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

**Estudiantes:**

Andrew López Herrera - 2021062132

**Asignación:**

Documento de requerimientos del software

**Fecha de entrega:**

Viernes 14 de marzo, 2025

Contenido

[1. Introducción 3](#_Toc192885558)

[1.1. Propósito 3](#_Toc192885559)

[1.2. Alcance 3](#_Toc192885560)

[1.3. Resumen del producto 4](#_Toc192885561)

[1.3.1. Perspectiva del producto 4](#_Toc192885562)

[1.3.2. Funciones del Producto 5](#_Toc192885563)

[1.3.3. Características del usuario 7](#_Toc192885564)

[1.3.4. Limitaciones 7](#_Toc192885565)

[1.4. Definiciones 8](#_Toc192885566)

[1.5. Siglas y abreviaturas 8](#_Toc192885567)

[2. Referencias 8](#_Toc192885568)

[3. Requisitos 9](#_Toc192885569)

[3.1. Funciones 9](#_Toc192885570)

[3.2. Requisitos de usabilidad 12](#_Toc192885571)

[3.3. Interfaces externas 13](#_Toc192885572)

[3.4. Requisitos de rendimiento 16](#_Toc192885573)

[3.5. Requisitos de la base de datos lógica 18](#_Toc192885574)

[3.6. Restricciones de diseño 21](#_Toc192885575)

[3.7. Atributos del sistema de software 22](#_Toc192885576)

[3.8. Información de soporte 23](#_Toc192885577)

[4. Verificación 26](#_Toc192885578)

[5. Apéndices 29](#_Toc192885579)

[5.1. Supuestos y dependencias 29](#_Toc192885580)

[5.2. Matriz de trazabilidad 31](#_Toc192885581)

# 1. Introducción

## 1.1. Propósito

El propósito de crear esta aplicación es para automatizar los procesos de guardado de inventario y facturación. Actualmente, el cliente, SRL Technologies, 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 Technologies, 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.

## 1.2. Alcance

Esta aplicación consta de cuatro objetivos para su funcionamiento:

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 Technologies. 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 Technologies 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.

## 1.3. Resumen del producto

### 1.3.1. Perspectiva del producto

El producto es totalmente nuevo y creado desde cero.

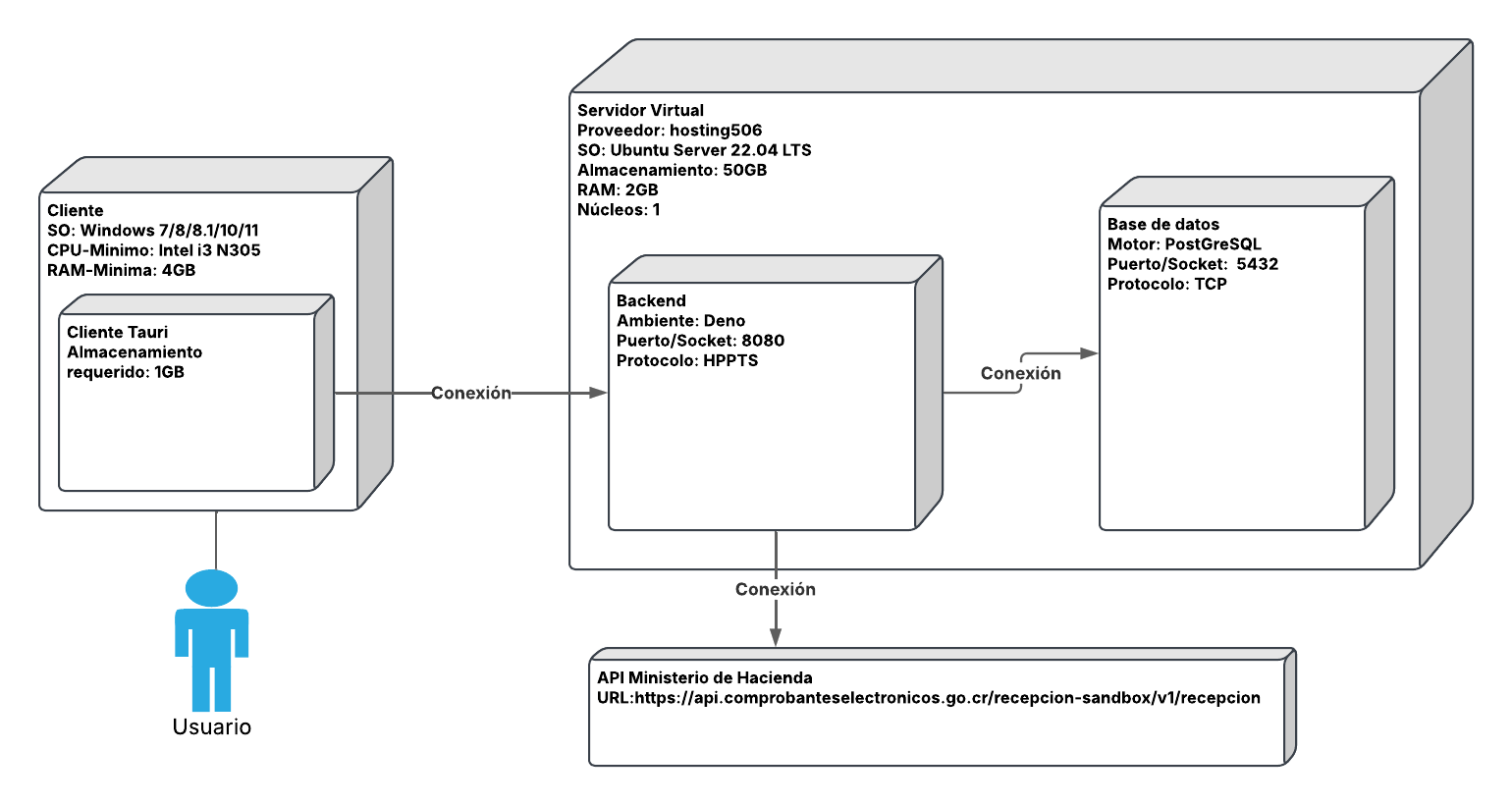


Diagrama 1 - Implementación del sistema

### 1.3.2. Funciones del Producto

* Gestión de negocio
* Los empleados del negocio contarán con las herramientas necesarias para llevar a cabo las tareas diarias del establecimiento a través de una herramienta centralizada
* Gestión de empleados
* Se podrá administrar e identificar los datos y accesos de manipulación de la información con base en las responsabilidades de los empleados del negocio
* Administración de inventario
* El cliente será capaz de los productos en stock con los que cuenta el negocio, así como los detalles específicos de los mismos.
* Adicionalmente, el administrador del negocio podrá manipular esta información, ya sea agregando, eliminando o corrigiendo los datos de dichos artículos.
* Se le permitirá al administrador de este negocio una visualización estratégica en pro de la optimización de sus productos en razón a las ventas y consumo de estos.
* Facturación
* Cada una de las ventas serán procesadas automáticamente, tomando en cuenta la gestión del inventario en conjunto con esta función.
* La facturación electrónica es parte de este proceso, tanto como para el negocio como para los clientes del mismo.

