**Servicio de Estandarización y Mejora del Proceso de Factura Electrónica en Colombia**

Arquitectura de Software

**Versión 3.0**

Control de Versiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 2013-10-31 | 0.1 | Creación del documento de vistas del sistema | Augusto César Peña Rodríguez |
| 2013-12-05 | 0.2 | QA documento Vistas del sistema | Claudia Ráquira |
| 2013-12-06 | 1.0 | Ajuste del documento de vistas del sistema | Augusto César Peña Rodríguez |
| 2014-10-15 | 2.0 | Ajustes por comentarios de DIAN | Augusto César Peña Rodríguez |
| 2015-01-07 | 3.0 | Ajustes de acuerdo a la incorporación del Componente. |  |

|  |  |  |
| --- | --- | --- |
| **Elaborado por:** | **Revisado por:** | **Aprobado por:** |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Augusto César Peña R.**  Arquitecto de Sistema  Indra | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Patricia Hernández Pulido**  Gerente Proyecto  Indra | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Ricardo Santacruz**  Líder Técnico  Indra |

Tabla de Contenido

[1. INTRODUCCIÓN 6](#_Toc408576320)

[1.1 Propósito 6](#_Toc408576321)

[1.2 Alcance 6](#_Toc408576322)

[1.3 Organización del documento 7](#_Toc408576323)

[1.4 Definiciones, Acrónimos y Abreviaturas 7](#_Toc408576324)

[1.5 Referencias 8](#_Toc408576325)

[1.6 Panorama 10](#_Toc408576326)

[2. CONTEXTO DEL PROYECTO 10](#_Toc408576327)

[2.1 Problema a Resolver 10](#_Toc408576328)

[2.2 Objetivos 11](#_Toc408576329)

[2.3 Stakeholders 11](#_Toc408576330)

[2.4 Descripción del Contexto 11](#_Toc408576331)

[2.5 Modelo de Dominio 13](#_Toc408576332)

[2.6 Aspectos Críticos 15](#_Toc408576333)

[3. CALIDAD 15](#_Toc408576334)

[3.1 Atributos de calidad 15](#_Toc408576335)

[3.1.1 Seguridad 15](#_Toc408576336)

[3.1.2 Desempeño 16](#_Toc408576337)

[3.1.3 Disponibilidad 16](#_Toc408576338)

[3.1.4 Escalabilidad 16](#_Toc408576339)

[3.2 Escenarios de Atributos de Calidad 17](#_Toc408576340)

[4. REPRESENTACIÓN ARQUITECTÓNICA 21](#_Toc408576341)

[5. METAS Y RESTRICCIONES DE LA ARQUITECTURA 22](#_Toc408576342)

[5.1 Metas 22](#_Toc408576343)

[5.2 Restricciones 23](#_Toc408576344)

[6. PERSPECTIVA DE CASOS DE USO 23](#_Toc408576345)

[6.1 Diagrama de Casos de Uso - Registro de Participantes 24](#_Toc408576346)

[6.2 Diagrama de Casos de Uso - Factura 25](#_Toc408576347)

[6.3 Diagrama de Casos de Uso - Consulta y Parametrización 26](#_Toc408576348)

[7. PERSPECTIVA LÓGICA 27](#_Toc408576349)

[7.1 Componentes del sistema 27](#_Toc408576350)

[7.2 Paquetes de Diseño 30](#_Toc408576351)

[7.3 Clases de Diseño Significativo para la Arquitectura 34](#_Toc408576352)

[7.4 Realizaciones de Casos de Uso 39](#_Toc408576353)

[8. PERSPECTIVA DE PROCESOS 44](#_Toc408576354)

[9. PERSPECTIVA DE DESPLIEGUE 49](#_Toc408576355)

[10. PERSPECTIVA DE IMPLEMENTACIÓN 53](#_Toc408576356)

[10.1 Panorama 53](#_Toc408576357)

[10.2 Niveles 56](#_Toc408576358)

[11. PERSPECTIVA DE DATOS 62](#_Toc408576359)

**Lista de Figuras**

[Figura 1. Diagrama de Contexto 12](#_Toc408576360)

[Figura 2. Modelo de dominio 14](#_Toc408576361)

[Figura 3. Diagrama de casos de Uso - Registro de participantes 24](#_Toc408576362)

[Figura 4. Diagrama de Casos de Uso - Factura 25](#_Toc408576363)

[Figura 5. Diagrama de Casos de Uso - Consulta y Parametrización 26](#_Toc408576364)

[Figura 6. Diagrama de Componentes 28](#_Toc408576365)

[Figura 7. Diagrama de Paquetes Catálogo 31](#_Toc408576366)

[Figura 8. Diagrama de paquetes FacturaIntegracion 32](#_Toc408576367)

[Figura 9. Diagrama de paquetes Componente 33](#_Toc408576368)

[Figura 10. Diagrama de clases - Recepción de facturas 36](#_Toc408576369)

[Figura 11. Diagrama de clases - Procesamiento de facturas 38](#_Toc408576370)

[Figura 12. Diagrama de secuencia - Recibir Factura 40](#_Toc408576371)

[Figura 13. Diagrama de secuencia – Recibir archivos con facturas por FTP 41](#_Toc408576372)

[Figura 14. Diagrama de secuencia – Recibir factura por FTP 42](#_Toc408576373)

[Figura 15. Diagrama de secuencia - Verificar Factura Electrónica 43](#_Toc408576374)

[Figura 16. Diagrama de proceso de recepción de factura por web service 44](#_Toc408576375)

[Figura 17. Diagrama de proceso de recepción por FTP 45](#_Toc408576376)

[Figura 18. Diagrama de proceso de verificaciones de facturas nacionales 46](#_Toc408576377)

[Figura 19. Diagrama de proceso de verificaciones de facturas internacionales 48](#_Toc408576378)

[Figura 20. Diagrama de despliegue de Catálogo 50](#_Toc408576379)

[Figura 21. Diagrama de despliegue 52](#_Toc408576380)

[Figura 22. Panorama de implementación de Catálogo 54](#_Toc408576381)

[Figura 23. Panorama de implementación de Componente 55](#_Toc408576382)

[Figura 24. Diagrama de niveles de Catálogo 57](#_Toc408576383)

[Figura 25. Diagrama de niveles de Componente 60](#_Toc408576384)

[Figura 26. Diagrama conceptual de entidades de datos de Catálogo 62](#_Toc408576385)

[Figura 27. Diagrama conceptual de entidades de datos de Componente 63](#_Toc408576386)

# INTRODUCCIÓN

La definición de una solución arquitectónica a nivel de software, se fundamenta en la necesidad de satisfacer, en etapas tempranas, atributos de calidad como desempeño, seguridad, mantenibilidad, etc., creando una estructura de alto nivel que también sirva como guía durante el desarrollo del sistema. La creación de esta arquitectura, genera las bases de los sistemas de información misionales del negocio y debe obedecer a lineamientos trazados de una arquitectura de negocio.

Esta definición de la solución arquitectónica para el Servicio de Estandarización y Mejora del Proceso de Factura Electrónica en Colombia, está enmarcada en la arquitectura MUISCA propia de la DIAN y en la utilización de los subsistemas que esta provee.

## Propósito

Este documento provee un panorama comprensible de la Arquitectura del Sistema, utilizando un número de perspectivas diferentes que resaltan diferentes aspectos del sistema. La intención del documento es capturar y convenir las principales decisiones estructurales que se han tomado sobre el sistema.

Esta sección define el rol o propósito del Documento de Arquitectura de Software, dentro del conjunto de los documentos del proyecto y es complementado por otros documentos como la especificación de casos de uso, especificaciones suplementarias y todo aquel documento que defina los requerimientos funcionales y no funcionales. Define igualmente la audiencia del documento a nivel técnico, los arquitectos de soluciones INDRA, DIAN, líderes técnicos, de desarrollo, desarrolladores y todos los roles que intervengan en la construcción del sistema, con una indicación de cómo se espera que utilicen el documento.

## Alcance

El alcance del documento de arquitectura pretende resolver e incluir la solución a los requerimientos funcionales y técnicos del sistema a construir, apoyándose en la creación de diferentes vistas de esta, representadas a través de diagramas. Todas las definiciones del documento de arquitectura deberán constituir los lineamientos para la construcción del sistemas y ser utilizado por roles como el líder de desarrollo, líderes técnicos, desarrolladores, líder de pruebas y todo aquel que esté involucrado en la construcción del sistema.

Se incluye la relación de utilización o integración con la arquitectura MUISCA y sus respectivos subsistemas, desarrollando la utilización de sus servicios.

## Organización del documento

El presente documento se organiza con base en la plantilla propuesta por INDRA, la cual se adaptó a las necesidades del proyecto de Fábrica de Software para DIAN.

Se presenta inicialmente una descripción del documento, en la que se incluyen aspectos como su propósito, la terminología y definiciones requeridas para su desarrollo y entendimiento, y otros documentos relevantes.

En la segunda parte se listan las generalidades del proyecto como el detalle del problema a resolver, los objetivos del proyecto, sus stakeholders y el contexto de la solución.

En la tercera sección, a la que se le ha dado el nombre de Calidad, se describen los atributos de calidad deseables y priorizados que rigen las decisiones de diseño tomadas en las diferentes perspectivas de la arquitectura de software.

En la cuarta sección el proceso de análisis utilizado para la definición de la Arquitectura de Software del sistema a implementar, adicionalmente se provee un diagrama conceptual que muestra claramente cada uno de los módulos/subsistemas a tener en cuenta dentro de la definición arquitectural.

En la quinta sección se detallan las metas y restricciones tanto tecnológicas como de negocio para el proyecto.

En la sexta, séptima, octava, novena y décima sección se desarrolla con más detalle la propuesta de la arquitectura del sistema y se muestran los modelos arquitecturales de cada punto de vista.

## Definiciones, Acrónimos y Abreviaturas

Ver glosario del proyecto. Archivo GLOS\_NDFACE002\_Glosario ubicado en la siguiente ruta:

http://gdocumental.dian.gov.co/cliente/DIAN/SGT/Documentos%20Cliente/Fabrica%20de%20Software/Equipo%20Por%20Demanda/Nuevos%20Desarrollos%20(ND)/NDFACE002/03.%20Ejecuci%C3%B3n/Requerimientos/1.%20Entregables/GLOS\_NDFACE002\_Glosario.doc

Adicionalmente, a continuación se incluye la definición de algunos términos manejados en el documento.

