Texto

El contenido generado por IA puede ser incorrecto.

**Instituto Tecnológico de Costa Rica**

**Escuela de Computación**

**IC-7841 Proyecto de Diseño de Software**

**Grupo: 60**

**2025, Semestre I**

**Profesor:** Luis Diego Noguera Mena

**Estudiante:**

Andrew López Herrera - 2021062132

**Asignación:**

Documento de diseño del software

**Fecha de entrega:**

Viernes 4 de abril, 2025

# Contenido

[Contenido 1](#_Toc194702397)

[Tabla de Imágenes 3](#_Toc194702398)

[1. Fecha de la versión y estatus 4](#_Toc194702399)

[2. Organización 4](#_Toc194702400)

[3. Autor(es) 4](#_Toc194702401)

[4. Historia de cambios 4](#_Toc194702402)

[5. Introducción 4](#_Toc194702403)

[a. Propósito 4](#_Toc194702404)

[b. Alcance 5](#_Toc194702405)

[c. Contexto 6](#_Toc194702406)

[d. Resumen 7](#_Toc194702407)

[6. Referencias 8](#_Toc194702408)

[7. Glosario 8](#_Toc194702409)

[8. Interesados 9](#_Toc194702410)

[9. Perspectivas de diseño 10](#_Toc194702411)

[a. Contexto 10](#_Toc194702412)

[b. Composición 27](#_Toc194702413)

[c. Lógica 31](#_Toc194702414)

[d. Dependencias 37](#_Toc194702415)

[e. Información 38](#_Toc194702416)

[f. Uso de patrones 39](#_Toc194702417)

[g. Interfaces 39](#_Toc194702418)

[h. Interfaz de usuario 39](#_Toc194702419)

[i. Estructura 40](#_Toc194702420)

[j. Interacción 40](#_Toc194702421)

[k. Dinámica de estados 40](#_Toc194702422)

[l. Algoritmos 40](#_Toc194702423)

[m. Recursos 45](#_Toc194702424)

[10. Apéndice - Alternativas de diseño 47](#_Toc194702425)

# Tabla de Imágenes

[Diagrama 1 - Diagrama de contexto del sistema SRLGestock [1] 10](#_Toc194701601)

[Diagrama 2 - Casos de uso de la gestión usuario [1] 11](#_Toc194701602)

[Diagrama 3 - Casos de uso de la gestión de facturas [1] 14](#_Toc194701603)

[Diagrama 4 - Caso de uso de gestión de la empresa [1] 20](#_Toc194701604)

[Diagrama 5 - Casos de uso de la gestión del inventario. [1] 21](#_Toc194701605)

[Diagrama 6 - Diagrama de componente del sistema SRL Gestock [1] 27](#_Toc194701606)

[Diagrama 7 - Diagrama de clases para gestionar información de la empresa [1] 31](#_Toc194701607)

[Diagrama 8 - Diagrama de clases de la lógica de factura 33](#_Toc194701608)

[Diagrama 9 - Diagrama de clases de la lógica del producto [1] 34](#_Toc194701609)

[Diagrama 10 - Diagrama de clases de la lógica del usuario [1] 36](#_Toc194701610)

[Diagrama 11 - Diagrama entidad relación de la base de datos [1] 38](#_Toc194701611)

[Diagrama 12 - Diagrama de flujo sobre el proceso de facturación electrónica [1] 40](#_Toc194701612)

[Diagrama 13 - Diagrama de flujo del proceso de inicio de sesión 43](#_Toc194701613)

[Diagrama 14 - Diagrama de implementación del sistema SRLGestock 45](#_Toc194701614)

[Diagrama 15 - Diagrama de clases de la primera propuesta alternativa de la facturación 48](#_Toc194701615)

[Diagrama 16 - Diagrama de clases de la segunda propuesta alternativa de la facturación 49](#_Toc194701616)

# Fecha de la versión y estatus

**Fecha:** 04 de abril del 2025

**Estado:** Revisado

# Organización

No aplica

# Autor(es)

**Autor:** Andrew Denilson López Herrera

# Historia de cambios

No aplica

# Introducción

## Propósito

El propósito de crear esta aplicación es para automatizar los procesos de guardado de inventario y facturación. Actualmente, el cliente, SRL Technology, deben manejar el inventario con una hoja de cálculo, por lo que, toma más tiempo realizar una compra en el local, ya que se debe verificar el stock desde la hoja de cálculo. Además, la facturación electrónica se debe de realizar de manera manual, por lo que se tarda aún más una compra.

Se desea realizar una aplicación que automatice estos procesos y las compras se hagan de manera rápida. Además, se puede tener un mejor control del inventario, ya que es el sistema quien maneja estos datos, por lo que se hace sencilla la tarea. Esta aplicación es importante para la empresa, ya que, esta espera crecer con el tiempo y una hoja de cálculo no va a ser la mejor herramienta para trabajar con varios empleados. En caso de que la empresa decida tener más sucursales la hoja de cálculo tampoco va a ser de gran ayuda, debido a que no se puede tener un control central de los productos que entran y salen de las sucursales.

El cliente, SRL Technology, recalcó la importancia que tiene que la aplicación haga las facturas electrónicas, por lo que, la facturación electrónica es una de las razones más importantes para que esta aplicación sea creada. Se debe hacer que este proceso sea poco tedioso y mostrarse amigable con el usuario.

## Alcance

Este producto podrá realizar tareas de gestión de inventarios y facturación de la empresa SRL Technology, el objetivo de esto es automatizar los procesos de gestión de inventarios y facturación de la empresa, ahorrando así más tiempo en realizar tareas engorrosas y tardadas. Actualmente, se tiene planeado que el sistema realice cuatro principales actividades en la empresa.

La primera actividad es gestionar los usuarios que realizan actividades de gestión de inventarios y facturación, para ello, el programa debe ser capaz de crear usuarios e iniciar sesión en la aplicación. Además, con esto se evita que la aplicación pueda ser accedida por terceros no autorizados y también se protege la integridad de los datos.

La segunda actividad es gestionar el inventario de la empresa, para ello se utilizará una base de datos que almacene la información del inventario de la empresa. Para ello, se dará al usuario una interfaz gráfica por el que pueda gestionar el inventario. Esta gestión le permitirá al usuario crear, mostrar, modificar y eliminar los productos registrados.

La tercera actividad es gestionar la facturación de la empresa. La empresa requiere realizar facturas, por lo que se le dará al usuario la interfaz gráfica para que pueda facturar n productos con m unidades por producto. De esta manera, se puede realizar una gestión automatizada no solo en las facturas, también el inventario, ya que las facturas creadas reducirán el inventario actual de manera automática.

Por último, la cuarta actividad consiste en crear facturas electrónicas y enviarlas al Ministerio de Hacienda. Este proceso es tedioso, por lo que el sistema debe crear toda la estructura de los archivos de manera automática y verificar todo el proceso de facturación sea válido.

## Contexto

La aplicación debe tener un cliente de escritorio de PC para Windows 10/11 de arquitectura x64 que debe ser capaz de conectarse con el backend para realizar las operaciones de facturación y control de inventario de las tiendas de SRL Technology. Esta aplicación será construida con el framework Tauri, debido que permite el uso multiplataforma.

La aplicación debe tener un backend para gestionar las operaciones que solicite el cliente. Este backend será desarrollado en Deno. Se montará sobre un servidor virtual de hosting506, utilizando, inicialmente, el plan GLORIA VPS S que ofrece un servidor un barato y lo suficientemente potente para las tareas que hará la empresa inicialmente. Se utilizará la empresa hosting506 por la comunicación directa que se da entre personas ticas. El servidor ejecutará Ubuntu Server en su versión 22.04, ya que no tiene costo para SRL Technology y es eficiente en el uso de recursos, además es ligero y seguro.

El motor de la base de datos que se utilizará PostgreSQL, debido a que es un motor de licencia libre, este permite que la empresa pueda realizar operaciones sin tener que pagar por licencias costosas. Este se ejecutará, por el momento, en el mismo servidor que utiliza el backend, ya que, la empresa no registra grandes movimientos de datos y se pueden utilizar ambos ambientes, backend y motor de bases de datos, en un solo computador.

Por último, el backend debe conectarse con el Ministerio de Hacienda para realizar las facturas electrónicas, por lo que debe usarse la API que ofrece dicho ministerio. Se utilizará la versión más reciente.

## Resumen

En este documento se podrán encontrar los diseños necesarios para la creación del sistema SRL Gestock. Se pueden encontrar los diagramas de casos de uso que se lograron identificar a partir de la reunión con el cliente, además, se pueden encontrar las descripciones de estos casos de usos debajo de cada diagrama de casos uso. Este diagrama muestra el contexto del programa.

El documento también muestra los diagramas componentes que ayudan al lector a comprender la composición del programa. Además, se pueden observar una descripción de cada componente que tiene el programa.

También se muestra los diagramas de clase del backend, ya que es necesario mantener un sistema seguro, por lo tanto, se decidió realizar el diagrama de clases para crear un diseño seguro en el backend.

El diagrama entidad-relación se encuentra en el documento para mostrar como es la estructura de la base de datos. Es importante realizar este diagrama, debido a que es necesario guardar la información en algún lugar. La información que se respalda en esta base de datos es importante, además debe ser fácil de recuperar por el backend, de ahí la necesidad del modelado de la base de datos.

Por último, se puede encontrar el diagrama de implementación del sistema, ya que es un sistema con la arquitectura frontend-backend, es necesario definir cuales son los dispositivos que se utilizarán en el sistema, además de cuál debe de ser los requisitos mínimos que deben tener estos dispositivos. Por esa razón, se decidió diseñar la implementación del sistema.

|  |  |
| --- | --- |
| [1] | A. D. L. Herrera, «Lucidchart,» 4 abril 2025. [En línea]. Available: <https://lucid.app/lucidchart/888488bd-9829-4e55-bbcf-928ed8797b81/edit?viewport_loc=85%2C-134%2C1616%2C752%2C-Fv6-YT0Orcn&invitationId=inv_80701699-c076-4875-b9b4-0db66c4a3ca6>. |

# Referencias

# Glosario

**Backend**: Parte del software que maneja la lógica de negocio, bases de datos y servidores.

**Frontend**: Parte del software con la que interactúa el usuario, como la interfaz gráfica.

**Base de datos**: Sistema organizado para almacenar y gestionar datos.

**HTTPS**: Protocolo seguro de comunicación en la web, usa cifrado SSL/TLS.

**Ruta**: Dirección específica de URL que el servidor maneja.

**Enrutador**: Componente que asigna solicitudes a rutas o controladores en una aplicación.

**Clase**: Plantilla para crear objetos con propiedades y métodos en programación orientada a objetos.

**Interface**: Contrato que define métodos que una clase debe implementar sin especificar cómo.

**Diagrama de clases**: Representación de clases, sus atributos, métodos y relaciones.

**Diagrama de flujos**: Representación del flujo de control o proceso en un sistema.

**Diagrama de contexto**: Diagrama que muestra el sistema y su interacción con entidades externas.

**Diagrama de componente**: Diagrama que describe cómo se organizan los componentes de un sistema.

**Diagrama de casos de uso**: Diagrama que describe las interacciones entre usuarios y el sistema.

**Descripción de casos de uso**: Explicación detallada de cómo se realiza una tarea en un sistema.

**Lucidchart**: Herramienta en línea para crear diagramas y gráficos.

**IA (Inteligencia Artificial)**: Tecnología que permite a las máquinas realizar tareas que requieren inteligencia humana.

