



UNIVERSIDAD DE CONCEPCIÓN  
FACULTAD DE INGENIERÍA

# RENOVACIÓN DE PUNTO DE VENTAS DE OPENCODE: MEJORA DEL RENDIMIENTO, CALIDAD DEL SOFTWARE Y EXPERIENCIA DEL CLIENTE

**Por: Sebastián Rodrigo Águila Montecino**

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Informático

Julio 2024

Concepción, Chile

**Profesor Patrocinante: Geoffrey Hecht**

**Supervisor: Luis Alberto Meneses González**

© 2024, Sebastián Rodrigo Águila Montecino

Ninguna parte de esta tesis puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso por escrito del autor.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento

# AGRADECIMIENTOS

Quiero expresar mis más profundos agradecimientos a todos quienes han sido parte de este largo proceso universitario. Agradezco de corazón a todos los que me han brindado su apoyo, ayuda y compañía:

A mis padres, Mabel Montecino y Rodrigo Águila, y a mi hermano Benjamín Águila, quienes como familia siempre estuvieron a mi lado, alentándome y apoyándome durante toda esta etapa.

A mi novia, Dalett Marín, quien, al igual que mi familia, ha sido una compañera leal que ha estado siempre presente, haciendo de la universidad un proceso mucho más ameno y llevadero.

A don Alberto Meneses, por haberme permitido trabajar en su empresa Opencode y desarrollar este proyecto junto a él.

A mi profesor guía, Geoffrey Hecht, por permitirme estar bajo su tutela, por su disposición y atención al detalle. Esta memoria no hubiera sido lo mismo sin sus valiosos consejos.

Finalmente, quiero agradecer a todos mis amigos que he hecho durante estos cinco años. Gracias por haber formado parte de esta etapa, brindándome su ayuda, compañía, risas y buenas experiencias, contribuyendo a que haya disfrutado mucho más de esta etapa.

## Resumen

Opencode, una empresa de desarrollo de software empresarial ubicada en Concepción, ofrece una vasta gama de soluciones para la administración, gestión y operación de negocios. Uno de sus productos, el Punto de Venta Web, presenta ciertas deficiencias en su rendimiento, calidad de su código y diseño. Además, no presenta ciertas características imprescindibles en este tipo de aplicaciones y resulta complicado su uso en ciertas situaciones para los clientes.

Por estos motivos, la presente memoria de título tiene como objetivo desarrollar un nuevo Punto de Venta Web para OpenCode, el cual mejore todos los aspectos deficientes que tiene el actual software. Esta actualización busca modernizar el sistema, mejorar la eficiencia de operación, la satisfacción del cliente y por ende la competitividad en el mercado de la compañía.

## Abstract

Opencode, a business software company located in Concepción, offers a wide range of solutions for business administration, management, and operations. One of its products, the web-based Point of Sale, has certain deficiencies in performance, code quality, and design. Furthermore, it lacks certain essential features for this type of application and is complicated for customers to use in certain situations.

For these reasons, this thesis aims to develop a new web-based Point of Sale for Opencode, which will improve all the deficient aspects of the current software. This update seeks to modernize the system, improve operational efficiency, customer satisfaction, and consequently, the company's market competitiveness.

# Índice

<b>AGRADECIMIENTOS</b>	<b>I</b>
<b>Resumen</b>	<b>II</b>
<b>Abstract</b>	<b>III</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Objetivos</b>	<b>2</b>
2.1. Objetivos Generales . . . . .	2
2.2. Objetivos Específicos . . . . .	2
<b>3. Antecedentes y Problemática</b>	<b>3</b>
3.1. Punto de Venta actual de Opencode . . . . .	3
3.1.1. Características . . . . .	3
3.1.2. Problemas y Carencias del Software . . . . .	4
<b>4. Propuesta de Mejora</b>	<b>6</b>
<b>5. Metodología</b>	<b>8</b>
5.1. Beneficios de la Metodología Seguida . . . . .	8
5.2. Requerimientos . . . . .	9
5.2.1. Requerimientos funcionales . . . . .	9
5.2.2. Requerimientos no funcionales . . . . .	11
<b>6. Diseño y Arquitectura</b>	<b>12</b>
6.1. Estructura . . . . .	12
6.2. Tecnologías Usadas . . . . .	13
6.3. Base de Datos . . . . .	15
6.4. Nuevo Punto De Venta . . . . .	18
6.4.1. Flujo de trabajo y funcionamiento . . . . .	18
6.4.2. Inicio de sesión y de turno de trabajo . . . . .	18
6.4.3. Implementación POS de Transbank . . . . .	21
6.4.4. Facturación Electrónica SII . . . . .	23
6.4.5. Simplificación de la interfaz . . . . .	25
<b>7. Pruebas y Resultados</b>	<b>36</b>
7.1. Medición de usabilidad con SUS . . . . .	36
7.2. Pruebas API . . . . .	40
7.3. Métricas de software . . . . .	45
7.3.1. Análisis con Visual Studio . . . . .	45
7.4. Pruebas con Sonarqube . . . . .	47
<b>8. Conclusión</b>	<b>49</b>

Índice

v

---

**Referencias**

**50**

## Índice de figuras

5.1. Diagrama de casos de uso del nuevo Punto de Venta . . . . .	11
6.1. Diagrama C4 del nuevo Punto de Venta . . . . .	13
6.2. Diagrama ER de la BD del Punto de Venta . . . . .	15
6.3. Diagrama de flujo del inicio de sesión del Punto de Venta actual .	19
6.4. Diagrama de flujo del nuevo inicio de sesión del Punto de Venta .	21
6.5. Visualización de la aplicación de escritorio de Transbank . . . . .	23
6.6. Comparación inicios de sesión . . . . .	26
6.7. Antes y después ventana Establecer Empresa . . . . .	27
6.8. Diferencias en el bloqueo de las cajas . . . . .	28
6.9. Modales de cierre de cajas . . . . .	29
6.10. Comparativa ventana de inicio de turno . . . . .	30
6.11. Comparativa de los paneles principales . . . . .	31
6.12. Métodos de pago Punto de Venta actual . . . . .	32
6.13. Modal métodos de pago nuevo Punto de Venta . . . . .	33
6.14. Comparativa de los paneles de administrador . . . . .	34
6.15. Modal operaciones de documentos POS nuevo . . . . .	35
7.1. Resultados encuesta SUS . . . . .	37
7.2. Gráfica resultado encuesta SUS . . . . .	40
7.3. Resultados Login controlador Auth . . . . .	40
7.4. Resultados Login controlador Usuarios . . . . .	41
7.5. Resultados IniciarTurno . . . . .	41
7.6. Resultados VerEstadoTurno . . . . .	42
7.7. Resultados ValidarSupervisor . . . . .	42
7.8. Resultados GuardarVenta . . . . .	43
7.9. Resultados ObtenerDocumentosCaja . . . . .	43
7.10. Resultados BuscarProducto . . . . .	44
7.11. Resultados Codigo . . . . .	44
7.12. Resultados BuscarCliente . . . . .	45
7.13. Métricas Visual Studio Punto de Venta actual . . . . .	45
7.14. Métricas Visual Studio Punto de Venta nuevo . . . . .	46
7.15. Métricas Sonarqube . . . . .	47

# 1. Introducción

Opencode, un negocio dedicado al desarrollo de software ubicado en la ciudad de Concepción, Chile, ofrece una amplia gama de productos de software de escritorio y web. Entre sus productos, cuentan con un Punto de Venta Web, el cual presenta deficiencias en su funcionamiento, carece de ciertas características y resulta algo complejo en su uso. Este proyecto tiene como objetivo principal mejorar la eficiencia, funcionalidad y experiencia del usuario al realizar transacciones comerciales mediante el desarrollo de una nueva versión moderna y optimizada del software.

Un Punto de Venta, comúnmente conocido como POS, es un sistema informático que permite gestionar las transacciones comerciales en un establecimiento, como tiendas, restaurantes o empresas de servicios. Este sistema integra diversas funcionalidades, como el control de inventario, el procesamiento de pagos, la emisión de facturas y la gestión de clientes. En el caso de Opencode, este punto de venta está conectado con su software de gestión integral de empresas, conocido por sus siglas en inglés ERP (Enterprise Resource Planning), que incluye todas las herramientas y procesos necesarios para dirigir una empresa exitosamente.

Considerando las limitaciones del POS actual de la empresa, caracterizado por su lentitud, diseño anticuado y lógica de funcionamiento complicada, se plantea la implementación de un nuevo POS basado en tecnologías actuales y enfoques innovadores. Para ello, se desarrollará una API en ASP .NET Core para la gestión de datos y la integración con sistemas existentes. ASP .NET Core es una plataforma de aplicaciones gratuita y de código abierto respaldada por Microsoft, sobre la cual Opencode basa toda su infraestructura de datos. Además, se ha elegido el framework Next.js para la creación de un frontend ágil y atractivo.

Para la concreción de este proyecto se adoptó una metodología adaptada a las condiciones de trabajo en la empresa e inspirada en el desarrollo de software secuencial, donde cada semana se establecieron objetivos donde todos debían ser completados para pasar a las siguientes fases.

Durante la extensión de este informe serán explicados los requerimientos del nuevo software, las mejoras del nuevo sistema frente al antiguo, se demostrará que la aplicación responde en tiempos dentro de lo deseado por la empresa y que el

software cumple con las expectativas de los clientes de Opencode, para finalmente concluir con una reflexión acerca de todo el proceso de desarrollo de este proyecto.

## 2. Objetivos

### 2.1. Objetivos Generales

El propósito general de este proyecto de memoria de título es reemplazar el antiguo punto de ventas (POS) de Opencode por una versión nueva y mejorada, abordando las deficiencias del POS actual mencionadas anteriormente.

Así, el objetivo principal es mejorar significativamente el rendimiento, diseño y funcionalidad del POS actual de Opencode mediante el desarrollo de una nueva versión que responda con eficiencia a las demandas del mercado y las expectativas de los usuarios.

### 2.2. Objetivos Específicos

1. Optimizar la velocidad de procesamiento y respuesta del nuevo POS para agilizar las transacciones comerciales y mejorar la experiencia del usuario.
2. Diseñar una interfaz moderna e intuitiva que facilite la navegación y utilización del sistema tanto para los empleados como para los clientes.
3. Implementar una lógica de funcionamiento coherente y eficaz que garantice la precisión y fiabilidad de las operaciones realizadas en el POS.
4. Asegurar la compatibilidad y adaptabilidad del nuevo POS con los sistemas existentes de Opencode para una transición fluida y sin interrupciones.
5. Entregar un proyecto con código limpio y buenas prácticas de desarrollo que permita a futuros desarrolladores mantener e integrar nuevas características fácilmente.
6. Obtener la aprobación y satisfacción del dueño de la empresa con la entrega de un POS mejorado que cubra de manera integral las necesidades y expectativas de sus clientes.

## 3. Antecedentes y Problemática

### 3.1. Punto de Venta actual de Opencode

#### 3.1.1. Características

El actual Punto de Venta de Opencode fue desarrollado en 2017, utilizando la tecnología ASP.NET Core [7], un marco de desarrollo web popular para compilar aplicaciones web en la Plataforma .NET.

A la fecha de presentación de esta memoria, este software es utilizado por alrededor de 5 a 10 clientes y provee a sus usuarios con las siguientes características:

- **URL única para cada cliente de Opencode para acceder a la aplicación web:** Cada cliente de Opencode tiene su propio subdominio para acceder al software.
- **Inicio de sesión con perfiles de usuario:** Los trabajadores de las empresas que utilizan el software cuentan con sus propios perfiles para acceder al sistema.
- **Soporte de múltiples empresas por cada cliente:** Cada cliente de Opencode puede operar con más de una empresa. Por ejemplo, un cliente A puede tener dos empresas, B y C, registradas sin necesidad de pagar dos veces el software o tener distintos subdominios.
- **Aperturas y cierres de caja:** Cada usuario puede abrir cajas para comenzar a vender y también cerrarlas para finalizar la jornada y rendir cuentas.
- **Venta con boletas y facturas:** El sistema es compatible con ventas tanto con boletas como con facturas.
- **Emisión de notas de crédito:** El Punto de Venta es capaz de generar notas de crédito para anular o modificar boletas y facturas.
- **Facturación electrónica automática:** Una vez generado cualquiera de los documentos mencionados en los dos puntos anteriores, el sistema automáticamente los sube al Servicio de Impuestos Internos.

- **Informes de caja:** El software tiene la capacidad de consultar los informes de caja emitidos en cualquier fecha pasada.
- **Informes y gráficos de ventas:** El Punto de Venta web permite a sus usuarios ver informes con información relevante acerca de las ventas, productos y usuarios, entre otros.

### 3.1.2. Problemas y Carencias del Software

A continuación se listarán las problemáticas identificadas del Punto de Venta actual de Opencode:

- **Ausencia de multitenant:** Como se mencionó en el primer punto de las características del software, cada cliente tiene su propio subdominio para acceder al sistema. Esto significa que por cada cliente hay que tener una instancia desplegada en los servidores, utilizando así más recursos.
- **Interfaz confusa:** La interfaz del Punto de Venta actual no es clara ni intuitiva, lo que dificulta a los usuarios entender qué hace cada opción presente en la pantalla.
- **Falta de integración de Transbank:** Para vender con tarjetas, el usuario debe ingresar en el Punto de Venta el monto a pagar y después manualmente digitarlo en la máquina de pagos con tarjetas. Esto da pie a la posibilidad de cometer errores y deposita la confianza en que el usuario siempre hará bien su trabajo.
- **Irregularidades en el funcionamiento del sistema:** Aunque la aplicación web funciona, se ha identificado y reportado por parte del dueño de Opencode que el sistema no es regular. Aunque en momentos funciona bien, hay veces que su rendimiento decae, con tiempos de espera largos y funcionalidades que, sin razón aparente, dejan de funcionar, entre otros problemas.
- **Lógica de uso no óptima:** El flujo de trabajo del actual Punto de Venta es confuso. Operaciones como el ingreso del turno en que un cajero operará el sistema deberían ser realizadas automáticamente por la aplicación. Sin embargo, actualmente deben ser ejecutadas manualmente por el usuario, dando pie a la posibilidad de cometer errores y, por tanto, discordancias en los datos.