* MDB: Sigla para Message Driven Bean. Es la tecnología en Java para el manejo de mensajería por medio de JMS. Permite la orquestación y ejecución de procesos asíncronos.
* FTP: Protocolo de transferencia de archivos (File Transfer Protocol) entre un cliente y un servidor.
* JMS: Java Message Service. Tecnología Java para el manejo de mensajería.
* Quartz: Librería Java open source que permite la programación de procesos por medio de frecuencias o periodicidades de tiempo.
* SOAP: Simple Object Access Protocol. Protocolo basado en XML y usado para el intercambio de mensajes en los web services.
* Spring: Framework open source para aplicaciones empresariales bajo Java. Contiene un conjunto de componentes que facilitan la implementación o uso de diferentes tecnologías.
* XML: Siglas para eXtensible Markup Language. Lenguaje estándar independiente de plataforma.
* XSD: XML Schema Definition. Lenguaje estándar para definir la estructura de documentos XML.

## Referencias

Se tomaron como referencia los siguientes documentos:

|  |  |
| --- | --- |
| **Documento** | **Ubicación (Ruta)** |
| Documento del Plan de proyecto "FSDIAN\_NDFACE002\_PlanProyecto.docx". | http://gdocumental.dian.gov.co/cliente/DIAN/SGT/Documentos%20Cliente/Fabrica%20de%20Software/Equipo%20Por%20Demanda/Nuevos%20Desarrollos%20(ND)/NDFACE002/02.%20Planeaci%C3%B3n/FSDIAN\_NDFACE002\_PlanProyecto.docx |
| Glosario del proyecto  “GLOS\_NDFACE002\_Glosario.docx” | http://gdocumental.dian.gov.co/cliente/DIAN/SGT/Documentos%20Cliente/Fabrica%20de%20Software/Equipo%20Por%20Demanda/Nuevos%20Desarrollos%20(ND)/NDFACE002/03.%20Ejecuci%C3%B3n/Requerimientos/1.%20Entregables/GLOS\_NDFACE002\_Glosario.doc |
| Documento Visión del sistema "VISI\_NDFACE002\_VisionSistema.docx". | http://gdocumental.dian.gov.co/cliente/DIAN/SGT/Documentos%20Cliente/Fabrica%20de%20Software/Equipo%20Por%20Demanda/Nuevos%20Desarrollos%20(ND)/NDFACE002/03.%20Ejecuci%C3%B3n/Requerimientos/1.%20Entregables/VISI\_NDFACE002\_VisionSistema.doc |
| Requerimientos funcionales "REFU\_NDFACE002\_RequerimientosFuncionales.doc" | http://gdocumental.dian.gov.co/cliente/DIAN/SGT/Documentos%20Cliente/Fabrica%20de%20Software/Equipo%20Por%20Demanda/Nuevos%20Desarrollos%20(ND)/NDFACE002/03.%20Ejecuci%C3%B3n/Requerimientos/1.%20Entregables/REFU\_NDFACE002\_RequerimientosFuncionales.doc |
| Requerimientos suplementarios "REFU\_NDFACE002\_RequerimientosSuplementarios.doc" | http://gdocumental.dian.gov.co/cliente/DIAN/SGT/Documentos%20Cliente/Fabrica%20de%20Software/Equipo%20Por%20Demanda/Nuevos%20Desarrollos%20(ND)/NDFACE002/03.%20Ejecuci%C3%B3n/Requerimientos/1.%20Entregables/RESU\_NDFACE002\_RequerimientosSuplementarios.doc |
| Especificaciones de casos de uso. | http://gdocumental.dian.gov.co/cliente/DIAN/SGT/Documentos%20Cliente/Forms/AllItems.aspx?RootFolder=%2Fcliente%2FDIAN%2FSGT%2FDocumentos%20Cliente%2FFabrica%20de%20Software%2FEquipo%20Por%20Demanda%2FNuevos%20Desarrollos%20%28ND%29%2FNDFACE002%2F03%2E%20Ejecuci%C3%B3n%2FRequerimientos%2F1%2E%20Entregables&FolderCTID=0x012000E9C5C12E05C25D449460C04E33E0D02A&View=%7B2BC69FA0-A5BF-4F6D-A6DE-8259CC522515%7D |
| Documento de consultoría "FSDIAN\_NDFACE002\_PoliticaFirmaElectronica-v1.2.docx " | http://gdocumental.dian.gov.co/cliente/DIAN/SGT/Documentos%20Cliente/Fabrica%20de%20Software/Equipo%20Por%20Demanda/Nuevos%20Desarrollos%20(ND)/NDFACE002/03.%20Ejecuci%C3%B3n/Entregables%20Consultoria/Politica%20Firma/FSDIAN\_NDFACE002\_PoliticaFirmaElectronica-v1.2.docx |
| Documento de consultoría "FSDIAN\_NDFACE002\_RecomendacionesDefinicionFormatoFacturaElectronica-v1.2.docx " | http://gdocumental.dian.gov.co/cliente/DIAN/SGT/Documentos%20Cliente/Fabrica%20de%20Software/Equipo%20Por%20Demanda/Nuevos%20Desarrollos%20(ND)/NDFACE002/03.%20Ejecuci%C3%B3n/Entregables%20Consultoria/Formato%20Factura/FSDIAN\_NDFACE002\_RecomendacionesDefinicionFormatoFacturaElectronica-v1.2.docx |
| Documento de consultoría "FSDIAN\_NDFACE002\_Req\_Tecnicos\_Funcionales\_Ofact\_Prvtec v1.2.docx " | http://gdocumental.dian.gov.co/cliente/DIAN/SGT/Documentos%20Cliente/Fabrica%20de%20Software/Equipo%20Por%20Demanda/Nuevos%20Desarrollos%20(ND)/NDFACE002/03.%20Ejecuci%C3%B3n/Entregables%20Consultoria/Req\_Tecnicos\_Funcionales\_Ofact\_Prvtec/FSDIAN\_NDFACE002\_Req\_Tecnicos\_Funcionales\_Ofact\_Prvtec%20v1.2.docx |

Tabla . Referencias

## Panorama

Este documento inicia con el contexto del proyecto y los escenarios de calidad que rigen la definición de la arquitectura del software. Luego muestra la representación de la solución arquitectónica, iniciando con sus metas y restricciones. Se definen las vistas del sistema, lógica, procesos, despliegue e implementación, sumando una vista de casos de uso o funcional y una vista de datos.

# CONTEXTO DEL PROYECTO

## Problema a Resolver

Actualmente la DIAN cuenta con un mecanismo de intercambio y almacenamiento de las Facturas Electrónicas a través de un Sistema de Transferencia de Archivos FTP en el que dichas facturas son recibidas en paquetes, sin tener una discriminación de las mismas ni validaciones particulares de recepción ni de contenido.

La información de los participantes que intervienen en el proceso de Facturación Electrónica no se encuentra centralizada y los acuerdos particulares con relación al intercambio de las facturas, cuya manifestación explícita no se encuentra registrada dentro de los Sistemas DIAN, no es de conocimiento por parte de la entidad.

Por otro lado, no se tiene un control de las características y condiciones particulares de los Proveedores del Servicio de Facturación electrónica seleccionados y los Software utilizados por los Obligados a Facturar para la emisión de las Facturas Electrónicas.

El modelo de facturación que existe actualmente, no cuenta con un formato estándar claramente definido, ni tampoco con un mecanismo a través del cual sea posible garantizar la integridad y la autenticidad de las diferentes facturas emitidas, lo que hace difícil el intercambio de las mismas.

Las características expuestas anteriormente impiden la masificación del modelo de facturación electrónica en Colombia y dificultan el control del mismo por parte de la DIAN, haciendo que los procesos tributarios continúen siendo complejos aumentando la evasión del fraude fiscal.

## Objetivos

* Diseñar la arquitectura del sistema de Facturación Electrónica.
* Identificar las restricciones tanto tecnológicas como de negocio para el proyecto.
* Identificar y priorizar los escenarios de calidad de acuerdo a los motivadores de negocio, restricciones y requerimientos no funcionales.
* Proveer diferentes vistas de la arquitectura que sean claras y relevantes para cada stakeholder del proyecto.

## Stakeholders

En esta sección se definen los Stakeholders más importantes dentro del ciclo de vida del proyecto, los cuales son todas aquellas personas u organizaciones que afectan o son afectadas por el proyecto, ya sea de forma positiva o negativa, para los cuales se describen sus necesidades y expectativas.

| **Nombre** | **Descripción** |
| --- | --- |
| Gerente de proyecto DIAN - Indra | Asegurar la planeación, ejecución y puesta en marcha de los objetivos del proyecto. |
| Coordinador del proyecto Mejora y masificación de la dirección general | Liderar y orientar a los interesados en la correcta toma de decisiones y planteamientos según el resultado de los entregables. |
| Dirección de Gestión de ingresos | Dirigir y plantear en materia de administración tributaria y otros ingresos el correcto enfoque del alcance del Proyecto de Facturación electrónica. |
| Técnica aduanera | Plantear las directrices y el direccionamiento de los flujos de información para encaminar el proyecto de forma adecuada en el ámbito de las exportaciones e importaciones. |
| Dirección de Gestión Fiscal | Determinar la afectación que surte el proyecto desde el punto de vista patrimonial y fiscal. |
| Dirección de Gestión Jurídica | Asesorar a los interesados del proyecto en la formulación, coordinación y ejecución de políticas, planes y estrategias, dentro del ámbito jurídico Institucional. |

Tabla . Stakeholders

## Descripción del Contexto

El siguiente diagrama pretende dar una visión global del Sistema de Facturación Electrónica y sus interacciones con elementos internos y externos a DIAN. Las interacciones internas con subsistemas Muisca se detallan más adelante.



Figura . Diagrama de Contexto

El sistema de Factura Electrónica está compuesto por 3 subsistemas, los cuales se describen a continuación:

* Catálogo de Participantes: Aplicación Web bajo el modelo Muisca que soporta las funcionalidades relacionadas con la administración de la información relacionada con los diferentes actores o participantes en la facturación electrónica.
* Componente DIAN: Aplicación encargada de la recepción y verificación de facturas electrónicas. Esta aplicación no está estructurada como las demás aplicaciones bajo el modelo Muisca.
* Integración: Aplicación bajo el modelo Muisca que sirve de puente entre el Componente DIAN y los demás subsistemas Muisca incluyendo el Catálogo.

