

*Universidad de las Ciencias Informáticas*  
*Facultad 2*



*Sistema de Gestión Policial (SIGEPOL)*  
*“Módulo de Consejos Disciplinarios”*

**Trabajo de Diploma**

Presentado para optar por el título de Ingeniero en Ciencias Informáticas.

***Autores:***

Leydis Katlhen Guerra López

Darien Cabrera Peláez

***Tutor:***

Ing. Elieser Bello Ross

“Año 53 Aniversario del Triunfo de la Revolución”

13 de junio del 2011, Ciudad de la Habana, Cuba.

## **Declaración de Autoría**

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

\_\_\_\_\_

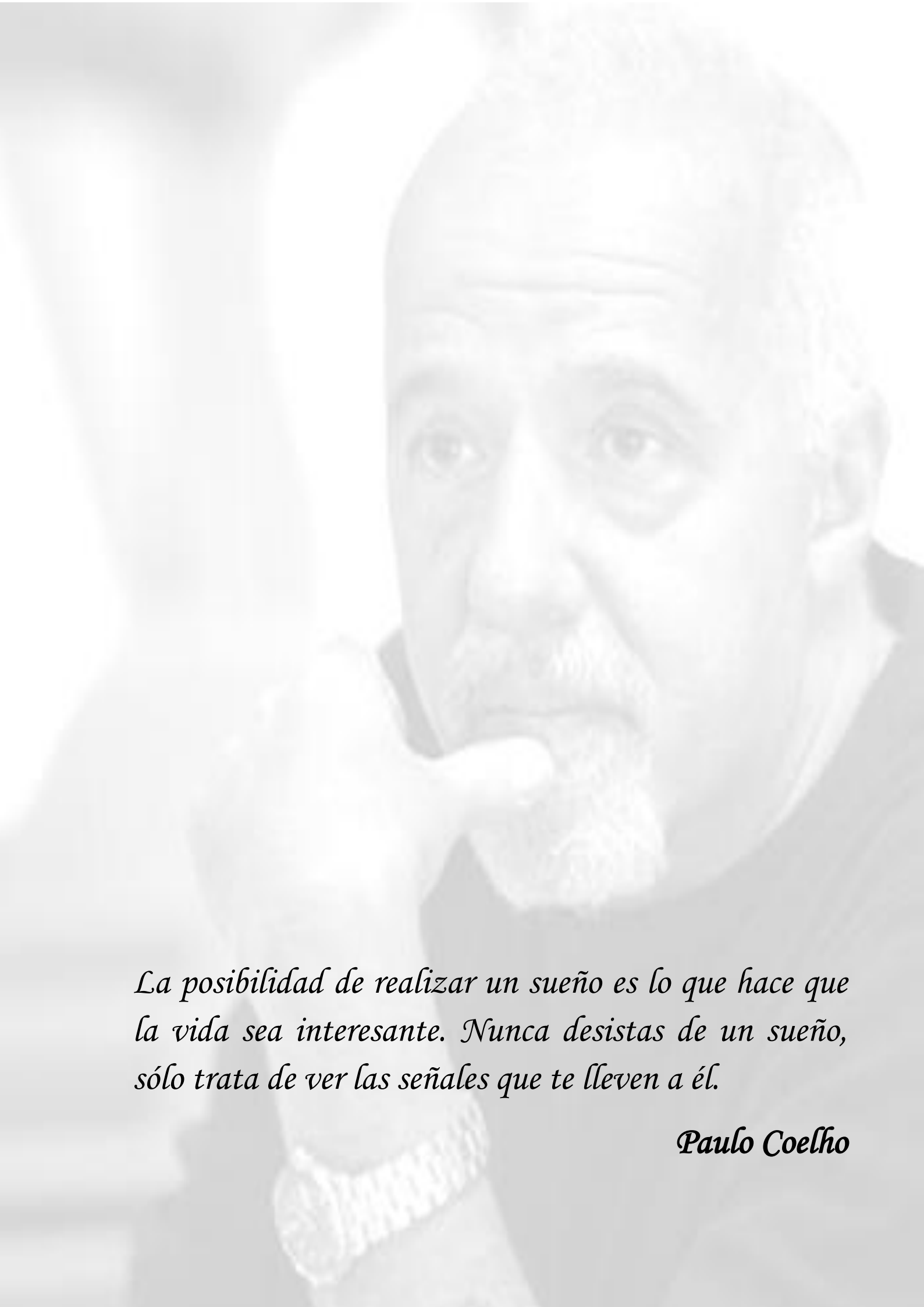
**Autor: Darien Cabrera Peláez.**

\_\_\_\_\_

**Autor: Leydis Katlhen Guerra López.**

\_\_\_\_\_

**Tutor: Ing. Elieser Bello Ross**



*La posibilidad de realizar un sueño es lo que hace que la vida sea interesante. Nunca desistas de un sueño, sólo trata de ver las señales que te lleven a él.*

**Paulo Coelho**

## *Dedicatoria*

### *Leydis Katlhen:*

*Les dedico este trabajo en primer lugar a mi papá y mi mamá, quienes me han dado fuerzas para seguir adelante a lo largo de estos cinco años, aparte de su amor incondicional y su cariño. A mi hermano que lo quiero mucho y a mi tía por su guía profesional. A mi novio por su apoyo y comprensión, y a todos mis amigos.*

### *Darien:*

*A mis padres por todo el amor y cariño que me han dado en todos estos años, mis hermanos por la dedicación y preocupación durante mi vida estudiantil. A mi familia que los quiero mucho, a mi novia y a todos mis amigos que siempre han estado a mi lado.*

*Agradecimientos*

*Le agradecemos a todo el equipo de desarrollo de Sigepol, y a todos los que de una manera u otra hicieron posible el desarrollo del trabajo.*

## **Resumen**

En la actualidad la seguridad ciudadana es uno de los grandes problemas que enfrentan las sociedades. Su impacto en la vida de los ciudadanos hace que los gobiernos tomen medidas y realicen planes alternativos para disminuir los niveles de inseguridad. En la población de la República Bolivariana de Venezuela están presentes problemas como el desempleo y la vivienda; pero uno de los más preocupantes es la inseguridad ciudadana. La cantidad de delitos que se cometen en esta nación ha originado gran incertidumbre.

Teniendo en cuenta la preocupación mostrada por el gobierno venezolano por erradicar las diversas causas generadoras de delitos, se crea el Viceministerio del Sistema Integrado de Policía (VISIPOL) que promueve la creación del Sistema de Gestión Policial (SIGEPOL), que tiene como objetivo gestionar la información policial a nivel nacional, que sirva de base a la toma de decisiones y ayude a combatir el delito en la República Bolivariana de Venezuela. Como parte del SIGEPOL se define el Módulo de Consejo Disciplinario que gestiona la información de los candidatos a los consejos disciplinarios, la conformación de la lista nacional y las listas regionales; así como la creación y supervisión de los Consejos Disciplinarios.

El documento que se presenta a continuación recoge los resultados del trabajo investigativo realizado. Para el desarrollo de la aplicación web que se propone se emplea: RUP como metodología de desarrollo de software, Visual Paradigm como herramientas CASE, Grails como framework a utilizar y Groovy como lenguaje de programación.

Con el desarrollo de este módulo, se podrán conformar los Consejos Disciplinarios de forma automatizada y permitirá que exista un registro centralizado de la información de los casos evaluados en los consejos de los cuerpos de policías, lo que contribuirá a disminuir el delito.

## Índice

|   |    |
|---|----|
| Introducción .....  | 1  |
| Capítulo 1 Fundamentación del Tema .....                                      | 5  |
| 1.1. Introducción .....   | 5  |
| 1.2. Seguridad Ciudadana .....  | 5  |
| 1.3. Seguridad Ciudadana en Venezuela .....                                   | 5  |
| 1.4. Descripción de la Dirección específica a la que responde el Módulo ..... | 6  |
| 1.4.1. Propuesta del Proceso Automatizado de Consejos Disciplinarios.....     | 7  |
| 1.5. Conclusiones .....   | 8  |
| Capítulo 2 Metodología y Herramientas de Desarrollo .....                     | 10 |
| 2.1. Introducción.....  | 10 |
| 2.2. Metodologías de Desarrollo de Software .....                             | 10 |
| 2.2.1. Metodología RUP.....   | 10 |
| 2.2.2. Lenguaje Unificado de Modelado (UML) .....                             | 11 |
| 2.3. Herramienta CASE .....   | 11 |
| 2.3.1. Visual Paradigm para UML .....   | 12 |
| 2.4. Plataforma de Desarrollo de Software .....                               | 13 |
| 2.4.1. Plataforma Java .....  | 13 |
| 2.4.2. Lenguaje de Programación Java.....                                     | 14 |
| 2.4.3. Lenguaje de Programación Groovy.....                                   | 14 |
| 2.4.4. Máquina Virtual de Java (JVM) .....                                    | 15 |
| 2.5. Frameworks .....   | 15 |
| 2.5.1. Dojo .....   | 15 |
| 2.5.2. Grails .....   | 16 |
| 2.6. Entorno Integrado de Desarrollo (IDE) .....                              | 17 |
| 2.6.1. SpringSource Tool Suite (STS) .....                                    | 17 |
| 2.7. Gestor de Base de Datos .....  | 18 |
| 2.7.1. Gestor de Base de Datos Oracle 10g release 2 .....                     | 18 |
| 2.8. Conclusiones.....  | 18 |
| Capítulo 3 Diseño del Sistema.....  | 19 |
| 3.1. Introducción.....  | 19 |
| 3.2. Patrones de Diseño.....  | 19 |
| 3.3. Patrones de Arquitectura.....  | 20 |
| 3.4. Descripción de la Arquitectura.....                                      | 23 |

|  |    |
|--|----|
| 3.5. Diagrama de Paquetes.....                           | 25 |
| 3.6. Diagrama de clases.....                             | 27 |
| 3.6.1. Diagramas de Clases de la Capa Presentación ..... | 27 |
| 3.7. Descripción de las principales clases .....         | 34 |
| 3.8. Conclusiones.....                                   | 40 |
| Capítulo 4 Implementación de la solución.....            | 41 |
| 4.1. Introducción.....                                   | 41 |
| 4.2. Modelo de Implementación.....                       | 41 |
| 4.2.1. Diagrama de Componentes .....                     | 41 |
| 4.2.2. Descripción de los Componentes.....               | 41 |
| 4.3.Descripción de los casos de pruebas .....            | 47 |
| 4.3.1. SC Registrar Conformación de listas .....         | 47 |
| 4.3.1. SC Crear Consejo Disciplinario.....               | 51 |
| 4.4. Conclusiones.....                                   | 52 |
| 5. Conclusiones Generales.....                           | 53 |
| 6. Recomendaciones .....                                 | 54 |
| 7. Referencias Bibliográficas.....                       | 55 |
| 8. Bibliografía.....                                     | 57 |
| ANEXO.....   | 58 |



### **Introducción**

La seguridad ciudadana es uno de los primordiales deberes desde el punto de vista social que todo país debe garantizar, de forma tal, que los ciudadanos queden satisfechos con la actuación de los cuerpos de seguridad. Garantizando la seguridad se puede asegurar una convivencia pacífica y una disminución de la violencia en un país; así como contribuir a la prevención de delitos y faltas.

En el año 2010 se evidenció una vez más que en Venezuela la muerte y la violencia se convirtieron en algo común entre sus 29 millones de habitantes. Venezuela se encuentra en los primeros puestos del ranking de países con mayor número de homicidios de la región, registrando en este año, 17.600 asesinatos, según cifras de la organización Observatorio Venezolano de Violencia. (1)

La preocupación demostrada por el gobierno venezolano por erradicar diversas causas y condiciones generadoras de delitos y de otras condiciones de conductas delictuales se expresan de forma clara y palpable en la implementación de un conjunto de políticas y misiones sociales aplicadas en estos últimos años.

Por esta razón el Comandante Hugo Rafael Chávez Frías, Presidente Constitucional de la República Bolivariana de Venezuela, a través del Reglamento Orgánico del Ministerio del Poder Popular para Relaciones Interiores y Justicia, según Decreto N° 6.733 de fecha 09 de Junio de 2009 y publicado en la Gaceta Oficial de la República Bolivariana de Venezuela N° 39.196 de fecha 09 de Junio 2009, creó el Viceministerio del Sistema Integrado de Policía (VISIPOL) que tiene como misión promover y alcanzar en el plano interno del Sistema Integrado de Policía una sinergia en el funcionamiento de sus procesos y en la interrelación de los diferentes Cuerpos de Policía y de Vigilancia y Seguridad Privada; y en el plano externo la armonía y complementación con los órganos que participan en la formulación y ejecución de políticas públicas y procedimientos que contribuyan al pacífico disfrute de los derechos de los ciudadanos y de las instituciones. (2)

El Viceministerio del Sistema Integrado de Policía, cuenta con la Oficina Nacional de Supervisión Disciplinaria de los Cuerpos de Policías, que establece las responsabilidades individuales de funcionarios policiales por la realización de faltas, infracciones y desviaciones, así como promover las buenas prácticas. Estos Consejos Disciplinarios darán seguimiento al debido proceso de efectivos involucrados en infracciones permitiendo la participación del pueblo.

La selección de los integrantes de estos consejos se realiza con base en las listas nacionales y regionales para la conformación de los Consejos Disciplinarios de Policía. Los criterios aplicados incluyen ser de nacionalidad venezolana o tener más de 10 años de residencia en el país, ser persona hábil mayor de 21 años con título de educación media diversificada, tener buena conducta e idoneidad moral, no haber sido destituido o destituida de un órgano o ente del Estado, no tener antecedentes penales y, en los casos de las listas regionales, encontrarse residenciado en la región para la que se postula. Para más información en cuanto a las leyes que rigen los Consejos Disciplinarios referirse al anexo. (3)

Entre las principales problemáticas que presenta esta dirección están las siguientes:

No existe un registro centralizado de la información resultante de los procesos de investigaciones de los casos evaluados en Consejos Disciplinarios, que permita conocer los procedimientos, las faltas más frecuentes y elementos de interés que posibiliten evaluar las causas y condiciones que las favorecen.

Se realiza un control manual de la información de los funcionarios miembros de los Consejos Disciplinarios y de los que forman parte de las listas regionales y nacionales de integrantes de los Consejos Disciplinarios. Esto dificulta desarrollar actividades dirigidas a la formación inicial y continua de estos funcionarios, así como garantizar el disfrute si fuese necesario de las garantías y medidas de protección contempladas en la Ley de Protección de Víctimas y Testigos.

Para solucionar las situaciones antes mencionadas, se plantea el siguiente problema científico:

¿Cómo gestionar la información referente a la conformación, modificación y actuación de los Consejos Disciplinarios de los Cuerpos de Policía?

El objeto de investigación lo constituye el proceso de gestión de los Consejos Disciplinarios en los Cuerpos de Policía, siendo el campo de acción los procesos de gestión de los Consejos Disciplinarios en los Cuerpos de Policía de la República Bolivariana de Venezuela.

Entre las principales funcionalidades del Módulo de Consejos Disciplinarios del Sistema de Gestión Policial están:

- Registrar los funcionarios que integran los consejos disciplinarios.
- Gestionar la lista nacional y las listas regionales de candidatos a integrar los Consejos Disciplinarios.
- Registrar los resultados de investigaciones de los casos evaluados en Consejos Disciplinarios.

- Registrar las intervenciones directas a los Consejos Disciplinarios ante una mala actuación de los mismos.

Para darle solución al problema planteado se definió como objetivo general desarrollar el módulo de Consejos Disciplinarios del Sistema de Gestión Policial a partir del análisis de los requisitos de software, haciendo uso de las tecnologías y metodologías definidas en el proyecto.

Para cumplir con el objetivo mencionado, se plantea las siguientes tareas a realizar:

1. Estudiar e investigar las tecnologías y herramientas a usar en el desarrollo del módulo.
2. Investigar las leyes que rigen el registro de funcionarios y el control de los Órganos Policiales y de Seguridad Ciudadana en Venezuela para conformar el Marco Legal.
3. Analizar los procesos de negocio de los Consejos Disciplinarios.
4. Estudiar los Casos de Uso correspondiente al módulo Consejos Disciplinarios.
5. Realizar el diseño del módulo a partir del análisis del negocio.
6. Implementar los elementos de diseño en términos de componentes de implementación.

Para cumplir con las tareas antes mencionadas se desarrollará un producto de software flexible, seguro, confiable y orientado a las necesidades del cliente, acorde con las tecnologías de punta en el ámbito informático, las leyes venezolanas vigentes y los requerimientos necesarios.

### **Métodos de Investigación Utilizados**

#### **Métodos teóricos**

- **Analítico - Sintético:** Para procesar toda la información enfocada hacia la investigación de los procesos del sistema, permitiendo organizar y simplificar el análisis de todo el volumen de datos a recopilar en fracciones más factibles.
- **Histórico:** Para el estudio de antecedentes que existen respecto a las herramientas, metodologías y lenguajes de programación que son utilizados en el desarrollo del sistema.
- **Lógico: -método de modelación:** se ve reflejado durante todo el desarrollo, que se utilizan diagramas que reflejan la estructura y sus relaciones como son el diagrama de paquetes, diagrama de clases.

#### **Métodos empíricos**

- **Observación:** Revisión de documentos y sitios web para apoyar el estudio de las diferentes herramientas y metodologías utilizadas en la aplicación.

