

# **Universidad de las Ciencias Informáticas**

## **Facultad 4**



**Título: “Diseño e implementación de la Capa de Presentación de las funcionalidades relacionadas con el registro de decisiones y la ejecución de la pena del módulo Situación Jurídica.”**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN INFORMÁTICA**

### **AUTORES:**

Alien Gongora Rodríguez

Yordanys Pupo Diéguez

### **TUTOR:**

Ing. Arturo César Arias Orizondo

**Ciudad de la Habana, Mayo del 2008.**

**Año 50 de la Revolución**

## DECLARACIÓN DE AUTORÍA

---

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Alien Gongora Rodríguez.

---

Yordanys Pupo Diéguez.

---

Ing. Arturo César Arias Orizondo.

## DATOS DE CONTACTO

---

Tutor: Ing. Arturo César Arias Orizondo

Graduado en el 2003 Graduado en el 2003 de Ingeniero Informático del Instituto Superior Politécnico José Antonio Echeverría (ISPJAE), Título de Oro, promedio 5 puntos.

Ha impartido las asignaturas de Sistemas de Bases de Datos, Ingeniería de Software I y II y Gestión de Software en la Universidad de Ciencias Informáticas, participando en la elaboración de los programas de estas asignaturas.

Miembro del tribunal de acreditación de competencias del Departamento de Ingeniería de Software durante los cursos 2004-2005, 2005-2006.

Vicedecano de producción e investigación por dos cursos: 2004-2005, 2005-2006.

Miembro del consejo científico de la UCI desde el 2006.

Miembro de la comisión de carrera durante los cursos 2004-2005, 2005-2006.

Ha impartido cursos de postgrado sobre RUP y UML.

Participó en el Congreso y Feria TechnoInternet 2004 celebrado en Santiago de Chile, Chile.

Ha participado en evaluaciones de calidad de productos de software para Venezuela.

Diseñador de base de datos del proyecto SAFRE, primero de exportación de la UCI.

Líder del proyecto de desarrollo de software para la gestión del sistema penitenciario venezolano desde el 2005.

## **AGRADECIMIENTOS**

---

Agradecemos a todas aquellas personas que colaboraron con el desarrollo de este trabajo, en especial a Arturo César Arias Orizondo, Yanet Vega Miniet y Javier López Del Castillo Caymares por dedicar parte de su tiempo en la revisión y corrección de errores.

### **Alien Góngora Rodríguez**

A todas aquellas personas maravillosas que de una forma u otra contribuyeron para que hoy este aquí, aquellas personas de las que aprendí mucho, y me hicieron reflexionar y aprender que el sentido de la vida es sentirse responsable de aquellos a quien uno ama. A todos mis compañeros de aulas y amigos que siempre estuvieron presente cada momento de alegría y tristeza.

A toda mi familia, a mis maravillosos hermanos, a mi tía Idelina. Especialmente a mi mamá y a mi papá, que aunque ella no haya estado presente físicamente siempre ha sido mi principal inspiración y guía, el supo darme la confianza y ser el mejor de los padres, me enseñó a no detenerme ante las adversidades de la vida, a salir adelante, a confiar en mi persona y demostrarme que todo es posible cuando se desea y quiere...

### **Yordanys Pupo Diéguez**

A mis padres quienes me dieron la principal formación con las que hoy cuento, por brindarme su amor, confianza y apoyo incondicional, por estar siempre a mi lado haciendo posible cada meta que me he puesto en la vida.

A mis amigos y compañeros de aula que de una forma u otra me han dado una gran ayuda cuando más lo necesitaba y no tenía a nadie con quien contar.

A la revolución cubana que junto a nuestro gran líder FIDEL ha hecho realidad el gran sueño de encontrarme hoy en esta universidad optando por el título de Ingeniero en Ciencias Informáticas.

Control Penal es la especialidad penitenciaria encargada de llevar el control del expediente carcelario de los privados de libertad dentro del cual se gestiona su situación legal. En el contexto venezolano, esta actividad no se lleva de manera uniforme ni sistemática. Es por ello que como parte de la solución de modernización del sistema penitenciario el software desarrollado incluye un subsistema para el manejo de la situación jurídica de los internos para apoyar el trabajo de los especialistas del área de Control Penal.

El alcance de este trabajo abarca el diseño y la implementación de la capa de presentación de las funcionalidades que dentro del control penal se encargan de llevar el registro de las decisiones del proceso penal que son tomadas sobre un individuo privado de libertad y la gestión de la ejecución de la pena para aquellos individuos que cuentan con una sentencia definitivamente firme.

Para realizar estas funciones mencionadas fue necesario utilizar, dentro del marco arquitectónico del proyecto, una metodología para la construcción de la capa de presentación así como las herramientas necesarias para generar los artefactos. Finalmente se obtuvo el diseño e implementación, que según las validaciones realizadas con los clientes y usuarios finales, satisfacen sus necesidades reales.

- Sistema Penitenciario
- Control Penal
- Situación Jurídica
- Implementación
- Capa de Presentación
- JEE (Java Platform, Enterprise Edition)

## TABLA DE CONTENIDOS

---

<b>AGRADECIMIENTOS</b> .....	I
<b>DEDICATORIA</b> .....	II
<b>RESUMEN</b> .....	III
<b>PALABRAS CLAVE</b> .....	IV
<b>TABLAS Y FIGURAS</b> .....	VII
<b>INTRODUCCIÓN</b> .....	1
<b>DESARROLLO</b> .....	6
<b>1 Descripción detallada del problema.</b> .....	6
1.1 Decisiones del proceso penal .....	6
1.2 Ejecución de la pena .....	6
1.3 ¿Cómo se llevan en la práctica esos procesos? .....	7
1.4 Aporte de la solución, beneficios, resultados .....	7
<b>2 Arquitectura, herramientas y técnicas utilizadas.</b> .....	8
2.1 Arquitectura del sistema .....	8
2.1.1 Arquitectura Cliente-Servidor .....	8
2.1.2 Arquitectura en capas .....	8
2.1.3 Estructura de paquetes .....	10
2.1.4 Patrones utilizados .....	11
2.2 Ambiente de Desarrollo Integrado .....	13
2.2.1 Eclipse .....	13
2.3 Tecnologías .....	14
2.3.1 JEE .....	14
2.3.2 Spring .....	14
2.3.3 JSON y JSON-RPC .....	14

2.3.4 JavaScript.....	15
2.3.5 AJAX.....	15
2.3.6 Dojo toolkit.....	16
2.4 Herramientas .....	16
2.4.1 Visual Paradigm Suite .....	16
2.4.2 ER/Studio.....	16
2.5 Flujo de trabajo para Programador de Interfaz Usuario.....	17
<b>3 Diseño de la solución .....</b>	<b>22</b>
3.1 Definición de casos de uso y actores del sistema.....	22
3.2 Agrupación de funcionalidades por paquetes .....	23
3.3 Diseño de la solución por funcionalidades .....	24
3.3.1 Funcionalidades de las decisiones sobre el proceso penal .....	25
3.3.2 Funcionalidades de ejecuciones.....	31
3.3.3 Funcionalidades de los delitos.....	34
3.3.4 Funcionalidades de unificar penas .....	36
3.3.6 Funcionalidades del cómputo.....	43
3.4. Interacción entre capa de presentación y negocio.....	47
3.5 Componentes Reutilizables.....	50
3.6 Modelo físico de datos .....	52
<b>BIBLIOGRAFÍA.....</b>	<b>55</b>
<b>ANEXOS.....</b>	<b>56</b>
<b>GLOSARIO .....</b>	<b>63</b>

**Figuras**

Figura 1: Arquitectura Cliente Servidor.....	8
Figura 2: Arquitectura en tres capas .....	9
Figura 3: Estructura de paquetes del sistema.....	10
Figura 4: Flujo del Patrón MVC.....	11
Figura 5: Patrón Facade .....	12
Figura 6: Tecnologías que integra Ajax.....	15
Figura 7: Flujo de trabajo para PIU .....	17
Figura 8: Diagrama de Casos de Uso del Sistema.....	22
Figura 9: Paquetes del módulo Situación Jurídica .....	23
Figura 10: Vista Procesos Legales.....	26
Figura 11: Relación de dependencia entre las vistas de la Decisión Judicial.....	26
Figura 12: Gestionar Decisión Judicial .....	27
Figura 13: Decisión Suspensión Condicional del Proceso.....	28
Figura 14: Decisión Sentencia Condenatoria .....	28
Figura 15: Decisión que implica Egreso o fin del Proceso .....	28
Figura 16: Diagrama extensiones web de las Decisiones Judiciales .....	29
Figura 17: Relación de dependencia entre las vistas que componen la Ejecución .....	30
Figura 18: Vista de la ejecución.....	32
Figura 19: Vista nueva ejecución.....	33
Figura 20: Vista delitos cometidos.....	35
Figura 21: Relación de dependencia entre acumular ejecución y los documentos asociados .	36
Figura 22: Vista Acumular Ejecuciones.....	37
Figura 23: Diagrama extensiones web de la Ejecución de la Pena.....	38
Figura 24: Vista de la condena .....	40
Figura 25: Relación de dependencia entre la nueva pena y los documentos asociados.....	40
Figura 26: Vista Gestionar Condena.....	41

Figura 27: Diagrama extensiones web de la Condena .....	42
Figura 28: Vista Gestionar Cómputo de la Pena.....	44
Figura 29: Vista Gestionar Cómputo.....	45
Figura 30: Diagrama extensiones web del Cómputo de la Pena .....	46
Figura 31: Clases del diseño de Decisiones Judiciales para la capa de presentación .....	47
Figura 32: Clases del diseño de Ejecución de la Pena para la capa de presentación.....	48
Figura 33: Diagrama de clases del diseño de la lógica del negocio.....	49
Figura 34: Componente Documentos Asociados .....	50
Figura 35: Componente TribunalTable .....	51
Figura 36: Modelo físico de datos .....	52

## **Tablas**

Tabla 1: Caso de Uso CRUD-D decisión sobre proceso penal.....	25
Tabla 2: Caso de Uso: CRUD ejecuciones.....	31
Tabla 3: Caso de Uso: CRUD delitos .....	34
Tabla 4: Caso de Uso: Unificar penas .....	36
Tabla 5: Caso de Uso: CRUD-D pena.....	39
Tabla 6: CRUD cómputo .....	43

A partir de la llegada a la presidencia de la República Bolivariana Venezolana del Comandante Hugo Rafael Chávez Frías, se inició una ola de transformaciones sociales, por lo que a finales del año 1999 se aprobaron nuevos cambios en la Constitución de la República Bolivariana de Venezuela, los cuales estaban encaminados a una mejora de la sociedad. En este marco nace el proyecto de Humanización del Sistema Penitenciario, producto al artículo 272 de la constitución el cual estipula: “El estado garantizará un sistema penitenciario que asegure la rehabilitación del interno o interna y el respeto a sus derechos humanos. Para ello, los establecimientos penitenciarios contarán con espacios para el trabajo, el estudio, el deporte y la recreación, funcionarán bajo la dirección de penitenciaristas profesionales con credenciales académicas universitarias (...)” (Orizondo 2006). La cooperación brindada por Cuba en este campo incluye la atención a la salud de los internos, asesoría especializada y el desarrollo de un sistema informático para gestionar y automatizar los procesos penitenciarios conocido por Sistema de Gestión Penitenciaria (SIGEP).

El SIGEP constituye la solución de software para la informatización de la gestión de los privados de libertad de la República Bolivariana de Venezuela. El sistema, como solución integral para la informatización de los procesos penitenciarios, abarca varias de las especialidades carcelarias dentro de las que se encuentran: Control Penal, Tratamiento, Custodia y Seguridad, Administración Penal y Salud.

Control Penal es la especialidad penitenciaria cuya función es llevar el control de la documentación legal de los privados de libertad, tanto de procesados como de penados. Los especialistas de esta área, utilizando la información jurídica de cada individuo, deben alertar sobre el cumplimiento de los términos procesales y de la ejecución de la pena. Los tribunales constituyen el principal proveedor de esta información.

Control Penal es una de las áreas claves de un establecimiento penitenciario. De la eficiencia del funcionamiento de esta, depende en buena medida, la garantía de preservar los derechos de los privados de libertad.

En la actualidad, los establecimientos penitenciarios de la República Bolivariana de Venezuela, no cuentan con una organización adecuada de los procesos de Control Penal. Esta especialidad no está formalmente creada, solo algunas funciones son asumidas por los especialistas de Registro y Control y cada penal lleva la información de una manera dispersa y no uniforme.

El expediente se conforma solo con los documentos que envía el tribunal y para obtener una información hay que reconstruir la historia procesal del individuo de manera manual. Por tal motivo, muchos casos de anomalías legales no son detectados a menos que se revise exhaustivamente toda la documentación almacenada en el expediente, que puede ser voluminosa.

Teniendo en cuenta esta problemática, el SIGEP concibe un subsistema para el área de Control Penal y dentro de este, un módulo para el manejo de la situación jurídica de los privados de libertad. Este módulo permite llevar el control de los procesos penales que puede afrontar un individuo, las fases (control, juicio, apelación, casación, revisión) por las que transcurren estos procesos y las decisiones judiciales que se toman en cada fase. Además se registra lo relacionado con la ejecución de la pena, etapa de la que se recogen los datos referidos a la sentencia, la pena impuesta y el cómputo de la pena.

Durante la etapa de definición del sistema se obtuvieron los requisitos y se realizó el análisis del módulo situación jurídica. Una vez planteados estos requisitos de software, surge la necesidad de obtener un diseño de la solución que permita la posterior implementación.

Para el desarrollo de este sistema se realizó una solución Web sustentada por una arquitectura de software de tres capas, modelo que permite separar las tareas de implementación y que estas sean ejecutadas por programadores especializados en cada una de ellas, teniendo en cuenta la complejidad de las mismas y las distintas tecnologías que se requieren integrar para construir cada capa arquitectónica.

En este contexto y una vez obtenidos los requisitos del software, surge la necesidad de diseñar e implementar la capa de presentación, que integrada al resto de las capas arquitectónicas de la aplicación, den respuesta a los requisitos funcionales planteados para el control de la ejecución de la pena y las decisiones sobre el proceso penal, correspondientes al módulo Situación Jurídica del Subsistema Control Penal.

Dadas las condiciones mencionadas con anterioridad se plantea como **Objetivo General**: Diseñar e implementar la capa de presentación del módulo Situación Jurídica del Subsistema Control Penal, correspondiente a las funcionalidades que garantizan el control de la ejecución de la pena y el registro de decisiones judiciales durante el proceso penal.

Para darle solución al presente problema, se define como **Objeto de Estudio**: Diseño e implementación de aplicaciones de gestión para la web basada en JEE y tecnologías Open Source.

Por lo que se especifica el siguiente **Campo de Acción**: Ingeniería de software, disciplinas de diseño e implementación, Lenguaje Unificado de Modelado (UML), tecnologías web, programación en java, frameworks, arquitectura de software.

Posibles resultados: Diseño e implementación de las funcionalidades de la capa de presentación de las decisiones del proceso penal y la ejecución de la pena del módulo Situación Jurídica del Subsistema Control Penal.

Para darle cumplimiento al objetivo general se han trazados las siguientes **Tareas a Cumplir**:

- Diseñar la solución a partir de los requisitos funcionales especificados, utilizando las herramientas, tecnologías y la arquitectura definida para el SIGEP.
- Implementar la solución diseñada.
- Refinar el modelo de datos a partir de las consideraciones del diseño detectadas.
- Realizar casos de prueba a la interfaz para verificar su correcto funcionamiento.

### **Actualidad e importancia del trabajo.**

La automatización del proceso de la Ejecución de la Pena y las Decisiones Judiciales permite mantener la información disponible en todo momento, para los especialistas o Funcionarios de Control Penal (FCP), así tendrán conocimiento y control sobre las fases por las que transcurre el proceso penal de un interno, las decisiones tomadas sobre cada una de estas y la ejecución de la pena. La utilización de la solución de software permitirá a los especialistas de esta área, utilizar de manera adecuada la información jurídica de cada individuo, de modo que dará correcto cumplimiento a los términos procesales y la ejecución de la pena.

### **Métodos, herramientas y tecnologías para resolver el problema.**

Como metodología de referencia para guiar el trabajo de implementación, se utilizó lo que propone RUP para la disciplina de implementación. No obstante el equipo de proyecto diseñó un flujo de trabajo para los programadores de interfaz de usuario que permitió adaptar lo propuesto por RUP al esquema arquitectónico seleccionado para desarrollar la aplicación.