Inicialmente los participantes ingresan al Catálogo para identificarse como Obligado a Facturar, Facturador Electrónico Voluntario, Proveedor Tecnológico, Adquirente Electrónico o Adquirente Internacional, ingresando la información requerida. Este registro es necesario para que puedan ser habilitados o autorizados para operar dentro del sistema, teniendo en cuenta el manejo de estados para cada tipo de participantes, de acuerdo a unas reglas y requisitos de cumplimiento para su correspondiente habilitación y autorización.

Una vez el participante ha sido autorizado dentro del sistema, podrá acceder y realizar las operaciones de registro, activación, inactivación y verificación de cumplimiento de requisitos sobre los Software que utiliza u ofrece para la emisión de facturas electrónicas, esto de acuerdo al tipo de participante si es Obligado a Facturar o Proveedor Tecnológico. Cuando el Software ha sido previamente registrado y activado, el participante realizará la respectiva asociación del Software de tal manera que pueda utilizarlo para sus propias transacciones de facturación u ofrecerlo a otros facturadores electrónicos.

Luego, por medio del Componente DIAN re realiza la recepción de facturas electrónicas emitidas por los participantes, a través del Software que ha sido previamente activado y autorizado ante la DIAN**.** La factura recibida es procesada ejecutando una serie de verificaciones en cuanto a estructura y contenido. Cuando la factura ha sido validada se efectuará el registro de la verificación realizada de tal forma que quede disponible esta información de transacciones de facturación para posteriores consultas por parte de los participantes.

El Componente DIAN provee dos mecanismos para recibir las facturas electrónicas, los cuales son: un web service que recibe una factura por cada mensaje SOAP, y un proceso que monitorea una ruta configurable a un sistema de archivos donde se pueden subir archivos comprimidos en formato ZIP con lotes de facturas electrónicas. En ambos casos las facturas electrónicas deben estar en formato XML bajo una estructura definida.

## Modelo de Dominio

En esta sección se describe el modelo de dominio de la solución. Este modelo muestra las entidades de negocio significativas, sus atributos y relaciones.



Figura . Modelo de dominio

Un participante puede ser un obligado a facturar o un adquirente, los cuales poseen registro en el RUT. El tipo de participante define el estado del participante en un periodo de fechas determinado. Las verificaciones del tipo de participante hacen parte del Catálogo.

El participante usa un software registrado en el Catálogo, el cual tiene un estado y está vinculado a un ambiente de software. El ambiente de software indica si el software pertenece a un ambiente de habilitación o a un ambiente de operación en producción.

La factura se registra con la relación del ambiente de software que la envía y almacena los datos propios de la misma como el tipo de moneda, el valor, e impuestos. La factura es verificada por medio del Componente DIAN.

## Aspectos Críticos

En esta sección se describe los aspectos críticos identificados en el sistema a nivel de requerimientos.

* La recepción de las facturas es una funcionalidad de frente al usuario final y punto de inicio del proceso de verificación de facturas.
* Almacenamiento de las facturas recibidas y el resultado de las verificaciones de las mismas.
* Integración de subsistemas Muisca que son requeridos para las verificaciones de las facturas electrónicas.

# CALIDAD

En esta sección se presentan los atributos de calidad y escenarios de calidad relevantes para el sistema.

## Atributos de calidad

Los atributos de calidad definen las características que debe tener el sistema, de forma que se contextualiza cada atributo de calidad respecto a los requerimientos del sistema. Inicialmente, en esta sección se identifican los atributos de calidad a tener en cuenta, dándoles un entendimiento bajo el contexto del proyecto; luego se describen los escenarios de calidad relacionados con los atributos de calidad, en los cuales se establecen las metas u objetivos medibles que guían el diseño de la arquitectura de software del sistema.

### Seguridad

El sistema de Factura Electrónica requiere la capacidad de controlar, monitorear y auditar de forma confiable, las acciones relacionadas al personal autorizado dentro de la funcionalidad del sistema, además de la administración de los datos almacenados orientada hacia la integridad y conservación de la información.

Se requiere de mecanismos de seguridad que permitan la autenticación y autorización de usuarios en el sistema. Primero se debe garantizar que el acceso de los usuarios al sistema sea concedido si y solo si el proceso de autenticación ha sido exitoso; y luego, se debe garantizar que el acceso a la información y funcionalidades del sistema a los usuarios sea restringido de acuerdo a sus roles.

Se deben proporcionar los mecanismos para que durante la ejecución de la aplicación no se presenten problemas de suplantación de identidad o que un funcionario pueda realizar una acción sobre el sistema que no debería estar en posibilidad de hacer.

### Desempeño

#### Respecto al desempeño se consideran dos aspectos relevantes en el sistema: concurrencia y utilización de recursos.

#### Concurrencia: Durante una ventana de tiempo, se realizan múltiples procesamientos asociados a la administración de la información de participantes y recibo de facturas de forma individual y masiva. Los usuarios de acuerdo a los roles que desempeñan dentro de la organización, deben poder utilizar la funcionalidad requerida, visualizando o interactuando con sus respectivas interfaces. El acceso de diferentes usuarios a una funcionalidad del sistema en el mismo momento es una característica requerida en el sistema.

#### Utilización de Recursos: Se considera muy valioso la capacidad de procesamiento del hardware sobre el que se desplegará el sistema, por lo que se deben considerar estrategias para realizar procesamiento en paralelo, de tal forma que permitan optimizar el uso del recurso de procesamiento. Esto para aplicar las reglas de negocio a los archivos XML, correspondientes a facturas electrónicas que se reciban, y generar respuesta de aprobación o desaprobación a los participantes.

### Disponibilidad

Este atributo indica el nivel o grado en que un sistema o componente está operativo y accesible para su uso, lo cual es requerido en el sistema principalmente para la funcionalidad de recepción de facturas electrónicas.

Se deben definir estrategias de alta disponibilidad con elementos de software o hardware redundantes y/o en cluster, para reducir los puntos únicos de falla y los efectos en cadena que puede causar una falla en un componente.

### Escalabilidad

La escalabilidad se puede llevar a cabo mediante dos estrategias: aumentar capacidad de procesamiento en los nodos donde está desplegado el sistema (escalabilidad vertical) o agregar más nodos para desplegar el sistema en ellos (escalabilidad horizontal).

Se debe tener en cuenta que el escalamiento vertical es más restringido que el escalamiento horizontal, ya que hay limitaciones en la cantidad de memoria que se puede asignar a la máquina virtual de Java, incluso puede perjudicar el desempeño porque aumenta el tiempo del garbage collector.

Por lo mencionado anteriormente, el esfuerzo en el diseño del sistema se enfocará en el aprovechamiento del escalamiento horizontal, con el fin de que el sistema se adapte al crecimiento esperado de participantes, representado en el aumento de la información de facturación que debe procesar.

## Escenarios de Atributos de Calidad

En los escenarios de calidad se presentan los casos concretos que se plantean respecto a los atributos de calidad, definiendo metas u objetivos medibles.

A continuación se presentan los principales escenarios de calidad identificados y priorizados:

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| **ESCA001 - Verificación de la autenticidad de origen de la información y no repudio.** | El origen de todos los archivos recibidos por el sistema, debe ser comprobado utilizando mecanismos de firma digital. |
| **Atributo de Calidad** | Seguridad. |
| **Justificación** | La autenticidad del origen de las facturas electrónicas debe confirmarse. |
| **Prioridad** | Alta. |
| **Dificultad** | Alta. |
| **Fuente** | Sistema externo, tercer proveedor. |
| **Estimulo** | Envío de facturas electrónicas en formato XML. |
| **Artefacto** | Componente DIAN. |
| **Ambiente** | Pruebas de aceptación en operación normal. |
| **Respuesta** | Resultado exitoso o no de la validación del origen de la información. |
| **Medida** | El 100% de los archivos recibidos por actores externos al sistema debe ser validado en cuanto a su origen. |

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| **ESCA002 - Presentación de opciones para un usuario final o funcionario.** | Un funcionario o usuario final entra al sistema y solo tiene acceso a las funcionalidades, según roles previamente asignados. |
| **Atributo de Calidad** | Seguridad. |
| **Justificación** | Los usuarios solo deben tener acceso a las funcionalidades para las cuales hayan sido autorizados. |
| **Prioridad** | Alta. |
| **Dificultad** | Media. |
| **Fuente** | Funcionarios DIAN, usuarios externos a DIAN. |
| **Estimulo** | Ingreso a funcionalidades. |
| **Artefacto** | Sistema. |
| **Ambiente** | Pruebas de aceptación en operación normal. |
| **Respuesta** | El sistema solo permite utilizar funcionalidades preestablecidas por rol. |
| **Medida** | El funcionario o usuario externo tiene autorización de utilizar todas las opciones que le muestra el sistema, según su perfil; y cualquier intento de ingresar a una funcionalidad que no corresponda a su perfil será bloqueado por el sistema. |

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| **ESCA003 - Disponibilidad del sistema.** | El sistema debe estar operativo garantizando que se responden las peticiones de los usuarios. |
| **Atributo de Calidad** | Disponibilidad. |
| **Justificación** | Tener operativa la funcionalidad de recepción de facturas para atender a los usuarios. |
| **Prioridad** | Alta. |
| **Dificultad** | Media. |
| **Fuente** | Participante, Operador tecnológico. |
| **Estimulo** | Envío de facturas electrónicas. |
| **Artefacto** | Componente DIAN. |
| **Ambiente** | Pruebas de aceptación en operación normal. |
| **Respuesta** | Se reciben las facturas y se valida su estructura. |
| **Medida** | El componente presenta una disponibilidad del 99.5%, atendiendo todas las peticiones de envío de facturas durante 365 días, en horario 7x24. Esto indica que durante 365 días de operación del sistema, se podría presentar hasta 2 días (48 horas) en el cual no se encuentra operativo, sin tener en cuenta las ventanas de tiempo de mantenimiento del sistema, software base o hardware. |

