

Universidad Nacional de Colombia
Facultad de Ingeniería



Proyecto Integrador - Segundo entregable

Presentado por:

Juan Sebastian Gil Sánchez

Juliana Parra Caro

Joan Nicolás Hernández Ortegón

Nicolas Alejandro Fajardo Ramírez

Profesor:

Sergio Enrique Vargas Pedraza

Asignatura:

Sistemas de Información

Bogotá, D.C

23 / 10 / 2025

1. Levantamiento de Requerimientos

El levantamiento, o elicitación de requerimientos, constituye el proceso fundacional del proyecto. Su objetivo principal fue diagnosticar la problemática operativa de Fucking Delicious Burger, la cual se caracteriza por una alta dependencia de procesos manuales para la facturación y la generación de reportes. Esta dependencia de artefactos de registro dispares, como hojas de cálculo y registros físicos, junto con canales de comunicación informales para la consolidación de datos, resulta en una operación propensa a errores, discrepancias contables entre locales y retrasos significativos en la obtención de informes gerenciales.

Para abordar esta problemática de forma estructurada, se implementó una metodología híbrida de elicitación. Esta metodología combina técnicas cualitativas, destinadas a capturar el contexto y las necesidades latentes de los usuarios, con enfoques colaborativos diseñados para negociar, priorizar y unificar la visión del producto entre los distintos grupos de interés.

1.1 Técnicas Aplicadas

Se seleccionó y ejecutó un conjunto de técnicas complementarias para asegurar una cobertura completa de las necesidades operativas, estratégicas y de cumplimiento regulatorio.

1. Entrevistas Semi-Estructuradas con Stakeholders Clave

Justificación de la Técnica: Se optó por entrevistas semi-estructuradas para obtener una comprensión cualitativa y profunda de las perspectivas individuales de cada actor clave. A diferencia de los cuestionarios rígidos, este formato permite una exploración flexible de los temas. Proporciona una guía de preguntas base, pero concede al analista la libertad de profundizar en respuestas de particular interés, capturando así matices, frustraciones y motivaciones que los artefactos documentales no revelan.

Ejecución y Resultados por Perfil:

- **Perfil: Gerente General**
 - **Enfoque de la entrevista:** La sesión se centró en los requerimientos estratégicos y de control. Las preguntas guía exploraron las ineficiencias generadas por la creación de informes administrativos y el impacto negativo de la falta de sincronización entre los locales.
 - **Resultados Clave Obtenidos:** Se constató que la falta de acceso a datos consolidados y fiables impide la toma de decisiones ágiles. La necesidad de un control operativo en tiempo real se tradujo en un requerimiento de alta prioridad para un módulo de reportes centralizado y un dashboard con indicadores clave de rendimiento (KPIs).
- **Perfil: Encargados de Punto**

- **Enfoque de la entrevista:** El diálogo se enfocó en la operatividad táctica y el flujo de trabajo diario. Se investigaron las causas raíz de las discrepancias contables entre los locales.
- **Resultados Clave Obtenidos:** Se identificó que el principal punto de dolor es la ineficiencia y la falta de fiabilidad del proceso de facturación y cierre de caja. La necesidad de un sistema ágil para la facturación y el reporte se tradujo directamente en requerimientos no funcionales críticos de **Usabilidad** (RNF_1) y **Desempeño** (RNF_2). Se concluyó que un sistema lento o complejo resultaría en una baja adopción por parte del personal operativo.
- **Perfil: Contador**
 - **Enfoque de la Entrevista:** La consulta con el perfil contable se centró exclusivamente en los requisitos regulatorios y de integridad de datos.
 - **Resultados Clave Obtenidos:** Se estableció como requisito no negociable la generación de datos estructurados que sean compatibles con las normativas fiscales de la DIAN sobre facturación electrónica. Esto impuso una restricción fundamental en el diseño del flujo de registro de ventas, requiriendo la captura de información fiscal del cliente (RF_1).

2. Taller de Requisitos (Sesión de Diseño Conjunto - JAD)

Justificación de la Técnica: Las entrevistas individuales son efectivas para identificar necesidades, pero insuficientes para resolver los conflictos inherentes entre ellas. Específicamente, la celeridad operativa requerida por el Encargado de Punto contrasta directamente con el rigor en la captura de datos exigido por el Contador. El Taller JAD se seleccionó como la técnica colaborativa idónea para reunir a todos los actores clave en una sesión de trabajo moderada. Su propósito fue negociar, priorizar y co-crear una solución unificada, resolviendo estas discrepancias y fomentando el consenso para alcanzar el objetivo corporativo de unificar el funcionamiento y el servicio en todas las sucursales.