**Servidor**: Computadora o software que proporciona servicios a otros dispositivos en una red.

**Puerto**: Punto de acceso lógico para la transmisión de datos en una red.

**IP (Dirección IP)**: Dirección única que identifica a un dispositivo en una red.

**Procesador**: Componente de la computadora que ejecuta instrucciones y procesa datos.

**RAM (Memoria de Acceso Aleatorio)**: Memoria volátil que almacena datos temporales que el procesador necesita.

**Sistema operativo**: Software que gestiona los recursos de hardware y proporciona servicios a las aplicaciones.

**API (Interfaz de Programación de Aplicaciones)**: Conjunto de reglas que permite que una aplicación interactúe con otra.

# Interesados

**SRL Technology:** Empresa dueña del software, es quien necesita el sistema para realizar operaciones de facturación y gestión de inventario.

**Desarrolladores:** Son la o las personas que necesitan conocer el diseño del software para saber como implementarlo.

# Perspectivas de diseño

## Contexto

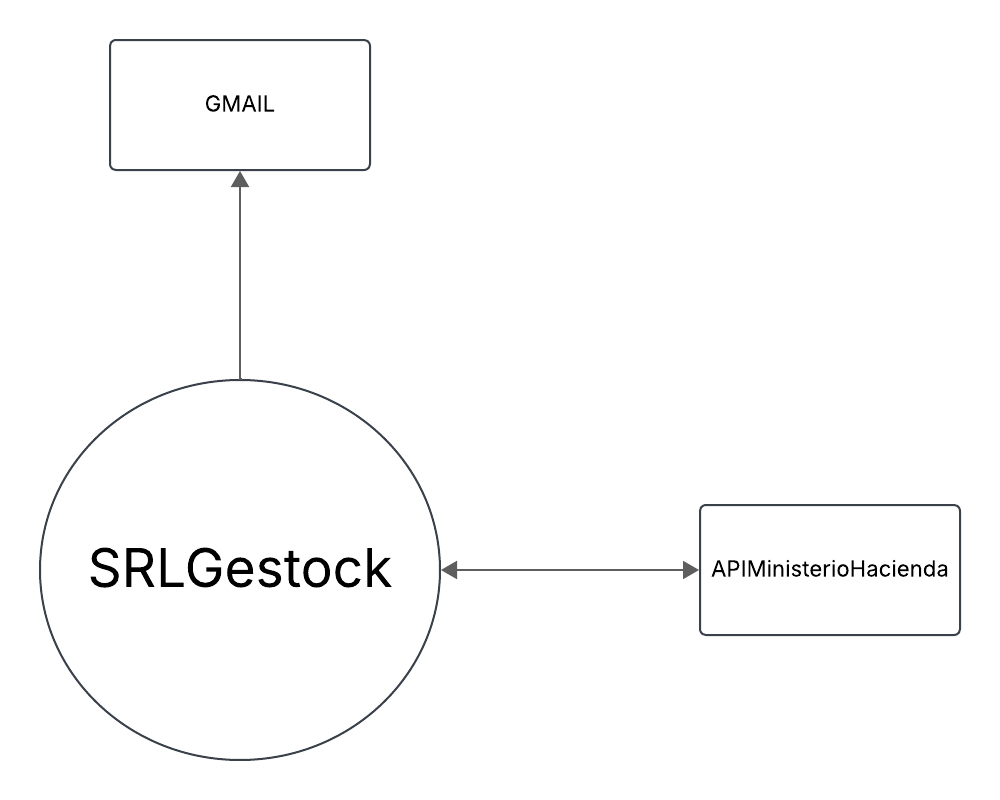


Diagrama 1 - Diagrama de contexto del sistema SRLGestock [1]

Este diagrama representa el sistema de SRLGestock interactuando con sistemas externos para realizar las acciones necesarias para que se pueda trabajar. A continuación, se explica cada a que se refiere cada entidad.

* **SRLGestock:** El sistema creado para la empresa SRL Technology, se encarga de gestionar el inventario y las facturas.
* **APIMinisterioHacienda:** El sistema de facturación electrónica del Ministerio de Hacienda, se encarga de aprobar las facturas electrónicas generadas.
* **GMAIL:** El sistema de correos electrónicos que se encarga de enviar los mensajes necesarios a los usuarios.

Con la intención de mostrar de manera más detallada el contexto de la aplicación se procede a mostrar los diagramas de caso de uso y las descripciones de cada uno para de los casos de usos. El caso de uso de **Iniciar Sesión** que se muestra en todos los diagramas de casos de uso se refiere al mismo caso de uso, pero con el objetivo de mostrar diagramas de casos de uso más visibles, se separaron en varias partes.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama 2 - Casos de uso de la gestión usuario [1]

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-001 |
| Nombre | Iniciar Sesión |
| Autor | Andrew Denilson López Herrera |
| Fecha | 1 de abril del 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Esta opción permite al usuario ingresar con su usuario al sistema SRL Gestock. | |
| Precondiciones:  1. Haber creado un usuario anteriormente.  2. Tener internet.  3. Iniciar la aplicación manualmente. | |
| Flujo normal:  1. Escribir el correo electrónico asociado al usuario.  2. Escribir la contraseña enviada por el sistema o creada por el usuario.  3. Presionar el botón iniciar sesión.  4. Se ingresa al menú principal. | |
| Flujo alternativo:  4. Las credenciales son incorrectas.  5. El usuario ingresa las credenciales correctas.  6. Presionar el botón iniciar sesión.  7. Se ingresa al menú principal.  -----------------------------------------------------------------------------------------------------------  4. El usuario no existe.  5. El usuario solicita un nuevo usuario.  6. El usuario vuelve a realizar el flujo normal.  -----------------------------------------------------------------------------------------------------------  1. El usuario no recuerda la contraseña.  2. Se presiona el botón de olvidé contraseña.  3. Se llena el formulario. 4. El usuario realiza el flujo normal. | |
| Postcondiciones:  El sistema crea una nueva sesión de usuario. | |
| Excepciones:  1. No hay internet, no se puede acceder al sistema. | |

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-002 |
| Nombre | Recuperar contraseña |
| Autor | Andrew Denilson López Herrera |
| Fecha | 1 de abril del 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Esta opción permite al usuario recibir una nueva contraseña en caso de que la haya perdido. | |
| Precondiciones:  1. Haber creado un anteriormente.  2. Tener conexión a internet.  3. Haber presionado el botón olvidé la contraseña. | |
| Flujo normal:  1. El usuario ingresa el correo electrónico.  2. Presionar el botón de confirmación de correo.  3. Enviar el correo electrónico con la contraseña aleatoria.  4. Iniciar sesión con la nueva contraseña. | |
| Flujo alternativo:  3. El correo electrónico del usuario no existe.  4. Crear un nuevo usuario. | |
| Postcondiciones:  1. El sistema registra la nueva contraseña aleatoria. | |
| Excepciones:  1. No hay conexión a internet, no se puede recuperar la contraseña. | |

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-003 |
| Nombre | Crear usuario |
| Autor | Andrew Denilson López Herrera |
| Fecha | 1 abril del 2025 |
| Actores | Administrador |
| Descripción: Esta función permite a los administradores crear un nuevo usuario en el sistema SRL Gestock. | |
| Precondiciones:  1. Haber iniciado sesión con una cuenta de administrador. | |
| Flujo normal:  1. El administrador debe ingresar al menú de administración.  2. El administrador debe seleccionar la opción de crear usuario.  3. El administrador debe llenar el formulario del nuevo usuario.  4. El administrador envía los datos del usuario.  5. El sistema confirma la creación del usuario. | |
| Flujo alternativo:  5. El sistema indica que el correo electrónico ya está inscrito en el sistema.  6. El administrador indica otro correo electrónico. | |
| Postcondiciones:  1. El sistema envía la contraseña al usuario por correo electrónico. | |
| Excepciones:  1. No hay conexión a internet, no se puede crear el usuario. | |

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama 3 - Casos de uso de la gestión de facturas [1]

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-004 |
| Nombre | Ver facturas |
| Autor | Andrew Denilson López Herrera |
| Fecha | 1 de abril del 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Permite a los usuarios ver las facturas que se han creado en la tienda. | |
| Precondiciones:  1. Haber iniciado sesión. | |
| Flujo normal:  1. Se presiona el botón para ver facturas.  2. Se muestran las facturas al usuario. | |
| Flujo alternativo:  Ninguno | |
| Postcondiciones:  Ninguno | |
| Excepciones:  1. No hay conexión a internet, no se pueden observar las facturas. | |

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-005 |
| Nombre | Ingresar producto a factura. |
| Autor | Andrew Denilson López Herrera |
| Fecha | 1 de abril del 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Esta función permite ingresar los productos del inventario a una factura. | |
| Precondiciones:  1. Haber iniciado sesión.  2. Haber empezado una facturación. | |
| Flujo normal:  1. El usuario ingresa el nombre o el consecutivo del producto.  2. Le aparecen los productos con la similitud parcial o exacta.  3. El usuario selecciona el producto deseado.  4. El producto se ingresa a la factura.  5. Se escoge la cantidad de producto que se desea escoger. | |
| Flujo alternativo:  Ninguna | |
| Postcondiciones:  Ninguna | |
| Excepciones:  1. No hay conexión a internet, no se puede buscar el produto.  2. El producto no existe en el inventario.  3. El producto no tiene suficiente stock disponible. | |

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-006 |
| Nombre | Enviar factura por correo |
| Autor | Andrew López Herrera |
| Fecha | 1 de abril del 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Esta opción permite a los usuarios o administradores programar un envío de una factura por medio de correo electrónico a un cliente de SRL Technology. | |
| Precondiciones:  1. Tener conexión a internet.  2. Haber inicializado una factura. | |
| Flujo normal:  1. Presionar el checkbox de para ingresar un cliente.  2. Llenar el formulario del cliente.  3. Facturar la compra. | |
| Flujo alternativo:  Ninguna. | |
| Postcondiciones:  1. Esperar que se reciba la respuesta de la factura electrónica a hacienda.  2. Generar el PDF con los detalles de la compra.  3. Enviar el PDF de la factura, el XML de la factura y las repuesta XML de hacienda al correo electrónico del cliente. | |
| Excepciones:  1. No hay internet, no se puede facturar.  2. El correo electrónico de la persona no existe.  3. El sistema de correos electrónicos no sirve. | |