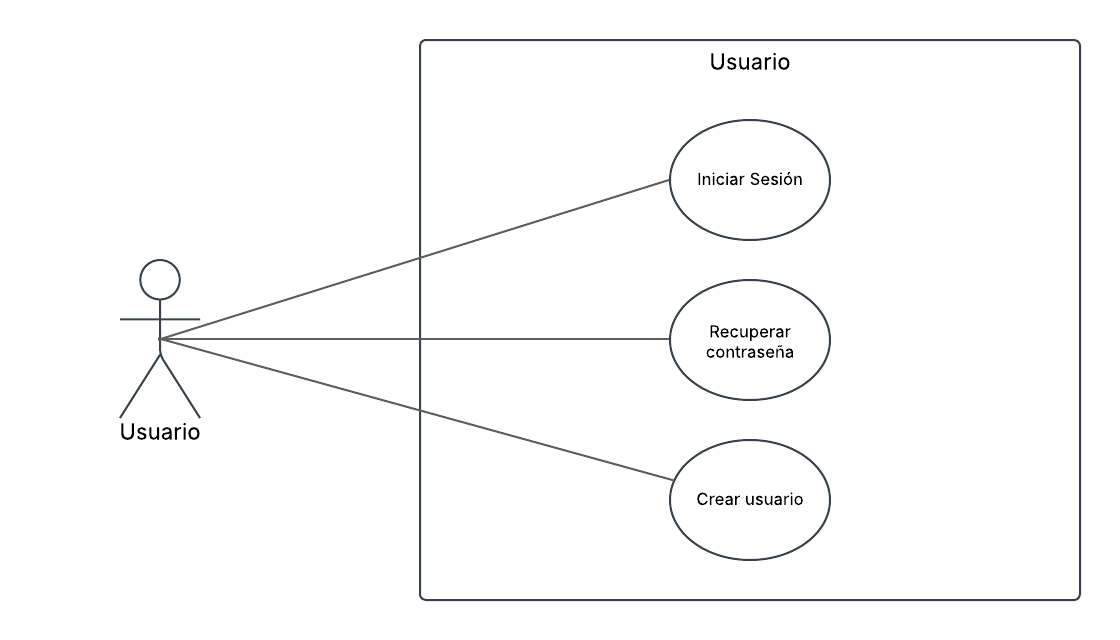


Diagrama 2 - Casos de uso de los usuarios

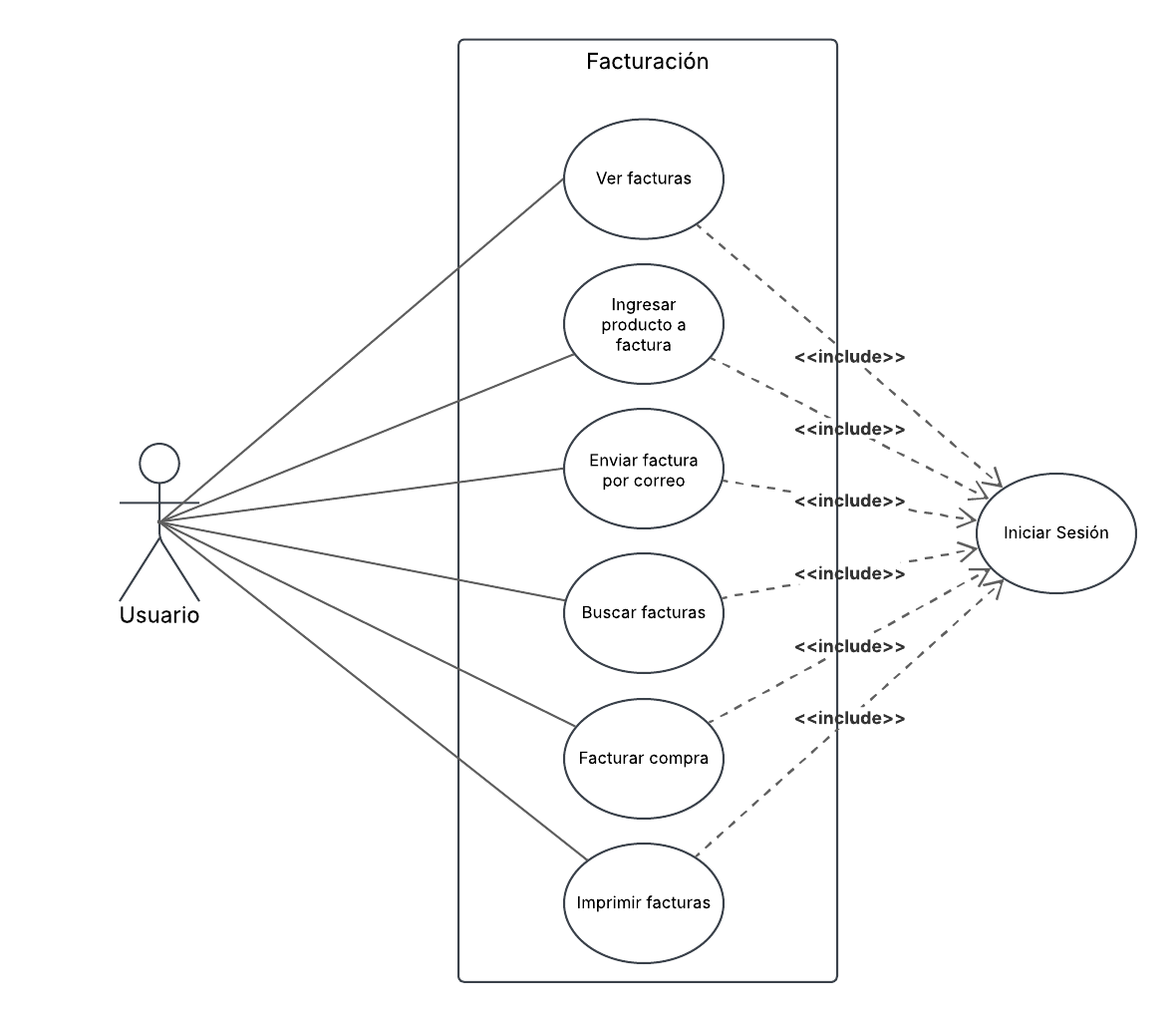


Diagrama 3- Casos de uso de la facturación

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama 4 - Caso de uso de empresa