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| **ESCA004 - Recibir archivos XML de forma masiva.** | Se deben recibir archivos XML masivamente, identificando el éxito o fallo de la recepción de dichos archivos. |
| **Atributo de Calidad** | Desempeño |
| **Justificación** | Este escenario está orientado a garantizar que el sistema soporte la recepción de las facturas electrónicas de forma masiva. |
| **Prioridad** | Alta. |
| **Dificultad** | Alta. |
| **Fuente** | Sistema externo, tercer proveedor. |
| **Estimulo** | Durante 10 segundos se envían progresivamente 7000 facturas en XML utilizando la opción de FTP. |
| **Artefacto** | Componente DIAN – Recepción. |
| **Ambiente** | Pruebas de aceptación en operación normal. |
| **Respuesta** | Confirmación positiva del recibo de los archivos XML. |
| **Medida** | El 100% de las facturas electrónicas enviadas y que llegan al servicio de recepción del Componente DIAN son reconocidas o no como facturas, de tal forma que, para las facturas reconocidas, quedan encoladas las peticiones para las verificaciones en procesos posteriores. |

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| **ESCA005 - Aplicar reglas de negocio a archivos XML de forma masiva.** | Se deben procesar las facturas recibidas aplicando las validaciones definidas en los requerimientos funcionales, para identificar el éxito o fallo de las mismas y persistir estos resultados. |
| **Atributo de Calidad** | Desempeño. |
| **Justificación** | Este escenario está orientado a garantizar que el sistema soporte el procesamiento masivo de las facturas que se reciben. |
| **Prioridad** | Alta. |
| **Dificultad** | Alta. |
| **Fuente** | Componente DIAN - Recepción. |
| **Estimulo** | Petición para el procesamiento de facturas recibidas. |
| **Artefacto** | Componente DIAN – Procesamiento. |
| **Ambiente** | Pruebas de aceptación en operación normal. |
| **Respuesta** | Conformidad o no de las validaciones aplicadas a las facturas. |
| **Medida** | Se aplican las validaciones a 30 facturas electrónicas de forma concurrente y para todas las facturas queda almacenada la respuesta de éxito o fallo de dichas validaciones. |

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| **ESCA006 - Tolerancia a fallos.** | El sistema debe presentar la característica de recuperabilidad ante fallos que ocurren en alguno de sus componentes. |
| **Atributo de Calidad** | Disponibilidad. |
| **Justificación** | Este escenario es relevante debido a que se debe tener alta disponibilidad en el sistema. |
| **Prioridad** | Alta. |
| **Dificultad** | Alta. |
| **Fuente** | Externa al sistema. |
| **Estimulo** | Caída de un servidor. |
| **Artefacto** | Componente DIAN – Recepción y Procesamiento. |
| **Ambiente** | Pruebas de aceptación en operación normal. |
| **Respuesta** | El sistema continúa atendiendo las peticiones de los usuarios. |
| **Medida** | Cuando se presenta la caída de un servidor en el que está desplegado o instalado un elemento del componente DIAN, éste debe estar en capacidad de recuperarse en el momento en que se reinicie el servidor afectado, y mientras tanto seguir atendiendo las peticiones que puedan atender los otros servicios no afectados. |

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| **ESCA007 - Tiempo de respuesta a peticiones del usuario.** | El participante consulta los datos del catálogo y el sistema responde según sea exitosa o no la petición. |
| **Atributo de Calidad** | Desempeño. |
| **Justificación** | Se debe tener en cuenta el tiempo de respuesta de cara al usuario final, ya que es una medida del usuario respecto a la calidad del servicio informático y puede influir en decisiones de diseño. |
| **Prioridad** | Alta. |
| **Dificultad** | Alta. |
| **Fuente** | Participantes y funcionarios DIAN. |
| **Estimulo** | Consulta de datos del catálogo de participantes con por lo menos un filtro de búsqueda. |
| **Artefacto** | Catálogo. |
| **Ambiente** | Pruebas de aceptación en operación normal. |
| **Respuesta** | Resultado de la consulta de acuerdo a los filtros ingresados. |
| **Medida** | El sistema presenta tiempos de respuesta para las consultas entre 3 y 5 segundos dependiendo de la complejidad de las mismas. [[1]](#footnote-1) |

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| **ESCA008 - Crecimiento en la demanda de capacidad o carga del sistema.** | El sistema debe tener la capacidad de soportar el aumento de usuarios o peticiones a través del tiempo. |
| **Atributo de Calidad** | Escalabilidad. |
| **Justificación** | Se espera que la cantidad de actores en el sistema incremente con el tiempo, por lo que la escalabilidad es un aspecto importante para el sistema. |
| **Prioridad** | Alta. |
| **Dificultad** | Media. |
| **Fuente** | Obligados a facturar y adquirentes. |
| **Estimulo** | Incremento de la cantidad de obligados a facturar y de adquirentes. |
| **Artefacto** | Sistema. |
| **Ambiente** | Pruebas de aceptación. |
| **Respuesta** | El sistema debe soportar la operación. |
| **Medida** | Permitir aumentar la capacidad del sistema, para soportar el incremento de obligados a facturar y adquirentes, a partir de agregar más capacidad de hardware o infraestructura. |

# REPRESENTACIÓN ARQUITECTÓNICA

En esta sección se describe el proceso de análisis utilizado para la definición de la Arquitectura de Software del sistema a implementar, adicionalmente se provee un diagrama conceptual que muestra claramente cada uno de los módulos/subsistemas a tener en cuenta dentro de la definición arquitectural.

La arquitectura es representada por medio de modelos UML a un alto nivel de abstracción, los cuales permiten dar una visión de los elementos arquitectónicos que componen el sistema, identificando los componentes de mayor importancia e identificados como críticos que pueden representar algún grado de riesgo y requieran de mayor elaboración y detalle.

Para la representación de la arquitectura se utiliza el modelo de vistas 4+1, el cual tiene en cuenta un análisis desde 5 vistas, adicionalmente se incluye una vista del modelo de datos.

La siguiente tabla muestra las diferentes perspectivas sobre las cuales se define la arquitectura de software del sistema.

|  |  |
| --- | --- |
| **Perspectiva** | **Descripción** |
| **Perspectiva de Casos de Uso** | Esta perspectiva comprende la representación de los Casos de Uso y Actores que describen el comportamiento de las funcionalidades más significativas del sistema. Esta vista se fundamenta en los requerimientos del usuario. |
| **Perspectiva Lógica** | Esta perspectiva contiene la descomposición del modelo de diseño en términos de una jerarquía de paquetes y niveles. Adicionalmente documenta las clases del diseño más significativas de los componentes genéricos que hacen parte del sistema. |
| **Perspectiva de Procesos** | Esta perspectiva considera aquellos aspectos que requieren tratamiento especial de los requerimientos no funcionales relacionados con el desempeño, la escalabilidad y la capacidad de procesamiento, define las alternativas de solución para el tratamiento de problemas referentes a concurrencia, distribución y tolerancia a fallos. Considerando componentes multihilos para la ejecución de un proceso que comprende un conjunto de tareas que conforman una unidad ejecutable. |
| **Perspectiva de Despliegue** | Esta perspectiva proporciona las configuraciones de hardware y la representación de los nodos de procesamiento que se deben contemplar para el despliegue de los componentes del sistema. |
| **Perspectiva de Implementación** | Esta perspectiva comprende los componentes que son necesarios para la implementación del sistema, define la configuración y organización de cada uno de los módulos de software en el entorno de desarrollo. Se utilizan diagramas que permiten representar con precisión los componentes de software integrados en un sistema y sus dependencias. |
| **Perspectiva de Datos** | Esta perspectiva contiene una representación del modelo de datos considerando las entidades más significativas para el sistema y sus relaciones. |

Tabla . Perspectivas de arquitectura de software

# METAS Y RESTRICCIONES DE LA ARQUITECTURA

A continuación se exponen las metas y restricciones identificadas para la arquitectura de Servicio de Estandarización y Mejora del Proceso de Factura Electrónica en Colombia.

## Metas

* Proveer la exposición de un servicio que permita la integración con todo tercero que esté interesado en ingresar al modelo de Factura Electrónica, acogiéndose al estándar definido para envío de las facturas electrónicas.
* Mantener la información de la DIAN centralizada, evitando replica de datos entre aplicaciones y asignando las debidas responsabilidades a los artefactos de software que apoyan el modelo de Factura Electrónica, para establecer un dominio de información.
* Generar trazas de auditoría acordes con las necesidades del negocio.
* Responder ante la demanda creciente principalmente del procesamiento de recibo y verificación de facturas.
* Definir los mecanismos de seguridad, acordes al manejo del documento factura electrónica, que permitan su autenticidad e integridad.
* Asegurar que la aplicación producto del Servicio de Estandarización y Mejora del Proceso de Factura Electrónica en Colombia, pueda desplegarse en una plataforma de alta disponibilidad, según los principios arquitecturales MUISCA.

## Restricciones

* Utilizar los subsistemas MUISCA que proveen diferentes servicios a nivel de arquitectura, logrando que Factura Electrónica sea un subsistema que cumpla con los principios estructurales inherentes que esta arquitectura posee.
* La aplicación debe ser construida bajo el lineamiento de las aplicaciones MUISCA respecto a no almacenar datos en la sesión HTTP.
* La aplicación debe poder desplegarse en JBoss 6.3 EAP.
* Para la persistencia de datos estructurados se debe usar la base de datos Oracle 11g Release 2 Enterprise Edition.
* La plataforma de desarrollo será Java SE Development Kit 7 update 45.

# PERSPECTIVA DE CASOS DE USO

Esta sección muestra el modelo de casos de uso que representan la funcionalidad central y más significativa del sistema final, o los que tienen mayor impacto en las decisiones de diseño de la arquitectura del sistema.

Se presenta el diagrama de casos de uso divido en tres vistas para su mejor entendimiento. Uno para el registro de los participantes, otro que relaciona las funcionalidades de la factura y el último para consultas y parametrizaciones.

