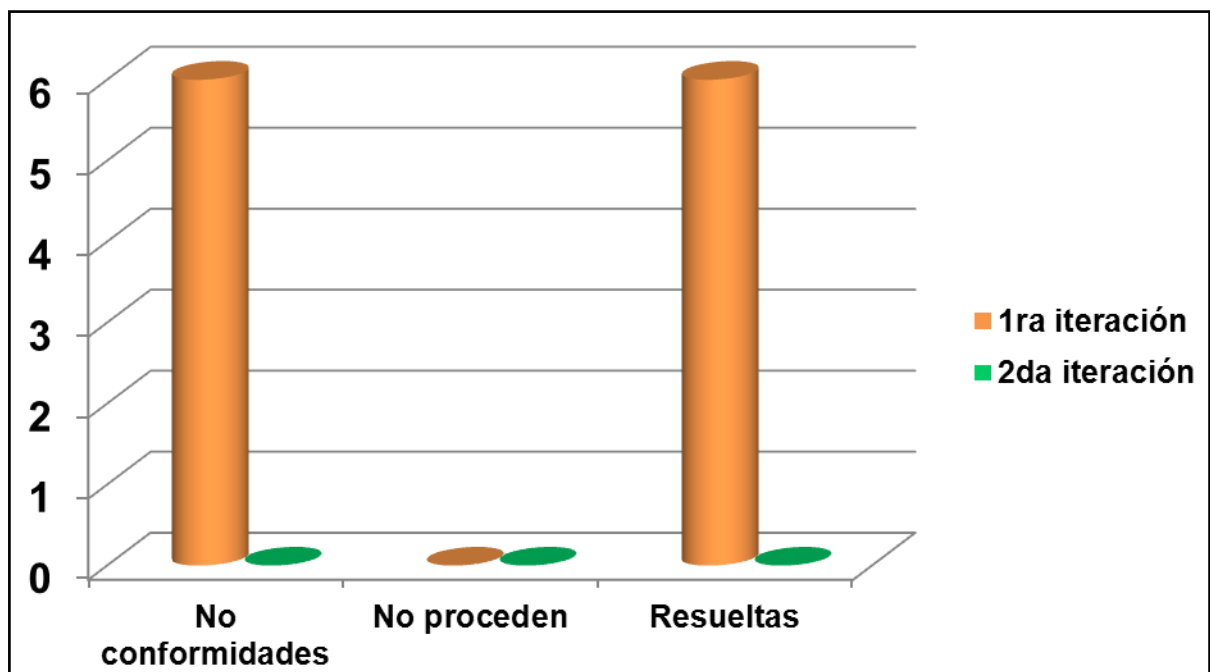


Figura 22 Resultados obtenidos en las pruebas al módulo Emulación.

3.8. Conclusiones parciales

En este capítulo se mostraron y explicaron los artefactos generados durante la implementación, entre ellos los diagramas de componentes y el diagrama de despliegue. Se incluyeron elementos importantes relacionados con la implementación. Además, para comprobar el correcto funcionamiento de la aplicación y satisfacción de los requisitos funcionales capturados se puso en práctica una estrategia de prueba para garantizar la calidad de la solución propuesta.

Después de tener todos estos elementos se arriba a las siguientes conclusiones:

- Los artefactos generados lograron mostrar con claridad la definición y descripción de la solución que da respuesta a los requisitos funcionales y a las características que debe cumplir el sistema, los cuales fueron determinados con el cliente.
- La estrategia de prueba puesta en práctica garantizó la satisfacción de las necesidades que debe cubrir el sistema y lograr que la solución propuesta tenga un mayor acierto.

CONCLUSIONES GENERALES

Luego de una exhaustiva investigación y la obtención de la solución que responde a los módulos Administración de Biblioteca y Emulación perteneciente a SIDEPA, se concluye lo siguiente:

- Se detallaron los elementos necesarios para la búsqueda de la solución al problema propuesto, donde el estudio y análisis de la metodología, herramientas, tecnologías y lenguajes permitió demostrar la elección de estos para el desarrollo de los módulos Administración de biblioteca y Emulación.
- Se generaron los artefactos necesarios para documentar el diseño del sistema, diagramas de clases, diagramas de colaboración y modelo de datos, permitiendo materializar con precisión los requisitos expuestos por el cliente para la posterior implementación de los módulos Administración de biblioteca y Emulación.
- Se implementaron todas las funcionalidades definidas para los módulos Administración de biblioteca y Emulación haciendo uso de los estándares de codificación existentes por el grupo de trabajo, logrando la integración de estos al SIDEPA.
- Se realizaron las pruebas de calidad a los módulos Administración de biblioteca y Emulación, donde las no conformidades encontradas fueron corregidas, demostrando que cada una de las funcionalidades responde apropiadamente a los requisitos funcionales definidos con el cliente.