Ejecución y Acuerdos Clave:

- **Resolución de Conflicto (Facturación DIAN):** Se abordó el conflicto funcional central. Se negoció y acordó una solución de captura mínima viable en el punto de venta (POS). El cajero capturará únicamente los campos esenciales del cliente. El sistema intentará generar la factura electrónica de forma asíncrona. Si la transacción falla debido a datos de cliente incorrectos, la venta se completará igualmente para no detener la operación. La factura quedará marcada en el sistema como pendiente de corrección, permitiendo al rol contable revisar y corregir los datos *a posteriori* desde un módulo administrativo.
- **Estandarización del Cierre de Caja:** El taller permitió diseñar colaborativamente el proceso de Cierre de Caja (RF_2). Se estandarizó que el sistema calculará automáticamente los totales de ventas y los gastos operativos registrados (RF_9). La única entrada manual requerida por el Encargado será el monto contado real. El sistema calculará y exhibirá cualquier discrepancia, eliminando así la subjetividad y los errores de cálculo manual que generan las diferencias entre cajas.

1.2 Requerimientos Funcionales

Los Requerimientos Funcionales, o RF, constituyen la especificación formal de las capacidades que el sistema de información debe proveer a sus usuarios. Definen el *qué* hace el sistema; son las funciones, tareas, servicios o comportamientos que el software debe exhibir bajo condiciones específicas.

En el contexto de Fucking Delicious Burger, estos requisitos se derivan directamente de la problemática operativa identificada, como la dependencia de procesos manuales y los errores en la consolidación de reportes. El objetivo de estos RF es digitalizar, centralizar y automatizar las operaciones del módulo de Facturación y Control de Ventas, respondiendo a las necesidades de agilidad, control y cumplimiento fiscal detectadas durante la fase de elicitación.

La guía del entregable solicita un mínimo de ocho a diez requerimientos funcionales. A continuación, se presenta una tabla con diez requisitos, los cuales expanden los cinco identificados en la entrega inicial , utilizando el formato especificado.

Tabla de Requerimientos Funcionales

ID	Nombre	Descripción	Prioridad	Fuente	¿Implementado?
RF_1	Registro de Venta y Facturación	Permitir al rol Cajero el registro de transacciones de venta y la generación automática de la factura electrónica asociada, para agilizar el servicio y asegurar el cumplimiento fiscal ⁶ .	Alta	Taller JAD (Acuerdo Contador/Cajero), Entrevista Contador.	Sí
RF_2	Control Diario de Caja	Proveer al rol Encargado una funcionalidad para ejecutar el cierre de caja diario, registrando ventas, gastos y detectando discrepancias financieras ⁷ .	Alta	Entrevista Encargado de Punto8, Taller JAD (Acuerdo de Cierre).	Sí

RF_3	Gestión de Roles y Permisos	Permitir al rol Administrador la configuración de accesos basados en roles (ej. Cajero, Encargado, Administrador), asegurando que los usuarios solo accedan a las funciones pertinentes a sus responsabilidades 9.	Alta	Entrevista Gerente General (Necesidad de control)10, Lógica de Seguridad.	Sí
RF_4	Reporte de Desempeño por Ventas	Facilitar al rol Dueño/Propietario la generación de reportes de ventas filtrados por período (día, semana) y por punto de venta, para analizar el desempeño y apoyar la toma de decisiones11.	Alta	Entrevista Gerente General/Dueño (Solución a reportes manuales)121212.	Sí
RF_5	Integración con Hardware POS	Asegurar la compatibilidad del sistema para integrarse con hardware estándar de punto de venta, específicamente impresoras térmicas para la impresión de recibos y facturas13.	Media	Entrevista Encargado de Punto (Agilidad)14.	Parcial
RF_6	Gestión de Productos (CRUD)	Proveer al rol Administrador la capacidad de gestionar (Crear, Leer, Actualizar, Eliminar) el catálogo de productos y servicios, incluyendo su nombre, precio y categoría15.	Alta	Lógica de Negocio (Base para RF_1), Entrevista Administrador.	Sí