## Diagrama de Casos de Uso - Registro de Participantes

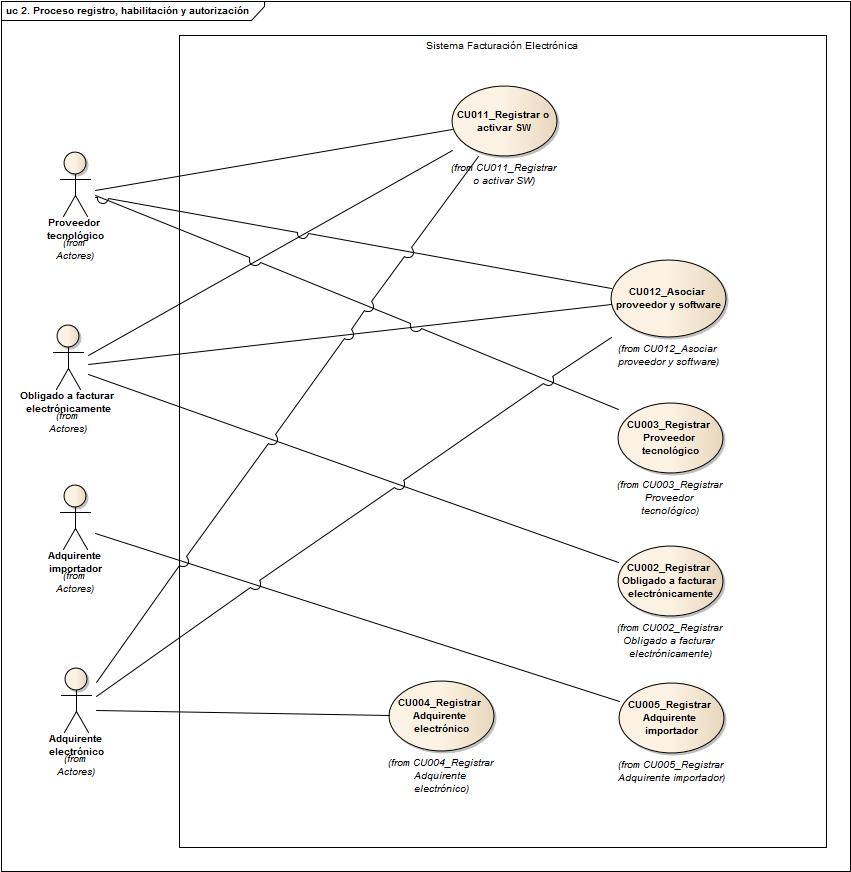


Figura . Diagrama de casos de Uso - Registro de participantes

Los casos de uso de registro de participantes conforman una de las principales formas de utilizar el sistema y se espera que estas funcionalidades sean ejecutadas con mayor frecuencia durante el inicio del modelo de factura electrónica o cuando la DIAN saque resoluciones en los que ingrese al modelo a nuevos obligados a facturar.

## Diagrama de Casos de Uso - Factura

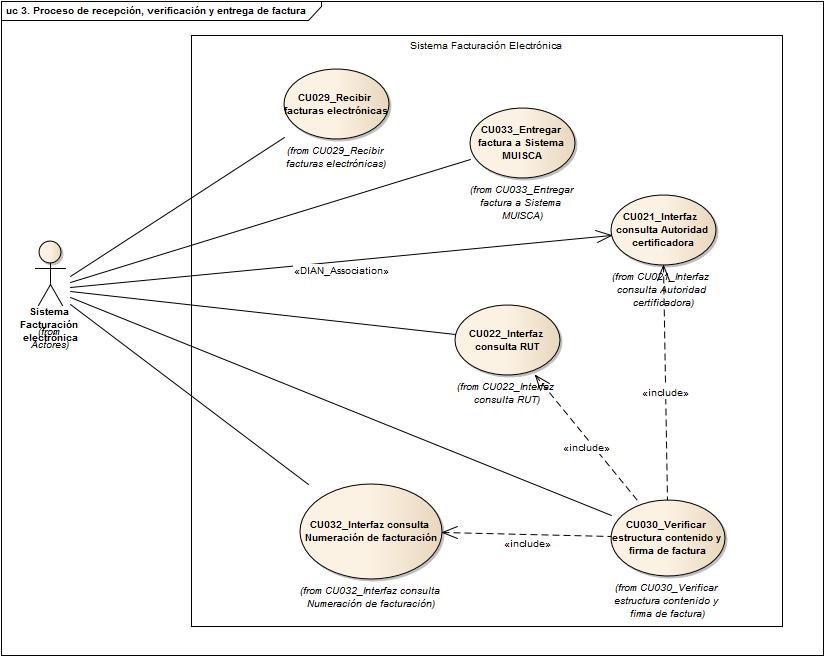


Figura . Diagrama de Casos de Uso - Factura

Los casos de uso que están relacionados directamente con la factura, son funcionalidades B2B ó relacionadas con la verificación de la factura.

## Diagrama de Casos de Uso - Consulta y Parametrización

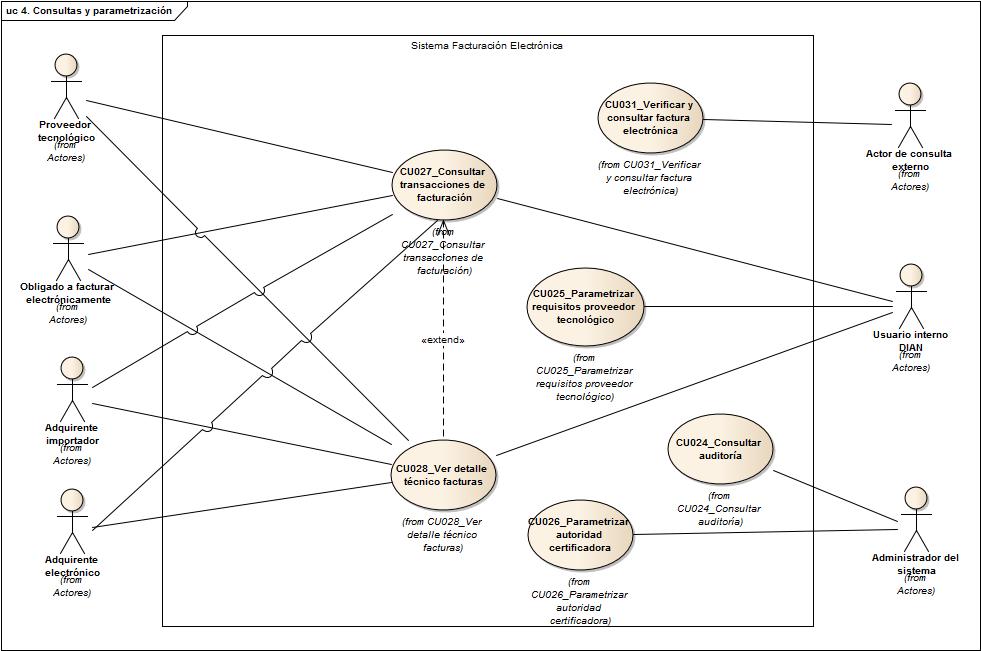


Figura . Diagrama de Casos de Uso - Consulta y Parametrización

Todos los participantes tienen interacción con el sistema en cuanto a las consultas y la posibilidad, para los obligados a facturar y adquirentes, de consultar las transacciones relacionadas con la factura electrónica y verificar la factura respectivamente.

Estas últimas funcionalidades se basan en la verificación de la factura por parte de la aplicación y la persistencia del resultado de esta. Se debe también contemplar la consulta de la auditoría y las parametrizaciones que deben realizar los funcionarios de la DIAN.

# PERSPECTIVA LÓGICA

En esta sección se describen los componentes del Sistema y el diagrama de paquetes donde se muestra la estructura de paquetes a nivel de implementación y las dependencias entre los mismos.

## Componentes del sistema

El siguiente diagrama muestra los componentes principales del sistema de Facturación Electrónica y sus relaciones con los componentes o subsistemas Muisca, tanto transversales como misionales.



Figura . Diagrama de Componentes

El sistema de Facturación Electrónica está compuesto por Componente DIAN, Catálogo y FacturaIntegracion.

**Catálogo**

El Catálogo es una aplicación Web bajo el modelo Muisca, que cubre toda la funcionalidad relacionada con la administración de la información de participantes del proceso de facturación electrónica y el software usado por ellos, que a su vez tendrá comunicación con el Componente DIAN para enviar facturas electrónicas.

El Catálogo se integra a los subsistemas Muisca RUT, Arquitectura, Comunicaciones Correo y Gestión Documental. La integración a estos subsistemas se realiza mediante el llamado de comandos de acción de cada uno de los subsistemas con que se integra.

La integración con RUT se requiere para consultar datos del RUT de los participantes que se registran en el Catálogo, con el fin de hacer validaciones o aplicar reglas de negocio de acuerdo a los requerimientos funcionales.

La integración con Arquitectura se requiere para hacer uso del modelo de seguridad Muisca, la consulta de valores dominio, y la definición de tareas programadas.

La integración con Comunicaciones Correo es requerida para enviar correos electrónicos a los participantes para notificar de ciertos eventos que ocurren con el estado de la información relacionada a ellos.

La integración con Gestión Documental se requiere para almacenar documentos que agregan los participantes durante el proceso de registro de su información.

**Componente DIAN**

Este componente se divide en dos: recepción y procesamiento.

Recepción cubre la funcionalidad de recibir las facturas electrónicas por medio de Web Service, validar la estructura de las facturas, registrar el resultado de la validación y dejar disponible la factura en el sistema de archivos para que sea tomada por el componente de procesamiento.

Procesamiento se encarga de ejecutar las validaciones sobre las facturas previamente recibidas. Por cada validación se registra el resultado para su posterior consulta. El procesamiento que realiza el componente se basa en la ejecución de actividades encadenadas o coordinadas por medio de mensajes JMS. Para cada actividad se aplica una regla de procesamiento que identifica el procesador que la actividad debe ejecutar.

Para algunas de las validaciones se debe realizar consultas a otros subsistemas Muisca, para lo cual se define el componente FacturaIntegracion.

**FacturaIntegracion**

Este componente se define como un proxy o puente entre el Componente DIAN y los subsistemas a los cuales debe integrarse. Esto se hace buscando mejorar la mantenibilidad y el bajo acoplamiento entre las integraciones, ya que si no existiera FacturaIntegracion, cada integración que requiera el Componente DIAN sería una relación de acoplamiento directa.

De esta forma FacturaIntegracion permite que el Componente DIAN pueda consultar información de los subsistemas: RUT, Numeración de Facturación.

En el RUT se consultan los datos del registro de RUT que tiene el obligado a facturar de la factura recibida para validar el estado del RUT.

En Numeración de Facturación se consultan los rangos de numeración y la clave técnica para validar el número de factura y el CUFE (Código Único de Factura Electrónica).

## Paquetes de Diseño

Para Catálogo y FacturaIntegracion se plantea una estructura de paquetes bajo el modelo Muisca.



