

**SOFTWARE PARA EL ANÁLISIS Y CONTROL DE LOS DATOS REPORTADOS
POR LAS CONTROLARÍAS MUNICIPALES A LA CONTROLARÍA
DEPARTAMENTAL DE SANTANDER**

**RONALD FRANCISCO GRANADOS PLATA
JUAN FERNANDO NOGUERA MUÑOZ**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECHANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA**

2012

**SOFTWARE PARA EL ANÁLISIS Y CONTROL DE LOS DATOS REPORTADOS
POR LAS CONTROLARÍAS MUNICIPALES A LA CONTROLARÍA
DEPARTAMENTAL DE SANTANDER**

**RONALD FRANCISCO GRANADOS PLATA
JUAN FERNANDO NOGUERA MUÑOZ**

Trabajo De Investigación para optar al titulo de Ingeniero de Sistemas

Director

**JAIME OCTAVIO ALBARRACÍN FERREIRA
Profesor Ingeniería de Sistemas**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECHANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA**

2012

CONTENIDO

	Pág.
INTRODUCCION	13
1. PRESENTACION	14
1.1 DEFINICION DEL PROBLEMA	14
1.2 OBJETIVOS	15
1.2.1 Objetivo General	15
1.2.2 Objetivos Específicos	15
1.2.2.1 Módulo de registro	15
1.2.2.2 Módulo de comparación por formatos.	16
1.2.2.3 Módulo de comparación por municipio.	16
1.2.2.4 Módulo de contraste:	17
1.3 JUSTIFICACION	18
1.4 IMPACTO Y VIABILIDAD	19
1.4.1 Impacto	19
1.4.2 Viabilidad	20
1.5 ALCANCE	20
2. MARCO TEORICO	21
2.1 CONTROL FISCAL	21
2.2 MODELO VISTA CONTROLADOR (MVC)	24
2.3 JAVA ENTERPRISE EDITION 6.0	25
2.4 JAVASERVER FACES	27
2.4.1 JavaBeans	29
2.4.2 API JavaServlet	29
2.4.2.1 Ciclo de Vida de un JavaServlet.	29

2.4.2.2 Características	30
2.4.3 Managed Bean	31
2.4.4 Persistencia de Java	31
2.4.5 Conectividad Con La Base De Datos Java (JDBC)	32
2.4.6 Servidor GlassFish	32
2.4.7 Ajax	33
2.5 NETBEANS	33
2.6 SERVIDOR WEB	34
2.7 SERVIDOR DE APLICACIONES	34
2.8 SOFTWARE LIBRE	35
2.8.1 Ventajas del Software Libre	35
2.9 JAVA DEVELOPMENT KIT (JDK)	36
2.10 JAVA RUNTIME ENVIRONMENT (JRE)	36
3. BASES DE DATOS	37
3.1 MODELO JERÁRQUICO	38
3.2 MODELO DE RED	39
3.3 MODELO RELACIONAL	39
3.4 MODELO ENTIDAD-RELACIÓN	40
3.4.1 Claves	41
3.4.2 Integridad Referencial	42
3.4.3 Formas Normales	43
3.5 GESTORES DE BASES DE DATOS	45
3.5.1 PostgreSQL	46
4. PLAN DE TRABAJO	47
4.1 ESPECIFICACIÓN Y ANÁLISIS DE REQUERIMIENTOS	49
4.2 DISEÑO	51
4.3 DESARROLLO	52

5.METODOLOGIA DE TRABAJO	53
5.1 FASE INICIAL	53
5.2 ESPECIFICACION DE REQUERIMIENTOS	54
5.3 ANALISIS	56
5.3.1 Diagramas de casos de uso	57
5.4 DISEÑO	65
5.4.1 Modelo entidad – relación	67
5.4.2 Modelo de datos	68
5.5 IMPLEMENTACION	69
5.6 PRUEBAS	79
CONCLUSIONES Y RECOMENDACIONES	81
BIBLIOGRAFIA	82

LISTA DE FIGURAS

	Pág.
Figura 1. Control Fiscal	22
Figura 2. Modelo Vista-Controlador	25
Figura 3. Esquema de Funcionamiento JavaEE 6.0	26
Figura 4. Modelo Jerárquico	38
Figura5. Modelo de Red	39
Figura6. Modelo Relacional	40
Figura7. Modelo Entidad – Relación	42
Figura 8. Caso de uso Administrador Sistema	57
Figura 9. Caso de uso Entidad Pública	57
Figura 10. Caso de uso Funcionario Contraloría	58
Figura11. Modelo Entidad-Relación	67
Figura 12. Modelo de Datos	69
Figura13. Organización Sistema	70
Figura14. Vistas Contraloría	71
Figura15. Vistas Entidades Públicas	72
Figura16. Carpeta contenedora de código Css y Js	73
Figura17. Carpeta contenedora de recursos de la aplicación	74
Figura18. Carpeta contenedora la capa Lógica de Negocio y Capa de Datos	74
Figura19. Controladores Sistema	75
Figura20. Contenedor Capa Lógica de Negocio	76
Figura21. Contenedor Capa de Datos de la Aplicación	76
Figura22. Métodos Lógica de Negocios	77
Figura23. Motor de Persistencia	78
Figura24. Motor de Persistencia	78

LISTA DE TABLAS

	Pág.
Tabla 1. Casos de Uso - Registrar Usuario	58
Tabla 2. Casos de Uso - Validar Usuario	60
Tabla 3. Casos de Uso - Eliminar Usuario	61
Tabla 4. Casos de Uso - Gestionar Vigencias	62
Tabla 5. Casos de Uso - Diligenciar Formatos	62
Tabla 6. Casos de Uso - Editar Información	63
Tabla 7. Casos de Uso - Consultar Información	64

RESUMEN

TITULO: SOFTWARE PARA EL ANÁLISIS Y CONTROL DE LOS DATOS REPORTADOS POR LAS CONTROLARÍAS MUNICIPALES A LA CONTROLARÍA GENERAL DE SANTANDER*

AUTORES: Juan Fernando Noguera Muñoz
Ronald Francisco Granados Plata**

PALABRAS CLAVE: Sistema de Información Web, Control Fiscal, Patrimonio Público, Información, Framework, Modelo Vista Controlador, JavaServer faces, Java.

DESCRIPCIÓN

La Contraloría General de Santander es un órgano de control con autonomía administrativa y presupuestal para administrar sus asuntos, en los términos y en las condiciones establecidas en la Constitución y en las leyes encargados de realizar el control fiscal del patrimonio público que el estado destina para el departamento de Santander.

El control fiscal es el conjunto de actividades realizadas para lograr, mediante herramientas como sistemas y procedimientos, la regularidad y corrección de la administración del patrimonio público.

La contraloría como ente auditor del departamento de Santander no posee herramientas sofisticadas para brindar un apoyo en la toma de decisiones que a diario deben realizarse.

Este proyecto permitirá realizar el reporte de cada uno de los formatos o cuentas que por decreto debe diligenciar cada entidad sujeta de control en el departamento de Santander a través de un sistema web. Permitirá a los auditores empleados de la contraloría vigilar si la gestión fiscal de los entes sujetos de control, servidores públicos y particulares, se ha ejecutado conforme a las normas legales y los principios de economía, eficiencia, eficacia y equidad.

Cada entidad sujeta de control tendrá la obligación de reportar los datos solicitados por la contraloría para realizar su respectiva auditoría a través del desarrollo de este proyecto.

Este sistema de información web estará basado en el framework JavaServer Faces que constituye un marco de trabajo de interfaces de usuario para aplicaciones web basadas en tecnología Java y en el patrón MVC (Modelo Vista Controlador). Este proyecto utiliza herramientas y procesos que permiten que la información fluya de forma más eficiente y eficaz, es decir, que el tráfico de esta sea rápido, pertinente, consistente, etc.

* Proyecto de Grado modalidad Trabajo de Investigación

** Facultad De Ingeniería Fisicomecánicas, Escuela de Ingeniería de Sistemas e Informática,
Director: Ph.D. Jaime Octavio Albarracín

ABSTRACT

TITLE: SOFTWARE FOR ANALYSIS AND CONTROL OF DATA REPORTED BY CONTRALORIAS MUNICIPALES TO THE CONTRALORIA GENERAL OF SANTANDER*

AUTHORS: Juan Fernando Muñoz Noguera**
Ronald Francisco Granados

KEYWORDS: Web Information System, Fiscal Control, Public Property, Information, Framework, Model View Controller, JavaServer Faces, Java.

DESCRIPTION

The Contraloría General de Santander is a supervisory institute with administrative and budgetary autonomy to manage their affairs under the terms and conditions established by the Constitution and the laws responsible for conducting the fiscal control of public property that the state intended for the Santander department.

Fiscal control is the set of activities undertaken to achieve, using tools such as systems and procedures, regularity and correctness of management of public assets.

The Contraloría does not have sophisticated tools to provide support in making daily decisions that must be performed.

This project allows entities report each of the forms or bills requested by the Contraloría through internet. It must be filled by each entity subject to control in the department of Santander.

Allow employees of the Contraloría like auditors determine whether the fiscal management of the entities subject to control, public and private, has been enforced under the laws and the principles of economy, efficiency, effectiveness and equity.

Each entity subject to control will be required to report the data requested by the Contraloría to perform their respective audit through the development of this project.

This information system will be based on JavaServer Faces that provides a framework of user interfaces for web applications based on Java technology and patternMVC (Model View Controller). This project uses tools and processes that allow information to flow more efficiently and effectively, meaning that this traffic is fast, relevant and consistent.

* Project of Grade

** Faculty of Physical-Mechanical Engineering. Systems and Computer Engineering. Director Ph.D. Jaime Octavio Albarracín

INTRODUCCION

El manejo de la información a través del tiempo ha ido mejorando de forma más efectiva y a su vez eficaz de tal forma que tener grandes cantidades de estas ya no es un problema como hace poco podría suceder. Las organizaciones a medida que crecen y evolucionan generan mayor cantidad de información la cual si no tiene el control y la administración suficiente generará una pérdida en el desarrollo de la organización y en consecuencia el fin de ésta.

El desarrollo de las nuevas tecnologías y más específicamente de las TIC's han demostrado ser una solución a las problemáticas en el control, desarrollo e interpretación de la información. Esta solución son un conjunto de procesos, los cuales tiene un fin específico que en conjunto brindan la solución a la problemática planteada en el continuo desempeño de toda organización.

En este proyecto se busca manejar y administrar una gran cantidad de información que provienen de diferentes puntos y se concentran en uno específico. Para ser estudiada se requiere un gasto de recursos humanos y monetarios que se reducirán al implementar esta herramienta, destinada para hacer el mismo trabajo pero de forma más eficaz y efectiva.

1. PRESENTACION

1.1 DEFINICION DEL PROBLEMA

La Contraloría general de Santander es la entidad encargada de verificar que se administren bien las rentas y gastos de las entidades territoriales, es decir, de controlar los presupuestos de todas las oficinas y servicios del Gobierno y de investigar cualquier mal uso de los dineros públicos.

La contraloría general de Santander en el departamento se encarga de auditar a organizaciones como hospitales, alcaldías, instituciones educativas, entidades descentralizadas del departamento, asambleas ya la gobernación de Santander. Éste control se lleva a cabo a través de un sistema Web llamado SIA en el que cada uno de estos municipios reportan alrededor de 44 informes mes a mes los cuales llevan consigo especificando el manejo de los tesoros del departamento. Estos informes son evaluados por la contraloría general de Santander con el objetivo de verificar el destino de estos recursos.

En total son 81 municipios los que conforman el territorio santandereano lo que hace difícil el manejo de 44 informes por cada uno de éstos. Debido a la manera como cada municipio presenta estos datos es complejo realizar un contraste para llegar al análisis necesario.

Realizar un análisis específico y muy detallado de estos datos es una tarea obligatoria para cumplir con los fines propuestos por esta entidad además del desarrollo económico del departamento y de cada uno de los municipios que lo conforman.

1.2 OBJETIVOS

1.2.1 Objetivo General

Desarrollar una aplicación web para analizar y organizar los datos reportados por las contralorías municipales del departamento de Santander.