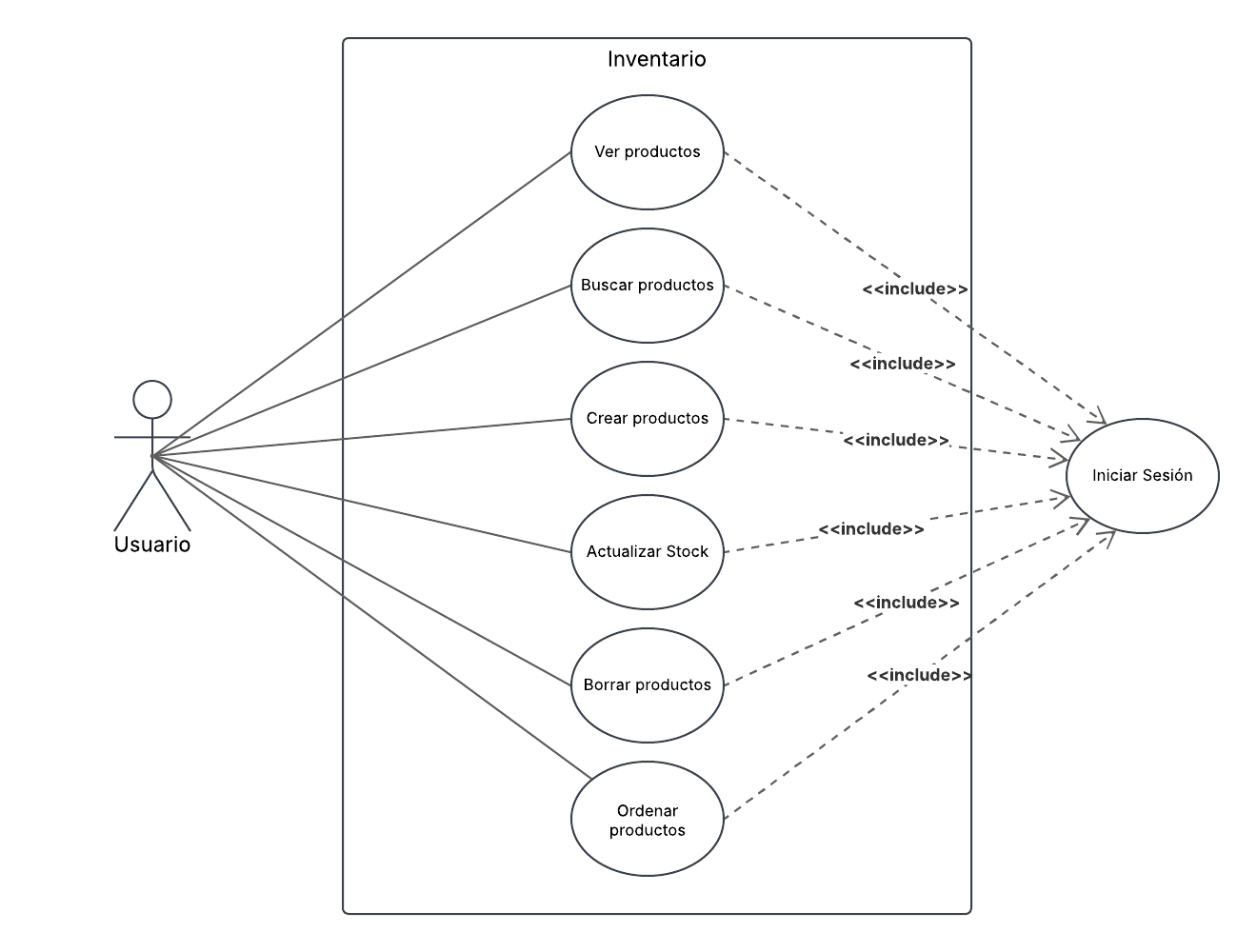


Diagrama 5 - Casos de uso del inventario

### 1.3.3. Características del usuario

Los usuarios son empleados o el mismo dueño del negocio. Algunas personas tienen problemas de visión, no se sabe exactamente cual, pero se sabe ya que utilizan lentes diariamente. Tienen experiencia utilizando sistema de facturación con diseños amigables y sencillos. Estas personas saben leer, es muy importante para realizar las facturas. Tienen conocimientos generales en dispositivos tecnológicos. Son usuarios con nulo conocimiento de computación avanzada. Son personas que deben realizar tareas de manera rápida para evitar el acumulamiento de clientes en el establecimiento. Puede haber personas con nula experiencia en facturación.

### 1.3.4. Limitaciones

El dinero es una de las limitaciones más importantes, ya que SRL Technologies menciona que puede haber meses en los que hayan bastantes ventas, mientras que otros meses se realicen pocas ventas. Relacionado al dinero, otra limitación es la potencia del servidor, ya que se utiliza un servidor con recursos limitados para ahorrar costos de por equipo. Por el momento, se intentará utilizar una máquina virtual con 2GB de RAM que es el recurso que notamos más comprometido.

La API del Ministerio de Hacienda representa una limitación, ya que no se encuentra mucha documentación en internet y no se sabe muy bien cómo se usa y como se debe implementar este en el proyecto. Además, no se sabe que errores pueda tener este software.

La falta de equipo de prueba es otra limitación bastante importante, ya que, antes de hacer que el cliente pague mensualmente un servidor que no utilizará hasta que el sistema esté listo. Por lo que, se requiere equipo con especificaciones similares para realizar las pruebas y asegurarnos que el sistema no tenga ningún problema en la ejecución del backend.

## 1.4. Definiciones

**Cliente:** se refiere al software que interactúa directamente con el cliente y se encarga de realizar las peticiones al backend. Esta es la aplicación instalable en las computadoras de la tienda y muestra la interfaz de usuario para realizar operaciones.

**Servidor:** se refiere al software central que recibe todas las peticiones del cliente. Este software se encarga de ser un puente de entre los clientes y la base de datos. No interactúa directamente con el usuario.

## 1.5. Siglas y abreviaturas

SRL: Sheila, Rodney, Luis

# 2. Referencias

No aplica

# 3. Requisitos

### 3.1. Funciones

**RS001: El sistema debe iniciar sesión.**

**Descripción:** El sistema debe permitir a los usuarios ingresar con un nombre de usuario y contraseña válidos.

**Importancia:** Alta

**Estimado inicial:** 8 horas

**RS002: El sistema debe crear usuarios.**

**Descripción:** El sistema debe permitir la creación de nuevos usuarios con roles y permisos asignados.

**Importancia:** Alta

**Estimado inicial:** 10 horas

**RS003:** **El sistema debe permitir la recuperación de contraseñas.**

**Descripción:** El sistema debe permitir que los usuarios recuperen sus contraseñas mediante un correo electrónico de recuperación.

**Importancia:** Alta

**Estimado inicial:** 6 horas

**RS004:** **El sistema debe crear productos al inventario.**

**Descripción:** El sistema debe permitir la creación de productos con detalles como nombre, cantidad, precio y descripción.

**Importancia:** Alta

**Estimado inicial:** 8 horas

**RS005: El sistema debe actualizar el stock del inventario.**

**Descripción:** El sistema debe permitir la actualización del stock de productos de acuerdo a las ventas o nuevas adquisiciones.

**Importancia:** Alta

**Estimado inicial:** 7 horas

**RS006: El sistema debe mostrar los productos del inventario.**

**Descripción:** El sistema debe permitir ver una lista de productos con detalles como nombre, cantidad, precio y descripción.

**Importancia:** Alta

**Estimado inicial:** 6 horas

**RS007: El sistema debe borrar los productos del inventario.**

**Descripción:** El sistema debe permitir la eliminación de productos del inventario.

**Importancia:** Media

**Estimado inicial:** 6 horas

**RS008: El sistema debe buscar productos por consecutivo o nombre.**

**Descripción:** El sistema debe permitir buscar productos de manera eficiente utilizando su nombre o consecutivo.

**Importancia:** Alta

**Estimado inicial:** 7 horas

**RS009: El sistema debe ordenar productos por cantidad de ventas.**

**Descripción:** El sistema debe permitir ordenar los productos en el inventario según la cantidad de ventas.

**Importancia:** Baja

**Estimado inicial:** 6 horas

**RS010: El sistema debe imprimir facturas.**

**Descripción:** El sistema debe permitir generar e imprimir facturas de ventas.

**Importancia:** Alta

**Estimado inicial:** 10 horas

**RS011: El sistema debe crear el archivo XML con el formato exigido por hacienda para la facturación electrónica.**

**Descripción:** El sistema debe generar archivos XML con el formato requerido por el Ministerio de Hacienda para la facturación electrónica.

