Facturación electrónica

DC3 Software

Análisis y diseño: especificación de requerimientos y diseño del sistema

Daniela Cepeda

Carlos Quijano

Camilo Cepeda

Cristian Triana

Versión 1

12/04/2019

Tabla de contenido

[1. Introducción](#_Toc2617657)

[1.1 Propósito](#_Toc2617658)

[1.2 Alcance](#_Toc2617659)

[1.3 Definiciones, Acrónimos y Abreviaciones](#_Toc2617660)

[2 Descripción General](#_Toc2617661)

[2.1 Perspectiva del producto](#_Toc2617662)

[2.2 Funcionalidades del Producto](#_Toc2617663)

[2.2.1 < Funcionalidad 1 >](#_Toc2617664)

[2.3 Características de los Actores](#_Toc2617665)

[2.4 Restricciones Generales](#_Toc2617666)

[3 Requerimientos No Funcionales](#_Toc2617667)

[3.1 Interfaz Externa](#_Toc2617668)

[3.1.1 Interfaz de usuario](#_Toc2617669)

[3.1.2 Interfaz de Hardware](#_Toc2617670)

[3.1.3 Interfaces de Software](#_Toc2617671)

[3.1.4 Interfaces de comunicación](#_Toc2617672)

[3.2 Usabilidad](#_Toc2617673)

[3.3 Desempeño](#_Toc2617674)

[3.4 Seguridad](#_Toc2617675)

[3.5 Confiabilidad](#_Toc2617676)

[3.6 Disponibilidad](#_Toc2617677)

[3.7 Mantenibilidad](#_Toc2617678)

[3.8 Portabilidad](#_Toc2617679)

[3.9 Escalabilidad](#_Toc2617680)

[4 Requerimientos Funcionales](#_Toc2617681)

[4.1 Lista de Casos de Uso](#_Toc2617682)

[4.2 Diagrama de Casos de Uso](#_Toc2617683)

[4.3 Documentación de Casos de Uso](#_Toc2617684)

[5 Modelos de Diseño](#_Toc2617685)

[5.1 Modelo lógico / de relación](#_Toc2617686)

[5.2 Modelo De Arquitectura](#_Toc2617687)

[5.3 Modelos dinámicos](#_Toc2617688)

[5.4 Modelos de Infraestructura](#_Toc2617689)

[6 Diseño de Interfaces](#_Toc2617690)

[7 Inspección](#_Toc2617691)

# 1. Introducción

El software está diseñado para realizar facturación electrónica para empresas pequeñas de Colombia según el Decreto 2242 de 2015, en este documento se especificarán los requerimientos mediante la necesidad del cliente, además de un diseño básico de la arquitectura del software.

## Propósito

El propósito de este documento es evidenciar el proceso de construcción de los requerimientos y diseño del software mediante el análisis a las necesidades del cliente.

Roles:

* Cristian Triana:
* Camilo Cepeda:
* Daniela Cepeda:
* Carlos Quijano:

## Alcance

El software va dirigido a establecimientos pequeños a los cuales se les aplica el impuesto del IVA para que la DIAN pueda realizar el respectivo cobro.