El presente documento está compuesto por 4 capítulos:

En el Capítulo 1 “**Fundamentación del Tema**” se exponen los fundamentos generales que sirven de soporte teórico en la solución del problema. Se describen los procesos principales del Módulo de Consejos Disciplinarios.

En el Capítulo 2 “**Metodología y Herramientas de Desarrollo**” se describen teóricamente las herramientas y lenguajes de programación que se utilizará, para comprobar que son las idóneas para el desarrollo del Módulo de Consejos Disciplinarios. Además se plantea la metodología a emplear en el desarrollo del mismo.

En el Capítulo 3 “**Diseño del Sistema**” se construye la solución propuesta, se modelan diagramas de clases que representan las funcionalidades del Módulo de Consejos Disciplinarios aplicando los patrones de arquitectura y diseño seleccionados.

En el Capítulo 4 “**Implementación de la Solución**” se representa el diagrama de componentes que detalla la forma en que está estructurado el sistema, reflejando la transformación de los elementos del modelo del diseño en términos de componentes, ficheros que contienen código fuente, ejecutables, así como las dependencias entre ellos.

## **Capítulo 1 Fundamentación del Tema**

### **1.1. Introducción**

El objetivo fundamental de este capítulo es realizar la investigación sobre la conformación y supervisión de los Consejos Disciplinarios; particularmente los procesos de registrar los funcionarios, gestionar la lista nacional y las listas regionales de los candidatos para integrar los consejos y exponer los fundamentos generales que sirven de soporte teórico en la solución y concepción del problema.

### **1.2. Seguridad Ciudadana**

La seguridad ciudadana es la acción integrada que desarrolla un país, con la colaboración de la ciudadanía, destinada a asegurar la convivencia pacífica y contribuir a la prevención de ocurrencias de delitos e infracciones. Está encaminada a prevenir males como la violencia, las infracciones y el crimen; acciones que generan diversos peligros para la sociedad y para el desarrollo de las actividades económico-socio-culturales en general.

Se enmarca en la situación o condición de una comunidad donde de manera generalizada sus habitantes puedan hacer pacífico ejercicio de la generalidad de sus derechos, sin verse impedidos de ello por acciones de violencia, de cualquier tipo o naturaleza. (4)

La falta de seguridad ciudadana ha pasado a ser en los últimos tiempos uno de los temas centrales de preocupación de los ciudadanos y, por lo tanto, una de las cuestiones a resolver por los responsables políticos de principios de este siglo. (5)

### **1.3. Seguridad Ciudadana en Venezuela**

En la actualidad la convivencia pacífica en las sociedades se encuentra amenazada por la existencia de tensiones y conflictos que generan conductas violentas y que han surgido por diferentes causas. Entre las que se pueden señalar a dos de ellas que son complementarias, una es la crisis económica que afecta a la mayor parte de los países del mundo y la crisis de valores, que ha generado pobreza, marginalidad, desempleo, drogadicción, alcoholismo, corrupción, pérdida de identidad, pérdida de confianza en el otro, etc. (6)

En Venezuela la inseguridad de la población ha crecido hasta convertirse en el país con mayor número de homicidios, si bien en Caracas la cifra es de 135 homicidios por cada

100.000 personas, en comparación con otros países esta cifra está muy por encima con valores de: México 22, Brasil 20 y Chile 4 por cada 100.000 habitantes. Según el Observatorio Venezolano de la Violencia los homicidios son la principal causa de muerte entre los jóvenes del país. En un estudio de la UNESCO realizado en 57 países, Venezuela es el país con más muertes por armas de fuego. (7)

En ese sentido el Ministerio del Poder Popular para las Relaciones Interiores y Justicia (MPPRIJ) conjunto con los Órganos de Seguridad Ciudadana ha tomado medidas para el combate y prevención del delito, así como asegurar una convivencia pacífica en el país. En este marco se crea el Viceministerio del Sistema Integrado de Policía (VISIPOL) y dentro de él la puesta en marcha del Sistema de Gestión Policial con la finalidad de ayudar a coordinar las acciones de los órganos policiales.

#### **1.4. Descripción de la Dirección específica a la que responde el Módulo**

Los Consejos Disciplinarios tienen como objetivo establecer las responsabilidades individuales de funcionarios policiales que cometen faltas, infracciones y delitos que tributen a desviaciones, así como también promover las buenas prácticas. Entre las competencias de esta instancia está decidir mediante recomendación u opiniones los procedimientos disciplinarios que le siguen a los funcionarios en los casos de falta sujeta a la destitución, la organización buscará erradicar las conductas desviadas en el seno de la policía y de informar a los directivos sobre las irregularidades que se presenten, para más información sobre el marco legal referirse al anexo. (8)

En la República Bolivariana de Venezuela cada Cuerpo de Policía controla y almacena por separado la información resultante de los procesos de investigaciones de los casos evaluados en Consejos Disciplinarios que permite conocer los procedimientos, las faltas más frecuentes y elementos de interés. Por lo que existen funcionarios policiales que tienen sus procesos listos para la destitución y no se han podido excluir de la institución. Incluso, hay funcionarios, en flagrancia, cometiendo delitos y aún siguen activos.

Por lo tanto, un control manual e individual de la información de los funcionarios miembros de los Consejos Disciplinarios y de los que forman parte de las listas regionales y nacionales de integrantes de los Consejos Disciplinarios de los Cuerpos de Policía, trae como consecuencia directa el inconveniente de que no existe un control centralizado de la información. Esto dificulta desarrollar actividades dirigidas a la formación inicial y continua de estos funcionarios,

así como garantizar el disfrute si fuese necesario de las garantías y medidas de protección contempladas en la Ley de Protección de Víctimas y Testigos. Además de no tener una visión general de la documentación de los consejos. A continuación, una breve descripción acerca de las funcionalidades generales de dichos Consejos Disciplinarios.

### **1.4.1. Propuesta del Proceso Automatizado de Consejos Disciplinarios**

Los Consejos Disciplinarios se constituyen para conocer y opinar sobre los procedimientos y faltas sujetas a sanción de destitución que sean cometidas por funcionarios policiales.

Los procesos con los que se encuentran relacionados se inician cuando se introduce la cédula de un postulante a candidato para buscarlo. Si no se encuentra registrado se introducen los datos especificados en dependencia del tipo de persona, las cuáles pueden ser funcionario policial activo, funcionario policial jubilado y personas civiles.

Después de estar registrado el nuevo candidato, se verifica los antecedentes en el Sistema Integrado de Información Policial (SIIPOL) y además el historial personal en caso de que el candidato sea un funcionario policial, luego se registra si el candidato está aprobado o denegado. Posteriormente se verifica en el Poder Moral, donde será aprobado o denegado.

A continuación se pasa a conformar las listas, que puede ser de dos tipos, la lista nacional y las listas regionales. Primero se escoge un candidato en condición de titular y dos candidatos en condición de suplentes para integrar el tipo de lista seleccionada y luego de ser conformadas estas pueden ser publicadas.

Para conformar la propuesta del cuerpo de policía se obtiene todos los funcionarios policiales de dicho cuerpo de policía, y se selecciona un funcionario en condición de titular y dos funcionarios en condición de suplentes.

Además si el candidato comete alguna infracción o falta, este pierde su condición como candidato a formar parte de un consejo disciplinario. Si la persona que se elimina por pérdida de condición es integrante de un consejo disciplinario y tiene condición de "Titular" el suplente que le sigue pasa a ser "Titular" y si la persona que se elimina tiene condición de primer suplente, el suplente que le sigue toma la condición como primer suplente.

Después que están creadas las listas y se tiene la propuesta del cuerpo de policía se pasa a crear el consejo disciplinario, se selecciona el cuerpo de policía al que desea crearle el

consejo. Este consejo disciplinario tendrá una propuesta completa y aprobada de titular y suplentes del mismo cuerpo de policía que lo solicito que no este asignado a algún otro consejo disciplinario y una propuesta de titular y suplentes de la lista nacional y una propuesta de titular y suplentes de la lista regional.

Posteriormente se crean las guías de supervisión a los Consejos Disciplinarios, para cuando se desee supervisar. Estas supervisiones las realiza un funcionario, el cual debe poner una decisión a dicha supervisión y si tiene alguna recomendaciones estas son introducidas en averiguaciones administrativas.

A raíz de estas supervisiones se pueden encontrar consejos disciplinarios ilegales, estos son los consejos que se encuentran en los cuerpos de policías constituidos ilegalmente por algún motivo y estos son registrados para tener constancia de ellos.

### **1.5. Conclusiones**

En este capítulo se realizó un análisis de los principales conceptos que deben conocerse para comprender la solución que se propone. Se investigó sobre el marco constitucional y legal de la seguridad ciudadana en cuanto a los Consejos Disciplinarios que se encuentra en el anexo y se describió como transcurren las funcionalidades generales a las que responde el módulo de Consejos Disciplinarios.



## **Capítulo 2 Metodología y Herramientas de Desarrollo**

### **2.1. Introducción**

En este capítulo se describe teóricamente la metodología, se fundamenta e investiga la elección de las tecnologías y herramientas, necesarias para llevar a cabo el desarrollo del módulo.

### **2.2. Metodologías de Desarrollo de Software**

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software. Estas metodologías contribuyen a mejorar la calidad y a realizar un software en el tiempo esperado y con el coste estimado. (9)

Para la selección de esta metodología se debe tener en cuenta la duración del proyecto, así como la frecuencia de los cambios a los que este puede estar sujeto, pues no existe una metodología de desarrollo de software universal. Por lo que las características de cada proyecto exigen que el proceso sea configurable.

#### **2.2.1. Metodología RUP**

La metodología RUP (Rational Unified Process en inglés), es un proceso de desarrollo de software, el cual es un ejemplo de las llamadas Metodologías Pesadas, las cuales ponen especial énfasis en la planificación y control del proyecto.

Existen además las denominadas Metodologías Ágiles, dirigidas sobre todo a equipos de desarrollo pequeños, con mayor fuerza en aspectos humanos asociados al trabajo en equipo, involucran al cliente en el proceso como parte activa del propio equipo de desarrollo y están orientadas sobre todo a la generación de código con ciclos cortos de desarrollo, pero no resulta factible utilizar este tipo de metodologías teniendo en cuenta que el cliente procede de otro país y no puede formar parte del equipo de desarrollo, además el equipo de desarrollo es grande e inestable, pues en su mayoría son estudiantes y profesores que pueden pasar a cumplir otras funciones en cualquier momento, por lo que se necesitará una buena organización y abundante documentación para que el avance del proyecto no se vea afectado y se asegure la continuidad del mismo.

Con el objetivo de cumplir con los requerimientos del cliente a la hora de entregar un producto de software, es necesario el uso de una metodología de desarrollo que nos permita darle organización al trabajo, para así realizar la entrega con las variables de tiempo, costo y calidad dentro de los valores previamente acordados.

Para elegir la metodología de desarrollo adecuada es necesario mirar al proyecto que se quiere desarrollar desde adentro para determinar su alcance, posible duración y complejidad. Luego de analizar lo que se propone con el trabajo y teniendo en cuenta el posible volumen de procesos a informatizar se llega a la conclusión de que la metodología más factible sería RUP.

Esta metodología es especialmente efectiva para proyectos de gran magnitud como SIGEPOL, además de poseer una gran adaptabilidad cuando se habla de proyectos a largo plazo. Por otro lado la filosofía de casos de usos y arquitectura es perfecta para describir la amplia gama de funcionalidades que se requiere implementar. Da la posibilidad de generar la documentación necesaria que garantizaría la continuidad del proyecto.

### **2.2.2. Lenguaje Unificado de Modelado (UML)**

El UML es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. (10)

Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos, permite la modelación de sistemas con tecnología orientada a objetos. Los diagramas son entes importantes de UML, cuya finalidad es presentar como diversas perspectivas de un sistema, a las cuales se les modela. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementarlo. El modelo gráfico de UML tiene un vocabulario en el que se identifican: elementos, relaciones y diagramas.

La causa fundamental por la cual es usado en el desarrollo del sistema es que el equipo de desarrollo tiene experiencia en el trabajo con el mismo.

### **2.3. Herramienta CASE**

Las herramientas CASE (Ingeniería de Software Asistida por Ordenador) son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el coste de las mismas en términos de tiempo y dinero.

El desarrollo de un nuevo software requiere que las tareas sean organizadas y completadas de forma correcta y eficiente. Las Herramientas CASE fueron desarrolladas para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software. Permiten incrementar la productividad y el control de calidad en cualquier proceso de elaboración de software, ya que transforman el desarrollo de software en un proceso automatizado. (11)

### **2.3.1. Visual Paradigm para UML**

Visual Paradigm para UML es una herramienta CASE profesional, fácil de utilizar y que soporta el ciclo de vida completo del desarrollo de software; usa UML como lenguaje de modelado, característica que ayuda a la construcción de aplicaciones con calidad, de manera intuitiva, a menor coste; además de que facilita la comunicación entre los miembros del equipo de desarrollo, al garantizar el uso de un lenguaje estándar común. (12)

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

- Disponibilidad en múltiples plataformas (multiplataforma): Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris.
- Brinda un número considerable de estereotipos, lo que permite un mayor entendimiento de los diagramas.
- Generación de código e ingeniería inversa: brinda la posibilidad de generar código a partir de los diagramas, para plataformas como Java; también se obtienen diagramas a partir del código, lo que facilita en gran medida el entendimiento entre analistas e implementadores.
- Integración con distintos Ambientes de Desarrollo Integrados (IDE): se integra fácilmente con varios IDE de desarrollo, entre los que se encuentra Eclipse, que es el que se utilizará para el desarrollo de la aplicación.
- Generación de documentación: brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.

### 2.4. Plataforma de Desarrollo de Software

Una plataforma de software constituye el ambiente de hardware o en el caso de software un conjunto de librerías y programas en el cual se ejecutan aplicaciones. Un ejemplo de plataforma lo constituye un entorno de programación, sistema operativo, o generalmente la composición de ambos.

#### 2.4.1. Plataforma Java

La plataforma Java es un entorno o plataforma de computación creada por la Sun Microsystems, capaz de ejecutar aplicaciones desarrolladas con el uso del lenguaje de programación Java y un conjunto de herramientas de desarrollo. En este caso, la plataforma no es un hardware específico o un sistema operativo, sino una máquina virtual encargada de la ejecución, y un conjunto de librerías estándares que ofrecen funcionalidades comunes. (13)

Para el desarrollo de la aplicación se seleccionó la Edición Empresarial: J2EE o Java EE (Java Platform, Enterprise Edition).

Una de las principales razones por la que se seleccionó esta plataforma es porque se cuenta con limitaciones legales por parte del cliente: el decreto Ley No. 3.390 de la Gaceta Oficial de la República Bolivariana de Venezuela establece que la Administración Pública Nacional empleará prioritariamente Software Libre Desarrollado con Estándares Abiertos, en sus Sistemas, Proyectos y Servicios Informáticos, permitiendo el uso de otro tipo de software solo en casos de no ser posible la adquisición de Software Libre Bajo Estándares Abiertos, en tal caso, los órganos y entes de la Administración Pública Nacional deberán solicitar ante el Ministerio de Ciencia y Tecnología, la autorización para adoptar otro tipo de soluciones bajo normas y criterios establecidos por ese Ministerio. (14)

J2EE es independiente del sistema operativo, es una plataforma madura y con una amplia gama de proveedores y documentación. Además el usuario no necesitará de requerimientos de hardware ni de software para acceder al sistema, solo se debe equipar con buen hardware el servidor de aplicaciones.

Java es uno de los lenguajes de programación más elaborados y más utilizados para la creación de software de empresa. La evolución de Java, que ha pasado de ser un medio para desarrollar *applets* para ser ejecutados en navegadores, a un modelo de programación capaz de manejar las aplicaciones de una empresa de hoy en día. (15)

Los componentes principales de la plataforma son la Máquina Virtual de Java (JVM) y el lenguaje de programación Java.

### **2.4.2. Lenguaje de Programación Java**

Este lenguaje nace bajo la filosofía de la Programación Orientada a Objetos (OOP), lo que permite lograr una mayor reutilización del código y una forma más eficiente de producir software, es una técnica centrada principalmente en los datos y la manera de llegar a ellos. Fue creado a partir de C++ y diseñado para integrarse sin mucha complejidad a entornos de red, ideal para ambientes bajo filosofía Cliente-Servidor. Es portable, sus programas pueden ser ejecutados en multiplataforma, esto posibilita hacer más liviana la carga de trabajo de un sistema, debido a que puede distribuirse en diferentes equipos de cómputo sin los problemas de incompatibilidad de estructuras de datos, o sea, se pueden utilizar recursos situados en diferentes ubicaciones geográficas, característica que favorece el desarrollo de SIGEPOL dado la modularidad y distribución del mismo como sistema.

Es un lenguaje que es compilado, generando ficheros de clases compilados, pero estas clases compiladas, son en realidad interpretadas por la máquina virtual de java. Siendo la máquina virtual de java la que mantiene el control sobre las clases que se estén ejecutando. Además es seguro ya que al ejecutar el código java la máquina virtual realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros.