## Herramientas de modelado

Las herramientas de modelado de software utilizadas para generar y representar los artefactos del sistema a realizar son:

- **ER/Studio:** *Herramienta para y consultar el modelo de datos*
- **Visual Paradigm Suite 3.1:** *Herramienta para realizar el modelado de los procesos a desarrollar.*

## Herramienta de desarrollo

Las herramientas de desarrollo utilizadas para llevar a cabo el desarrollo de las funcionalidades son:

- **Eclipse SDK:** *Ambiente de desarrollo integrado.*
  - **WTP (Web Tools Platform):** *Plataforma de herramientas Web.*
  - **SpringIDE:** *Plug-in para la configuración de los archivos de Spring Framework.*
  - **SubEclipse:** *Plug-in para la integración con el controlador de versiones Subversión.*
  - **AJDT (AspectJ Development Tools):** *Plug-in que provee herramientas para el desarrollo orientado aspectos, usando AspectJ Framework.*
- **Contenedor Web (Apache Tomcat):** *Contenedor Web para desplegar la aplicación.*
- **Subversión:** *Herramienta para la gestión de versiones.*

## **Tecnologías**

Las tecnologías utilizadas para el desarrollo de las funcionalidades son:

- **Spring MVC:** *Módulo MVC (Modelo Vista Controlador) de Spring Framework para la web.*
- **JSON:** *(Javascript Object Notation) formato ligero para intercambio de datos.*
- **JSON-RPC:** *(Javascript Object Notation – Remote Procedure Call) Llamadas a métodos remotos, haciendo uso JSON para el intercambio de datos.*
- **JavaScript:** *Lenguaje script utilizado para unir el conjunto de tecnologías usados en la web. .*
- **Ajax:** *Conjunto de tecnologías, usando como método de transporte el XMLHttpRequest.*
- **Dojo Toolkit:** *Librería de clases javascript.*

### **1 Descripción detallada del problema.**

Uno de los objetivos del SIGEP es gestionar información que permita conocer la situación jurídica de los individuos(as) que se encuentran bajo la custodia del Sistema Penitenciario. Un individuo(a) puede tener varios Expedientes Penitenciarios, cada uno de ellos correspondiente al tránsito de este por el Sistema Penitenciario (Ver Anexo1). Cada Expediente Penitenciario puede tener varios procesos judiciales por los cuales el individuo(a) puede estar penado o procesado, un estado, las detenciones y la ejecución de la pena correspondiente. De cada proceso judicial se conocen los delitos, los abogados y el Órgano del Ministerio Público que lo atienden, una síntesis del hecho por el que haya sido sentenciado el individuo(a), la jurisdicción del proceso y las etapas por las que transcurre (Ver Anexo 2). Las etapas dependiendo de su tipo tienen asociadas un conjunto de decisiones que son emitidas al terminar cada fase (Ver Anexo 3). Los datos de la ejecución de la pena se registran a nivel de Expediente Penitenciario, puesto que la ejecución de la pena es única para todos los procesos judiciales con sentencia definitivamente firme (Ver Anexo 4). El cómputo se registra como parte de la etapa de Ejecución de la pena y se mantiene un historial de las modificaciones que este ha sufrido durante la ejecución de la pena (Ver Anexo 5).

#### **1.1 Decisiones del proceso penal**

Las decisiones son determinaciones que toman los tribunales sobre el proceso penal que se le sigue a un individuo y que pueden darle continuidad a este proceso, cerrarlo o pasarlo a ejecución a través de una sentencia condenatoria. Las decisiones son propias de cada fase y solo se puede dictar una decisión por etapa. Para aquellas que pasan el proceso a ejecución hay que especificar los delitos probados y la síntesis del hecho.

#### **1.2 Ejecución de la pena**

La ejecución de la pena de los individuos es llevada por un tribunal de ejecución. De ella se registran además de los datos del tribunal, la sentencia, los delitos probados, la condena, el cómputo, la pena principal impuesta y las penas accesorias.

El cómputo puede variar durante la ejecución de la pena por diversos motivos: redención de pena por buena conducta, revocatoria de redención de la pena, la acumulación de ejecuciones y la rectificación de errores. Solo el último cómputo es el activo.

Puede suceder en casos aislados que un individuo tenga varios tribunales de ejecución. En tal situación lo que debe ocurrir es que ambas ejecuciones se unifiquen y solo un tribunal de ejecución se haga cargo de velar por el cumplimiento de la pena del privado de libertad. Cuando se unifiquen las ejecuciones se debe actualizar el cómputo y la pena resultante de la acumulación.

### **1.3 ¿Cómo se llevan en la práctica esos procesos?**

La DGCR (Dirección General de Custodia y Rehabilitación del Recluso) no contaba con algún tipo de aplicación informática que se ajustara al problema de los Establecimientos Penitenciarios. Los expedientes de los privados de libertad se llevan mayormente en papel, salvo algunas iniciativas aisladas que utilizan hojas de cálculo para mantener alguna información. Esta es escasa y no siempre es enviada por el tribunal por lo que es frecuente que el expediente penal se encuentre desactualizado. En estas condiciones, conocer el estado del proceso penal de los individuos puede ser complicado, pues requiere la revisión completa de la documentación enviada por los tribunales. Sin una información sistematizada y ordenada resulta difícil identificar anomalías en la situación jurídica de los individuos y por tanto, la institución carece de mecanismos que alerten por ejemplo, sobre casos particulares de retardo procesal.

### **1.4 Aporte de la solución, beneficios, resultados**

La solución aporta un grupo de componentes y funcionalidades que pueden ser reutilizadas por otros programadores, con el objetivo de ganar tiempo de desarrollo. Además brinda una actualización del flujo de trabajo más ajustado a las necesidades del Programador de Interfaz Usuario (PIU), con las actividades necesarias para implementar las funcionalidades de la capa de presentación de una aplicación web, con el entorno de arquitectura del SIGEP.

El resultado observado luego de la implantación de la solución, en prisiones venezolanas como parte de las pruebas piloto realizadas, fue el esperado, por lo que se obtuvo una mayor organización, rapidez y seguridad en los procesos, siendo beneficiados el sistema penitenciario en su conjunto.

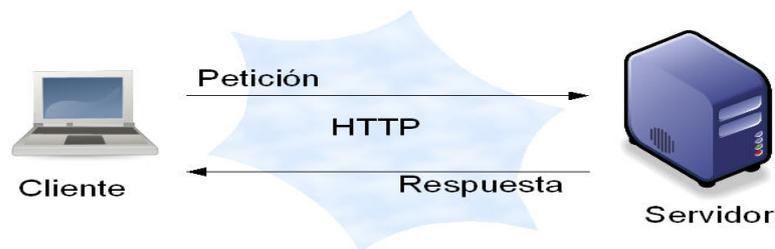
## 2 Arquitectura, herramientas y técnicas utilizadas.

### 2.1 Arquitectura del sistema

La arquitectura definida para el sistema, esta basada en los estilos arquitectónicos Cliente-Servidor y Arquitectura en Capas, la cual, cuenta con la capa Presentación, Lógica de Negocio y Acceso a Datos, representadas de la siguiente forma.

#### 2.1.1 Arquitectura Cliente-Servidor

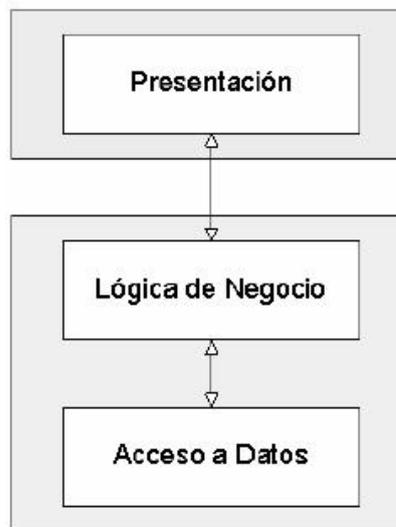
La arquitectura cliente servidor se encuentra dentro de la clasificación de estilo de llamada y retorno. El cliente y el servidor generalmente están localizados en diferentes sistemas, sin embargo pueden encontrarse en el mismo. El cliente es la entidad que hace la petición por un servicio. El servidor es la entidad que provee el servicio correspondiente a la petición. El servicio debe procurar el resultado, el cual es retornado.



**Figura 1: Arquitectura Cliente Servidor**

#### 2.1.2 Arquitectura en capas

La arquitectura en capa define el estilo en capas como una organización jerárquica de forma tal que cada una proporciona servicios a la inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. Lo que posibilita un diseño basado en niveles de abstracción creciente, posibilitando a los implementadores, particionar un problema en una secuencia de pasos incrementales. Este estilo de desarrollo en varios niveles, facilita que en caso que ocurra algún cambio, sólo se tendrían que realizar las correcciones necesarias en el nivel requerido sin tener que revisar código de otros niveles. En dicha arquitectura a cada nivel se encomienda una tarea simple, lo que permite que el diseño pueda ampliarse con facilidad en caso de que las necesidades aumenten.



**Figura 2: Arquitectura en tres capas**

**Capa de presentación:** Es la encargada de interactuar con el usuario, obtener, validar y enviar los datos al servidor. Debido al uso de Spring MVC estará compuesta por tres tipos de objetos: Modelo, Vista y Controlador.

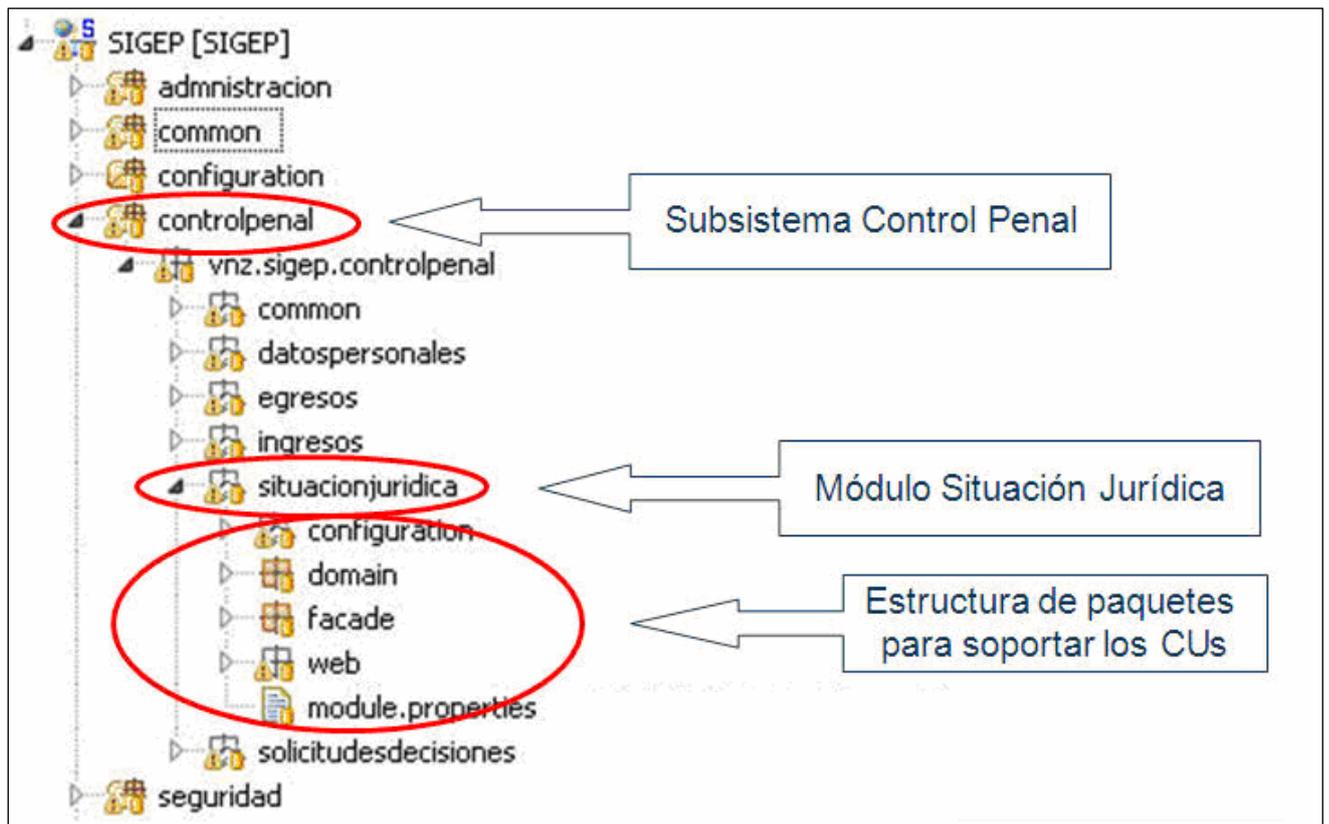
- **Controlador:** Es el responsable de procesar las entradas del usuario en forma de peticiones HTTP (Hyper Text Transfer Protocol), invocando las funcionalidades necesarias, expuestas por la capa de lógica de negocio y devolviendo un modelo requerido para ser mostrado.
- **Modelo:** Contiene los datos resultantes de la ejecución de la lógica de negocio, los cuales deberían ser mostrados en la respuesta.
- **Vistas:** Son las encargadas de mostrar los datos del modelo que han sido suministrados por un controlador como respuesta a una petición. La forma de mostrar el modelo podrá ser con diferentes tipos de vistas como JSP (Java Server Page), HTML (Hyper Text Markup Language), documentos PDF, documentos de Excel e imágenes.

**Capa de Lógica de Negocio:** Es la encargada de procesar toda la información que será persistida en la base de datos o mostrada en la capa de presentación. La definición de arquitectura del sistema propone que los objetos de dominio no presenten ningún tipo de lógica de negocio, dejando caer esta responsabilidad sobre los objetos de negocio, permitiendo usar a los objetos de dominio como Transfer Objects que se mueven entre las capas arquitectónicas de la aplicación.

**Capa de Acceso a Datos:** Es la encargada de persistir, consultar, actualizar y eliminar los objetos de dominios recibidos de la capa lógica.

### 2.1.3 Estructura de paquetes

La arquitectura del sistema posee una estructura de paquetes que divide tanto lógicamente como físicamente las tres capas arquitectónicas y junto a ello los recursos que se generan.



**Figura 3: Estructura de paquetes del sistema**

Esta estructura de paquete permite mantener una buena organización de las actividades a realizar dentro de las diferentes capas. La capa raíz llamada SIGEP, agrupa todos los subsistemas de la aplicación, dentro de los cuales se encuentran los diferentes módulos, los cuales están estructurados de forma tal que estén presente las tres capas lógicas. El paquete web contendrá todos los recursos que se generen dentro de la capa de presentación, ya sean Controllers, Commands, Validator y PropertyEditors. El paquete configuration contendrá las configuraciones del mismo, en forma de mapeo de peticiones, controladoras y vistas. Una vista más detallada de la organización de los paquetes que sustentan la solución de la implementación del módulo situación jurídica se muestra en el Anexo 6.

#### 2.1.4 Patrones utilizados

“Un patrón proporciona una solución común a un problema común en un contexto dado” (Booch; Rumbaugh & Jacobson 1999). A modo de reflexión podemos decir que un patrón describe un problema que ocurre frecuentemente dentro de un entorno y el núcleo de la solución al problema de forma que pueda ser reutilizado.

Los patrones utilizados durante la solución fueron:

- MVC
- FACADE
- Alta cohesión y Bajo acoplamiento
- Arquitectura en n Capas

**Modelo Vista Controlador (MVC):** Es un patrón de arquitectura de software que separa los datos de una aplicación en la interfaz, la lógica y el modelo de datos en tres componentes distintos. Este patrón está íntimamente ligado al framework Spring MVC que se propuso utilizar para la solución y debido a esto se ha aprovechado el uso del mismo.

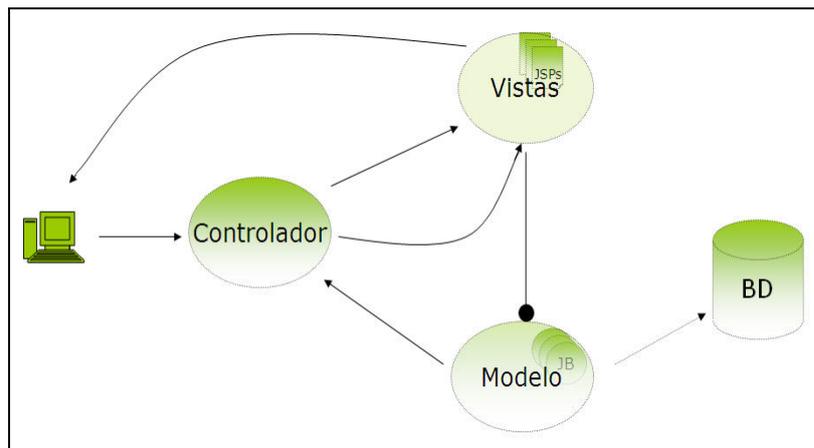
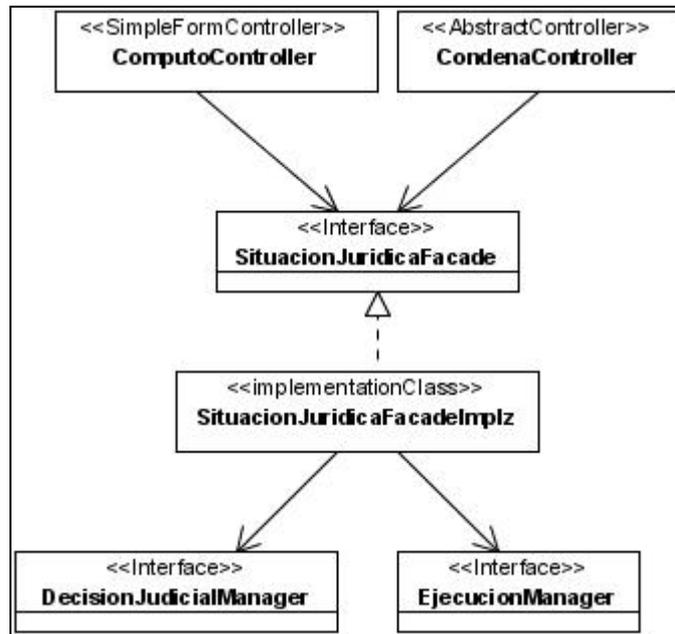


Figura 4: Flujo del Patrón MVC

**Facade:** Este patrón provee una interfaz unificada y sencilla que funciona como intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas. Limita en gran escala las dependencias entre objetos y simplifica la complejidad del sistema. En el entorno de la aplicación es usado para brindar servicios entre la capa de presentación y la capa lógica.



**Figura 5: Patrón Facade**

**Alta Cohesión y Bajo Acoplamiento:** Estos patrones pueden ser vistos de forma separada, aunque están fuertemente ligados, de hecho si aumenta demasiado la cohesión del sistema se obtiene un bajo acoplamiento entre las clases, y por el contrario si se reduce el acoplamiento, aumentada la cohesión.

El patrón **Alta Cohesión** establece que la información almacenada en una clase debe de ser coherente en la mayor medida posible relacionada con la clase, con esto queremos decir las clases obtenidas para la solución manejan solamente información referente a la acción que realiza la misma a la misma y limitan sus funcionalidades a realizar actividades concretas.