1.2.2 Objetivos Específicos

1.2.2.1 Módulo de registro. Permitirá a cada entidad pública registrar, obtener y validar los datos de los siguientes formatos:

Formato 1. Catálogo de Cuentas

Formato 2A. Resumen de Caja Menor

Formato 2B. Relación de Gastos de Caja

Formato 3. Cuentas Bancarias

Formato 4. Pólizas de Aseguramiento

Formato 5A. Propiedad, Planta y Equipo – Adquisiciones y Bajas

Formato 5B. Propiedad, Planta y Equipo – Inventario

Formato 6. Ejecución Presupuestal de Ingresos

Formato 7. Ejecución Presupuestal de Gastos

Formato 7A. Relación de pagos

Formato 8A. Modificaciones al Presupuesto de Ingresos

Formato 8B. Modificaciones al Presupuesto de Egresos

Formato 9. Ejecución PAC de la Vigencia

Formato 10. Ejecución Reserva Presupuestal

Formato 11. Ejecución Presupuestal de Cuentas por Pagar

Formato 11A: Vigencias Futuras

Formato 12. Proyectos de Inversión

Formato 12A. Presupuesto Proyectos de Inversión

Formato 13A. Nuevo Formato de Contratación

Formato 13B. Ingresos por Estampillas

1.2.2.2 Módulo de comparación por formatos. Permitirá visualizar gráficamente el formato de una o varias entidades públicas según el que sea seleccionado:

Formato 1. Catálogo de Cuentas

Formato 2A. Resumen de Caja Menor

Formato 2B. Relación de Gastos de Caja

Formato 3. Cuentas Bancarias

Formato 4. Pólizas de Aseguramiento

Formato 5A. Propiedad, Planta y Equipo – Adquisiciones y Bajas

Formato 5B. Propiedad, Planta y Equipo – Inventario

Formato 6. Ejecución Presupuestal de Ingresos

Formato 7. Ejecución Presupuestal de Gastos

Formato 7A. Relación de pagos

Formato 8A. Modificaciones al Presupuesto de Ingresos

Formato 8B. Modificaciones al Presupuesto de Egresos

Formato 9. Ejecución PAC de la Vigencia

Formato 10. Ejecución Reserva Presupuestal

Formato 11. Ejecución Presupuestal de Cuentas por Pagar

Formato 11A: Vigencias Futuras

Formato 12. Proyectos de Inversión

Formato 12A. Presupuesto Proyectos de Inversión

Formato 13A. Nuevo Formato de Contratación

Formato 13B. Ingresos por Estampillas

1.2.2.3 Módulo de comparación por municipio. Permitirá visualizar gráficamente uno o varios formatos de una sola entidad pública según el que se haya seleccionado:

Formato 1. Catálogo de Cuentas

Formato 2A. Resumen de Caja Menor
Formato 2B. Relación de Gastos de Caja
Formato 3. Cuentas Bancarias
Formato 4. Pólizas de Aseguramiento
Formato 5A. Propiedad, Planta y Equipo – Adquisiciones y Bajas
Formato 5B. Propiedad, Planta y Equipo – Inventario
Formato 6. Ejecución Presupuestal de Ingresos
Formato 7. Ejecución Presupuestal de Gastos
Formato 7A. Relación de pagos
Formato 8A. Modificaciones al Presupuesto de Ingresos
Formato 8B. Modificaciones al Presupuesto de Egresos
Formato 9. Ejecución PAC de la Vigencia
Formato 10. Ejecución Reserva Presupuestal
Formato 11. Ejecución Presupuestal de Cuentas por Pagar
Formato 11A: Vigencias Futuras
Formato 12. Proyectos de Inversión
Formato 12A. Presupuesto Proyectos de Inversión
Formato 13A. Nuevo Formato de Contratación
Formato 13B. Ingresos por Estampillas

1.2.2.4 Módulo de contraste: Permitirá relacionar los diferentes formatos concernientes a cada entidad utilizando tablas o diagramas estadísticos (diagramas de torta, diagramas de barras o histogramas).

a) Exportar el formato obtenido a diferentes tipos de documentos como son archivos de pdf y csv.

1.3 JUSTIFICACION

La contraloría al igual que la mayoría de organizaciones que se desempeñan hoy en día en nuestro medio, es una entidad la cual tiene como fin el estudio de una gran cantidad de información. Este estudio se ha llevado a cabo a través de herramientas que aunque hasta el día de hoy han sido de gran ayuda se está convirtiendo en un procedimiento el cual no es el deseado ya que no está prestando la finalidad requerida. Esto se debe a que basados en nuestras capacidades físicas de la época se crearon métodos como solución al manejo del flujo de la información que ellos requieren analizar; hoy en día ya es demasiada información y los métodos tradicionales de uso no están dando resultados.

Las TIC's entendidas como el estudio de la Tecnología de la Información y la Comunicación las cuales manejan elementos y métodos que sirven para el mejoramiento, difusión e interpretación de la información, que es la base de toda organización, día a día juega un papel más importante en la vida de éstas. Este desarrollo se hace a través de diferentes herramientas y procesos que permiten que la información fluya de forma más eficiente y eficaz, es decir que el tráfico de ésta sea rápido, pertinente, consistente, etc. Gastando el mínimo de recursos posibles.

Nuestro proyecto enfatiza un mejoramiento en el análisis de los datos recolectados provenientes de las subcontralorías del departamento de Santander utilizando TIC's como herramienta para resolver este problema, innovando en el manejo de la información que recibirá la contraloría para realizar los debidos análisis que cumplen su función como organización.

1.4 IMPACTO Y VIABILIDAD

1.4.1 Impacto

La tecnología al pasar del tiempo ha ido jugando un papel muy importante en la vida de las organizaciones de tal forma que ha ido influyendo en el crecimiento de estas e irrumpiendo en la mayoría de ámbitos que componen nuestra sociedad tanto económico, tecnológico y social, entre otros.

Tecnológicamente la contraloría a través de este proyecto dará un paso más hacia la evolución e innovación tecnológica que se vive actualmente en nuestra sociedad y que nos exige que constantemente estemos haciendo uso de ella y a la vez estar al día sobre adelantos los cuales sean relevantes para nosotros. Los métodos de estudio que se tienen actualmente no son los más indicados para poder analizar un gran número de informes que se tienen que tener en cuenta y que al pasar del tiempo se vuelven más complicados para entenderlo; el software a desarrollar aportará una herramienta que hace uso de la tecnología e ingeniería para realizar la misma tarea pero de una forma más específica y precisa.

Económicamente el cambio reflejado será de gran cobertura ya que la tarea que realiza el software actualmente se hace, pero gastando gran cantidad de dinero que se va en recurso tanto humano como físico.

Socialmente vemos que éste proyecto nos muestra dos fases una positiva y una negativa. Positivamente obtendremos una mejor forma de trabajar al hacerlo con diferentes herramientas las cuales demandan un adelanto en nuestro conocimiento, pero una vez obtenido nos llevará a hacer tareas de análisis de una manera más rápida y segura, es decir con un rango de error muy bajo.

1.4.2 Viabilidad

Actualmente la Contraloría General de Santander enfrenta un reto ante el cual se ha desenvuelto de la mejor forma posible gracias a sus metodologías tradicionales de trabajo.

Las contralorías municipales de Santander reportan actualmente su información de manera poco precisa, pues son demasiadas hojas de Excel con información pobre en análisis de cuentas. Podemos decir que son datos lo que estas contralorías de Santander reportan, definiendo datos como una expresión mínima de contenido sobre un tema determinado. El desarrollo de este proyecto será el encargado de transformar estos datos en la información requerida por la contraloría para cumplir sus objetivos como entidad fiscalizadora de los bienes de la nación.

1.5 ALCANCE

El alcance de este proyecto permitirá desarrollar una herramienta capaz de contrastar y presentar a través de gráficas, tablas y cuadros de comparación la diferente información que las contralorías municipales de Santander reportan a la Contraloría General de Santander con el fin de brindar información de alta calidad para facilitar la comprensión de dichos reportes.

El lenguaje de programación al que se recurrirá para el desarrollo de esta herramienta será Java EE, PostgreSQL, PG Designer, Jsf.

Cabe resaltar que son herramientas con licencia de Software Libre.

2. MARCO TEORICO

2.1 CONTROL FISCAL

Es una función pública que ejerce la Contraloría General de la República, la cual vigila la gestión fiscal de la administración y de los particulares o entidades que manejen fondos o bienes de la Nación. La preceptiva constitucional contempla, dentro de las facultades del Contralor General de la República, aplicables a los contralores departamentales, distritales y municipales, la de exigir informes sobre su gestión fiscal a los empleados oficiales de cualquier orden y a toda persona o entidad pública o privada que administre fondos o bienes de la Nación.

El control fiscal tiene un fundamento jurídico, toda su actividad está enmarcada dentro del ordenamiento legal.

El Control Fiscal como función del Estado, está distribuido entre diversas Entidades de Derecho Público. La actuación de estas Entidades debe estar enmarcada dentro de la súper legalidad, es decir, dentro de las normas constitucionales y dentro de la legalidad, leyes, reglamentos y decretos que predominan en nuestros sistemas de gobierno.

Figura 1. Control Fiscal



El desarrollo de este trabajo de investigación apoya el funcionamiento de la contraloría general de Santander, quien es el ente encargado de ejercer el control fiscal sobre el patrimonio del estado.

Los objetivos que se propone la contraloría como entidad encargada de vigilar el patrimonio del estado son:

- Representar y vigilar en nombre de la comunidad la gestión fiscal y administración de los particulares o entidades que manejen fondos o bienes del departamento, municipios, entidades descentralizadas y demás del departamento y de los municipios.
- Evaluar los resultados obtenidos por las diferentes organizaciones y entidades del Estado en la correcta, eficiente, económica, eficaz y equitativa administración del patrimonio público, de los recursos naturales y del medio ambiente.
- Generar una cultura del control del patrimonio del Estado y de la gestión pública.

- Establecer las responsabilidades fiscales e imponer las sanciones pecuniarias que correspondan y las demás acciones derivadas del ejercicio de la vigilancia fiscal.
- Procurar el resarcimiento del patrimonio público.

Para el cumplimiento de su misión y de sus objetivos, en desarrollo de las disposiciones consagradas en la Constitución Política y en apoyo con las tecnologías de la información, le corresponde a la Contraloría General de Santander:

- Ejercer la Vigilancia de la gestión fiscal del Departamento de Santander, los municipios y las entidades descentralizadas del orden departamental y municipal a través, entre otros, de un control financiero, de gestión y de herramientas basadas en las tecnologías de la información que faciliten resultados, fundados en la eficiencia, la equidad, la economía y la valoración de los costos ambientales.
- Ejercer la vigilancia de la gestión fiscal conforme a los sistemas de control, procedimientos y principios que establezcan la Ley y el Contralor General de Santander, mediante resolución.
- Ejercer el control fiscal, posterior y selectivo, sobre las cuentas que rinden la Gobernación de Santander, los municipios y demás entidades sometidas bajo su control.
- Ejercer funciones administrativas y financieras propias de la entidad, para el cabal cumplimiento y desarrollo de las actividades de la gestión del control fiscal.
- Conceptuar sobre la calidad y eficiencia del control interno en los términos previstos en la Constitución y la Ley.

2.2 MODELO VISTA CONTROLADOR (MVC)

Uno de los patrones más conocidos en el desarrollo web es el patrón MVC (Modelo Vista Controlador). Este patrón permite separar la lógica de control, la lógica de negocio y la lógica de presentación.

El Modelo, las Vistas y los Controladores se tratan como entidades separadas, es decir, cualquier cambio producido en el Modelo se refleja automáticamente en cada una de las Vistas.

Ventajas

- Permite normalizar y estandarizar el desarrollo de software.
- Maneja componentes por separado.
- Hay una API bien definida que permite modificar o reemplazar el modelo, la vista o el controlador por separado sin dificultad alguna.
- La conexión entre el modelo y sus vistas se produce en tiempo de ejecución no en tiempo de compilación.

Modelo

La parte del modelo se refiere exactamente a aquella parte de la realidad sobre la que se desea actuar como por ejemplo itinerarios de viaje, supermercados, etc. Este es el que utiliza directamente los datos, el modelo gestiona y controla toda la metodología que se deba aplicar a estos para convertirlos en información.

Esta capa, no posee referencias a las vistas ni a los controladores. El sistema es quien se encarga de mantener enlazadas las vistas al modelo.

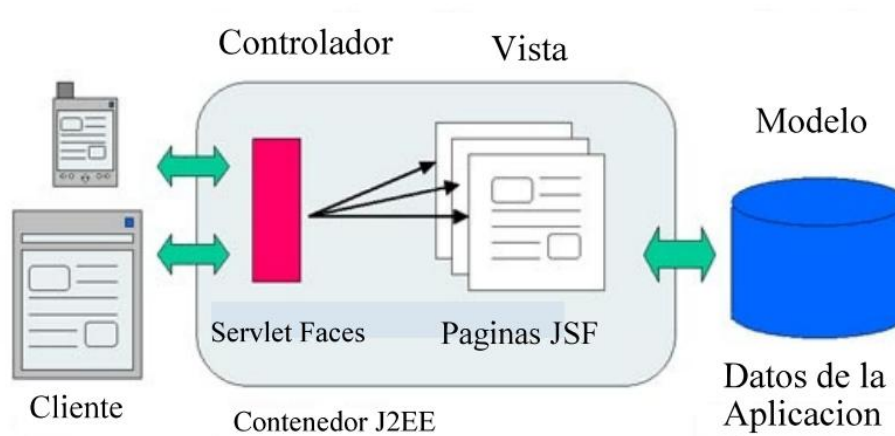
Vista

Es la encargada de realizar la presentación visual de los datos que controla el modelo. Está referenciado al modelo, pues es quien debe generar una representación visual de los datos entrantes.

Controlador

Es quien da significado a las acciones del usuario en el navegador, es decir, cuando se realiza alguna operación ya sea sobre el modelo o sobre la vista.

Figura 2. Modelo Vista-Controlador¹



2.3 JAVA ENTERPRISE EDITION 6.0

Actualmente se puede observar la necesidad de aplicaciones distribuidas, transaccionales y portables que sean veloces, seguras y confiables. Su objetivo es proporcionar a los desarrolladores un conjunto de APIs acortando el tiempo de desarrollo, reduciendo la complejidad de la aplicación, y mejorando el rendimiento de las aplicaciones.

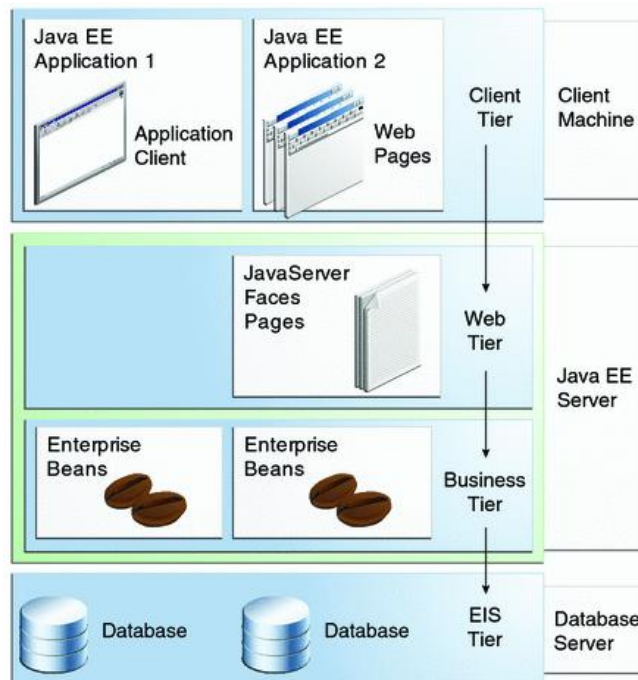
Esta plataforma utiliza un modelo de aplicación distribuida multicapa para aplicaciones empresariales. La lógica de las aplicaciones se divide en varios componentes según su función.