### **2.4.3. Lenguaje de Programación Groovy**

Groovy es un lenguaje orientado a objetos para la Plataforma Java, como alternativa al lenguaje de programación Java. Es un lenguaje dinámico, similar a Python, Ruby y Perl que puede usarse como lenguaje de scripting dentro de la Plataforma Java. También utiliza una sintaxis con llaves para delimitar bloques, y se compila dinámicamente hacia bytecodes para la Máquina Virtual Java, funcionando así con cualquier librería y código Java.

El lenguaje Groovy es un superconjunto del lenguaje Java, esto hace que su aprendizaje sea más fácil para programadores Java. En general se puede renombrar un archivo .java en .groovy y va a funcionar, aunque hay algunas pocas incompatibilidades. Además Groovy a diferencia de Java, inyectan métodos y propiedades nuevas en las clases del JDK, o incluso en los tipos primitivos. (16)

Entre las características que lo distingue se incluyen el tipado estático y dinámico, closures, sobrecarga de operadores, sintaxis nativa para la manipulación de listas y maps, soporte nativo para expresiones regulares, iteración polimórfica y expresiones embebidas dentro de strings. (17)

### **2.4.4. Máquina Virtual de Java (JVM)**

El término JVM se refiere a la especificación abstracta de una máquina de software para ejecutar programas Java. Es el núcleo de ese lenguaje de programación, por lo que resulta imposible ejecutar cualquier programa Java sin la ejecución de alguna implantación de la JVM, o sea, el código no se ejecuta directamente sobre un procesador físico, sino sobre un procesador virtual Java. Es la encargada de traducir los bytecode en las instrucciones nativas, permitiendo la portabilidad de las aplicaciones.

## **2.5. Frameworks**

En el desarrollo de software, un marco de trabajo (framework en inglés) es una estructura de soporte definida sobre la cual un proyecto puede ser organizado y desarrollado; puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre aplicaciones para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Para desarrollar una aplicación Web es necesario tener en cuenta que esta filosofía está basada en el estilo arquitectónico Cliente-Servidor, por lo que es necesario definir ambos comportamientos. Por ello se propone, para el lado del cliente, la utilización del siguiente marco de trabajo para el trabajo con javascript.

### **2.5.1. Dojo**

Dojo es un marco de trabajo que contiene APIs y controles (widgets) para facilitar el desarrollo de aplicaciones Web que utilicen el modelo AJAX. Con la utilización de Dojo, se gana en facilidad y agilización del proceso de desarrollo de este tipo de aplicaciones; permite reutilizar código y promueve buenas prácticas de desarrollo; resuelve problemas comunes de compatibilidad entre navegadores. Dojo contiene un sistema de empaquetado inteligente, efectos de Interfaz de Usuario (UI), abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de APIs con AJAX. Además presenta la habilidad de degradar cuando AJAX/Java Script no es completamente soportado en el cliente. (18)

### **2.5.2. Grails**

Para darle soporte a toda la gestión de información y manejo de páginas dinámicas que llegan a los clientes, se plantea para el lado del servidor la utilización del siguiente marco de trabajo, dadas las potencialidades del mismo que se exponen a continuación.

Grails es un framework de desarrollo para aplicaciones Web creado en Java y Groovy. Pero también es una herramienta de construcción de proyectos y pruebas a la vez, que permite añadir plugins desarrollados por terceros. Tiene la peculiaridad de que abarca las tres capas del desarrollo web: acceso a base de datos, capa de negocio y vista. Grails se basa en otros muchos frameworks opensource conocidos, principalmente Spring e Hibernate, proporcionando un entorno de desarrollo estandarizado y ocultando gran parte de los detalles de configuración al programador.

Es un framework de desarrollo “full stack”, esto significa que proporciona todos los componentes necesarios para desarrollar una aplicación. Entre los componentes se encuentran el contenedor de servlets, la base de datos (hsqj por defecto), las diversas capas del MVC, la capa de servicios que incorpora el framework y servicios de logging.

La consecuencia inicial de ser “full stack” y de tener una estructura definida por convención garantiza crear una aplicación y ponerla a funcionar en un corto periodo de tiempo. Se lanza el comando “grails create-app miAplicacion” y unos momentos después tienen una aplicación funcionando mediante el comando “grails run-app”, sin ninguna configuración inicial y sin tener que instalar otros componentes. (19)

La estructura de las aplicaciones Grails es también parte de su convención. Es una estructura definida, siempre igual, donde cada artefacto es lo que es por su ubicación, no por la configuración en algún fichero XML. Los controladores en la carpeta controllers, los objetos de dominio en domains, y así sucesivamente.

Este framework se basa en dos principios fundamentales que aumentan significativamente la productividad:

- **Convención mejor que configuración.** Elimina la configuración a realizar de forma manual por la implantación de convenciones para definir el tipo y las características de los artefactos, en lugar de hacerlo en archivos de configuración.

- **DRY (Don't repeat yourself).** La participación de Spring Container en Grails permite inyección de dependencias, de forma que cada actor en la aplicación debe definirse una única vez, haciéndose visible a todos los demás de forma automática. (20)

### **Alta productividad**

Grails tiene tres características que intentan incrementar su productividad comparándolo con los framework Java tradicionales:

- Inexistencia de configuración XML.
- Entorno de desarrollo preparada para funcionar desde el primer momento.
- Funcionalidad disponible mediante métodos dinámicos. (21)

La principal característica de Grails es que está diseñado para que se programe en Groovy, un lenguaje dinámico con la misma sintaxis que Java, pero que añade muchísimas nuevas funcionalidades, permitiendo programar más con menos código. (22)

## **2.6. Entorno Integrado de Desarrollo (IDE)**

Un entorno de desarrollo integrado (en inglés Integrated Development Environment o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. Las herramientas que normalmente componen un entorno de desarrollo integrado son las siguientes: un editor de texto, un compilador, un intérprete, unas herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, opcionalmente, un sistema de control de versiones. (23)

### **2.6.1. SpringSource Tool Suite (STS)**

SpringSource Tools Suite está basado en Eclipse, pero además incorpora muchas herramientas y asistentes cuyo objetivo es el de facilitar y agilizar el desarrollo de nuestras aplicaciones utilizando tecnologías como Spring Framework, Spring MVC y por supuesto, Grails.

Entre las características más destacadas que STS proporciona se encuentran:



- Soporte para Spring 3.
- Asistentes para la creación de proyectos Spring.
- Herramientas para la gestión de beans.
- Editores gráficos de archivos de configuración de Spring. (24)

### **2.7. Gestor de Base de Datos**

Los sistemas de gestión de bases de datos (en inglés Database Management System, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante. (25)

#### **2.7.1. Gestor de Base de Datos Oracle 10g release 2**

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System). Es una herramienta cliente/servidor para la gestión de bases de datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales. Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas. El software del servidor puede ejecutarse en multitud de sistemas operativos.

El uso de este gestor de base de datos permite el manejo de grandes volúmenes de información lo cual es de gran importancia debido a que el sistema que se propone debe llevar el control de todos los trámites realizados por SIGEPOL.

Otra de las razones por la que se utiliza Oracle es que cuenta con un equipo de soporte, de la misma empresa, el cual dará solución a cualquier problema con la base de datos en un corto período de tiempo, además que incluye un conjunto de herramientas que hacen que sea más fácil la integración con otros sistemas que usen Oracle en caso de que sea necesario.

### **2.8. Conclusiones**

En este capítulo se estudiaron e investigaron las metodologías y se fundamentó la elección de las herramientas y tecnologías que utilizan el equipo de desarrollo para la elaboración de la aplicación.

## Capítulo 3 Diseño del Sistema

### 3.1. Introducción

En este capítulo se describe la propuesta de diseño del Módulo de Consejos Disciplinarios. Se muestra la arquitectura del Sistema de Gestión Policial, se presenta el diagrama de paquetes que se propone para estructurar los elementos del diseño analizados desde dos enfoques, horizontal desde el punto de vista modular, y vertical, entrando en la estructura de capas en las que estará dividido cada módulo. Se describen además los diagramas de clases de cada uno de estos paquetes divididos por capas, y la descripción de las principales clases del diseño.

### 3.2. Patrones de Diseño

Un patrón de diseño es una descripción de clases y objetos, comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. Identifica clases, instancias, roles, colaboraciones y la distribución de responsabilidades. (26)

Entre los que utilizamos en el módulo se encuentran:

#### **Inversión de Control (IoC)**

La inversión de control es un patrón utilizado en Grails, donde las dependencias de un componente no deben gestionarse desde el propio componente, para que contenga sólo la lógica necesaria para hacer su trabajo.

El objetivo de esta técnica es mantener los componentes lo más sencillos que sea posible, incluyendo únicamente código que tenga relación con la lógica de negocio, y dejar fuera todo el código de fontanería. Así la aplicación será más fácil de comprender y mantener. (20)

Cuando se crea un componente de la aplicación, Grails configura Spring para que gestione su ciclo de vida (cuándo se crea, cuántas instancias se mantienen vivas a la vez, cómo se destruyen, etc.) y sus dependencias, (qué otros componentes necesita para realizar su trabajo y cómo conseguirlos). (20)

Este patrón es usado principalmente cuando se crea una instancia de un servicio dentro de un controlador, esto permite que el desarrollador no tenga que preocuparse por gestionar el ciclo de vida de dicho servicio, lo cual ahorra tiempo y disminuye la posibilidad de cometer errores por parte del equipo de desarrollo a la hora de crear instancias de otras clases.

**GOF** (Gang of Four)

### **Instancia única o Solitario (Singleton)**

La intención del patrón es mantener un punto de acceso global a una instancia de una clase garantizando que esta sea única.

Consecuencias:

- Acceso controlado a la única instancia. Puede tener un control estricto sobre cómo y cuándo acceden los clientes a la instancia.
- Espacio de nombres reducido. El patrón Singleton es una mejora sobre las variables globales.
- Permite el refinamiento de operaciones y la representación. Se puede crear una subclase de Singleton.
- Permite un número variable de instancias. El patrón hace que sea fácil cambiar de opinión y permitir más de una instancia de la clase Singleton.
- Más flexible que las operaciones de clase (static en C#, Java). (27)

Este patrón es principalmente usado cuando se declara una instancia de un servicio en un controlador, esta instancia tendrá un acceso total a todas las acciones que puede realizar dicho servicio y además puede ser utilizada en cualquier parte del controlador donde fue declarada. El uso de Singleton permite una mejor organización del código al existir una única instancia por clase.

### **3.3. Patrones de Arquitectura**

Un patrón arquitectónico “representa, en términos de componentes y las relaciones entre ellos, posibles soluciones para incorporar en el diseño, aspectos que mejoran la usabilidad del sistema final”. (27)

Entre los principales patrones que se emplearán en el desarrollo del módulo se encuentran:

#### **Modelo-Vista-Controlador (MVC)**

El patrón de arquitectura conocido como Modelo-Vista-Controlador (MVC), separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario; es decir separa en tres capas diferentes los datos de una aplicación, la interfaz de usuario, y la lógica de control:

El Modelo incorpora la capa del dominio y persistencia, que es la encargada de guardar los datos en un medio persistente (ya sea una base de datos, un archivo de texto, XML, registro, etc.). En el modelo es donde se hace el levantamiento de todos los objetos que el sistema debe de utilizar, es el proveedor de los recursos.

La Vista se encarga de presentar la interfaz al usuario, en sistemas *web*, esto es típicamente HTML, aunque pueden existir otro tipo de vistas. En la vista solo se deben de hacer operaciones simples.

El Controlador es el que escucha los cambios en la vista y se los envía al modelo, el cual le regresa los datos a la vista. La forma más sencilla de implementar este patrón es pensando en capas, como regla, los accesos a la base de datos se hacen en el modelo, la vista y el controlador no deben de saber si se usa o no una base de datos. El controlador es el que decide que vista se debe de imprimir y que información es la que se envía.

Este modelo de arquitectura presenta varias ventajas:

- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.
- La aplicación es más escalable y puede mantenerse mejor.
- El reparto de tareas dentro del equipo de trabajo es más fácil.
- La depuración de la aplicación es más sencilla y, finalmente, puede utilizarse un marco de trabajo (o framework) probado. (27)

Al incorporar el modelo de arquitectura MVC al diseño, las piezas se pueden construir por separado y luego unir las en tiempo de ejecución. Si uno de los Componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas.

### **Tres Capas (Layers)**

El modelo tres capas permite desglosar la aplicación en partes lógicas que favorecen a una mejor organización en el desarrollo de la misma. Este patrón es fácilmente acoplable con el MVC. La arquitectura quedaría separada en lo relacionado con la gestión de interfaz (presentación), negocio y acceso a datos, además de las clases del dominio.

### **Patrones de Aplicaciones Empresariales**

#### **Template View**

La mejor manera de trabajar es componer la página web con contenido dinámico como lo hace una página estática, sino poner en los marcadores que se pueden resolver en las llamadas para obtener información dinámica. Desde la parte estática de la página actúa como una plantilla para la respuesta en particular.

Este permite escribir la presentación en la estructura de la página e incrustar marcadores en ella para indicar dónde incluir el contenido dinámico, facilita mucho el desarrollo al permitir incluir contenidos dinámicos dentro de las vistas, en el caso de SIGEPOL este patrón está presente a la hora de incluir la etiqueta `{nombre variable}` dentro de una vista, se usa principalmente a la hora de trabajar con los select, tablas y otros elementos que muestran información desde la base de datos.

### **Front Controller**

Con el uso de este patrón, se garantiza que exista un único objeto que recibe todas las peticiones provenientes de varias vistas, acto seguido se encarga de procesar las solicitudes. Esto permite centralizar toda la gestión de las solicitudes en un único objeto, ganando en organización.

En el caso del sistema que se propone este patrón se encuentra en los controladores principales que engloban los procesos de varias vistas como es el ejemplo del controlador `CandidatoController` que se encarga de recibir todas las solicitudes de las interacciones del usuario referentes a la gestión de los candidatos que se registran en los consejos disciplinarios.

### **Data Transfer Object**

En varias ocasiones desde una vista se envían objetos que pertenecen a varias clases del dominio en estos casos se crea una clase que se encarga de capturar estos objetos y enviarlos de manera independiente a la vista.

En el caso del sistema que se describe el patrón `Data Transfer Object` está presente a la hora de crear los `command` ya que estos son los encargados de capturar los objetos cuando no existe una entidad que los pueda tratar, la función principal del `command` es poder trabajar con varias clases del dominio al mismo tiempo de forma sencilla y rápida.

### 3.4. Descripción de la Arquitectura

Establecer las bases sólidas que soporten la construcción, escalabilidad, correcto funcionamiento y mantenimiento de una aplicación informática, requiere un estudio detallado de los patrones a lo largo del desarrollo del software, las buenas prácticas de programación, el entorno de desarrollo de la misma, las prestaciones que ha de satisfacer y los requerimientos necesarios para su puesta en marcha. La descripción de la arquitectura define los principales aspectos tomados en cuenta en el diseño de un sistema. La arquitectura de SIGEPOL está organizada desde dos enfoques, uno vertical y otro horizontal.

A continuación se muestra la siguiente figura donde se representan los componentes más importantes de la arquitectura del sistema, posteriormente se explican cada una de sus partes.

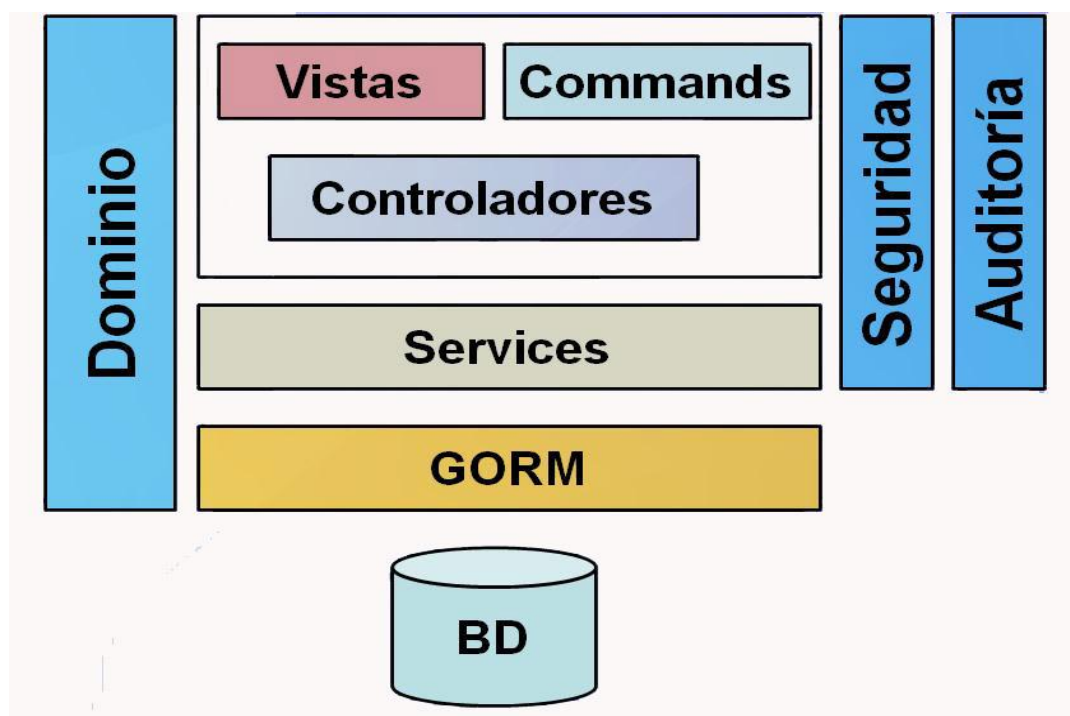


Fig. 1. Modelo de la Arquitectura.