El patrón **Bajo Acoplamiento** es la manera de tener las clases lo menos ligadas entre sí cuando existen demasiadas relaciones, de forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima afectación posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre clases. Este patrón es utilizado dentro de la implementación de los controladores utilizados.

## **2.2 Ambiente de Desarrollo Integrado.**

El Ambiente de Desarrollo Integrado utilizado para la elaboración del sistema es Eclipse, el mismo incorporará un conjunto de plugins para trabajar con parte de las tecnologías usadas en el desarrollo de la aplicación.

### **2.2.1 Eclipse**

Eclipse es una plataforma de software de código abierto independiente de una plataforma para desarrollar. El ambiente de desarrollo integrado de Eclipse emplea módulos (*plug-in*) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos donde las funcionalidades están todas incluidas.

### **WTP (Web Tools Platform)**

Es una plataforma de herramientas web de Eclipse, provee varias APIs para desarrollo de aplicaciones sobre la Web y JEE. Incluyen editores gráficos de código fuente para una variedad de lenguajes, asistentes y aplicaciones incorporadas para simplificar el desarrollo de servicios web, además de herramientas y APIs para soportar el despliegue, ejecución y prueba de aplicaciones.

Soporta integración con servidores Web dentro de Eclipse como ambiente de ejecución de primera clase para aplicaciones web. También incluye la configuración de servidores y su asociación con los proyectos web, permitiendo la depuración sobre el servidor de los recursos y las clases.

### **Spring IDE**

Sirve como interfaz gráfica para la configuración de los archivos usados por Spring Framework. Permite el completamiento de etiquetas, valores de atributos y elementos en estos archivos de configuración.

### **SubEclipse**

Adiciona integración para el control de versiones Subversion, permitiendo operaciones de sincronización y actualización. Es muy útil para el desarrollo colectivo, en el que intervienen un conjunto de desarrolladores trabajando sobre el mismo proyecto, poniendo a disposición del equipo de desarrollo facilidades para el trabajo en equipo.

## 2.3 Tecnologías

### 2.3.1 JEE

La plataforma JEE es el nombre de un entorno o plataforma originaria de Sun Microsystems, para el desarrollo de aplicaciones empresariales utilizando el Lenguaje de programación Java y un conjunto de herramientas de desarrollo, JEE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, Enterprise Java Beans, Servlets, Portlets, Java Server Pages y varias tecnologías de servicios web, definiendo cómo coordinarlos.

### 2.3.2 Spring

Es un framework de peso ligero y código abierto que provee servicios empresariales en forma de POJOs<sup>1</sup>, interviene en todas las capas arquitectónicas de una aplicación y contribuye al desarrollo de aplicaciones JEE. Su principal novedad está en la inversión de control o inyección de dependencia que permite independizar las diferentes capas de una aplicación.

Para la implementación de la capa de interfaz usuario se hace uso como tecnología fundamental el módulo Spring Web MVC, uno de los principales módulos de Spring, el cual provee una implementación Modelo-Vista-Controlador para aplicaciones web permitiendo una separación de estos tres elementos y haciendo uso de una gran variedad de funcionalidades que ayudan a la solución del problema. Los restantes módulos de Spring Framework se muestran en el Anexo 7.

### 2.3.3 JSON y JSON-RPC

JSON, acrónimo de (JavaScript Object Notation), es un formato ligero para el intercambio de datos, es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. JSON-RPC (JavaScript Object Notation - Remote Procedure Call) es una tecnología que permite hacer llamadas a procedimientos o métodos desde el cliente web.

La utilización de JSON como formato de intercambio y JSON-RPC para hacer llamadas a métodos remotos agiliza el intercambio de datos, y reducen el uso de peticiones y controladores innecesarios que solo eran utilizados para obtener y validar datos.

---

<sup>1</sup> POJO: Es el acrónimo de Plain Old Java Object. Clase del lenguaje de programación Java. Este nombre se les da a las clases que no son de algún tipo especial (EJBs, Java Beans, etcétera) y no cumplen ningún otro rol ni implementan alguna interfaz especial.

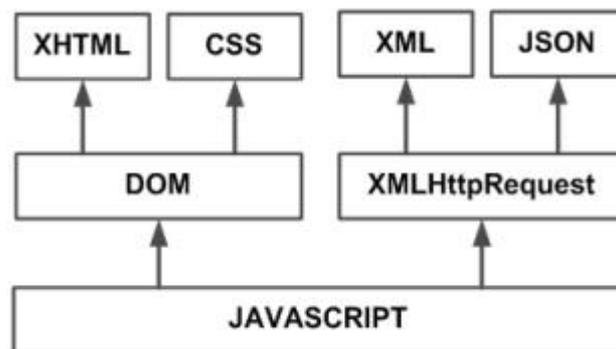
### 2.3.4 JavaScript

Lenguaje de programación interpretado por un cliente web, presenta una sintaxis parecida Java y C++. No es un lenguaje orientado a objetos, debido a que no dispone de Herencia y otros paradigmas de la Programación Orientada a Objetos, es un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base y extendiendo su funcionalidad. Permite interactúa con diferentes tecnologías web como CSS, DOM, JSON y está condicionado en la unión de las diferentes tecnologías web.

### 2.3.5 AJAX

Es la unión de varias tecnologías que se desarrollan de forma autónoma y que se unen de formas nuevas y sorprendentes, entre estas tecnologías podemos encontrar:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.



**Figura 6: Tecnologías que integra Ajax**

AJAX permite al usuario una mejor interacción con las aplicaciones web, evitando recargar en gran medida la aplicación cada vez que se realiza una petición al servidor web, por lo que el intercambio de información con el usuario se produce en un segundo plano.

### **2.3.6 Dojo toolkit**

Framework AJAX o librería de clases (*Asynchronous JavaScript + XML* o JavaScript asíncrono + XML) el cuál brinda una variedad de clases y widgets para facilitar el desarrollo de aplicaciones web. Dojo puede ser interpretado por diferentes navegadores web y posee un sistema de empaquetado muy parecido al del JDK de java.

La gran variedad de clases componentes y widget, es un punto muy importante en la elaboración en la capa presentación de la aplicación, posibilita a los programadores de interfaz una serie de funcionalidades y elementos dinámicos que facilita la programación en el cliente.

## **2.4 Herramientas**

El lenguaje Unificado de modelado (UML) utilizado para visualizar, especificar, construir y documentar un sistema de software por medio de conceptos orientados a objetos, ha constituido la base principal para la construcción de diagramas en las diferentes herramientas de modelados.

### **2.4.1 Visual Paradigm Suite**

Visual Paradigm Suite es un conjunto de herramientas de modelado que permiten realizar el modelado dentro del proceso de desarrollo de software.

A continuación se mencionan las principales herramientas presentes dentro de esta suite:

- Visual Paradigm for UML Enterprise Edition
- Visual Paradigm Smart Development Environment (SDE) Enterprise Edition.
- Visual Paradigm DB Visual Architect Frameworks

### **2.4.2 ER/Studio**

ER/Studio es una herramienta de modelado para diseñar bases de datos. Ayuda a descubrir, documentar y reutilizar los datos activos. Con un soporte de ida y vuelta de bases de datos, los arquitectos de datos tienen el poder para analizar a donde las fuentes de datos existentes tan bien como el diseño e implementación de bases de datos de alta calidad.

## 2.5 Flujo de trabajo para Programador de Interfaz Usuario.

Para obtener cada uno de los artefactos generados dentro del rol PIU es necesario establecer un orden en las actividades a realizar. Estas actividades especifican cómo y cuándo realizar cada artefacto por los integrantes del equipo de desarrollo. Este grupo de actividades, dio lugar al **Flujo de trabajo para Programador de Interfaz Usuario** dentro de la arquitectura del SIGEP. A continuación se hace una breve descripción de las actividades que integran el flujo de trabajo utilizadas para darle solución a la implementación del módulo Situación Jurídica. Es importante mencionar que se realizó una variante para reajustar algunas actividades que no fueron utilizadas, debido a la extensión del flujo dentro de la capa de presentación, por lo que solo se hará referencia a las utilizadas.

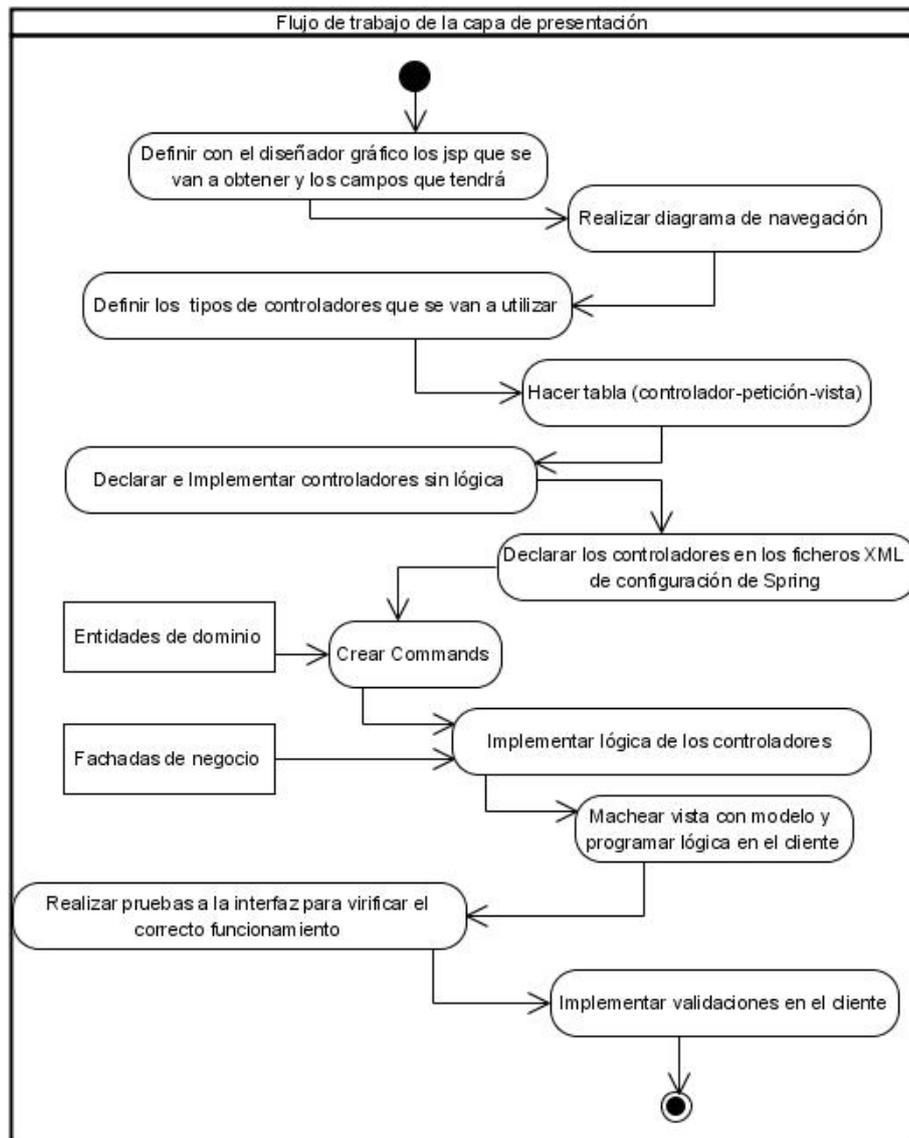


Figura 7: Flujo de trabajo para PIU

**Actividad 1:** Definir con el diseñador gráfico las jsp (Java Server Pages) que se deben obtener y los campos que tendrán.

**Descripción:** Definiremos vista a las distintas pantallas que provee la descripción del caso de uso, y estados a los cambios en ciertos campos o conjuntos de estos. Inicialmente se definirán jsp generales que contienen la información necesaria de cada vista, y estas serán debidamente fraccionadas en jsp más pequeñas según se entienda. A partir de los cambios de estado de una vista, el programador y el diseñador gráfico llegarán al acuerdo de cuales son las jsp que quiere por separado, para que el programador pueda trabajar en cada uno de estos segmentos de jsp de manera individual, si le es necesario.

**Actividad 2:** Realizar diagrama de navegación (vistas e interacciones cliente servidor)

**Descripción:** Dibujar un diagrama de navegación que muestre las interacciones entre las vistas y el servidor, que de alguna manera reflejen las posibles peticiones y las correspondientes respuestas. A partir de este diagrama se obtienen los nombres lógicos de las vistas y los nombres de las URLs de las peticiones.

**Actividad 3:** Dado el diagrama anterior, definir tipos de controladores que deben ser utilizado.

**Descripción:** Identificar el tipo de controlador que se utilizará para resolver cada petición y sus respectivos nombres. Para esto se tomará en cuenta el flujo y el tiempo de vida dentro del caso de uso de cada uno de los controladores donde se utilizarán instancias de `AbstractController` para las peticiones simples que no necesitan de ninguna lógica compleja, los `MultiActionController`, generalmente, para las peticiones asíncronas que apoyen o de soporte con peticiones al servidor para complementar otras peticiones, y los `SimpleFormController` donde se trabaje directamente con formularios.

**Actividad 4:** Hacer tabla (controlador-petición-vista). Se definen los nombres de las peticiones y las respuestas por controlador.

**Descripción:** Realizar una tabla con tres columnas, cuyos encabezados son [petición, controlador y vistas], dejar reflejados, las peticiones, los controladores y las posibles vistas a devolver en cada caso.

**Actividad 5:** Declarar e Implementar controladores sin lógica.

**Descripción:** Crear las clases de cada uno de los controladores que se determinaron en las actividades anteriores, declarar los métodos que se deberán sobrescribir o implementar en cada caso, así como declarar los atributos y dependencias con otras clases que tenga, tales como fachada de negocio. Comentar, a partir de las especificaciones establecidas para generar adecuadamente los javadocs, el encabezado de cada clase a implementar, así como cada método declarado.

**Actividad 6:** Declarar los controladores en los ficheros xml de configuración de Spring.

**Descripción:** Los controladores se mapearan en el correspondiente fichero sigep-subsistema-modulo-servlet.xml, que se encuentra en el paquete vnz.sigep.subsistema.modulo.configuration (y también en el fichero sigep-subsistema-modulo-servlet-test.xml, que se encuentra en el mismo paquete; esto es para procurar tener un xml de configuración exclusivo a soportar y apoyar las pruebas de unidad necesarias a cada controlador posteriormente implementado). A partir de los HandlerMappings que se hayan declarado en el mismo fichero, se configurará la correspondiente petición o las peticiones con el controlador en el mismo.

**Actividad 7:** Crear Commands (posible interacción con el diseñador, necesitan las entidades para utilizarlas como Command) y se machean los atributos del jsp con los del command en caso de hacer falta.

**Descripción:** Realizar los command necesarios para crear un objeto desde un formulario en la WEB o enviar los datos para los jsp. Esto se debe a que cada atributo del objeto command toma el valor referente al campo en el formulario que contiene el mismo nombre del atributo. Los commands deben colocarse en el paquete vnz.sigep.[nombre\_del\_sistema].[nombre\_del\_modulo].web.command

**Actividad 8:** Implementar lógica de los controladores. (Posible interacción: debe estar definida las Fachadas de Negocio y las entidades, y se piden métodos necesarios en caso de que no existan.

**Descripción:** Dar solución a la implementación de las funcionalidades necesarias para resolver cada problemática que requieren los casos de usos.

**Actividad 9:** Crear las vistas, separarlas física y lógicamente a partir de las consideraciones que se hayan tomado en cuenta para crearlas.

**Descripción:** Definir las vistas para cada una de las peticiones a realizar. Es necesario destacar que existen algunos casos en que uno o mas jsp se encuentra formando parte de otra, esto se debe a la complejidad de la vista, la cual resulta mucho más sencillo si se fraccionan en jsp más pequeñas y luego se hace referencia a ellas de manera de inclusión con la expresión: `<%@include file="NombreDeLaVista.jsp"%>`

Los correspondientes ficheros JSP obtenidos deberán ser colocados en la carpeta `/WEB-INF/jsp/[subsistema]/[modulo/]` cuyos nombres deben de alguna manera clara y breve relacionarse o describir el fin para el cual han sido destinadas.

**Actividad 10:** Declarar las vistas. (Declarar el resolvedor de vistas, y mapearlas cuando se necesite.)

**Descripción:** A partir del resolvedor o los resolvedores de vistas que se hayan determinado utilizar se configurarán las vistas asociadas a cada controlador que haya sido implementado o que de alguna manera se asocie al caso de uso que se está programando (estas se mapearán en el fichero `sigep-subsistema-modulo-servlet.xml`, que se encuentra en el paquete `vnz.sigep.subsistema.modulo.configuration`).

**Actividad 11:** Vincular las vistas al modelo y programar lógica en el cliente

**Descripción: Vincular vista con modelo**

El amarre de los atributos del command con elementos html, se realizará fundamentalmente usando el tag de Spring bind, En el caso de los formularios, se recomienda usar para el amarre el tag de Spring FormTag, pues este permite ahorrar gran cantidad de código que sería necesario escribir usando el tag `<bind>`.

En ocasiones se necesita amarrar atributos del command con objetos javascript, y no con elementos html. Para estos casos se recomienda usar el formato JSON para el intercambio de información entre el cliente y el servidor. Mediante el formato JSON, los datos del cliente se amarran a los atributos del command usando el propertyEditor `BaseJSONPropertyEditor` o alguna subclase. Y en el cliente, se amarra el objeto JSON serializado con un objeto javascript usando el tag de Spring bind y la función `dojo.json.evalJson(jsonStr);`

## Programar lógica en el cliente

Se debe crear un archivo .js para cada jsp que necesite de código javascript. En ese archivo se deberán programar todas las funciones y objetos que se necesite en la jsp. A continuación se mencionan algunas consideraciones sobre estos archivo js:

- Deben ser ubicados en la carpeta WebContent/js/[nombre\_del\_subsistema]/[nombre\_del\_módulo], la primera línea de los archivos debe ser `dojo.provide("sigep.[nombre_del_subsistema].[nombre_del_módulo].[nombre_del_archivo]");` donde [nombre\_del\_archivo] es el nombre del archivo sin incluir la extensión (o sea, si el archivo es menu.js, en [nombre\_del\_archivo] solo se pone menu).
- En la página en que se va usar el archivo js debe ponerse:

```
<script>
```

```
    dojo.require("sigep.[nombre_del_subsistema].[nombre_del_módulo].[nombre_del_archivo]");
```

```
</script>
```

**Actividad 12:** Mostrar errores de validaciones del cliente y del servidor en el jsp. Externalización de ficheros y realizar pruebas a la interfaz para verificar el correcto funcionamiento e interacción de esta con el supuesto usuario final.