¹¹SICUMA: Sistemas de Información Cooperativos Universidad de Málaga. *Tutorial de JavaServer Faces.P. 18*{En línea}. {02 de Octubre de 2011}. Disponible en: <http://www.sicuma.uma.es/sicuma/Formacion/documentacion/JSF.pdf>

Java EE define los componentes de la siguiente manera:

- Los clientes de aplicaciones, que son componentes que se ejecutan en el ordenador cliente.
- Tecnología de Java Servlet, JavaServer Faces, son componentes que se ejecutan en el ordenador del servidor.
- JavaBeans (EJB), son componentes de negocio que se ejecutan en el servidor.

Figura 3. Esquema de Funcionamiento JavaEE 6.0



Cuando hablamos de clientes de aplicaciones, se debe distinguir entre si es un cliente web o si es una aplicación cliente.

Un cliente web se compone de dos partes, páginas web dinámicas que contiene varios tipos de lenguajes de marcas como HTML, XML, etc., que son generados por los componentes web que se ejecutan en la capa web, y de un navegador web que responde a las peticiones del cliente.

Una aplicación cliente se ejecuta en una máquina cliente y proporciona gran cantidad de servicios y tareas a comparación de lo que se puede implementar con un lenguaje de marcas. Es usualmente una interfaz gráfica de usuario que brinda servicios conforme a las reglas del negocio.

Las aplicaciones cliente pueden conectarse igualmente con aplicaciones web a través de un JavaServlet.

Los componentes de una aplicación web son Java Servlets o páginas web creadas usando un framework como JavaServer Faces.

2.4 JAVASERVER FACES

JavaServer faces proporciona un entorno de trabajo, conocido como framework, vía web, es decir, gestiona las acciones producidas por el navegador del usuario y los traduce en eventos que son enviados al servidor. El lenguaje de programación del lado del cliente está basado en HTML.

JavaServer Faces maneja interfaces de usuario del lado del servidor, y están basadas en tecnología JAVA y en el patrón MVC o conocido como modelo vista controlador.

Esta tecnología separa el comportamiento y la presentación de una aplicación web. Separar la lógica del negocio de la de presentación proporciona una división estratégica en el equipo de desarrollo y así mismo reutilizar los componentes desarrollados.

JavaServer Faces proporciona una arquitectura amplia para manejar el estado de los componentes, procesar los datos, validar la entrada del usuario y manejar eventos.

Cada aplicación desarrollada en JSF se ejecuta en un contenedor de Servlets de Java y contienen:

- Componentes JavaBeans
- Oyentes de eventos
- Páginas
- Clases de utilidad de lado del servidor, como beans para acceder a las bases de datos
- Librerías de etiquetas personalizadas para dibujar componentes de Interfaz de usuario
- Librerías de etiquetas personalizadas para representar manejadores de eventos, validadores y otras acciones.
- Componentes de interfaz de usuario como objetos con estado en el servidor.

JavaServer Faces posee componentes de interfaz de usuario. Estos son elementos configurables y reutilizables. Uno de estos componentes puede ser un botón, una tabla, los cuales pueden estar compuestos de más componentes. JSF proporciona:

- Conjunto de clases UI component para especificar el estado y comportamiento de componentes utilizados en las interfaces de usuario.
- Un modelo de eventos y métodos que acceden a la información que definen como manejar los eventos de los componentes.
- Un modelo de conversión que define como conectar conversores de datos a un componente.
- Un modelo de validación que define como registrar validadores con un componente.

2.4.1 JavaBeans

Es una clase que contiene atributos. Un atributo es un valor identificado con un nombre. Pertenece a un tipo determinado y puede ser leído y/o escrito a través de métodos getter y setter, que son los métodos para obtener y establecer los valores de atributo.

Es necesario usar beans en una aplicación desarrollada en JSF para todos los datos accedidos por una página. Estos se caracterizan por ser el conducto de comunicación entre la interfaz de usuario y la aplicación desarrollada.

Un JavaBean es un componente reutilizable del software, que puede ser manipulado, es decir, se usan cuando se necesita conectar las clases java con las páginas web o archivos de configuración.

2.4.2 API JavaServlet

Provee un simple y consistente mecanismo para extender las funcionalidades de un servidor web y el acceso a un sistema de negocio existente. Se puede entender un JavaServlet como un componente de una aplicación que se ejecuta en el contexto de otro programa que se ejecuta en el lado del servidor para dar servicio a la aplicación en internet.

2.4.2.1 Ciclo de Vida de un JavaServlet. Los JavaServlet también poseen un ciclo de vida:

- Un servidor carga e inicializa el servlet
- El servlet maneja cero o más peticiones del cliente
- El servidor elimina el servlet

La inicialización de un `JavaServlet` se completa antes de manejar peticiones de clientes y antes de que el servlet sea destruido. Los Servlets no tienen problemas de concurrencia durante su inicialización. El servidor no puede recargar un servlet sin primero haberlo destruido.

Después de la inicialización, el servlet puede manejar peticiones de clientes. Estas respuestas son manejadas por la misma instancia del servlet por lo que hay que tener cuidado con acceso a variables compartidas por posibles problemas de sincronización entre requerimientos concurrentes.

Los Servlets se ejecutan hasta que el servidor los destruye, por cierre el servidor o por petición del administrador del sistema.

2.4.2.2 Características

- Son independientes del servidor utilizado y del sistema operativo, es decir, no interesa si el servidor donde reside la aplicación está escrito en otro lenguaje pues el resultado será el mismo.
- Un servlet puede llamar otro servlet o incluso sus métodos.
- Pueden obtener fácilmente información acerca del cliente que este conectándose con la aplicación. Esa información es dirección ip, el puerto que utiliza la aplicación, obtener el método solicitado del navegador del cliente bien sea get o post.
- Permiten el uso de cookies (fragmento de información que se almacena en el disco duro del visitante de una aplicación web a través de su modo de petición al servidor) y sesiones, es decir, se puede guardar información del visitante con el fin de mantener una comunicación.
- Pueden relacionar un cliente con una base de datos que este ubicada en un servidor distinto.
- Son clases que procesan peticiones y ejecutan respuestas.

2.4.3 Managed Bean

La parte importante que diferencia a JavaServer Faces de las demás metodologías de desarrollo web es la separación de la presentación de la aplicación y la lógica de negocio que se debe manejar. JSF usa los beans para lograr esta separación. Las paginas Jsf se refieren a las propiedades del JavaBean, y la lógica de programa está contenida en el código de implementación del bean. Los beans se encargan de conectar las clases java con las vistas de la aplicación web o con los archivos de configuración.

2.4.4 Persistencia de Java

Es la función de leer y guardar la información de un objeto de forma permanente. Esta usa un enfoque de mapeo objeto- relacional para cerrar la brecha entre un modelo orientado a objetos y una base de datos relacional. Se compone de:

- API de persistencia
- Lenguaje de consulta
- Mapeo de metadatos objeto-relacional

Java maneja varias herramientas para la persistencia como son:

- EclipseLink JPA 2.0
- Hibernate JPA 1.0 y 2.0
- TopLink Essentials JPA 1.0
- KODO
- OpenJPA JPA 2.0 y 1.0

Son herramientas que facilitan el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos.

2.4.5 Conectividad Con La Base De Datos Java (JDBC)

Permite invocar comandos SQL desde los métodos utilizados en la programación de las aplicaciones java. Se utiliza cuando se desea construir una aplicación de negocio donde se hace necesario el uso de JavaBeans y se desea acceder a la base de datos directamente. Está compuesto por dos partes:

- Una interfaz utilizada por los componentes de la aplicación que realizan la conexión con la base de datos.
- Una interfaz que se encarga de enlazar el controlador JDBC con la plataforma Java EE.

2.4.6 Servidor GlassFish

Es una implementación compatible de la plataforma Java Enterprise Edition. El servidor GlassFish incluye gran número de herramientas soporte para el desarrollador. Igualmente brinda soporte a las interfaces de programación como JSP, JavaServer Faces entre otras.

Posee herramientas como:

- Consola de administración para manejar usuarios, recursos y aplicaciones.
- Una línea de comandos para la administración del servidor
- Una línea de comandos que permite lanzar aplicaciones cliente
- Una línea de comandos para extraer el esquema de información de la base de datos
- Una línea de comandos para encapsular los contenedores y librerías de la aplicación cliente en archivos JAR
- Realizar copias de la base de datos

2.4.7 Ajax

Brinda la capacidad de construir aplicaciones web dinámicas e interactivas. Usando Ajax en la creación de una aplicación web se puede obtener información de un servidor sin interferir con la parte visual de un cliente.

Anteriormente se creaban aplicaciones web estáticas, es decir si se deseaba actualizar una página web, tenía que recargar toda la página completa imposibilitando al cliente interactuar con la aplicación. Recargar las páginas web repetidamente sobrecarga la red, y puede deteriorar el rendimiento de la aplicación.

2.5 NETBEANS

Es un entorno de desarrollo, de código abierto. Una herramienta en la cual se puede escribir programas, compilar, corregir errores, y ejecutar programas. Está escrito en Java. Es un producto libre y gratuito sin restricciones de uso.

Permite que las aplicaciones sean desarrolladas a partir de componentes de software o módulos. Son archivos java que contienen clases de java escritas para interactuar con las apis de netbeans. Estos módulos pueden ser desarrollados independientemente, pudiéndose extender y extender lo que sea necesario.

Esta plataforma ofrece servicios comunes a las aplicaciones de escritorio permitiéndole al desarrollador enfocarse en la lógica específica de la aplicación deseada.

Permite administrar las interfaces de usuario, las configuraciones del usuario, el almacenamiento de cualquier tipo de dato, las ventanas.

2.6 SERVIDOR WEB

Se trata de un programa que procesa una o varias aplicaciones del lado de un servidor. Esto es una conexión o comunicación iniciada a través de los navegadores web. El protocolo utilizado por los navegadores es http. La comunicación inicia en el cliente cuando hace una petición al servidor de la aplicación web, donde el servidor procede a enviar una respuesta a esta petición a través del navegador en el ordenador del cliente, quien interpreta y despliega en pantalla la solicitud. Un ejemplo de petición es escribir en la barra de direcciones `www.google.com.co`, allí está solicitando al servidor de aplicaciones de google desplegar la vista que se especifica en la url.

La tarea de un servidor web es alojar sitios y/o aplicaciones y permitir que otros ordenadores ejecuten sus peticiones en el ordenador del servidor visualizando los procesos ejecutados en el ordenador del cliente.

Los servidores permiten el uso de diversas estrategias para la tolerancia a fallos y la estabilidad, como el balanceo de carga.

2.7 SERVIDOR DE APLICACIONES

El servidor de aplicaciones se diferencia de los servidores web en que presenta la lógica de negocio a través de protocolos, mientras que el servidor web se encarga de responder solicitudes del ordenador cliente con paginas HTML.

En los servidores de aplicaciones se transmiten paginas HTML, pero lo más importante es que transmiten también datos de la lógica del programa.

Un servidor de aplicaciones gestiona los recursos requeridos por la aplicación incluyendo la seguridad, procesamiento de transacciones y mensajería.

2.8 SOFTWARE LIBRE

Se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

El software libre brinda cuatro libertades a los usuarios de este tipo de software:

- La libertad de usar el programa con cualquier propósito
- La libertad de estudiar cómo funciona el programa, y adaptarlo a las necesidades
- La libertad de distribuir copias
- La libertad de mejorar el programa y redistribuir

El software libre es propiedad de todos, cada persona tiene derecho a usar el software, modificarlo y copiarlo.

Algunos gobiernos han ido adoptando el software libre porque les permite ahorrar dinero en implementar servidores de tecnología Linux. También se conoce por la seguridad que maneja este tipo de sistemas.

2.8.1 Ventajas del Software Libre

- Coste
- Innovación tecnológica
- Requisitos de hardware y durabilidad de las soluciones
- Independencia del proveedor
- Adaptación del software

2.9 JAVA DEVELOPMENT KIT (JDK)

Es un software que brinda herramientas al desarrollador para la creación de programas en Java. Incluye programas como appletviewer que es un visor de applets para generar vistas previas. El compilador de java es Javac, java que es el intérprete del lenguaje Java, y un javadoc que genera documentación de las clases Java de un programa.

2.10 JAVA RUNTIME ENVIRONMENT (JRE)

Es un conjunto de utilidades que permiten la ejecución de programas Java.

Está conformado por una máquina virtual de java, un conjunto de bibliotecas java y otros componentes necesarios para que una aplicación escrita en lenguaje Java pueda ser ejecutada. Actúa como un intermediario entre el sistema operativo y Java.

La máquina virtual de java es el programa que ejecuta el código java previamente compilado mientras que las librerías de clases estándar son las que implementan el API de Java. Ambas JVM y API deben ser consistentes entre sí, de ahí que sean distribuidas de modo conjunto.

3. BASES DE DATOS

Se define base de datos¹ como un conjunto de archivos almacenados de forma integrada y compartida, el cual implica el uso de un software manejador. Un ejemplo de base de datos puede ser una biblioteca, pues almacena diferentes tipos de libros que brindan información detallada de diferentes temas.

Cada base de datos se compone de tablas que son las encargadas de guardar los datos. Cada tabla tiene 1 o más columnas y filas. Las columnas guardan parte de la información sobre cada elemento que se quiere guardar, y las filas conforman registros.

Pueden clasificarse de acuerdo al contexto en el que se desean utilizar o las necesidades que se desean satisfacer.

Existen distintos software encargados de gestionar las bases de datos como MySQL, PostgreSQL, Oracle, DB2, SQL Server.

MySQL y PostgreSQL son basados en software libre.