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-007 |
| Nombre | Buscar facturas |
| Autor | Andrew Denilson López Herrera |
| Fecha | 1 de abril del 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Esta función permite a los usuarios buscar facturas por número de factura. | |
| Precondiciones:  Ninguna | |
| Flujo normal:  1. Se selecciona ver facturas.  2. Se pone el número de factura en la entrada de búsqueda.  3. Se presiona buscar.  4. Se encuentra la factura.  5. Se presiona para observar los detalles. | |
| Flujo alternativo:  4. No se encuentra la factura, debido a que se ingresó incorrectamente el número de factura.  5. Se vuelve a ingresar el número de factura correctamente.  6. Se verifica que se haya encontrado la factura, si no, se vuelve a repetir el flujo alterno. | |
| Postcondiciones:  Ninguna. | |
| Excepciones:  1. No hay internet, no se pueden encontrar facturas. | |

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-008 |
| Nombre | Facturar compra |
| Autor | Andrew Denilson López Herrera |
| Fecha | 2 de abril del 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Permite a los usuarios facturar los artículos seleccionador en la compra. | |
| Precondiciones:  1. Tener conexión a internet.  2. Haber iniciado una facturación. | |
| Flujo normal:  1. Se presiona facturar.  2. Se muestra la vista previa de la factura.  3. Se completa la facturación. | |
| Flujo alternativo:  Ninguno. | |
| Postcondiciones:  1. Se rebaja la cantidad de productos del inventario.  2. Se registra la factura en la base de datos.  3. Se envía la factura XML al Ministerio de Hacienda.  4. Se recibe el XML de respuesta del Ministerio de Hacienda.  5. Se envía el correo electrónico con la factura al cliente, si es que se seleccionó la opción. | |
| Excepciones:  1. No hay conexión a internet, no se puede crear la factura.  2. El sistema del Ministerio de Hacienda no responde. | |

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-009 |
| Nombre | Imprimir facturas. |
| Autor | Andrew Denilson López Herrera |
| Fecha | 1 de abril de 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Permite al usuario imprimir la factura de la compra que ha realizado la persona. | |
| Precondiciones:  1. Haber facturado la compra. | |
| Flujo normal:  1. Seleccionar la impresora.  2. Presionar el botón de imprimir factura.  3. Esperar a que imprima la factura.  4. Entregar la factura. | |
| Flujo alternativo:  3. La impresora no responde.  4. Reiniciar la impresora o conectar otra impresora  5. Volver a realizar el flujo normal. | |
| Postcondiciones:  Ninguna. | |
| Excepciones:  1. No hay una impresora conectada. | |

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama 4 - Caso de uso de gestión de la empresa [1]

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-010 |
| Nombre | Actualizar datos de la empresa. |
| Autor | Andrew Denilson López Herrera |
| Fecha | 2 de abril de 25 |
| Actores | Administrador. |
| Descripción:  Esta opción permite cambiar los datos de la empresa que se muestran en la factura. | |
| Precondiciones:  1. Tener conexión a internet.  2. Haber iniciado sesión como administrador. | |
| Flujo normal:  1. En el menú principal, presionar el botón de administración.  2. Presionar el botón datos de la empresa.  3. Cambiar los datos del formulario.  4. Presionar guardar.  5. Se muestra un mensaje de éxito al cambiar la contraseña. | |
| Flujo alternativo:  5. Hay datos vacíos.  6. Llenar los espacios vacíos | |
| Postcondiciones:  1. Se envían los datos a la base de datos. | |
| Excepciones:  1. No hay internet, no se pueden actualizar los datos. | |

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama 5 - Casos de uso de la gestión del inventario. [1]

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-011 |
| Nombre | Ver productos |
| Autor | Andrew Denilson López Herrera |
| Fecha | 2 de abril del 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Le permite al usuario observar los productos que se encuentra en el inventario en una lista. | |
| Precondiciones:  1. Tener conexión a internet. | |
| Flujo normal:  1. Presionar el botón de inventario en el menú principal.  2. Seleccionar el botón ver productos.  3. Se abre la lista de productos del inventario. | |
| Flujo alternativo:  Ninguno. | |
| Postcondiciones:  Ninguno. | |
| Excepciones:  1. No hay conexión a internet, no se puede observar. | |

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-012 |
| Nombre | Buscar productos |
| Autor | Andrew Denilson López Herrera |
| Fecha | 2 de abril del 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Permite a los usuarios buscar un producto por coincidencia parcial o exacta por medio del consecutivo o el nombre del producto. | |
| Precondiciones:  1. Tener conexión a internet. | |
| Flujo normal:  1. Presionar el botón de inventario en el menú principal.  2. Seleccionar el botón ver productos.  3. Se abre la lista de productos del inventario.  4. Se presiona sobre la entrada de búsqueda de productos.  5. Se escribe el nombre o el consecutivo del producto a buscar.  6. Se presiona el botón buscar.  7. Se recibe la lista de coincidencias. | |
| Flujo alternativo:  Ninguna. | |
| Postcondiciones:  Ninguna. | |
| Excepciones:  1. No hay internet, no se puede buscar el producto. | |

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-013 |
| Nombre | Crear productos |
| Autor | Andrew Denilson López Herrera |
| Fecha | 2 de abril del 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Esta función permite a los usuarios ingresar nuevos productos a la base de datos del SRL Technology. | |
| Precondiciones:  1. Tener conexión a internet. | |
| Flujo normal:  1. En el menú principal, presionar el botón de inventarios.  2. Presionar el botón de crear nuevo producto.  3. Llenar el formulario del producto.  4. Presionar crear producto.  5. Se guarda el producto en la base de datos.  6. Se muestra un mensaje de éxito. | |
| Flujo alternativo:  6. El producto está duplicado. 7. No se necesita ingresar el producto nuevamente.  -----------------------------------------------------------------------------------------------------------  6. El producto está duplicado.  7. Se ingresa un producto con consecutivo diferente. | |
| Postcondiciones:  Ninguna. | |
| Excepciones:  1. No hay conexión a internet, no se puede crear el producto. | |

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-014 |
| Nombre | Actualizar stock |
| Autor | Andrew Denilson López Herrera |
| Fecha | 2 de abril del 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Esta función permite a los usuarios actualizar los valores de los productos. | |
| Precondiciones:  1. Tener conexión a internet. | |
| Flujo normal:  1. En el menú principal, seleccionar inventario.  2. Seleccionar buscar productos.  3. Buscar y seleccionar el producto.  4. Cambiar los valores del formulario.  5. Presionar el botón guardar cambios.  6. Se envía los datos a la base de datos.  7. Se muestra un mensaje de éxito. | |
| Flujo alternativo:  7. El consecutivo es igual a otro producto.  8. Se elimina el producto actual.  -----------------------------------------------------------------------------------------------------------  7. El consecutivo es igual a otro producto.  8. Se escribe un consecutivo diferente. | |
| Postcondiciones:  Ninguna. | |
| Excepciones:  1. No hay conexión a internet, no se puede actualizar el producto. | |

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-015 |
| Nombre | Borrar productos. |
| Autor | Andrew Denilson López Herrera |
| Fecha | 2 de abril del 2025 |
| Actores | Administrador. |
| Descripción:  Esta opción les permite a los administradores eliminar un producto del inventario. | |
| Precondiciones:  1. Tener conexión a internet.  2. Haber iniciado sesión como administrador. | |
| Flujo normal:  1. En el menú principal, seleccionar inventario.  2. Seleccionar buscar productos.  3. Buscar y seleccionar el producto.  4. Presionar el botón eliminar producto.  5. Esperar los tres segundos de confirmación.  6. Ingresar la contraseña de usuario.  7. Enviar la solicitud a la base de datos.  8. Ver mensaje de éxito. | |
| Flujo alternativo:  7. La contraseña es incorrecta.  8. Ingresar nuevamente la contraseña.  9. Volver a presionar el botón eliminar producto. | |
| Postcondiciones:  1. Volver a la búsqueda de productos.  2. Actualizar la lista de productos. | |
| Excepciones:  1. No hay conexión a internet.  2. Se elimina un producto que no existe. | |

|  |  |
| --- | --- |
| Descripción de casos de uso | |
| Identificador | CU-016 |
| Nombre | Ordenar productos. |
| Autor | Andrew Denilson López Herrera |
| Fecha | 2 de abril del 2025 |
| Actores | Usuario o administrador. |
| Descripción:  Les permite a los usuarios ordenar los productos por cantidad de productos vendidos, de manera ascendente, descendente o simplemente no ordenar. | |
| Precondiciones:  1. Tener conexión a internet. | |
| Flujo normal:  1. En el menú principal, seleccionar inventario.  2. Seleccionar buscar productos.  3. Seleccionar el combobox ordenar por.  4. Seleccionar el tipo de ordenamiento.  5. Se presenta la lista ordenada. | |
| Flujo alternativo:  Ninguna. | |
| Postcondiciones:  Ninguna. | |
| Excepciones:  No hay conexión a internet, no se puede solicitar una lista ordenada. | |