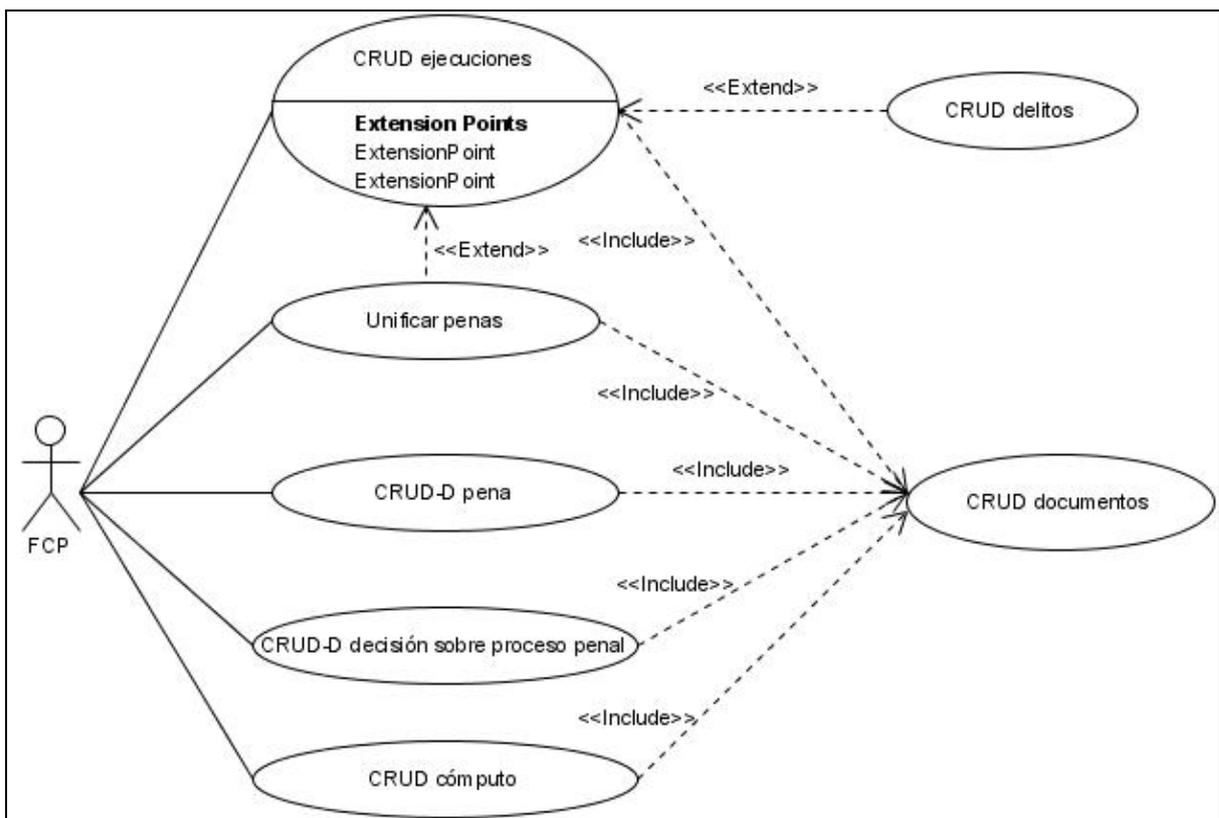
**Descripción:** Los mensajes de errores en el cliente identifican campos requeridos, datos incorrectos o muestran alertas en casos de cometerse posibles errores. Los mensajes de errores en el servidor alertaran sobre el suceso que se produjo, impidiendo que se realizara la acción deseada. Los mensajes a mostrar serán los establecidos por los lineamientos de control. Los ficheros de externalización de mensajes en los js deben ser ubicados en la carpeta: WebContent/js/[nombre\_del\_subsistema]/common/nls y para los jsp serán ubicados en la carpeta: SIGEP/[nombre\_del\_subsistema]/[nombre\_del\_módulo]/configuration.

### 3 Diseño de la solución

El módulo Situación Jurídica, perteneciente al subsistema Control Penal, permite gestionar la información sobre la situación jurídica de los individuos(as) que se encuentran bajo la custodia del Sistema Penitenciario. Este módulo contiene información sobre los procesos penales del individuo(a), las fases por las que transcurre cada uno de los procesos, las decisiones tomadas en cada fase y sobre la ejecución de la pena.

#### 3.1 Definición de casos de uso y actores del sistema

Una vez obtenidos los requisitos funcionales relacionados con el registro de decisiones y la ejecución se formaron los casos de uso que dan lugar a las funcionalidades a implementar.

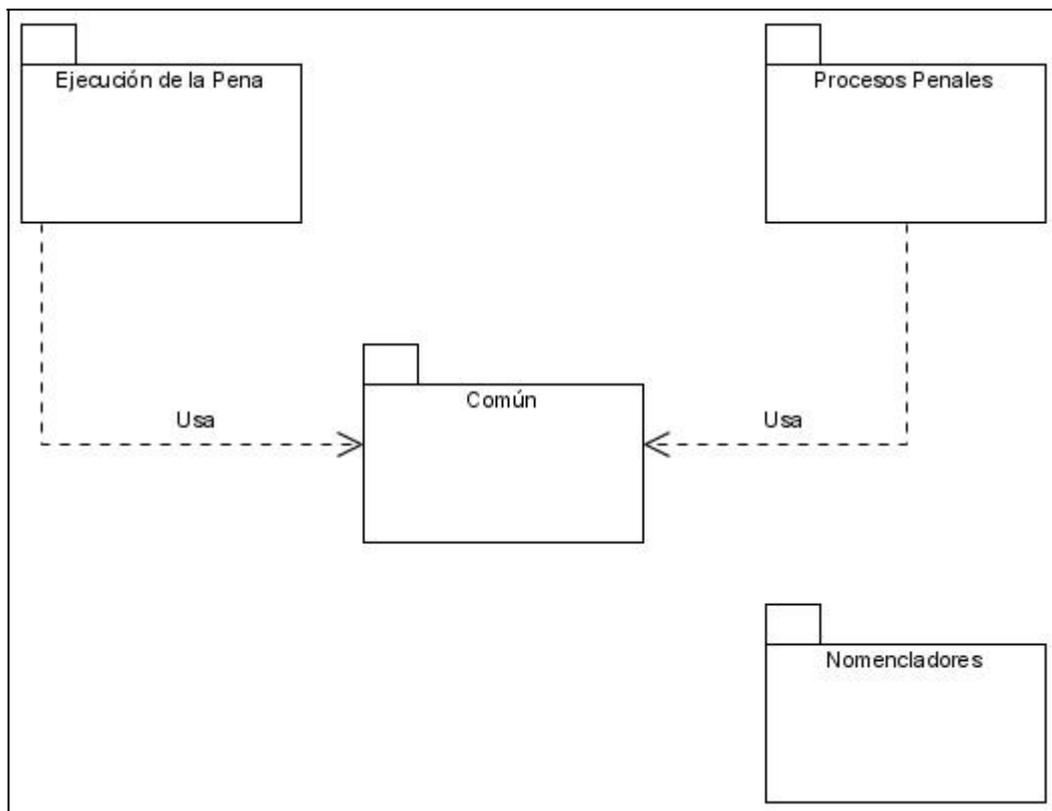


**Figura 8: Diagrama de Casos de Uso del Sistema**

Se identificó como actor del sistema, el Funcionario del Control Penal, el cual será el encargado de gestionar la información referente a las Decisiones Judiciales y la Ejecución de la Pena.

### 3.2 Agrupación de funcionalidades por paquetes

Bajo las condiciones mencionadas anteriormente se ha dividido el modulo en diferentes paquetes que organizan y agrupan los casos de usos que estén relacionados utilizando la arquitectura propuesta para el sistema. Una representación de esos paquetes se muestra en la siguiente figura.



**Figura 9: Paquetes del módulo Situación Jurídica**

El paquete **“Común”**: Se encarga de agrupar clases o funcionalidades que son comunes para todo el modulo, estas funcionalidades pueden llamarse desde otros paquetes con el objetivo de reutilizar código y aprovechar un mejor rendimiento de la aplicación. Cada módulo deberá agrupar sus funcionalidades comunes en un paquete como éste y estos a su vez podrán utilizar funciones comunes que estén en capas de un nivel más alto en el sistema.

El paquete **“Ejecución de la Pena”**: Agrupa aquellas clases que se utilizan para imponer las penas, realizar su cómputo y establecerle una condena al interno. Este paquete usa un grupo de funcionalidades que se encuentran en el paquete común.

El paquete “**Procesos Legales**”: Agrupa las clases pertenecientes al proceso legal, las fases por las que se transita el mismo y dentro de estas las Decisiones Judiciales que se toman en cada una, teniendo en cuenta que pueden cambiar el estado del proceso dependiendo del tipo de decisión. Para llevar a cabo su completo funcionamiento requiere de funciones contenidas en el paquete común.

El paquete “**Nomencladores**”: Agrupa las clases que brindan parámetros de configuración del sistema que son predefinidos con anterioridad.

### **3.3 Diseño de la solución por funcionalidades**

Para el diseño de la solución fue necesario generar un grupo de artefactos que modelan y describen las funcionalidades a realizar. Para ello se obtuvo el diagrama de casos de usos del sistema con sus respectivas descripciones, los diagramas de clases de la lógica de presentación, los diagramas de extensiones web y un modelo físico de datos refinado.

Cada caso de uso cuenta con un diagrama de extensiones web que brinda a modo general información de las interacciones cliente – servidor, representando las peticiones realizadas, las vistas a las que se retornará y el flujo de datos entre ellas. En este diagrama se identifican elementos tales como: vistas, formularios, javascripts usados, peticiones realizadas y los controladores que responden a las mismas. Las vistas tienen funcionalidades específicas que serán mostradas en detalles a nivel de caso de uso para su mejor entendimiento.

### 3.3.1 Funcionalidades de las decisiones sobre el proceso penal

Caso de Uso: CRUD-D decisión sobre proceso penal		
Objetivo	Registrar, modificar y consultar las decisiones tomadas sobre el proceso penal del individuo(a).	
Actores	FCP	
Resumen	El FCP indica que desea gestionar la decisión tomada en una fase del proceso penal. El sistema permite consultar, modificar y registrar una decisión judicial.	
Precondición	El FCP se ha identificado y autenticado correctamente ante el sistema y tiene permisos para utilizar este servicio. Se ha buscado y seleccionado el expediente del individuo(a). Se ha registrado un proceso penal. Se ha seleccionado una fase del proceso penal.	
Garantías	Mínimas	El sistema notifica al FCP cualquier evento que impida gestionar las decisiones sobre el proceso penal.
	De éxito	Se han registrado los datos de la decisión sobre el proceso penal.
Acción que inicia el caso de uso	El FCP selecciona una fase e indica que desea registrar una decisión sobre el proceso penal.	
Descripción del CRUD	El caso de uso tiene tres(3) flujos básicos: Registrar decisión judicial. Editar decisión judicial. Consultar decisión judicial.	
Relaciones	CU incluidos	CRUD documentos
	CU extensión	No aplicable
Requisitos especiales	De una fase sólo puede registrarse una decisión.	
Asuntos pendientes	¿Cuál es el límite máximo para suspender el proceso?	

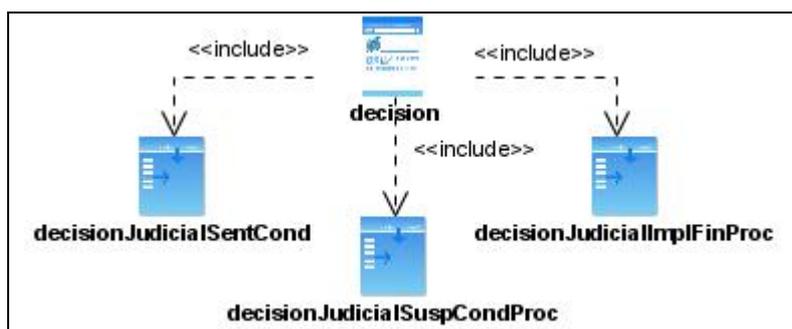
**Tabla 1: Caso de Uso CRUD-D decisión sobre proceso penal**

La figura 10: muestra la vista Procesos Legales mediante la cual se puede adicionar o eliminar una decisión judicial a la fase. Esta vista permite realizar otras funcionalidades tales como: gestionar procesos legales, unificar y desunificar procesos y gestionar fases, que no son objetos de estudio y por lo tanto no haremos referencias a ellas.

Procesos				< < 1 ... > >		Total: 1	
Proceso	Número de Expediente	Última Fase	Estado				
Causa 1	GP01-P-2006-019699	Control	Ejecución				
		Adicionar Proceso	Modificar	Eliminar	Detalles		
		Adicionar Fase	A Ejecución	Unificar	Desunificar		
Fases				< < 1 ... > >		Total: 1	
No	Número de Expediente	Fase del Proceso	Tribunal				
1	GP01-P-2006-019699	Control	Tribunal Sexto de Control del Circuito Judicial Penal Carabobo extensión Valencia				
		Modificar	Eliminar	Detalles	Decisión	Eliminar Decisión	

**Figura 10: Vista Procesos Legales**

La vista que gestiona la información de la decisión está compuesta por tres fragmentos de jsp que se muestran o se ocultan en dependencia del tipo de decisión a registrar. El diagrama que se muestra a continuación representación las relaciones entre dichas vistas.



**Figura 11: Relación de dependencia entre las vistas de la Decisión Judicial**

El registro de las decisiones se lleva a cabo mediante la vista que se muestra a continuación.

**DECISIONES DE CONTROL**

---

**Tipo**

**Observaciones**

---

**Documentos Asociados a la Decisión**

¿Es un Tribunal Itinerante el que emite la decisión?

**No. Documento**

**Tipo Documento**

**Fecha Elaboración**

**Fecha Recepción**

---

**Datos del Tribunal**

**Tribunal Itinerante**

**Estado**

**Circuito Judicial Penal**

**Extensión**

**Tipo de Tribunal**

**Número de Tribunal**

---

**Documentos** | < < 1 ... > > | **Total: 1**

No. Documento	Tipo Documento	Fecha Elaboración	Fecha Recepción	Tribunal
20-101	Boleta de excarcelación	01/05/2008	05/05/2008	Tribunal 206 de Control del Circuito Judicial Penal Área metropolitana

---

**Figura 12: Gestionar Decisión Judicial**

El “tipo de decisión” se aplica en dependencia del tipo de fase seleccionada para registrar la decisión y puede incluir datos de otras decisiones que impliquen sentencia condenatoria, suspensión condicional del proceso o los delitos que fueron cometidos por el interno. La vista de los delitos será mostrada únicamente si la decisión judicial a adicionar implica egreso o cierre del proceso. A continuación se muestran tres vistas que se han separado físicamente de la vista principal y que en dependencia del tipo de decisión seleccionada se incluidas dentro de esta.

Pena Principal		Penas Accesorias	
<b>Tipo Pena</b>	Presidio	<b>Pena</b>	La inhabilitación política durante el tiempo de la condena
		<input type="button" value="Nuevo"/> <input type="button" value="Adicionar"/> <input type="button" value="Cancelar"/>	
<b>Período de Sentencia</b>		<b>Penas Accesorias</b>	< < <b>1</b> ... > >   <b>Total: 1</b>
<b>Años</b>	10	<b>Penas Accesorias</b>	
<b>Meses</b>	5	La sujeción a la vigilancia de la autoridad por una cuarta parte del tiempo de la condena, desde que esta termine	
<b>Días</b>	2		<input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>

Figura 13: Decisión Suspensión Condicional del Proceso

ESTA DECISIÓN JUDICIAL VA A CERRAR EL PROCESO					
<b>Síntesis de los Hechos</b>	<b>Delitos</b>   < < <b>1</b> ... > >   <b>Total: 1</b>				
Robo con fuerza, homicidio y corrupción.	<table border="1"> <thead> <tr> <th>Tipo de Delito</th> <th>Probado</th> </tr> </thead> <tbody> <tr> <td>Violación, seducción, prostitución o corrupción de menores y ultrajes al pudor</td> <td><input checked="" type="checkbox"/></td> </tr> </tbody> </table>	Tipo de Delito	Probado	Violación, seducción, prostitución o corrupción de menores y ultrajes al pudor	<input checked="" type="checkbox"/>
Tipo de Delito	Probado				
Violación, seducción, prostitución o corrupción de menores y ultrajes al pudor	<input checked="" type="checkbox"/>				
<input type="button" value="Atrás"/> <input type="button" value="Terminar"/> <input type="button" value="Cancelar"/>					

Figura 14: Decisión Sentencia Condenatoria

Período de Suspensión		Condiciones	
<b>Años</b>	5	<b>Condición</b>	Realizar labores de trabajo carcelarias.
<b>Meses</b>	3	<input type="button" value="Nuevo"/> <input type="button" value="Adicionar"/> <input type="button" value="Cancelar"/>	
<b>Días</b>	1	<b>Condiciones</b>	< < <b>1</b> ... > >   <b>Total: 1</b>
		<b>Condiciones</b>	
		No cometer ninguna indisciplina.	<input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>

Figura 15: Decisión que implica Egreso o fin del Proceso

El diagrama de extensiones web que representa las interacciones cliente servidor de este caso de uso es el siguiente.