Figura . Diagrama de Paquetes Catálogo

En el paquete web se define el paquete wbo para ubicar las clases WBO que controlan las páginas Web. Los WBO usan a los delegados que están ubicados en el paquete general.delegados. Los delegados llaman a los comandos de acción ubicados en el paquete acciones. Los comandos de acción usan a los comandos de servicios ubicados en el paquete servicios. Los servicios se apoyan en los DAOs que se ubican en el paquete dao. Las clases TO se ubican en el paquete general.to y son usados por las acciones, servicios y daos.



Figura . Diagrama de paquetes FacturaIntegracion

Para FacturaIntegracion se definen los mismos paquetes a excepción del paquete web, ya que no es necesario puesto la intención de FacturaIntegracion es proveer delegados que permitan realizar la integración entre el Componente y los diferentes subsistemas Muisca.



Figura . Diagrama de paquetes Componente

Respecto al Componente se tienen dos paquetes principales com.indra.mmdd.corporate y co.gov.dian.muisca.facturaelectronica. En com.indra.mmdd.corporate se ubican las clases que conforman el core del componente. En co.gov.dian.muisca.facturaelectronica se ubican las clases que implementan el procesamiento específico para las facturas.

El paquete com.indra.mmdd.corporate.activities contiene las clases base para las actividades, las cuales contienen lógica de publicación de mensajes JMS, gestión de errores, trazabilidad y auditoría.

El paquete com.indra.mmdd.corporate.processors contiene las interfaces que deben implementar los procesadores específicos.

El paquete com.indra.mmdd.corporate.domain contiene las entidades JPA que maneja el core del componente para su funcionamiento.

El paquete co.gov.dian.muisca.facturaelectronica.activities contiene las actividades específicas para las validaciones de la factura, las cuales heredan de las clases base ubicadas en com.indra.mmdd.corporate.activities.

El paquete co.gov.dian.muisca.facturaelectronica.processors contiene los procesadores específicos para las validaciones de la factura, los cuales implementan las interfaces definidas en com.indra.mmdd.corporate.processors.

El paquete co.gov.dian.muisca.facturaelectronica.domain contiene las entidades JPA que se necesitan del modelo de negocio de la Facturación Electrónica, por ejemplo la entidad de verificación de factura en la que se registra el resultado de las verificaciones.

En el paquete co.gov.dian.muisca.facturaelectronica.delegados se ubican las clases que sirven para llamar a los servicios de integración con subsistemas Muisca que se implementan sobre el subsistema Muisca FacturaIntegracion. Estos delegados son usados por los procesadores.

El paquete co.gov.dian.muisca.facturaelectronica.validation contiene las clases que hacen las validaciones de las facturas recibidas en XML. Estas validaciones se hacen contra archivos XSD que definen la estructura de los archivos de facturas electrónicas. Además se validan los archivos ZIP, de tal forma que realmente sean archivos comprimidos con datos.

## Clases de Diseño Significativo para la Arquitectura

A continuación se muestran las principales clases que intervienen en la recepción de las facturas y el posterior procesamiento de las mismas.

En general, la recepción de facturas se hace por medio de clases llamadas conceptualmente conectores. Estos conectores heredan de clases base que hacen parte del core del Componente. Existe un conector para el procesamiento de las peticiones de los web service y existen dos conectores para soportar la recepción de facturas por FTP.

En el caso del web service se recibe una sola factura a la vez, la cual se espera recibir dentro del mensaje SOAP de forma comprimida (zip). El web service realiza una tarea de verificación y reconocimiento del documento recibido. Luego, crea registros de trazabilidad o auditoría indicando la recepción exitosa de la factura. Si la verificación fue exitosa se envía un mensaje a una cola JMS para dejar encolado el proceso de las validaciones a la factura. El web service se libera retornando una respuesta con un mensaje de éxito o error.

En el caso de la recepción por FTP, se espera recibir archivos comprimidos (zip) con un conjunto de facturas. El archivo ZIP no debe contener directorios, solamente el conjunto de facturas. Se usa Quartz para programar dos procesos que se ejecutan automáticamente. El primer proceso se encarga de leer los archivos desde el sistema de archivos donde se reciben las facturas, por cada archivo detectado se realizan los siguientes pasos:

* Se valida que el archivo sea un ZIP correcto con archivos por dentro.
* Se mueve el archivo ZIP a una ruta configurada y se descomprime.
* Se crean registros en base de datos por cada factura que viene en el ZIP.

El segundo proceso de recepción de facturas por FTP se encarga de consultar en base de datos los registros creados en el proceso anterior, que corresponden a facturas pendientes de procesar. Por cada factura a procesar realiza un reconocimiento de la misma, identifica la actividad a ejecutar y envía un mensaje a una cola JMS según la actividad identificada.



Figura . Diagrama de clases - Recepción de facturas

|  |  |
| --- | --- |
| **Clase/Interfaz** | **Descripción** |
| InvoiceIncomingEndpoint | Clase que define el web service para recibir las facturas. Delega el procesamiento de las peticiones que recibe a la clase WSReceptionConnector. |
| WSReceptionConnector | Conector con la lógica para recibir y reconocer la factura recibida por el web service. Esta clase hereda de DocsReceptionByWSConnectorAbstract. |
| DocsReceptionByWSConnectorAbstract | Clase abstracta que define la base para el conector de web service. Esta clase establece la firma de algunos métodos abstractos que deben ser implementados en el conector. Hereda de DocsReceptionConnectorAbstract. |
| DocsReceptionConnectorAbstract | Clase abstracta que implementa los métodos activateWorkFlow() que se encarga de identificar la siguiente actividad del workflow a ejecutar para el procesamiento de la factura; y el método publishMessage() que se encarga de enviar el mensaje a la cola JMS apropiada. |
| ReceiveDocumentsFromFilesystem | Clase que define el primer proceso asíncrono ejecutado con Quartz para la recepción de facturas por FTP. Delega el procesamiento de cada archivo ZIP a la clase FilesystemPollingReceptionConnector. |
| FilesystemPollingReceptionConnector | Conector con la lógica para validar cada archivo ZIP que se recibe por FTP. Hereda de DocsReceptionByFilesystemPollingConnectorAbstract. |
| DocsReceptionByFilesystemPollingConnectorAbstract | Clase abstracta que define la base para el conector de recepción de archivos ZIP. Esta clase establece la firma de algunos métodos abstractos que deben ser implementados en el conector. Hereda de DocsReceptionConnectorAbstract. |
| ReceiveChildrenDocuments | Clase que define el segundo proceso asíncrono ejecutado con Quartz para la recepción de facturas por FTP. Delega el procesamiento de cada factura recibida a la clase ChildrenDocsPoolingReceptionConnector. |
| ChildrenDocsPoolingReceptionConnector | Conector con la lógica para validar y reconocer cada factura recibida dentro de un archivo ZIP. Hereda de ChildDocsReceptionByDbPoolingConnectorAbstract. |
| ChildDocsReceptionByDbPoolingConnectorAbstract | Clase abstracta que define la base para el conector de validación y reconocimiento de facturas recibidas en archivos ZIP. Esta clase establece la firma de algunos métodos abstractos que deben ser implementados en el conector. Hereda de DocsReceptionConnectorAbstract. |

Tabla . Clases de recepción de facturas

El siguiente diagrama muestra las clases principales para el procesamiento de las facturas, esto implica ejecutar las validaciones sobre cada factura. Por cada validación se define un Message Driven Bean (MDB) y un procesador. Cada MDB lee mensajes de una cola JMS independiente. Cada procesador tiene la lógica específica para realizar la validación sobre la factura.

La clase FaceValidationActivity define el concepto de actividad de workflow. Esta clase realiza lo siguiente:

* Identifica e instancia el procesador apropiado.
* Inicia la ejecución del procesador, obtiene el resultado y lo persiste.
* Identifica la próxima actividad a ejecutar.
* Publica mensaje JMS a la cola destinada para la próxima actividad.



Figura . Diagrama de clases - Procesamiento de facturas

|  |  |
| --- | --- |
| **Clase/Interfaz** | **Descripción** |
| Validacion1MDB, Validacion2MDB, ValidacionNMDB | Message Driven Beans definidos por cada validación de la factura. |
| Procesador1, Procesador2, ProcesadorN | Procesadores definidos por cada validación de la factura. Deben implementar la interfaz ActionProcessor. |
| ActionProcessor | Interfaz que define el método performAction() donde se debe implementar la lógica concreta para hacer las validaciones de factura. |
| FaceValidationActivity | Actividad general que es ejecutada por cada MDB. Hereda de WorkFlowActivityAbstract. Implementa el método getAdditionalParams() para incluir parámetros adicionales que se necesitan propagar a los procesadores para la ejecución de las validaciones. |
| WorkFlowActivityAbstract | Clase abstracta que implementa el método processDocumentReceived(), el cual es el punto de acceso a la ejecución de la actividad. Además implementa el método createActionProcessor() encargado de identificar e instanciar el procesador apropiado. Hereda de DocumentActivityAbstract. |
| DocumentActivityAbstract | Clase abstracta que implementa el método publishMessage() encargado de publicar el mensaje JMS y define el método abstracto getAdditionalParams(). |

Tabla . Clases de procesamiento de facturas

## Realizaciones de Casos de Uso

Esta sección ilustra cómo el software funciona presentando realizaciones de casos de uso, o escenarios de Recibir Factura Electrónica y Verificar Factura Electrónica. Se representa cómo los diferentes elementos del modelo de diseño contribuyen a su funcionalidad.

Los diagramas de secuencia se muestran a alto nivel con el fin de hacer más clara las principales clases y métodos que intervienen en cada funcionalidad.



Figura . Diagrama de secuencia - Recibir Factura

La clase InvoiceIncomingEndpoint define el punto de acceso del web service que recibe las facturas electrónicas. Esta clase invoca el método processRequest de la clase WSReceptionConnector.

En el método checkFileOK() se realizan validaciones de la factura que se recibe. Se espera que la factura esté en formato comprimido ZIP y sólo se recibe una factura por web service. Luego de reconocer que es un ZIP válido con una sola factura, se procede a obtener los datos principales del XML de la factura (NIT, número de factura, fecha) por medio de la clase DocumentValidator. La clase DocumentValidator se apoya en JAXB para realizar el procesamiento del XML.