#### DOMINIO

Se encuentra representado verticalmente teniendo en cuenta que contiene todas las vistas, controladores, command y servicios donde están los procesos que ocurren en los Consejos Disciplinarios.

- **Vistas:** en una aplicación del MVC, es responsable de mostrar al usuario el estado del sistema y las acciones que tiene a su disposición. Estas no son responsables de

modificar los datos o incluso de obtener los datos; estas simplemente sirven para mostrar los datos del modelo que han sido suministrados por un controlador.

- **Controladores:** Contendrá los componentes que reciben las órdenes del usuario, gestionan la aplicación de la lógica de negocio sobre el modelo de datos, y determinan qué vista debe mostrarse a continuación. Cuando se dice que los componentes de la capa de control "gestionan la aplicación de la lógica de negocio" se refiere a que son responsables de que ésta se aplique. En términos de aplicaciones web, los controladores son por tanto los responsables de interceptar las peticiones HTTP del navegador y generar la respuesta correspondiente, ya sea en el propio controlador o delegando el trabajo en una vista GSP.
- **Command:** responsables de validar la información que llega con un formulario cuando no existe ninguna entidad involucrada en el proceso. Permite el acceso a atributos de diferentes clases, también posibilita insertar en varias tablas en una misma acción.
- **Servicios:** realizan las acciones de lógica de negocio de la aplicación. Un servicio es una clase cuyo nombre sigue el patrón XXXService.groovy y que se aloja en la carpeta grails-app/services. Para usar el servicio, cualquier controlador, librería de etiquetas, etc. puede declarar una variable userService, lo que activará la inyección automática. Son los encargados de manejar el acceso a datos, además de lograr que la lógica de negocio sea reutilizable permitiendo invocar el mismo proceso desde distintos controladores o acciones sin repetir código.

### **SEGURIDAD**

Se encarga de controlar los aspectos relacionados con la seguridad de la aplicación mediante el control de la autenticación de usuario así como de gestionar las políticas de seguridad definidas. Esta capa se desarrolla utilizando el framework Spring Security, donde se filtran las llamadas a los controladores de la aplicación, verificando que el usuario esté autenticado y que posea el rol que le permite hacer uso de las funcionalidades del sistema sobre las cuales tiene permiso de acceso. Se representa en la arquitectura de manera vertical debido a que la seguridad es gestionada en todo el sistema.

### **AUDITORÍA**

Es la responsable de guardar todos los cambios que realiza el usuario en el sistema, es decir guardaría el rol del usuario, el cambio que efectuó, la tabla donde realizó dicho cambio y la hora y fecha del mismo.

### **GORM**

GORM (Grails Object Relational Mapping), un gestor de persistencia escrito en Groovy, para controlar el ciclo de vida de las entidades y proporcionar una serie de métodos que son creados en tiempo de ejecución para cada clase del dominio que facilitan enormemente las búsquedas. (20)

GORM está construido sobre Hibernate, una herramienta de mapeo objeto-relacional, que se encarga de relacionar las clases del dominio con tablas de una base de datos, y las propiedades de las clases con campos en las tablas, además garantiza que cada operación que sea realizada sobre los objetos del modelo de datos será traducida por Hibernate a las sentencias SQL necesarias para quedar reflejado en la base de datos. GORM proporciona los métodos de búsqueda y modificación que permiten manipular las clases del dominio, además agrega un campo id a la clase y se encargara de generar un valor único para cada instancia de la clase logrando una mayor sencillez en la búsqueda de datos específicos. (20)

Permite la definición de validaciones automáticas sobre estas clases, de manera que antes de persistir en la base de datos un objeto de este tipo, se ejecutan esas validaciones que evitan que se salve si no cumple las restricciones definidas, devolviendo la colección de errores generada. Se generan automáticamente los métodos necesarios para gestionar la persistencia del objeto y asimismo por defecto aplica la técnica del bloqueo optimista para gestionar esta problemática.

### **BASE DE DATOS**

Contiene todas las estructuras necesarias para poder almacenar la información del dominio que debe persistir y está contenida físicamente en el servidor de base de datos. Es única y común a todos los módulos de SIGEPOL y debe garantizar que el modelo de datos esté al menos, en tercera forma normal.

## **3.5. Diagrama de Paquetes**

Los diagramas de paquetes se usan para reflejar la organización de paquetes y sus elementos, muestran como un sistema está dividido en agrupaciones lógicas mostrando las



dependencias entre esas agrupaciones. Un paquete de diseño es una colección de clases, relaciones, realizaciones de casos de usos, diagramas y otros paquetes. Es usado para estructurar el modelo de diseño mediante su división en partes más pequeñas y agrupar elementos relacionados de dicho modelo con propósitos organizacionales. (28)

Para el diseño de este módulo las clases se agruparon por paquetes según su funcionalidad y las entidades que gestionan. Los paquetes definidos son: paquete Común, paquete Control de las Listas Nacional y Regional, paquete Control de los Consejos Disciplinarios y paquete Control de Supervisión de los Consejos Disciplinarios. Cada paquete contiene dos diagramas de clases correspondientes a la Capa de Presentación y a la Capa de Dominio. Se divide en dos diagramas debido a la complejidad de estos y el gran número de clases contenidos en los mismos.

A continuación se describe el objetivo general de cada paquete para lograr una mayor comprensión:

**Paquete Control de las Listas Nacional y Regional:** agrupa todas las funcionalidades referentes a la gestión de los candidatos, la verificación de estos en el SIIPOL, en el Poder Moral y la conformación de las listas nacional y regional. Estas acciones son realizadas por el Organizador del Consejo Disciplinario.

**Paquete Control de Consejos Disciplinarios:** se encuentran las funcionalidades necesarias para la gestión de los consejos disciplinarios, realizadas a los cuerpos de policías. Además de las funcionalidades referentes a la gestión de la propuesta de dicho cuerpo de policía y registrar la pérdida de condición.

**Paquete Control de Supervisión de los Consejos Disciplinarios:** se encuentran todas las funcionalidades necesarias para la creación de guías de supervisión y para gestionar las supervisiones que se le realizan a los consejos disciplinarios. Además de gestionar los consejos disciplinarios ilegales y las propuestas de los cuerpos de policías incompletas.

**Paquete Común:** se representa de un color diferente porque está compuesto por diferentes funcionalidades que son comunes para todos los módulos del sistema.

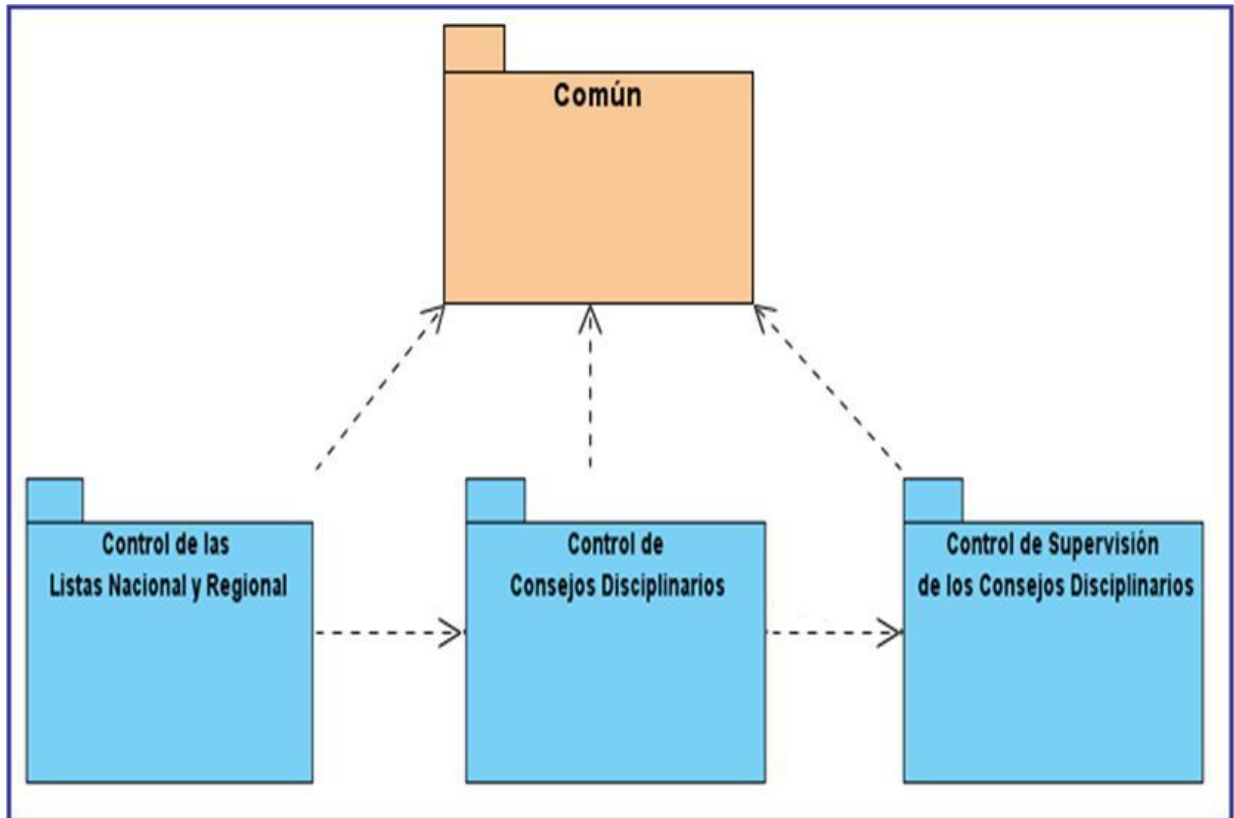


Fig. 2. Diagrama de Paquetes.

### 3.6. Diagrama de clases

#### 3.6.1. Diagramas de Clases de la Capa Presentación

A continuación se muestran los diagramas de clases pertenecientes al Módulo Consejos Disciplinarios, primero se exponen los diagramas pertenecientes a la capa de presentación y luego los que pertenecen a la capa de dominio, una vez expuesto todos los diagramas se procede a la descripción de las principales clases presentes en los mismos.

En la capa de presentación está implementado el patrón Modelo-Vista-Controlador en la cual van a estar ubicadas las vistas, los controladores, los servicios y los command, todas son responsables de mostrar al usuario el estado actual del modelo de datos, y presentarle las distintas acciones disponibles. Se representa el color de las clases como se especifica en la figura de la descripción de la arquitectura del sistema.

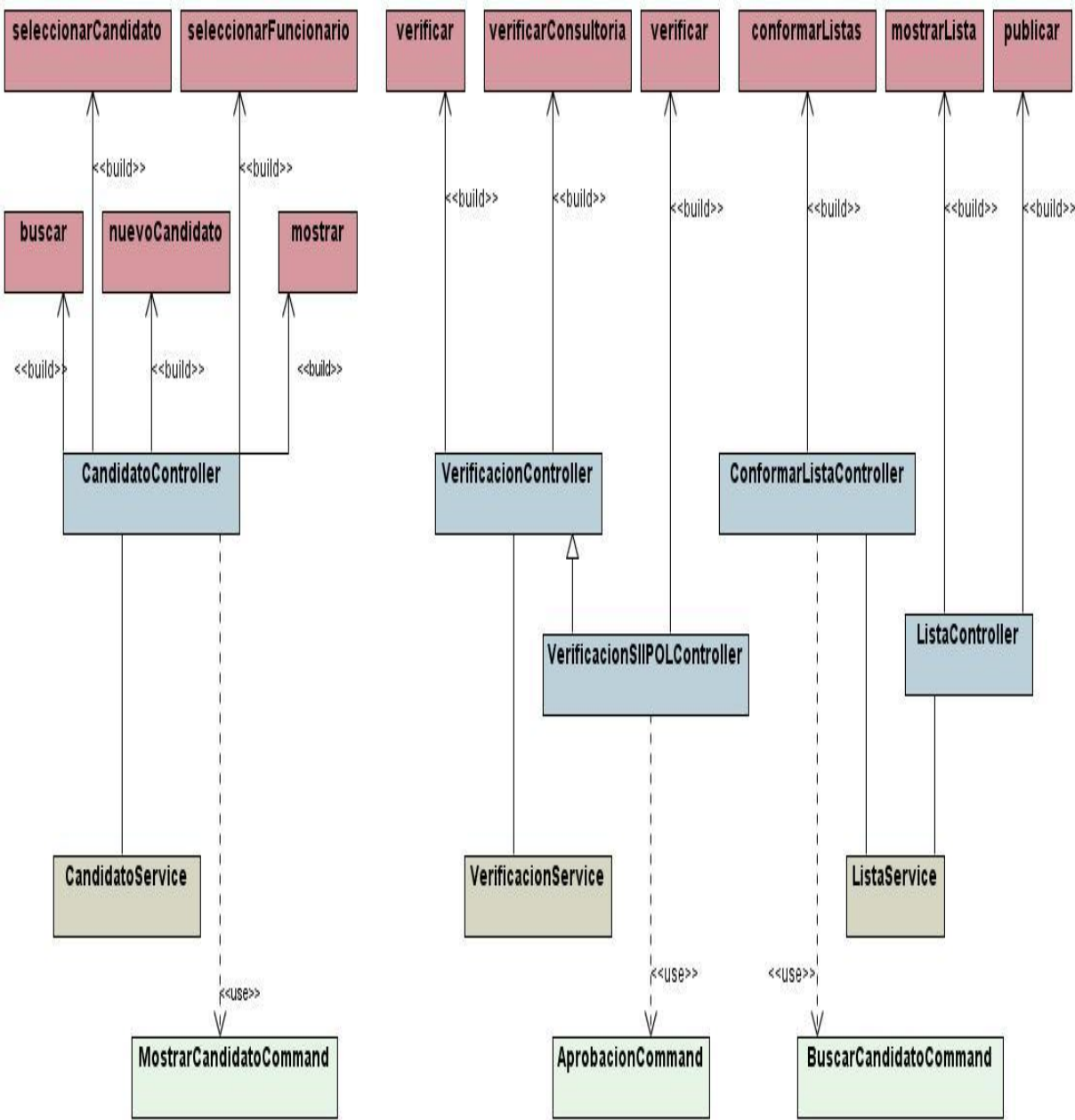


Fig. 3. Diagrama de Clases del Paquete Control de Listas Nacional y Regional(a).

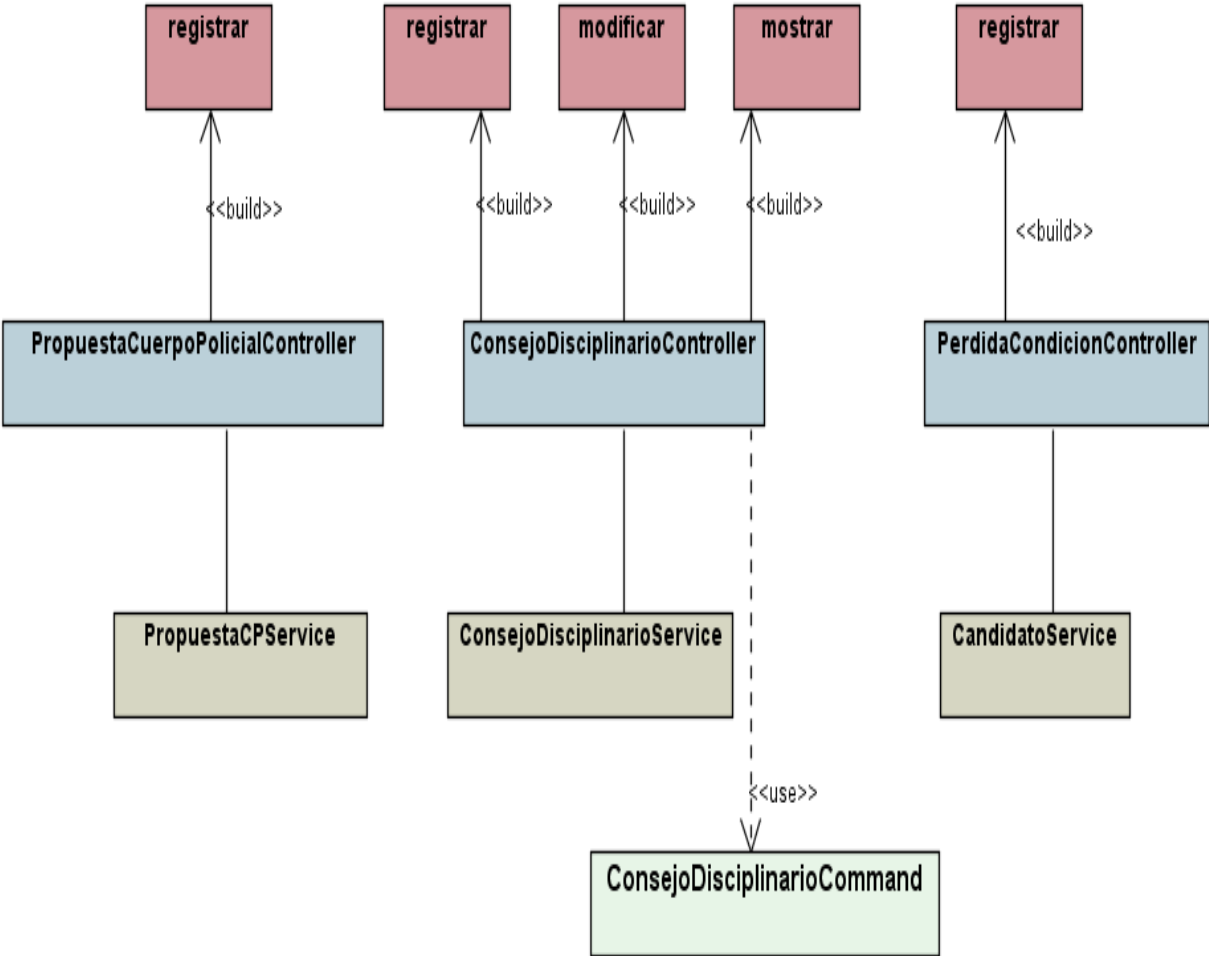


Fig. 4. Diagrama de Clases del Paquete Control de Consejos Disciplinarios(a).