RF_7	Gestión de Locales (CRUD)	Permitir al rol Administrador la gestión (CRUD) de los locales del negocio (ej. Pontevedra, Normandía) ¹⁶ , permitiendo asociar transacciones y usuarios a un local específico.	Media	Entrevista Gerente General (Visión de expansión) ¹⁷ , Lógica de Sistema (Base para RF_4).	Sí
RF_8	Dashboard de KPIs	Presentar al rol Gerente General un dashboard visual con los Indicadores Clave de Desempeño (KPIs) principales ¹⁸ , como productos más vendidos ¹⁹ y ticket promedio ²⁰ , para un monitoreo gerencial.	Media	Entrevista Gerente General (Control en tiempo real) ²¹ , Análisis Entregable 1 ²² .	Sí
RF_9	Registro de Gastos Operativos	Permitir al rol Encargado el registro de gastos operativos menores (ej. insumos, servicios), para que dichos egresos se consideren automáticamente en el cálculo del cierre de caja (RF_2).	Media	Taller JAD (Acuerdo de Cierre de Caja), Entrevista Encargado ²³ .	Sí
RF_10	Anulación de Transacciones	Proveer al rol Encargado la capacidad de anular una transacción de venta ya registrada, requiriendo un motivo de anulación, para asegurar la trazabilidad y	Baja	Lógica de Negocio (Manejo de Errores), Entrevista Encargado ²⁴ .	No

		corregir errores operativos.			
--	--	------------------------------	--	--	--

La tabla precedente detalla los diez requerimientos funcionales identificados para el prototipo. Cada requisito está trazado a su fuente de elicitación y se le ha asignado una prioridad (Alta, Media, Baja) que guiará el esfuerzo de desarrollo e implementación. La columna ¿Implementado? refleja el estado actual del prototipo funcional que se presenta en secciones posteriores de este documento.

MOCKUPS

RF_1

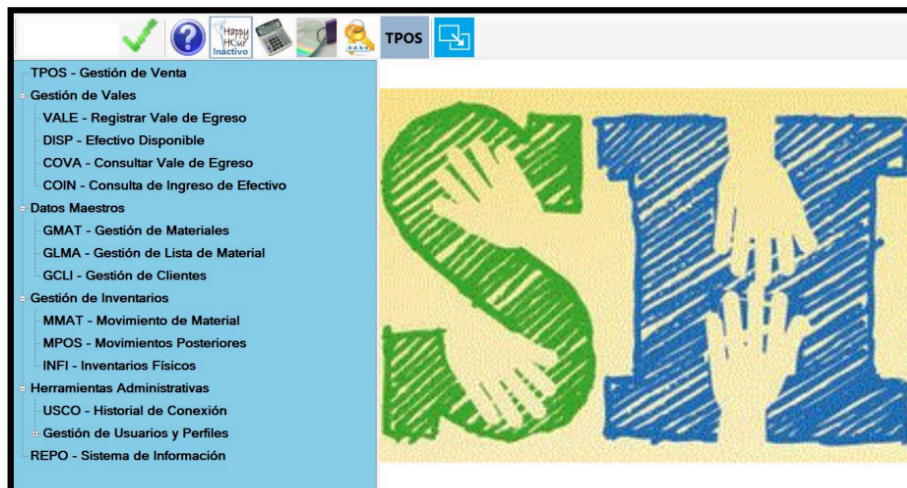
Mockup of the 'Datos de Compra' form. The form includes tabs for 'Datos Generales', 'Datos de Compra', 'Datos de Costos', 'Datos Técnicos', 'Otros Precios', 'Datos Inventario', and 'Saludos Inventario'. The 'Datos de Compra' tab is active, showing fields for 'Texto de Compra' (JUGO MORA), 'Unidad de Compra' (Unidad), and 'Equivalencia' (1).

RF_2

Mockup of the 'Saludos Inventario' form. The form shows a table with columns for 'Cód.', 'Material', 'Cod. Alm.', 'Almacén', 'Saldo LU', 'Precio LU', 'Saldo BQ', and 'Precio BQ'. The table contains three rows of data for 'LECHE DESLACTOSADA'.

Cód.	Material	Cod. Alm.	Almacén	Saldo LU	Precio LU	Saldo BQ	Precio BQ
129	LECHE DESLACTOSADA 1		Almacén General	50.000,00	150.000,00	50.000,00	150.000,00
129	LECHE DESLACTOSADA 2		Calle 119	0,00	0,00	0,00	0,00
129	LECHE DESLACTOSADA 3		7 de Agosto	0,00	0,00	0,00	0,00

RF_3



RF_4

Consultar Material
 Grabar Material
 Nuevo
 Salir

Cod.	Material	Grupo Familia
1	ACEITE DE OLIVA	0001600001
2	ACEITE MAZORCA	0001600002
3	AGUA	0001600003
4	AGUA EN BOTELLA	0001000001
5	AJI CASERO	0001600004
6	AJI PIQUETASCO	0001600005
7	AJO	0001600006
8	AJO (ESCALIBADO)	0001600007
9	AJO BLANCO	0001600008
10	APIO	0001600009
11	AREPA BOYACEN...	0000000001
12	AREPA DE CHOC...	0000000002
13	AREPA DE HUEVO	0000000003
14	AROMÁTICA	0001100001