## Definiciones, Acrónimos y Abreviaciones

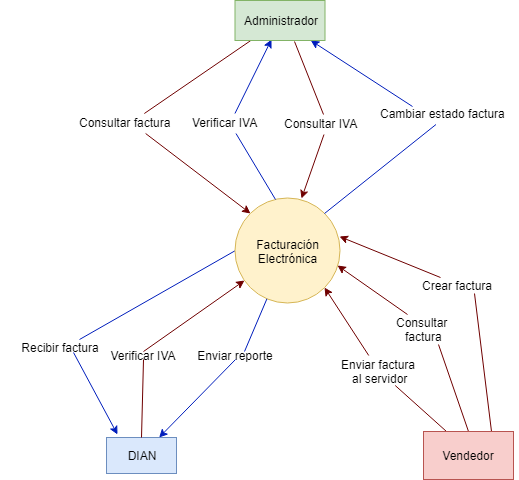
* DIAN: Dirección de impuestos y aduanas nacionales de Colombia.
* IVA: Impuesto al valor agregado.
* Facturación electrónica: Es un documento que sirve para describir el costo de los servicios y desglosar los impuestos correspondientes a pagar.
* SW: Software.
* Tributar: Cantidad de dinero que los ciudadanos deben pagar al estado para sostener el gasto público.
* Stakeholders: Son las personas involucradas en el proyecto, ej. El cliente, los usuarios finales, el equipo de desarrollo, etc.
* XML: Extensible Markup Language, es un lenguaje de almacenamiento y transporte de datos.
* SOAP: Servicio de arquitectura orientado a servicios web.
* Firebase: Plataforma para el desarrollo de aplicaciones web.
* Google Cloud: Es un gestor de almacenamiento en la nube.
* MB: Mega Byte.
* MD5: Message-Digest Algothm 5, es un algoritmo que permite encriptar un mensaje.

# Descripción General

Este SW permitirá al cliente gestionar sus ventas y llevar un control sobre los impuestos del IVA que debe tributar al estado, adicionalmente la aplicación podrá gestionar el inventario, de igual manera se tomará en cuenta los stakeholders involucrados en el uso del SW,

Resultados de entrevistas:

## Perspectiva del producto



## Funcionalidades del Producto

El software se encargará de gestionar las ventas que realice una empresa pequeña de Colombia, además, calculará el IVA que debe tributar al estado y administrara el inventario de la empresa.

### Consultar factura. Este módulo permite al cliente, administrador y cajero/a verificar la información sobre la compra realizada, ej. Valor de la compra, fecha de la compra, IVA.

### Consultar IVA de las facturas semanales. Permite visualizar el cálculo del IVA basado en las ventas realizadas cada semana.

### Editar la factura (Por motivos de cambio o error de facturación). Permite al administrador realizar cambios en una factura en caso de que haya algún error dentro de una compra.

### Cambiar estado de la factura. Permite cambiar el estado de una factura entre anulado/aprobado en caso que un cliente realice una devolución o un cambio.

### Crear la factura (Cuando realiza una venta). Permite generar un nueva factura.

### Enviar factura al servidor (Al crear la factura, se enviará automáticamente al servidor). Permite enviar las facturas generadas en formato XML al servidor de la DIAN anualmente.

## Características de los Actores

|  |  |
| --- | --- |
| Actor | Descripción |
| Vendedor | Es la persona encargada de facturar al cliente por las compras realizadas. |
| Administrador | Es el encargado de gestionar las facturas del sistema, tales como, crear factura, editar estado de factura, etc. |
| Cliente | Podrá visualizar las facturas de sus compras. |
| DIAN | Es la entidad encargada de recibir las facturas y verificar si cumplen con los estándares del decreto 2242 de 2015 |

## Restricciones Generales

El cliente no cuenta con una base de datos, por lo que el software proveerá este servicio para la administración del inventario.

# Requerimientos No Funcionales

## Interfaz Externa

< Cuando hay requerimientos que no son necesariamente de software o requerimientos que no cuentan como peticiones funcionales >

### Interfaz de usuario

< Requerimientos particulares de interfaz gráfica (ejemplo: diseño responsivo)>

### Interfaz de Hardware

* El usuario debe contar con un dispositivo móvil (ejemplo: Tablet, Smartphone, etc.)
* La aplicación estará disponible para sistemas operativos Android e IOS.

< Periféricos o hardware especializado>

### Interfaces de Software

* La aplicación se conectará a los servidores de la DIAN para enviar las facturas de las ventas realizadas en un formato XML, y serán enviadas mediante protocolo SOAP.
* La aplicación se conectará a la base de datos creada en Firebase y Google Cloud.

< Conexiones con otros sistemas >

### Interfaces de comunicación

* El formato de intercambio de datos con la aplicación del cliente debe ser XML.
* La aplicación usará el protocolo HTTP/HTTPS para conectarse a internet.