Cabe mencionar que estas falencias fueron determinadas por el dueño de la empresa previo al inicio de este proyecto. Si bien él tenía claro cuales eran estas problemáticas, en el caso de las irregularidades del funcionamiento y la mala lógica de uso, no estaban establecidos que aspectos eran los que realmente causaban estas situaciones.

## 4. Propuesta de Mejora

En esta sección se describirá y detallará la propuesta de mejora del Punto de Venta de Opencode, indicando las decisiones tomadas para afrontar cada problemática mencionada en el punto anterior y explicando qué beneficios supondrán para el software:

- **Implementación de API con multitenant:** La arquitectura multitenant [6] permite que una sola instancia de software sirva a muchos clientes. En el Punto de Venta actual, para que los clientes tengan acceso al software, se crea una nueva instancia de la aplicación web en los servidores de Opencode y se les asigna un subdominio único para ingresar desde la web. Con la implementación de multitenant, solo será necesaria una instancia de la API y de la aplicación web en ejecución para que todos los clientes de Opencode puedan acceder al software.
- **Sofisticación y limitación de funciones de la UI:** El Punto de Venta actual no distingue roles de usuario, por lo que dispone de funciones irrelevantes y potencialmente peligrosas para ciertos usuarios del sistema. Para solucionar esta problemática, se han identificado dos tipos de usuarios: cajeros y supervisores/administradores. La interfaz solo pondrá a disposición las funciones relevantes para los cajeros, mientras que las funciones avanzadas requerirán autorización.
- **Integración automatizada de Transbank:** Para evitar errores humanos al manipular las máquinas de pago con tarjetas, se creará un software de escritorio que permita conectar los puntos de venta de Transbank con el software web. De esta manera, se eliminará la necesidad de manipular manualmente la máquina, reduciendo el riesgo de errores.
- **Separación de API y aplicación web:** Uno de los problemas del Punto de Venta web actual es que todo el código está en un solo proyecto, ya que ASP.NET Core es un framework full stack. A largo plazo, y debido a la naturaleza de los procesos de desarrollo en la empresa, esto puede hacer que el proyecto crezca con iteraciones poco prolijas, dificultando su mantenimiento. Separar la API de la aplicación web permitirá un desarrollo más limpio y manejable.

- **Flujo de uso simplificado:** Para evitar confusiones y errores de los usuarios, se automatizarán tareas que antes debían realizar manualmente, como indicar la caja que están utilizando, seleccionar el turno en que están trabajando, independizar los turnos de las cajas y hacerlos dependientes de los usuarios. Así, cada usuario deberá iniciar sesión para usar la caja y cerrarla antes de operar en otra, entre otros ejemplos.

Estas propuestas, a excepción de la implementación de multitenant e integración de Transbank, fueron establecidas en colaboración con el dueño de Opencode. Con él, teniendo en cuenta las problemáticas mencionadas en la sección anterior, se realizaron reuniones donde se establecieron los aspectos específicos donde la nueva aplicación podía mejorar. En este proceso se utilizó la experiencia previa del memorista como cajero de los supermercados Jumbo, en conjunto de lo aprendido en la compañía durante los años de funcionamiento del POS actual, para moldear la visión de como debía ser este nuevo software.

## 5. Metodología

Considerando el enfoque particular de Opencode, una empresa pequeña que no sigue estándares formales en el desarrollo de proyectos, la metodología seguida para la implementación del nuevo POS se basó en un enfoque colaborativo y flexible que se adapta a las necesidades y dinámica de la organización. Se establecieron etapas en las que se acordaba un tiempo determinado para desarrollar una serie de funcionalidades que debía incluir el software.

### 1. Definición de Requisitos con el Jefe de la Empresa:

- Iniciar cada etapa del proyecto con una reunión con el jefe de la empresa para discutir y acordar los requisitos y características que se desean implementar en el nuevo POS.

### 2. Desarrollo Iterativo y Presentación de Avances:

- Avanzar en el desarrollo del POS de forma iterativa, presentando avances periódicamente al jefe para su revisión y posibles correcciones.

### 3. Validación con el Jefe para Avanzar a la Siguiente Etapa:

- Obtener la confirmación del jefe para avanzar a la siguiente etapa del desarrollo una vez que los requisitos acordados se hayan implementado y validado.

### 4. Utilización del POS Anterior como Guía:

- Utilizar el POS anterior como referencia para identificar las funcionalidades clave que deben ser incorporadas en el nuevo POS, asegurando la continuidad y mejora de las características existentes.

## 5.1. Beneficios de la Metodología Seguida

- Flexibilidad para adaptarse a los procesos internos de Opencode.
- Colaboración directa con el jefe de la empresa para alinear el desarrollo con las necesidades del negocio.
- Iteraciones frecuentes que permiten ajustes y mejoras continuas.

- Aprovechamiento de la experiencia del POS anterior como guía para el desarrollo del nuevo sistema.

Esta metodología, centrada en la comunicación constante con el jefe de la empresa y en la iteración continua de los avances, fue acordada y establecida en conjunto con el dueño de Opencode al inicio del proyecto, con el objetivo de garantizar un desarrollo efectivo y alineado con las expectativas de la empresa para la implementación exitosa del nuevo POS.

El proceso fue inspirado en un modelo de desarrollo de software secuencial donde cada semana se realizaron reuniones con el jefe de Opencode para establecer los nuevos objetivos y requerimientos a desarrollar, o revisar los avances en caso de no estar todas las metas completas.

## 5.2. Requerimientos

Para la descripción de los requerimientos del software es necesario especificar los tres distintos usuarios con los que cuenta la plataforma. A continuación se describirán brevemente:

- **Cajero:** Es el usuario responsable de realizar las transacciones de venta en el POS. Su principal función es procesar las compras de los clientes de manera eficiente y precisa, asegurando que todas las transacciones sean registradas correctamente en el sistema.
- **Administrador/Supervisor:** El administrador y/o supervisor es quien está encargado de entregar apoyo y autorización a los cajeros para que puedan operar las cajas sin inconvenientes. Si bien un administrador en la jerarquía de un negocio está por encima de un supervisor, en algunos casos esta figura de supervisor no existe y el administrador se encarga de cumplirla, o bien ambos la desempeñan al mismo tiempo.

### 5.2.1. Requerimientos funcionales

Los requerimientos funcionales del nuevo Punto de Venta fueron establecidos mediante el estudio del funcionamiento del software anterior. Se hizo un análisis de sus características, las cuales se utilizaron posteriormente para definir estos nuevos requisitos. Finalmente, estos fueron presentados y posteriormente aprobados por

el jefe de la empresa.

La plataforma permitirá	Cajero	Administrador/Supervisor
Ingresar con el perfil de su empresa y posteriormente su usuario dentro de la empresa	✓	✓
Abrir caja	✓	✓
Cerrar caja	✓	✓
Ingresar productos (escáner y teclado)	✓	✓
Buscar productos	✓	✓
Cambiar precios y hacer descuentos	✓	✓
Realizar ventas con boletas y facturas con cualquier medio de pago	✓	✓
Registrar y editar datos de los clientes	✓	✓
Entregar autorización para realizar ciertas operaciones	✗	✓
Consultar y visualizar documentos emitidos con sus detalles	✓	✓
Emitir notas de crédito y débito	✓	✓
Generar XML's	✓	✓
Subir XML's al SII	✓	✓

### Casos de Uso:

Para representar los requerimientos funcionales y proporcionar una ilustración desde la perspectiva del “actor” o usuario, se puede utilizar un diagrama de casos de uso (UCD). Este diagrama ofrece un resumen del sistema fácil de comprender y permite obtener retroalimentación por parte del cliente o los usuarios finales. En la siguiente figura se puede apreciar el diagrama de casos de uso actualizado para la versión de este software. Cabe mencionar que, en la práctica, dependiendo del negocio, supervisor y administrador pueden realizar la función de cajeros. En este caso se aplica herencia de actor.

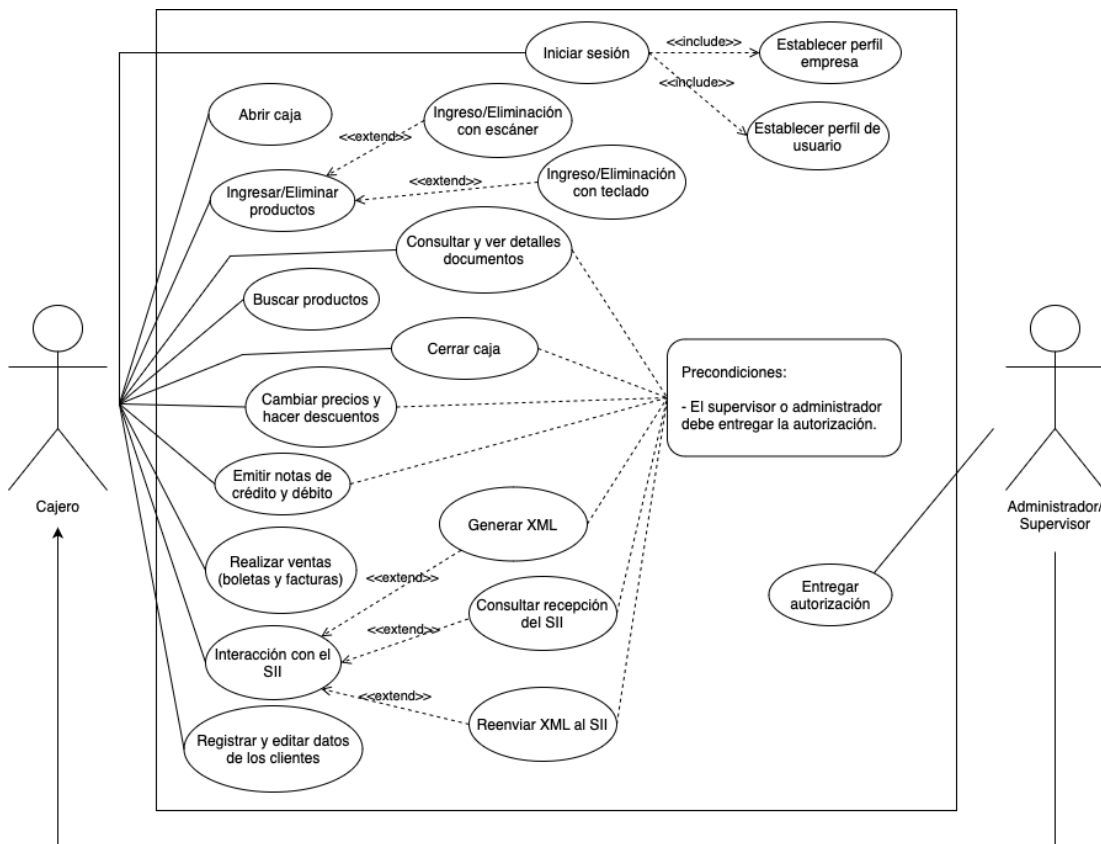


Figura 5.1: Diagrama de casos de uso del nuevo Punto de Venta

### 5.2.2. Requerimientos no funcionales

- **Rendimiento:** El sistema debe ser capaz de soportar la carga de las 5-10 empresas utilizando 1-4 cajas cada una, teniendo un tiempo de respuesta inferior a 3 segundos.
- **Seguridad:** El software debe proteger la integridad y confidencialidad de los datos de cada empresa y de sus usuarios que lo utilicen.
- **Usabilidad:** El sistema debe estar enfocado en la experiencia del usuario, brindando una navegación intuitiva y sencilla.
- **Consistencia:** El sistema debe mantener la integridad de la información de los usuarios.
- **Escalabilidad:** Como se espera que el sistema reciba actualizaciones con el tiempo para mejorar y soportar un aumento en su carga de trabajo, debe ser capaz de adaptarse a estos cambios de manera sencilla.

## 6. Diseño y Arquitectura

Una de las principales características del Punto de Venta es que una aplicación web, es por esto que para el desarrollo de la interfaz de usuario se eligieron frameworks basados en javascript, que es el lenguaje predominante en lo que respecta al desarrollo web. Para el procesamiento de los datos por el lado del backend, se decidió utilizar ASP .NET Core, pues es el framework en el cual Opencode tiene desarrolladas la mayoría de sus APIs y así también se pudo aprovechar funcionalidades ya desarrolladas por ellos útiles para el nuevo software. Más adelante en esta sección se explicarán en detalle las tecnologías utilizadas.

### 6.1. Estructura

Las principales piezas que interactúan en el nuevo Punto de Venta son la aplicación web y la API. El usuario opera en la aplicación web que a su vez hace uso de la API que procesa todos los datos necesarios y generados por el frontend en el navegador. La API por su parte se comunica con una base de datos MySQL que almacena toda esta información y también se conecta con el Servicio de Impuestos Internos o SII por sus siglas, para oficializar todas las ventas generadas por el negocio a través de la web.