**Importancia:** Alta

**Estimado inicial:** 24 horas

**RS012: El sistema debe enviar el archivo XML a hacienda para registrar la factura electrónica.**

**Descripción:** El sistema debe enviar el archivo XML de la factura creada al API de hacienda por medio del endpoint.

**Importancia:** Alta

**Estimado inicial:** 72 horas

**RS013: El sistema puede enviar opcionalmente el PDF de factura, XML de factura y el XML de acuse de recibido a un cliente por medio de correo electrónico.**

**Descripción:** El sistema debe permitir el envío de los archivos generados (PDF, XML y acuse de recibido) por correo electrónico al cliente.

**Importancia:** Media

**Estimado inicial:** 8 horas

**RS014: El sistema debe facturar una compra.**

**Descripción:** El sistema debe permitir la creación de facturas a partir de compras realizadas.

**Importancia:** Alta

**Estimado inicial:** 10 horas

**RS015: El sistema debe mostrar las facturas creadas.**

**Descripción:** El sistema debe permitir ver un listado de todas las facturas creadas.

**Importancia:** Media

**Estimado inicial:** 7 horas

**RS016: El sistema debe buscar facturas por número de factura.**

**Descripción:** El sistema debe permitir la búsqueda de facturas por su número de factura.

**Importancia:** Media

**Estimado inicial:** 6 horas

**RS017: El sistema debe permitir el ingreso de productos en una factura.**

**Descripción:** El sistema debe permitir agregar productos a la factura durante su creación.

**Importancia:** Alta

**Estimado inicial:** 8 horas

**RS018: El sistema debe buscar productos mientras se crea la factura.**

**Descripción:** El sistema debe permitir buscar productos en el inventario mientras se está creando la factura.

**Importancia:** Alta

**Estimado inicial:** 7 horas

**RS019: El sistema debe actualizar los datos de la empresa.**

**Descripción:** El sistema debe permitir actualizar los datos de la empresa, como dirección, teléfono y correo electrónico.

**Importancia:** Media

**Estimado inicial:** 6 horas

## 3.2. Requisitos de usabilidad

**Efectividad**

* **Tasa de éxito en tareas clave:** El usuario debe completar el 100% de las tareas sin error en todas las tareas.
* **Reducción de errores:** Se debe crear una interfaz gráfica responsiva. Se deben poner botones que al ser presionado muestren un mensaje de lo que se espera recibir. Se debe utilizar el placeholder para indicar el formato esperado de datos.
* **Tiempo de recuperación:** Se deben poner mensajes de errores en tiempo real cuando los usuarios estén llenando los campos de entrada de texto. No se debe eliminar los datos de un formulario cuando el usuario ingrese datos incorrectos, solamente se debe indicar el problema.

**Eficiencia**

* **Tiempo de ejecución de tareas:** El usuario no debe durar más de 10 segundos iniciando sesión. El usuario no debe durar más de 3 minutos creando un usuario. El usuario no debe tardar más de 7 segundos en buscar y encontrar un producto. El usuario no debe durar más de 2 minutos y medio creando o actualizando un producto. El usuario no debe durar más de 2 minutos en llenar la información del cliente para la factura electrónica. El usuario no debe durar más de 2 minutos actualizando la información de la empresa.
* **Carga cognitiva:** No deben de pasar más de 5 pantallas para realizar una acción. Todas las pantallas deben de tener un diseño similar (objetos en los mismos lugares de ser posible).
* **Rendimiento del sistema:** Estos datos se presentan el apartado 3.4.

**Satisfacción**

* **Facilidad de aprendizaje:** Se propone mostrar pantallas de ayuda que indican que hace cada herramienta de la pantalla.
* **Diseño accesible:** El diseño debe cumplir con las normas A de la WCAG 2.2.
* **Experiencia del usuario (UX):** Diseño limpio, atractivo y que genere confianza. Se opta por un diseño minimalista para evitar la carga de muchos componentes.
* **Encuestas de satisfacción:** Se deben realizar pruebas con los usuarios finales quienes indicarán lo siguiente: grado de facilidad de aprendizaje, grado de agrado de la interfaz gráfica, grado de facilidad de tareas y aspectos a mejorar.

**Prevención de daños o riesgos**

* **Seguridad de datos:** Las personas solo pueden realizar 5 intentos de ingresar a las cuentas cada cinco minutos. Se solicitará contraseñas robustas a los usuarios.
* **Prevención de errores críticos:** Al crear facturas, se dará una visualización general de la factura para que el usuario se asegure que todo es correcto. Para eliminar un producto del inventario deberá escribir el nombre del producto con el que fue registrado en el inventario.

## 3.3. Interfaces externas

**Interfaz Cliente Tauri**

**Nombre:** Cliente Tauri

**Descripción del propósito:** La interfaz gráfica de usuario permite a los usuarios gestionar y administrar el inventario de productos, facturar ventas, recuperar contraseñas y crear usuarios. Es la cara visible del sistema para la administración de inventarios y facturación electrónica.

**Fuente de la entrada o destino de la salida:** La entrada de datos se realiza mediante campos de texto en la interfaz gráfica de usuario. La salida de los datos se realiza mediante solicitudes HTTPS que interactúan con el backend del sistema.

**Rango válido, precisión y/o tolerancia:** No aplica.

**Unidades de medida:** Cantidad de unidades o paquetes.

**Temporización / Tiempo de respuesta esperado:**

**Acción:** Tiempo máximo esperado

**Facturación:** Máximo 3 segundos

**Búsqueda de productos:** Máximo 1 segundo

**Inicio de sesión:** Máximo 1 segundo

**Gestión de inventario (ingreso, actualización, eliminación):** Máximo 1 segundo

**Envío de correo de recuperación de contraseña:** Máximo 5 segundos

**Creación de usuario:** Máximo 2 segundos

**Búsqueda de facturas:** Máximo 3 segundos

**Envío de factura por correo (PDF):** Máximo 20 segundos

**Ordenamiento de productos por ventas:** Máximo 2 segundos

**Relaciones con otras entradas/salidas:** No aplica.

**Formatos de datos:** La entrada de datos no tiene un formato específico, ya que se reciben como identificadores separados. La salida de datos es en formato JSON.

**Formatos de comandos:** No aplica.

**Interfaz Backend Deno**

**Nombre:** Backend Deno

**Descripción del propósito:** El backend es responsable de gestionar la lógica de negocio, la conexión con la base de datos y la comunicación con la API del Ministerio de Hacienda. Administra el registro de usuarios, el inicio de sesión y la facturación electrónica.

**Fuente de la entrada o destino de la salida:** La entrada de datos se realiza a través de las rutas o endpoints del backend. La salida de datos se comunica con la base de datos mediante sockets TCP.

**Rango válido, precisión y/o tolerancia:** No aplica.

**Unidades de medida:** Cantidad de unidades o paquetes.

**Temporización:** No aplica.

**Relaciones con otras entradas/salidas:** No aplica.

**Formatos de datos:** La entrada se recibe en formato JSON, y la salida de datos para las facturas electrónicas es en formato XML. Los datos que se envían a la base de datos también se transmiten en formato JSON o como parámetros en procedimientos almacenados.

**Formatos de comandos:** deno <acción> [opciones] <archivo>

**Base de Datos**

**Nombre:** Cliente Electron.js

**Descripción del propósito:** La base de datos se encarga de almacenar toda la información relevante sobre la empresa, productos, usuarios y facturas. Proporciona almacenamiento persistente para los datos del sistema.