## Composición

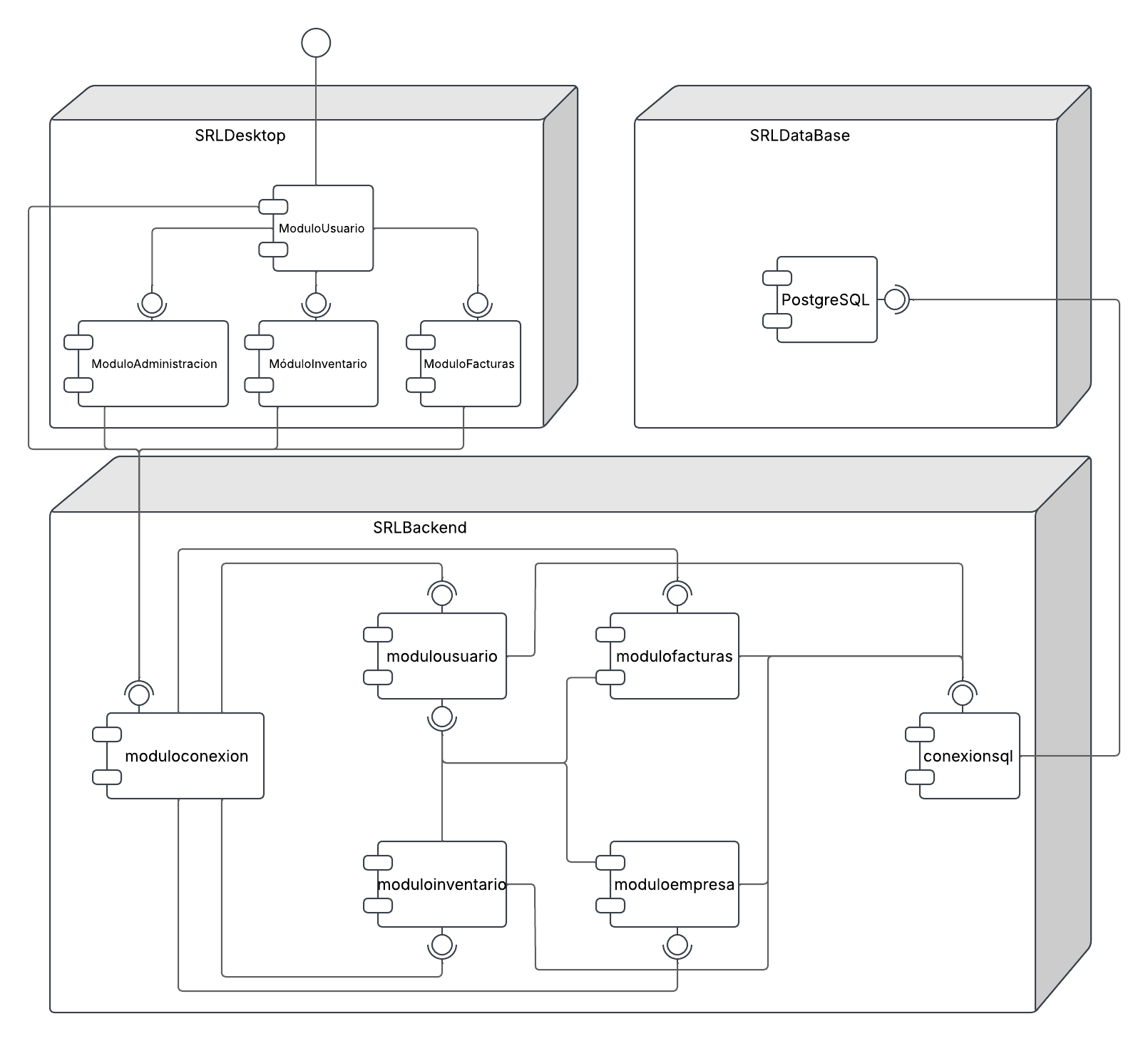


Diagrama 6 - Diagrama de componente del sistema SRL Gestock [1]

Este diagrama muestra los componentes que interactúan en el sistema SRL Gestock. Estos componentes hacen referencia a los paquetes de la aplicación.A continuación, se procede a explicar lo que se muestra en el diagrama.

* **SRL Desktop**
  + **ModuloUsuario:** Este módulo muestra la interfaz gráfica de inicio de sesión, recuperación de contraseñas y creación de usuarios.
  + **ModuloAdministracion:** Este módulo muestra a los administradores del sistema la interfaz gráfica para modificar los datos de la empresa.
  + **ModuloInventario:** Este módulo muestra a los usuarios todos los menús referentes al inventario de la empresa. Permite a los usuarios crear, modificar, ver y eliminar los productos de la empresa.
  + **ModuloFacturas:** Este módulo muestra a los usuarios las herramientas de facturas, tales como crear facturas, ver facturas y buscar facturas.
* **SRLBackend**
  + **moduloconexion:** Permite a SRLDesktop conectarse a SRLBackend por medio de rutas HTTPS. Además, contiene todas las rutas para que se puedan realizar las diferentes acciones de los módulos.
  + **modulousuario:** Su principal función es administrar las sesiones de usuarios. Este módulo permite a los usuarios realizar las acciones por medio de autenticaciones, esto se realiza de esta manera para proporcionar mayor seguridad al sistema.
  + **modulofacturas:** Permite a los usuarios crear facturas para los clientes de SRL Technology, así mismo, realiza la tarea de facturación electrónica automatizada. También, permite a los usuarios ver las facturas.
  + **moduloinventario:** Permite a los usuarios realizar CRUD del inventario de la empresa. Su función es meramente pasar información de la conexión a la base de datos y viceversa.
  + **moduloempresa:** Su principal función es mantener actualizado los datos de la empresa.
  + **conexionsql:** Permite al SRLBackend conectarse por medio de TCP a PostgreSQL. Permite recuperar toda la información de la base de datos.
* **SRLDataBase**
  + **PostgreSQL:** El motor de la base de datos utilizado en el sistema.

Relaciones en el Diagrama:

1. SRLDesktop a SRLBackend:
   1. Descripción: El componente SRLDesktop se comunica con el componente SRLBackend. Esta comunicación permite que la interfaz de usuario (SRLDesktop) interactúe con la lógica de negocio y los servicios proporcionados por el backend (SRLBackend).
   2. Módulos Involucrados:
      1. SRLDesktop: MóduloUsuario, ModuloAdministracion, ModuloInventario, ModuloFacturas.
      2. SRLBackend: modulousuario, moduloinventario, modulofacturas, moduloempresa, moduloconexion.
   3. Naturaleza: Cliente-Servidor. El SRLDesktop actúa como cliente, enviando solicitudes al SRLBackend, que actúa como servidor.
2. SRLBackend a SRLDataBase:
   1. Descripción: El componente SRLBackend se comunica con el componente SRLDataBase. Esta comunicación permite que el backend acceda y manipule los datos almacenados en la base de datos PostgreSQL.
   2. Módulos Involucrados:
      1. SRLBackend: conexionsql.
      2. SRLDataBase: PostgreSQL.
   3. Naturaleza: Acceso a datos. El SRLBackend utiliza conexionsql para interactuar con la base de datos PostgreSQL.
3. Relaciones Internas de SRLDesktop:
   1. Descripción: Dentro del componente SRLDesktop, los módulos MóduloUsuario, ModuloAdministracion, ModuloInventario y ModuloFacturas interactúan entre sí.
   2. Naturaleza: Interdependencia de módulos. Los módulos comparten información y funcionalidades para proporcionar una experiencia de usuario completa.
4. Relaciones Internas de SRLBackend:
   1. Descripción: Dentro del componente SRLBackend, los módulos modulousuario, moduloinventario, modulofacturas, moduloempresa y moduloconexion interactúan entre sí.
   2. Naturaleza: Interdependencia de módulos. Los módulos colaboran para implementar la lógica de negocio y los servicios del sistema.
5. Relación de conexionsql y PostgreSQL:
   1. Descripción: El modulo conexionsql se conecta a la base de datos PostgreSQL.
   2. Naturaleza: Conexión de base de datos. El modulo conexionsql sirve de puente para que el backend pueda realizar las operaciones CRUD en la base de datos.

## Lógica

Antes de empezar, se debe aclarar que el paquete conexionapi con su clase Conexion y el paquete conexionsql con clase **ClientePostgreSQL** se refieren a las mismas clases y paquetes en todos los diagramas de clases mostrados en este documento. Por lo tanto, se procede a explicar por aparte la clase **ClientePostgreSQL** y la clase **Conexion**.

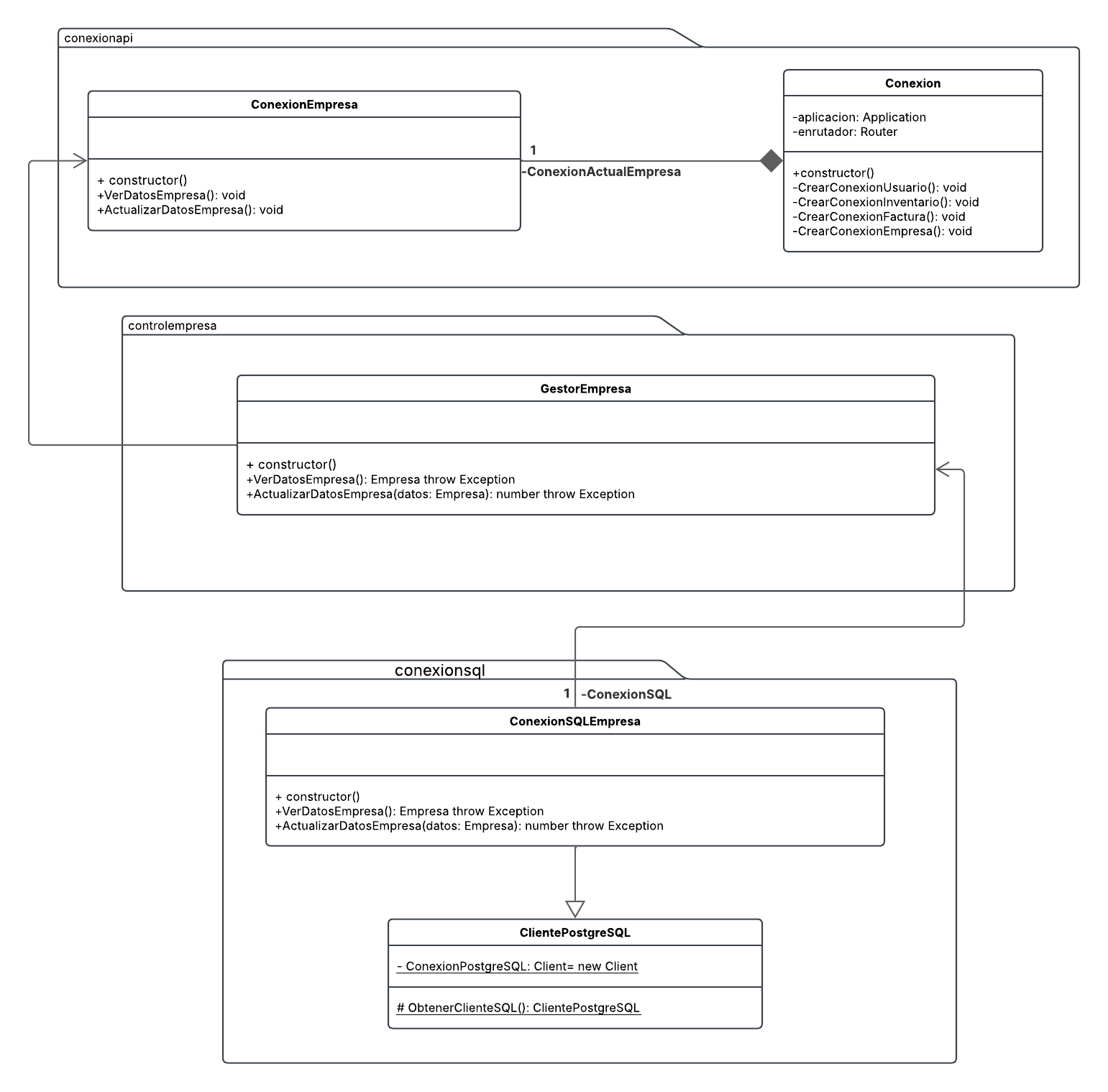


Diagrama 7 - Diagrama de clases para gestionar información de la empresa [1]

Este diagrama representa la lógica que se utiliza para modificar y visualizar la información de la empresa. Ahora mismo se procede a explicar las clases mostrados en

* **ConexionFactura:** Esta es la clase que crea las rutas para realizar las gestiones del inventario por medio de HTTPS.
* **GestorEmpresa:** Este es el gestor principal para modificar y observar la información de la empresa.
* **ConexiónSQLEmpresa:** Esta es la clase que contiene los métodos para crear los script para solicitar la información a la base de datos.