A continuación se presenta un diagrama C4 que describe la estructura del sistema:

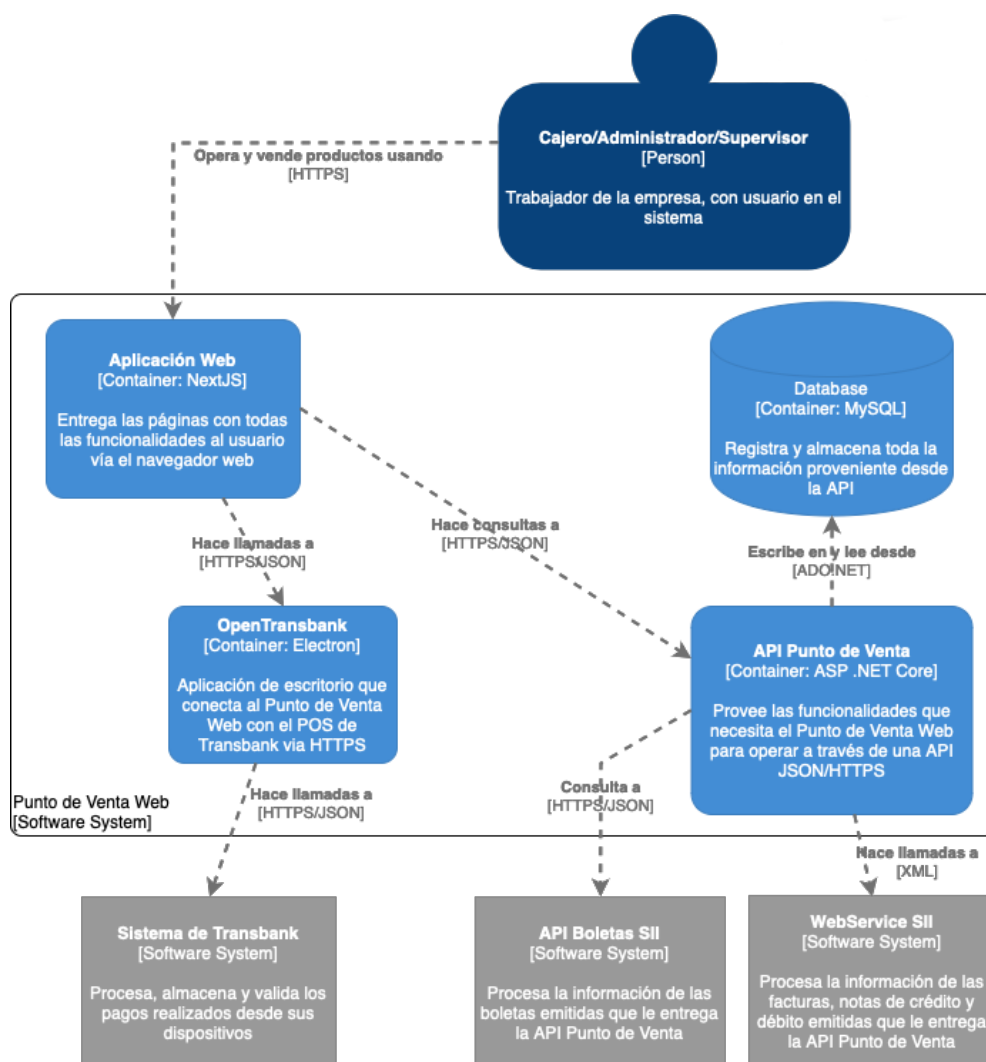


Figura 6.1: Diagrama C4 del nuevo Punto de Venta

## 6.2. Tecnologías Usadas

En este apartado, se listan las principales tecnologías y herramientas utilizadas durante todo el desarrollo del proyecto, con el fin de detallar y explicar brevemente la relevancia de cada una de ellas en este caso concreto:

- **Next.js [9]**: Un framework de React que permite la generación de aplicaciones web con renderizado del lado del servidor y generación de sitios estáticos. El jefe de OpenCode quiere modernizar las tecnologías que utiliza, siendo Next.js es un framework basado React, la biblioteca para interfaces de usuario web y nativas más popular a la fecha, y que además está en constante desarrollo y evolución, añadiendo constantemente nuevas

características e innovaciones.

- **Electron** [4]: Un framework para crear aplicaciones de escritorio utilizando tecnologías web como HTML, CSS y JavaScript. Se eligió Electron debido que al igual que Next.js utiliza JavaScript, entonces así se siguió con la misma línea de programación, y además es multiplataforma haciendo muy fácil el despliegue para los diferentes sistemas operativos que existen.
- **MySQL** [8]: Un sistema de gestión de bases de datos relacional de código abierto, ampliamente utilizado para almacenar y administrar datos. Todas las bases de datos de OpenCode son MySQL y es el principal motivo de la elección de esta tecnología.
- **GitHub** [5]: Una plataforma de alojamiento para el control de versiones y la colaboración en proyectos de software utilizando Git. Se eligió Github por sobre otras alternativas ya que es el sistema más popular en su rubro, sumado a que ya se contaba con conocimiento de como utilizarlo.
- **Tailwind CSS** [3]: Un framework de CSS que proporciona clases utilitarias para construir diseños personalizados de manera rápida y eficiente. La ventaja de Tailwind frente a otras tecnologías es que se parece más a utilizar CSS puro y no entrega diseños ya hechos como ofrecen otras opciones, entregando así más libertad a la hora de diseñar.
- **DBeaver** [2]: Una herramienta universal de administración de bases de datos que ofrece una interfaz gráfica para interactuar con bases de datos SQL y NoSQL. El motivo de la elección de DBeaver se basa en que es software de código abierto y por lo tanto de uso gratuito, a la vez que es uno de los más populares de su categoría.
- **ASP .NET Core**: Un framework de desarrollo de software de Microsoft que permite crear aplicaciones para Windows, web, móvil y más, utilizando varios lenguajes de programación. OpenCode desarrolla todas sus APIs en .NET, por tanto se pudo aprovechar código ya creado en especial para la comunicación con el SII que ahorró horas de trabajo y desarrollo.

## 6.3. Base de Datos

Este proyecto cuenta con dos bases de datos MySQL. La primera base de datos es la que guarda las conexiones a las bases de datos de las empresas respectivas y la segunda es la base de datos de la empresa con la que se ingresó en sí. Cabe destacar que para este proyecto no se diseñó una base de datos nueva, sino que se utilizan las bases de datos ya existentes de cada cliente. Esto se debe a que el Punto de Ventas debe ser compatible con el ERP Web de la Opencode, por lo tanto se debe usar la misma base de datos.

Para mejor entendimiento de cómo es el diseño de la base de datos utilizada, se presentará un diagrama ER con las tablas y atributos presentes en la base de datos. Este diagrama es un resumen de lo que el Punto de Ventas utiliza y no una representación de la base de datos en su totalidad, esto debido a que es de gran tamaño y el resto de tablas y atributos no tienen relevancia en el tópico de este informe.

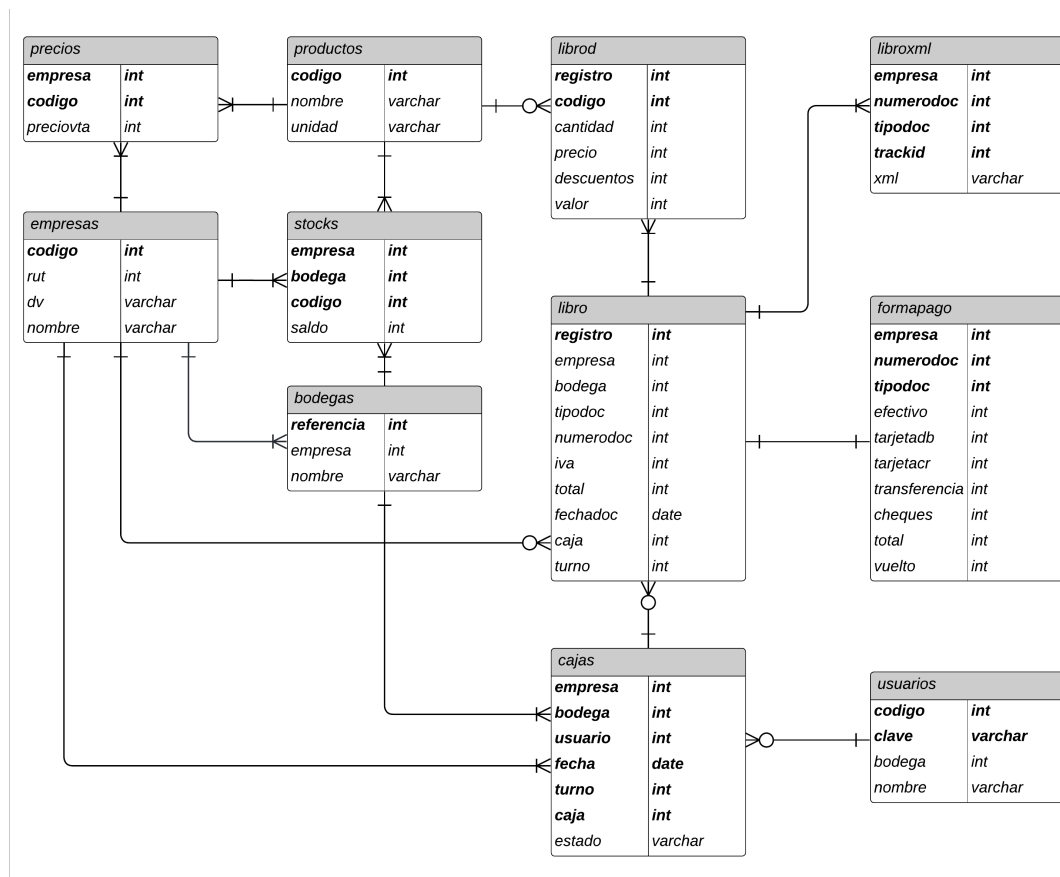


Figura 6.2: Diagrama ER de la BD del Punto de Venta

Es importante mencionar que esta base de datos no cuenta con relaciones. Esto se debe a que, como fue mencionado por miembros de OpenCode, cuando programaban triggers, estos fallaban debido a la presencia de claves foráneas.

A pesar de la ausencia de relaciones explícitas en la base de datos por la ausencia de claves foráneas, como se mencionó en el párrafo anterior, en la práctica las tablas y los datos si están conectados. Así a continuación se describirán estas entidades y sus principales relaciones:

- **empresas:** Entidad en la que se establecen las diferentes empresas que pueden existir con el cliente que contrató el software.
- **bodegas:** Entidad en la que se indican las bodegas que tiene cada empresa.
  - Relaciones:
    - empresa: Refiere a la empresa que pertenece la bodega.
- **usuarios:** Entidad donde se almacenan los datos de los usuarios y sus permisos en el software.
  - Relaciones:
    - bodega: Indica a que bodega pertenece el usuario.
- **cajas:** Entidad donde se guardan los detalles de los turnos de cada usuario en las diferentes cajas.
  - Relaciones:
    - usuario: Refiere al usuario que realizó el turno.
    - bodega: Indica a que bodega pertenece el usuario .
    - empresa: Indica a que empresa pertenece el usuario.
- **productos:** Entidad encargada de identificar cada producto registrado de la empresa.
- **precios:** Entidad encargada de asignar los precios de cada producto.
  - Relaciones:
    - empresa: Indica de que empresa es el precio del producto.
    - codigo: Refiere al producto de la lista de productos.

- **stocks**: Entidad encargada de indicar el stock de cada producto por bodega y empresa.
  - Relaciones:
    - empresa: Indica de que empresa es el stock.
    - bodega: Refiere a la bodega donde se registra el stock del producto.
    - codigo: Relaciona al producto con su stock.
- **libro**: Entidad donde se guardan todos los datos correspondientes a las ventas.
  - Relaciones:
    - empresa: Indica de que empresa fue la venta.
    - bodega: Hace referencia a la bodega donde se toman los productos.
- **librod**: Entidad donde se almacenan los detalles correspondientes a cada producto de cada venta.
  - Relaciones:
    - registro: Indica la venta en el libro a la que corresponde el detalle.
- **libroxml**: Entidad donde se guardan los archivos XML generados por cada venta.
  - Relaciones:
    - empresa: Indica de que empresa fue la venta.
    - tipodoc: Hace referencia al tipo de documento emitido detallado en la tabla libro.
    - numerodoc: Hace referencia al numero de documento en libro.
    - trackid: Indica el número de XML hecho de una venta (el primero, segundo, etc).
- **formapago**: Entidad encargada de almacenar el detalle de como se pago cada venta.
  - Relaciones:

- empresa: Indica de que empresa fue la venta.
- tipodoc: Hace referencia al tipo de documento emitido detallado en la tabla libro.
- numerodoc: Hace referencia al numero de documento en libro.

## 6.4. Nuevo Punto De Venta

En la siguiente sección se especificará los detalles de los cambios realizados en el nuevo Punto de Venta, se describirá el nuevo flujo de trabajo que los usuarios tendrán al utilizar el software, las nuevas funcionalidades y la nueva interfaz, indicando que mejora de la actual aplicación.

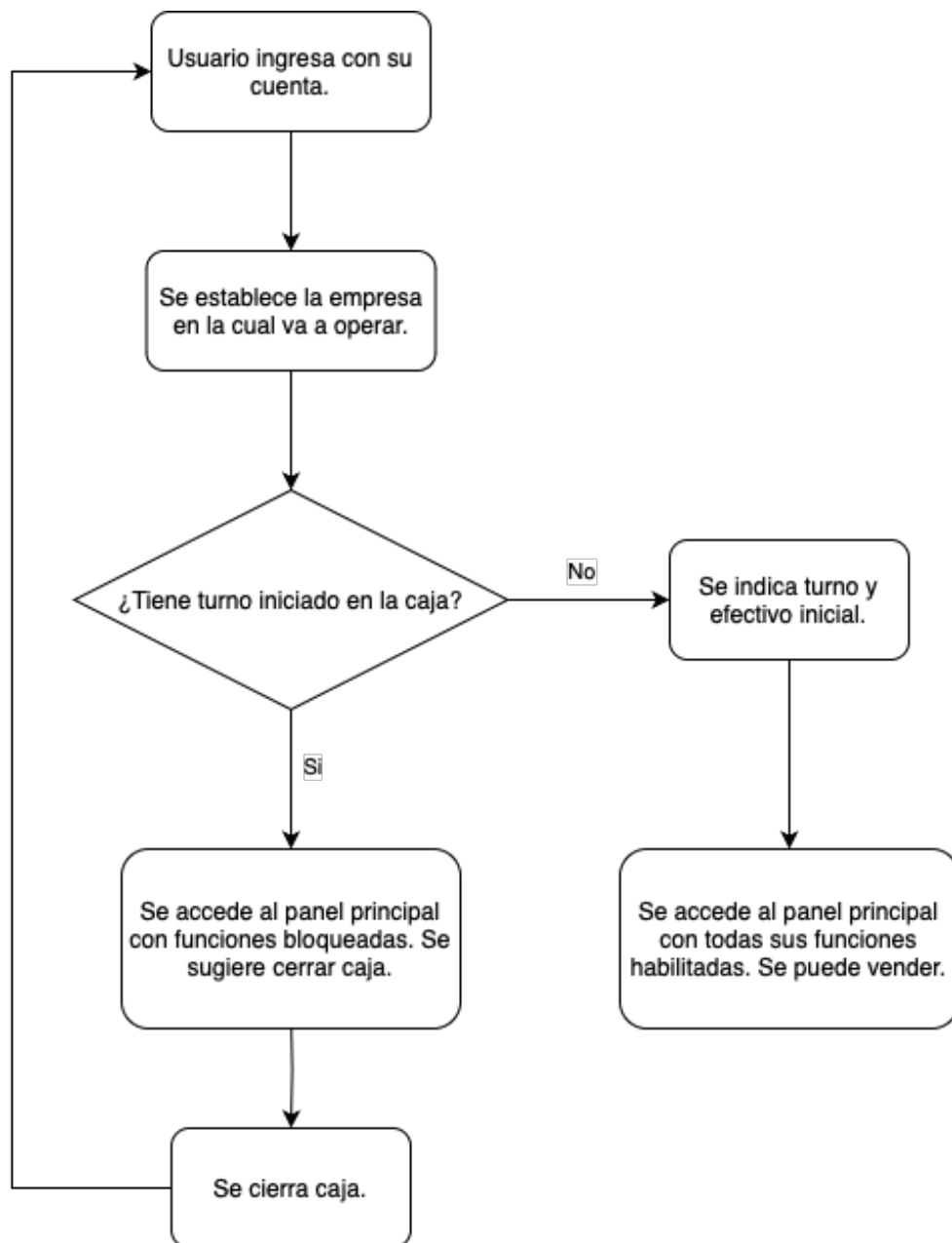
### 6.4.1. Flujo de trabajo y funcionamiento

En el actual Punto de Venta de Opencode algo que aqueja al dueño de la empresa y a los usuarios es su funcionamiento no intuitivo. A continuación se indicará como se cambió el flujo de trabajo desde la perspectiva del usuario final y como esto mejora la experiencia y uso de la aplicación.