< Conexiones con sistemas móviles, internet o incluso comunicación interna >

## Usabilidad

* El sistema debe ser intuitivo con el fin de facilitar y reducir el tiempo de capacitación sobre este.
* La aplicación utilizará una paleta de colores que sea agradable para el usuario.

## Desempeño

* El sistema debe recuperar la factura del usuario y mostrarla en menos de 3 segundos.
* El módulo login debe permitir el ingreso al usuario en menos de 3 segundos.
* La aplicación tendrá un límite de memoria caché de 200MB.

## Seguridad

* Las contraseñas de la aplicación serán encriptadas mediante el algoritmo MD5.
* Las contraseñas en la aplicación deben estar ocultas.

## Confiabilidad

* El usuario no puede experimentar más de dos fallas por mes en la aplicación.
* El sistema debe realizar una copia de seguridad de los datos almacenados cada día.
* El software debe seguir funcionando incluso cuando la aplicación no tenga acceso a internet.

## Disponibilidad

* La aplicación debe estar disponible 24/7.
* La aplicación debe soportar «cinco nueves» en disponibilidad: Esto significa que la aplicación estará disponible un 99,999% del tiempo al año. Indica que la aplicación no puede estar caída por más de 5,26 minutos al año.

## Mantenibilidad

* La aplicación está diseñada para tener un fácil mantenimiento, ya que cuenta con módulos independientes.
* El tiempo promedio para reparar un error de capa 2 en el SW debe no ser mayor a 8 horas.

## Portabilidad

* La aplicación estará disponible para sistemas operativos Android e IOS.
* La aplicación se conectará a los servidores de la DIAN.

## Escalabilidad

* La aplicación permite mantener a 5 usuarios activos al mismo tiempo

# Requerimientos Funcionales

## Lista de Casos de Uso

1. Como administrador necesito **consultar factura** para verificar la compra.
2. Como administrador necesito **consultar IVA** para saber cuál es el valor que debo tributar a la DIAN.
3. Como administrador necesito **cambiar estado de una factura** para evitar que genere un valor de una compra cancelada.
4. Como vendedor necesito **consultar factura** para validar los datos de la compra.
5. Como vendedor necesito **crear una factura** para registrar una compra.
6. Como vendedor necesito **enviar una factura al servidor** para guardar el registro.
7. Como vendedor necesito **enviar una factura al cliente** para guardar registro de sus ventas.
8. Como administrador necesito **iniciar sesión** para poder acceder al sistema.
9. Como vendedor necesito **iniciar sesión** para poder acceder al sistema.
10. El sistema debe **verificar el IVA** de las facturas.
11. El sistema debe **actualizar el XML** cuando el administrador cambie el estado de la factura.
12. El sistema debe **enviar reporte** a la DIAN.
13. El sistema debe **guardar facturas** generadas.
14. El sistema debe **crear un XML** anualmente.

## Diagrama de Casos de Uso



## Documentación de Casos de Uso

|  |  |
| --- | --- |
| ID Caso de uso: CU-004 | Nombre: Consultar facturas como vendedor. |
| **Descripción** | El vendedor de la panadería accede al sistema (validando su sesión) para consultar las facturas que hay en la panadería y validar los datos de las compras |
| **Actores** | Los actores que hacen parte de este caso de uso es el vendedor de la panadería (Actor principal) y el software del sistema (Actor secundario). |
| **Entradas y Pre-condiciones** | * PRE-1: El vendedor de la panadería debe estar inscrito en la base de datos del sistema. * PRE-2: El vendedor debe de validar su sesión en el sistema. |
| **Procesamiento** | 1. **Validar datos de las compras con las facturas**  * El vendedor debe de ingresar al sistema validando sus datos. * El vendedor debe de ingresar a la sección de facturas para ver cada una de ellas. |
| **Salidas** | El sistema le mostrara al vendedor un mensaje con todos los ítems comprados en la factura. |
| **Excepciones** | Cuando no hay facturas registradas en la aplicación, mostrara una alerta al vendedor informándole que aún no hay facturas registradas. |