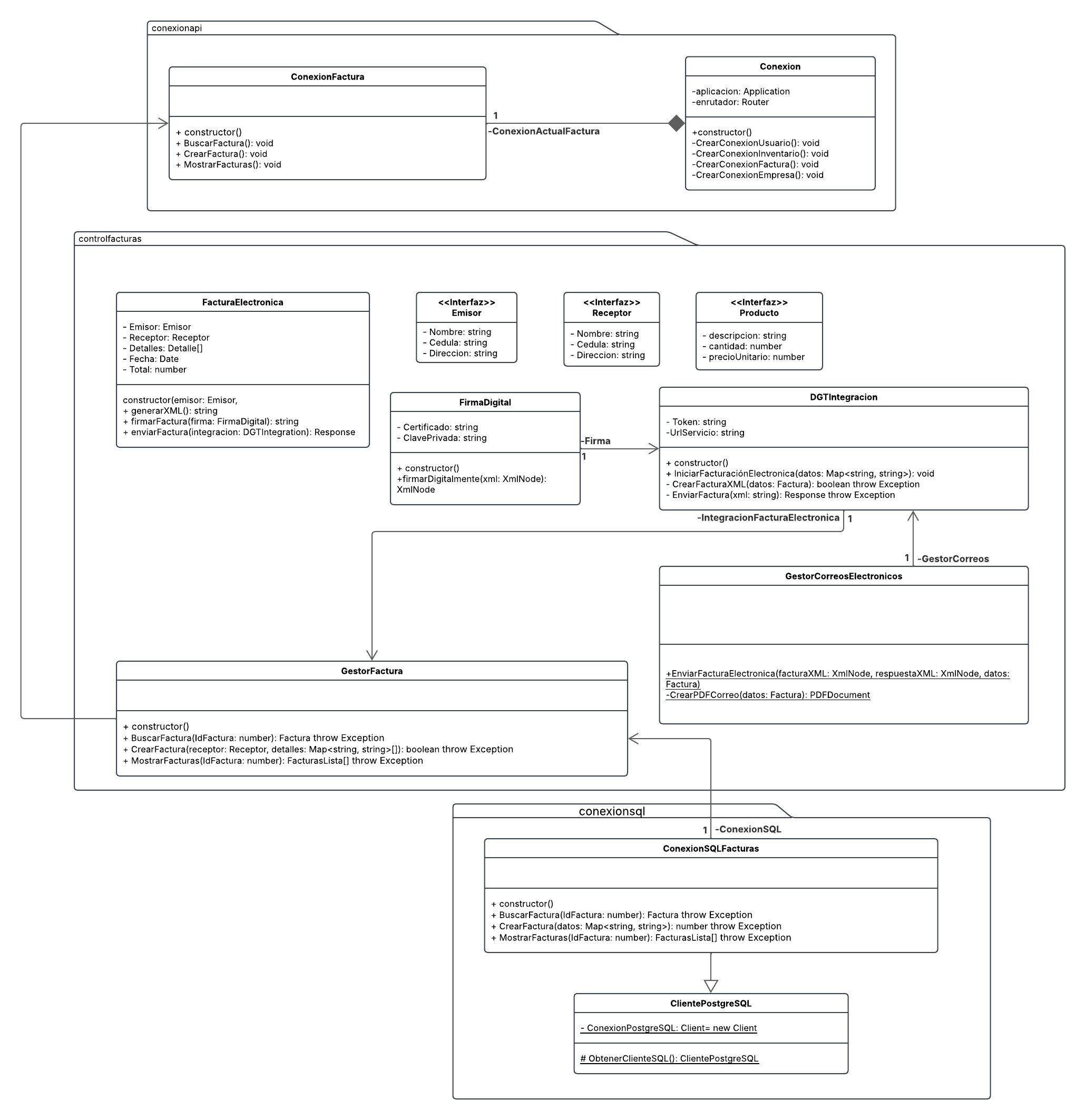


Diagrama 8 - Diagrama de clases de la lógica de factura

Este diagrama de clases muestra la lógica del sistema backend para administrar las facturas electrónicas de la empresa. A continuación, se procede a explicar las clases del diagrama:

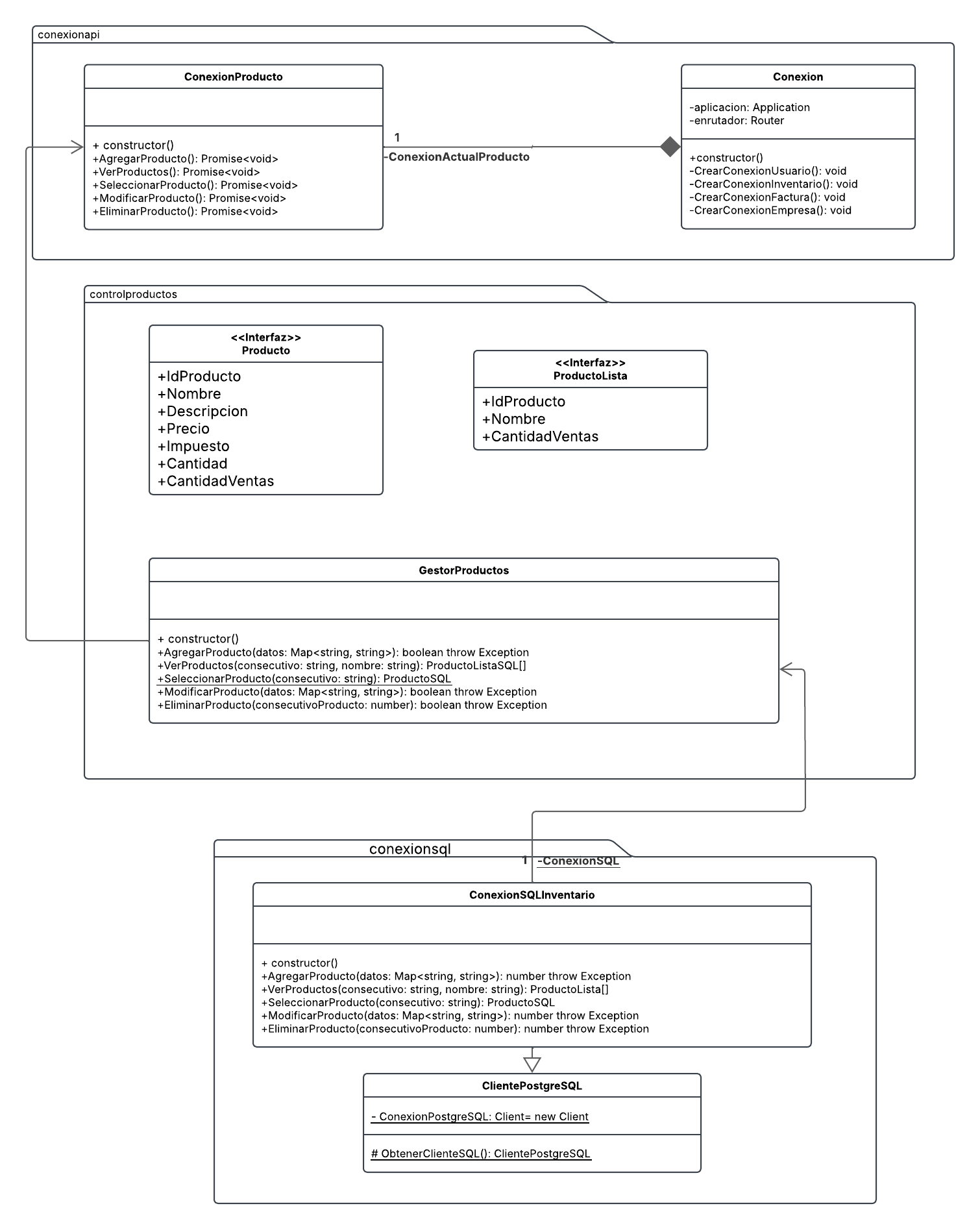


Diagrama 9 - Diagrama de clases de la lógica del producto [1]

* **ConexionFactura:** Se encarga de gestionar la conexión con el sistema de facturación electrónica, permitiendo la creación, consulta y administración de facturas dentro del entorno de integración.
* **ConexionActualFactura:** Representa una instancia activa y específica de la conexión de facturación, útil para mantener estados y operaciones en curso en tiempo real.
* **FacturaElectronica:** Modela una factura electrónica con todos sus componentes esenciales como emisor, receptor, productos, fecha de emisión y el monto total.
* **Emisor:** Define los datos clave de quien emite la factura, incluyendo su nombre, número de identificación y dirección registrada.
* **Receptor:** Especifica los datos básicos del destinatario de la factura, tales como nombre, número de identificación y dirección.
* **Producto:** Representa los elementos facturados, ya sean productos o servicios, con su respectiva descripción, cantidad y precio unitario.
* **FirmaDigital:** Encargada de firmar digitalmente las facturas usando certificados y claves, garantizando su validez, integridad y autenticidad.
* **DOTIntegracion:** Administra la integración del sistema con plataformas externas de facturación, generando y enviando documentos en formatos estructurados como XML.
* **GestorEmpresa:** Controla el flujo interno de facturas en el sistema empresarial, permitiendo su creación, búsqueda y presentación organizada.
* **GestorCorreosElectronicos:** Se ocupa del envío de facturas electrónicas mediante correo electrónico, asegurando una entrega eficaz al cliente.
* **ConexionSQLFacturas:** Gestiona el acceso a la base de datos para registrar, consultar y mostrar facturas, utilizando una conexión SQL directa.