### 6.4.2. Inicio de sesión y de turno de trabajo

Para el software actual un inicio de sesión y de turno sigue los siguientes pasos:

1. El usuario inicia sesión con su cuenta dentro de la empresa.
2. Establecer la empresa (dentro de las multiples que cada cliente puede configurar) con la que se quiere operar.
3. En caso de no tener un turno iniciado en la caja, se debe indicar que turno se va a trabajar, en que caja y el efectivo inicial. Si ya había una sesión con caja abierta anteriormente, este paso se salta y se pasa al siguiente.
4. Ahora en la ventana principal ya puedes vender, sin embargo si hay una sesión con caja abierta la plataforma te muestra un modal con un mensaje pidiendo que cierres caja y vuelvas a abrir, pues las fechas de la caja con las del computador no calzan. Si ese es el caso no puedes realizar ventas, pero si utilizar el resto de funcionalidades, por tanto debes cerrar la caja y volver al paso 1.



**Figura 6.3:** Diagrama de flujo del inicio de sesión del Punto de Venta actual

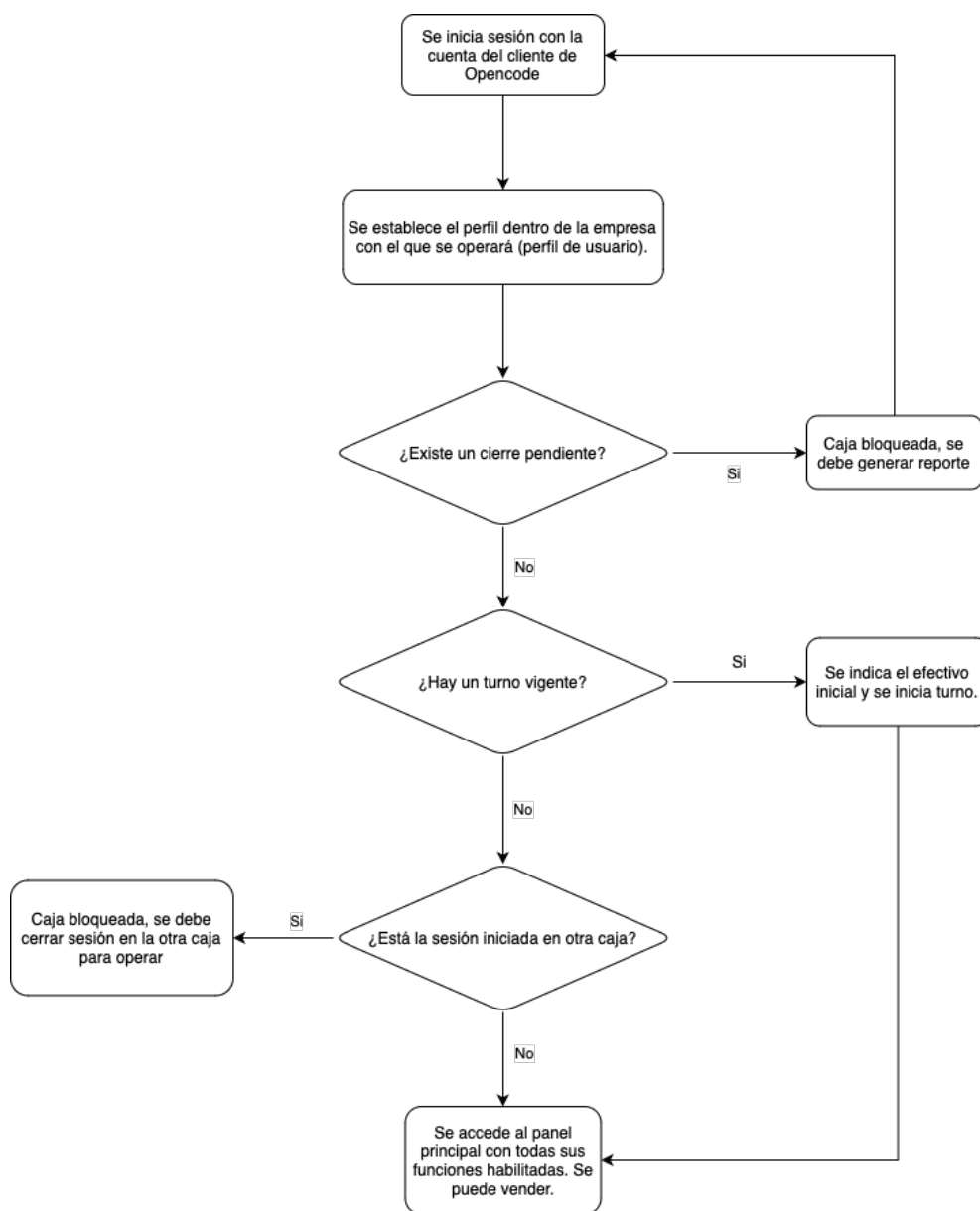
Los principales problemas detectados en el inicio de sesión y de turno son los siguientes:

- **Los turnos se manejan manualmente.** Un usuario puede iniciar con el mismo número de turno, en la misma caja, y en la misma fecha de un turno que ya sucedió causando así una incongruencia en los datos al haber dos turnos iguales con usuarios distintos.

- **La aplicación pide seleccionar número de caja independiente de donde se este operando.** Como se mencionó en el punto anterior, cada vez que se inicia sesión uno puede seleccionar el número de caja en el que se va a operar, por tanto las cajas físicas pueden cambiar de número de acuerdo a como lo indique el usuario, lo que da pie a incongruencias si algún usuario se equivoca en ingresar el número de la caja en la que fue asignado.
- A pesar de tener un turno expirado, **la aplicación permite al usuario seguir usandola con fecha no correspondiente al día real.**
- **El sistema permite tener turnos concurrentes.** Si un turno no ha expirado y el usuario no ha cerrado su caja, este puede iniciar sesión en otra caja permitiendo el sistema atender en dos cajas a la vez.

El nuevo flujo de trabajo, que soluciona las problemáticas mencionadas es el siguiente:

1. Iniciar sesión con la cuenta del cliente.
2. Iniciar sesión con el perfil de usuario dentro de la empresa. El sistema consulta el número de caja correspondiente al equipo donde se inició sesión. Si no logra obtener el dato lo pregunta por pantalla antes de continuar.
3. Una vez ingresado el usuario puede ocurrir lo siguiente:
  - El software automáticamente detecta cual fue el último turno del usuario. Si este tiene una caja sin cerrar, se bloquea la aplicación y se obliga a emitir el reporte de cierre de caja de ese turno.
  - Si no hay cierres pendientes y turnos vigentes, permite continuar al panel principal, preguntando solo por el efectivo inicial e ingresando el turno al sistema.
  - En caso de que haya turno vigente y este la sesión iniciada en otro equipo, bloquea el inicio de sesión sugiriendo desconectarse de la otra caja.
  - Si existe un turno vigente y no hay sesión iniciada en ninguna otra caja, el software continua directo al panel principal.



**Figura 6.4:** Diagrama de flujo del nuevo inicio de sesión del Punto de Venta

De esta forma, el inicio de sesión se simplifica, quitando pasos a realizar al usuario y siendo el sistema mucho más claro al indicar cuales son los pasos a seguir en cada situación posible que se presente.

### 6.4.3. Implementación POS de Transbank

Uno de los hitos a lograr de esta nueva versión del software fue la implementación del POS de Transbank. La importancia de este objetivo radica en que en el software actual, los cajeros deben ingresar manualmente los valores en la

máquina de pago con tarjetas. Esto puede dar pie a errores humanos y por tanto posibles incongruencias en los datos. Por último, cabe mencionar también que la automatización del pago de tarjetas facilita y agiliza el trabajo para el usuario.

Para funcionar, el POS de Transbank requiere 3 cosas: conexión a la corriente eléctrica, conexión a internet vía ethernet y finalmente conexión mediante usb a la máquina donde va a operar. Para poder controlar los dispositivos, Transbank provee de diferentes SDK's en distintos lenguajes de programación a sus clientes, facilitando considerablemente la integración de sus productos a software de terceros.

Dado que el Punto de Venta es una aplicación web, surgió la problemática de cómo controlar el POS de Transbank desde el navegador, ya que por razones de seguridad, desde los navegadores web no se puede acceder directamente al equipo en el que está instalado, impidiendo así el alcance al POS. Para resolver esto, se decidió desarrollar una pequeña aplicación de escritorio que ejecuta una API que controla dicho POS, a la cual el Punto de Venta Web puede conectarse vía HTTP.

Siguiendo con la línea del proyecto, el cual tiene implementado el frontend en Typescript, se decidió utilizar Electron, un framework de Javascript/Typescript para crear aplicaciones de escritorio. La ventaja de Electron es que soporta compilación multiplataforma, permitiendo operar en cualquier sistema operativo, y que genera un ejecutable de muy fácil instalación.

La API de esta aplicación implementa los siguientes dos métodos:

- `generarCobro`: este sólo necesita recibir el monto e inicia el cobro desde el POS de transbank.
- `obtenerNombreEquipo`: este método retorna el nombre del equipo para posteriormente consultar en la base de datos a que número de caja corresponde.

El funcionamiento de la aplicación es muy simple. Con el POS de transbank previamente conectado solo se debe iniciar este software y automáticamente se iniciará la conexión. Además el programa cuenta con un botón para reconectar en caso de que sea necesario e indica también el estado de conexión actual, para así verificar si la máquina de pago con tarjetas está funcionando correctamente.



**Figura 6.5:** Visualización de la aplicación de escritorio de Transbank

#### 6.4.4. Facturación Electrónica SII

Para oficializar las ventas realizadas en el POS, se deben enviar los Documentos Tributarios Electrónicos (DTE) emitidos por dichas operaciones al Servicio de Impuestos Internos (SII) utilizando su sistema de facturación electrónica. La API del Punto de Venta realiza este envío automáticamente después de haber guardado todos los detalles de cada transacción producida en la aplicación.

Cuando se genera una venta, el sistema emite un DTE creando un archivo XML, en el cual se describen los detalles de la operación utilizando las etiquetas establecidas en la documentación proporcionada por el SII.

Las empresas que deseen utilizar la facturación electrónica deben solicitar un Código de Asignación de Folios (CAF) para cada tipo de documento que deseen emitir. Por ejemplo, si un contribuyente solicita 1000 folios para Boletas Electrónicas y se le asignan los números del 501 al 1500, el SII le entregará:

- Un CAF que debe utilizarse exclusivamente para las Boletas Electrónicas, dentro del rango de 501 a 1500.
- Una clave privada que se usará para timbrar electrónicamente las boletas dentro del rango de 501 a 1500, a medida que se vayan emitiendo.

Este CAF debe concatenarse al final del XML generado previamente.

Por último, el XML del DTE debe ser firmado digitalmente utilizando la Firma

Electrónica Avanzada (FEA), que garantiza la autenticidad del documento y la integridad de la información. La FEA es un certificado digital que se obtiene de un prestador acreditado por el SII. Esta firma se ubica al final del archivo, después del CAF.

Con todo esto realizado, se pueden enviar los DTE al Servicio de Impuestos Internos utilizando su API para las Boletas Electrónicas y su Web Service para todos los otros tipos de documentos (Facturas, Notas de Crédito y Notas de Débito Electrónicas).

Una vez enviado el DTE, el SII retorna un identificador único llamado trackid, confirmando la recepción exitosa del documento. Luego, para verificar el estado de validación del DTE emitido, es necesario realizar una nueva consulta al servicio, donde ellos indicarán si el documento fue aceptado, rechazado u objetado.