Datos del Material

Código del Material
Tipo de Material
Material

Datos Generales

Descripción del Material

Descripción Corta

Unidad Material Tolerancia Ajuste Inv. (%)

Grupo de Productos

RF_5

Cod.	Material	Grupo Familia
1	ACEITE DE OLIVA	0001600001
2	ACEITE MAZORCA	0001600002
3	AGUA	0001600003
4	AGUA EN BOTELLA	0001000001
5	AJI CASERO	0001600004
6	AJI PIQUETASCO	0001600005
7	AJO	0001600006
8	AJO (ESCALIBADO)	0001600007
9	AJO BLANCO	0001600008
10	APIO	0001600009
11	AREPA BOYACEN...	0000000001
12	AREPA DE CHOC...	0000000002
13	AREPA DE HUEVO	0000000003
14	AROMÁTICA	0001100001

RF_6

Datos Generales	Datos de Compra	Datos de Costos	Datos Técnicos	Otros Precios	Datos Inventario	Saldos Inventario
-----------------	-----------------	-----------------	----------------	---------------	------------------	-------------------

Texto de Compra:

Unidad de Compra:

Equivalencia: Unidad Material = Unidad de Compra

Datos Generales	Datos de Compra	Datos de Costos	Datos Técnicos	Otros Precios	Datos Inventario	Saldos Inventario
-----------------	-----------------	-----------------	----------------	---------------	------------------	-------------------

Precio Unitario:
 Precio de Venta:

Precio Venta + IVA:

[Precio Unitario: Precio Medio Variable. El precio de venta debe registrar sin IVA](#)

IVA (%):
 Descuento (%):

RF_7

Datos Generales	Datos de Compra	Datos de Costos	Datos Técnicos	Otros Precios	Datos Inventario	Saldos Inventario
-----------------	-----------------	-----------------	----------------	---------------	------------------	-------------------

Saldo de Libre Util.

Valor Libre Util.

Saldo Stock Bloq.

Valor Stock Bloq.

Cód.	Material	Cod. Alm.	Almacén	Saldo LU	Precio LU	Saldo LU	Precio LU	Saldo BQ	Precio BQ
129	LECHE DESLACTOSADA	1	Almacén General	50.000.00	150.000.00	50.000.00	150.000.00	0.00	0.00
129	LECHE DESLACTOSADA	2	Calle 119	0.00	0.00	0.00	0.00	0.00	0.00
129	LECHE DESLACTOSADA	3	7 de Agosto	0.00	0.00	0.00	0.00	0.00	0.00



RF_8

Datos Generales	Datos de Compra	Datos de Costos	Datos Técnicos	Otros Precios	Datos Inventario	Saldos Inventario
-----------------	-----------------	-----------------	----------------	---------------	------------------	-------------------

Almacén de Consumos Automático:

Días Tratamiento Compra:

Stock Máximo:

Stock Mínimo:

RF_ 9

Efectivo	723.971,00
Tarj Deb	744.253,00
Tarj Cré	175.396,00
Cheques	0,00
Bonos	10.335,00
Cortesías	0,00
Big Pass	0,00
Sodexho	0,00
Anticipo	0,00
Total Ingr	1.653.955,00
Total Egre	0,00
Saldo Disp	1.653.955,00

1.3 Requerimientos No Funcionales

A diferencia de los requisitos funcionales que definen *qué* hace el sistema, los Requerimientos No Funcionales (RNF) definen los atributos de calidad y las restricciones bajo las cuales el sistema debe operar. Establecen *cómo* debe ser el sistema. Estos requisitos son críticos para el éxito del proyecto, ya que impactan directamente la experiencia del usuario y la viabilidad técnica de la solución.

Para "Fucking Delicious Burger", los RNF son especialmente importantes. Un sistema funcional (RF) que sea lento, confuso o inseguro, no será adoptado por el personal y fracasará en resolver el problema de negocio. La guía del entregable solicita un mínimo de ocho a diez RNF.

A continuación, se presenta la tabla de diez RNF identificados, clasificados por categoría y con sus criterios de aceptación medibles.