En el método createDocument() se crea un registro en base de datos que representa el documento o factura recibida y un registro de auditoría con información adicional, además se crea el archivo físico de la factura en un directorio configurado para recibir las facturas que luego se procesarán.

El método activateWorkFlow() identifica la primer actividad a ejecutar, obtiene el nombre de la cola JMS para iniciar la ejecución de dicha actividad y envía el mensaje a la cola JMS llamando al método publishMessage(). El mensaje que se publica en la cola JMS no contiene todo el contenido de la factura, solamente tiene el identificador del documento creado, y la identificación de la actividad a ejecutar.

Finalmente, el método createWSResponse() crea la respuesta de éxito o error que se retorna al cliente del web service.

El siguiente diagrama muestra el comportamiento del componente al recibir archivos comprimidos con facturas por FTP.



Figura . Diagrama de secuencia – Recibir archivos con facturas por FTP

El método executeInternal() de la clase ReceiveDocumentsFromFilesystem se ejecuta automáticamente a través de Quartz. Se obtienen los archivos por medio del método getFileList(), en el cual se hace un checksum inicial de cada archivo. Luego, por cada archivo se crea un hilo con el cual se ejecuta el método run() de la clase FilesystemPollingReceptionConnector.

El método processingIncomingDocument() coordina el procesamiento de un archivo recibido. El primer método que se invoca es checkFileIsReady() en el cual se calcula nuevamente el checksum y lo compara con el inicialmente calculado, si son diferentes quiere decir que el archivo aún no ha terminado de transferirse completamente y no se procesa; de lo contrario se continúa el proceso del archivo.

El método checkFileOK() realiza una validación del archivo para identificar que sea un ZIP.

El método createDocument() crea registros de documentos en base de datos representando el archivo recibido, mueve el ZIP de la carpeta de recepción a una ubicación configurada, descomprime y por cada factura del ZIP crea un registro de documento que representa las facturas recibidas en el ZIP.

En este procesamiento se considera el manejo de la transacción ya que al hacer un rollback se deben dejar los archivos tal como estaban al inicio del procesamiento.

El procesamiento específico de cada factura del ZIP se hace de forma separada por medio de otro proceso Quartz.



Figura . Diagrama de secuencia – Recibir factura por FTP

La clase ReceiveChildrenDocuments se ejecuta automáticamente por medio de Quartz. Lo primero que se hace es consultar los documentos creados que representan facturas recibidas sin procesar, a través del método getDocumentList(). Luego, por cada documento consultado se crea un hilo en el cual se inicia la ejecución del método run() de la clase ChildrenDocsPoolingReceptionConnector.

El procesamiento de una factura se realiza en el método processIncomingDocument(), donde se verifica el archivo de la factura (método checkFileOK()), se crea un registro de documento para llevar la trazabilidad del procesamiento de la factura (método createDocument()), se identifica la primer validación a realizar (método activateWorkFlow()) y se publica el mensaje en la cola JMS (método publishMessage()). De ahí en adelante se continúa con la ejecución del procesamiento de la factura.

El siguiente diagrama muestra el comportamiento general del componente para realizar una validación sobre la factura.



Figura . Diagrama de secuencia - Verificar Factura Electrónica

Por cada validación sobre la factura existe un Message Driven Bean (MDB) que lee los mensajes publicados en una cola JMS específica (para cada validación también debe definirse una cola JMS). Cuando el MDB lee el mensaje de la cola JMS, ejecuta una actividad de validación. La clase base para las actividades de validación es FaceValidationActivity.

La clase FaceValidationActivity, en su método processDocumentReceived(), mediante la configuración de las reglas de procesamiento almacenadas en base de datos, obtiene el procesador especifico a instanciar (clase que implemente la interfaz ActionProcessor).

Al ejecutar el método performAction() del procesador, se ejecuta el método doCustomAction(), en el cual se hace la validación de negocio sobre la factura. En este punto se muestra un llamado a un Delegado, el cual a su vez hace un llamado a un delegado de FacturaIntegracion para invocar un servicio de otro subsistema Muisca, en los casos donde la validación requiere consultar información de otros subsistemas Muisca.

Como se puede ver en el diagrama toda la lógica ejecutada en un procesador se ejecuta bajo el contexto de una transacción de base de datos, para garantizar que la operación de negocio se ejecute correctamente o se realice un rollback en caso de error.

Finalmente, se publica un mensaje a otra cola JMS para que la siguiente actividad se ejecute sobre la factura.

# PERSPECTIVA DE PROCESOS

En esta sección se presenta el enfoque del Componente para aprovechar el procesamiento en paralelo.

El siguiente diagrama muestra las actividades de alto nivel realizadas durante la recepción de una factura por medio de web service.



Figura . Diagrama de proceso de recepción de factura por web service

Cada vez que se recibe una petición por medio del web service se realizan las actividades mostradas. El sistema puede atender las peticiones al web service de forma concurrente, de tal forma que cada petición es un proceso paralelo independiente.



Figura . Diagrama de proceso de recepción por FTP

Respecto a la recepción por FTP, se usa Quartz configurado en cluster por medio de la base de datos. Esto permite que las instancias que tienen desplegado Componente – Recepción se sincronicen y se presente un balanceo de carga en la ejecución de este proceso.

Los dos procesos anteriores realizan un reconocimiento de la factura para identificar el workflow configurado que debe ejecutarse para la factura y así enviar un mensaje a una cola JMS para realizar las verificaciones de la factura en procesos independientes. Estos procesos de recepción realizan las actividades mínimas requeridas con el fin de que sean livianos y se cumplan con los tiempos de desempeño definidos en los escenarios de calidad.

Se tienen dos workflows configurados: uno para facturas nacionales y otro para facturas internacionales.

El siguiente diagrama muestra las actividades del workflow para facturas nacionales.



Figura . Diagrama de proceso de verificaciones de facturas nacionales

Una misma factura nacional pasa por el flujo que se muestra en la figura anterior, el cual es orquestado por medio de mensajes a colas JMS y Message Driven Beans que leen cada cola JMS.

Las agrupaciones punteadas indican procesos que se pueden ejecutar de forma paralela para diferentes facturas. Esta es la característica que provee el Componente para aprovechar el procesamiento en paralelo de varias facturas. Este esquema permite configurar la cantidad de instancias de cada MDB, favoreciendo el desempeño del procesamiento de las facturas.

La última actividad llamada “Realizar control de validaciones” se encarga de consolidar el resultado de las validaciones anteriores e identificar si la factura fue exitosa o no.

Respecto al proceso de las verificaciones de facturas internacionales, se aplica el mismo concepto de paralelismo y se reutilizan las clases Java que tienen la lógica para ejecutar las verificaciones de la factura. La diferencia con el proceso anterior es que no se ejecutan las verificaciones de validar obligado y validar numeración de factura.



Figura . Diagrama de proceso de verificaciones de facturas internacionales

# PERSPECTIVA DE DESPLIEGUE

En esta vista se identifica la distribución de los componentes de software en servidores físicos y la relación entre estos.

Para el despliegue de Catálogo se tiene un esquema típico de los subsistemas Muisca. El siguiente diagrama muestra este esquema.



Figura . Diagrama de despliegue de Catálogo

En el diagrama anterior se ve que el despliegue de Catálogo se hace sobre JBoss 6.3 EAP. Este esquema de despliegue permite dar la escalabilidad y disponibilidad requeridas por el Catálogo, ya que se pueden subir o bajar la cantidad de instancias necesarias para atender a los clientes de acuerdo a las cargas que presente el sistema.

Para el despliegue del Componente deben existir dos entornos, uno para la operación normal y otro para el proceso de habilitación. Los participantes de la facturación electrónica deben pasar por el proceso de habilitación antes de iniciar la operación real. Para el entorno de operación se plantea una topología en alta disponibilidad y tolerancia a fallos con cluster de JBoss y redundancia de nodos. Para el entorno de habilitación, como se espera menos carga, no se plantea una topología en alta disponibilidad.

El Componente se compone de diferentes elementos que se pueden desplegar de forma separada y con nodos redundantes. Estos elementos son: Componente – Recepción, Componente – Procesamiento, Mensajería.

El siguiente diagrama presenta el esquema de despliegue de los elementos que conforman el Componente, así como sus dependencias.

Como se puede ver en el diagrama, se plantean como mínimo dos instancias de servidores de aplicaciones por cada elemento del sistema, de tal forma que si falla una instancia, la otra sigue activa para atender las solicitudes. Además, se mantienen colas JMS persistentes y con un esquema de backup que permite recuperarse de fallos en este punto.



Figura . Diagrama de despliegue

Los bordes azules representan los cluster de JBoss que se configuran para desplegar cada elemento del Componente. Hay cluster de JBoss para Componente – Recepción, Componente – Procesamiento y Mensajería.

En los nodos donde se despliega Componente – Procesamiento se requiere desplegar FacturaIntegracion, Numeración de Facturación, RUT y MuiscaCore.

Componente, tanto Recepción como Procesamiento, usan la base de datos FAE en el entorno de operación y FAE\_HAB en el entorno de habilitación.

FacturaIntegracion usa la base de datos FAE\_CAT, la cual es la misma que usa Catálogo. FAE\_CAT es la misma base de datos para habilitación y operación.

La escalabilidad se soporta porque se pueden activar nuevas instancias de recepción, procesamiento y mensajería; de acuerdo al incremento de las peticiones en cualquiera de estos elementos.

# PERSPECTIVA DE IMPLEMENTACIÓN

## Panorama

Esta sección muestra los elementos estructurales y de alto nivel, con sus relaciones de la vista de implementación.

Para el Catálogo se plantea una estructura con 3 capas: presentación, negocio y datos; y un elemento general que es transversal a las capas. Las relaciones de uso deben ser de la capa de presentación a la capa de negocio, y de esta a la capa de datos.



Figura . Panorama de implementación de Catálogo

Para FacturaIntegracion aplica la misma estructura del diagrama anterior con la diferencia que no existe capa de presentación.

El siguiente diagrama muestra la estructura de alto nivel del Componente.