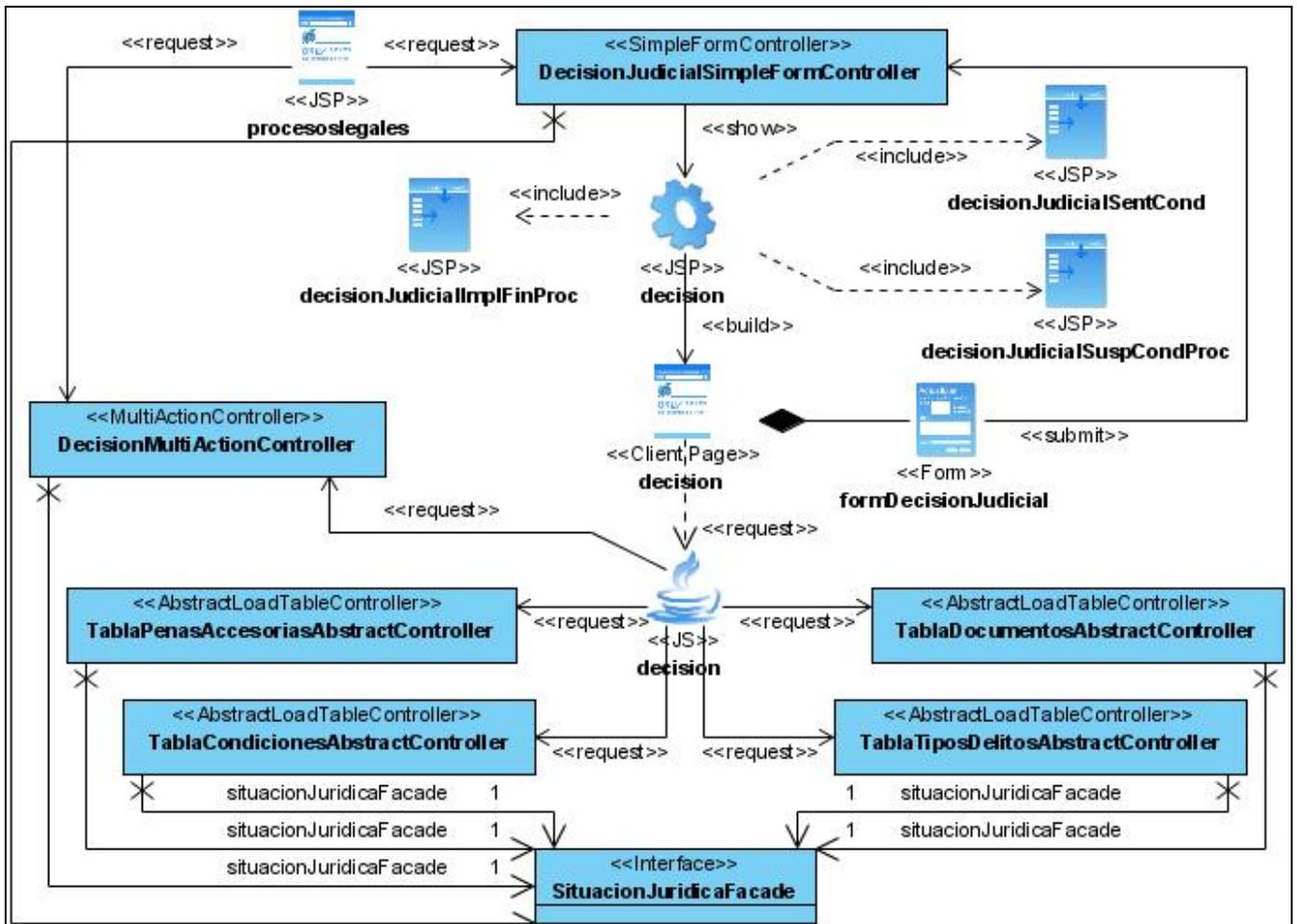
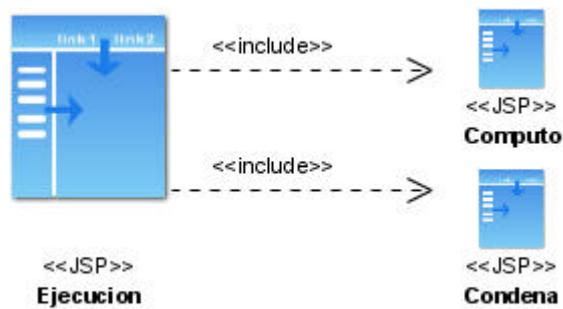


Figura 16: Diagrama extensiones web de las Decisiones Judiciales

## EJECUCION DE LA PENA

El conjunto de casos de uso CRUD Ejecuciones, CRUD Computo, CRUD Pena están estrechamente relacionados debido a que la Pena y El Computo son parte de una Ejecución, lo que condiciona que la solución fuese elaborada de forma tal que se pueda observar una ejecución con su respectiva Condena y Computo.



**Figura 17: Relación de dependencia entre las vistas que componen la Ejecución**

La figura 17, representa las relaciones entre las vistas de la Ejecución de la Pena, donde se puede observar que la misma está compuesta por el Cómputo y la Pena impuesta.

### 3.3.2 Funcionalidades de ejecuciones.

Caso de Uso: CRUD ejecuciones		
<b>Objetivo</b>		Adicionar, consultar, modificar y eliminar las ejecuciones de la pena.
<b>Actores</b>		FCP
<b>Resumen</b>		El sistema muestra las ejecuciones registradas y permite modificarlas, eliminarlas y adicionar nuevas. El FCP puede realizar estas operaciones hasta que indique que ha terminado el caso de uso.
<b>Precondición</b>		El FCP se ha identificado y autenticado correctamente ante el sistema y tiene permisos para utilizar este servicio. Se ha buscado y seleccionado el expediente del individuo(a).
<b>Garantías</b>	<b>Mínimas</b>	El sistema notifica al FCP cualquier evento que impida adicionar, consultar, eliminar o modificar las ejecuciones de la pena del individuo.
	<b>De éxito</b>	Adicionar y modificar: Se registraron las ejecuciones introducidas por el FCP. Eliminar: Se eliminaron las ejecuciones que el FCP ha indicado que deben ser eliminadas.
<b>Acción que inicia el caso de uso</b>		El FCP indica que desea gestionar las ejecuciones del individuo.
<b>Descripción del CRUD</b>		El caso de uso tiene cuatro (4) flujos básicos: <ul style="list-style-type: none"> <li>• Consultar ejecución.</li> <li>• Adicionar ejecución.</li> <li>• Modificar ejecución.</li> <li>• Eliminar ejecución.</li> </ul>
<b>Relaciones</b>	<b>CU incluidos</b>	CRUD documentos
	<b>CU extensión</b>	CRUD delitos
<b>Requisitos especiales</b>		La fase seleccionada como que pasó el proceso a ejecución se debe marcar como última.
<b>Asuntos pendientes</b>		No aplicable

**Tabla 2: Caso de Uso: CRUD ejecuciones**

La figura 18, representa la entrada al Caso de Uso CRUD Ejecución. Esta posee varias tablas, donde el FCP selecciona una Ejecución llevada por un tribunal y se muestra en las demás los procesos penales en ejecución y los delitos probados del interno. Además provee las funcionalidades de Gestionar y Acumular Ejecución.

Tribunales de Ejecución		< < <u>1</u> ... > >	Total: 1
Nro Expediente	Tribunal	Estado	
GP01-P-2006-019699	Tribunal Primero de Ejecución del Circuito Judicial Penal Carabobo extensión Valencia	Sin Acumular	
<input type="button" value="Nuevo"/> <input type="button" value="Acumular"/> <input type="button" value="Eliminar"/>			
Procedencia de los procesos penales en ejecución		< < <u>1</u> ... > >	Total: 1
Número Expediente	Tribunal		
GP01-P-2006-019699	Tribunal Sexto de Control del Circuito Judicial Penal Carabobo extensión Valencia		
Delitos Probados		< < <u>1</u> ... > >	Total: 1
Nombre Delito			
Robo			

**Figura 18: Vista de la ejecución**

La Figura 19, muestra la pantalla de adicionar una nueva ejecución, la cuál tiene número de expediente, la jurisdicción (Ordinaria, Adolescente, Militar) y el tribunal que llevará la ejecución, una síntesis de los hechos y los delitos cometidos. El vínculo “Seleccionar” muestra una vista donde se seleccionan los delitos cometidos.

<b>Número de Expediente</b>		<b>Jurisdicción</b>				
<input type="text" value="0130_14623"/>		<input type="text" value="Ordinaria"/>				
<b>Datos del Tribunal</b>						
<b>Tribunal Ordinario</b>						
<b>Estado</b>	<b>Circuito Judicial Penal</b>	<b>Extensión</b>	<b>Tipo de Tribunal</b>			
<input type="text" value="Anzoátegui"/>	<input type="text" value="Anzoátegui"/>	<input type="text" value="Barcelona"/>	<input type="text" value="Ejecución"/>			
<b>Número de Tribunal</b>						
<input type="text" value="Segundo"/>						
<b>Síntesis del Hecho</b>		<b>Delitos</b>				
<input type="text" value="Violación de información y derechos gubernamentales."/>		<p>▶ <b>Seleccionar</b></p> <p>  &lt; &lt; 1 ... &gt; &gt;   <b>Total: 2</b></p> <table border="1"> <tr> <td><b>Delito</b></td> </tr> <tr> <td>Violación de secretos políticos y militares</td> </tr> <tr> <td>Cooperación a la guerra contra la Patria</td> </tr> </table>		<b>Delito</b>	Violación de secretos políticos y militares	Cooperación a la guerra contra la Patria
<b>Delito</b>						
Violación de secretos políticos y militares						
Cooperación a la guerra contra la Patria						
		<input type="button" value="Eliminar"/>				
		<input type="button" value="Aplicar"/>	<input type="button" value="Cerrar"/>			

Figura 19: Vista nueva ejecución

### 3.3.3 Funcionalidades de los delitos.

<b>Caso de Uso: CRUD delitos</b>		
<b>Objetivo</b>	Adicionar, consultar, modificar y eliminar los delitos de un proceso penal.	
<b>Actores</b>	FCP	
<b>Resumen</b>	El sistema muestra los delitos del individuo que han sido registrados y permite eliminarlos, modificarlos y agregar nuevos. El FCP puede realizar estas operaciones hasta que indique que ha terminado el caso de uso.	
<b>Precondición</b>	El FCP se ha identificado y autenticado correctamente ante el sistema y tiene permisos para utilizar este servicio. Se ha buscado y seleccionado el expediente del individuo(a). Se ha seleccionado un proceso penal.	
<b>Garantías</b>	<b>Mínimas</b>	El sistema notifica al FCP cualquier evento que impida adicionar, consultar, eliminar o modificar delitos.
	<b>De éxito</b>	Adicionar y modificar: Se registraron los delitos del proceso penal. Eliminar: Se eliminaron del sistema los delitos que el FCP ha indicado que deben ser eliminados.
<b>Acción que inicia el caso de uso</b>	El FCP indica que desea gestionar los delitos de un proceso penal.	
<b>Descripción del CRUD</b>	El caso de uso tiene cuatro (4) flujos básicos: <ul style="list-style-type: none"> <li>• Consultar delitos</li> <li>• Adicionar delitos.</li> <li>• Modificar delitos.</li> <li>• Eliminar delitos.</li> </ul>	
<b>Relaciones</b>	<b>CU incluidos</b>	No aplicable
	<b>CU extensión</b>	No aplicable
<b>Requisitos especiales</b>	No aplicable	
<b>Asuntos pendientes</b>	No aplicable	

**Tabla 3: Caso de Uso: CRUD delitos**

La siguiente vista permite seleccionar un conjunto de delitos cometidos y adicionarlos a la tabla a una nueva Ejecución.

**SELECCIONAR DELITOS**

- Delitos contra las buenas costumbres y el buen orden de las familias
- Delitos contra las personas
- Delitos contra la propiedad
- Delitos contra la cosa pública
- Delitos contra la libertad
- Delitos contra la Conservación de los Intereses Públicos y Privados
- Faltas en general
- Delitos contra la Independencia y la Seguridad Nacional
- Delitos relacionados con el tráfico ilícito y el consumo de sustancias estupefacientes y psicotrópicas
- Delitos contra los derechos de las mujeres a una vida libre de violencia
- Delitos contra la administración de justicia
- Delitos contra el orden público
  - Importación, fabricación, comercio, detención y porte de armas
  - Instigación a delinquir
  - Agavillamiento
  - Excitación a la guerra civil, organización de cuerpos armados o intimidación al público
- Delitos contra la fé pública
  - Falsedad en los actos y documentos
  - Falsedad de pasaportes, licencias, certificados y otros semejantes
  - Fraudes cometidos en el comercio, las industrias y las almonedas
  - Quiebras
  - Falsificación de monedas o títulos de crédito público
  - Falsificación de sellos, timbres públicos y marcas

Aplicar Cerrar

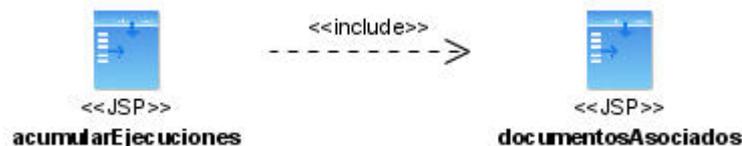
**Figura 20: Vista delitos cometidos**

### 3.3.4 Funcionalidades de unificar penas

Caso de Uso: Unificar penas		
<b>Objetivo</b>	Unificar varias ejecuciones en una única ejecución de la pena.	
<b>Actores</b>	FCP	
<b>Resumen</b>	El FCP selecciona varias ejecuciones e indica unificarlas. El sistema permite unificarlas en una ya existente o en una nueva, solicitando los datos correspondientes.	
<b>Precondición</b>	El FCP se ha identificado y autenticado correctamente ante el sistema y tiene permisos para utilizar este servicio.  Se ha buscado y seleccionado el expediente del individuo(a).	
<b>Garantías</b>	<b>Mínimas</b>	El sistema notifica al FCP cualquier evento que impida unificar las penas.
	<b>De éxito</b>	Se cambió el estado de las ejecuciones acumuladas a "unificada".
<b>Acción que inicia el caso de uso</b>		El FCP indica que desea unificar ejecuciones.
<b>Relaciones</b>	<b>CU incluidos</b>	CRUD documentos
	<b>CU extensión</b>	CRUD ejecuciones
<b>Requisitos especiales</b>		No aplicable
<b>Asuntos pendientes</b>		Extensión Asociar solicitud

**Tabla 4: Caso de Uso: Unificar penas**

La vista acumular ejecuciones requiere las funcionalidades de los documentos asociados. La figura 21 muestra la relación de dependencia que existe entre ellas.



**Figura 21: Relación de dependencia entre acumular ejecución y los documentos asociados**

Para acumular las ejecuciones se debe seleccionar una de ellas o marcar el campo “Crear nueva ejecución”. En caso de seleccionar una de las ejecuciones, solo debe registrar el documento que avala la decisión. En caso de crear una Nueva Ejecución, debe especificar el Número de Expediente, la Jurisdicción, seleccionar el Tribunal que puede ser unos de los que atiende las ejecuciones a acumular o elegir otro permitiendo la selección de un nuevo Tribunal de Ejecución y registrar los documentos asociados que avalan la decisión Acumular Ejecuciones.

**ACUMULAR EJECUCIONES**

---

**Seleccione la ejecución donde desea acumular o cree una nueva.** | < < 1 ... > > | **Total: 2**

Número Expediente	Nombre Tribunal
aaaaaaa	Tribunal Primero de Ejecución del Circuito Judicial Penal Anzoátegui extensión El Tigre
GP01-P-2006-015204	Tribunal Primero de Ejecución del Circuito Judicial Penal Carabobo extensión Valencia

**Crear nueva ejecución**

---

**Número Expediente**       **Jurisdicción**       **Tribunal**

**Datos del Tribunal**

**Tribunal Ordinario**

**Estado**       **Circuito Judicial Penal**       **Extensión**       **Tipo de Tribunal**

**Número de Tribunal**

**Documentos Asociados a la Decisión**

**Número**       **Tipo de Documento**       **Fecha de Elaboración**       **Fecha de Recepción**

---

**Documentos** | < < 1 ... > > | **Total: 1**

Número	Tipo de Documento	Elaboración	Recepción
4234	Auto acumulación de penas	29/04/2008	30/04/2008

---

**Figura 22: Vista Acumular Ejecuciones.**

La figura 23 muestra el diagrama de extensiones web asociado al caso de uso CRUD ejecuciones.