**Fuente de la entrada o destino de la salida:** La entrada de datos es a través de los parámetros de los procedimientos almacenados enviados desde el backend. Los datos procesados son devueltos al backend.

**Rango válido, precisión y/o tolerancia:** No aplica.

**Unidades de medida:** Cantidad de unidades o paquetes.

**Temporización:** No aplica.

**Relaciones con otras entradas/salidas:** No aplica.

**Formatos de datos:** La entrada se realiza a través de parámetros en procedimientos almacenados. Los datos se envían como objetos con una estructura similar a JSON.

**Formatos de comandos:** Lenguaje SQL.

**API Ministerio de Hacienda**

**Nombre:** API Open Source de CRLibre

**Descripción del propósito:** Esta API facilita el proceso de facturación electrónica en Costa Rica, permitiendo que el backend interactúe con el Ministerio de Hacienda para registrar las facturas electrónicas.

**Fuente de la entrada o destino de la salida:** La entrada a la API proviene del backend de la empresa a través de una solicitud POST. La salida es recibida por el backend de la empresa.

**Rango válido, precisión y/o tolerancia:** No aplica.

**Unidades de medida:** No aplica.

**Temporización:** No aplica.

**Relaciones con otras entradas/salidas:** Se desconoce, ya que se trata de un sistema de terceros.

**Formatos de datos:** Los datos se reciben en formato JSON y la respuesta es en formato XML.

**Formatos de comandos:** No aplica.

**Correo Electrónico**

**Nombre:** Denomailer

**Descripción del propósito:** Esta librería es utilizada para enviar correos electrónicos a los clientes con los archivos de las facturas electrónicas (PDF, XML y acuse de recibido).

**Fuente de la entrada o destino de la salida:** La entrada proviene del backend de la empresa, y la salida es dirigida a los clientes a través de correo electrónico.

**Rango válido, precisión y/o tolerancia:** No aplica.

**Unidades de medida:** No aplica.

**Temporización:** No especificado en el documento.

**Relaciones con otras entradas/salidas:** Se desconoce, ya que se trata de un sistema de terceros.

**Formatos de datos:** La entrada es en formato JSON y no devuelve respuesta.

**Formatos de comandos:** No aplica.

## 3.4. Requisitos de rendimiento

**1) Requisitos del Servidor**

**Servidor Virtual**

* **Plan GLORIA VPS S:**

**Sistema Operativo:** Ubuntu Server 22.04

**Especificaciones:**

1 núcleo de procesador virtual

2GB de RAM

50GB de almacenamiento interno

Ancho de banda: 500Mbps

Transferencia mensual: 2TB

Estas especificaciones son suficientes para las operaciones actuales de la PYME. En caso de que la empresa crezca, se recomienda considerar el plan GLORIA VPS M, con las siguientes especificaciones mejoradas:

* **Plan GLORIA VPS M**

**Sistema Operativo:** Ubuntu Server 22.04

**Especificaciones:**

2 núcleos de procesador

4GB de RAM

80GB de almacenamiento interno

Ancho de banda: 500Mbps

Transferencia mensual: 4TB

Este plan puede soportar sin problemas las exigencias de un negocio en crecimiento. Además, los servidores virtuales permiten mejorar el hardware sin necesidad de migraciones. Según la recomendación de la IA de hosting506, Deno y PostgreSQL deberían funcionar perfectamente en el plan GLORIA VPS S, siempre que el tráfico de datos de la empresa sea bajo.

**2) Requisitos Mínimos de Hardware del cliente:**

* **Almacenamiento Interno:** 500MB
* **Memoria RAM:** 300MB
* **Procesador:** Intel i3 N305
* **Conexión a Internet:** 10Mbps de descarga y 10Mbps de subida

**3) Requisitos de Rendimiento:**

El sistema debe ser capaz de soportar tres usuarios concurrentes realizando operaciones de facturación. El sistema debe permitir facturar hasta 200 productos diferentes por factura.

**Seguridad de Datos:** Los datos sensibles (contraseñas, pines, etc.) deben estar ocultos o censurados. Se debe permitir ver estos datos de manera opcional.

**4) Requisitos de Tiempo de Procesamiento**

**Facturación:** El tiempo de respuesta desde que se presiona el botón "Facturar" hasta recibir el mensaje de éxito o fracaso no debe superar los 3 segundos.

**Búsqueda de Producto:** El tiempo para buscar un producto para facturación no debe superar el 1 segundo.

**Inicio de Sesión:** El tiempo desde que se presiona "Iniciar sesión" hasta completar el inicio de sesión no debe superar los 1 segundo.

**Gestión de Productos:** El tiempo para ingresar, actualizar o eliminar un producto del inventario no debe superar el 1 segundo.

**Recuperación de Contraseña:** El correo para recuperar contraseñas no debe tardar más de 5 segundos.

**Creación de Usuario:** El tiempo para crear un usuario no debe superar los 2 segundos.

**Búsqueda de Factura:** El tiempo para buscar una factura no debe superar los 3 segundos.

**Envío de Factura por Correo Electrónico:** El tiempo para enviar la factura en formato PDF al correo del cliente no debe superar los 20 segundos, siempre que el sistema del Ministerio de Hacienda lo permita.

**Ordenar Productos en Inventario:** El tiempo para ordenar los productos del inventario por cantidad de ventas no debe superar los 2 segundos.

**5) Requisitos de Seguridad**

**Comunicaciones Seguras:** Se debe utilizar certificado HTTPS para la comunicación entre el cliente y el servidor.

**Restricciones de Entrada:** Los datos de entrada deben estar restringidos a 50 caracteres (excepto para descripciones de productos, que pueden tener un máximo de 200 caracteres).Esta restricción debe ser aplicada y verificada en el backend.

**Métodos de Enrutamiento**: Solo se deben utilizar métodos POST en el enrutador del backend.

**Cross-Origin Resource Sharing (CORS):** Se debe habilitar CORS entre el cliente y el backend.

**Limitación de Caracteres en el Cliente:** Las entradas de texto en el cliente deben estar limitadas a 50 caracteres, con descripciones que no excedan los 200 caracteres.

**Conjunto de Caracteres Aceptado:** El sistema debe aceptar únicamente el conjunto de caracteres extendido de ASCII.

## 3.5. Requisitos de la base de datos lógica

Para la base de datos se requiere un servidor virtual como los mencionado anteriormente, ya que, por el momento se ejecutarán el backend y la base de datos se ejecutarán en el mismo servidor, se requieren al menos 25GB de almacenamiento solamente para guardar los datos que genere la empresa. La base de datos que se utilizará es PostgreSQL, ya que es de licencia libre para uso comercial. Esta base de datos se utilizará todos los días de la semana, aproximadamente durante 18 horas al día. La base de datos se ejecutará en un solo servidor, por lo que no se requiere realizar ningún tipo de replicación. Las facturas deben almacenarse al menos durante 5años consecutivos.

Los aspectos de seguridad que debe cumplir la base de datos son lo siguientes:

* Se debe utilizar un certificado SSL para la comunicación entre el backend y la base de datos.
* Solamente se debe acceder a los datos de la base de datos por medio de procedimientos almacenados y funciones.
* El backend debe utilizar un usuario que solo tenga acceso a los procedimientos almacenados y funciones.
* La base de datos se debe acceder desde el backend para realizar las operaciones de la empresa. Los desarrolladores si pueden ingresar directamente.