RECOMENDACIONES

Al concluir este trabajo se proponen las siguientes recomendaciones:

- Comprobar el rendimiento de los módulos Administración de biblioteca y Emulación frente a bases de datos de grandes volúmenes de información, para validar que el diseño propuesto satisface tales condiciones, puesto que actualmente el sistema está probado para una cantidad limitada de datos que no ponen al límite la solución propuesta.
- Realizar las pruebas de aceptación para obtener un producto que satisfaga completamente los requisitos capturados con el cliente.
- Desarrollar los manuales de usuario de los módulos Administración de biblioteca y Emulación para capacitar al personal que hará uso de estos en la aplicación posteriormente.
- Continuar perfeccionando los requisitos del sistema mediante la actualización de los cambios en acuerdo con el cliente, teniendo en cuenta nuevas legislaciones o especificaciones que puedan surgir en el futuro.

REFERENCIAS BIBLIOGRÁFICAS

1. **Osorio, M.** *Diccionario de Ciencias Jurídicas Políticas y Sociales*. Buenos Aires, Argentina : Heliasta, 1981.
2. **Pérez, Irina Morciego.** *Diseño e Implementación del módulo Instrucción Escolar del SIGDATI*. La Habana : s.n., 2011.
3. **Ministerio de Gobierno y Justicia de Panamá.** Dirección General del Sistema Penitenciario de Panamá. [En línea] [Citado el: 15 de Enero de 2012.] <http://www.sistemapenitenciario.gob.pa/submenu.php?cod=25&id=1>.
4. **Dirección de Establecimientos Penitenciarios. MININT.** *Informe de factibilidad e ideas preliminares sobre la informatización del sistema penitenciario cubano*. La Habana : s.n.
5. **Estrada, Yelena Hernández.** *Análisis del Módulo Proceso Confiscatorio de Bienes del proyecto Sistema de Gestión Fiscal*. La Habana : s.n., 2009.
6. **Carrillo Pérez, Isaías, Pérez González, Rodrigo y Rodríguez Martín, Aureliano David.** *Metodología de desarrollo de software*. 2008.
7. **Boza Villalba, Renier y Schery Sánchez, Liset.** *Análisis y Diseño del proceso Dictámenes para el proyecto Sistema de Informatización de la Gestión de las Fiscalías II (SIGEF)*. 2011.
8. **SG Software Guru.** Revista digital SGúia. *Embarcadero E/R Studio*. [En línea] 2009. [Citado el: 25 de Enero de 2012.] <http://sg.com.mx/guia/node/1454>.
9. **Calderón Sanmartín, Jhon Alexander.** Blog "My Space to share". [En línea] Enero de 2008. [Citado el: 3 de Febrero de 2012.] <http://jcalderon.wordpress.com/2008/01/04/instalacion-y-configuracion-de-apache-tomcat-60-en-windows-xp>.
10. La Flecha. [En línea] Noviembre de 2003. [Citado el: 10 de Febrero de 2012.] <http://www.laflecha.net/canales/empresas/noticias/oracle-11g-la-nueva-base-de-datos-de-oracle>.
11. **Brito, Nacho.** *Manual de desarrollo web con Grails. JavaEE, como siempre debió haber sido*. 2009. ISBN: 978-84-613-2651.
12. mil veinticuatro. [En línea] 2008. [Citado el: 15 de Febrero de 2012.] <http://www.1024.com.uy/revista/index.php/revistas/galileo-galilei/item/25-grails-el-santo-grial-de-java>.

13. Dojo Toolkit. *Ecured*. [En línea] [Citado el: 2 de Marzo de 2012.] http://www.ecured.cu/index.php/Dojo_toolkit.
14. **Larman, Craig**. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999. ISBN: 970-17-0261-1.
15. **Jacobson, Ivor, Brooch, Gravy y Rum Baugh, James**. *El Lenguaje Unificado de Modelado. Manual de Referencia*. 2000.
16. **García, Fran**. Informática, Internet y otras cosas del querer. [En línea] 10 de Junio de 2009. [Citado el: 10 de Abril de 2012.] <http://www.frangarcia.net/txp/articles/319/caracteristicas-de-groovy>.
17. **Ecured**. Flujo de trabajo Requerimientos. *Ecured*. [En línea] 14 de Diciembre de 2010. http://www.ecured.cu/index.php/Flujo_de_Trabajo_Requerimiento.
18. **Pressman, Roger S**. *Ingeniería de Software. Un enfoque Práctico 5ta edición*. s.l. : McGraw-Hill, 2005.
19. **Monferrer Pérez, Sahily y Pérez Díaz, Juan Enrique**. *Análisis y Diseño del Módulo Sumario del proyecto Sistema de Gestión Fiscal*. La Habana : s.n., 2009.
20. Singleton. *Ecured*. [En línea] [Citado el: 1 de Junio de 2012.] http://www.ecured.cu/index.php/Patr%C3%B3n_Singleton.
21. **Duran Rivas, Yaksel y Sanz Mantilla, Daylin**. *Diseño e implementación del módulo Importación de los Depósitos Temporales* . La Habana : s.n., 2011.
22. Diagrama de despliegue. *Ecured*. [En línea] [Citado el: 15 de Mayo de 2012.] http://www.ecured.cu/index.php/Diagrama_de_despliegue.