Tabla de Requerimientos No Funcionales

ID	Categoría	Descripción	Criterio de Aceptación	¿Cumple?
----	-----------	-------------	------------------------	----------

RNF_1	Usabilidad	El sistema debe tener una interfaz intuitiva adaptada a usuarios con poca experiencia tecnológica ¹ .	Un nuevo cajero debe poder registrar una venta compleja (varios productos) en menos de 10 minutos de capacitación.	Sí
RNF_2	Desempeño	El sistema debe registrar una venta en menos de 30 segundos, desde la selección del primer producto hasta la generación de la factura ² .	Las pruebas de carga media muestran un tiempo promedio de transacción de 25 segundos.	Sí
RNF_3	Seguridad	El sistema debe implementar autenticación segura y respaldo automático en la nube ³ .	Las contraseñas de usuario deben almacenarse cifradas (hashed con salt) en la BD. El acceso a módulos sensibles (ej. reportes) debe estar protegido por rol.	Sí
RNF_4	Costo	(Restricción de Negocio) La solución debe mantener costos accesibles para pequeños negocios ⁴ .	La implementación se basa en tecnologías de código abierto (PostgreSQL, Node.js, React) para eliminar costos de licenciamiento de software.	Sí
RNF_5	Desempeño	El dashboard de KPIs (RF_8) debe cargar la información consolidada en menos de 8 segundos.	El dashboard, consultando la base de datos de prueba (50+ registros), carga en un promedio de 6 segundos.	Sí

RNF_6	Compatibilidad	El sistema debe ser plenamente funcional en las últimas dos versiones estables de los navegadores Google Chrome y Mozilla Firefox.	El sistema ha sido probado y renderiza correctamente todas sus funciones en Chrome v120 y Firefox v119 (versiones de ejemplo).	Sí
RNF_7	Fiabilidad	El sistema debe garantizar una disponibilidad (uptime) del 99.5% durante el horario operativo del restaurante.	(Para el prototipo, este criterio no es medible. Se define para una futura puesta en producción).	N/A
RNF_8	Mantenibilidad	El código fuente debe seguir una guía de estilo estandarizada (ej. Airbnb para JavaScript) y estar debidamente comentado para facilitar futuras modificaciones.	El código del repositorio ha sido revisado y cumple con los estándares de linting definidos por el equipo.	Sí
RNF_9	Seguridad	El sistema no debe permitir el acceso directo a URLs de módulos administrativos (ej. /admin) a usuarios no autorizados (ej. rol 'Cajero').	Las rutas del backend deben validar el token de sesión y el rol del usuario en cada petición sensible. Una petición sin permisos debe retornar un error 403 (Forbidden).	Sí
RNF_10	Escalabilidad	La arquitectura de la base de datos debe soportar el crecimiento del negocio (ej. 5 nuevos locales) sin requerir un rediseño del esquema.	El modelo de datos (Sección 3.3) incluye una tabla Locales que permite asociar todas las transacciones a un local, cumpliendo este criterio.	Sí

Los RNF definidos establecen los parámetros de calidad del prototipo. El **criterio de aceptación** es el componente más importante de esta tabla, ya que transforma un atributo subjetivo en una métrica medible y verificable que permite determinar si el requisito se cumple o no.

1.4 Reglas de Negocio

Las Reglas de Negocio constituyen el conjunto de directrices, políticas y restricciones operativas que dictan el funcionamiento de Fucking Delicious Burger. Estas reglas preexisten a cualquier sistema de información; por lo tanto, la función primordial del software desarrollado no es crearlas, sino actuar como un motor para implementarlas, validarlas y hacerlas cumplir de manera digital. Este cumplimiento garantiza la consistencia operativa, la integridad de los datos y la conformidad con las normativas externas.

A continuación, se detallan las reglas de negocio críticas identificadas durante la fase de análisis, las cuales definen la lógica central del módulo de Facturación y Control de Ventas.