Ahora, se procede a mostrar el diseño de la base de datos del sistema. Esto se representará mediante un diagrama de relacional.

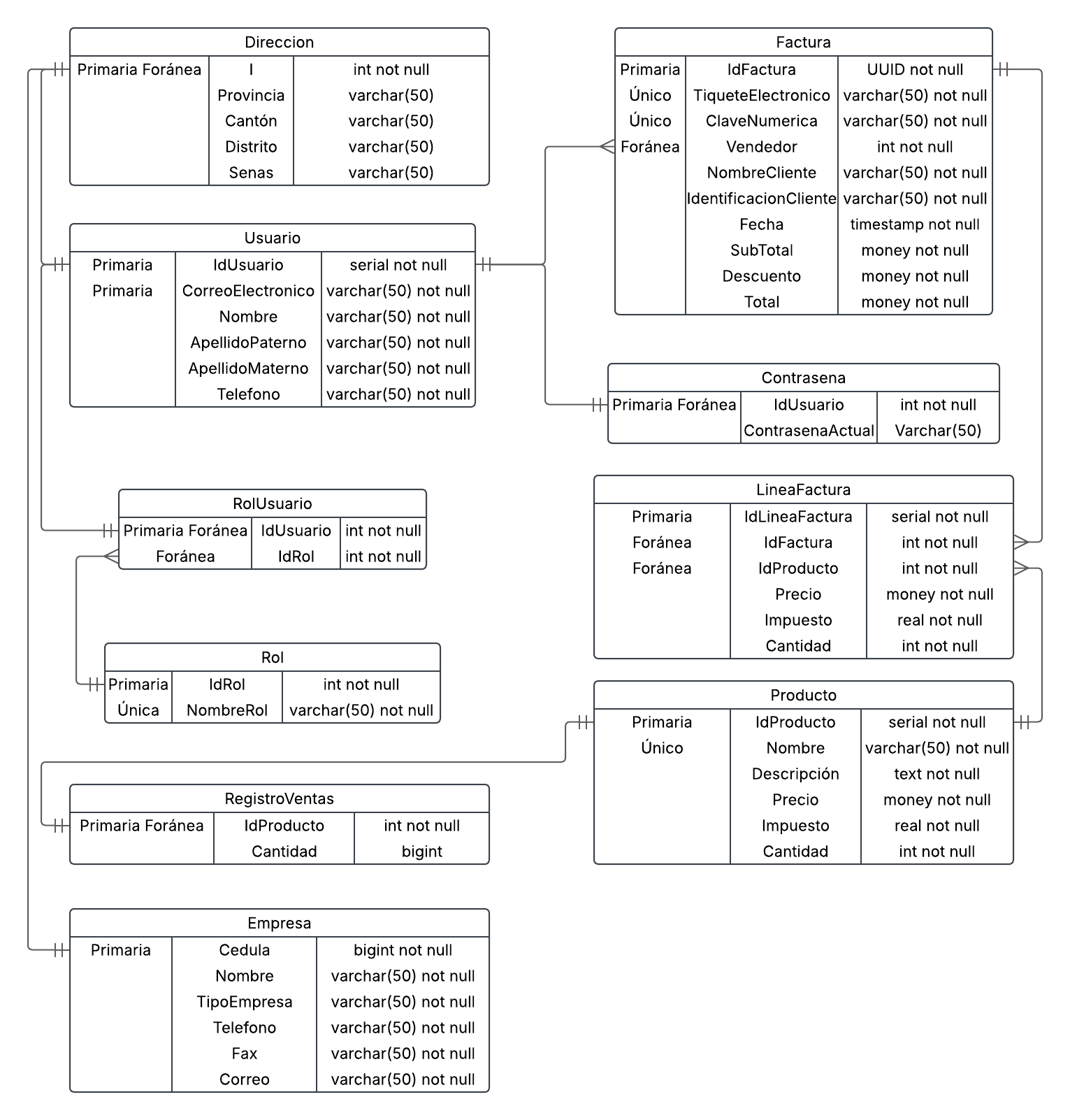


Diagrama 6 - Diagrama relacional de la base de datos

Las funciones necesarias para que el sistema funcione correctamente deben recibir los siguientes datos:

* Para el inicio de sesión se requiere del correo electrónico y de la contraseña.
* Para crear un usuario se requiere del nombre, apellido paterno, apellido materno, número de teléfono, provincia de residencia, cantón de residencia, distrito de residencia y señas del lugar.
* Para crear/modificar un producto se requiere del número consecutivo o número de barras, nombre del producto, descripción del producto, precio, porcentaje del impuesto del impuesto, cantidad de productos ingresados.
* Para buscar un producto se necesita del número de consecutivo o nombre del producto, cualquiera de las dos se puede aceptar en una sola función.
* Para eliminar un producto se necesita del número del consecutivo.
* Para solicitar los productos ordenados por número de ventas, ya sea de menor a mayor o viceversa, se solicita 1 si es de menor a mayor y 2 si es de mayor a menor. Estos datos son varchar.
* Para facturar una compra se necesitan los siguientes datos se enviaran los datos de la factura en formato JSON a un procedimiento almacenado que luego guardará los datos en las tablas. El formato del JSON es el siguiente “{vendedor(int), nombreCliente(varchar(50)), identificacion(varchar(50)), subtotal(money), total(money), productos(json[])}”, la fecha se calcula en el servidor de la base de datos. El formato de los productos es el siguiente {idProducto(int), precio(money), impuesto(real), cantidad(int)}
* Para buscar una factura se debe ingresar el número de factura.

## 3.6. Restricciones de diseño

El sistema debe cumplir con la Ley de Protección de la Persona frente al tratamiento de sus datos personales - Nº 8968. Debido a que, se guarda información personal de los usuarios del sistema.

Para las facturas se debe cumplir el Reglamento de comprobantes electrónicos para efectos tributarios No. 41820-H

## 3.7. Atributos del sistema de software

**Confiabilidad:** Este programa debe presentar una confiabilidad del 99%. Ya que es un sistema esencial para las operaciones de SRL Technologies. Los factores para garantizar esta confiabilidad son los siguientes:

* Se asegurará la calidad del software por medio de un plan de pruebas en cada sprint.
* Se utilizará una interfaz sencilla de usar con mensajes claros.
* Se utiliza un servidor virtual de una empresa nacional para garantizar mayor velocidad desde el cliente hasta el servidor.
* Se utilizan tecnologías seguras para evitar al máximo cualquier vulnerabilidad.
* Se dará soporte a los clientes para que estos se sientan más confiados con el sistema.

**Disponibilidad:** El sistema está disponible el 99%. El tiempo no disponible es por las noches para realizar mantenimiento al servidor. Los factores para asegurar la disponibilidad son los siguientes:

* Se utiliza un servidor virtual de hosting506 que asegura tener un tiempo en línea del 99.8%.
* Se utiliza un ancho de banda de 500Mbps que asegura que los tres usuarios puedan conectarse simultáneamente.
* Se utiliza un sistema asincrónico de un solo hilo que asegura que el servidor pueda ejecutarse con especificaciones bajas.
* Se utiliza un servidor lo suficientemente potente para las operaciones de SRL Technologies.