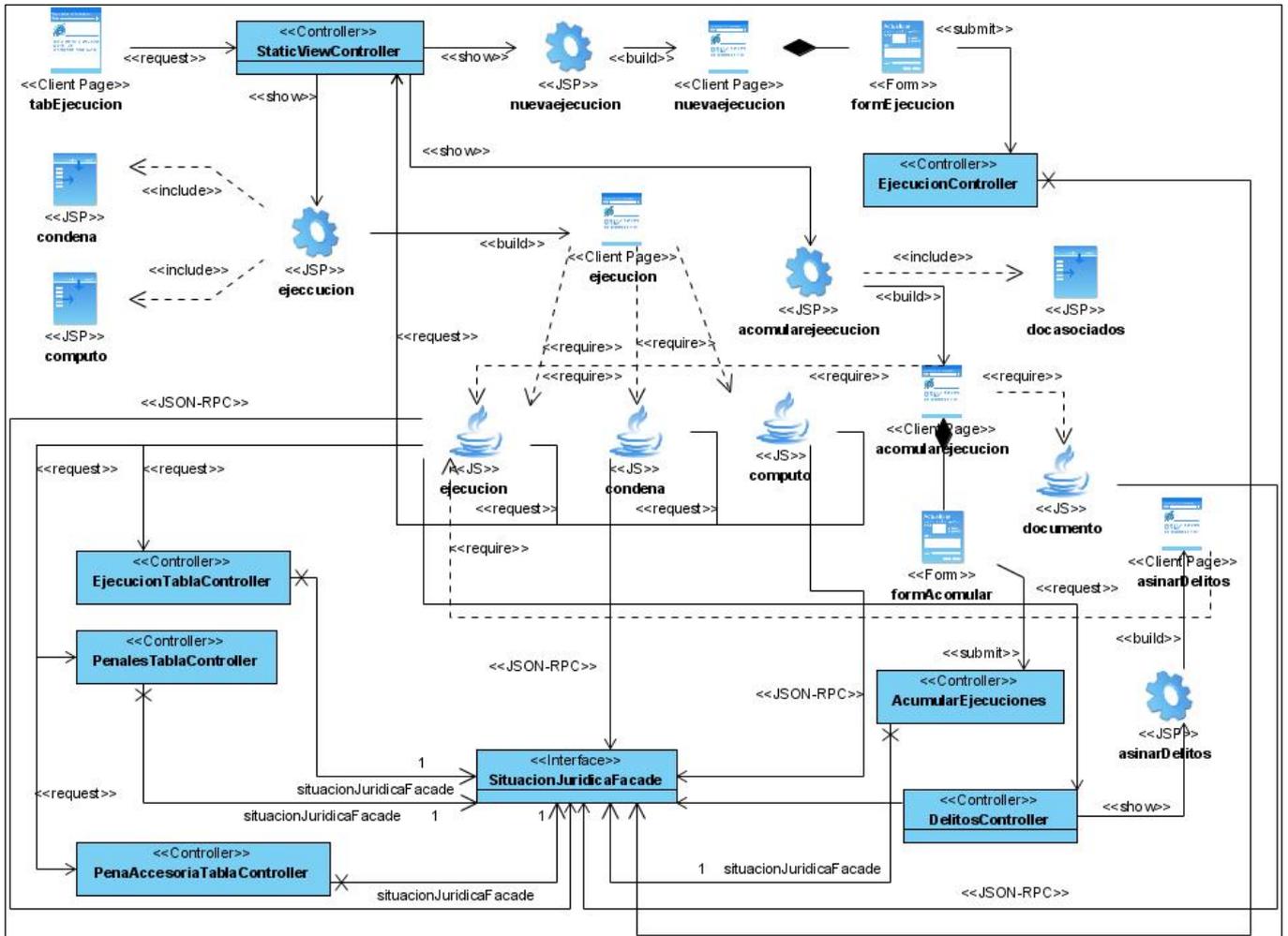


Figura 23: Diagrama extensiones web de la Ejecución de la Pena

### 3.3.5 Funcionalidades de la pena.

Caso de Uso: CRUD-D pena		
<b>Objetivo</b>	Registrar, consultar y modificar la condena del individuo(a).	
<b>Actores</b>	FCP	
<b>Resumen</b>	El sistema muestra la pena del individuo(a) que ya ha sido registrada y permite modificarla.	
<b>Precondición</b>	<p>El FCP se ha identificado y autenticado correctamente ante el sistema y tiene permisos para utilizar este servicio.</p> <p>Se ha buscado y seleccionado el expediente del individuo(a).</p> <p>Se ha seleccionado una ejecución de la pena.</p>	
<b>Garantías</b>	<b>Mínimas</b>	El sistema notifica al FCP cualquier evento que impida registrar, consultar o modificar la pena del individuo(a)
	<b>De éxito</b>	Registrar y modificar: Se registró la pena que ha introducido el FCP.
<b>Acción que inicia el caso de uso</b>	El FCP indica que desea gestionar la pena del individuo(a).	
<b>Descripción del CRUD</b>	<p>El caso de uso tiene cuatro (3) flujos básicos:</p> <ul style="list-style-type: none"> <li>• Consultar pena.</li> <li>• Modificar pena.</li> <li>• Adicionar pena.</li> </ul>	
<b>Relaciones</b>	<b>CU incluidos</b>	CRUD documentos
	<b>CU extensión</b>	No aplicable
<b>Requisitos especiales</b>	La condena sólo puede ser emitida por el tribunal de la última fase del proceso.	
<b>Asuntos pendientes</b>	<p>Preguntar si las siguientes penas son principales o accesorias:</p> <ul style="list-style-type: none"> <li>• Multa.</li> <li>• Caución de no ofender o dañar.</li> <li>• Amonestación o apercibimiento.</li> <li>• Pérdida de los instrumentos o armas con que se cometió el hecho punible y de los efectos que de él provengan.</li> </ul>	

Tabla 5: Caso de Uso: CRUD-D pena

La figura 24 muestra la vista perteneciente al caso de uso CRUD-D pena, la misma esta compuesta por varias tablas, donde el FCP selecciona en la primera una Condena y se muestra el Tipo de Pena, la Pena Impuesta y el conjunto de Penas Accesorias. Además de proveer las funcionalidades de Gestionar la Condena.

**Condena** | < < 1 ... > > | **Total: 1**

Fecha	Sentencia
09/05/2007	Sentencia condenatoria

**Tipo de Pena**

Prisión

**Pena impuesta**

Años	Meses	Días
6	0	0

| < < 1 ... > > | **Total: 2**

**Penas Accesorias**

La inhabilitación política durante el tiempo de la condena

La sujeción a la vigilancia de la autoridad por una cuarta parte del tiempo de la condena, desde que esta termine

**Figura 24: Vista de la condena**

La vista nueva pena requiere las funcionalidades de los documentos asociados. La figura 25 muestra la relación de dependencia que existe entre ellas.



**Figura 25: Relación de dependencia entre la nueva pena y los documentos asociados**

En la figura 26 se permite gestionar la información asociada a la gestión de la condena, la cuál tiene Origen de Sentencia (Decisión por la que se realiza la Condena), la Pena Impuesta en Años Meses y Días, los Tipo de Pena que permiten la selección de las Penas Accesorias para adicionar en la tabla de Penas Accesorias y registrar los documentos asociados que avalan el Origen de la Sentencia.

**CONDENA**

---

**Origen de la sentencia**

**Tipo de Pena**

**Pena Impuesta**

**Años**

**Meses**

**Días**

**Penas Accesorias**

**Pena**

| < < 1 ... > > | **Total: 1**

Penas
Suspensión del empleo

---

**Documentos Asociados a la Decisión**

**Número**

**Tipo de Documento**

**Fecha de Elaboración**

**Fecha de Recepción**

---

**Documentos** | < < 1 ... > > | **Total: 1**

Número	Tipo de Documento	Elaboración	Recepción
A-01-101-22	Copia de la sentencia	29/04/2008	30/04/2008

---

**Figura 26: Vista Gestionar Condena**

La figura 27 muestra el diagrama de extensiones web asociado al caso de uso CRUD-D pena.

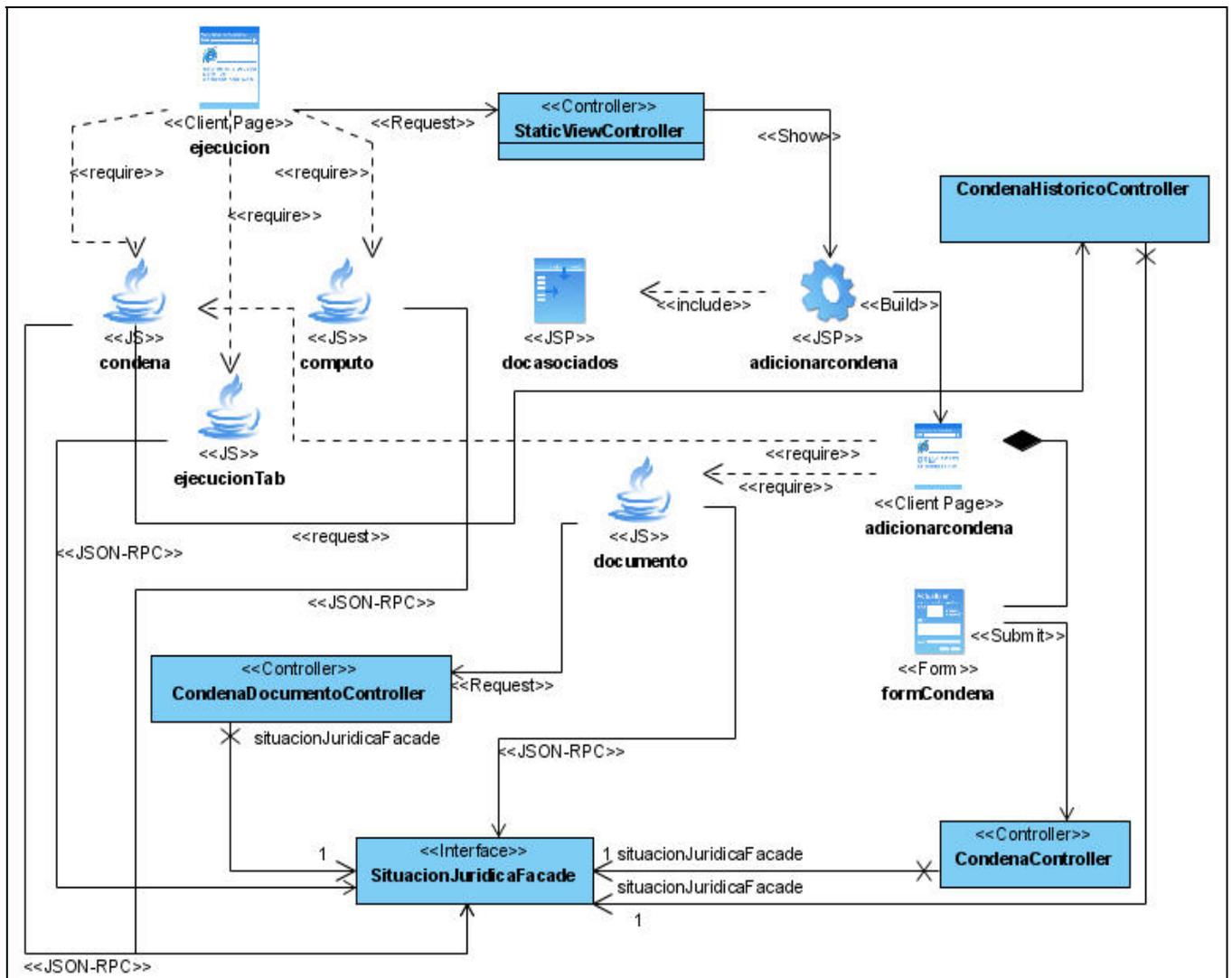


Figura 27: Diagrama extensiones web de la Condena

### 3.3.6 Funcionalidades del cómputo

Caso de Uso: CRUD cómputo		
Objetivo	Adicionar, modificar, consultar y eliminar cómputos de la pena al individuo(a).	
Actores	FCP	
Resumen	El FCP indica que desea gestionar los cómputos del individuo(a), el sistema muestra un listado de los cómputos registrados y permite adicionar nuevos, modificarlos y eliminarlos.	
Precondición	El FCP se ha identificado y autenticado correctamente ante el sistema y tiene permisos para utilizar este servicio. Se ha buscado y seleccionado el expediente del individuo(a). Se ha seleccionado una ejecución de la pena.	
Garantías	Mínimas	El sistema notifica al FCP cualquier evento que impida adicionar, eliminar, modificar o consultar el cómputo.
	De éxito	Adicionar: Se registró el cómputo de la pena del individuo(a) y la decisión que lo avala. Modificar: Se modificó el cómputo de la pena. Eliminar: Se eliminó el cómputo de la pena indicado por el FCP.
Acción que inicia el caso de uso	El FCP indica que desea gestionar el cómputo	
Descripción del CRUD	El caso de uso tiene cuatro (4) flujos básicos: Consultar cómputo. Adicionar cómputo. Modificar cómputo. Eliminar cómputo.	
Relaciones	CU incluidos	CRUD documentos
	CU extensión	Asociar solicitud
Requisitos especiales	No aplicable	
Asuntos pendientes	No aplicable	

**Tabla 6: CRUD cómputo**

La entrada al caso de uso CRUD Cómputo se realiza a través de la figura 28, la misma está compuesta por varias tablas, donde el FCP selecciona en la primera un Cómputo y se muestra en la siguiente información respecto al mismo. Además de proveer las funcionalidades de Gestionar un Cómputo.

**Cómputo de la Pena**

Fecha Cumplimiento: 12/12/2012      Cómputo Activo

| < < 1 ... > > |      Total: 1

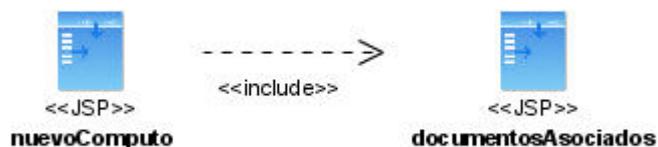
Fecha Cumplimiento	Motivo de Cómputo
12/12/2012	Cómputo ordinario

Por Cumplir	Cumplido
5 Año(s)	0 Año(s)
7 Mes(es)	4 Mes(es)
4 Día(s)	26 Día(s)

Estado de Progresividad	Fecha
Destacamento de Trabajo	
Régimen abierto	
Libertad Condicional	
Confinamiento	

**Figura 28: Vista Gestionar Cómputo de la Pena**

La vista nuevo computo requiere las funcionalidades de los documentos asociados. La figura 28 muestra la relación de dependencia que existe entre ellas.



**Figura 28: Relación de dependencia entre las la vista nuevo computo y documentos asociados**

La figura 29, muestra la vista de Gestionar el Cómputo de la Pena, el cuál tiene Motivo del Computo (Decisión por la que se realiza el Cómputo), la Fecha Cumplimiento de Pena, el Tiempo por Cumplir y el Tiempo Cumplido en Años, Meses y Días, los Estados de Progresividad con la Fecha correspondiente y los documentos asociados que avalan el Motivo de Cómputo.

**REGISTRAR NUEVO CÓMPUTO DE LA PENA**

---

<b>Motivo del Cómputo</b>	<b>Tiempo por Cumplir</b>	<b>Tiempo Cumplido</b>
<input type="text" value="Cómputo ordinario"/>	<b>Años</b> <input type="text" value="5"/>	<b>Años</b> <input type="text" value="0"/>
<b>Fecha Cumplimiento de la Pena</b>	<b>Meses</b> <input type="text" value="7"/>	<b>Meses</b> <input type="text" value="4"/>
<input type="text" value="12/12/2012"/>	<b>Días</b> <input type="text" value="4"/>	<b>Días</b> <input type="text" value="26"/>

---

**Términos**

**Estado de Progresividad / Fecha que Corresponde**

<b>Destacamento de Trabajo</b>	<b>Régimen Abierto</b>	<b>Libertad Condicional</b>	<b>Confinamiento</b>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

---

**Documentos Asociados a la Decisión**

<b>Número</b>	<b>Tipo de Documento</b>	<b>Fecha de Elaboración</b>	<b>Fecha de Recepción</b>
<input type="text"/>	<input type="text" value="-Seleccione-"/>	<input type="text"/>	<input type="text"/>

---

**Documentos** | < < 1 ... > > | **Total: 1**

Número	Tipo de Documento	Elaboración	Recepción
GP01-P-2006-019-699	Cómputo de la pena	09/05/2007	

---

**Figura 29: Vista Gestionar Cómputo**

La figura 30 muestra el diagrama de extensiones web asociado al caso de uso CRUD cómputo.

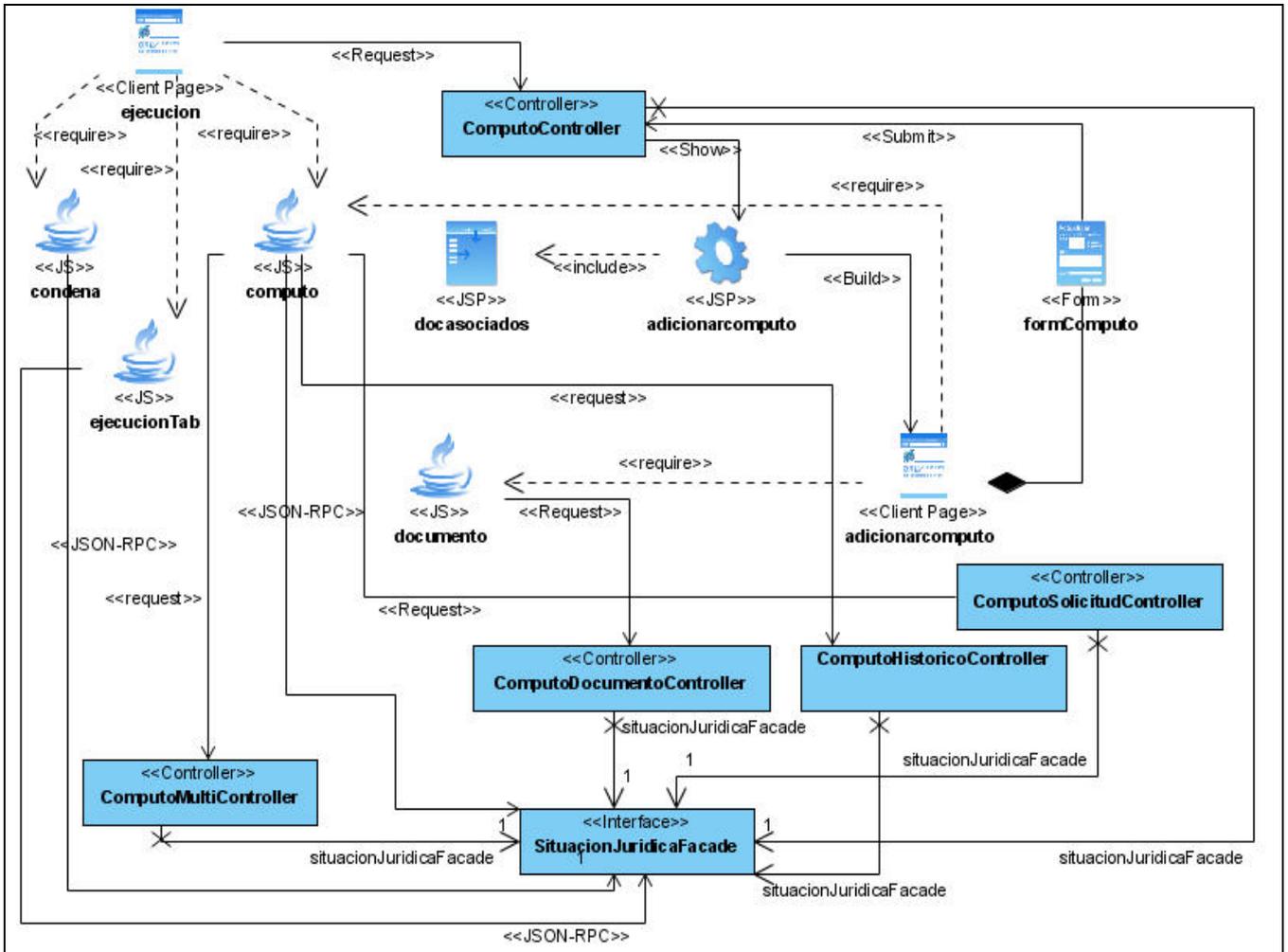


Figura 30: Diagrama extensiones web del Cómputo de la Pena

### 3.4. Interacción entre capa de presentación y negocio

Para el diseño del módulo se identificaron un conjunto de clases que modelan la solución de las funcionalidades a implementar. Las cuales han sido separadas dependiendo de la capa arquitectónica donde se encuentran. La primera parte pertenece a la capa de presentación y la segunda parte representa los objetos de dominio perteneciente a la capa lógica de negocio.