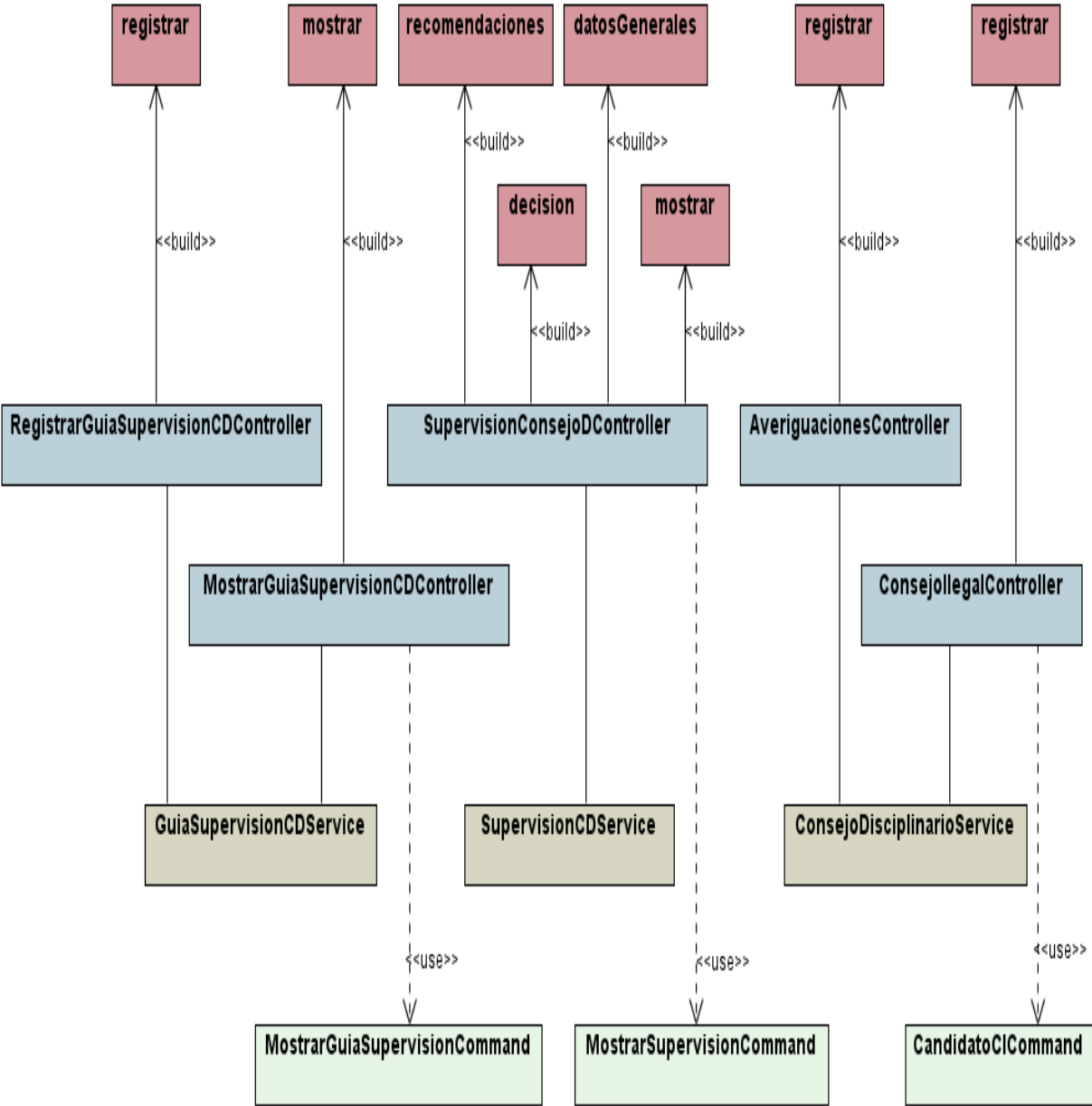


Fig. 5. Diagrama de Clases del Paquete Control de Supervisión de los Consejos Disciplinarios(a).

### 3.6.2. Diagramas de Clases de la Capa de Dominio

En los diagramas de la capa de dominio las clases del módulo se encuentran representadas del color azul como se especifica en la fig.1.Modelo de la Arquitectura. y las clases que se encuentran en naranja son clases externas al módulo que utilizamos.

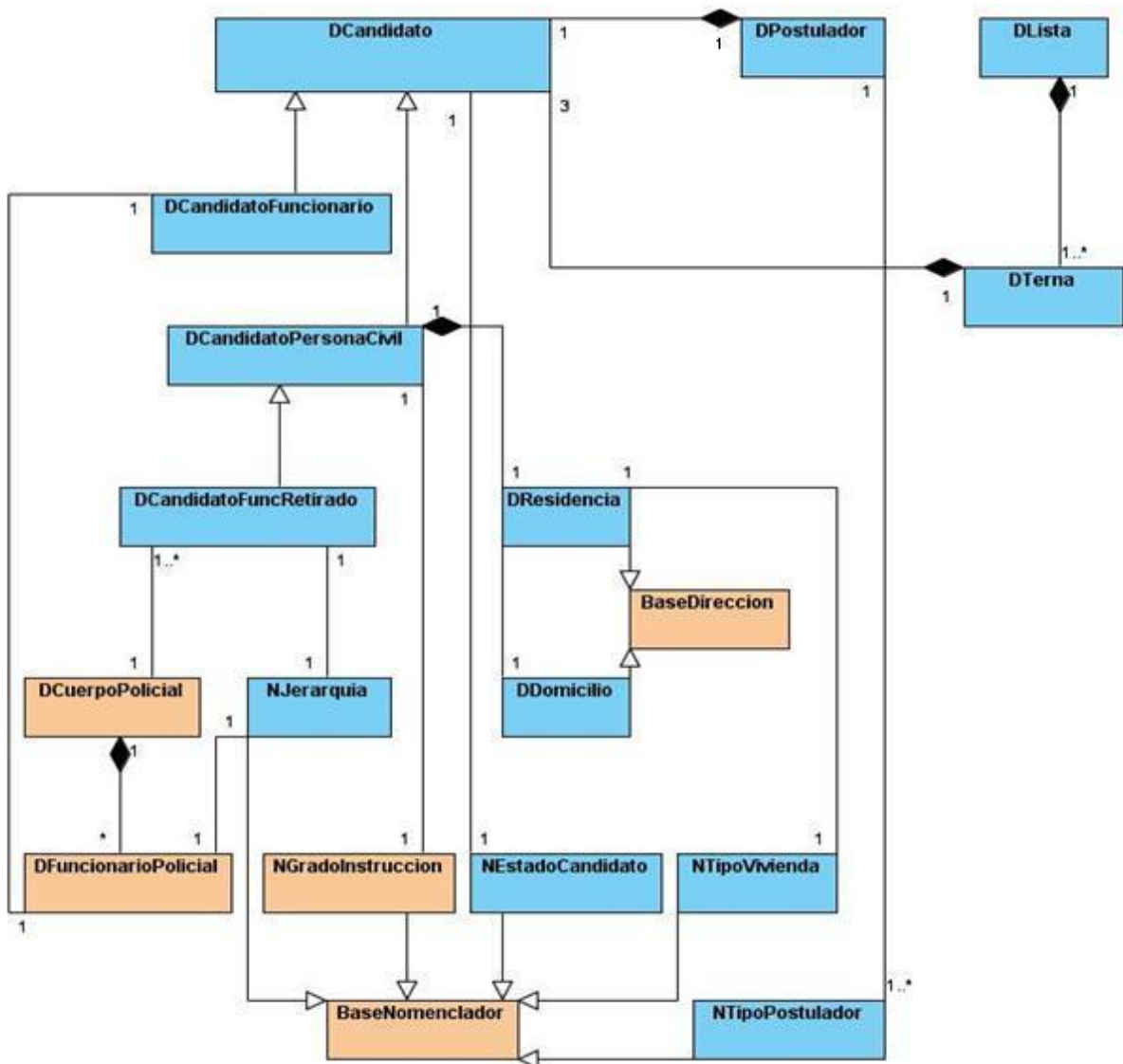


Fig. 6. Diagrama de Clases del Paquete Control de Listas Nacional y Regional (b).

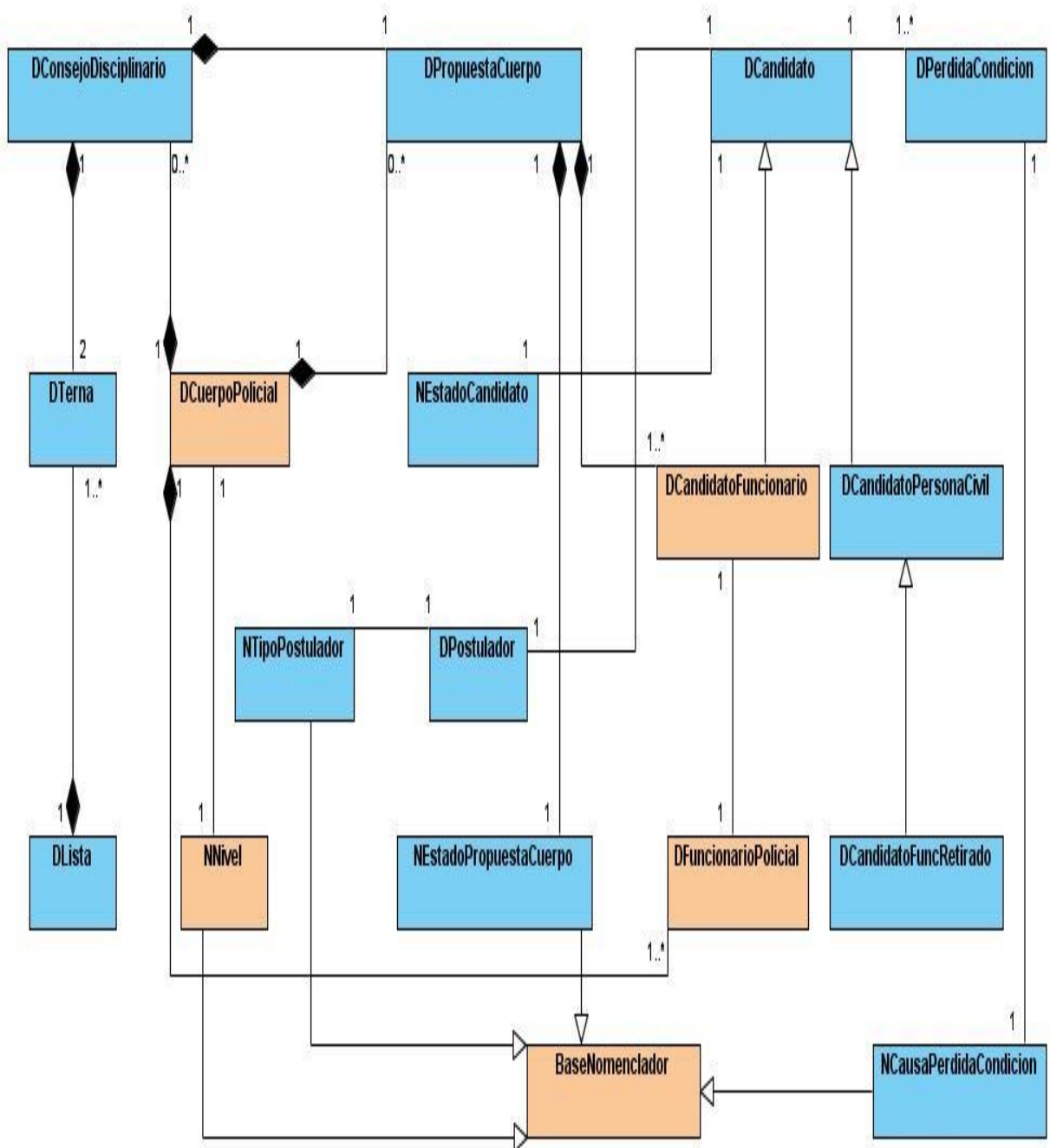


Fig. 7. Diagrama de Clases del Paquete Control de Consejos Disciplinarios (b).

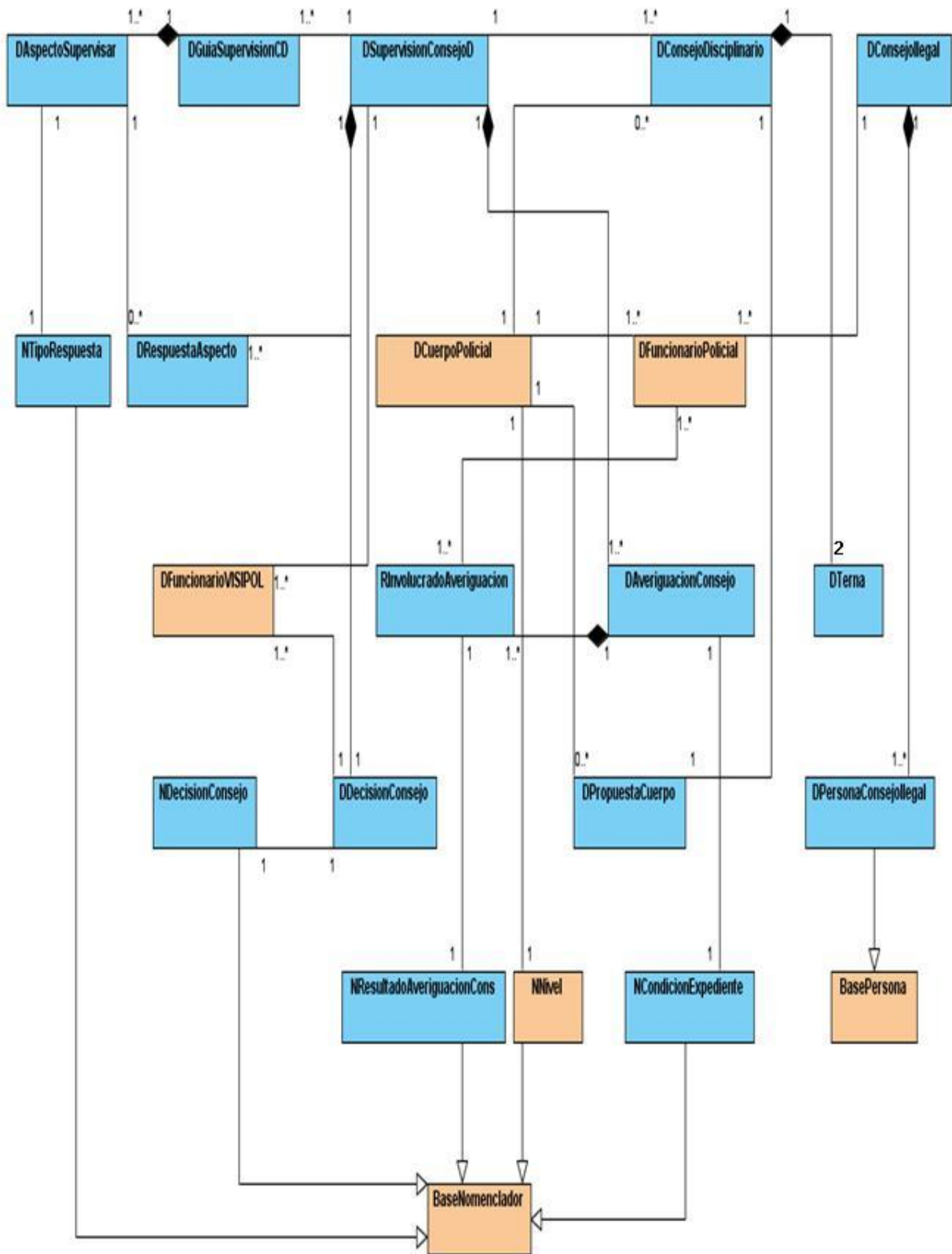


Fig. 8. Diagrama de Clases del Paquete Control de Supervisión de los Consejos Disciplinarios (b).



### 3.7. Descripción de las principales clases

Las siguientes clases descritas a continuación son las principales clases del módulo de Consejo Disciplinario, pues como se puede apreciar en los diagramas pertenecientes a la Capa de Presentación y a la Capa de Dominio, existen un gran número de clases involucradas en el proceso.