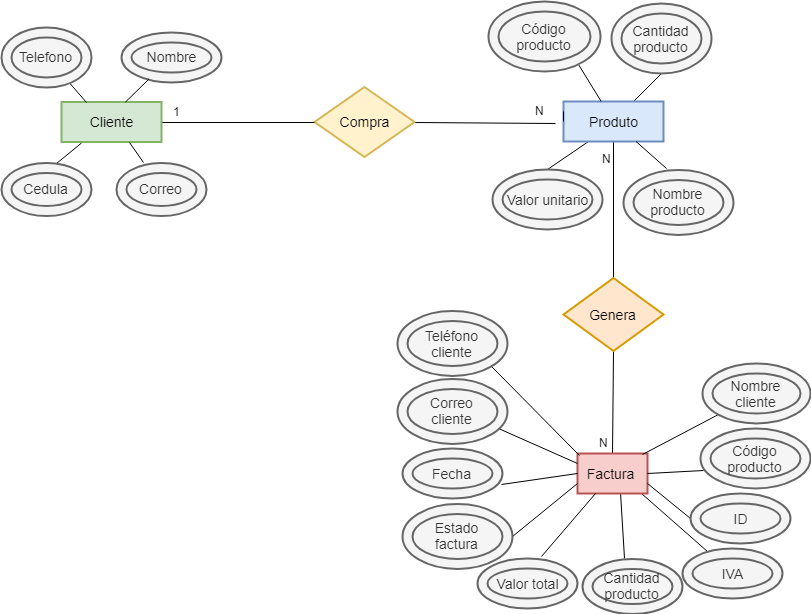
|  |  |
| --- | --- |
| ID Caso de uso: CU-005 | Nombre: Crear factura |
| **Descripción** | El vendedor de la panadería accede al sistema (validando su sesión) para crear una factura y validar la compra del cliente. |
| **Actores** | Los actores que hacen parte de este caso de uso es el vendedor de la panadería (Actor principal) y el software del sistema (Actor secundario). |
| **Entradas y Pre-condiciones** | * PRE-1: El vendedor de la panadería debe estar inscrito en la base de datos del sistema. * PRE-2: El vendedor debe de validar su sesión en el sistema. |
| **Procesamiento** | 1. **Crear la factura al cliente.**  * El vendedor debe de ingresar al sistema validando sus datos. * El vendedor valida los ítems seleccionados por el cliente y procede a ingresarlos en el sistema para crear la factura en el sistema. |
| **Salidas** | El sistema le mostrara al vendedor la confirmación de la creación de la factura. |
| **Excepciones** | Cuando el vendedor se equivoca al ingresar los ítems en la factura, esta tiene un botón de reiniciar la factura apenas es creada, para su pronta modificación. |

|  |  |
| --- | --- |
| ID Caso de uso: CU-006 | Nombre: Enviar factura al servidor. |
| **Descripción** | El vendedor de la panadería accede al sistema (validando su sesión) para enviar las facturas al servidor. |
| **Actores** | Los actores que hacen parte de este caso de uso es el vendedor de la panadería (Actor principal) y el software del sistema (Actor secundario). |
| **Entradas y Pre-condiciones** | * PRE-1: El vendedor de la panadería debe estar inscrito en la base de datos del sistema. * PRE-2: El vendedor debe de validar su sesión en el sistema. |
| **Procesamiento** | 1. **Enviar factura al servidor.**  * El vendedor debe de ingresar al sistema validando sus datos. * El vendedor valida ingresa a la lista de las facturas del día y procede a enviar las facturas al sistema (con un método de sincronización del sistema). |
| **Salidas** | El sistema le mostrara al vendedor la confirmación de la sincronización de las facturas del día. |
| **Excepciones** | En el caso de que la vendedora olvide realizar la sincronización de las facturas diarias con el sistema, el sistema almacena las facturas para ser sincronizadas el día siguiente. |