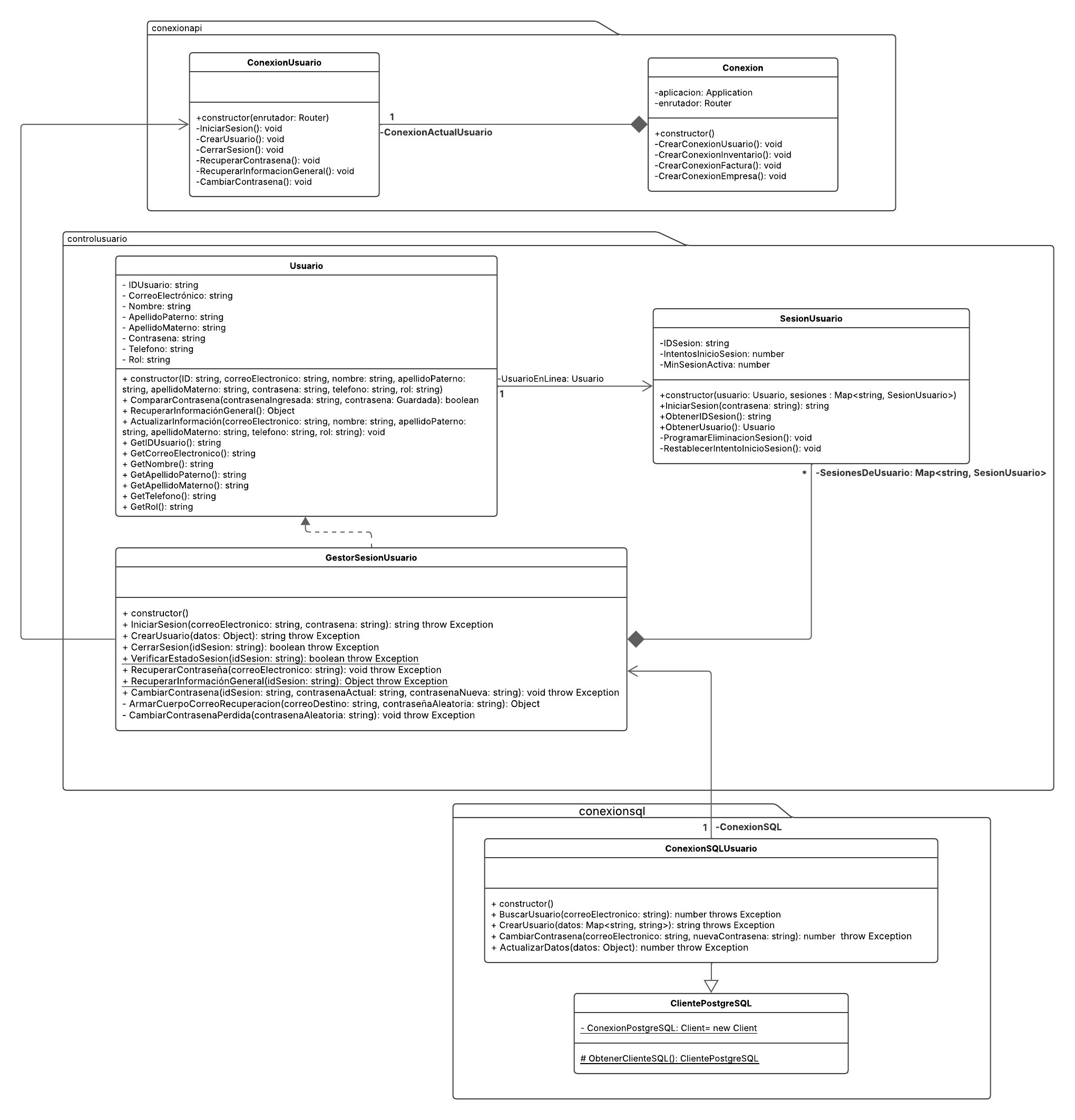


Diagrama 10 - Diagrama de clases de la lógica del usuario [1]

* **ConexionUsuario:** Esta clase maneja las operaciones específicas relacionadas con la autenticación y gestión de usuarios. Incluye métodos para iniciar y cerrar sesiones, crear nuevos usuarios, recuperar contraseñas, obtener información general del usuario y cambiar contraseñas.
* **Usuario:** Esta clase representa la entidad de un usuario en el sistema. Contiene atributos como el ID del usuario, correo electrónico, nombre, apellidos, contraseña, teléfono y rol. También incluye métodos para comparar contraseñas, recuperar información general y actualizar la información del usuario.
* **SesionUsuario:** Esta clase modela la sesión de un usuario autenticado. Contiene atributos como el ID de la sesión, el número de intentos de inicio de sesión fallidos, el tiempo mínimo de sesión activa y el usuario asociado a la sesión. Incluye métodos para iniciar sesión, obtener el ID de la sesión, obtener el usuario asociado y restablecer el contador de intentos de inicio de sesión.
* **GestorSesionUsuario:** Esta clase se encarga de la lógica de negocio relacionada con la gestión de sesiones de usuario. Incluye métodos para iniciar y cerrar sesiones, crear nuevos usuarios, verificar el estado de una sesión, recuperar contraseñas, obtener información general del usuario, cambiar contraseñas y generar correos electrónicos de recuperación de contraseña.
* **ConexionSQLUsuario:** Esta clase actúa como una interfaz para interactuar con la base de datos SQL. Incluye métodos para buscar usuarios por correo electrónico, crear nuevos usuarios en la base de datos, cambiar contraseñas y actualizar la información de los usuarios.

Ahora se debe explica las clases comunes en los diagramas de clases que son **ClientePostgreSQL** y **Conexion**.

* **ClientePostgreSQL:** Esta clase ofrece el mismo cliente de PostgreSQL para todas las conexiones de SQL del backend.
* **Conexion:** Esta es la clase que tiene almacenada las aplicaciones HTTPS necesarias para que el backend pueda escuchar las peticiones del frontend.

## Dependencias

No aplica.

## Información

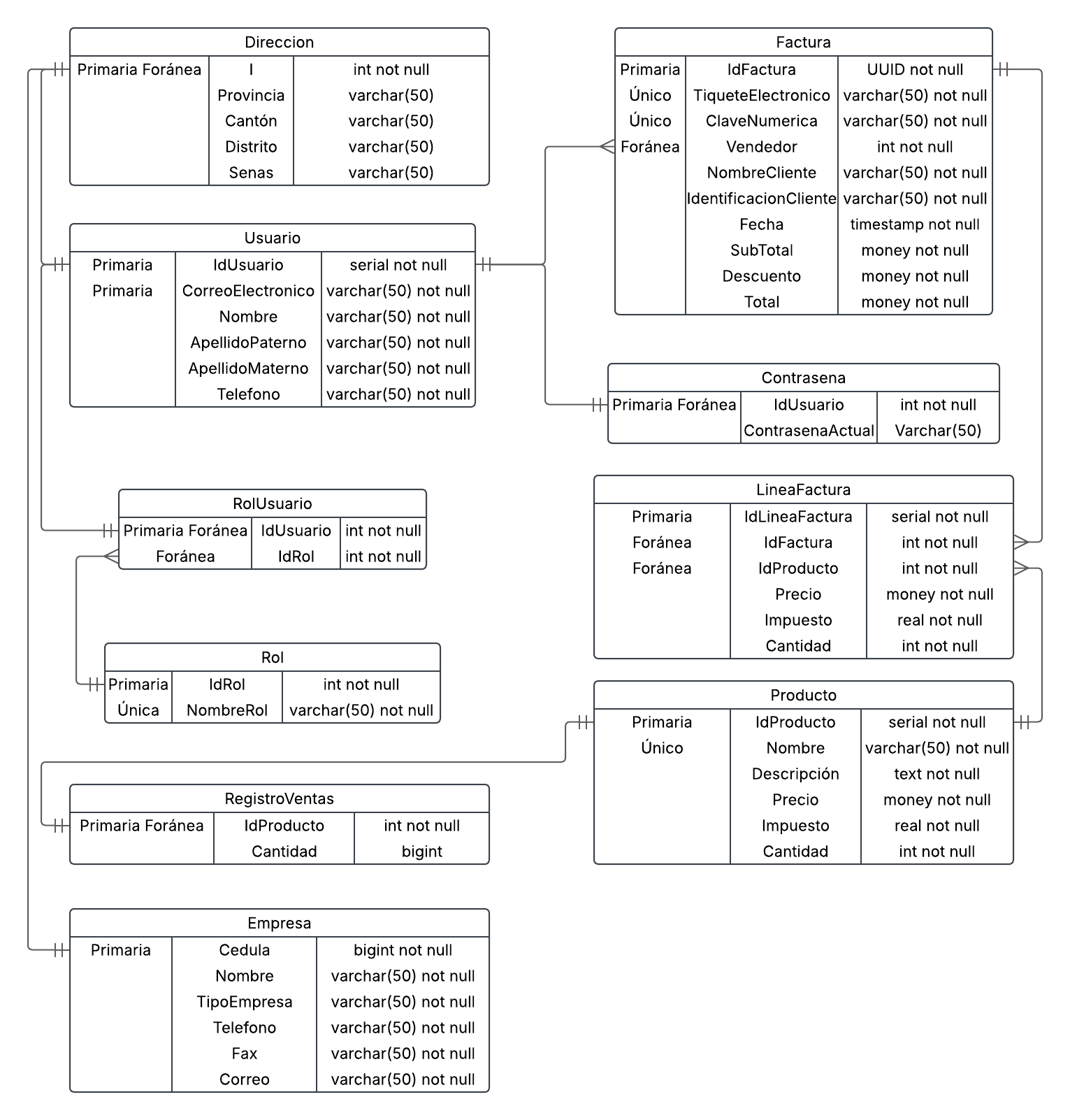


Diagrama 11 - Diagrama entidad relación de la base de datos [1]

Este diagrama muestra las tablas y las relaciones entre las tablas de la base de datos del sistema SRL Gestock. A continuación, se procede a explicar la necesidad de cada una de las tablas y sus relaciones.

* **Empresa:** Cada empresa tiene información asociada de si misma y cada factura tiene información de la empresa. Esta tabla, de una sola fila, se utiliza para poner la información en las facturas creadas
* **Dirección:** Esta es la tabla que administra las direcciones de los usuarios.
* **Usuario:** Esta tabla permite registrar a los usuarios de un sistema de manera que puedan acceder solamente los usuarios que están guardados en esta tabla.
* **RolUsuario:** Esta es una tabla intermedia que permite a los usuarios saber cual es su rol, para evitar el uso de nombres de roles con diferentes patrones.
* **Rol:** Indica los roles que pueden tener los usuarios, de esta manera los usuarios pueden o no acceder a ciertas funciones dependiendo del rol.
* **Contrasena:** Cada usuario requiere una contraseña para ingresar al sistema, por esa razón, es necesario guardar las contraseñas en una tabla.
* **Factura:** El requiere realizar facturas, por lo tanto, es necesario guardar las facturas que se han creado en el sistema.
* **LineaFactura:** Esta tabla se diseñó con el objetivo de guardar los productos comprados en una factura, por esa razón, tiene una clave foránea a producto.
* **Producto:** La parte más importante del sistema, permite realizar el registro de productos, además de llevar el control de la cantidad de unidades que hay por producto.
* **RegistroVentas:** Les permite a los usuarios saber cuantas unidades se han vendido por producto.

## Uso de patrones

No aplica.

## Interfaces

No aplica.

## Interfaz de usuario

Pendiente.

## Estructura

No aplica.

## Interacción

No aplica.

## Dinámica de estados

No aplica.

## Algoritmos

Se utiliza diagramas de flujo, ya que el punto 5.12.4 Examples del estándar utilizado lo permite.