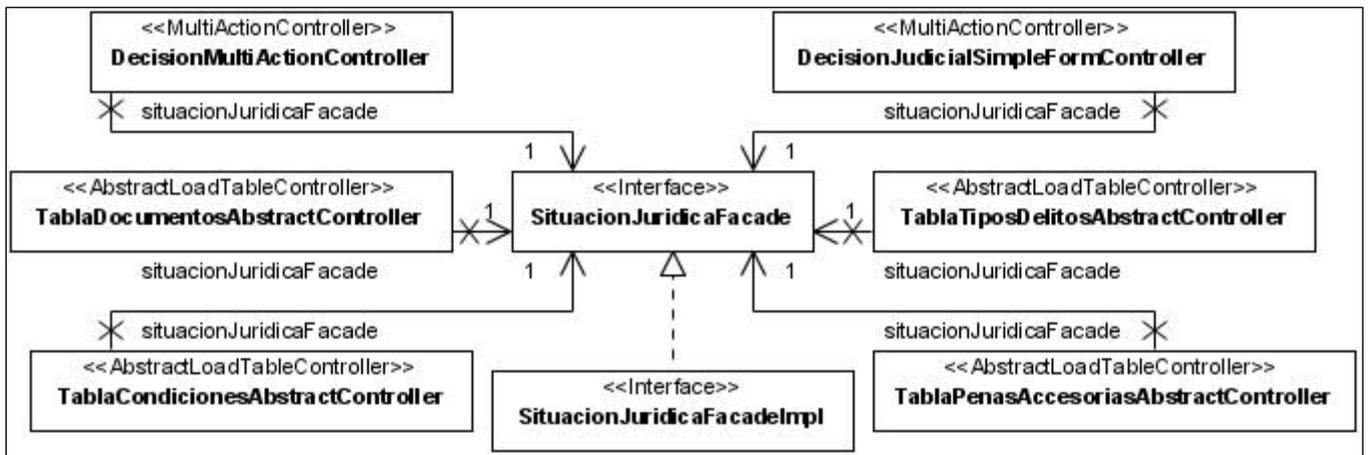


Figura 31: Clases del diseño de Decisiones Judiciales para la capa de presentación

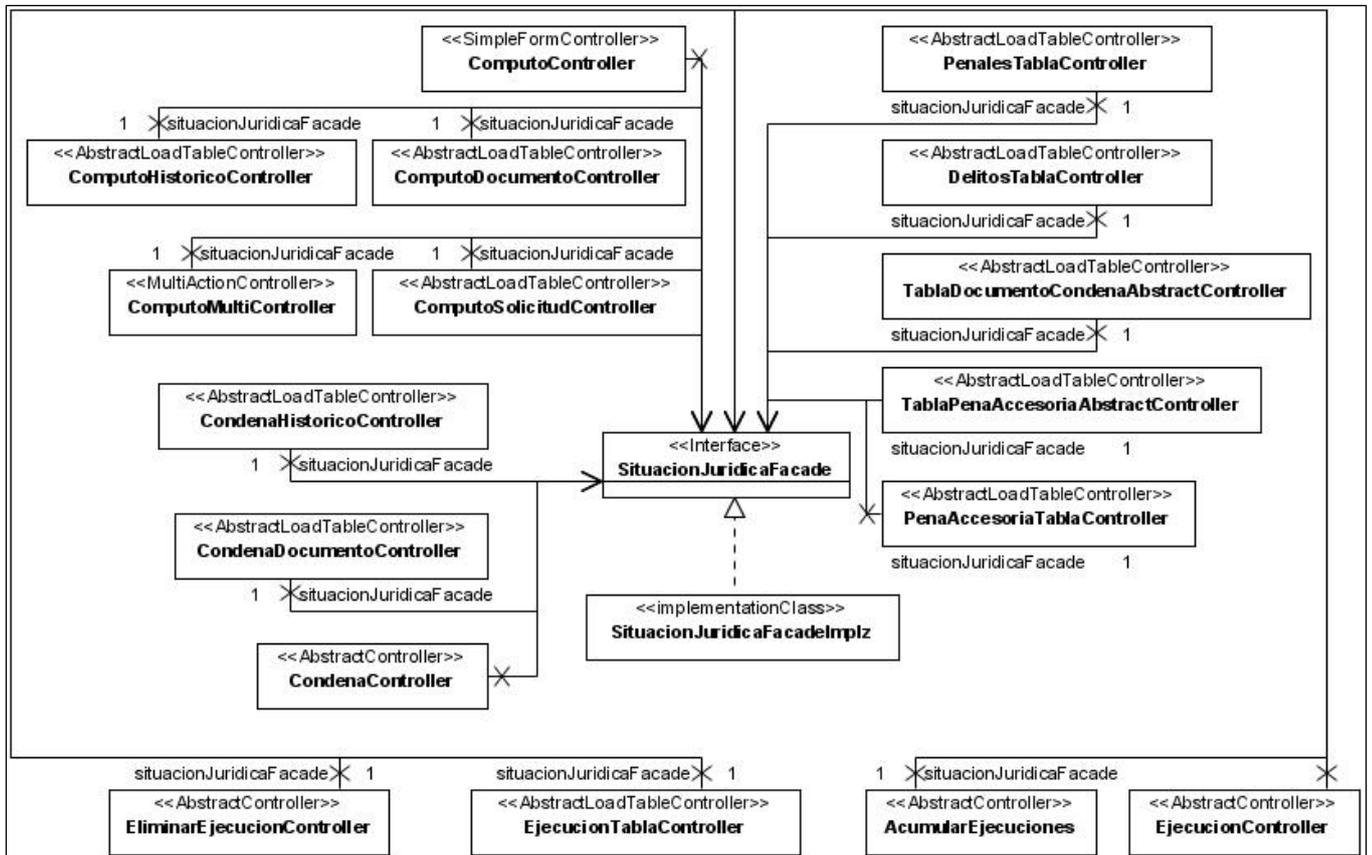


Figura 32: Clases del diseño de Ejecución de la Pena para la capa de presentación



### 3.5 Componentes Reutilizables

Los componentes reutilizables dentro de la solución del sistema son los siguientes:

- **Documentos asociados.**
- **Tablas (NavigableTable).**
- **Tribunales (TribunalTable).**

La reutilización de los **documentos asociados** constituyó una gran ventaja dentro de la solución debido a que el tiempo de implementar la misma funcionalidad varias veces sería grande y esto traería consigo retardo en las demás actividades. Como su uso era muy frecuente se decidió crear una sala vista que sería incluida dentro de las jsp que lo requirieren, así los demás casos de uso pueden acceder a la misma sin mucha complejidad y disminuyendo los errores.

Los documentos asociados son la funcionalidad que representa el caso de uso CRUD Documentos. Este caso de uso es incluido o usado en los demás casos de usos del sistema, lo que proporciona que el mismo sea común en cada una de las pantallas en la que se registran documentos. Entre los datos que se encuentran en los documentos están: número, tipo, Fecha de Elaboración y Recepción del mismo.

**Documentos Asociados a la Decisión**

Número  \*      Tipo de Documento       Fecha de Elaboración       Fecha de Recepción

**Documentos**      | < < 1 ... > > |      **Total: 0**

Número	Tipo de Documento	Elaboración	Recepción

**Figura 34: Componente Documentos Asociados**

El componente **NavegableTable** o comúnmente conocido como tabla, es utilizado en una gran cantidad de vistas dentro del módulo. Mediante él se pueden representar datos en forma de listas, permitiendo el ordenado de elementos, ya sea por nombres o números y facilita un grupo de funcionalidades básicas tales como: adicionar, modificar y eliminar elementos de la tabla realizando peticiones AJAX al servidor que hacen más rápido y cómodo el proceso de envío y actualización de datos. Este componente puede ser incluido en cualquier vista permitiendo ser utilizado por todos los programadores.

El componente **TribunalTable**, facilita seleccionar los tribunales que atienden el proceso judicial, ya sea para emitir una decisión o imponer una condena. El proceso de obtención de un tribunal parte de la selección del Estado, el circuito judicial, la extensión del circuito (si la tiene) y el tipo de tribunal; en correspondencia con estos datos, aparecen los números de tribunales registrados en esa localización. Una vez formada la cadena se puede identificar dicho tribunal.

Datos del Tribunal			
<b>Tribunal Itinerante</b>			
<b>Estado</b>	<b>Circuito Judicial Penal</b>	<b>Extensión</b>	<b>Tipo de Tribunal</b>
<input type="text" value="-Selecione-"/>	<input type="text" value="-Selecione-"/>	<input type="text" value="-Selecione-"/>	<input type="text" value="-Selecione-"/>
<b>Número de Tribunal</b>			
<input type="text"/>			

**Figura 35: Componente TribunalTable**



## CONCLUSIONES

---

Como resultados de este trabajo se obtuvo el diseño de la capa de Presentación del módulo Situación Jurídica, se implementó la solución del problema usando los artefactos brindados por el diseño utilizando las herramientas, tecnologías y convenciones de código definidas en la arquitectura del proyecto. Se refinó el modelo de datos de acuerdo con las entidades involucradas en la solución del módulo. Se logró cumplir con lo requerido con el cliente, el módulo fue probado y aceptado por el mismo y se encuentra funcionando en un penal de la República de Bolivariana de Venezuela.

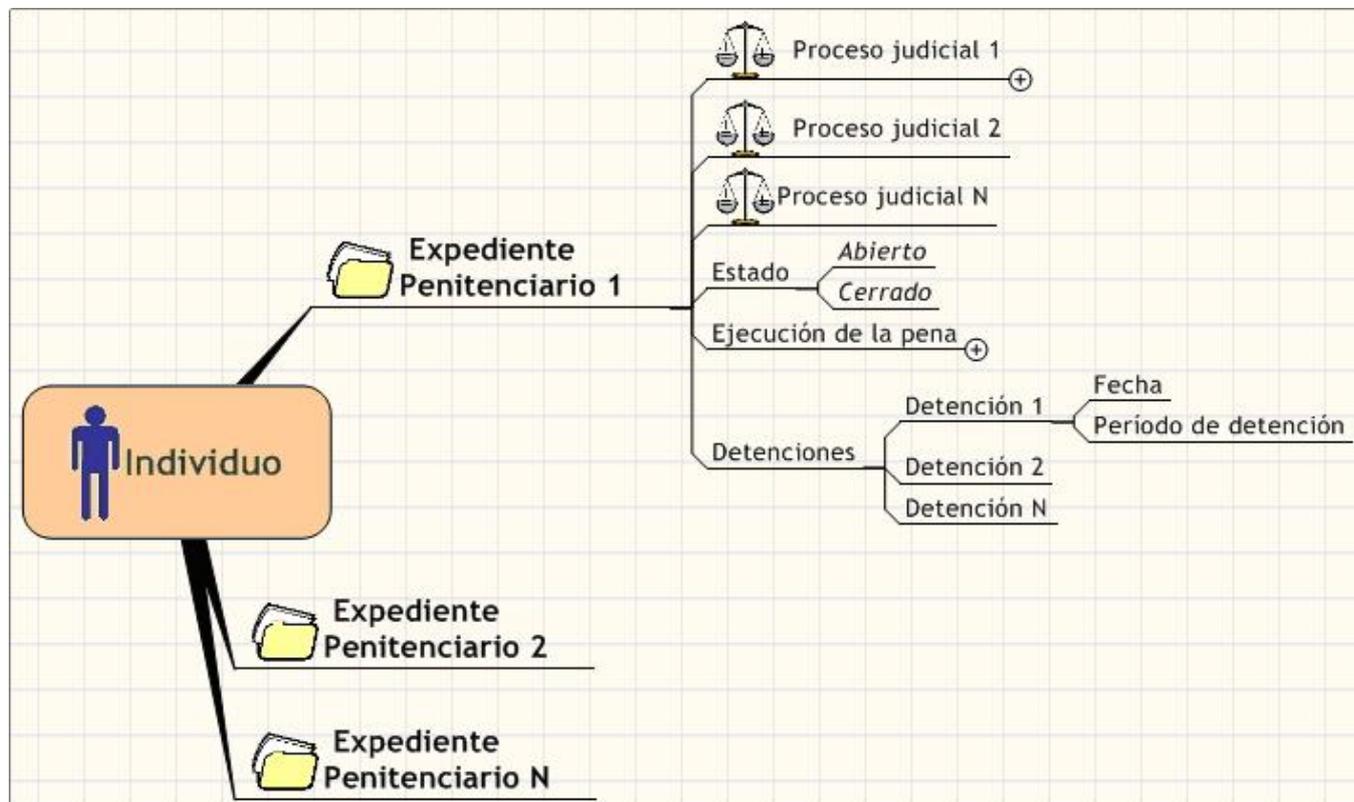
## RECOMENDACIONES

---

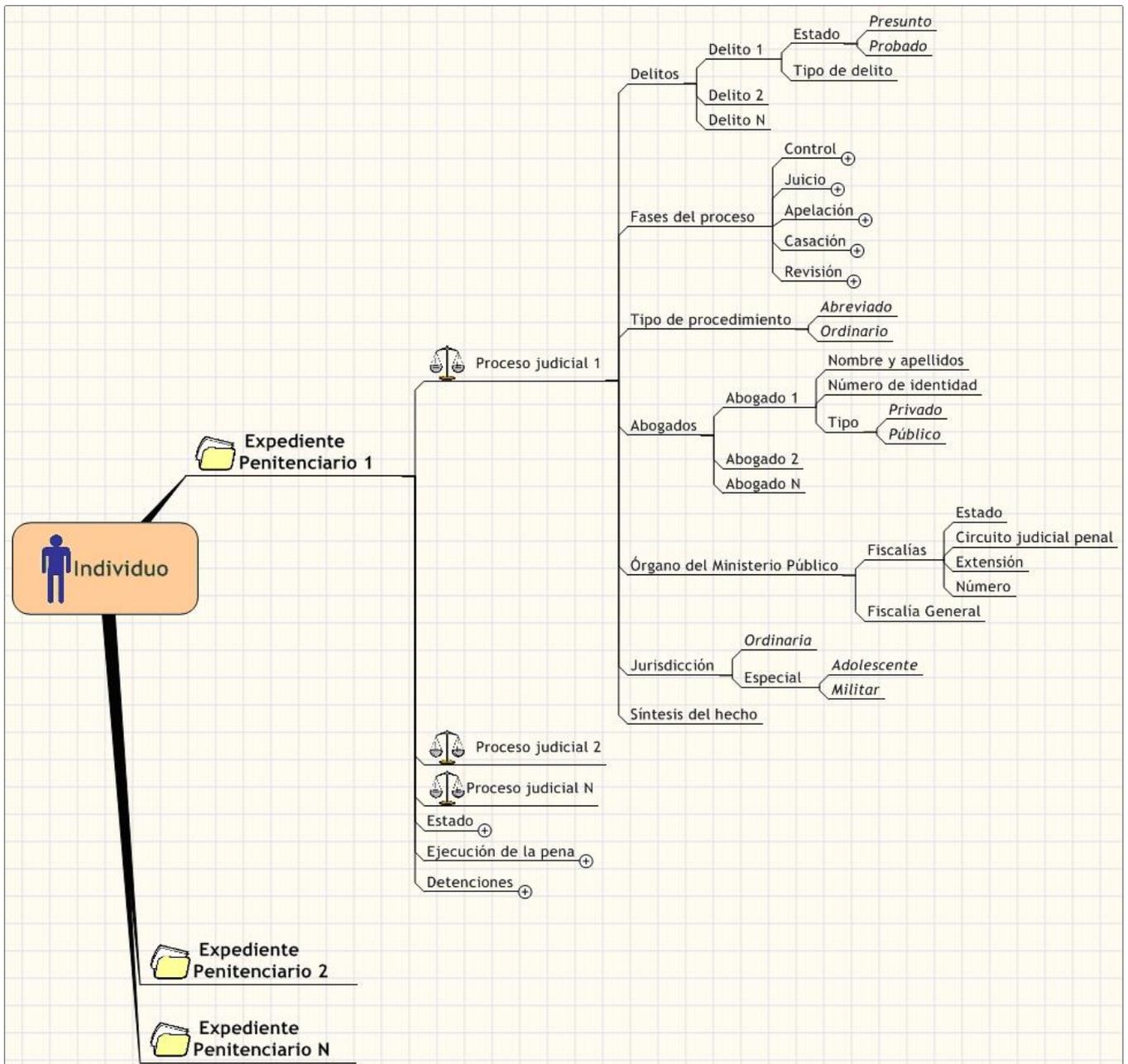
Se recomienda valorar el uso de la arquitectura definida en este trabajo para desarrollar interfaces web a aquellos proyectos que se desarrollen bajo la plataforma JEE.