|  |  |
| --- | --- |
| ID Caso de uso: CU-007 | Nombre: Enviar factura al cliente. |
| **Descripción** | El vendedor de la panadería accede al sistema (validando su sesión) para enviar las facturas al cliente. |
| **Actores** | Los actores que hacen parte de este caso de uso es el vendedor de la panadería (Actor principal) y el software del sistema (Actor secundario). |
| **Entradas y Pre-condiciones** | * PRE-1: El vendedor de la panadería debe estar inscrito en la base de datos del sistema. * PRE-2: El vendedor debe de validar su sesión en el sistema. * PRE-3: El cliente debe ser registrado en el sistema o ya estar en el sistema. |
| **Procesamiento** | 1. **Enviar factura al cliente.**  * El vendedor debe de ingresar al sistema validando sus datos. * El vendedor valida ingresa a la lista de las facturas del día y procede a enviar las facturas al cliente (buscando al cliente en el sistema). |
| **Salidas** | El sistema identificara al cliente y enviara la factura vía E.-mail al usuario que lo solicite. |
| **Excepciones** | No existen excepciones en este caso de uso. |

# Modelos de Diseño

## Modelo lógico / de relación

Modelo entidad relación

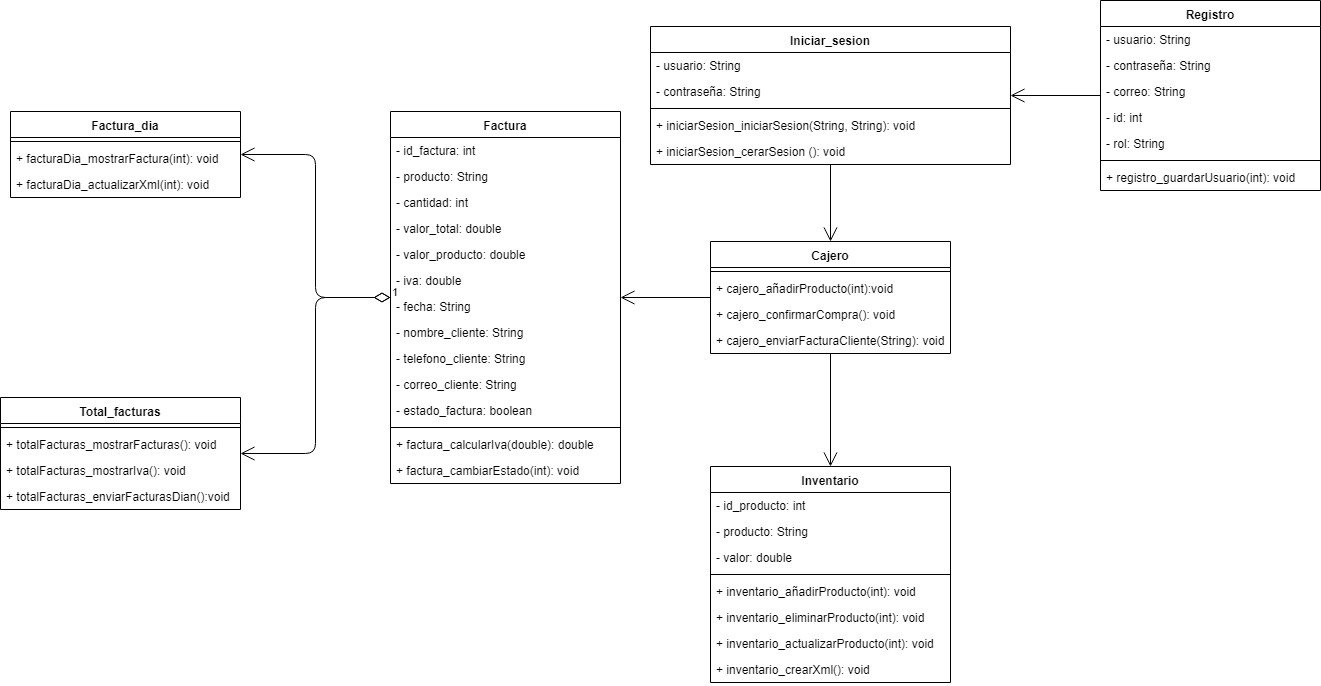


Diagrama de clases

## Modelo De Arquitectura

< Arquitectura general de toda la aplicación>

Ejemplo de diagrama de arquitectura: este es un diagrama por capas en el que además de mencionar las capas (como en el ejemplo), se debe mencionar los componentes específicos y tecnologías utilizadas. Por ejemplo, si en la capa de datos se utilizará Oracle, esta herramienta debe mencionarse como parte del diagrama.