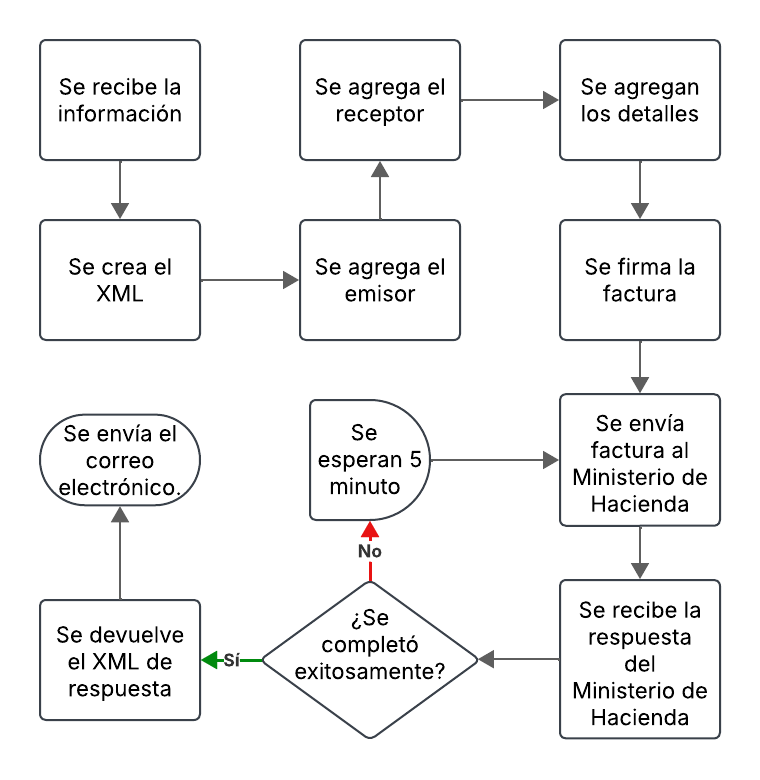


Diagrama 12 - Diagrama de flujo sobre el proceso de facturación electrónica [1]

**Pasos del Proceso:**

1. **Se recibe la información:**
   * El proceso comienza con la recepción de la información necesaria para generar la factura electrónica.
   * Esto puede incluir datos del cliente, productos o servicios, y otros detalles relevantes.
2. **Se crea el XML:**
   * Se genera el archivo XML que contiene la información de la factura electrónica.
   * Este archivo sigue un formato específico requerido por el Ministerio de Hacienda.
3. **Se agrega el emisor:**
   * Se agregan los datos del emisor o empresa que emite la factura.
   * Esto incluye información como el nombre, identificación y dirección de la empresa.
4. **Se agrega el receptor:**
   * Se agregan los datos del receptor o cliente al que se le emitirá la factura.
   * Esto incluye información como el nombre, identificación y dirección del cliente.
5. **Se agregan los detalles:**
   * Se agregan los detalles de los productos o servicios facturados.
   * Esto incluye información como la descripción, cantidad y precio de cada ítem.
6. **Se firma la factura:**
   * Se firma digitalmente el archivo XML de la factura.
   * Esto garantiza la integridad y autenticidad de la factura electrónica.
7. **Se envía factura al Ministerio de Hacienda:**
   * Se envía el archivo XML de la factura al Ministerio de Hacienda para su validación.
   * Este envío se realiza a través de un servicio web o plataforma designada por el Ministerio.
8. **¿Se completó exitosamente?**
   * Se verifica si la factura fue procesada y validada con éxito por el Ministerio de Hacienda.
   * Si la respuesta del Ministerio indica que la factura fue aceptada, el proceso continúa.
   * Si la respuesta indica que hubo un error o rechazo, se esperan cinco minutos y se vuelve a intentar.
9. **Se esperan 5 minutos:**
   * Se espera un tiempo de 5 minutos para recibir la respuesta del Ministerio de Hacienda.
   * Este tiempo de espera es necesario para que el Ministerio se recupere de algún error o se restablezca la conexión.
10. **Se recibe la respuesta del Ministerio de Hacienda:**
    * Se recibe la respuesta del Ministerio de Hacienda, que indica el estado de la factura.
    * Esta respuesta puede incluir un mensaje de aceptación o un mensaje de error con detalles sobre el rechazo.
11. **Se devuelve el XML de respuesta:**
    * Se devuelve la respuesta del Ministerio de Hacienda con la información aceptada.
12. **Se envía el correo electrónico:**
    * Se envía un correo electrónico al cliente con la factura electrónica y el XML de respuesta (si aplica).

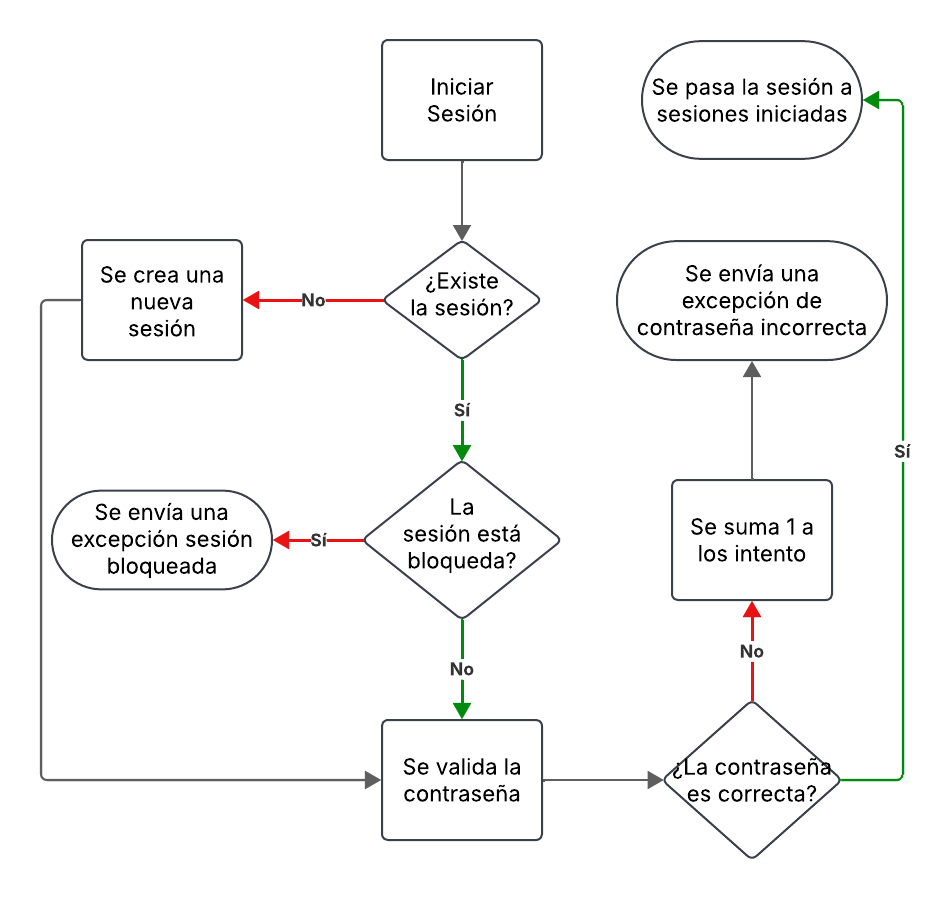


Diagrama 13 - Diagrama de flujo del proceso de inicio de sesión [1]

**Pasos del Proceso:**

1. **Iniciar Sesión:**
   * El proceso comienza cuando un usuario intenta iniciar sesión en el sistema.
2. **¿Existe la sesión?**
   * Se verifica si ya existe una sesión activa para el usuario.
   * **Si no existe una sesión**, se crea una nueva sesión.
3. **¿La sesión está bloqueada?**
   * Se verifica si la sesión está bloqueada debido a múltiples intentos fallidos de inicio de sesión.
   * **Si la sesión está bloqueada**, se envía una excepción de sesión bloqueada y se finaliza el proceso.
   * **Si la sesión no está bloqueada**, se valida la contraseña.
4. **Se valida la contraseña:**
   * Se verifica si la contraseña ingresada por el usuario es correcta.
5. **¿La contraseña es correcta?**
   * **Si la contraseña es correcta**, se pasa la sesión a la lista de sesiones iniciadas y se finaliza el proceso.
   * **Si la contraseña es incorrecta**, se suma 1 al contador de intentos fallidos y se envía una excepción de contraseña incorrecta.
6. **Se suma 1 a los intentos:**
   * Se incrementa el contador de intentos fallidos de inicio de sesión.
7. **Se envía una excepción de contraseña incorrecta:**
   * Se envía una excepción de contraseña incorrecta al usuario.
8. **Se pasa la sesión a sesiones iniciadas:**
   * Si la contraseña es correcta o si ya existía una sesión, se pasa la sesión a la lista de sesiones iniciadas.
   * El proceso finaliza con el usuario autenticado y la sesión activa.

Los dos diagramas de flujo mostrados, muestran los algoritmos más complejos del sistema SRLGestock, ya que puede llegar a ser confuso como es que se realizan esas tareas.