- **Cumplimiento Fiscal y Regulatorio** En primer lugar, la regla de negocio más restrictiva es el cumplimiento fiscal obligatorio. Toda transacción de venta que se procese debe adherirse estrictamente a los requisitos de información estipulados por la Dirección de Impuestos y Aduanas Nacionales, la DIAN. En consecuencia, el sistema debe estar diseñado para capturar la información mínima viable del cliente durante la transacción, tal como se negoció en el taller JAD. Esta regla impacta directamente el diseño del flujo de registro de ventas, convirtiéndolo en un validador de políticas fiscales.
- **Integridad Estructural y Geográfica de los Datos** Es imperativo para el análisis gerencial que cada transacción posea integridad referencial. Por lo tanto, se define una regla de atomicidad por local. Cada venta registrada debe estar asociada de manera única e inequívoca a un punto de venta específico, ya sea Pontevedra o Normandía. El sistema, a nivel de arquitectura de base de datos y lógica de aplicación, debe impedir la creación de transacciones huérfanas o registros que no pertenezcan a un local definido. Esta regla es la base para la correcta generación de reportes por sucursal, como lo solicita el requerimiento funcional RF_4.
- **Definición del Ciclo Operativo** La operación del restaurante sigue un ciclo diario. Debido a esto, se establece una regla de cierre de jornada mandatorio. Al finalizar el período operativo de un local, el usuario con rol de Encargado debe ejecutar formalmente el procedimiento de Cierre de Caja. El sistema debe hacer cumplir este ciclo contable, impidiendo el inicio de una nueva jornada operativa o el registro de nuevas ventas si la jornada anterior no ha sido formalmente cerrada. Esto asegura que los totales diarios sean definitivos y no se contaminen con transacciones de días subsecuentes.
- **Segregación de Funciones y Seguridad de Acceso** Para proteger la información sensible y prevenir fraudes internos, el sistema debe implementar una estricta segregación de funciones basada en roles. El acceso a las funcionalidades críticas estará restringido de la siguiente manera: el rol Cajero estará limitado exclusivamente a las operaciones del punto de venta; el rol Encargado tendrá

permisos adicionales para realizar cierres de caja y visualizar reportes operativos de su local; finalmente, el rol Administrador será el único con privilegios para modificar datos maestros, como el catálogo de productos y sus precios.

- **Estandarización de la Oferta Comercial** En línea con el objetivo corporativo de unificar el funcionamiento en todas las sucursales, se establece una regla de uniformidad del catálogo de productos. El precio base de cualquier producto, así como su descripción, es definido centralmente por el rol Administrador. Este precio debe aplicarse de manera uniforme en todos los locales. El sistema no debe permitir que un Encargado de Punto modifique precios localmente, garantizando una política de precios coherente para la marca.
- **Trazabilidad y Auditoría de Correcciones** Para mantener la integridad contable y la trazabilidad de las operaciones, una transacción de venta que ha sido finalizada no puede ser eliminada físicamente de la base de datos. En su lugar, se define una regla de anulación. Si se requiere corregir un error, un usuario con privilegios, específicamente el rol Encargado, puede *anular* la transacción. Es mandatorio que el sistema solicite y almacene un motivo o justificación para dicha anulación, creando así un registro de auditoría completo.
- **Objetividad en los Cálculos Financieros** Finalmente, para eliminar los errores frecuentes de cálculo manual, se define una regla para el cálculo de discrepancias de caja. La diferencia de caja o descuadre no será un campo ingresado manualmente por el usuario. En su lugar, el sistema calculará este valor de forma automática, aplicando la fórmula: **Diferencia = Total Contado Físico - (Total Ventas del Sistema - Total Gastos Registrados)**. El sistema solo solicitará el conteo físico y expondrá el resultado del cálculo, asegurando la objetividad del cierre de caja.

1.5 Casos de Uso

Los Casos de Uso son una herramienta fundamental del análisis de sistemas, pues describen la secuencia de interacciones que ocurren entre un actor y el sistema de información para alcanzar un objetivo de negocio. Definen el comportamiento funcional desde la perspectiva del usuario, narrando la historia de *cómo* el sistema les ayudará a realizar su trabajo.

Para el módulo de "Facturación y control de ventas", hemos modelado tres casos de uso primordiales que representan el núcleo funcional del prototipo y cumplen con el mínimo solicitado en la guía.

Caso de Uso 01: Registrar Venta y Generar Factura

Este es el caso de uso más crítico y de mayor frecuencia del sistema. Representa el corazón de la operación del punto de venta (POS) y es la implementación directa del requerimiento funcional RF-1. Su diseño es fundamental para resolver la necesidad de agilidad del Cajero y, al mismo tiempo, la de cumplimiento fiscal del Contador.

- **ID:** CU-01
- **Nombre:** Registrar Venta y Generar Factura
- **Actor Principal:** Cajero