A continuación se presenta un fragmento de ejemplo de un archivo XML que representa un DTE:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<EnvioDTE xmlns="http://www.sii.cl/SiiDte"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.sii.cl/SiiDte
  EnvioDTE_v10.xsd" version="1.0">
<SetDTE ID="SetDoc">
  <Caratula version="1.0">
    <RutEmisor>97975000-5</RutEmisor>
    <RutEnvia>7880442-4</RutEnvia>
    <RutReceptor>60803000-K</RutReceptor>
    <FchResol>2003-09-02</FchResol>
    <NroResol>0</NroResol>
    <TmstFirmaEnv>2003-10-13T09:33:22</TmstFirmaEnv>
    <SubTotDTE>
      <TpoDTE>33</TpoDTE>
      <NroDTE>1</NroDTE>
    </SubTotDTE>
  </Caratula>
<DTE version="1.0">
  <Documento ID="F60T33">
    <Encabezado>
      <IdDoc>
        <TipoDTE>33</TipoDTE>
        <Folio>60</Folio>
```

```
<FchEmis>2003-10-13</FchEmis>
</IdDoc>
<Emisor>
  <RUTEmisor>97975000-5</RUTEmisor>
  <RznSoc>RUT DE PRUEBA</RznSoc>
  <GiroEmis>Insumos de Computacion</GiroEmis>
  <Acteco>31341</Acteco>
  <CdgSIISucur>1234</CdgSIISucur>
  <DirOrigen>Teatinos 120, Piso 4</DirOrigen>
  <CmnaOrigen>Santiago</CmnaOrigen>
  <CiudadOrigen>Santiago</CiudadOrigen>
</Emisor>
```

**Listing 1:** Fragmento de un archivo XML

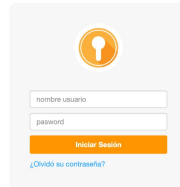
#### 6.4.5. Simplificación de la interfaz

Otro punto importante en los objetivos a lograr en el nuevo Punto de Venta de Opencode fue la renovación de interfaz. Se planteó hacer la interfaz más simple para facilitar el uso de la aplicación y mejorar la experiencia de los usuarios.

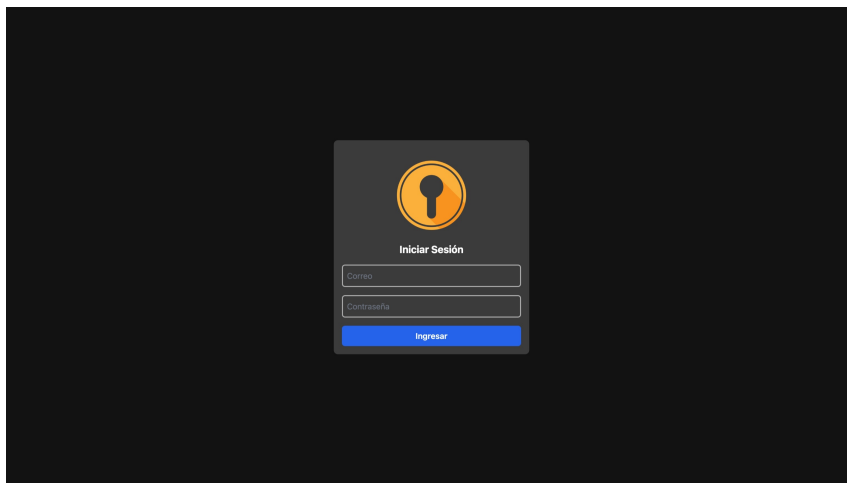
El dueño de Opencode tiene una preferencia por los temas oscuros, por tanto se optó por desarrollar la interfaz de usuario con ese estilo.

A continuación se presentarán las diferentes secciones del Punto de Venta actual y serán comparadas con la nueva versión, comentando las diferencias y las mejoras percibidas.

## 1. Inicio de sesión



(a) Antes



(b) Después

**Figura 6.6:** Comparación inicios de sesión

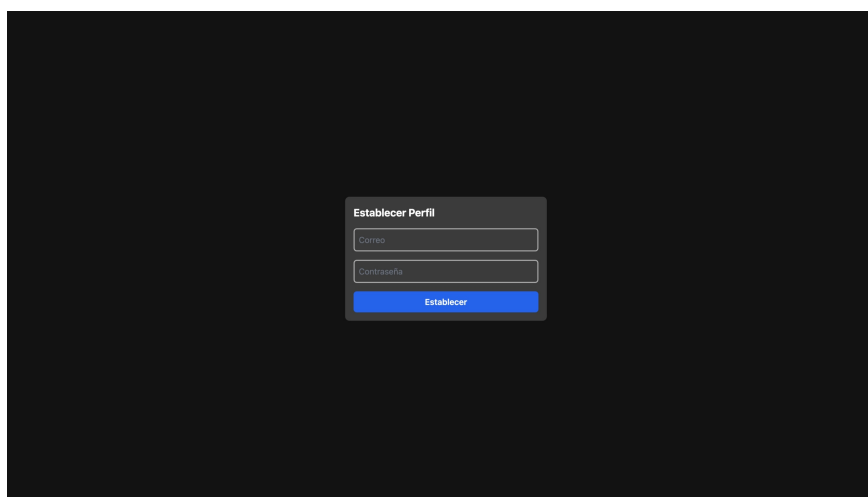
El inicio de sesión es la primera ventana que se les presenta al usuario a la hora de acceder a la aplicación por primera vez. Las principales diferencias visuales a destacar aquí son el cambio a tema oscuro y la pulcritud, pues en el Punto de Venta actual el modal de inicio de sesión no está centrado en la pantalla.

Funcionalmente estos inicios de sesión son diferentes, ya que en la versión actual se ingresa con la cuenta de usuario dentro de la empresa, mientras que en la versión nueva se está iniciando sesión con la cuenta de la empresa o cliente registrados en los sistemas de Opencode.

## 2. Establecer empresa



(a) Antes



(b) Después

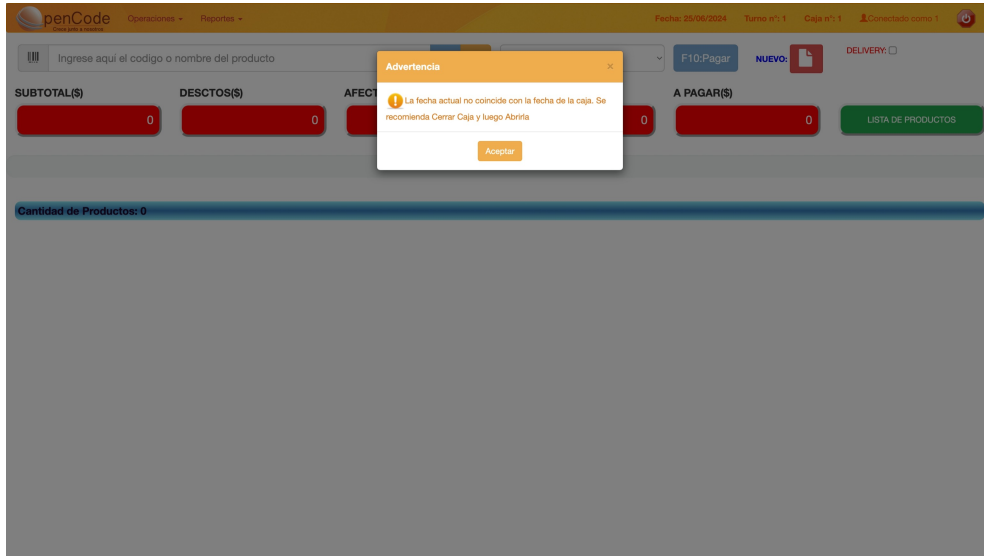
**Figura 6.7:** Antes y después ventana Establecer Empresa

Esta es la ventana que aparece después del inicio de sesión. Hay que recordar que OpenCode permite a sus clientes operar con diferentes empresas dentro del mismo software, sin necesidad de comprarlo dos veces. En el punto de venta actual se debe establecer la empresa para posteriormente empezar a trabajar con normalidad. En cambio, en la nueva versión, el usuario solo debe ingresar con su cuenta de usuario dentro de su empresa.

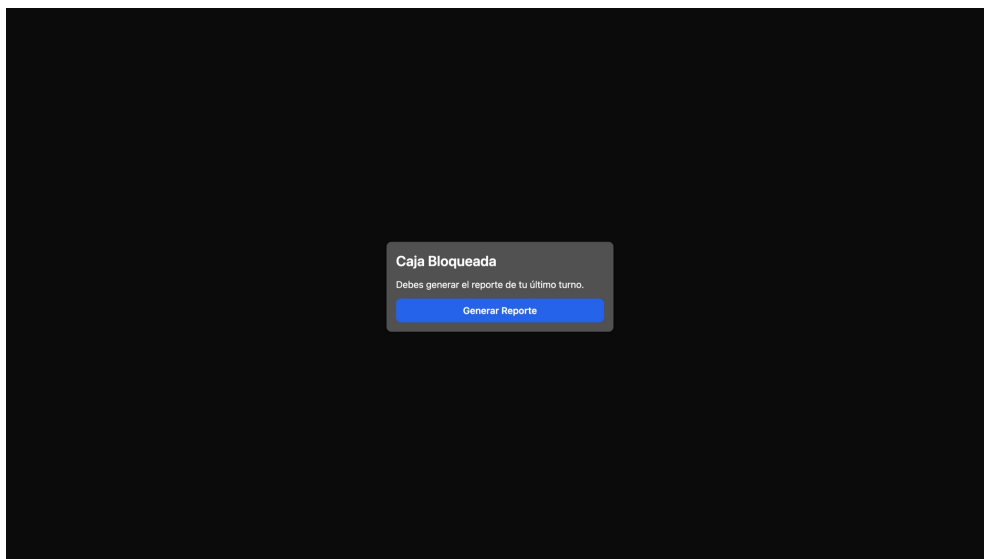
De cierta manera se puede decir, que se invirtió el orden de las dos ventanas mencionadas hasta el momento, y que ahora no se debe seleccionar la empresa

con click de ratón sino que solo basta con utilizar el perfil de cada una.

### 3. Caja bloqueada



(a) Antes



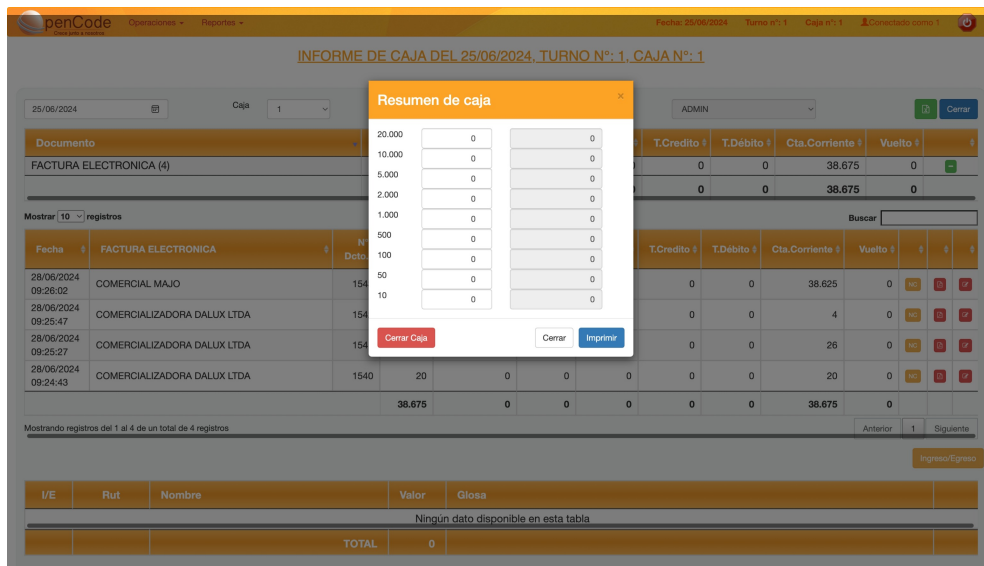
(b) Después

**Figura 6.8:** Diferencias en el bloqueo de las cajas

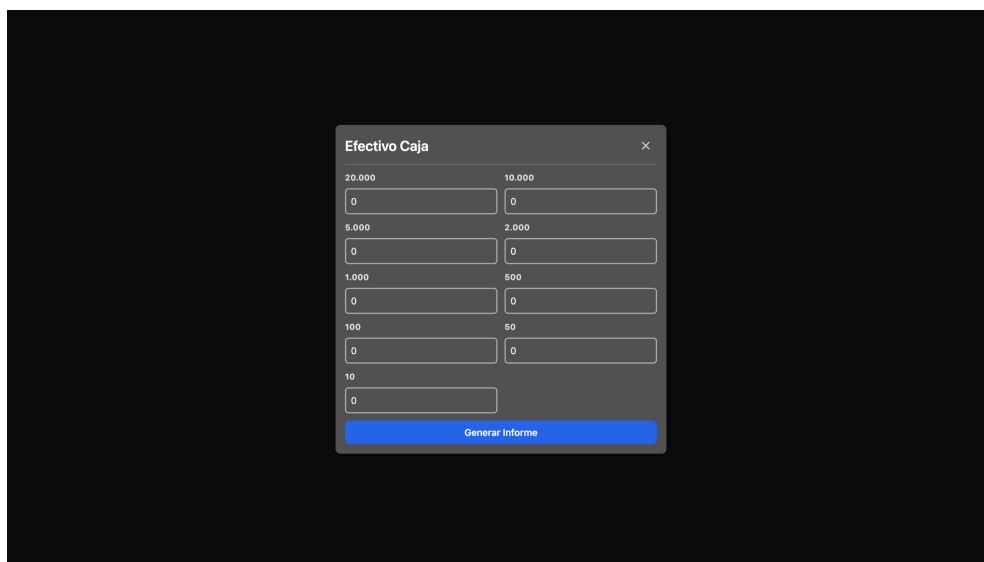
Si el usuario ingresa con sin haber rendido su turno algún día anterior, el sistema en ambos casos bloquea su uso. Las diferencias recaen en que el Punto de Venta actual solo muestra un modal y bloquea las ventas, permitiendo utilizar todo el resto del software, mientras que el nuevo POS no permite

realizar ninguna operación, obligando al usuario a rendir su caja.

#### 4. Cierre de caja



(a) Antes



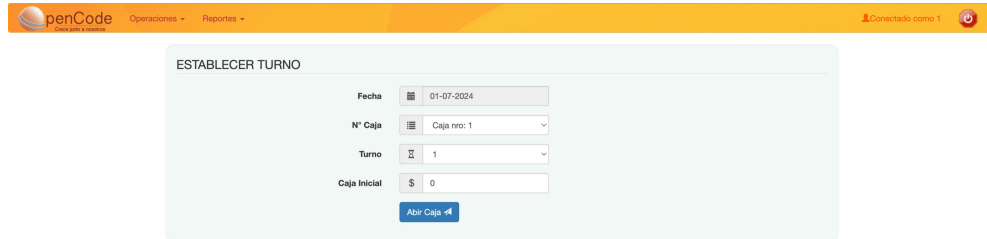
(b) Después

**Figura 6.9:** Modales de cierre de cajas

Para los cierres de cajas, ambas aplicaciones funcionan igual. Se muestra un modal con las denominaciones de billetes y monedas que existen en Chile y se digita la cantidad de cada uno al momento del cierre. La diferencia recae en

que el software actual da la opción de cerrar y de imprimir, cuando el nuevo Punto de Venta solo permite cerrar e imprime el reporte automáticamente.

## 5. Inicio de turno

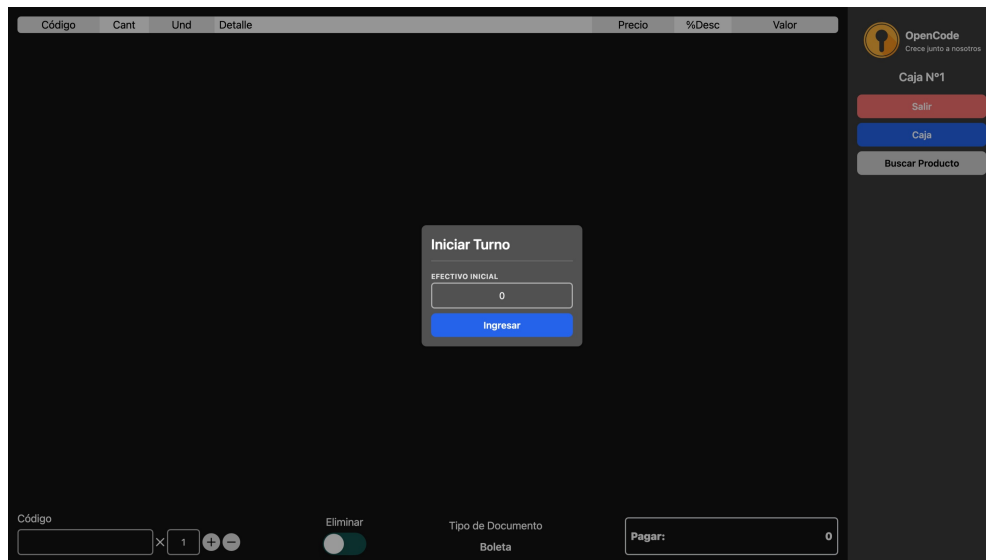


The screenshot shows a web interface for setting a shift. At the top, there is a navigation bar with the 'penCode' logo, 'Operaciones' and 'Reportes' menus, and a user connection status 'Conectado como 1'. The main content area is titled 'ESTABLECER TURNO' and contains the following fields:

- Fecha:** A date picker set to '01-07-2024'.
- N° Caja:** A dropdown menu set to 'Caja rnc: 1'.
- Turno:** A dropdown menu set to '1'.
- Caja Inicial:** A text input field with '\$ 0'.

Below these fields is a blue button labeled 'Abrir Caja'.

(a) Antes



The screenshot shows the main POS interface with a modal dialog titled 'Iniciar Turno'. The dialog has a field for 'EFECTIVO INICIAL' (Initial Cash) with the value '0' and a blue 'Ingresar' (Enter) button. The background interface includes a table header with columns: 'Código', 'Cant', 'Und', 'Detalle', 'Precio', '%Desc', and 'Valor'. On the right side, there is a sidebar with the 'OpenCode' logo, 'Caja N°1', and buttons for 'Salir', 'Caja', and 'Buscar Producto'. At the bottom, there is a search bar with 'Código', a quantity input '1', an 'Eliminar' toggle, a 'Tipo de Documento' dropdown set to 'Boleta', and a 'Pegar:' field with '0'.

(b) Después

**Figura 6.10:** Comparativa ventana de inicio de turno

Al iniciar un turno el Punto de Venta actual muestra un formulario con la fecha de apertura (que no se puede editar), un selector para elegir el turno en que se va a trabajar y una entrada de texto para ingresar el efectivo inicial con el que se atenderá.

Ahora el usuario solo ingresará el efectivo inicial. La fecha de apertura se da por hecho que es la fecha actual en el momento y el sistema detecta automáticamente el número de la caja y el turno en que se trabajará.

## 6. Panel principal

Cod	Producto	Cantidad	Und	(\$)Precio	Desc.	(\$)Valor
12	CONECTOR DE CADENA (NEGRO)	1	SET	107,10	0	107
32	CADENA CORTINA VERTICAL 89MM	1	METRO	202,30	0	202
1	TOPE CADENA (GRIS)	1	UN	141.013,81	0	141.014
43	CADENA SIN-FIN x3MTx4.5*6MM (GRIS)	1	UNIDAD	406,98	0	407
54	TAPA DUO GENEFA STANDARD REDONDA NEGRA 259-961	1	SET	697,34	0	697
63	EMBRAGUE DUO CORDON 38 371	1	UNIDAD	1.361,36	0	1.361
12	CONECTOR DE CADENA (NEGRO)	1	SET	107,10	0	107
33	CORDON CORTINA VERTICAL BLANCO	1	METRO	202,30	0	202

(a) Antes

Código	Cant	Und	Detalle	Precio	%Desc	Valor
123	1	UND	Tubo Conduit Libre Halogenos 16mm x 3mts 1250nw	2.073	0.0	2.073
21	1	UND	Tb PVC Cond. 32mm x 6mts C-III Cem	42.572	0.0	42.572
12	1	UND	Tb PVC Cond. 90mm x 6mts C-I Cem	7.235	0.0	7.235
22	1	UND	Tb PVC Cond. 40mm x 6mts C-III Cem	5.260	0.0	5.260
44	1	UND	Tb Sch.40 Naran 6 x 6mts c/Camp	57.701	0.0	57.701
121	1	UND	Lubricante Natural p/PVC 500cc	1.457	0.0	1.457