Estos gestores se encargan de introducir, organizar y recuperar la información que se almacena en la base de datos.

Según la variabilidad de los datos, pueden ser estáticas o dinámicas. Estáticas cuando son bases de datos de solo lectura. Y dinámicas cuando la información almacenada se está modificando constantemente, permitiendo actualizaciones, borrado y adición de datos además de las operaciones de consulta.

¹ALBARRACIN, Jaime. Curso bases de datos. p. 12

Las bases de datos se rigen por un modelo de administración de datos llamado “Modelo de datos”, el cual es un esquema en donde se plasma la estructura interna de funcionamiento de la información. Esta estructura interna de funcionamiento describe como esta ordenada y como será utilizada la información.

Varios modelos se han establecido a medida que avanza la tecnología:

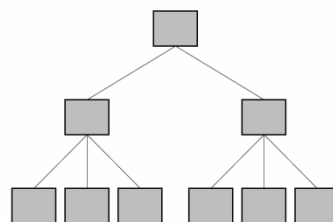
- Modelo Jerárquico
- Modelo de red
- Modelo relacional
- Modelo entidad-relación

3.1 MODELO JERÁRQUICO

Son bases de datos que como su nombre lo indica, organiza la información en una estructura jerárquica, es decir, similar a un árbol visto al revés, en donde un nodo padre puede tener varios hijos. El nodo que no posee padre se le llama nodo raíz, y a los nodos hijos se les llama hojas.

Una de sus limitaciones es la incapacidad de representar eficientemente la redundancia de datos.

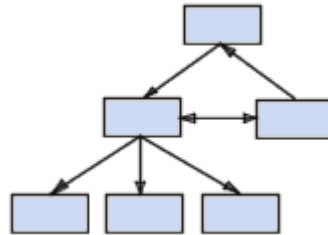
Figura 4. Modelo Jerárquico



3.2 MODELO DE RED

Es un modelo en el que se representan objetos y sus relaciones. Este modelo permite a cada registro tener múltiples registros padres e hijos, formando una especie de enrejado con la información.

Figura5. Modelo de Red



Su ventaja sobre el modelo jerárquico es que permite realizar un modelado más natural de las relaciones entre entidades.

3.3 MODELO RELACIONAL

Es el modelo más utilizado actualmente. Se caracteriza por el uso de relaciones. Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados tuplas.

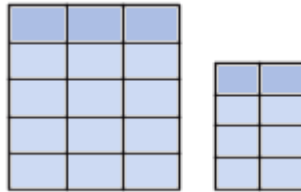
Los datos son almacenados en relaciones. Cada relación es un conjunto de datos.

En este modelo se considera la base de datos como una colección de relaciones. Una relación representa un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor.

Cada fila recibe el nombre de tupla o registro y cada columna el de atributo.

Este modelo maneja la información a través del Álgebra Relacional y el Cálculo Relacional. El álgebra permite describir la forma de realizar una consulta y el cálculo solo indica lo que se desea devolver.

Figura6. Modelo Relacional



3.4 MODELO ENTIDAD-RELACIÓN

Es un tipo de modelo de datos conceptual de alto nivel que se emplea en el diseño de las bases de datos relacionales. Este modelo muestra la estructura de la base de datos empleando todo tipo de herramientas conceptuales.

Para poder construir este modelo es necesario seguir ciertos pasos:

- Obtener y analizar los requerimientos del cliente.
- Crear un esquema conceptual de la base de datos. Este es el que contiene las descripciones detalladas de las entidades, relaciones y restricciones.
- Implementar un Software manejador de bases de datos que construya lo que el modelo de datos plasma.

Entidad: Objeto del mundo real sobre el que se desea guardar información. Por ejemplo, una persona. Estas entidades están compuestas de atributos, los cuales son los que definen este objeto. Continuando con el ejemplo de la persona, los atributos serían su número de cédula, nombre, sus apellidos, su dirección de residencia, etc. De estos atributos algunos no se repetirán, estos reciben el

nombre de clave de la entidad. Para la persona que definimos en el ejemplo, su clave sería su número de cedula.

Relación :Es la asociación que se hace con las entidades. Es necesario que refleje las interacciones que puedan existir entre las entidades. Estas relaciones pueden ser de 3 tipos:

➤ Relaciones 1-1

Las entidades que intervienen poseen registros que se relacionan uno a uno. Por ejemplo, un municipio tiene un área rural y esa área rural solo le corresponde a ese municipio.

➤ Relaciones 1-Muchos

Un registro de una entidad se relaciona con varios registros en otra entidad, pero esos registros solo se relacionan con esa entidad. Por ejemplo, un área rural tiene varias veredas pero cada una de esas veredas solo pertenecen a esa área rural.

➤ Relaciones Muchos-Muchos

Muchos registros de una entidad se relacionan con muchos registros en otra entidad. Por ejemplo, una parcela puede tener varios dueños y cada dueño puede tener varias parcelas.

3.4.1 Claves

También denominadas llaves. Son atributos que identifican una entidad dentro de un conjunto de entidades.

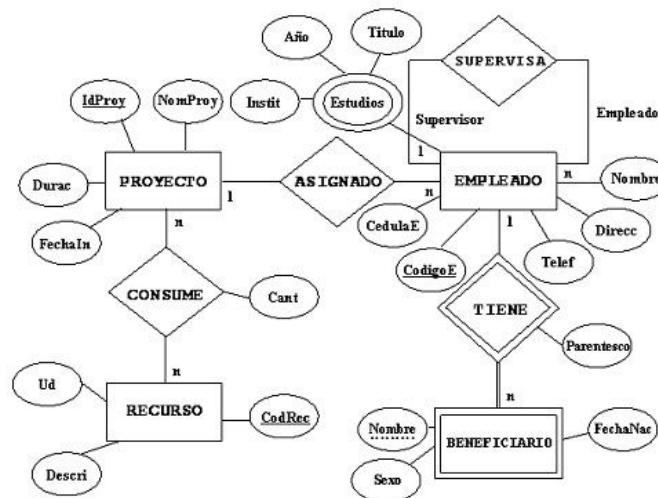
Entre las claves se encuentra una que recibe el nombre de “Superclave”, es el conjunto de atributos no vacío, que identifica en forma única una entidad dentro de un conjunto de entidades.

“Clave candidata” es una superclave para la cual ningún subconjunto es superclave, excepto el mismo.

“Clave primaria” o llave primaria es una clave candidata que identifica en forma única una tupla en la tabla.

“Clave foránea” o llave foránea es un atributo que es llave primaria en otra entidad con la cual se relaciona. Las llaves foráneas son en últimas las que permiten relacionar las tablas en las bases de datos.

Figura7. Modelo Entidad – Relación



3.4.2 Integridad Referencial

La integridad referencial garantiza que una entidad siempre se relaciona con otras entidades validas, es decir, si una entidad está asociada con otra y se desea eliminar una de ellas, la integridad referencial no lo permite y el software manejador de bases de datos es quien vela por no permitir este tipo de

asociaciones. Esto conlleva a que todos los datos ingresados son correctos, sin redundancia, sin datos perdidos o relaciones mal hechas.

3.4.3 Formas Normales

La tarea de crear la base de datos consiste en estructurar los datos, es decir, que no existan duplicaciones innecesarias. El proceso de Normalización es quien permite perfeccionar las bases de datos. Es un proceso en el que se aplican ciertas reglas. La normalización se define como el proceso de descartar la repetición de grupos, minimizar la redundancia, eliminar claves compuestas y separar los atributos que no sean de la clave.

Ventajas:

- ✓ Integridad de datos
- ✓ Consultas optimizadas
- ✓ Creación y ordenación de índices más rápido
- ✓ Ejecución más rápida de la actualización de las tablas
- ✓ Resolución de concurrencias mejorada

Para normalizar basta con observar si las tablas poseen información repetida, si es así, deben dividirse las tablas en tablas independientes para eliminar la duplicación.

Existen ciertos criterios para normalizar una base de datos, las formas normales. Para cubrir las necesidades de una base de datos, se considera que es suficiente aplicar las tres primeras formas normales².

Primera Forma Normal: “La relación está en primera forma normal cuando el valor de sus atributos en cada tupla es atómico”

²ALBARRACIN, Jaime. Curso bases de datos. p.162

Esto quiere decir que cuando se escoge en la tabla una fila con una columna siempre existirá 1 solo valor.

Segunda Forma Normal: “Una relación está en segunda forma normal si y solo si esta en Primera Forma Normal y todos sus atributos no clave dependen por completo de la clave principal”

Es decir que todos los atributos que no son clave principal deben depender únicamente de la clave principal.

Tercera Forma Normal: “Una relación esta en tercera forma normal si y solo si esta en segunda forma normal y todos los atributos no clave dependen de manera no transitiva de la clave principal.”

Está en tercera forma normal cuando se eliminan las dependencias y se generan todas las relaciones externas con otras tablas u otras bases de datos.

Forma Normal de Boyce & Codd (BCNF):”Una relación esta en forma normal de Boyce y Codd si y solo si todo determinante es una clave candidata”

Cuarta Forma Normal: “Una relación esta en cuarta forma normal si y solo si esta en BCNF y no contiene dependencias multivaluadas”.

En general las 3 primeras formas normales son suficientes para cubrir necesidades de la mayoría de las bases de datos.

3.5 GESTORES DE BASES DE DATOS

Es un gestor de base de datos que se encarga de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización. Maneja la privacidad, integridad, y seguridad de los datos. Permite la interacción del usuario con la base de datos a través de una interfaz gráfica, es decir, que cualquier operación realizada en la base de datos es gestionada por el software manejador que está siendo utilizado por el usuario.

El gestor almacena una descripción de datos que recibe el nombre de diccionario de datos, así como los usuarios permitidos y los permisos. En este diccionario se guardan todas las propiedades de la base de datos, descripción de la estructura utilizada y relaciones entre los datos.

Entre los gestores de base de datos libres se encuentran:

- PostgreSQL
- Firebird
- SQLite
- DB2 Express-C
- Apache Derby
- MariaDB
- MySQL

Y entre los no libres:

- Advantage Database
- dBase
- FileMaker
- Fox Pro

- gsBase
- IBM DB²
- IBM Informix
- MAGIC
- Microsoft Access
- Microsoft SQL Server
- Nexus DB
- Open Access
- Oracle
- Paradox

3.5.1 PostgreSQL

Sistema de gestión de base de datos relacional orientado a objetos y basado en software libre. Es un software mantenido por varios desarrolladores que trabajan de forma desinteresada y patrocinados por organizaciones comerciales (PostgreSQL Global Development Group).

4. PLAN DE TRABAJO

El plan de trabajo especifica las actividades y el análisis que se llevó a cabo para obtener el cumplimiento de cada uno de los objetivos propuestos.

Este proyecto se desarrolla en el marco de 4 módulos propuestos para cumplir unos objetivos planteados que brindan una solución a la problemática que actualmente posee la Contraloría General de Santander:

- Módulo de registro (Objetivo específico No. 1.2.2.1)
- Módulo vista por formato (Objetivo específico No. 1.2.2.2)
- Módulo vista por ente (Objetivo específico No. 1.2.2.3)
- Módulo de contraste (Objetivo específico No. 1.2.2.4)

Estos módulos procesan los distintos formatos que de acuerdo a unas vigencias específicas creadas por la contraloría, cada ente público debe reportar.

Formato 1. Catálogo de Cuentas: Se diligencia ingresando las subcuentas de acuerdo al código contable asignado en el catálogo de cuentas expedido por la Contaduría General de la Nación para los entes públicos, de acuerdo a la naturaleza de las cuentas manejadas por cada sujeto de control para mostrar los resultados de la gestión de la vigencia terminada.

Formato 2A. Resumen de Caja Menor: Este formato permite adicionar una o más cajas menores, dependiendo de las que utiliza la entidad.

Formato 2B. Relación de Gastos de Caja: En este formato, inicialmente debe indicar el número de la caja menor a relacionar en la columna No Caja, de acuerdo con el número registrado en el Formato 2A posteriormente debe registrar la

relación de gastos con respecto a los rubros establecidos en la columna Nombre del Gasto y posteriormente los valores ejecutados y autorizados. Esta relación debe diligenciarse teniendo en cuenta que el número de caja varía, de acuerdo a la cantidad de cajas menores que hayan constituido durante la vigencia rendida

Formato 3. Cuentas Bancarias: En este formato se relacionan las cuentas constituidas por las entidades sujetas de control y vigilancia en bancos y Entidades financieras

Formato 4. Pólizas de Aseguramiento: El objetivo de este formato es verificar el cumplimiento del artículo 107 de la ley 42 de 1993, en este se registra la totalidad de las pólizas vigentes a 31 de diciembre del año rendido.

Formato 5a. Propiedad, Planta y Equipo: Permite registrar todas las adquisiciones y bajas de elementos durante la vigencia rendida.

Formato 6. Ejecución Presupuestal de Ingresos: Este formato describe los ingresos agrupados en Ingresos Corrientes (Tributarios, No tributarios y Transferencias) y de Capital.

Formato 7. Ejecución Presupuestal de Gastos: Este formato describe los gastos agrupados en Gastos de Funcionamiento (de Personal, Generales y Transferencias), Servicios de la Deuda y Gastos de Inversión.

Formato 13A. Nuevo Formato de Contratación: En este formulario se debe relacionar los contratos celebrados durante el período reportado, así mismo, los que fueron celebrados en vigencias anteriores y se ejecutan o terminan en la vigencia rendida.

4.1 ESPECIFICACIÓN Y ANÁLISIS DE REQUERIMIENTOS

La especificación de requisitos genera un documento con una descripción completa de las necesidades y funcionalidades que el sistema debe ofrecer.

En esta fase se encontraron las falencias que posee el actual sistema de auditoría que manejan llamado SIA, el cual fue un avance del sistema tradicional de manejo de la información que era escrito en papel físico. Sin embargo esta herramienta software no está cumpliendo con las necesidades requeridas por la contraloría.