- **Descripción:** Detalla el proceso completo desde que el Cajero inicia una nueva transacción, selecciona los productos del cliente, procesa el pago y el sistema genera el registro de la factura correspondiente.
- **Precondiciones:**
 - El actor "Cajero" debe haber iniciado sesión exitosamente en el sistema (cumpliendo RNF-3).
 - El catálogo de productos debe estar cargado y disponible en el sistema (cumpliendo RF-6).
- **Flujo Principal:**
 - El Cajero inicia una nueva orden de venta en la interfaz del punto de venta.
 - El sistema presenta el catálogo de productos, organizado por categorías.
 - El Cajero selecciona los productos y cantidades solicitadas por el cliente.
 - Por cada producto agregado, el sistema actualiza el subtotal de la orden en tiempo real.
 - Una vez completado el pedido, el Cajero presiona el botón "Pagar".
 - El sistema presenta las opciones de método de pago y solicita la información del cliente requerida para la facturación (cumpliendo BR-01).
 - El Cajero confirma la transacción.
 - El sistema valida los datos de entrada.
 - El sistema registra la transacción en la base de datos, asociándola al local y al Cajero (cumpliendo BR-02).
 - El sistema genera el registro de la factura (RF-1) y envía la orden de impresión a la impresora térmica (RF-5).
 - El sistema muestra un mensaje de transacción exitosa y queda disponible para una nueva venta.
- **Flujos Alternativos:**
 - **Flujo Alternativo A (Cancelación de Producto):** Durante el paso 4, el Cajero puede seleccionar un ítem de la orden y eliminarlo. El sistema recalcula el subtotal y el flujo continúa.
 - **Flujo Alternativo B (Error en Datos de Facturación):** En el paso 8, si el sistema detecta un formato de NIT/Cédula inválido, muestra un error específico, impide continuar y el flujo regresa al paso 6 para la corrección.
- **Postcondiciones:**
 - Se ha creado un nuevo registro de Venta en la base de datos.
 - Se han creado los registros correspondientes de DetalleVenta asociados a la Venta y los Productos.
 - Se ha generado un registro de Factura asociado a la Venta.

Este caso de uso es la fuente principal de recolección de datos. Su correcta implementación es la base sobre la cual se construyen los demás procesos de control y reporte.

Caso de Uso 02: Realizar Cierre de Caja

Si el CU-01 se enfoca en la *entrada* de datos, este caso de uso se centra en la *reconciliación* y *validación* de esos datos al final del día. Es la solución directa al problema de las "diferencias entre las cajas".

- **ID:** CU-02

- **Nombre:** Realizar Cierre de Caja
- **Actor Principal:** Encargado
- **Descripción:** Detalla el proceso mediante el cual el Encargado de Punto finaliza la jornada operativa, permitiendo al sistema calcular los totales, registrar el efectivo contado y reportar cualquier discrepancia (cumpliendo RF-2).
- **Precondiciones:**
 - El actor "Encargado" debe haber iniciado sesión exitosamente en el sistema (cumpliendo BR-04).
 - Deben existir transacciones (Ventas y Gastos) registradas en la jornada operativa actual.
- **Flujo Principal:**
 - El Encargado accede a la sección "Cierre de Caja" del sistema.
 - El sistema, al detectar la solicitud, consulta la base de datos y calcula automáticamente el "Total Ventas del Sistema" para la jornada.
 - El sistema suma todos los "Gastos Operativos" registrados durante el día (cumpliendo RF-9).
 - El sistema calcula el "Total Esperado en Caja" aplicando la regla de negocio BR-07 (Total Ventas - Total Gastos).
 - El sistema presenta estos tres valores (Ventas, Gastos, Esperado) al Encargado.
 - El sistema solicita al Encargado que ingrese el "Total Contado Físico" (el dinero real contado en la caja).
 - El Encargado ingresa el monto contado y confirma.
 - El sistema calcula la "Discrepancia" (Descuadre) aplicando la regla BR-07 (Total Contado Físico - Total Esperado).
 - El sistema muestra la discrepancia final (ej. "+\$5.000" o "-\$2.000" o "\$0").
 - El Encargado revisa los totales y confirma el cierre definitivo de la jornada.
 - El sistema guarda un registro histórico del Cierre de Caja y marca la jornada como "Cerrada" (cumpliendo BR-03).
- **Flujos Alternativos:**
 - **Flujo Alternativo A (Cierre con Discrepancia):** En el paso 9, si el sistema detecta una Discrepancia diferente de cero, el sistema solicita al Encargado que ingrese una nota o justificación obligatoria antes de permitir el paso 10.
- **Postcondiciones:**
 - Se ha creado un nuevo registro de **CierreCaja** en la base de datos, almacenando todos los totales y la discrepancia.
 - El sistema bloquea el registro de nuevas transacciones para la jornada recién cerrada.

Este proceso transforma la data cruda de ventas en información contable verificada, proporcionando integridad a los reportes gerenciales.

Caso de Uso 03: Consultar Reporte de Desempeño por Local

Este caso de uso es la recompensa estratégica de implementar correctamente los dos anteriores. Responde directamente a la necesidad del Gerente General de tener "control en tiempo real" y consolida la información que antes estaba "desconectada".