#### Clase RegistrarCandidato

##### Propósito

Es una Vista, del paquete Control de Listas Nacional y Regional encargada de mostrar un formulario con los datos que se deben insertar de un candidato.

##### Descripción

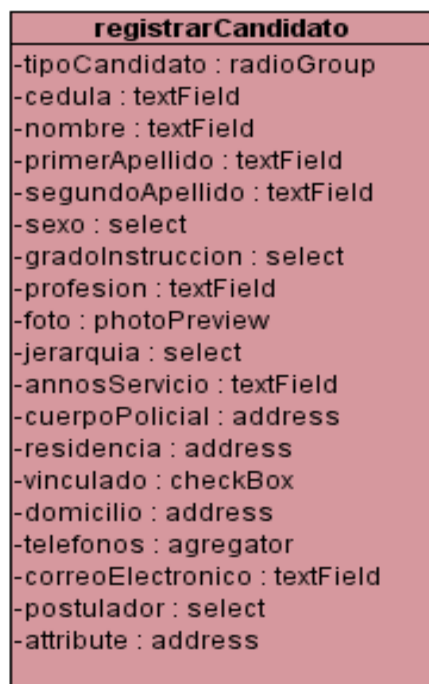


Fig. 9. RegistrarCandidato

#### Clase RegistrarCandidatoController

##### Propósito

Es un controlador, del paquete Control de Listas Nacional y Regional encargada de registrar los datos de los candidatos y decidir las vistas que se deben mostrar referentes a la gestión de los candidatos.

### Descripción

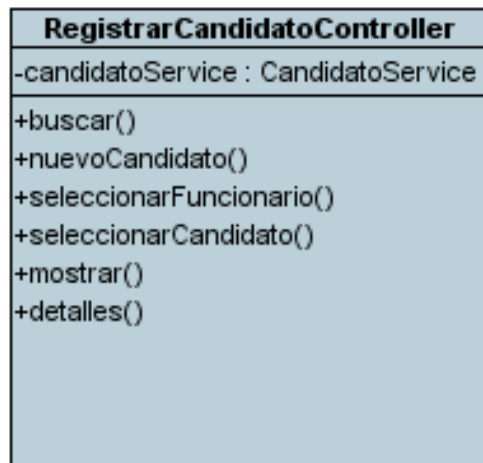


Fig. 10. RegistrarCandidatoController

### Clase ConformarListaController

#### Propósito

Es un controlador, del paquete Control de Listas Nacional y Regional encargada de recibir las solicitudes del Organizador del Consejo Disciplinario sobre la lista que desea conformar y la selección de los candidatos, además de decidir la vista que se deben mostrar referentes a la gestión de estas listas.

#### Descripción

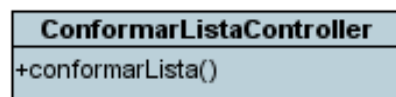


Fig. 11. ConformarListaController

### Clase ListaService

#### Propósito

Es un servicio, del paquete Control de Listas Nacional y Regional encargada de realizar las acciones de lógica de negocio sobre el modelo referente a la gestión de listas.

#### Descripción

| <b>ListaService</b>   |
|---|
| +mostrarCandidatosAprobados(tipoLista : ETipoLista, region : NRegionGeografica, estado : NEstado) : Lista<br>+mostrarListas(tipoLista : ETipoLista) : Lista<br>+registrarTerna(terna : DTerna) : void<br>+registrarLista(lista : DLista) : void |

Fig. 12. ListaService

### **Clase DCandidatoFuncionario**

#### **Propósito**

Es una clase de dominio, del paquete Control de Listas Nacional y Regional que contiene los datos de candidato funcionario.

#### **Descripción**

| <b>DCandidatoFuncionario</b>  |
|---|
| +getCedula()<br>+getNombre()<br>+getPrimerApellido()<br>+getSegundoApellido()<br>+getSexo()<br>+getApellidos()<br>+getGradoInstruccion()<br>+getTipoCandidato() |

Fig. 13. DCandidatoFuncionario

#### **Observaciones**

Hereda de la clase DCandidato.

### **Clase DCandidatoFuncRetirado**

#### **Propósito**

Es una clase de dominio, del paquete Control de Listas Nacional y Regional que contiene los datos de candidato funcionario retirados.

#### **Descripción**

| <b>DCandidatoFuncRetirado</b> |
|-------------------------------|
| -annosServicio : int          |
| +getTipoCandidato()           |

Fig. 14. DCandidatoFuncRetirado

### Observaciones

Hereda de la clase DCandidatoPersonaCivil.

### Clase DCandidatoPersonaCivil

### Propósito

Es una clase de dominio, del paquete Control de Listas Nacional y Regional que contiene los datos de candidato persona civil.

### Descripción

| <b>DCandidatoPersonaCivil</b> |
|-------------------------------|
| -nombre : String              |
| -primerApellido : String      |
| -segundoApellido : String     |
| -cedula : String              |
| -sexo : ESexo                 |
| -correoElectronico : String   |
| -profesion : String           |
| -vinculado : boolean          |
| -foto : byte[]                |
| +getTipoCandidato()           |

Fig. 15. DCandidatoPersonaCivil

### Observaciones

Hereda de la clase DCandidato.

### Clase PropuestaCuerpoPolicialController

### Propósito

Es un controlador, del paquete Control de Consejos Disciplinarios encargada de registrar las propuestas de los cuerpos de policías y decidir las vistas que se deben mostrar referentes a la gestión de estas propuestas.

**Descripción**

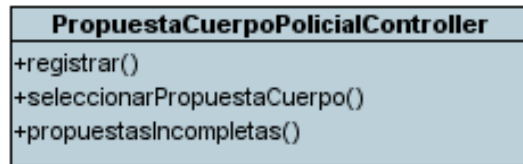


Fig. 16. PropuestaCuerpoPolicialController

**Clase ConsejoDisciplinarioController**

**Propósito**

Es un controlador, del paquete Control de Consejos Disciplinarios encargada de registrar las solicitudes del usuario y decidir las vistas que se deben mostrar referentes a la gestión de los consejos disciplinarios.

**Descripción**

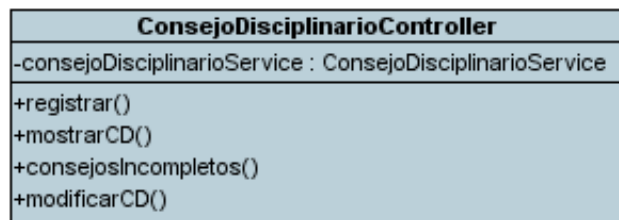


Fig. 17. ConsejoDisciplinarioController

**Clase ConsejoDisciplinarioService**

**Propósito**

Es un servicio, del paquete Control de Consejos Disciplinarios encargada de realizar las acciones de lógica de negocio sobre el modelo referente a la gestión de los consejos disciplinarios.

**Descripción**

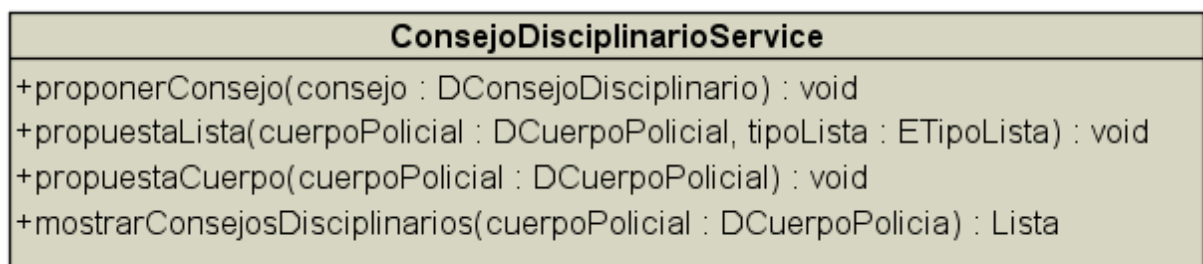


Fig. 18. ConsejoDisciplinarioService

### Clase RegistrarGuíaSupervisiónCDController

#### Propósito

Es un controlador, del paquete Control de Supervisión de los Consejos Disciplinarios encargada de registrar las guías de supervisión a los consejos disciplinarios y decidir las vistas que se deben mostrar referentes a la gestión de estas guías.

#### Descripción

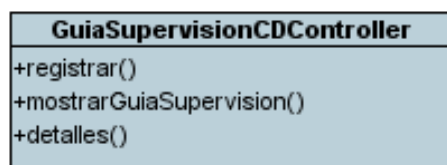


Fig. 19. RegistrarGuíaSupervisiónCDController

### Clase SupervisiónConsejoDisciplinarioController

#### Propósito

Es un controlador, del paquete Control de Supervisión de los Consejos Disciplinarios encargada de registrar las supervisiones a los consejos disciplinarios y decidir las vistas que se deben mostrar referentes a la gestión de las supervisiones.

#### Descripción

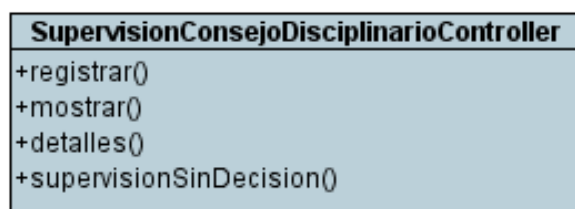


Fig. 20. SupervisiónConsejoDisciplinarioController

### Clase ConsejoIlegalController

#### Propósito

Es un controlador, del paquete Control de Supervisión de los Consejos Disciplinarios encargada de registrar los consejos disciplinarios ilegales y decidir las vistas que se deben mostrar referentes a la gestión de dichos consejos.

## Descripción

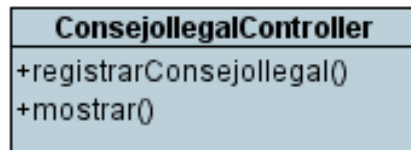


Fig. 21. ConsejoLegalController.

## 3.8. Conclusiones

En este capítulo se realizó una breve descripción de los patrones arquitectónicos de diseños utilizados en la arquitectura así como los principales aspectos tomados en el sistema, se describió el diseño del módulo Consejos Disciplinarios, se presentó el diagrama de paquetes, así como los diagramas de clases de cada uno de los paquetes. Además se realizó una descripción de las clases del diseño.

## Capítulo 4 Implementación de la solución

### 4.1. Introducción

En el presente capítulo se representan en términos de componentes los elementos del modelo del diseño definidos en el capítulo anterior. A continuación se modelan los diagramas de componentes, con los componentes principales que constituyen la aplicación del Módulo de Consejos Disciplinarios. Se realiza una breve descripción de cada componente, especificando su propósito y contenido.

### 4.2. Modelo de Implementación

El modelo de implementación está conformado por el diagrama de componentes y el diagrama de despliegue. Se describen los componentes a construir así como su organización y dependencia entre nodos físicos en los que funcionará el sistema.

#### 4.2.1. Diagrama de Componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema. (29)

Son utilizados para modelar la vista estática y dinámica de un sistema. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema. Seguidamente se muestra el diagrama de componentes correspondiente a la aplicación.

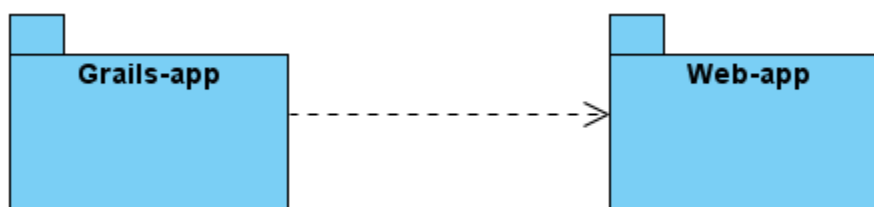


Fig.22. Diagrama de Componentes

#### 4.2.2. Descripción de los Componentes



## Subsistema Grails-app

### Propósito

Contenedor físico de los ficheros que conforman el núcleo de la aplicación.

### Contenido

- Subsistema Controllers
- Subsistema Domain
- Subsistema Services
- Subsistema View

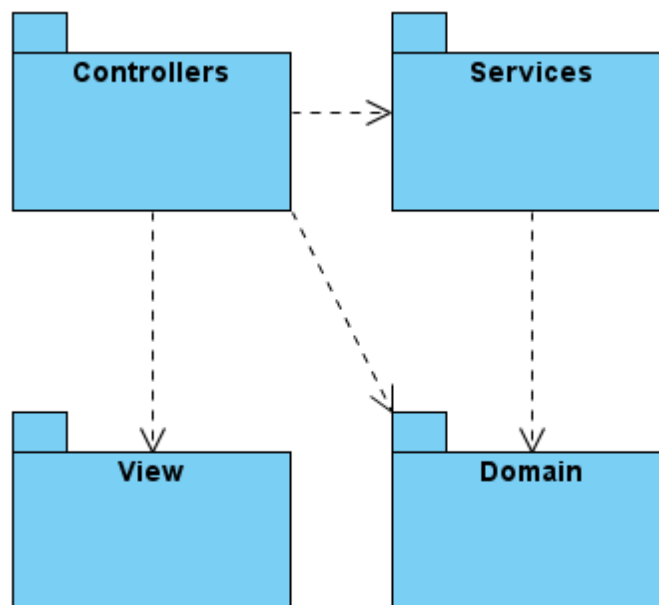


Fig.23. Diagrama de Componentes del subsistema grails-app.

## **Subsistema Controllers**

### Propósito

Contenedor físico de los ficheros que aplican la lógica del negocio y controlan el flujo en la aplicación, archivos con extensión Groovy.

### Contenido

- AveriguacionesController.groovy
- CandidatoController.groovy
- ConsejoDisciplinarioController.groovy
- ConsejoLegalController.groovy
- ConformarListaController.groovy
- ListaController.groovy
- MostrarGuiaSupervisionCDController.groovy
- PerdidaCondicionController.groovy
- PropuestaCuerpoPoliciaController.groovy

- RegistrarGuiaSupervisionCDController.groovy
- SupervisionConsejoDisciplinarioController.groovy
- VerificacionController.groovy
- VerificacionSIIPOLController.groovy
- AprobacionCommand.groovy
- BuscarCandidatoCommand.groovy
- BuscarConsejoCommand.groovy
- CandidatoCICommand.groovy
- CandidatoFuncRetiradoCommand.groovy
- ConsejoDisciplinarioCommand.groovy
- MostrarCandidatoCommand.groovy
- MostrarGuiaSupervisionCommand.groovy
- MostrarSupervisionCommand.groovy

### Subsistema Domain

#### Propósito

Contenedor físico de los ficheros que contienen de manera abstracta la información que el sistema va a manipular, archivos con extensión Groovy.

#### Contenido

- DConsejoDisciplinario.groovy
- DConsejoLegal.groovy
- DLista.groovy
- DPersonaConsejoLegal.groovy
- DPropuestaCuerpo.groovy
- DTerna.groovy
- ECondicion.groovy
- NEstadoPropuestaCuerpo.groovy
- NTipoLista.groovy
- DCandidato.groovy
- DCandidatoFuncionario.groovy
- DCandidatoFuncRetirado.groovy
- DCandidatoPersonaCivil.groovy
- DCandidatura.groovy
- DPerdidaCondicion.groovy
- DPostulador.groovy
- EEstadoCandidato.groovy
- ETipoCandidato.groovy
- NCausaPerdidaCondicion.groovy
- NEstadoCandidato.groovy
- NTipoPostulador.groovy
- DAspectoSupervisar.groovy
- DAveriguacionConsejo.groovy
- DDecisionConsejo.groovy

- DGuiaSupervisionCD.groovy
- DRespuestaAspecto.groovy
- DSupervisionConsejoD.groovy
- NCondicionExpediente.groovy
- NDecisionConsejo.groovy
- NResultadoAveriguacionCons.groovy
- NTipoRespuesta.groovy
- RInvolucradoAveriguacion.groovy
- DCuerpoPolicial.groovy
- NNivel.groovy
- BaseDireccion.groovy
- DDireccion.groovy
- DDomicilio.groovy
- DResidencia.groovy
- NEstado.groovy
- NMunicipio.groovy
- NParroquia.groovy
- NTipoVivienda.groovy
- NRegionGeografica.groovy
- DFuncionarioPolicial.groovy
- NGradoInstruccion.groovy
- NJerarquia.groovy
- NEstadoFuncionario.groovy
- BasePersona.groovy
- ESexo.groovy
- DFuncionarioVISIPOL.groovy

### **Subsistema Services**

#### **Propósito**

Contenedor físico de los ficheros se encargan de manejar la lógica del negocio, archivos con extensión Groovy.

#### **Contenido**

- CandidatoService.groovy
- ConsejoDisciplinarioService.groovy
- GuiaSupervisionCDService.groovy
- ListaService.groovy
- PerdidaCondicionService.groovy
- PropuestaCPService.groovy
- SupervisionCDService.groovy
- VerificacionService.groovy

### **Subsistema Views**

#### **Propósito**

Contenedor físico de las interfaces de usuarios, archivos con extensión GSP.