## Recursos

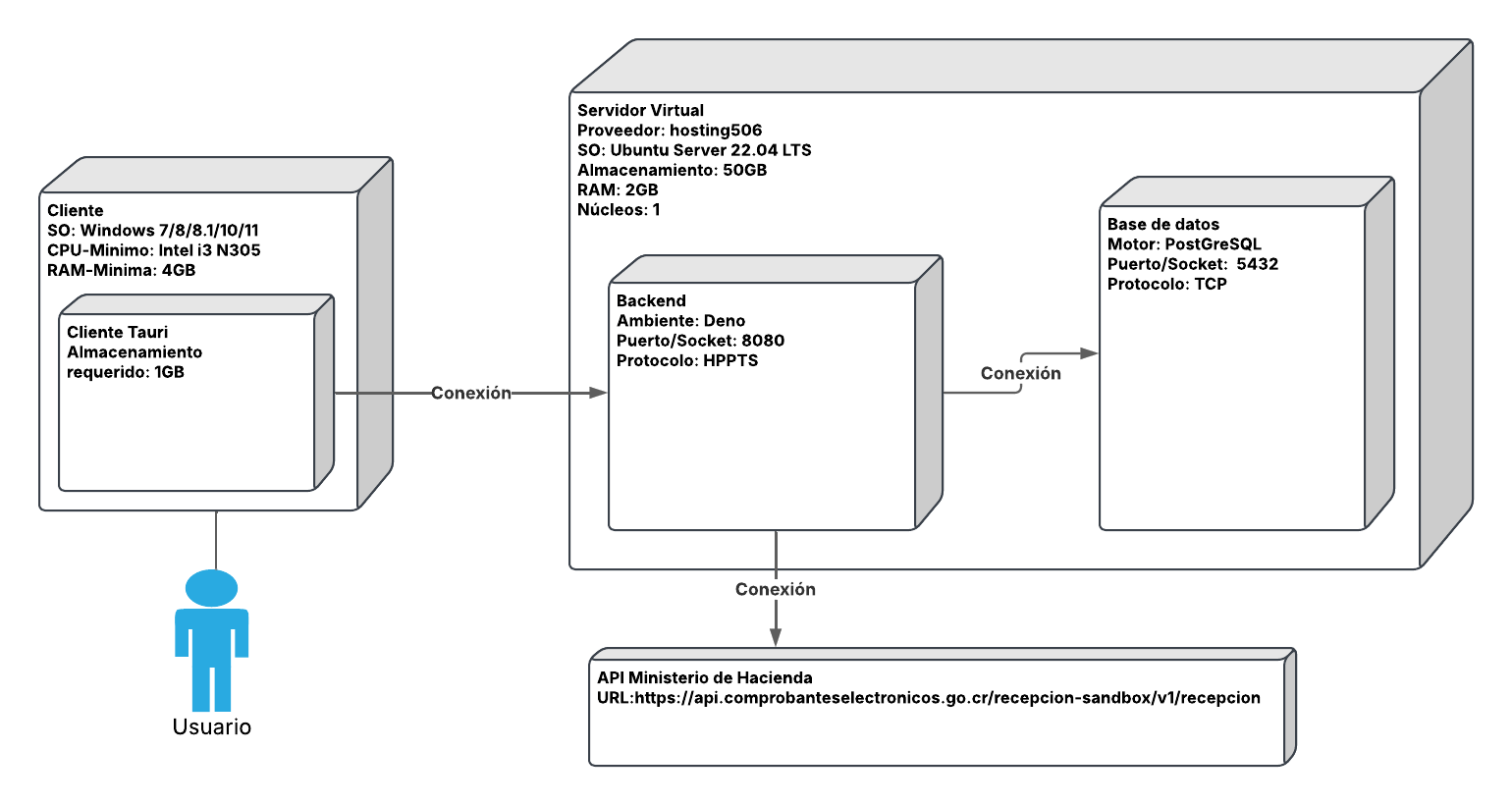


Diagrama 14 - Diagrama de implementación del sistema SRLGestock [1]

Este diagrama representa la arquitectura de un sistema diseñado para interactuar con la API del Ministerio de Hacienda de Costa Rica para la gestión de comprobantes electrónicos. La arquitectura se basa en un modelo cliente-servidor distribuido, donde un **usuario** interactúa a través de una aplicación **Cliente Tauri**. Esta aplicación, diseñada para ser multiplataforma, requiere un mínimo de 1GB de almacenamiento y está optimizada para sistemas operativos Windows desde la versión 7 hasta la 11, con requisitos mínimos de hardware de un procesador Intel i3 N305 y 4GB de RAM.

La aplicación **Cliente Tauri** se comunica con un **Backend** alojado en un **Servidor Virtual** proporcionado por hosting506. Este servidor, configurado con Ubuntu Server 22.04 LTS, cuenta con 50GB de almacenamiento, 2GB de RAM y un núcleo de procesamiento. El Backend, desarrollado en el entorno Deno, opera en el puerto 8080 utilizando el protocolo seguro HTTPS para la comunicación.

El **Backend** es responsable de la lógica de negocio y la orquestación de la comunicación con la **Base de Datos** PostgreSQL, que reside en el mismo **Servidor Virtual**. La base de datos, accesible a través del puerto 5432 y el protocolo TCP, almacena de forma persistente los datos del sistema.

Finalmente, el **Backend** se comunica con la **API del Ministerio de Hacienda** a través de la URL https://api.comprobanteselectronicos.go.cr/recepcion-sandbox/v1/recepcion. Esta comunicación permite la transmisión y recepción de comprobantes electrónicos, integrando el sistema con los servicios del Ministerio.

La arquitectura propuesta busca un equilibrio entre rendimiento, seguridad y escalabilidad. Se ha considerado el uso de HTTPS para la comunicación entre el cliente y el backend, así como la selección de tecnologías que permitan un manejo eficiente de la información y una fácil escalabilidad horizontal en caso de aumento de la demanda.

# Apéndice - Alternativas de diseño

Para crear los diseños alternativos se decidió utilizar la IA de Lucidchart para generar diagramas de clases. Esta se le ingresa una descripción del diagrama de clases y ella la genera por sí misma. Por esa razón, estos dos diagramas que se van a presentar tienen un diseño diferente a los diagramas anteriores, aunque de igual manera que los diagramas anteriores, se puede observar su diseño por medio de la referencia.

Estoy conforme con la mayor parte del sistema que se ha diseñado en este documento, pero hay una función en particular que aún me tiene dudoso sobre su implementación. La función de facturación electrónica es un poco compleja, debido a que se debe conectar el sistema SRLGestock a la API del Ministerio de Hacienda, el problema es que hay poca documentación y los diseños que se pueden crear pueden llegar a ser soluciones imprecisas para cumplir su función debidamente.

Por esa razón, se decidió crear diagramas de clases más para tener opciones de cambiar en caso de que se necesite una alternativa.

1. **Alternativa uno**

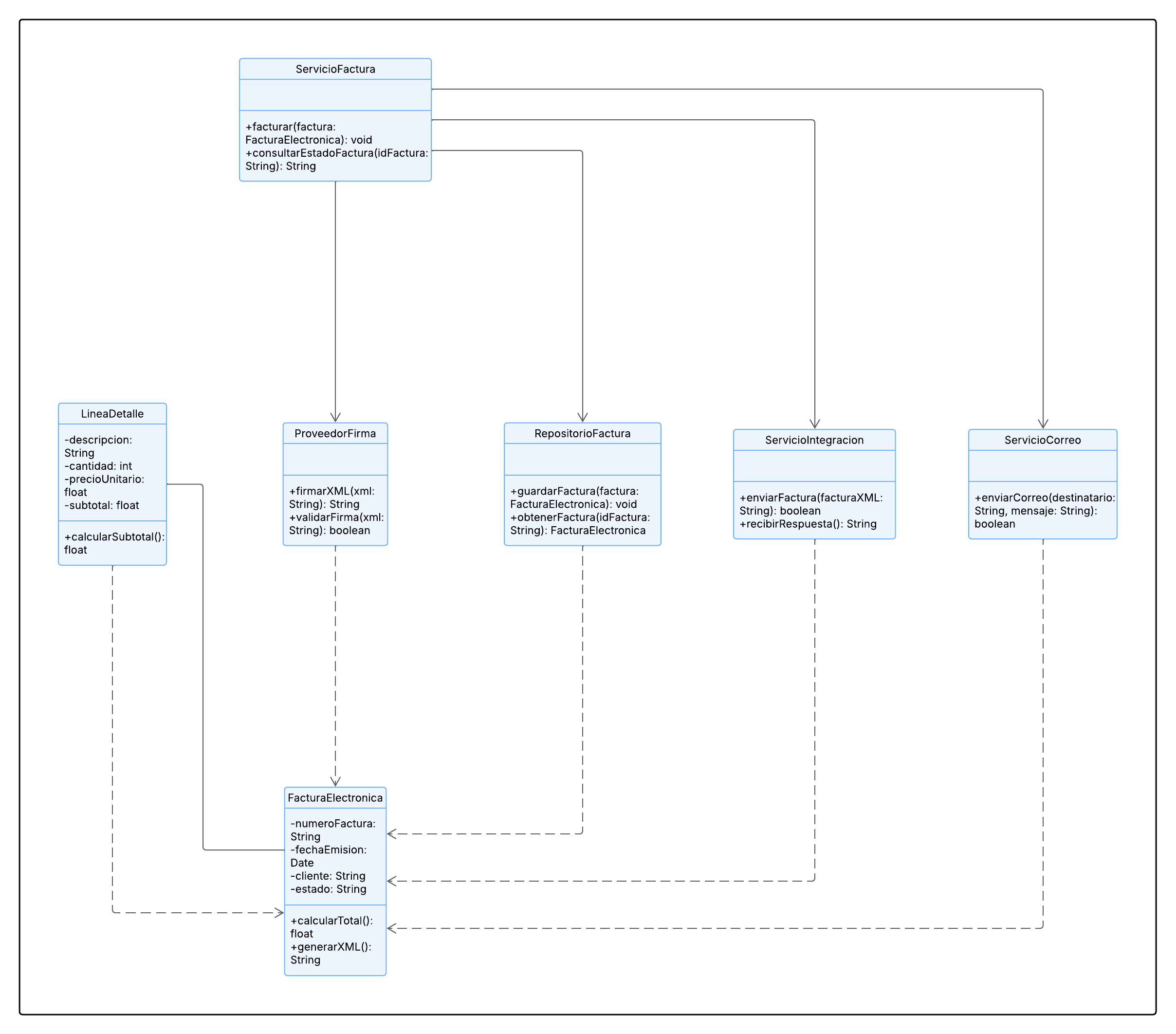


Diagrama 15 - Diagrama de clases de la primera propuesta alternativa de la facturación [1]

El sistema está pensado para que cada parte haga solo lo que le toca, sin mezclarse con cosas que no le corresponden. Por eso, la clase principal (ServicioFactura) se encarga de manejar todo el proceso de facturación, desde crear hasta enviar la factura. Así se tiene todo el flujo controlado en un solo lugar.

Las partes que hacen cosas más específicas, como firmar digitalmente, mandar la factura al gobierno o enviarla por correo, se dejaron en clases aparte. Esto se hizo para que, si más adelante hay que cambiar algo (por ejemplo, el método de firma o el sistema de correos), no haya que tocar el resto del sistema.

También se separó todo lo que tiene que ver con guardar y leer facturas de la base de datos. Así, si algún día se quiere usar otro tipo de base de datos, se puede cambiar sin complicaciones.

En resumen, este diseño hace que el sistema sea más fácil de entender, de mantener y de cambiar en el futuro sin romper todo.

1. **Alternativa dos**

**Diagrama

El contenido generado por IA puede ser incorrecto.**

Diagrama 16 - Diagrama de clases de la segunda propuesta alternativa de la facturación [1]

Este diseño se centra en mantener el sistema bien organizado y fácil de modificar si se necesita en el futuro. La clase Documento es abstracta porque representa la idea general de cualquier documento digital (no solo facturas), lo que deja abierta la posibilidad de agregar más tipos de documentos más adelante. FacturaDigital hereda de esta clase y añade los campos específicos de una factura, como fecha, total y XML generado.

Cada factura está compuesta por varios Detalle, que contienen los productos o servicios. Separar esta parte ayuda a manejar mejor los cálculos y mantener limpia la lógica de la factura.

La clase Cliente representa tanto al emisor como al receptor, y está conectada directamente con la factura para simplificar el acceso a su información.

Toda la lógica del proceso está centralizada en GestorFactura, que actúa como el coordinador: crea la factura, aplica la firma, la envía a Hacienda, la guarda y notifica al cliente. Este enfoque permite tener un punto de control claro y facilita pruebas o cambios sin afectar todo el sistema.

Los servicios externos (firma, integración con Hacienda, almacenamiento y notificaciones) están separados en clases propias. Esto no solo permite reutilizarlos si se agregan otros tipos de documentos, sino que también hace que el código sea más flexible y fácil de mantener si cambia alguno de esos servicios.

En resumen, se trata de un diseño que busca claridad, flexibilidad y facilidad de mantenimiento, sin complicar más de lo necesario.