11	1	UND	Tb PVC Cond. 75mm x 6mts C-I Cem	6.289	0.0	6.289

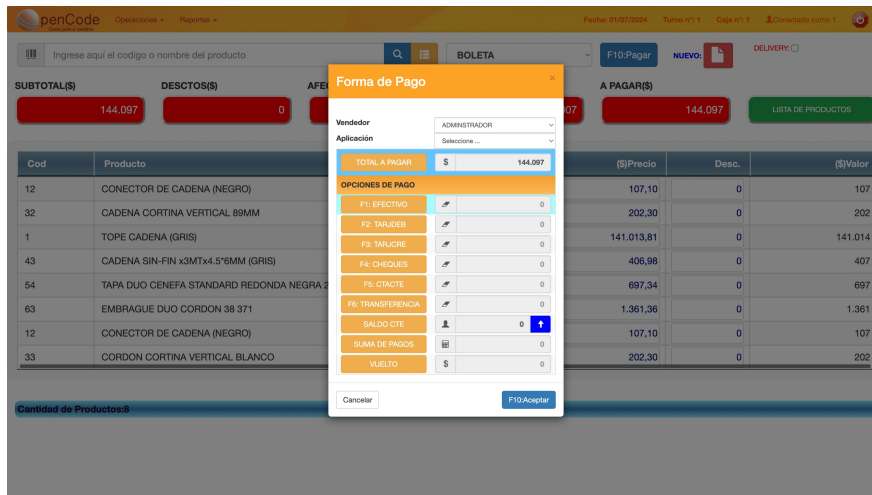
(b) Después

Figura 6.11: Comparativa de los paneles principales

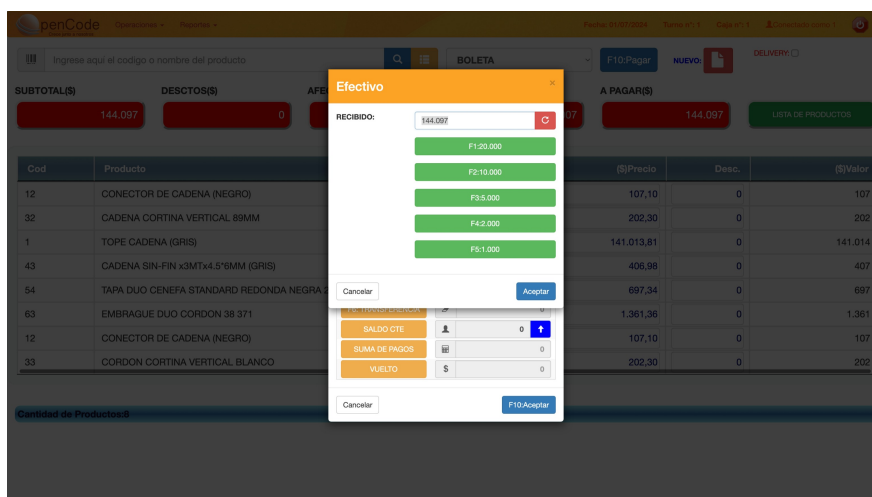
Continuando con la tónica, en el panel principal el cambio principal realizado fue la eliminación de funcionalidades disponibles para el usuario. Se suprimió la

barra de navegación superior y ahora solo se presenta el detalle de la venta y una barra lateral con la opción de desconectarse, ver el panel del administrador, y la opción de buscar productos. Esta barra lateral cuenta con espacio disponible para que en el futuro se pueda acceder a nuevas funcionalidades desde ella.

## 7. Modal medios de pago



(a) Modal de medios de pago



(b) Modal detalle del medio de pago

**Figura 6.12:** Métodos de pago Punto de Venta actual

Los operadores en el Punto de Venta actual, para ingresar un medio de pago, deben seleccionar el tipo de pago (efectivo, tarjeta, etc.) y después en otro modal deben ingresar el valor para este.

Código	Cant	Und	Detalle	Precio	%Desc	Valor
12	1	UND	Tb PVC Cond. 90mm x 6mts C-I Cem	7235	0.0	7235
12	1	UND	Tb PVC Cond. 90mm x 6mts C-I Cem	7235	0.0	7235
43	1	UND	Tb Sch.40 Naran 4 x 6mts c/Camp	27522	0.0	27522
32	1	UND	Tb PVC Cond. 75mm x 6mts C-III Gom	14068	0.0	14068
12	1	UND	Tb PVC Cond. 90mm x 6mts C-I Cem	7235	0.0	7235
32	1	UND	Tb PVC Cond. 75mm x 6mts C-III G	4068	0.0	14068

**Monto a Pagar: 77.363**

EFECTIVO	VUELTO
<input type="text" value="77363"/>	<input type="text" value="0"/>
TARJETA CRÉDITO	
<input type="text" value="0"/>	
TARJETA DÉBITO	
<input type="text" value="0"/>	
TRANSFERENCIA	
<input type="text" value="0"/>	
CHQUES	
<input type="text" value="0"/>	
CUENTA CORRIENTE	
<input type="text" value="0"/>	
FALTANTE	
<input type="text" value="0"/>	

Código:  x 1 Eliminar Tipo de Documento:  Pagar:

**Figura 6.13:** Modal métodos de pago nuevo Punto de Venta

En la nueva versión esto se ha simplificado, ahora solo hay un modal donde el usuario ingresa todos los valores correspondientes a cada forma de pago. El sistema le indica cuanto es el vuelto y lo que le falta si la suma de los totales son menores al total de la venta.

## 8. Panel administrador

**INFORME DE CAJA DEL 25/06/2024, TURNO N°: 1, CAJA N°: 1**

Documento	Total	Transferencia	Efectivo	Cheques	T.Credito	T.Débito	Cta. Corriente	Vuelto
FACTURA ELECTRONICA (4)	38.675	0	0	0	0	0	38.675	0
<b>TOTAL</b>	<b>38.675</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>38.675</b>	<b>0</b>

Fecha	FACTURA ELECTRONICA	N° Dcto.	Total	Transferencia	Efectivo	Cheques	T.Credito	T.Débito	Cta. Corriente	Vuelto
28/06/2024 09:25:02	COMERCIAL MAJO	1543	38.625	0	0	0	0	0	38.625	0
28/06/2024 09:25:47	COMERCIALIZADORA DALUX LTDA	1542	4	0	0	0	0	0	4	0
28/06/2024 09:25:27	COMERCIALIZADORA DALUX LTDA	1541	26	0	0	0	0	0	26	0
28/06/2024 09:24:43	COMERCIALIZADORA DALUX LTDA	1540	20	0	0	0	0	0	20	0
<b>TOTAL</b>			<b>38.675</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>38.675</b>	<b>0</b>

(a) Antes

**Detalle diario**

Selección Parámetros: Fecha: 07-06-2024, N° Caja: 0, Turno: 0

DOCUMENTO	CANT.	TOTAL	TRANSFERENCIA	CHEQUES	EFFECTIVO	T. CRÉDITO	T. DÉBITO	CTA. CTE.	VUELTO
FACTURAS ELECTRONICAS	2	56.775	0	0	47.020	0	9.758	0	0
<b>TOTALES:</b>	<b>2</b>	<b>56.775</b>	<b>0</b>	<b>0</b>	<b>47.020</b>	<b>0</b>	<b>9.758</b>	<b>0</b>	<b>0</b>

DOCUMENTO	N° DOC	TOTAL	TRANSFERENCIA	CHEQUES	EFFECTIVO	T. CRÉDITO	T. DÉBITO	CTA. CTE.	VUELTO
FACTURA	88	47.017	0	0	47.020	0	0	0	0
FACTURA	87	9.758	0	0	0	0	9.758	0	0

(b) Después

**Figura 6.14:** Comparativa de los paneles de administrador

Para el panel del administrador se mantuvo la misma esencia del software actual. La interfaz fue simplificada un poco y las operaciones disponibles para los documentos no se presentan en la misma columna de la tabla, ahora están en un modal que aparece al seleccionar el documento.

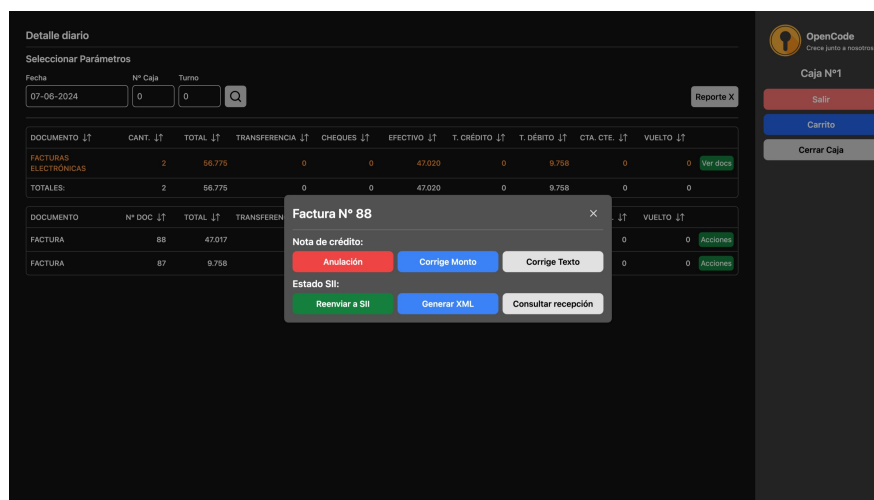


Figura 6.15: Modal operaciones de documentos POS nuevo

Este modal trae nuevas funciones relacionadas al servicio de impuestos internos, las cuales son

- Reenviar a SII: Reenvia XML del Documento Tributario Electrónico en caso que este haya sido rechazado.
- Generar XML: Genera un nuevo XML del DTE.
- Consultar recepción: Consulta si el servicio recibió el XML del DTE conforme y que está pendiente de aprobación.

## 7. Pruebas y Resultados

### 7.1. Medición de usabilidad con SUS

Se midió la usabilidad del nuevo software usando SUS (System Usability Scale) [1]. El estudio se realizó con Anselec, empresa que utiliza el Punto de Venta de Opencode. Se les dispuso un formulario digital para que todos sus trabajadores pudieran contestar los enunciados.

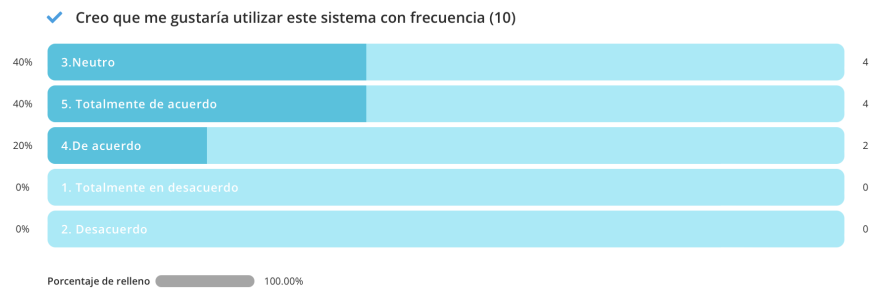
El SUS consta de 10 enunciados predeterminados, a los cuales se les debe responder según la escala de Likert. Los enunciados a responder son los siguientes:

1. Creo que me gustaría utilizar este sistema con frecuencia
2. Encontré el sistema innecesariamente complejo
3. Pensé que el sistema era fácil de usar
4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema
5. Encontré que las diversas funciones de este sistema estaban bien integradas
6. Pensé que había demasiada inconsistencia en este sistema
7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente
8. Encontré el sistema muy complicado de usar
9. Me sentí muy seguro usando el sistema
10. Necesitaba aprender muchas cosas antes de empezar con este sistema

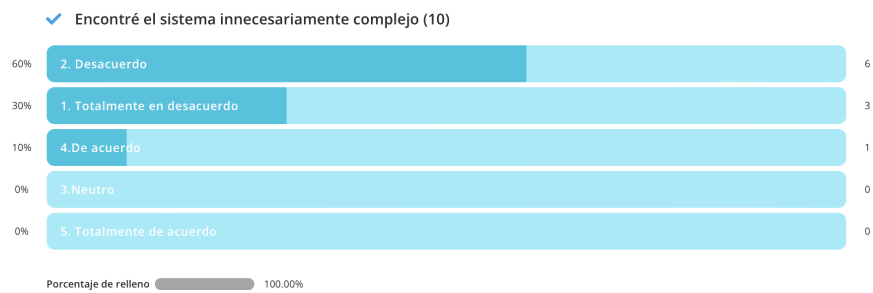
Donde cada pregunta se debe calificar entre el 1 y el 5, siendo 1 “totalmente en desacuerdo” y 5 “totalmente de acuerdo”.

El estudio fue realizado por 10 trabajadores de la empresa ya mencionada, donde se incluyó solamente a aquellos colaboradores dedicados a trabajar con el punto de venta de Opencode.

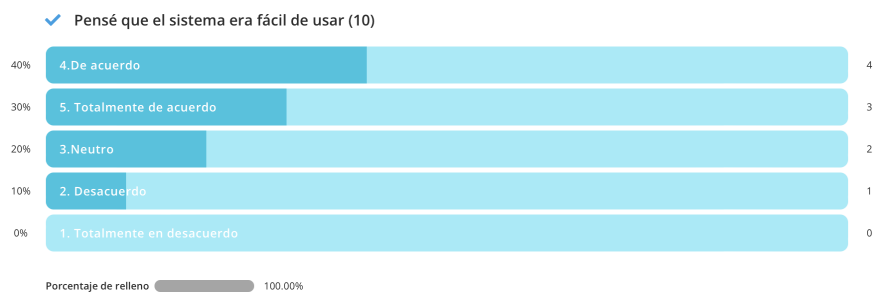
A continuación, se despliegan los resultados obtenidos:



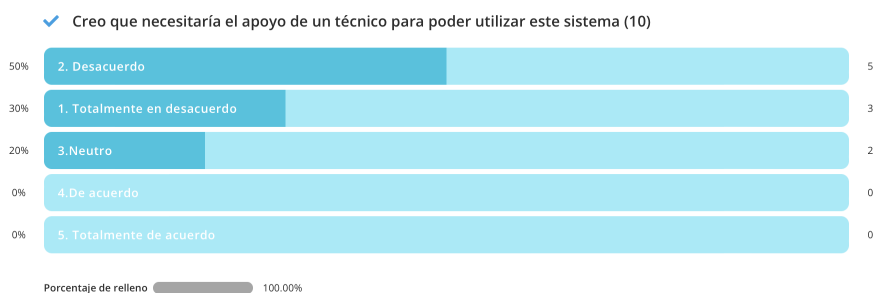
(a) Pregunta 1



(b) Pregunta 2

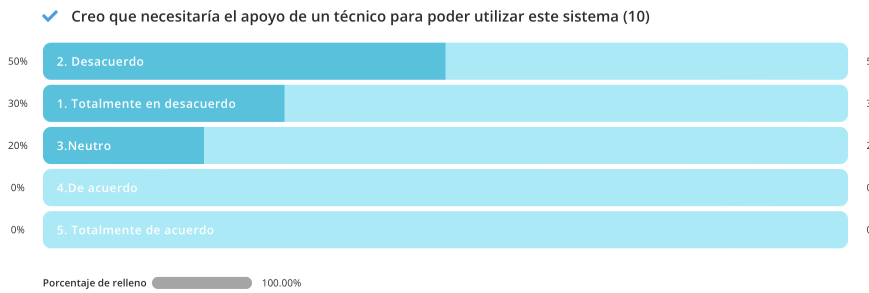


(c) Pregunta 3

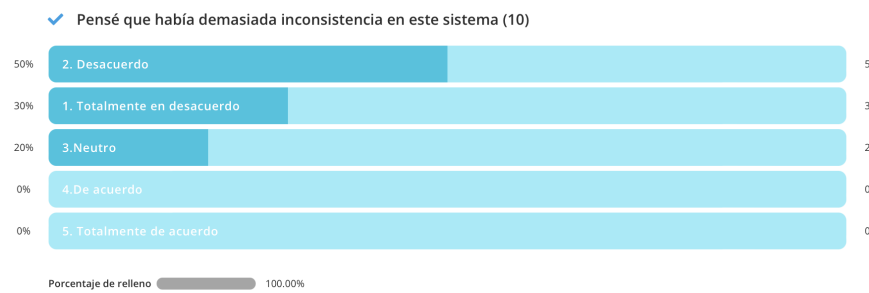


(d) Pregunta 4

Figura 7.1: Resultados encuesta SUS



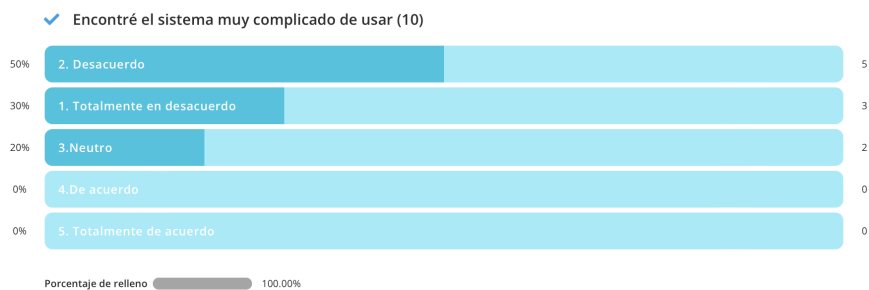
(e) Pregunta 5



(f) Pregunta 6

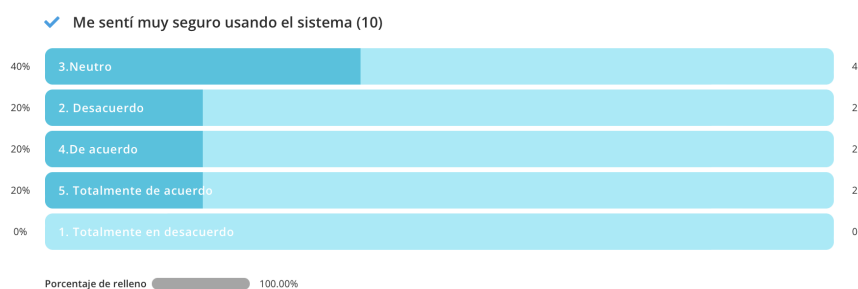


(g) Pregunta 7

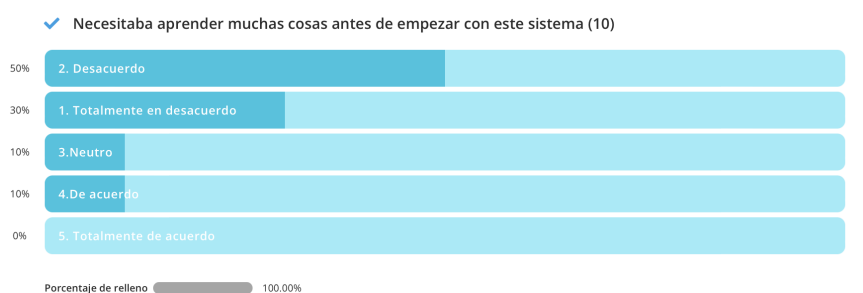


(h) Pregunta 8

Figura 7.1: Resultados encuesta SUS (cont.)



(i) Pregunta 9



(j) Pregunta 10

**Figura 7.1:** Resultados encuesta SUS (cont.)

Con estas respuestas se procede a realizar el cálculo para obtener la usabilidad del sistema:

**Suma de respuestas de enunciados impares, restado 5=**  
 $[(12+20+8+16+15+6+2+15+12+10+35+6+4+12+4+8+10)/10]-5= 14.5$

**Suma de respuestas de enunciados pares, restado a 25=**  
 $25-[(12+3+4+10+3+6+10+3+6+10+3+6+10+3+3+4)/10]= 17.9$

**Suma de ambos resultados, multiplicado por 2.5=**  
 $(14.5+17.9) \times 2.5= 81$

Este resultado no es un porcentaje, es el puntaje obtenido y se posiciona en una escala entre 1-100, donde el promedio es 68 y un resultado debajo de este, indica que hay varios aspectos a corregir, de esta manera se obtiene la usabilidad del sistema. Viéndose de la siguiente manera:



- Usuarios: Controlador que contiene los metodos relacionados a la gestión de los usuarios de la empresa
  - Login:

```
Status: 200 OK   Size: 652 Bytes   Time: 1.56 s

Response   Headers 5   Cookies   Results
1   {
2   "codigo": 1,
```

Figura 7.4: Resultados Login controlador Usuarios

- IniciarTurno:

```
Status: 200 OK   Size: 1 Bytes   Time: 1.47 s

Response   Headers 5   Cookies   Results
1   1
```

Figura 7.5: Resultados IniciarTurno

- VerEstadoTurno:

```
Status: 200 OK   Size: 18 Bytes   Time: 497 ms
Response        Headers5    Cookies    Results
1  "Cierre pendiente"
```

Figura 7.6: Resultados VerEstadoTurno

- ValidarSupervisor:

```
Status: 200 OK   Size: 45 Bytes   Time: 418 ms
Response        Headers5    Cookies    Results    Docs
1  {
2    "abrircajas": 1,
3    "autdescto": 1,
4    "cmbprecios": 1
5  }
```

Figura 7.7: Resultados ValidarSupervisor

- PtoVta: Controlador con los métodos necesarios para realizar ventas.
  - GuardarVenta:

```
Status: 200 OK   Size: 733.81 KB   Time: 6.12 s

Response   Headers 5   Cookies   Results
-----
1   {
2     "ok": "ok",
3     "numero": 60,
4     "fecha": "3 de JULIO de 2024",
5     "timbre": "Qk0+mQgAAAAAAD4AAAAoAAAA4AgAAMA
6   }
```

Figura 7.8: Resultados GuardarVenta

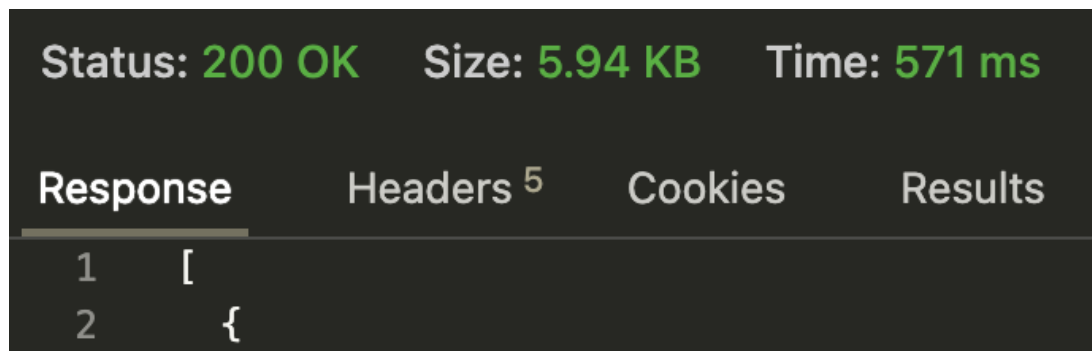
- ObtenerDocumentosCaja:

```
Status: 200 OK   Size: 1.36 KB   Time: 838 ms

Response   Headers 5   Cookies   Results
-----
1   {
2     "resumenes": [
```

Figura 7.9: Resultados ObtenerDocumentosCaja

- Productos: Controlador con los métodos que permiten obtener los productos a la venta.
  - BuscarProducto:



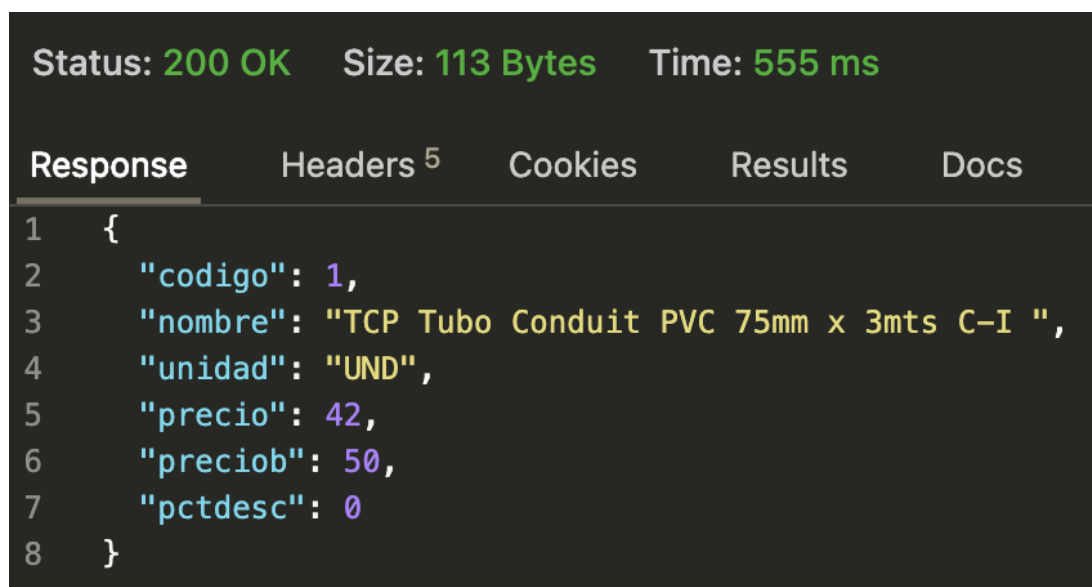
```
Status: 200 OK   Size: 5.94 KB   Time: 571 ms

Response   Headers 5   Cookies   Results

1   [
2   {
```

Figura 7.10: Resultados BuscarProducto

- Código:



```
Status: 200 OK   Size: 113 Bytes   Time: 555 ms

Response   Headers 5   Cookies   Results   Docs

1   {
2   "codigo": 1,
3   "nombre": "TCP Tubo Conduit PVC 75mm x 3mts C-I ",
4   "unidad": "UND",
5   "precio": 42,
6   "preciob": 50,
7   "pctdesc": 0
8   }
```

Figura 7.11: Resultados Código

- Clientes: Controlador que contiene los métodos para procesar la información relacionada a los clientes.
  - BuscarCliente:

```

Status: 200 OK   Size: 2.41 KB   Time: 581 ms

Response   Headers5   Cookies   Results
-----
1   [
2   {
3   "rut": 76760612,

```

Figura 7.12: Resultados BuscarCliente

- FacturacionElectronica: Controlador que contiene los métodos relacionados al servicio de impuestos internos y la facturación en línea. No se pudieron hacer pruebas con este controlador debido a que la API de certificación y pruebas del SII estuvo caída al momento de la realización estas pruebas.

## 7.3. Métricas de software

Para respaldar la mejora del nuevo Punto de Venta en comparación con la versión anterior, se realizaron mediciones de la calidad del código de ambas aplicaciones. En concreto, se llevaron a cabo dos pruebas: la primera utilizando Visual Studio y la segunda utilizando SonarQube.

### 7.3.1. Análisis con Visual Studio

Para este análisis, dado que Visual Studio es un IDE enfocado en el trabajo con la plataforma .NET, se comparó el proyecto original del Punto de Venta, que está completamente hecho en este framework, y la API del nuevo sistema, que también está desarrollada en esta plataforma. Los resultados son los siguientes:

PROYECTO	ÍNDICE DE MANTENIMIENTO	COMPLEJIDAD CICLOMÁTICA	PROFUNDIDAD DE HERENCIA	ACOPAMIENTO DE CLASES	LÍNEAS DE CÓDIGO FUENTE	LÍNEAS DE CÓDIGO FUENTE EJECUTABLE
WebPtoVta	83	3.226	4	378	35.047	17.965

Figura 7.13: Métricas Visual Studio Punto de Venta actual

PROYECTO	ÍNDICE DE MANTENIMIENTO	COMPLEJIDAD CICLOMÁTICA	PROFUNDIDAD DE HERENCIA	ACOPPLAMIENTO DE CLASES	LÍNEAS DE CÓDIGO FUENTE	LÍNEAS DE CÓDIGO FUENTE EJECUTABLE
apiPtoVtaWebERP	68	69	2	105	629	215
apiPtoVtaWebERP. Data	80	343	1	122	3.676	1.416
apiPtoVtaWebERP. Model	98	781	1	11	646	28

**Figura 7.14:** Métricas Visual Studio Punto de Venta nuevo

En las imágenes presentadas se puede apreciar que en todas las métricas excluyendo el índice de mantenimiento son mejores (de menor valor). Si bien en la comparación no se está comparando el nuevo Punto de Venta en su totalidad, estas pruebas nos dan una buena visión acerca de las mejoras realizadas. Con respecto al índice de mantenimiento, se puede apreciar que en ambas aplicaciones se presenta un buen valor representado con una marca verde en Visual Studio.

Por otro lado se puede notar que la complejidad ciclomática, medida que evalúa la cantidad de lógica de decisión en una función de código fuente es mucho menor, lo que trae los siguientes beneficios:

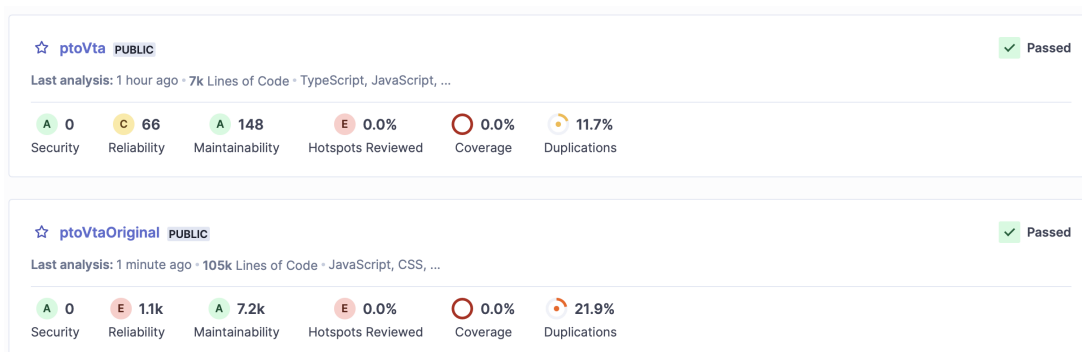
1. **Mantenibilidad:** Un software con menor complejidad ciclomática generalmente es más fácil de entender y modificar. Además, con menos caminos lógicos y decisiones en el código, hay menos posibilidades de introducir errores durante el desarrollo o mantenimiento.
2. **Legibilidad:** Menos complejidad significa que el código tiende a ser más claro y más fácil de seguir.
3. **Testabilidad:** Un código con menor complejidad ciclomática tendrá menos caminos de ejecución, lo que facilita la creación de casos de prueba completos y efectivos.
4. **Rendimiento:** Un código más simple y directo a menudo puede resultar en un mejor rendimiento, ya que hay menos lógica y operaciones complejas que el sistema debe ejecutar.
5. **Comprensión:** Para nuevos desarrolladores que se unan al proyecto, entender un código con menor complejidad será más rápido y fácil.

También otro punto importante a destacar es la cantidad de líneas de código. Si

se suma el total de líneas de código de los tres proyectos que componen la API del nuevo Punto de Venta (en la imagen: apiPtoVtaWebERP, apiPtoVtaWebERP.Data y apiPtoVtaWebERP.Model), obtenemos un total de 4.951. Por su lado la versión original del sistema tiene un total de 35.047, que si bien es del proyecto completo y no solo de la API, es siete veces más dicha cantidad. Más adelante en la sección se presentarán pruebas con el proyecto del frontend del sistema nuevo que confirmará está gran reducción en la cantidad de código frente al sistema antiguo.

## 7.4. Pruebas con Sonarqube

En esta prueba, se volvió a medir el proyecto completo del Punto de Venta original frente al proyecto del frontend del nuevo sistema. Estos fueron los resultados:



**Figura 7.15:** Métricas Sonarqube  
*ptoVta* refiere al nuevo POS y *ptoVtaOriginal* al actual

Los resultados obtenidos muestran una clara mejoría de la nueva aplicación con respecto a la antigua. En este caso en concreto, volviendo a lo mencionado en el punto anterior con relación a las líneas de código. Sonarqube identifica alrededor de 7 mil líneas de código totales en el proyecto del frontend del nuevo software. Esto sumado con los valores obtenidos de la API nos daría un total de alrededor de 12 mil líneas de código totales. Lo que significa que con un tercio de líneas de código se logró crear un software con las mismas capacidades, sin duda una reducción considerable.

Otros aspectos en los que se obtiene una mejor calificación incluyen la fiabilidad del código (reliability), que evalúa la identificación de fallos presentes que podrían resultar en comportamientos inesperados o errores durante la ejecución del software. El nuevo Punto de Venta obtuvo una calificación de letra C, presentando 66 casos

de posibles fallas, en comparación con la calificación de E del sistema original, que registró 1100 situaciones donde la aplicación podría fallar. También se redujo notablemente la duplicidad de código, que se refiere a fragmentos de código repetidos exacta o casi exactamente en múltiples lugares dentro del código fuente. La nueva versión del POS marcó un 11.7% de duplicidad, frente al 21.9% de la versión anterior.

## 8. Conclusión

El propósito de este proyecto fue crear una nueva versión del Punto de Venta Web de la empresa Opencode, con el objetivo de optimizar la plataforma y mejorar la experiencia de los usuarios.

Se identificaron como principales desventajas de la versión anterior un sistema poco amigable e irregular en cuanto a su rendimiento. El trabajo se enfocó en estas áreas, logrando mejoras significativas y añadiendo nuevas funcionalidades que se mencionan a continuación.

En primer lugar, se mejoró el rendimiento del POS, logrando agilizar las transacciones comerciales al integrar los sistemas existentes de Opencode con sistemas externos como Transbank y la página del Servicio de Impuestos Internos (SII). Esto permitió una experiencia más directa y rápida.

También se modernizó y simplificó la interfaz, haciendo que su uso sea mucho más intuitivo y menos laborioso para los clientes. Esta mejora se refleja directamente en una mayor apreciación por el servicio contratado.

Estas transformaciones y nuevas funcionalidades se realizaron pensando en lo mejor para la empresa, buscando mejorar la relación con sus consumidores y atender sus expectativas y demandas. Aunque fue un trabajo complejo debido a las numerosas mejoras necesarias y el desarrollo de un software nuevo desde cero, la integración exitosa de sistemas externos como Transbank y el SII añadió un valor adicional. Especialmente desafiante fue la integración con el SII, ya que este sistema estuvo fuera de servicio en numerosas ocasiones, dificultando las pruebas.

Finalmente, con esta nueva versión del Punto de Venta Web, se logró perfeccionar el modelo anterior, obteniendo la aprobación y satisfacción del dueño de la empresa. Este proyecto presentó una oportunidad para que el autor demostrara los conocimientos adquiridos a lo largo de la carrera y los aplicara en la creación de un software funcional para una empresa con trayectoria. Además de la experiencia obtenida, el proyecto destacó la importancia de valores imprescindibles como la responsabilidad, organización y disciplina. Las nuevas funcionalidades del software han otorgado un valor agregado y serán mantenidas por la empresa, permitiendo concluir que se cumplieron satisfactoriamente los objetivos propuestos en esta memoria.

## Referencias

- [1] Cris Busquets. Medir la usabilidad con el sistema de escalas de usabilidad (sus). <https://www.uifrommars.com/como-medir-usabilidad-que-es-sus/>. Revisado: 27-06-2024.
- [2] DBeaver Community. Dbeaver community | free universal database tool. <https://dbeaver.io>. Revisado: 27-06-2024.
- [3] Soham de Roy. ¿qué es tailwind css? guía para principiantes. <https://www.freecodecamp.org/espanol/news/que-es-tailwind-css-guia-para-principiantes/>. Revisado: 27-06-2024. Traducido por: Jorge Sierra.
- [4] OpenJS Foundation. ¿qué es electron? <https://www.electronjs.org/es/docs/latest/>. Revisado: 27-06-2024.
- [5] Inc. GitHub. Acerca de github y git. <https://docs.github.com/es/get-started/start-your-journey/about-github-and-git>. Revisado: 27-06-2024.
- [6] Diego Alexander Forero Higuera. Arquitectura multitenant: qué es y por qué es importante. <https://platzi.com/blog/multi-tenant-que-es-y-por-que-es-importante/>. Revisado: 27-06-2024.
- [7] Microsoft. ¿qué es asp.net core? <https://dotnet.microsoft.com/es-es/learn/aspnet/what-is-aspnet-core>. Revisado: 27-06-2024.
- [8] Oracle. ¿qué es mysql? <https://www.oracle.com/cl/mysql/what-is-mysql/>. Revisado: 27-06-2024.
- [9] Vercel. What is next.js? <https://nextjs.org/docs>. Revisado: 27-06-2024.