Figura . Panorama de implementación de Componente

|  |  |
| --- | --- |
| **Elemento** | **Descripción** |
| B2BIntegrationEngineLib | Donde están las actividades y procesadores para el procesamiento de las facturas, además de clases utilitarias. Usa FacturaIntegracion para realizar el llamado a servicios Muisca. |
| B2BIntegrationSchemas | Contiene los esquemas XSD que define la estructura de los archivos XML para las facturas. Además de las clases que representan cada uno de los elementos de dichos esquemas. |
| B2BIntegrationScheduler | Contiene el web service de recepción de facturas y las clases que se ejecutan por medio de Quartz para la recepción de archivos de facturas por FTP. Usa B2BIntegrationEngineLib y B2BIntegrationSchemas. |
| B2BIntegrationQueues | Contiene los Message Driven Beans de las validaciones que deben realizarse a las facturas. |

Tabla . Elementos panorama de implementación de Componente

## Niveles

En el siguiente diagrama se muestra una vista de implementación donde se relacionan los principales elementos arquitectónicos, tecnologías y librerías de Muisca usadas para el Catálogo.

Para FacturaIntegracion aplican los mismos elementos mostrados en el diagrama, a excepción de que no se tiene capa de presentación y no se usan las librerías Muisca de Gestión Documental, y Comunicaciones.



Figura . Diagrama de niveles de Catálogo

**Presentación**

|  |  |
| --- | --- |
| **Componente** | **Descripción** |
| **JSF 2.1** | Estándar para la creación de interfaces de usuario del lado del servidor. La tecnología JavaServer Faces incluye:   * Un conjunto de APIs para representar componentes de interfaz de usuario y la gestión de su estado, la gestión de eventos y la validación de entrada, la definición de la página de navegación, y el apoyo a la internacionalización y accesibilidad. * Librería de etiquetas personalizadas JavaServer Pages (JSP) para expresar una interfaz JavaServer Faces dentro de una página JSP. |
| **PrimeFaces 4.**0 | PrimeFaces es un conjunto de componentes JSF de código abierto con algunas extensiones. Sus principales características son:   * Rico conjunto de componentes (HTMLEditor, de diálogo, de Autocompletar, de Gráficos y muchos más). * Soporte Ajax basado en el estándar JSF 2.0 Ajax APIs. * Ligero, empaquetado en un jar, no requiere configuración ni dependencias adicionales. * Ajax Push soportado a través de websockets. * Extensa documentación. |
| **WBOs** | Representa cada uno de los Managed Beans que está asociado con los componentes de interfaz de usuario se utilizan en una página en particular. |

**Negocio**

|  |  |
| --- | --- |
| **Componente** | **Descripción** |
| **Delegados** | Clases que definen los puntos de acceso a la funcionalidad de negocio por parte de la capa de presentación. |
| **Comandos de Acción** | Los Comandos de Acción son elementos en donde se debe implementar la lógica de negocio, puede contener la orquestación de Comandos de Acción o Comandos de Servicio. |
| **Comandos de Servicio** | Los Comandos de Servicio son los únicos elementos habilitados para implementar el uso de la capa de datos mediante el administrador de persistencia. |

**Datos**

|  |  |
| --- | --- |
| **Componente** | **Descripción** |
| **Administrador de Persistencia** | Componente encargado de realizar operaciones de consulta, modificación, creación y eliminación sobre la base de datos. |
| **DAOs** | Objetos que contienen la lógica de acceso a datos. Se decide no usar JPA, y a cambio, usar DAOs. Los puntos que llevaron a tomar esta decisión son los siguientes:   * Volumen de datos grande. * Inserción de datos de forma simultánea. * Ejecución de procedimientos almacenados. * Facilidad de crear consultas complejas directamente en SQL. |
| **JDBC** | Java Database Connectivity, la tecnología usada para realizar las operaciones contra la base de datos. |

**General**

|  |  |
| --- | --- |
| **Componente** | **Descripción** |
| **TOs** | Objetos POJO que se utilizan para transferir datos entre capas. |
| **Enumeraciones** | Enumeraciones definidas para el subsistema. |
| **Constantes** | Clases o interfaces que definen valores constantes usados en el subsistema. |

Respecto a la seguridad en Catálogo y FacturaIntegracion, por ser subsistemas bajo el modelo Muisca, se usa el módulo de seguridad de Muisca.

En cuanto al Componente, se maneja seguridad en el web service de tal forma que se valida contra el identificador y la contraseña de acceso que tenga registrado el software que envía la factura. La implementación de esto se hace por medio del framework Spring, el cual provee componentes para facilitar la implementación de WS-Security, el cual se implementa con Username token profile. La respuesta del web service es firmada con un certificado digital.

Para garantizar el no repudio y la integridad de los datos de la factura se espera que las facturas recibidas estén firmadas. Una de las validaciones que se realiza sobre la factura al procesarla es validar la firma.

La recepción de facturas por FTP, por ser un proceso que se ejecuta automáticamente, no tiene implementadas validaciones de acceso. Sin embargo, se delega el acceso a los repositorios donde se almacenan las facturas, al sistema de FTP configurando las respectivas cuentas de usuario.



Figura . Diagrama de niveles de Componente

**B2BIntegrationScheduler**

|  |  |
| --- | --- |
| **Componente** | **Descripción** |
| **Web Service** | Web service de recepción de facturas. |
| **Jobs** | Clases Java con la implementación de los Jobs que leen la ruta de directorio donde se reciben facturas por FTP. |
| **Quartz** | Librería open source utilizada para programar los Jobs. |

**B2BIntegrationQueues**

|  |  |
| --- | --- |
| **Componente** | **Descripción** |
| **MDBs** | Se define un Message Driven Bean por cada validación de factura. Cada uno lee mensajes de colas JMS diferentes. |
| **EJB** | Tecnología que permite implementar los MDBs. |
| **JMS** | Colas JMS sobre las que leen los MDBs. |

**B2BIntegrationSchemas**

|  |  |
| --- | --- |
| **Componente** | **Descripción** |
| **Clases Java de Estructuras** | Clases Java autogeneradas por medio de JAXB a partir de los esquemas XSD. |
| **XSD** | Esquemas que definen la estructura del XML de factura y la estructura de la petición y respuesta del web service de recepción de facturas. |
| **JAXB** | Tecnología Java usada para el procesamiento de los XML de facturas creando instancias de clases Java equivalentes. |

**B2BIntegrationEngineLib**

|  |  |
| --- | --- |
| **Componente** | **Descripción** |
| **Conectores** | Conectores definidos por el core del componente que permiten la recepción de documentos por diferentes medios. En este caso se usan para la recepción de facturas por web service y por sistemas de archivos. |
| **Validación estructura** | Clases Java que permiten realizar la validación de la estructura del XML de la factura a partir del XSD. |
| **Actividades** | Clases Java que se ejecutan por el workflow de procesamiento de facturas. |
| **Procesadores** | Clases Java que son instanciadas y ejecutadas por cada actividad del workflow de procesamiento de facturas. |
| **Delegados** | Clases Java que se encargan del llamado de servicios a FacturaIntegracion para realizar la integración con subsistemas Muisca. |
| **JPA** | Java Persistence API. Framework de mapeo objeto relacional para facilitar las operaciones de persistencia. El componente está orientado a usar JPA y como maneja sus propias tablas de base de datos, diferentes a las de catálogo, se decide seguir usando JPA para el componente. |
| **Hibernate** | Implementación concreta que se usa para JPA. |

# PERSPECTIVA DE DATOS

En este capítulo se presenta una vista de alto nivel de las principales entidades de datos que maneja el sistema.

En el diagrama siguiente se muestran las principales entidades correspondientes al negocio de Facturación Electrónica.



Figura . Diagrama conceptual de entidades de datos de Catálogo

|  |  |
| --- | --- |
| **Entidad** | **Descripción** |
| FAE\_FACTURA | Entidad principal que almacena los datos de las facturas recibidas y verificadas. |
| FAE\_VERIF\_FACTURA | Se utiliza para almacenar el resultado de las verificaciones de las facturas. |
| FAE\_PARTICIPANTE | Almacena la información relacionada con los participantes en el proceso de Facturación Electrónica. |
| FAE\_PARTICIPANTE\_TIPO | Define el tipo que tiene un participante en un periodo de tiempo. |
| FAE\_PART\_TIPO\_SOFTWARE | Mantiene la relación entre la factura, el participante, y el software usado para el envío de la factura. |
| FAE\_SOFTWARE | Registro de los diferentes software que enviarán facturas. |
| FAE\_AMBIENTE\_SOFTWARE | Mantiene el control del ambiente en el que está un software registrado y los datos de acceso para permitir la recepción de facturas por parte de dicho software. |

Tabla . Diagrama conceptual de entidades de datos de Catálogo

El componente maneja entidades de datos adicionales que se utilizan para la configuración del mismo y la trazabilidad de los procesos que ejecuta.



Figura . Diagrama conceptual de entidades de datos de Componente

|  |  |
| --- | --- |
| **Entidad** | **Descripción** |
| COM\_WORKFLOWCONFIGURATION | Entidad principal que define la configuración de un workflow para el procesamiento de las facturas. En este contexto, se entiende por workflow una serie de actividades o pasos que se ejecutan secuencialmente. |
| COM\_WORKFLOWSTEP | Permite definir los diferentes pasos o actividades que tiene el workflow. |
| COM\_WORKFLOWSTEPRULES | Cada actividad o paso puede contener una o más reglas de procesamiento. |
| COM\_PROCESSINGRULE | Define las reglas de procesamiento. En esta entidad se indica la clase Java que implementa la regla de procesamiento correspondiente a un procesador. |
| COM\_DOCUMENT | Representa un archivo que se recibe para ser procesado por un workflow. |
| COM\_CHILDDOCUMENTSTOPROCESS | En el caso de archivos ZIP con varias facturas, un documento tendrá asociados varios registros hijos que representan a cada una de las facturas contenidas en el ZIP. |
| COM\_DOCAUDITRECORD | A medida que se procesa el documento por medio de las actividades del workflow, se crean registros de auditoría con los resultados. |
| COM\_DOCAUDITRECORDPROPERTY | Para cada registro de auditoría se pueden almacenar datos específicos que ofrecen detalles de trazabilidad del proceso. |

Tabla . Entidades de datos del Componente

1. Estos tiempos fueron acordados entre DIAN e Indra en el acta de reunión publicada en la ruta https://gdocumental.dian.gov.co/cliente/DIAN/SGT/Documentos%20Cliente/Fabrica%20de%20Software/Portafolio/04.%20Monitoreo%20y%20Control/Actas%20de%20Reuni%C3%B3n/Comite%20Arquitectura/AREU\_TiemposRespuesta\_2014-12-23.pdf [↑](#footnote-ref-1)