**Seguridad:** Se opta por un diseño seguro para evitar vulnerabilidades. Los factores para garantizar la seguridad son los siguientes:

* Se utiliza TypeScript como lenguaje para programar, debido a que es evita el desbordamiento de buffer.
* Se utilizan try/catch para evitar un ataque DOS por excepciones en la ejecución.
* Se utiliza un certificado HTTPS en la comunicación entre el cliente y el servidor para encriptar la información.
* Solamente se utilizarán métodos POST en las peticiones para evitar exponer los datos en la URL.
* Se limita el máximo de datos escritos en todo el sistema a 50 caracteres y 200 caracteres para descripciones.
* Se utiliza cors entre el cliente y el servidor para evitar que se pueda acceder al servidor desde otro cliente no oficial.
* Se utiliza Deno que es un ambiente de servidor bastante seguro.
* Se utiliza un certificado SSL entre el backend y la base de datos para encriptar los datos.
* El backend utiliza un usuario que solo tiene acceso a procedimientos almacenados.
* Los sistemas a instalar no deben instalarse con super usuarios.

**Mantenibilidad:** Para facilitar la mantenibilidad del producto, se propone realizar lo siguiente:

* Utilizar TypeScript que mejora considerablemente el uso de interfaces.
* Modularizar las funciones de la aplicación.
* Utilizar interfaces en las dependencias, no se permite depender de clases concretas.
* Utilizar el paradigma orientado a objetos en el backend.

**Portabilidad:** Para cumplir con la portabilidad de tienen los siguientes requisitos:

* Crear el backend con Deno, esto permite trasladar el proyecto a otra máquina en caso de ser necesario y no requiere mucha configuración.
* Crear el frontend con Tauri, ya que este puede realizar compilaciones a otros sistemas operativos y a una gran variedad de arquitecturas de microprocesador.
* Se utiliza la base de datos PostgreSQL ya que tiene distribuciones en muchos sistemas operativos, incluyendo los que se necesitan.

## 3.8. Información de soporte

Se procede a presentar los siguientes elementos que **no deben considerarse como requisitos** para el sistema que se está construyendo, por lo que solo es información que sirve para mostrar a nivel general lo que está realizando el sistema. A continuación, se procede a mostrar los aspectos:

1. **Formatos de Entrada/Salida de Muestra**

Se incluirán ejemplos de los formatos de entrada y salida del sistema, como el formato JSON que se utiliza para la comunicación entre el cliente y el backend, así como los archivos XML generados para la facturación electrónica.  
*Ejemplo de entrada de datos al backend*:

{

"nombre\_producto": "Producto A",

"cantidad": 10,

"precio": 500

}

*Ejemplo de salida de datos XML al API del Ministerio de Hacienda*:

<?xml version="1.0" encoding="UTF-8"?>

<FacturaElectronica>

<Clave>5060204000012345678901234567890123456789012345</Clave>

<NumeroConsecutivo>1234567890</NumeroConsecutivo>

<Emisor>

<NumeroIdentificacion>3101526147</NumeroIdentificacion>

</Emisor>

<Receptor>

<NumeroIdentificacion>123456789</NumeroIdentificacion>

</Receptor>

<Detalle>

<Linea>

<Descripcion>Producto A</Descripcion>

<MontoTotal>1000.00</MontoTotal>

</Linea>

</Detalle>

</FacturaElectronica>

1. **Estudios de Análisis de Costos y Resultados de Encuestas a Usuarios**

Basado en la entrevista que se tuvo con uno de los usuarios y la oferta de herramientas de para realizar el proyecto. El costo del sistema solo debe ser máximo de 25000 colones, ya que la empresa no cuenta con los recursos suficientes para mantener el sistema. El desglose del costo del sistema se presenta a continuación:

* **Servidor virtual:** 20 dólares (10000 colones aproximadamente)

El sistema contará con ese único costo por el momento.

1. **Información de Respaldo o Contexto**

El sistema está diseñado para cubrir las necesidades de facturación electrónica de la empresa SRL Technologies. La solución debe integrar de manera eficiente el manejo de inventarios y la generación de facturas electrónicas, cumpliendo con las regulaciones fiscales del Ministerio de Hacienda de Costa Rica.  
Además, se ha considerado que el sistema debe ser accesible desde varias estaciones de trabajo conectadas a una red local.

1. **Descripción de los Problemas a Resolver**

El sistema busca resolver los siguientes problemas:

* + **Facturación manual:** El proceso de facturación es actualmente manual, lo que genera errores y pérdida de tiempo.
  + **Gestión de inventarios:** La falta de un sistema automatizado para el control de inventarios dificulta el seguimiento de productos.
  + **Cumplimiento de normativas fiscales:** El sistema debe asegurar que las facturas electrónicas generadas cumplan con las normativas del Ministerio de Hacienda.

1. **Instrucciones Especiales de Empaque**

El producto será entregado de manera funcional a SRL Technologies, por lo que no hace falta empaquetar el cliente para como instalable en los equipos de la empresa.

# 4. Verificación

Tabla 1 - Verificación de los requerimientos

|  |  |  |
| --- | --- | --- |
| Requerimiento | Verificación | Método |
| RS001: El sistema debe iniciar sesión. | Se realizará una prueba funcional en la que se ingresarán diferentes combinaciones de nombre de usuario y contraseña para verificar que el sistema permita el acceso solo con credenciales válidas. También se verificará el comportamiento cuando se ingresen credenciales incorrectas. | Pruebas unitarias y pruebas de integración. |
| RS002: El sistema debe crear usuarios. | Se probará la creación de usuarios con diferentes roles y permisos para verificar que se asignen correctamente. Se comprobará que los usuarios creados puedan acceder a las funcionalidades correspondientes a sus roles. | Pruebas funcionales y pruebas de aceptación del usuario. |
| RS003: El sistema debe permitir la recuperación de contraseñas. | Se verificará que el sistema envíe un correo electrónico de recuperación a la dirección asociada a la cuenta y que la nueva contraseña funcione para ingresar nuevamente al sistema. | Pruebas de integración de correo electrónico y pruebas de funcionalidad de restablecimiento de contraseña. |
| RS004: El sistema debe crear productos al inventario. | Se verificará que al ingresar los detalles del producto el producto se registre correctamente en el sistema y sea visible en la lista de inventarios. | Pruebas de unidad y pruebas funcionales. |
| RS005: El sistema debe actualizar el stock del inventario. | Se agregarán nuevos productos y stock de productos existentes y se verificará sin el sistema actualiza los datos. | Pruebas de integración y pruebas de validación de base de datos. |
| RS006: El sistema debe mostrar los productos del inventario. | Se verificará que los productos sean visibles en una lista, con sus detalles completos, y que la interfaz se cargue correctamente. | Pruebas de interfaz de usuario y pruebas funcionales. |
| RS007: El sistema debe borrar los productos del inventario. | Se realizará una prueba para eliminar productos y verificar que estos ya no aparezcan en la lista del inventario. | Pruebas de eliminación de base de datos. |
| RS008: El sistema debe buscar productos por consecutivo o nombre. | Se realizará una prueba de búsqueda utilizando nombres y consecutivos de productos para verificar que la búsqueda sea precisa y rápida. | Pruebas de búsqueda y pruebas de rendimiento. |
| RS009: El sistema debe ordenar productos por cantidad de ventas. | Se verificará que los productos se ordenen correctamente según la cantidad de ventas y que la lista se actualice dinámicamente con nuevos datos. | Pruebas de integración y pruebas de interfaz de usuario. |
| RS010: El sistema debe imprimir facturas. | Se verificará que el sistema permita la generación de facturas en el formato adecuado y que la impresión sea exitosa. | Pruebas de funcionalidad de generación de documentos y pruebas de impresión. |
| RS011: El sistema debe crear el archivo XML con el formato exigido por hacienda para la facturación electrónica. | Se generarán facturas y se verificará que el archivo XML generado cumpla con el formato exigido por el Ministerio de Hacienda. | Pruebas de validación de archivos y pruebas de compatibilidad. |
| RS012: El sistema debe enviar el archivo XML a hacienda para registrar la factura electrónica. | Se realizará una prueba de integración en la que se envíen archivos XML al API de Hacienda y se verifique que la transacción sea exitosa. | Pruebas de integración con el API de Hacienda. |
| RS013: El sistema puede enviar opcionalmente el PDF de factura, XML de factura y el XML de acuse de recibido a un cliente por medio de correo electrónico. | Se verificará que el sistema pueda enviar los archivos correctamente al correo electrónico del cliente y que se reciban en el formato adecuado. | Pruebas de integración con el servicio de correo electrónico. |
| RS014: El sistema debe facturar una compra. | Se verificará que el sistema pueda generar una factura correctamente a partir de una compra realizada, incluyendo todos los detalles de los productos y precios. | Pruebas de generación de facturas. |
| RS015: El sistema debe mostrar las facturas creadas. | Se verificará que las facturas generadas se muestren correctamente en la lista de facturas. | Pruebas de interfaz de usuario. |
| RS016: El sistema debe buscar facturas por número de factura. | Se verificará que el sistema permita buscar facturas por su número y que los resultados sean correctos. | Pruebas de funcionalidad de búsqueda. |
| RS017: El sistema debe permitir el ingreso de productos en una factura. | Se probará que los productos puedan ser añadidos a la factura durante su creación y que los detalles sean correctos. | Pruebas de creación de factura. |
| RS018: El sistema debe buscar productos mientras se crea la factura. | Se verificará que el sistema permita buscar productos en tiempo real mientras se está creando la factura. | Pruebas de funcionalidad de búsqueda en tiempo real. |
| RS019: El sistema debe actualizar los datos de la empresa. | Se verificará que los datos de la empresa (dirección, teléfono, correo electrónico) se actualicen correctamente en el sistema. | Pruebas de actualización de base de datos. |