La fase de requerimientos comprendió las siguientes tareas:

- Análisis y diseño del formato No.1 Catalogo de cuentas
- Análisis y diseño del formato No. 2A Resumen de Caja Menor
- Análisis y diseño del formato No. 2B Relación de Gastos de Caja
- Análisis y diseño del formato No. 3 Cuentas Bancarias
- Análisis y diseño del formato No. 4 Pólizas de Aseguramiento
- Análisis y diseño del formato No. 5A Propiedad, Planta y Equipo – Adquisiciones y Bajas
- Análisis y diseño del formato No. 5B Propiedad, Planta y Equipo – Inventario
- Análisis y diseño del formato No. 6 Ejecución Presupuestal de Ingresos
- Análisis y diseño del formato No. 7 Ejecución Presupuestal de Gastos
- Análisis y diseño del formato No. 7A Relación de pagos
- Análisis y diseño del formato No. 8A Modificaciones al Presupuesto de Ingresos
- Análisis y diseño del formato No. 8B Modificaciones al Presupuesto de Egresos
- Análisis y diseño del formato No. 9 Ejecución PAC de la Vigencia
- Análisis y diseño del formato No. 10 Ejecución Reserva Presupuestal
- Análisis y diseño del formato No. 11 Ejecución Presupuestal de Cuentas por Pagar

- Análisis y diseño del formato No. 11A Vigencias Futuras
- Análisis y diseño del formato No. 12 Proyectos de Inversión
- Análisis y diseño del formato No. 12A Presupuesto Proyectos de Inversión
- Análisis y diseño del formato No. 13A Nuevo Formato de Contratación
- Análisis y diseño del formato No. 13B Ingresos por Estampillas
- Validar que los formatos no sean diligenciados con datos incorrectos o vacíos.
- Visualizar el nombre del rubro presupuestal junto con el código del rubro presupuestal en los formatos que manejan este campo. El rubro presupuestal es un código y un nombre que se asigna a un gasto o ingreso monetario. Este rubro es único y no se puede repetir.
- Visualizar el campo código contable junto con su respectivo nombre de la cuenta para los formatos que manejen alguno de estos dos campos. Estos campos son únicos y no se deben repetir.
- Prohibir a la entidad pública que reporta información, introducir campos erróneos en la base de datos, es decir, que si el código contable es un número no permitir que introduzcan letras en él.
- Verificar que el valor total por el que se realizan los contratos sea el mismo que aparece en el formato 7 (ejecución presupuestal de gastos).
- Permitir que el sistema a partir de la información reportada, diligencie el formato 20 que es el que la contraloría debe reportar a la auditoría general de la república.
- Realizar una comparación entre el campo compromisos y el campo pagos del formato 7 (ejecución presupuestal de gastos).
- Denegar a las diferentes entidades obligadas a reportar información cargar sus reportes finales con campos vacíos o abreviaturas como n/a, pues los formatos establecidos por legislatura obligan a que se diligencien uno a uno los campos de los formatos para el debido estudio del dinero entregado a las entidades públicas.

- Crear cuentas de acceso para cada uno de los involucrados con el sistema de auditoria bien sea entidad pública o funcionario de la contraloría.
- Llevar un control de la persona que debe diligenciar la información por parte de cada ente público.

4.2 DISEÑO

En esta fase se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo.

El diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente; debe ser una guía que puedan leer y entender los que construyan el código y los que prueban y mantienen el software.

En esta fase se debe realizar el diseño de:

- Entradas y salidas del sistema
- Base de datos
- Procesos
- Interfaces de usuario
- Programación

4.3 DESARROLLO

Es la fase de codificación del sistema. Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.

Se comprueba que el software ya desarrollado realice las tareas indicadas para solucionar la problemática.

5.METODOLOGIA DE TRABAJO

Describiremos la metodología usada para el desarrollo de este sistema. Las fases aplicadas son especificación de requerimientos, análisis, diseño, desarrollo de la aplicación (codificación) y pruebas.

5.1 FASE INICIAL

Inicialmente se realizaron visitas a la contraloría general de Santander en las cuales se documentó la problemática que posee actualmente esta entidad, las funciones que realizan, los mecanismos utilizados actualmente en la entidad para cumplir con sus objetivos y la debida toma de requerimientos.

Esta entidad se encarga a través de unos formatos pertenecientes a un sistema desarrollado por la auditoria general de la república que recibe el nombre de Sistema Integral de Auditoria (SIA) de realizar el debido control fiscal sobre los entes públicos del departamento de Santander. Los formatos necesarios para auditar a cada ente público son:

- Formato 1. Catálogo de Cuentas
- Formato 2A. Resumen de Caja Menor
- Formato 2B. Relación de Gastos de Caja
- Formato 3. Cuentas Bancarias
- Formato 4. Pólizas de Aseguramiento
- Formato 5a. Propiedad, Planta y Equipo
- Formato 6. Ejecución Presupuestal de Ingresos
- Formato 7. Ejecución Presupuestal de Gastos
- Formato 13A. Nuevo Formato de Contratación

Estos formatos deben ser diligenciados de manera obligatoria por los diferentes entes públicos de acuerdo a su tipo. Estos entes públicos se clasifican en 6 tipos:

- Alcaldías
- Asamblea
- Descentralizada
- Educativa
- Gobernación
- Salud

Existe un límite de tiempo definido como vigencia por la contraloría, donde se establecen fechas para que los formatos puedan ser cargados en el servidor de la aplicación de la auditoría, con un plazo de meses como máximo, en los que la entidad pública debe justificar el destino de los fondos recaudados del estado diligenciando los formatos respectivos.

5.2 ESPECIFICACION DE REQUERIMIENTOS

La especificación de requerimientos se realizó en sesiones de 2 horas, dos días cada semana. Se realizaron reuniones con los diferentes nodos que componen la contraloría general de Santander. Estos nodos son:

- García Rovira
- Guanentino
- Comunero
- Soto y Mares
- Vélez
- Central

Cada uno de estos nodos está encargado de ciertas zonas del departamento de Santander en el que deben realizar control sobre el patrimonio que el estado entrega a cada uno de los entes públicos que conforman el departamento de Santander.

Se pudo recopilar las sugerencias que la contraloría tiene para que su sistema les permita desempeñar las funciones requeridas eficientemente:

- Visualizar el nombre del rubro presupuestal junto con el código del rubro presupuestal en los formatos que manejan este campo. El rubro presupuestal es un código y un nombre que se asigna a un gasto o ingreso monetario. Este rubro es único y no se puede repetir.
- Visualizar el campo código contable junto con su respectivo nombre de la cuenta para los formatos que manejen alguno de estos dos campos. Estos campos son únicos y no se deben repetir.
- Prohibir a la entidad pública que reporta información, introducir campos erróneos en la base de datos, es decir, que si el código contable es un número no permitir que introduzcan letras en él.
- Verificar que el valor total por el que se realizan los contratos sea el mismo que aparece en el formato 7 (ejecución presupuestal de gastos).
- Permitir que el sistema a partir de la información reportada, diligencie el formato 20 que es el que la contraloría debe reportar a la auditoría general de la república.
- Realizar una comparación entre el campo compromisos y el campo pagos del formato 7 (ejecución presupuestal de gastos).
- Visualizar los distintos formatos reportados a través del navegador web y no tener que utilizar software adicional como Microsoft Excel.
- Visualizar los formatos reportados desde cualquier ubicación del mundo sin descargar archivos que utilicen software adicional pues retrasa el debido proceso de vigilancia por parte de la contraloría.

- Denegar a las diferentes entidades obligadas a reportar información cargar sus reportes finales con campos vacíos o abreviaturas como n/a, pues los formatos establecidos por legislatura obligan a que se diligencien uno a uno los campos de los formatos para el debido estudio del dinero entregado a las entidades públicas.
- Crear cuentas de acceso para cada uno de los involucrados con el sistema de auditoria bien sea entidad pública o funcionario de la contraloría.
- Llevar un control de la persona que debe diligenciar la información por parte de cada ente público.

5.3 ANALISIS

En esta fase se procede a analizar los requerimientos del cliente, clasificándolos, pues hay requerimientos que por sus características no pueden ser tratados igual. Se clasificaron de acuerdo a si son funcionales, no funcionales o si son requerimientos externos.

Requerimientos Funcionales: Son aquellos expresados anteriormente. Son los requisitos que solicito la contraloría para nuestro sistema.

Requerimientos No Funcionales: Los no funcionales son aquellos que imponen restricciones al diseño y funcionamiento del sistema.

- El ordenador utilizado para trabajar el sistema debe tener instalado el aplicativo Java Runtime Environment

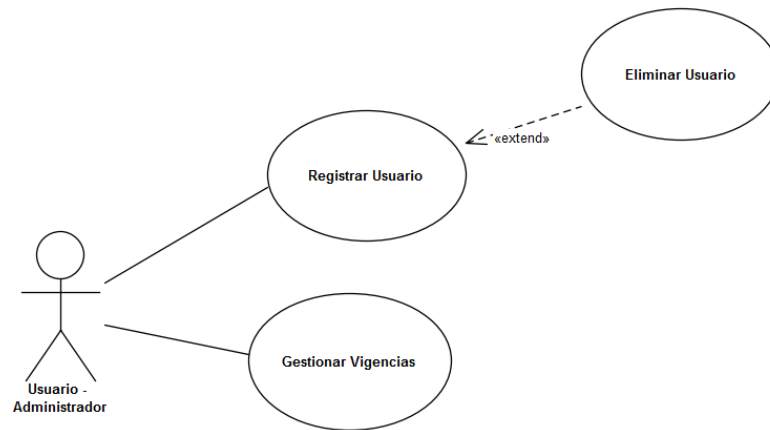
Requerimientos Externos:

- Para poder utilizar este sistema de información se hace necesario tener una conexión a internet sin importar la velocidad de transmisión. Se requiere estar conectado a internet para poder visualizar y diligenciar los formatos.

5.3.1 Diagramas de casos de uso

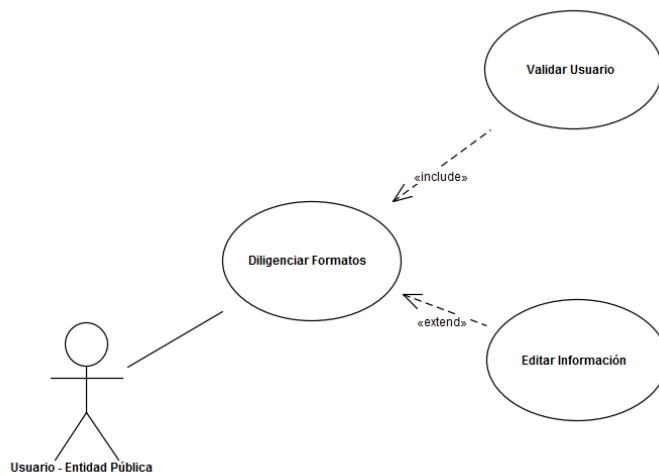
- Diagrama de casos de uso para el Administrador del sistema.

Figura 8. Caso de uso Administrador Sistema



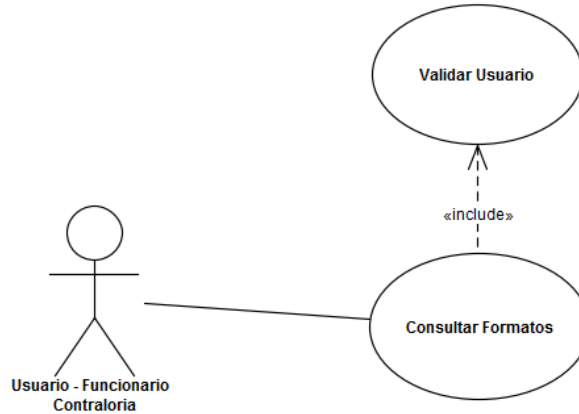
- Diagrama de casos de uso para el Usuario – Entidad Pública.

Figura 9. Caso de uso Entidad Pública



- Diagrama de casos de uso Usuario – Funcionario Contraloría.

Figura 10. Caso de uso Funcionario Contraloría



Actores Del Sistema

- ✓ Administrador
- ✓ Usuario - Funcionario Contraloría
- ✓ Usuario – Entidad Publica

CASOS DE USO

Tabla 1. Casos de Uso - Registrar Usuario

Caso de Uso:	Registrar usuario
Actores:	Administrador
Descripción:	El administrador debe utilizar este caso de uso para crear cuentas de usuario para permitir el acceso de cada entidad públicaal sistema y reportar la información requerida por la contraloría. Se habilitara una cuenta de

	administrador a cada ente público con un usuario y una contraseña.
Precondiciones:	Ninguno
Flujo Principal:	Se despliega la vista de registro en la que el administrador del sistema debe introducir la información personal del funcionario que está encargado de reportar los formatos establecidos por ley.
Pos condiciones:	Ninguna
Subflujos:	La interfaz de registro de usuario solicitará los campos como nombres, apellidos, número de cedula, teléfono, email, usuario, contraseña.
Excepciones	Se presentarán excepciones en caso de no llenar todos los campos o de introducir valores no permitidos en los distintos campos.

Tabla 2. Casos de Uso - Validar Usuario

Caso de Uso:	Validar usuario
Actores:	Usuario - Entidad Pública, Usuario – Funcionario Contraloría
Descripción:	Es necesario validar el acceso tanto de la persona encargada de diligenciar la información como del funcionario de la contraloría encargado de consultar la información reportada por las entidades públicas.
Precondiciones:	Se requiere que antes de acceder a este caso de uso se hayan registrado los usuarios en el sistema para así proceder a la validación de la persona que desea acceder al sistema.
Flujo Principal:	Se despliega la vista de insertar el nombre de usuario y la contraseña.
Pos condiciones:	Cada actor podrá dirigirse a su respectivo caso de uso. Para el usuario entidad pública se desplegara el menú en el que podrá escoger la opción diligenciar formato. Para el usuario funcionario contraloría se despliega en el menú la opción de consultar información reportada por las entidades públicas.
Subflujos:	Para el usuario entidad pública se desplegara la pantalla donde debe indicar que formatos son los que va a

	diligenciar. Para el usuario funcionario contraloría se desplegara la pantalla donde debe escoger el tipo de entidad público que desea auditar y seguidamente el ente como tal.
Excepciones	Se presentaran excepciones en el caso de introducir mal el usuario o la contraseña.