### **Contenido**

#### Averiguaciones

- registrar.gsp

#### Candidato

- buscar.gsp
- mostrar.gsp
- nuevoCandidato.gsp
- seleccionarCandidato.gsp
- seleccionarFuncionario.gsp
- \_mostrar.gsp

#### ConformarLista

- buscarCandidatosAprobados.gsp
- \_confirmarTernas.gsp
- \_seleccionarCandidato.gsp

#### ConsejoDisciplinario

- mostrarCD.gsp
- registrar.gsp

#### Consejo Ilegal

- registrar.gsp

#### Lista

- mostrar.gsp
- \_listaNacional.gsp
- \_listaRegional.gsp
- \_mostrarListaNacional.gsp
- \_mostrarListaRegional.gsp
- \_publicar.gsp

#### GuiaSupervision

- registrar.gsp
- mostrar.gsp

#### PerdidaCondicion

- registrar.gsp

#### PropuestaCuerpoPolicial

- registrar.gsp

### SupervisionConsejoDisciplinario

- datosGenerales.gsp
- recomendaciones.gsp
- \_aspectosSupervisados.gsp
- \_consejoSupervisado.gsp
- \_guiasSupervision.gsp
- \_supervisores.gsp
- mostrar.gsp
- decision.gsp

### Verificacion

- verificar.gsp
- \_verificar.gsp
- verificarConsultoriaJ.gsp
- \_verificarConsultoriaJ.gsp
- verificarSIIPOL.gsp

### **Subsistema Web-app**

#### **Propósito**

Contenedor físico de los ficheros usados para el diseño de la página, archivos con extensión CSS, JS, HTML, JPG, XML.

#### **Contenido**

- **Subsistema JS**
- **Subsistema imagen**
- **Subsistema css**
- **Subsistema web-inf**

#### **Subsistema JS**

##### **Propósito**

Contenedor físico de los ficheros que permiten el acceso a los objetos de las interfaces, archivos con extensión JS.

##### **Contenido**

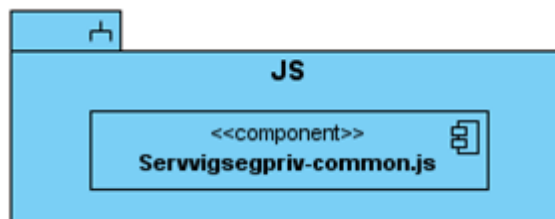


Fig.23. Diagrama de Componentes del subsistema JS.

### 4.3.Descripción de los casos de pruebas

La prueba es un proceso de ejecución de un programa con la intención de descubrir errores que puedan atentar contra el correcto funcionamiento del sistema. Cada producto a entregar debe ser probado para así poder mitigar los errores que surjan y alcanzar un alto nivel de aceptación por parte del cliente.

#### Métodos de Prueba

**Pruebas de caja blanca:** Se comprueban los caminos lógicos del software. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado (sobre el código).

**Pruebas de caja negra:** Pruebas que se llevan a cabo sobre la interfaz del software. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto y que la integridad de la información externa se mantiene (no se ve el código). Permiten encontrar:

- Funciones incorrecta o ausente.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para alcanzar un mejor entendimiento de las pruebas de caja negra asociada al módulo se especifican dos de los escenarios principales que son: las conformaciones de las listas y la creación de los consejos disciplinarios.

#### 4.3.1. SC Registrar Conformación de listas

| Escenario  | Descripción   | Postulado para | Titular | Suplente 1 | Suplente 2 | Condición | Respuesta del sistema   | Flujo central  |
|--|---|----------------|---------|------------|------------|-----------|---|--|
| EC 1.1 Registrar exitosamente Conformación de Listas | Registra exitosamente los datos correspondientes a la Conformación de Listas. | V              | V       | V          | V          | V         | Muestra el listado con los integrantes que acaban de ser insertado al tipo de lista seleccionada. | 1. El Organizador del Consejo Disciplinario selecciona la opción "Conformar Listas".<br>2. Muestra una interfaz solicitando los criterios de búsqueda<br>3. Comprueba que ha introducido los datos obligatorios.<br>4. Muestra el listado de candidatos aprobados obtenidos. |

|  |  |    |    |    |    |    |  |   |
|--|--|----|----|----|----|----|--|---|
|  |  |    |    |    |    |    |  | <p>5. El Organizador del Consejo Disciplinario selecciona un candidato y la condición.<br/>Nota: Para introducir el titular y los dos suplentes a la lista seleccionada se tiene que repetir el proceso de seleccionar la condición del candidato tres veces, indicando un solo titular y dos suplentes.</p> <p>6. Muestra los datos del candidato en el lugar predeterminado en la pantalla según la condición seleccionada.</p> <p>7. Muestra un mensaje de confirmación de inserción de los tres candidatos como integrantes del tipo de lista seleccionada</p> <p>8. Almacena en la BD los tres candidatos como integrantes del tipo de lista seleccionada.</p> <p>9. Muestra el listado con los integrantes que acaban de ser insertado al tipo de lista seleccionada.</p> |
| EC1.2<br>Datos<br>Obligatorios<br>Incompletos        | No se registran los datos de Conformación de Listas porque comprueba que no se han introducido todos los datos obligatorios. | NA | V  | V  | V  | V  | Muestra un mensaje indicando que debe introducir todos los datos obligatorios. | <p>1. El Organizador del Consejo Disciplinario selecciona la opción "Conformar Listas".</p> <p>2. Muestra una interfaz solicitando los criterios de búsqueda.</p> <p>3. Comprueba que no se han introducido los datos obligatorios.</p> <p>4. Muestra un mensaje indicando que debe introducir todos los datos obligatorios.</p>  |
|  |  | V  | NA | V  | V  | V  |  |   |
|  |  | V  | V  | NA | V  | V  |  |   |
|  |  | V  | V  | V  | NA | V  |  |   |
|  |  | V  | V  | V  | V  | NA |  |   |
| EC1.3 No<br>Existen<br>Candidatos<br>Aprobados       | No se registran los datos de Conformación de Listas porque comprueba que el listado de candidatos aprobados es vacío.        | V  | I  | I  | I  | V  | Muestra un mensaje indicando que la búsqueda no produjo resultado.             | <p>1. El Organizador del Consejo Disciplinario selecciona la opción "Conformar Listas".</p> <p>2. Muestra una interfaz solicitando los criterios de búsqueda</p> <p>3. Comprueba que ha introducido los datos obligatorios.</p> <p>4. Obtiene de la BD el listado de candidatos aprobados que cumplan con los criterios de búsqueda introducidos.</p> <p>5. Comprueba que el listado de candidatos aprobados es vacío.</p> <p>6. Muestra un mensaje indicando que la búsqueda no produjo resultado.</p>   |
| EC1.4<br>Selección<br>de<br>Candidatos<br>Incompleta | No se registran los datos de Conformación de Listas  | V  | NA | V  | V  | V  | Muestra un mensaje indicando que debe seleccionar tres candidatos              | <p>1. El Organizador del Consejo Disciplinario selecciona la opción "Conformar Listas".</p> <p>2. Muestra una interfaz solicitando los criterios de</p>   |
|  |  | V  | V  | NA | V  | V  |  |   |

|   |   |   |   |   |    |   |  |   |
|---|---|---|---|---|----|---|--|---|
|   | porque comprueba que no se han seleccionado a los tres candidatos requeridos como titular y suplentes.  | V | V | V | NA | V | para insertarlos como integrantes del tipo de lista seleccionada.  | búsqueda<br>3. Obtiene de la BD el listado de candidatos aprobados que cumplan con los criterios de búsqueda introducidos.<br>4. El Organizador del Consejo Disciplinario selecciona un candidato del listado y la condición.<br>5. Comprueba que no se han seleccionado a los tres candidatos requeridos como titular y suplentes.<br>6. Muestra un mensaje indicando que debe seleccionar tres candidatos para insertarlos como integrantes del tipo de lista seleccionada.   |
| EC1.5<br>Selección de Candidatos Incorrecta | No se registran los datos de Conformación de Listas porque comprueba que no se ha seleccionado a un candidato en condición de titular y dos candidatos en condición de suplentes. | V | I | V | V  | V | Muestra un mensaje indicando que debe seleccionar un candidato en condición de titular y dos candidatos en condición de suplentes. | 1. El Organizador del Consejo Disciplinario selecciona la opción "Conformar Listas".<br>2. Muestra una interfaz solicitando los criterios de búsqueda.<br>3. Comprueba que no se ha seleccionado a un candidato en condición de titular y dos candidatos en condición de suplentes.<br>4. Muestra un mensaje indicando que debe seleccionar un candidato en condición de titular y dos candidatos en condición de suplentes.  |
|   |   | V | V | I | V  | V |  |   |
|   |   | V | V | V | I  | V |  |   |
| EC1.7<br>Opción "Eliminar"                  | El Organizador del Consejo Disciplinario elimina tres integrantes de una misma fila del listado.  | V | V | V | V  | V | Elimina los tres integrantes seleccionados de una misma fila del listado.  | 1. El Organizador del Consejo Disciplinario selecciona la opción "Conformar Listas".<br>2. Muestra una interfaz solicitando los criterios de búsqueda<br>3. Comprueba que ha introducido los datos obligatorios.<br>4. Obtiene de la BD el listado de candidatos aprobados que cumplan con los criterios de búsqueda introducidos.<br>5. El Organizador del Consejo Disciplinario selecciona un candidato del listado y su condición.<br>6. Para introducir el titular y los dos suplentes a la lista seleccionada se tiene que repetir el proceso de seleccionar la condición del candidato tres veces, indicando un solo titular y dos suplentes.<br>7. Muestra un mensaje de confirmación de inserción de los tres candidatos como integrantes del tipo de lista |



|                       |   |   |   |   |   |   |  |
|-----------------------|---|---|---|---|---|---|--|
|                       |   |   |   |   |   |   | <p>seleccionada</p> <p>8. Almacena en la BD los tres candidatos como integrantes del tipo de lista seleccionada.</p> <p>9. Muestra el listado con los integrantes que acaban de ser insertado al tipo de lista seleccionada con la opción de Eliminar y con hipervínculos a mostrar la lista nacional y mostrar la lista regional.</p> <p>10. El Organizador del Consejo Disciplinario selecciona la opción "Eliminar".</p> <p>11. El Organizador del Consejo Disciplinario selecciona la opción "Si".</p>   |
| EC 1.8<br>Opción "No" | El Organizador del Consejo Disciplinario selecciona los tres integrantes de una misma fila del listado selecciona la opción eliminar. | V | V | V | V | V | <p>Cancela la operación.</p> <p>1. El Organizador del Consejo Disciplinario selecciona la opción "Conformar Listas".</p> <p>2. Muestra una interfaz solicitando los criterios de búsqueda</p> <p>3. Comprueba que ha introducido los datos obligatorios.</p> <p>4. Obtiene de la BD el listado de candidatos aprobados que cumplan con los criterios de búsqueda introducidos.</p> <p>5. El Organizador del Consejo Disciplinario selecciona un candidato del listado y su condición.</p> <p>6. Para introducir el titular y los dos suplentes a la lista seleccionada se tiene que repetir el proceso de seleccionar la condición del candidato tres veces, indicando un solo titular y dos suplentes.</p> <p>7. Muestra un mensaje de confirmación de inserción de los tres candidatos como integrantes del tipo de lista seleccionada</p> <p>8. Almacena en la BD los tres candidatos como integrantes del tipo de lista seleccionada.</p> <p>9. Muestra el listado con los integrantes que acaban de ser insertado al tipo de lista seleccionada con la opción de Eliminar y con hipervínculos a mostrar la lista nacional y mostrar la lista regional.</p> <p>10. El Organizador del Consejo Disciplinario selecciona la opción "Eliminar".</p> <p>11. El Organizador del Consejo Disciplinario selecciona la opción "No".</p> <p>12. Cancela la operación.</p> |

### 4.3.1. SC Crear Consejo Disciplinario

| Escenario                              | Descripción                                      | # CD | Nivel | Cuerpo Policía | Nacional | Regional | Respuesta del sistema  | Flujo central   |
|--|--|------|-------|----------------|----------|----------|--|---|
| EC 1.1 Registrar Consejo Disciplinario | Crear exitosamente el consejo disciplinario.     | V    | V     | V              | V        | V        | Almacena en la BD el consejo disciplinario creado del cuerpo de policía seleccionado usando el código del cuerpo de policía. | <ol style="list-style-type: none"> <li>1. El Organizador del Consejo Disciplinario selecciona la opción "Crear Consejo Disciplinario".</li> <li>2. Obtiene de la BD los niveles de cuerpo de policía que existen registrados.</li> <li>3. Muestra un formulario con datos a solicitar:</li> <li>4. El Organizador del Consejo Disciplinario introduce los datos necesarios para crear un consejo disciplinario.</li> <li>5. Muestra el listado de propuestas obtenidos para el cuerpo de policía en la interfaz "Seleccionar Propuesta del Cuerpo de Policía".</li> <li>6. El Organizador del Consejo Disciplinario selecciona un elemento del listado mostrado.</li> <li>7. Comprueba que la lista nacional, lista regional y cuerpo de policía están completas.</li> <li>8. Comprueba que el número de consejo disciplinario no exista para el cuerpo de policía seleccionado usando el código del mismo.</li> <li>9. Almacena en la BD el consejo disciplinario creado del cuerpo de policía seleccionado usando el código del cuerpo de policía.</li> </ol> |
| EC 1.2 Dato Obligatorio Vacío.         | Algunos de los campos obligatorios están vacíos. | NA   | V     | V              | V        | V        | Muestra un mensaje indicando que faltan datos por introducir y señala los datos que faltan.                                  | <ol style="list-style-type: none"> <li>1. El Organizador del Consejo Disciplinario selecciona la opción "Crear Consejo Disciplinario".</li> <li>2. Obtiene de la BD los niveles de cuerpo de policía que existen registrados.</li> <li>3. Muestra un formulario con datos a solicitar:</li> <li>4. El Organizador del Consejo Disciplinario introduce los datos necesarios para crear un consejo disciplinario.</li> <li>5. Comprueba que no se ha introducido los datos obligatorios.</li> <li>6. Muestra un mensaje indicando que faltan datos por introducir y señala los datos que faltan.</li> </ol>   |
|  |  | V    | NA    | V              | V        | V        |  |   |
|  |  | V    | V     | NA             | V        | V        |  |   |
|  |  | V    | V     | V              | NA       | V        |  |   |
|  |  | V    | V     | V              | V        | NA       |  |   |
| EC 1.3 Propuesta                       | Comprueba que se haya                            | V    | V     | V              | I        | V        | Muestra un mensaje   | <ol style="list-style-type: none"> <li>1. Comprueba que se haya obtenido titular y suplentes</li> </ol>   |

|   |   |   |   |   |   |   |   |  |
|---|---|---|---|---|---|---|---|--|
| de Listas Incompleta                            | obtenido titular y suplentes para la lista nacional y para la lista regional.                 | V | V | V | V | I | indicando que la propuesta de la lista no está completa.  | para la lista nacional y para la lista regional. 2. Muestra un mensaje indicando que la propuesta de la lista no está completa.  |
|   |   | V | V | V | I | I |   |  |
| EC1.4 Propuesta de Cuerpo de Policía Incompleta | Comprueba que la propuesta de cuerpo de policía no incluye al titular y al menos un suplente. | V | V | I | V | V | Muestra un mensaje indicando que no se obtuvo propuesta por cuerpo de policía.  | 1. Comprueba que la propuesta de cuerpo de policía no incluye al titular y al menos un suplente. 2. Muestra un mensaje indicando que no se obtuvo propuesta por cuerpo de policía.   |
| EC 1.5 Dato Introducido Incorrecto              | Comprueba que los datos introducidos son incorrectos.   | I | V | V | V | V | Muestra un mensaje indicando que existen datos incorrectos, indica los datos que están incorrectos. Los datos se mantienen para que el usuario los modifique si lo desea. | 1. Comprueba que los datos introducidos son incorrectos. 2. Muestra un mensaje indicando que existen datos incorrectos, indica los datos que están incorrectos. Los datos se mantienen para que el usuario los modifique si lo desea.                              |
|   |   | V | I | V | V | V |   |  |
|   |   | V | V | I | V | V |   |  |
|   |   | V | V | V | I | V |   |  |
|   |   | V | V | V | V | I |   |  |
| EC 1.6 Existe Número de Consejo                 | Comprueba que el número de consejo disciplinario.   | I | V | V | V | V | Muestra un mensaje indicando que el número de consejo disciplinario introducido ya existe.  | 1. Comprueba que el número de consejo disciplinario existe para el cuerpo de policía seleccionado usando el código del mismo. 2. Muestra un mensaje indicando que el número de consejo disciplinario introducido ya existe para el cuerpo de policía seleccionado. |
|   |   |   |   |   |   |   |   |  |

#### 4.4. Conclusiones

En este capítulo se representa la estructura física del sistema donde se identifican componentes y subsistemas de implementación además de especificar el propósito y contenido de cada uno y las relaciones que existen entre ellos.

### **5. Conclusiones Generales**

Al concluir el presente trabajo de diploma se ha cumplido el objetivo y las tareas de la investigación. Se cumplió el objetivo general trazado de desarrollar una aplicación informática para los procesos de gestión de los candidatos y creación y supervisión de los consejos disciplinarios como parte del Sistema de Gestión Policial para la República Bolivariana de Venezuela.

Con el estudio de las metodologías y las herramientas se definió un diseño que modela el sistema y brinda soporte a todos los requisitos funcionales y no funcionales. Un aspecto de gran utilidad en la confección y concepción del diseño fue la comprensión y utilización de patrones, que permitieron aplicar buenas prácticas y brindar soluciones probadas a problemas comunes.