# 5. Apéndices

## 5.1. Supuestos y dependencias

**Acceso a internet**

**Supuestos:** Se asume que la empresa SRL Technologies tiene acceso a internet en su local.

**Dependencias:** Si la empresa no cuenta con internet, el sistema no se puede utilizar, debido a que se requiere conexión a internet para conectarse a la base de datos y al backend.

**Computadoras**

**Supuestos:** La empresa debe contar con tres computadoras que cumplen con los requisitos mínimos de hardware para ejecutar la aplicación de facturación.

**Dependencias**: Al no tener al menos una computadora con los requisitos mínimos de hardware, la empresa no puede utilizar el sistema.

**Dinero**

**Supuestos:** La empresa cuenta con los recursos necesarios para pagar la mensualidad del servidor virtual de hosting506.

**Dependencias:** Si la empresa no cuenta con el dinero mensual para pagar el servidor el servicio puede verse cortado, debido a que hosting506 apagará el servidor.

**API Ministerio de Hacienda**

**Supuestos:** El Ministerio de Hacienda siempre tiene el API de facturación electrónica.

**Dependencias:** Si el sistema de facturación del Ministerio de Hacienda se ve afectado, la aplicación no va a funcionar correctamente al momento de generar una factura electrónica.

**Correo electrónico**

**Supuestos:** El sistema de GMAIL funciona correctamente todo el tiempo.

**Dependencias:** SI GMAIL no está funcionando correctamente los correos electrónicos no le van a llegar a los compradores de SLR Technologies.

**hosting506 disponibilidad**

**Supuestos:** Los servidores de hosting506 están activos durante el 99.8% del tiempo.

**Dependencias:** Los servicios de hosting506 pueden verse afectados durante más tiempo de lo previsto, ocasionando que el sistema no se pueda acceder hasta que los sistemas de hosting506 se vuelvan a restablecer.

**hosting506 persistencia de datos**

**Supuestos:** La información en hosting506 es improbable que se pierda.

**Dependencias:** Si SRL Technologies perdiera los datos de su inventario, podría perder mucho dinero hasta que se pueda recuperar.

**Integridad del sistema**

**Supuestos:** Se utilizan protocolos de seguridad para evitar la vulneración de los datos en el sistema.

**Dependencias:** Si algún atacante decidiera vulnerar el sistema, la privacidad de la información se vería comprometida y la empresa puede sufrir demandas por no proteger los datos.

**Integridad de la base de datos**

**Supuestos:** La base de datos no dañará, alterará o eliminará los datos del inventario.

**Dependencias:** El daño, alteración o eliminación de los datos de la base de datos puede provocar inconsistencia en el inventario y en el peor de los casos pérdidas de dinero.

**Backend disponibilidad**

**Supuestos:** El backend estará disponible siempre que se le realice una petición.

**Dependencias:** Si el backend no está disponible y requiere de un reinicio, no va a haber forma de que los usuarios puedan reiniciarlo, ya que no saben de computación avanzada.

## 5.2. Matriz de trazabilidad

Tabla - Matriz de trazabilidad

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID Requerimiento | Nombre | Asociado a | Categoría | Prioridad | Estimado Inicial |
| RS001 | Inicio de sesión | RS002, RS003 | Funcional | Alta | 8 horas |
| RS002 | Creación de usuarios | RS001 | Funcional | Alta | 10 horas |
| RS003 | Recuperación de contraseñas | RS001 | Funcional | Alta | 6 horas |
| RS004 | Creación de productos en inventario | RS005, RS006, RS007 | Funcional | Alta | 8 horas |
| RS005 | Actualización de stock | RS004, RS006 | Funcional | Alta | 7 horas |
| RS006 | Visualización del inventario | RS004, RS005, RS007, RS008, RS009 | Funcional | Alta | 6 horas |
| RS007 | Eliminación de productos | RS004, RS006 | Funcional | Media | 6 horas |
| RS008 | Búsqueda de productos | RS006, RS018 | Funcional | Alta | 7 horas |
| RS009 | Ordenamiento por ventas | RS006 | Funcional | Baja | 6 horas |
| RS010 | Impresión de facturas | RS014, RS016 | Funcional | Alta | 10 horas |
| RS011 | Generación de XML | RS014, RS012 | Funcional | Alta | 24 horas |
| RS012 | Envío del XML a Hacienda | RS011 | Funcional | Alta | 72 horas |
| RS013 | Envío de factura por correo | RS014, RS011 | Funcional | Media | 8 horas |
| RS014 | Facturación de una compra | RS017, RS018, RS010, RS011 | Funcional | Alta | 10 horas |
| RS015 | Visualización de facturas | RS014, RS016 | Funcional | Media | 7 horas |
| RS016 | Búsqueda de facturas | RS015 | Funcional | Media | 6 horas |
| RS017 | Ingreso de productos en factura | RS014, RS018 | Funcional | Alta | 8 horas |
| RS018 | Búsqueda de productos en factura | RS017, RS008 | Funcional | Alta | 7 horas |
| RS019 | Actualización de datos de la empresa | - | Funcional | Media | 6 horas |