Tabla 3. Casos de Uso - Eliminar Usuario

Caso de Uso:	Eliminar usuario
Actores:	Administrador
Descripción:	Este caso de uso permite eliminar usuarios del sistema que pueden ser exfuncionarios de algún ente público o de la contraloría misma.
Precondiciones:	Ninguna
Flujo Principal:	Se despliega la vista de insertar el nombre de usuario y la contraseña.
Pos condiciones:	Ninguna
Subflujos:	Para el administrador se despliega una pantalla en la que se podrá visualizar la lista de usuarios inscritos tanto para la contraloría general de Santander como para cada ente público.
Excepciones	Ninguna

Tabla 4. Casos de Uso - Gestionar Vigencias

Caso de Uso:	Gestionar Vigencias
Actores:	Administrador
Descripción:	Este caso de uso permite controlar las fechas tanto de apertura como de cierre del plazo que por ley se da para reportar los formatos por parte de las entidades públicas.
Precondiciones:	Se requiere que este creado el ente publico
Flujo Principal:	Se despliega la pantalla en la que se puede visualizar los tipos de entes y los entes que los conforman. Igualmente se podrá visualizar el campo pertinente para la apertura o cierre de la vigencia de cada ente público.
Poscondiciones:	Ninguna
Subflujos:	Ninguno
Excepciones	Se presentan excepciones en caso de indicar mal una fecha.

Tabla 5. Casos de Uso - Diligenciar Formatos

Caso de Uso:	Diligenciar Formatos
Actores:	Usuario - Entidad Publica
Descripción:	Este caso de uso permite a la entidad pública diligenciar todos los formatos requeridos por la contraloría.
Precondiciones:	Validación del usuario entidad publica
Flujo Principal:	Se despliega una pantalla en la que

	debe escoger los formatos que desea reportar uno a uno.
Pos condiciones:	Se actualiza la información reportada por la entidad pública.
Subflujos:	Ninguno
Excepciones	Se presentan excepciones en el caso de digitar un dato erróneo en alguna casilla. Ocurrirán excepciones en el caso de haber alcanzado la fecha límite establecida en la vigencia de los formatos.

Tabla 6. Casos de Uso - Editar Información

Caso de Uso:	Editar Información
Actores:	Usuario - Entidad Publica
Descripción:	Este caso de uso permite a la entidad pública corregir errores en los campos diligenciados anteriormente.
Precondiciones:	Debe ejecutarse el caso de uso diligenciar formatos.
Flujo Principal:	Se despliega en pantalla los formatos que debe reportar esa entidad. El usuario entidad pública escoge que formato desea editar y podrá visualizar la información que digitó en la cual podrá igualmente modificar campos.
Pos condiciones:	Se actualiza la información de los formatos en el sistema.

Subflujos:	Ninguno
Excepciones	Se presentan excepciones en el caso en que la vigencia haya alcanzado la fecha límite de reporte. Por consiguiente no se podrá editar la información. En el caso de digitar mal un valor en los campos, ocurrirán excepciones también.

Tabla 7. Casos de Uso - Consultar Información

Caso de Uso:	Consultar Información
Actores:	Usuario – Funcionario Contraloría
Descripción:	Este caso de uso permite a los funcionarios de la contraloría general de Santander visualizar uno a uno los formatos reportados por cada entidad pública del departamento de Santander.
Precondiciones:	Debe haberse validado el usuario y contraseña de acceso del funcionario de la contraloría.
Flujo Principal:	Se despliega en pantalla los diferentes tipos de entes y de los entes públicos que lo conforman. El funcionario podrá seleccionar el ente público que desea consultar.
Pos condiciones:	Se cumplen las tareas propuestas por parte de la contraloría general de Santander, pues está realizando la debida revisión del patrimonio público entregado a las entidades públicas del

	departamento de Santander.
Subflujos:	Ninguno
Excepciones	Se presentan excepciones en el caso de haber digitado mal el usuario o la contraseña de acceso al sistema.

5.4 DISEÑO

En esta etapa de diseño, ya se tiene un panorama completo de lo que debe ser el sistema. Utilizamos herramientas con las cuales pudimos establecer:

- Entradas y salidas del sistema
- Base de datos a utilizar para el sistema
- Procesos necesarios para garantizar calidad en el sistema
- Interfaces de usuario
- Programación

Para el desarrollo de este proyecto era importante crear un modelo de datos, pues la descripción de la estructura de la base de datos brinda una colección de herramientas conceptuales utilizadas para describir, relacionar y restringir los datos con los que se desea trabajar.

Sabiendo modelar la información recibida de la contraloría como lo hemos hecho, pudimos identificar la metodología a utilizar para dar solución a la problemática que maneja actualmente la contraloría.

Se realizó un trabajo de investigación de todos los formatos, para conocer qué tipo de información debe contener cada uno. Este fue el primer paso para elaborar nuestro modelo de datos, pues establecimos relaciones a partir de allí entre los distintos formatos que hay que reportar.

Para el desarrollo de este software decidimos realizar la programación de interfaces de usuario en JavaServer Faces, pues es un framework muy completo para el desarrollo de aplicativos web. Posee una amplia gama de herramientas en las que el desarrollador se puede apoyar para realizar la programación. Paralelo, utilizamos el gestor de base de datos PostgreSQL, pues posee todas las herramientas que otros gestores comerciales y está escrito en código abierto.

Soporta distintos tipos de datos entre los cuales están tipo fecha, monetario, elementos gráficos, datos sobre redes, cadenas de bits, entre muchos otros.

Incluye herencia entre tablas, permite la gestión de diferentes usuarios, como también de permisos para cada uno de ellos.

Posee una gran escalabilidad, es capaz de ajustarse al número de cpus y a la cantidad de memoria que posee el sistema, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.

Para la construcción de la base de datos fue necesario seguir un proceso de normalización, el cual consiste en aplicar reglas para mantener integridad en los datos, en las relaciones que se construyen, y evitar la redundancia que se pueda generar debido a la cantidad de información que se debe albergar en la base de datos.

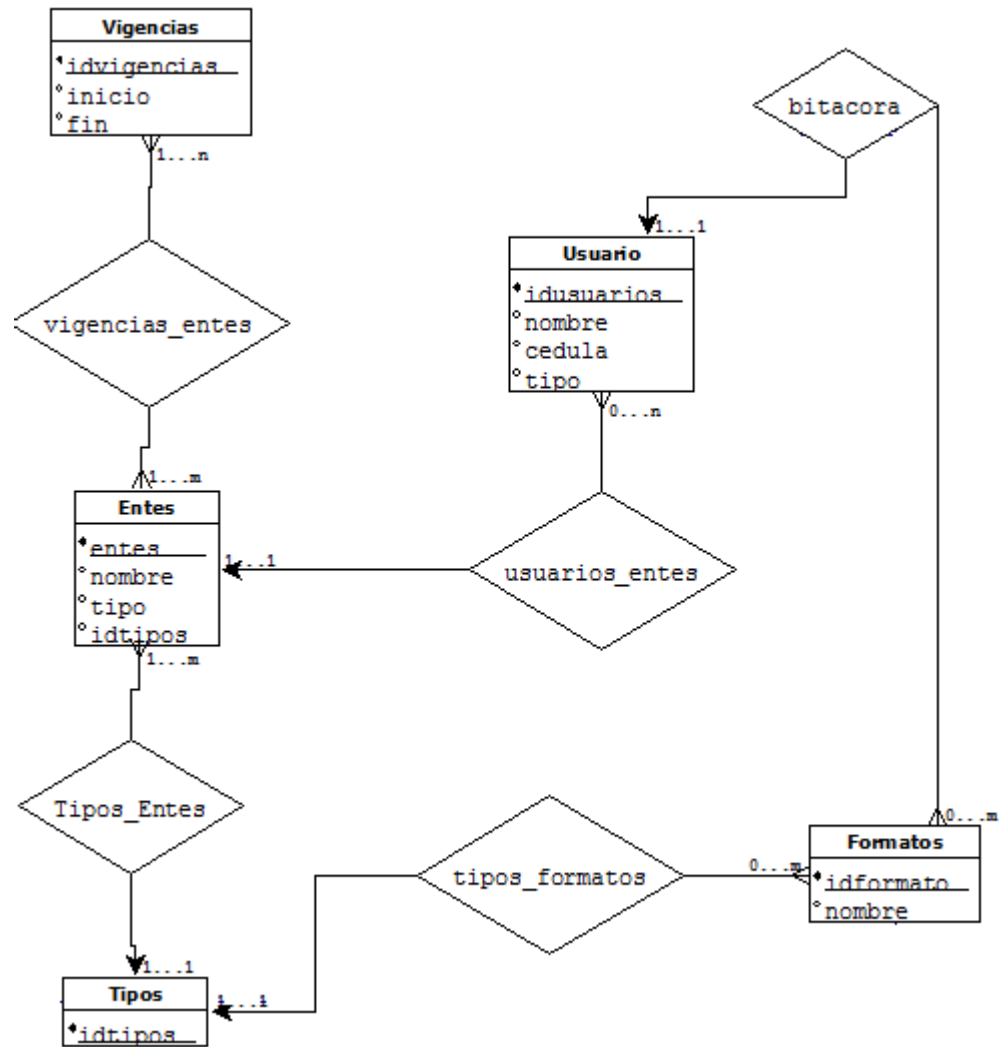
Se definieron de acuerdo al marco teórico para la normalización de las bases de datos, unas claves candidatas, una principal, una foránea para cada entidad.

Otro paso importante a desarrollar para obtener la base de datos del sistema es identificar las diferentes entidades que representan objetos a partir de la información junto con las relaciones que estas forman entre sí para darle sentido

al sistema de información. Este análisis genera un modelo entidad-relación, el cual es el preámbulo para generar el modelo de datos de la aplicación.

5.4.1 Modelo entidad – relación

Figura11. Modelo Entidad-Relación



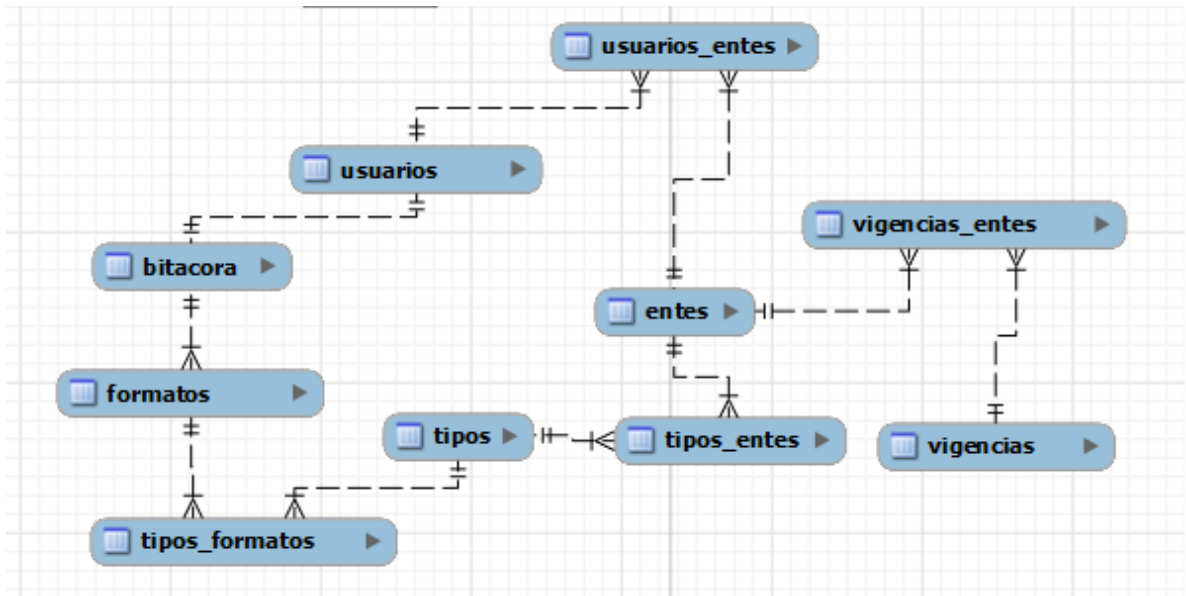
5.4.2 Modelo de datos

Según el análisis realizado a los requerimientos obtenidos, y la división por módulos que planteamos para el desarrollo de este proyecto realizamos un modelo de datos que comprende la manera en que se estructura la información en nuestro sistema. Tomamos en cuenta los aspectos más importantes dentro del procedimiento que se realiza paso a paso para auditar el patrimonio del estado. EL modelo se obtuvo a partir de un diagrama entidad relación propuesto por el equipo de desarrollo. Teniendo en cuenta las formas normales y la teoría propuesta para el tema de las bases de datos, se pudo obtener un modelo de datos final que minimiza los problemas de lógica y redundancia.

Es importante resaltar que en los nombres de los campos de las tablas se estipulo manejar letra minúscula, sin espacio, y separados con guiones de piso cuando fue necesario.

Los nombres con los que identificamos cada tabla son los mismos a los que hace relación, es decir, si estamos observando la tabla vigencias, podemos visualizar campos dentro de ella que hacen referencia a el manejo de las distintas fechas estipuladas por legislación para reportar la información que se requiere por la contraloría.

Figura 12. Modelo de Datos



5.5 IMPLEMENTACION

En esta fase comienza la codificación del proyecto, donde plasmamos lo que realizamos en la fase de requerimientos, análisis y diseño.