De igual manera el flujo de trabajo aquí expuesto pudiera servir como marco metodológico para la implementación de la capa de presentación.

- Booch; Rumbaugh & Jacobson. *El Lenguaje Unificado de Modelado*. 1999. 480 p. 0-201-30998-X
- Craig Walls. *Spring In Action Second Edition*. 2008. p. 1-933988-13-4
- Crane; Pascarello & James. *Ajax in Action*. Greenwich, Manning Publications Co., 2006. p.
- García. *Tutorial de Java Server Pages*. 2002. p.
- Gross. *Ajax Patterns and Best Practices*. Nueva York, Springer-Verlag, Inc., 2006. p. 1-59059-616-1  
978-1-59059-616-6
- Heilmann. *Beginning JavaScript with DOM Scripting and Ajax*. 2006. p.
- Keese. *Dijit 0.9: Tour of new features*, 2007. [2008]. Disponible en: <http://dojotoolkit.org/>
- Kuchana. *Software Architecture Design Patterns in Java*. Auerbach, 2004. p.
- Kusakov. *Develop HTML widgets with Dojo*. 2006. 36 p.
- Larman. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.*, Prentice-Hall  
Hispanoamerica, 1999. p. 8425
- Orizondo. *Proyecto Técnico de Asesoría Especializada, Colaboración Médica Odontológica,  
Comunicación Institucional y Solución Tecnológica para apoyar la modernización del Sistema  
Penitenciario de la República Bolivariana de Venezuela*. Venezuela, 2006. p.
- Pérez. *Introducción a JavaScript*. 2008. 185 p.
- Pimentel & Martínez. *Documento de Arquitectura de Software*. Cuba, 2008. p.
- Pimentel & Rivero. *ArBaWeb: Arquitectura Base sobre la Web*. La Habana, Universidad de las Ciencias  
Informáticas, 2007. p.
- SETH LADD. *Expert Spring MVC and Web Flow*. 2006. p. 978-1-59059-584-8
- Wiley. *Professional Java Development with the Spring Framework*. 2005. p. 0764574833

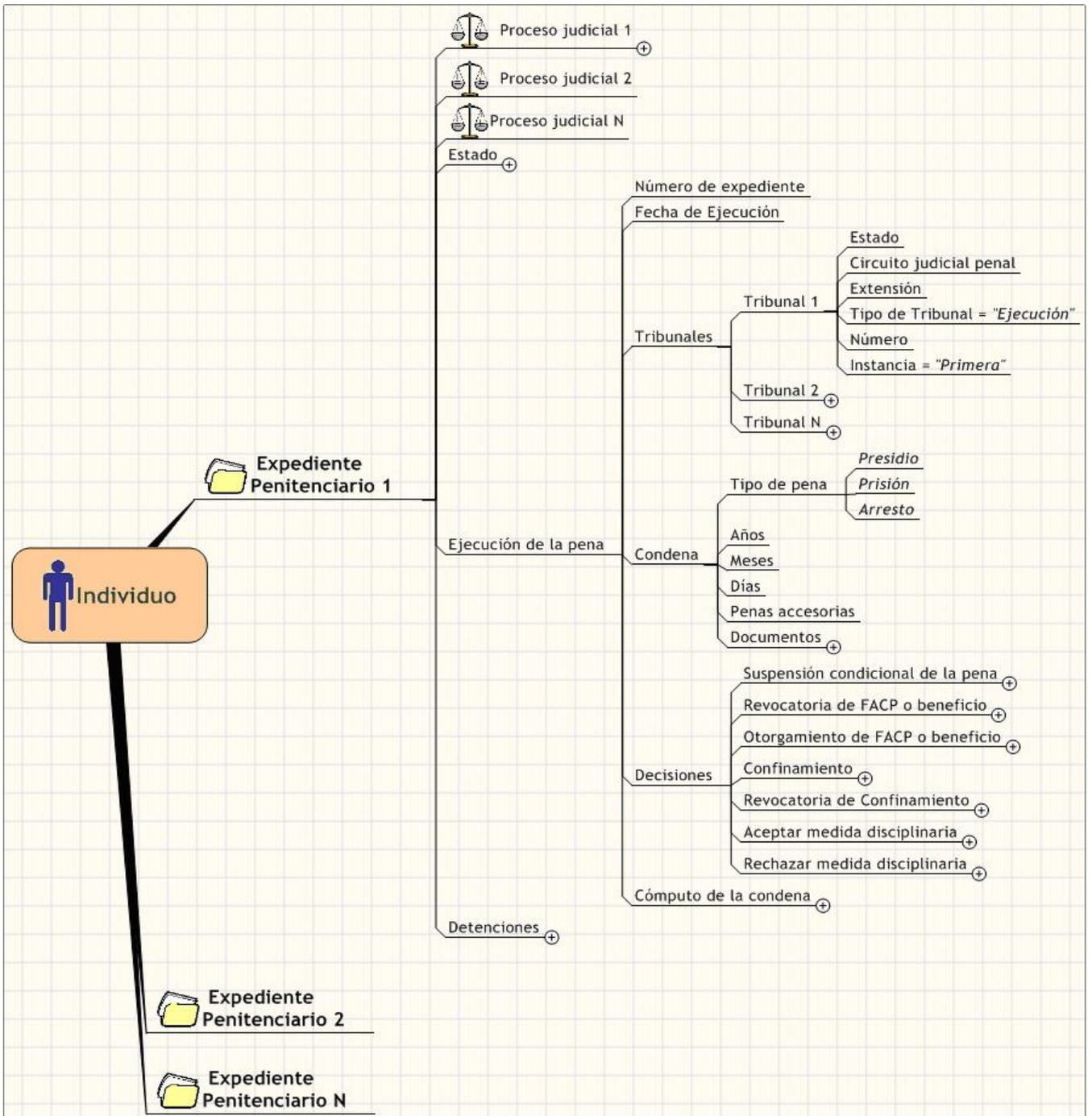


Anexo 1: Expediente del individuo.

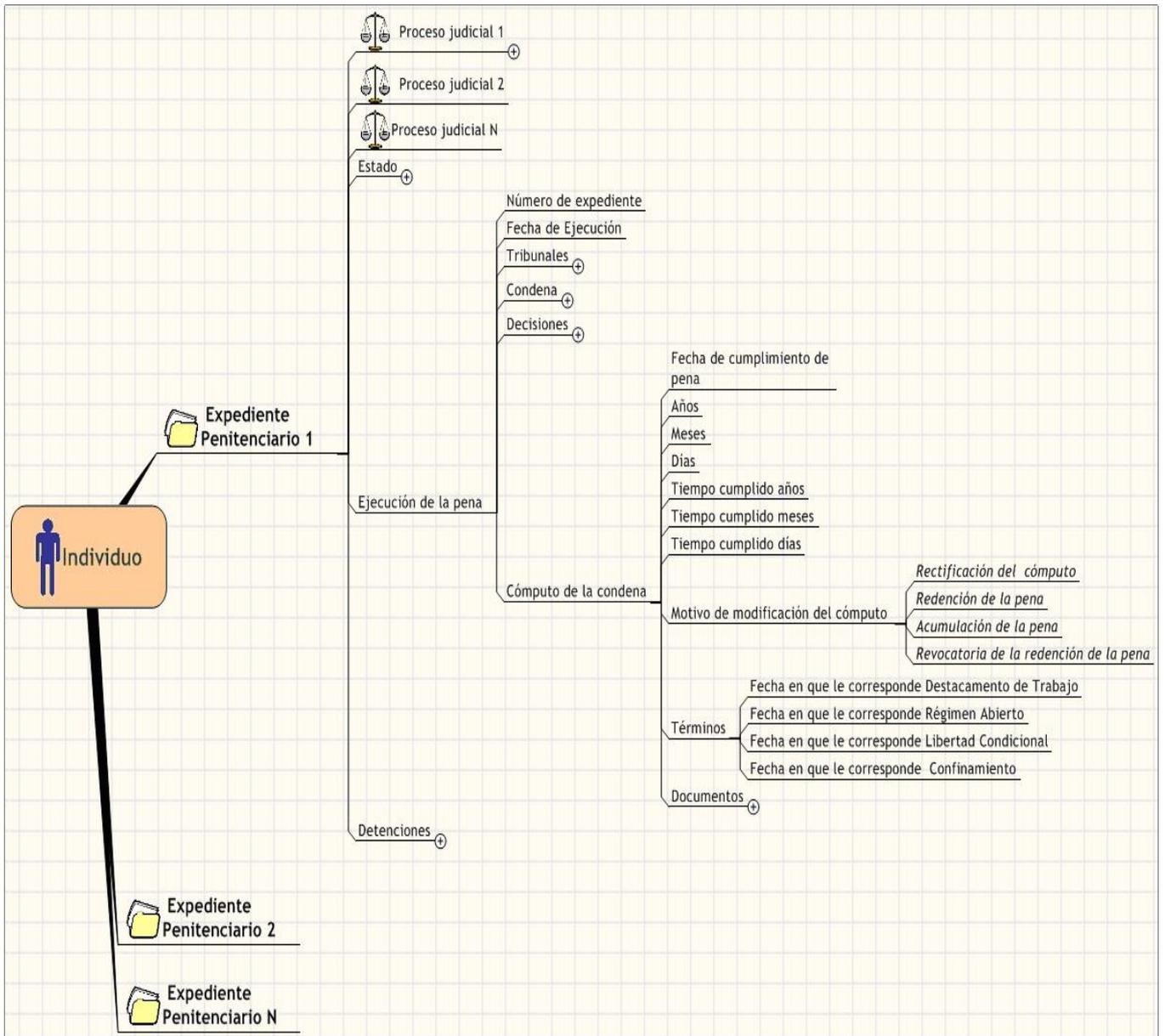


**Anexo 2: Proceso Judicial.**

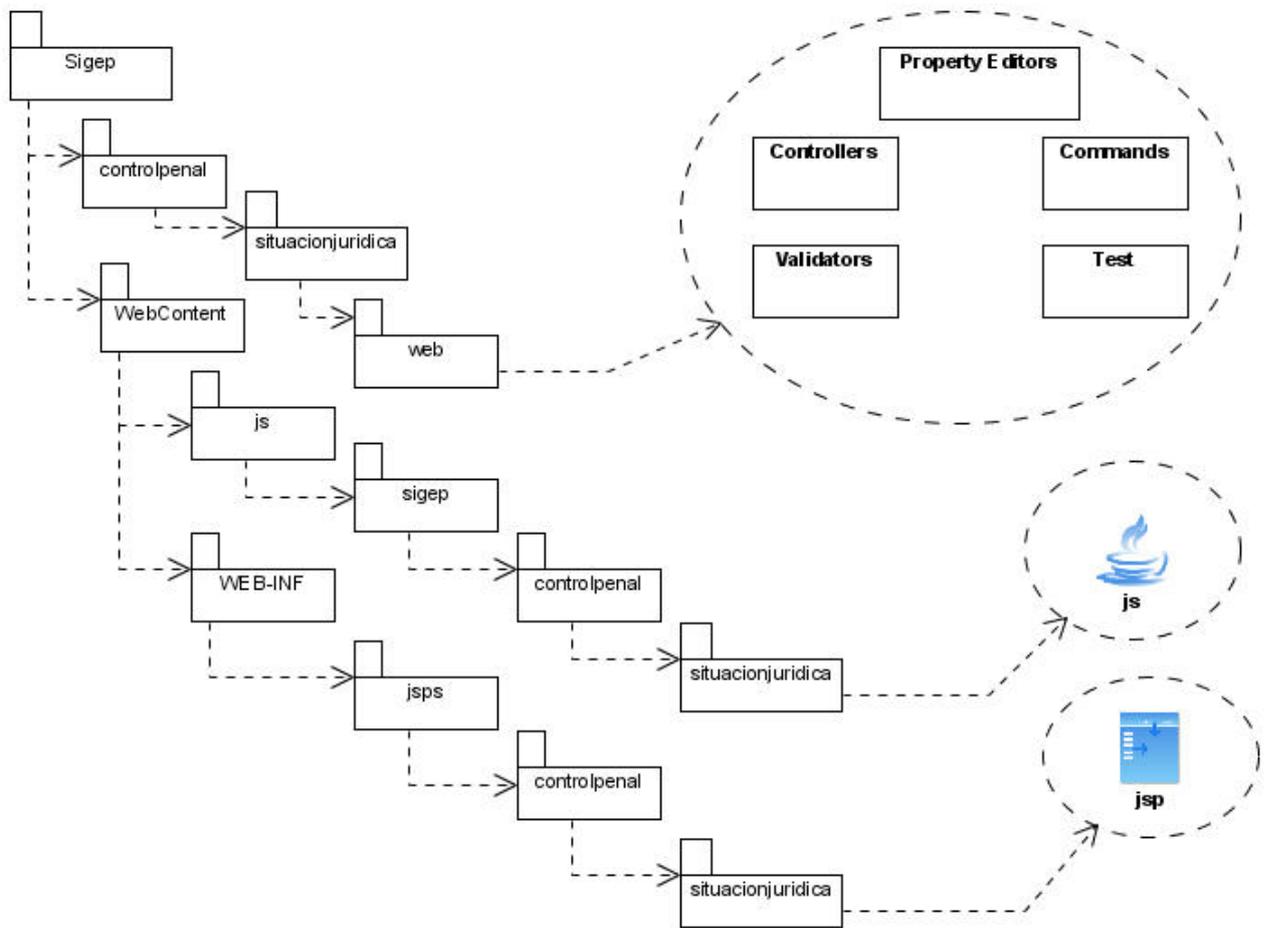




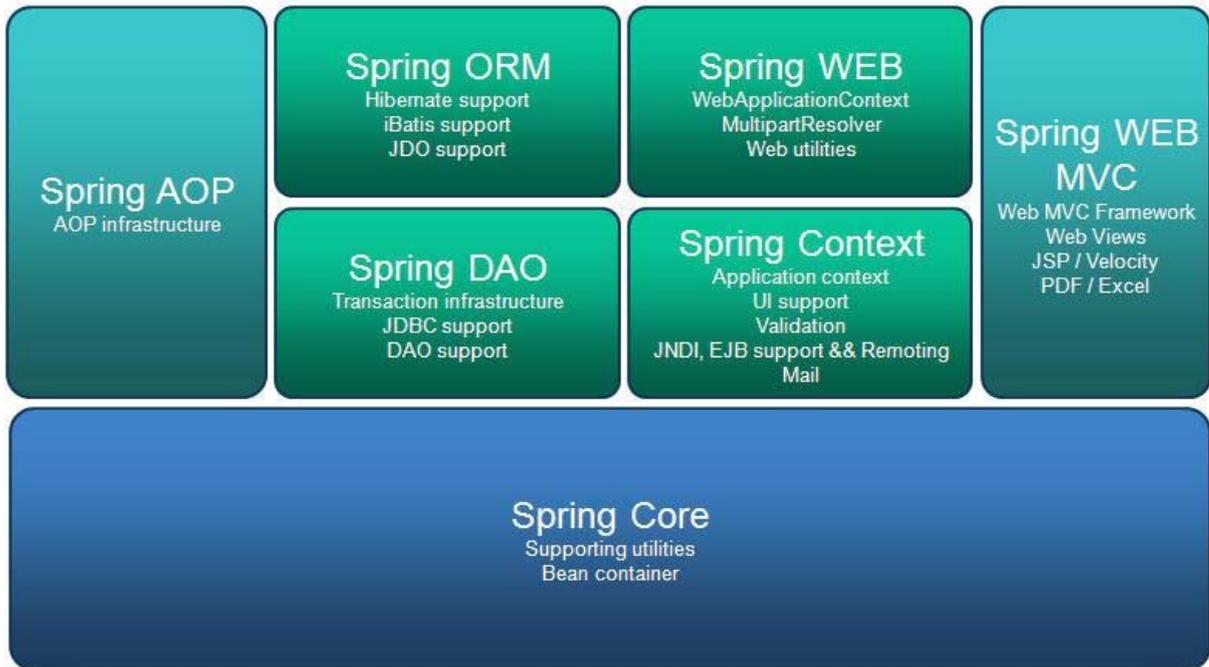
Anexo 4: Ejecución de la Pena.



**Anexo 5: Cómputo de la Condena.**



**Anexo 6: Paquetes del Modulo Situación Jurídica.**



**Anexo 7: Módulos de Spring.**

### **Sistema Penitenciario**

Al asignar al régimen penitenciario la calificación de Sistema Penitenciario se hace referencia al conjunto de normas, procedimientos y dependencias dispuestas por el Estado para la ejecución del régimen penitenciario entre los que se encuentran además los principios, programas, recursos humanos, dependencias e infraestructura que se encuentran relacionadas y destinadas a este régimen.

### **Control penal**

Es el proceso del Sistema Penitenciario encargado de garantizar el cumplimiento de la legalidad, la organización, control y tramitación de la documentación legal del procesado o penado durante el ingreso, reingreso, permanencia y egreso de los lugares de internamiento.

### **Situación Jurídica**

Situación jurídica es la encargada de gestionar información sobre los procesos judiciales de un individuo. La cual tiene como objetivo garantizar las fases por las que transcurre cada uno de los procesos, las decisiones tomadas en cada fase y la ejecución de la pena en el Sistema Penitenciario.