<https://arevalomaria.files.wordpress.com/2011/03/componentes.png?w=840>

## Modelos dinámicos

< Considere los modelos necesarios para entender el flujo del sistema (diagramas de flujo, de estados, etc.). Se puede presentar únicamente un diagrama de procesos del sistema en general. Los demás diagramas son opcionales>

Ejemplo de diagrama de procesos: <https://d2slcw3kip6qmk.cloudfront.net/marketing/blog/2017Q4/agile-lifecycle/user-journey-flow-template.png>

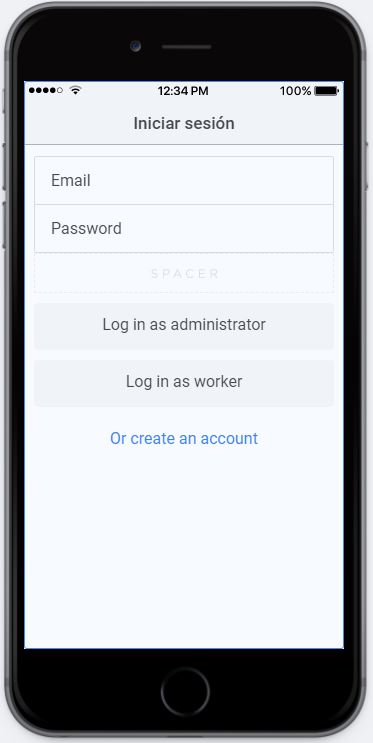
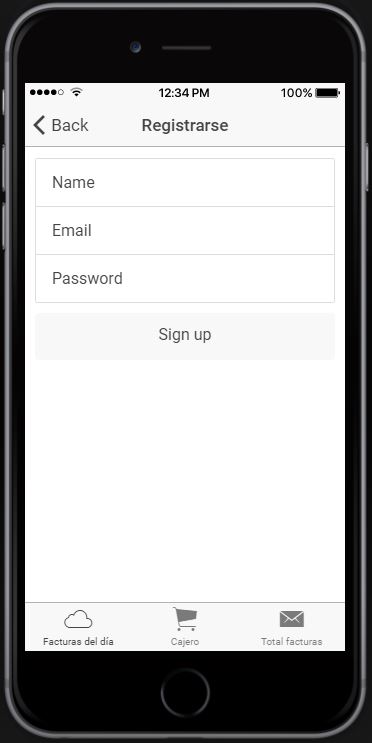
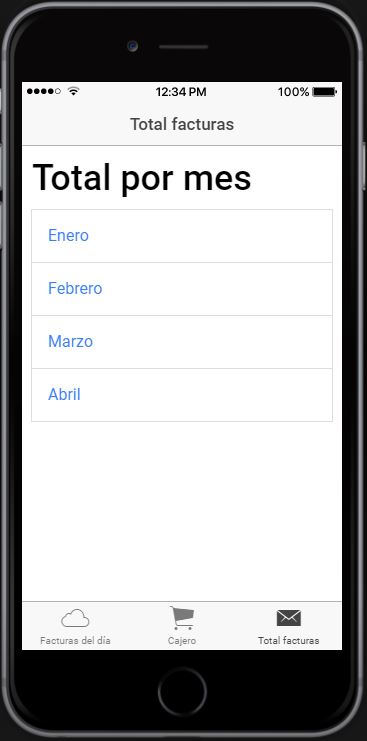
## Modelos de Infraestructura

< Considere los modelos de despliegue, comunicación o de infraestructura en general donde se represente los componentes del sistema >

Ejemplo de diagrama de despliegue: <https://sites.google.com/site/softwarearchitecturedocument/5-otras-vistas/5-1-vista-fisica>

# Diseño de Interfaces





# Inspección

< Explique el proceso que se llevó a cabo para realizar la inspección de este documento. Presente evidencias.>