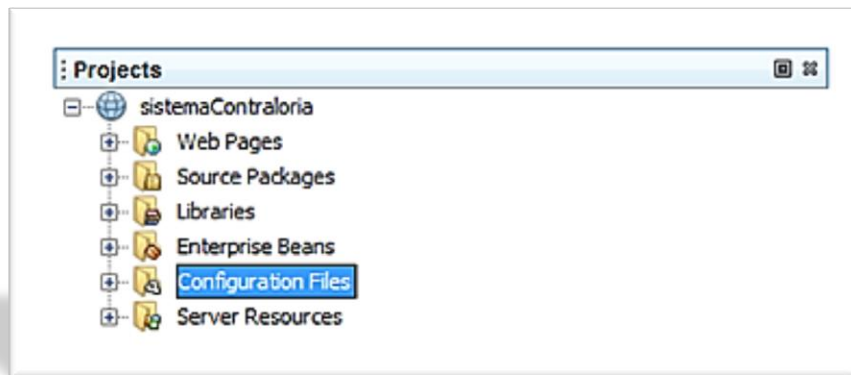
Se utilizó el framework para desarrollo de aplicaciones web JavaServer Faces, el lenguaje utilizado fue java a través de la herramienta Netbeans 7.0, con servidorGlassfish 3.1 y PostgreSQL para la gestión de la base de datos.

Netbeans y JavaServer Faces brindan variadas herramientas para el diseño de las vistas xhtml, como tablas, formularios, contenedores ejb, métodos para acceder las listas creadas con la información de la base de datos. Nuestra base de datos se llama Sistema Contraloria, en la cual se ejecutaron pruebas de almacenamiento de información verificando que no existieran errores en el almacenamiento ni incoherencias.

Se implementó el código fuente haciendo uso de prototipos así como pruebas y ensayos para corregir errores. Se crearon bibliotecas y componentes reutilizables en el desarrollo del proyecto para realizar un proceso más ágil y eficiente en la codificación.

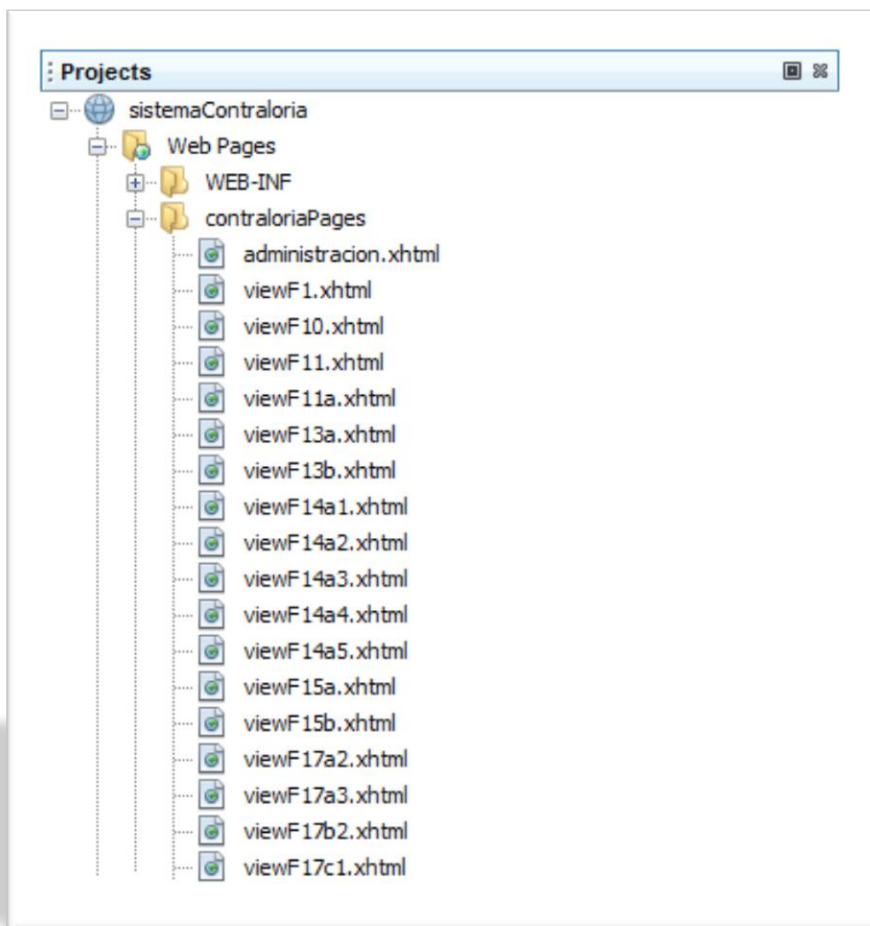
A continuación se detalla la organización de los diferentes directorios que se utilizaron para la construcción del sistema.

Figura13. Organización Sistema



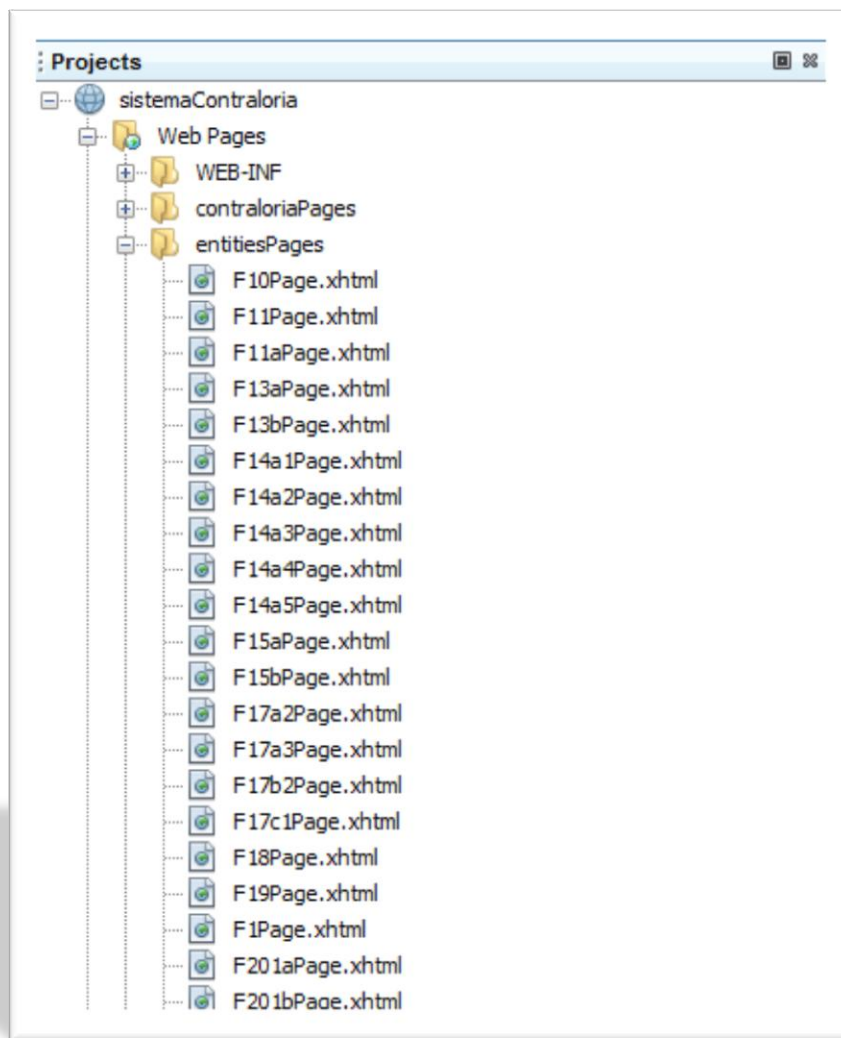
La figura permite observar las diferentes carpetas en las que se organiza la codificación del sistema para la contraloría general de Santander. La figura de un mundo es representación de que el proyecto que se realiza es un proyecto web. El gestor de desarrollo Netbeans, provee herramientas óptimas para brindar soporte al desarrollador de aplicaciones informáticas, incluyendo la organización de directorios y separación de componentes.

Figura14. Vistas Contraloría



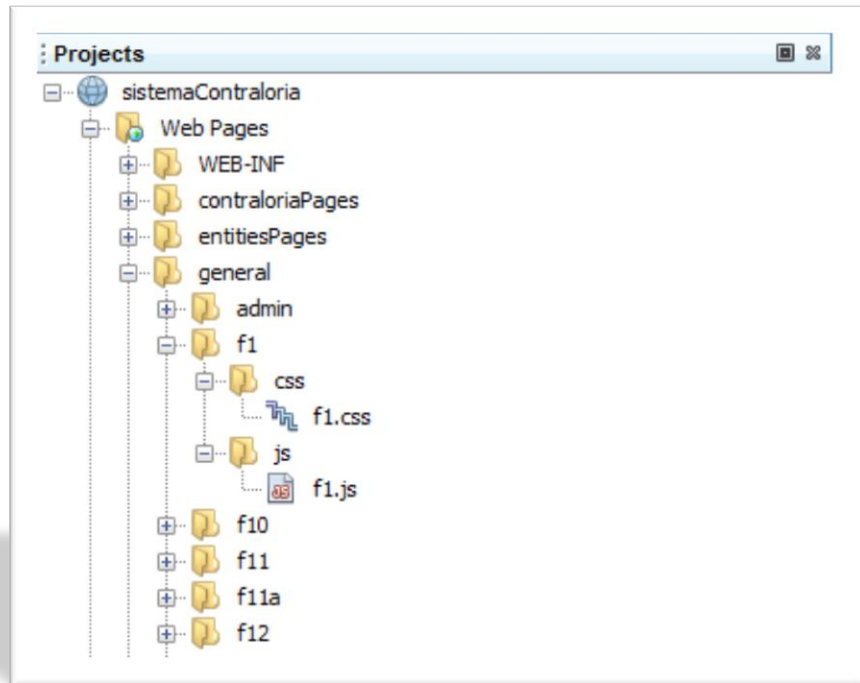
La figura No. 14 permite observar las vistas xhtml de toda la aplicación. Estas vistas están contenidas en un folder mayor llamado Web Pages. Dentro de este folder creamos otro llamado contraloriaPages que es el folder que contiene los formatos reportados por todas las entidades públicas del departamento de Santander, los cuales son objeto de estudio por parte de la contraloría.

Figura15. Vistas Entidades Públicas



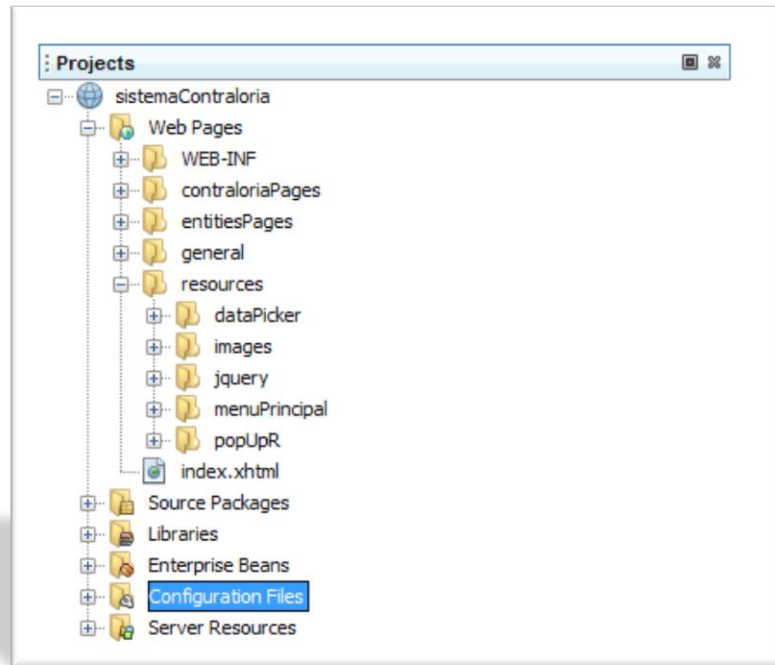
La figura 15 permite observar un folder llamado entitesPages, este folder es quien contiene las vistas que poseen los formularios de los formatos que deben reportar todos los entes públicos de Santander.

Figura16. Carpeta contenedora de código Css y Js



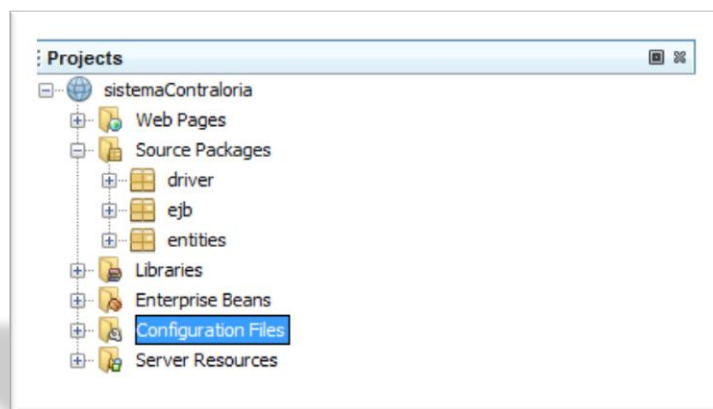
La figura 16 permite observar un folder llamado general, el cual está contenido en el folder mayor Web Pages, este folder contiene toda la parte de hojas de estilo y códigos JavaScript para cada uno de los formatos que se deben reportar. Esta es la presentación y estética del sistema desarrollado para la contraloría general de Santander.