Se estructuró un diagrama de componentes que representa la implementación del sistema en términos de componentes, quedando construida una aplicación donde estará centralizada toda la información resultante de los procesos de investigaciones de los casos evaluados en Consejos Disciplinario.

## **6. Recomendaciones**

Realizar las pruebas a la versión obtenida del Módulo de Consejos Disciplinarios, que permita evaluar y garantizar la calidad del sistema.

Continuar el ciclo de desarrollo del módulo, realizando nuevas iteraciones en cada uno de los procesos, así como implementar la funcionalidad de auditoría del sistema.

Seguir profundizando en el estudio del framework Grails y el lenguaje Groovy para aprovechar las facilidades y beneficios que brinda.

## 7. Referencias Bibliográficas

1. <http://informe21.com>. [Online] 2008. [Cited: enero 10, 2011.]  
<http://informe21.com/actualidad/venezuela-registro-17600-asesinatos-2010-segun-observatorio-venezolano-violencia> .
2. *Gaceta Oficial de la República Bolivariana de Venezuela*. Republica Bolivariana de Venezuela : Deposito Legal, 2010. Número 39.349.
3. **República Bolivariana de Venezuela**. Abrebrecha. [Online] agosto 11, 2008. [Cited: diciembre 15, 2010.] [http://www.abrebrecha.com/82547\\_Consejos-Disciplinarios-de-Polic%C3%ADa-listos-para-activarse-en-todo-el-pa%C3%ADs-.html](http://www.abrebrecha.com/82547_Consejos-Disciplinarios-de-Polic%C3%ADa-listos-para-activarse-en-todo-el-pa%C3%ADs-.html).
4. **Agencia Latinoamericana de Información**. America Latina en Movimiento. [Online] 11 06, 2007. [Cited: enero 13, 2011.] <http://alainet.org/active/20511&lang=es>.
5. **AYESTAS, JOSE ANTONIO PORTUGAL**. Curso de seguridad e inseguridad ciudadana. [Online] 04 13, 2005. [Cited: enero 13, 2011.] <http://www.mailxmail.com/curso-seguridad-inseguridad-ciudadana>.
6. **Zereceda, David Carhuamaca**. Seguridad ciudadana: ¿violencia o paz? [Online] [Cited: enero 15, 2011.] <http://www.uasb.edu.ec/padh/centro/pdfs17/davidcarhuamaca.pdf>.
7. Informe21. [Online] 2008. [Cited: enero 10, 2011.] <http://informe21.com/actualidad/venezuela-registro-17600-asesinatos-2010-segun-observatorio-venezolano-violencia..>
8. YVKE Radio Mundial. [Online] [Cited: enero 20, 2011.]  
<http://www.radiomundial.com.ve/yvke/noticia.php?464620>.
9. **Mendoza, María A**. Informatizate. [Online] julio 2006. [Cited: enero 9, 2011.]  
[Http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html..](Http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html..)
10. Entorno Virtual de Aprendizaje. *Introduccion\_a\_RUP\_y\_UML*. [Online] [Cited: noviembre 10, 2010.]  
[http://eva.uci.cu/mod/resource/view.php?id=34099/Introduccion\\_a\\_RUP\\_y\\_UML.pdf](http://eva.uci.cu/mod/resource/view.php?id=34099/Introduccion_a_RUP_y_UML.pdf) .
11. **Diego, Juan**. Herramientas CASE. [Online] mayo 24, 2008. <http://acquiredactoyo.blogspot.com/>.
12. Visual Paradigm International Ltd. *Visual Paradigm*. [Online] [Cited: febrero 7, 2011.]  
<http://www.visual-paradigm.com/>.
13. Programación con Java. [Online] [Cited: enero 30, 2011.] [http://profesores.fi-b.unam.mx/carlos/java/java\\_basico1\\_2.html](http://profesores.fi-b.unam.mx/carlos/java/java_basico1_2.html).
14. **Legislación Venezolana**. Fundación Software Libre América Latina. [Online] 2008. [Cited: enero 30, 2011.] <http://www.fsfla.org/svnwiki/legis/venezuela>.
15. Allamaraju. *Subrahmanyam*. [Online]
16. Dosideas. [Online] [Cited: febrero 1, 2011.] <http://www.dosideas.com/wiki/Groovy>.

17. **Alzola, Aitor.** Blog de Tecnologías. [Online] [Cited: febrero 3, 2011.] <http://blogs.vitoria-gasteiz.org/ti/tag/grails/page/3/>.
18. elwebmaster. [Online] enero 25, 2008. <http://www.elwebmaster.com/articulos/top-5-javascript-framework..>
19. **Alzola, Aitor.** Blog de Tecnologías. [Online] [Cited: enero 3, 2011.] <http://blogs.vitoria-gasteiz.org/ti/tag/grails/page/3/>.
20. **Brito, Nacho.** Manual de desarrollo web con Grails. España : s.n. [Online] 2009.
21. debugmodeon. [Online] 2008. [Cited: febrero 5, 2011.] <http://es.debugmodeon.com/articulo/introduccion-a-grails>.
22. **Herranz, José Ignacio.** Blog de Paradigma Tecnológico. [Online] [Cited: febrero 4, 2011.] <http://www.paradigmatecnologico.com/se-puede-hacer-una-gran-aplicacion-con-groovy-y-grails/> .
23. Entornos de Desarrollo Integrado. [Online] <http://petra.eutio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
24. SpringSource Tool Suite. *The Best Development Tool for Enterprise Java.* [Online] [Cited: febrero 20, 2011.] <http://www.springsource.com/developer/sts>.
25. **BuenasTareas.com.** Sistema De Gestión De Bases De Datos. [Online] Tecnología. <http://www.buenastareas.com/ensayos/Sistema-De-Gesti%C3%B3n-De-Bases-De/1493930.html>.
26. Patrones de Diseño. [Online] <http://www.google.com/cu/url?sa=t&source=web&cd=1&ved=0CBIQFjAA&url=http%3A%2F%2Fmsdn.microsoft.com%2Fes-es%2Flibrary%2Ffb972240.aspx&rct=j&q=%E2%80%99Los%20patrones%20de%20dise%C3%B1o%20son%20el%20esqueleto%20de%20las%20soluciones%20a%20problemas%20comun>.
27. **Ortiz, Jaime Sánchez.** Patrones de Diseño. [Online] junio 2006. [Cited: febrero 25, 2011.] <http://umlpanamericana.blogspot.com/>.
28. DIAGRAMA DE PAQUETES. *Base de Datos.* [Online] septiembre 2009. <http://maicolsg.blogspot.com/2009/09/diagrama-de-paquetes.html?zx=df8e9e6c2a81bc1b>.
29. Ingeniería del Software. *Modelo de Implementación: Diagramas de Componentes.* [Online] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.
30. **República Bolivariana de Venezuela.** Abrebrecha. [Online] agosto 11, 2008. [Cited: diciembre 15, 2010.] [http://www.abrebrecha.com/82547\\_Consejos-Disciplinarios-de-Polic%C3%ADa-listos-para-activarse-en-todo-el-pa%C3%ADs-.html](http://www.abrebrecha.com/82547_Consejos-Disciplinarios-de-Polic%C3%ADa-listos-para-activarse-en-todo-el-pa%C3%ADs-.html) .

## 8. Bibliografía

**Martínez Aguirre, Ing. Raúl. 2009.** Módulo de Dependencia Policial. Universidad de las Ciencias Informáticas: Tesis, 2009.

**Labrada Arce, Ing. Carlos Rafael. 2010.** Integración del Sistema de Mapificación Web y el Sistema de Gestión Policial. Universidad de las Ciencias Informáticas: Tesis, 2009.

**VENEZUELA, G. D. LEY DE COORDINACION DE SEGURIDAD CIUDADANA.** 2001, n° Disponible en: [http://www.mij.gov.ve/ley\\_coord\\_sc.htm](http://www.mij.gov.ve/ley_coord_sc.htm).

**Craig Larman.** UML y Patrones, Introducción al análisis y diseño Orientado a Objetos.

**Manual desarrollo web con grails.1.1.X** Disponible en: <http://www.manual-de-grails.es>

**Ley del estatuto de la Función Policial.**

<http://informe21.com>. [Online] 2008. [Cited: enero 10, 2011.] <http://informe21.com/actualidad/venezuela-registro-17600-asesinatos-2010-segun-observatorio-venezolano-violencia> .

**Gaceta Oficial de la República Bolivariana de Venezuela.** Republica Bolivariana de Venezuela : Deposito Legal, 2010. Número 39.349.

**República Bolivariana de Venezuela.** Abrebrecha. [Online] agosto 11, 2008. [Cited: diciembre 15, 2010.] [http://www.abrebrecha.com/82547\\_Consejos-Disciplinarios-de-Polic%C3%ADa-listos-para-activarse-en-todo-el-pa%C3%ADs-.html](http://www.abrebrecha.com/82547_Consejos-Disciplinarios-de-Polic%C3%ADa-listos-para-activarse-en-todo-el-pa%C3%ADs-.html).

**Agencia Latinoamericana de Información.** America Latina en Movimiento. [Online] 11 06, 2007. [Cited: enero 13, 2011.] <http://alainet.org/active/20511&lang=es>.

**AYESTAS, JOSE ANTONIO PORTUGAL.** Curso de seguridad e inseguridad ciudadana. [Online] 04 13, 2005. [Cited: enero 13, 2011.] <http://www.mailxmail.com/curso-seguridad-inseguridad-ciudadana>.

**Zereceda, David Carhuamaca.** Seguridad ciudadana: ¿violencia o paz? [Online] [Cited: enero 15, 2011.] <http://www.uasb.edu.ec/padh/centro/pdfs17/davidcarhuamaca.pdf>.

**YVKE Radio Mundial.** [Online] [Cited: enero 20, 2011.] <http://www.radiomundial.com.ve/yvke/noticia.php?464620>.

**Mendoza, María A.** Informatizate. [Online] julio 2006. [Cited: enero 9, 2011.] [Http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](Http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).

**Entorno Virtual de Aprendizaje. Introduccion\_a\_RUP\_y\_UML.** [Online] [Cited: noviembre 10, 2010.] [http://eva.uci.cu/mod/resource/view.php?id=34099/Introduccion\\_a\\_RUP\\_y\\_UML.pdf](http://eva.uci.cu/mod/resource/view.php?id=34099/Introduccion_a_RUP_y_UML.pdf) .

**Diego, Juan.** Herramientas CASE. [Online] mayo 24, 2008. <http://aquiredactoyo.blogspot.com/>.



**Visual Paradigm International Ltd.** *Visual Paradigm*. [Online] [Cited: febrero 7, 2011.] <http://www.visual-paradigm.com/>.

**Programación con Java.** [Online] [Cited: enero 30, 2011.] [http://profesores.fi-b.unam.mx/carlos/java/java\\_basico1\\_2.html](http://profesores.fi-b.unam.mx/carlos/java/java_basico1_2.html).

**Legislación Venezolana.** Fundación Software Libre América Latina. [Online] 2008. [Cited: enero 30, 2011.] <http://www.fsfla.org/svnwiki/legis/venezuela>.

**Dosideas.** [Online] [Cited: febrero 1, 2011.] <http://www.dosideas.com/wiki/Groovy>.

**Alzola, Aitor.** Blog de Tecnologías. [Online] [Cited: febrero 3, 2011.] <http://blogs.vitoria-gasteiz.org/ti/tag/grails/page/3/>.

**elwebmaster.** [Online] enero 25, 2008. <http://www.elwebmaster.com/articulos/top-5-javascript-framework..>

**Alzola, Aitor.** Blog de Tecnologías. [Online] [Cited: enero 3, 2011.] <http://blogs.vitoria-gasteiz.org/ti/tag/grails/page/3/>.

**Brito, Nacho.** Manual de desarrollo web con Grails. España : s.n. [Online] 2009.

**debugmodeon.** [Online] 2008. [Cited: febrero 5, 2011.] <http://es.debugmodeon.com/articulo/introduccion-a-grails>.

**Herranz, José Ignacio.** Blog de Paradigma Tecnológico. [Online] [Cited: febrero 4, 2011.] <http://www.paradigmatecnologico.com/se-puede-hacer-una-gran-aplicacion-con-groovy-y-grails/>.

**Entornos de Desarrollo Integrado.** [Online]

<http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.

**SpringSource Tool Suite.** *The Best Development Tool for Enterprise Java*. [Online] [Cited: febrero 20, 2011.] <http://www.springsource.com/developer/sts>.

**BuenasTareas.com.** Sistema De Gestión De Bases De Datos. [Online] Tecnología. <http://www.buenastareas.com/ensayos/Sistema-De-Gesti%C3%B3n-De-Bases-De/1493930.html>.

**Patrones de Diseño.** [Online]

<http://www.google.com.cu/url?sa=t&source=web&cd=1&ved=0CBIQFjAA&url=http%3A%2F%2Fmsdn.microsoft.com%2Fes-es%2Flibrary%2Fbb972240.aspx&rct=j&q=%E2%80%9C%20Los%20patrones%20de%20dise%C3%B1o%20son%20el%20esqueleto%20de%20las%20soluciones%20a%20problemas%20comun>.

**Ortiz, Jaime Sánchez.** Patrones de Diseño. [Online] junio 2006. [Cited: febrero 25, 2011.] <http://umlpanamericana.blogspot.com/>.

**DIAGRAMA DE PAQUETES.** *Base de Datos.* [Online] septiembre 2009. <http://maicolsg.blogspot.com/2009/09/diagrama-de-paquetes.html?zx=df8e9e6c2a81bc1b>.

**Ingeniería del Software. Modelo de Implementación:** *Diagramas de Componentes.* [Online] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>.

**República Bolivariana de Venezuela.** Abrebrecha. [Online] agosto 11, 2008. [Cited: diciembre 15, 2010.] [http://www.abrebrecha.com/82547\\_Consejos-Disciplinarios-de-Polic%C3%ADa-listos-para-activarse-en-todo-el-pa%C3%ADs-.html](http://www.abrebrecha.com/82547_Consejos-Disciplinarios-de-Polic%C3%ADa-listos-para-activarse-en-todo-el-pa%C3%ADs-.html) .

## ANEXO

Ley del estatuto de la Función Policial.

- Capítulo VII Del Control, Supervisión y Participación Ciudadana en el Desempeño Policial.

**Artículo 75.** Son instancias de control de la policía la Oficina de Control de Actuación Policial, la Oficina de Respuesta a las Desviaciones Policiales y el Consejo Disciplinario de Policía.

**Artículo 80.** El Consejo Disciplinario de Policía es un órgano colegiado, objetivo e independiente de apoyo a la Dirección del cuerpo de policía nacional, estatal o municipal, según el caso, encargado de conocer y decidir sobre las infracciones más graves sujetas a sanción de destitución, cometidas por los funcionarios policiales de cada cuerpo de policía nacional, estatal o municipal, según el caso. Las decisiones que tome el consejo Disciplinario de Policía, previa opinión del Director o Directora del cuerpo de policía nacional, estatal o municipal correspondiente, serán vinculantes para estos últimos una vez adoptados.

**Artículo 81.** El Consejo Disciplinario de Policía estará integrado por el funcionario policial de mayor jerarquía, o el que le sigue en jerarquía, de mayor antigüedad, en condición de personal activo, por un funcionario policial con rango no inferior a comisionado agregado de cualquier cuerpo policial del estado o municipio seleccionados de la lista regional de integrantes de los consejos disciplinarios de policía y por una persona seleccionada de la lista nacional de integrantes de los consejos disciplinarios de policía. El Consejo Disciplinario de Policía se constituirá temporalmente para conocer cada caso que le deba ser sometido y aplicará los procedimientos y las reglas previstos en el Capítulo VI de la presente Ley.

La integración, organización y funcionamiento de los consejos disciplinarios de policía tanto del Cuerpo de Policía Nacional Bolivariana, así como en los cuerpos de policía estatales y municipales, se rigen por lo establecido en la presente Ley, sus reglamentos y resoluciones. El Ministerio del Poder Popular con competencia en materia de seguridad ciudadana regulará, mediante resolución, la constitución, organización, funcionamiento y selección de las listas regionales y nacionales de integrantes de los consejos disciplinarios de policía.

**Artículo 82.** El Consejo Disciplinario de Policía tiene las siguientes competencias:

1. Decidir los procedimientos disciplinarios que se sigan a los funcionarios policiales del cuerpo de policía nacional, estatal o municipal, según el caso, en los casos de faltas

sujetas a la sanción de destitución aplicable de conformidad con las leyes que rigen la materia.

2. Mantener informado permanentemente al Director del cuerpo de policía nacional, estatal o municipal, según el caso, del resultado de los procedimientos y, de manera periódica, preparar informes a ser remitidos al Ministerio del Poder Popular con competencia en materia de seguridad ciudadana, sobre las faltas más conocidas y otros elementos de interés que posibiliten evaluar las causas y condiciones que las favorecen.
3. Las demás establecidas en los reglamentos y resoluciones de esta Ley.

**Artículo 96.** La medida de destitución comporta la separación definitiva del cargo del funcionario policial. El Consejo Disciplinario de Policía ponderará, para decidir sobre la medida de destitución, las circunstancias atenuantes y agravantes que concurrieren en cada caso.