Figura17. Carpeta contenedora de recursos de la aplicación



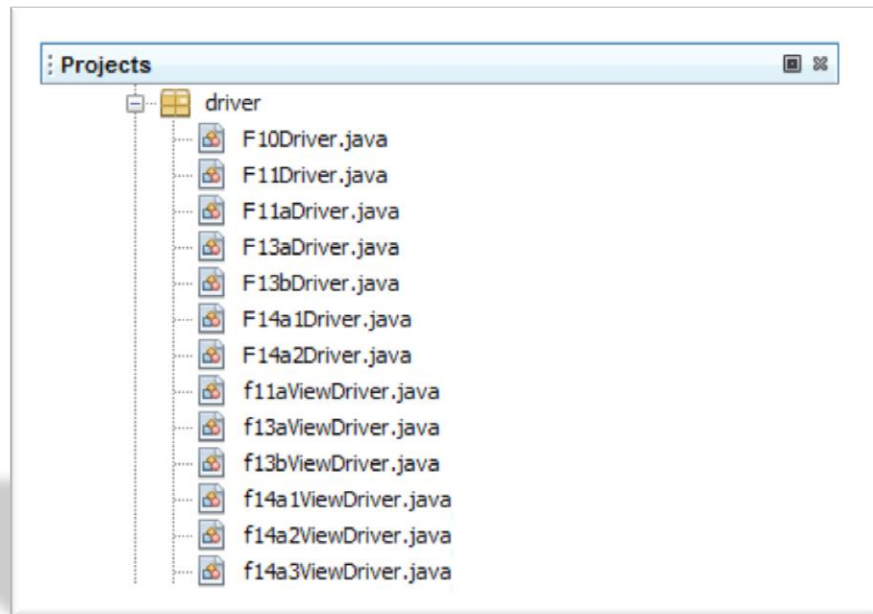
La figura 17 permite observar un folder llamado resources, el cual contiene todas las figuras, códigos JavaScript, complementos, entre otros que hemos implementado para la presentación de nuestra aplicación web.

Figura18. Carpeta contenedora la capa Lógica de Negocio y Capa de Datos



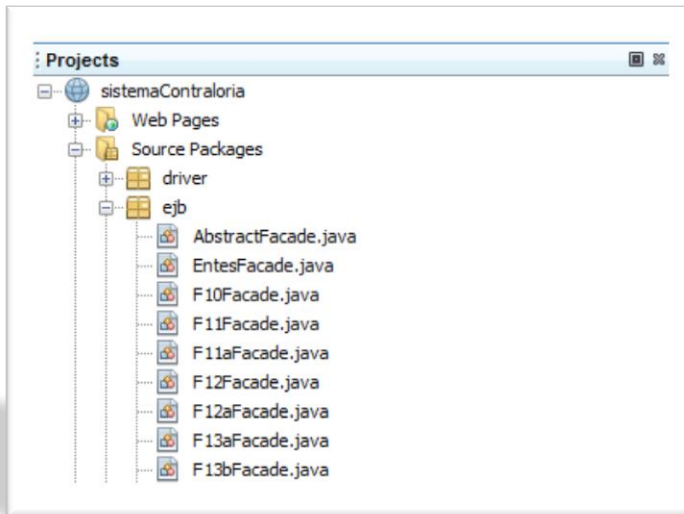
La figura 18 permite observar un folder mayor llamado Source Packages, el cual contiene la capa de lógica de negocio, capa de datos y capa de presentación de la aplicación objeto de este libro.

Figura19. Controladores Sistema



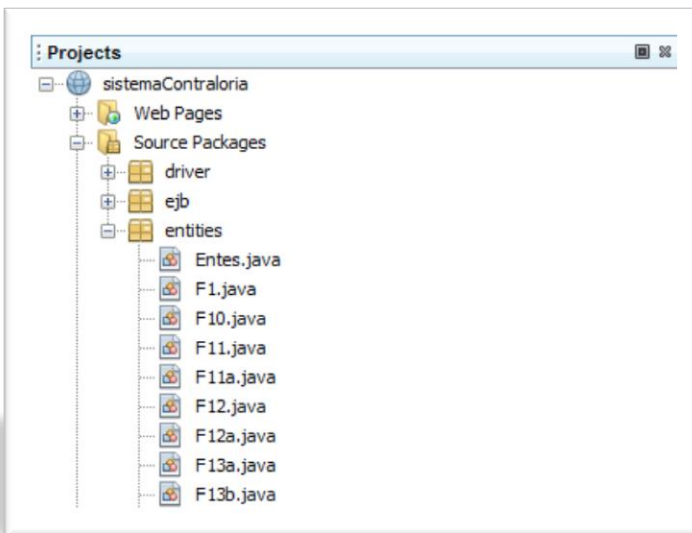
La figura 19 permite observar uno de los folder contenidos en el folder mayor Source Packages llamado driver. Este folder contiene los controladores que permiten que la capa de datos y la capa de lógica de negocio se comuniquen con las vistas xhtml.

Figura20. Contenedor Capa Lógica de Negocio



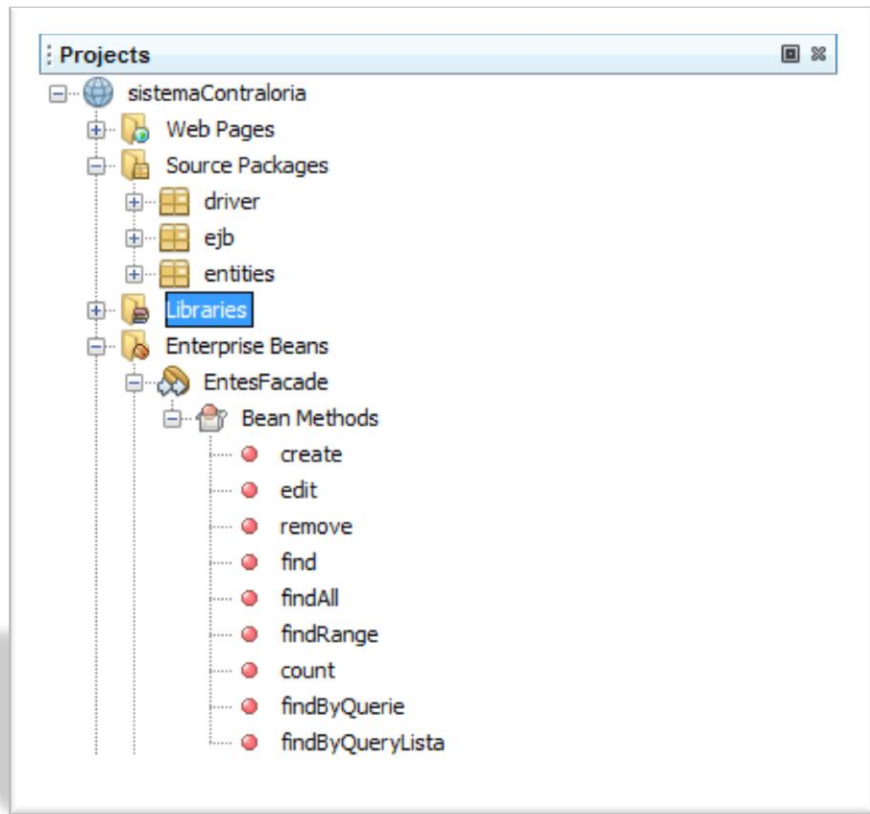
La figura 20 permite observar otro folder llamado ejb. Este folder contiene toda la capa de lógica de negocio de la aplicación que se desarrolló para este proyecto de investigación. Esta capa es la que permite intercambiar información con las capa de datos.

Figura21. Contenedor Capa de Datos de la Aplicación



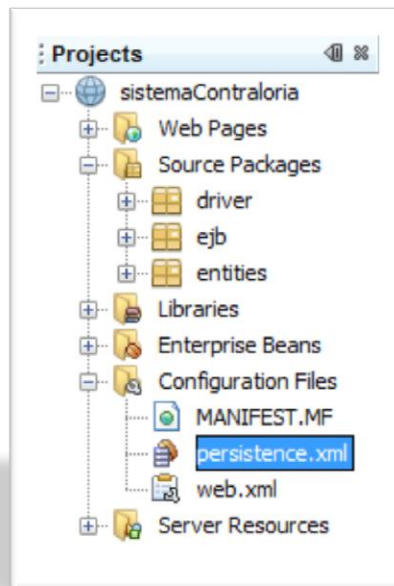
La figura 21 permite observar un folder llamado entities. Este folder contiene la capa de datos de la aplicación. La capa de datos es la encargada de relacionar la aplicación con la base de datos de esta.

Figura22. Métodos Lógica de Negocios



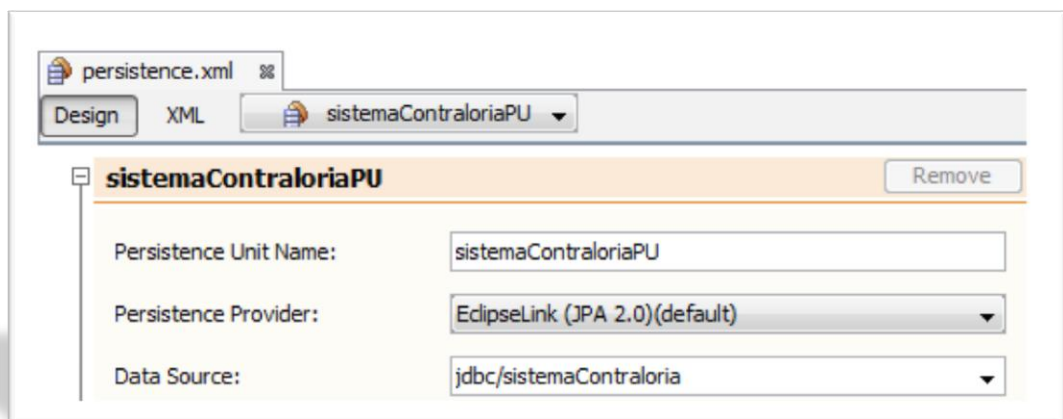
La figura 22 permite observar los diferentes métodos creados como lógica de negocio de esta aplicación. Están contenidos en un folder llamado Enterprise beans. Estos métodos están creados para todos y cada uno de los formatos a reportar por las entidades públicas. Así como también formaran parte de la parte de administración de la aplicación.

Figura23. Motor de Persistencia



La figura 23 permite observar un archivo contenido en un folder llamado Configuration Files. Este archivo se llama persistence, y es el que contiene el motor de persistencia utilizado para nuestra aplicación web. En la figura podemos apreciar el motor de persistencia escogido para nuestra aplicación.

Figura24. Motor de Persistencia



5.6 PRUEBAS

En esta fase, se ejecuta una serie de pasos que permiten verificar el cumplimiento de cada uno de los requerimientos planteados en la fase de especificación de requerimientos. Permiten encontrar fallas en el sistema, realizar ajustes, mantenimiento en caso de necesitarlo para garantizar el correcto funcionamiento. Las actividades se dividieron según el rol del usuario del sistema, administrador, auditor, y usuario entidad pública.

Para el rol de usuario entidad pública se diligenciaron los formatos correspondientes obteniendo como resultado una información consistente en la base de datos, datos reales y verídicos con ejecución de errores cuando se presentaban datos erróneos. Se denegaba el almacenamiento de la información errónea.

La interfaz de diligenciamiento de información es amigable, sencilla, consistente, lo cual permitía a la entidad pública el fácil diligenciamiento de la información en el sistema.

Para el rol de auditor y de administrador del sistema se visualizó correctamente la información reportada por los diferentes entes públicos, observando que estuviera correcto el tipo de entidad que se estaba auditando, junto con el usuario encargado y permitido para diligenciar la información. Se crearon los usuarios correctamente asignando el tipo de rol que desempeña con el sistema.

Para el desarrollo de estas pruebas tuvimos en cuenta:

- Garantizar una correcta navegación a través del sistema teniendo en cuenta los diferentes casos de uso planteados.
- Garantizar el ingreso correcto de la información a la base de datos sin corrupción o pérdida en los datos.

- Garantizar un tiempo de respuesta mínimo en el procesamiento y recuperación de la información
- Establecer sesiones de usuario de acuerdo a el rol que desempeñan en el sistema
- Garantizar la exportación de los formatos diligenciados a formatos pdf o csv.
- Garantizar el debido diligenciamiento de la información dentro de las vigencias establecidas por la contraloría general de Santander.

CONCLUSIONES Y RECOMENDACIONES

- El uso de herramientas escritas bajo licencias de software libre garantiza bajos costos en la implementación de estos y garantiza calidad.
- El desarrollo de este proyecto permite la investigación y el desarrollo de más herramientas que se pueden enlazar con este trabajo para crear más funcionalidades para la Contraloría General de Santander.
- Se puede desarrollar otro módulo para este proyecto que mida el cumplimiento del plan propuesto por los alcaldes de cada uno de los municipios que conforman el departamento de Santander.
- Hacer uso de una metodología de desarrollo para la construcción de cualquier aplicación software es necesario, pues garantiza un ordenamiento en la construcción de la herramienta. Además permite establecer y asignar tareas para la construcción de la herramienta.

BIBLIOGRAFIA

CONTRALORIA GENERAL DE SANTANDER. Disponible en internet:
<http://www.contraloriasantander.gov.co/contenido/cont_qs-contraloria.html>

HORSTMANN, Cay y GEARY, David. Core JavaServer Faces.3 ed.
Boston:Pearson Education, Inc., 2010.

JENDROCK, Eric; EVANS, Ian y GOLLAPUDI, Devika.The Java EE 6
Tutorial.[Online].[Redwood City, CA, U.S.A.]. Oracle Corporation, Julio.2011.
[Citado 20 Marzo, 2011]. Disponible en internet:
<<http://download.oracle.com/javasee/6/tutorial/doc/>>

RUMBAUGH, James; JACOBSON, Ivar y BOOCH, Grady.EL LENGUAJE
UNIFICADO DE MODELADO. MANUAL DE REFERENCIA. 2 ed. Madrid:
GRAFILLES, S.L., 2000. ISBN 84-7829-037-0.

SOMMERVILLE, Ian. INGENIERIA EL SOFTWARE. 7 ed. Madrid:Pearson
Educación, S.A., 2005. ISBN 84-7829-074-5

WEITZENFELD, Alfredo. Ingeniería Del Software Orientada A Objetos Con UML
Java e Internet. 1 ed. México: Thomson Editores, S.A, 2005.