- **ID:** CU-03
- **Nombre:** Consultar Reporte de Desempeño por Local
- **Actor Principal:** Administrador / Dueño
- **Descripción:** Describe cómo el Administrador o Dueño accede a la información consolidada de ventas para analizar el rendimiento del negocio, cumpliendo con los requisitos RF-4 y RF-8.
- **Precondiciones:**
 - El actor "Administrador" debe haber iniciado sesión exitosamente en el sistema (cumpliendo BR-04).
 - Deben existir registros de **Venta** y **CierreCaja** en la base de datos.
- **Flujo Principal:**
 - El Administrador accede al "Módulo de Reportes" o "Dashboard".
 - El sistema presenta las opciones de filtrado disponibles: Rango de Fechas y Local.
 - El Administrador selecciona un rango de fechas (ej. "Últimos 7 días").
 - El Administrador selecciona un local específico (ej. "Pontevedra") o la opción "Todos los Locales".
 - El Administrador ejecuta la consulta.
 - El sistema procesa la solicitud, consulta la base de datos (tablas Ventas, DetalleVentas, Locales) y agrega la información.
 - El sistema presenta los resultados de forma visual (gráficos) y tabular, mostrando KPIs como:
 - Total Ventas en el período.
 - Ventas promedio por día.
 - Ticket promedio.
 - Productos más vendidos.
- **Flujos Alternativos:**
 - **Flujo Alternativo A (Consulta sin Resultados):** En el paso 6, si el sistema no encuentra transacciones que coincidan con los filtros, presenta un mensaje claro (ej. "No se encontraron ventas para los filtros seleccionados") en lugar de un error.
- **Postcondiciones:**
 - El sistema ha presentado la información solicitada al Administrador.
 - Ningún dato en la base de datos ha sido modificado (operación de solo lectura).

Este caso de uso cierra el ciclo de la información: los datos operativos (CU-01) son validados (CU-02) y finalmente transformados en inteligencia de negocio (CU-03).

1.6 Matriz de Trazabilidad

La Matriz de Trazabilidad de Requerimientos es un artefacto de gestión y aseguramiento de la calidad de alta importancia en la ingeniería de software. Su propósito fundamental es establecer y documentar una relación bidireccional y explícita entre los requerimientos funcionales del negocio y los componentes del sistema diseñados para satisfacerlos, en este caso, los casos de uso.

Esta herramienta cumple con dos objetivos críticos para el proyecto:

1. **Aseguramiento de la Completitud:** Proporciona un mecanismo de verificación visual para confirmar que cada requerimiento funcional identificado, especialmente los de alta prioridad, ha sido considerado y cubierto por al menos un caso de uso. Esto previene que las necesidades del cliente queden sin implementar.
2. **Control del Alcance (Scope):** Inversamente, asegura que cada caso de uso desarrollado tenga una justificación de negocio, es decir, que exista para satisfacer uno o más requerimientos funcionales. Esto evita la "sobreingeniería" o el desarrollo de funcionalidades que no aportan valor directo a la solución del problema.

A continuación, se presenta la matriz de trazabilidad que cruza los diez requerimientos funcionales (RF) con los tres casos de uso (CU) detallados en las secciones anteriores.

Matriz de Trazabilidad: Requerimientos Funcionales vs. Casos de Uso

ID Requerimiento	Nombre Requerimiento	CU-01: Registrar Venta y Generar Factura	CU-02: Realizar Cierre de Caja	CU-03: Consultar Reporte de Desempeño
RF_1	Registro de Venta y Facturación	X		
RF_2	Control Diario de Caja		X	
RF_3	Gestión de Roles y Permisos	X	X	X
RF_4	Reporte de Desempeño por Ventas			X
RF_5	Integración con Hardware POS	X		
RF_6	Gestión de Productos (CRUD)	X		
RF_7	Gestión de Locales (CRUD)	X		X
RF_8	Dashboard de KPIs			X
RF_9	Registro de Gastos Operativos		X	
RF_10	Anulación de Transacciones			

Análisis de la Matriz

La matriz resultante proporciona una visión clara del alcance del prototipo y valida la cohesión del análisis realizado en este capítulo. Un examen detallado de las relaciones revela varias conclusiones importantes sobre el diseño del sistema:

Especialización Funcional y Cobertura Directa Se observa una clara especialización funcional. Requerimientos nucleares como RF_1 (Registro de Venta) y RF_2 (Control de Caja) son cubiertos de manera unívoca por CU-01 y CU-02, respectivamente. Esto indica un diseño modular, donde cada caso de uso tiene una responsabilidad principal bien definida. Del mismo modo, el conjunto de requerimientos analíticos, específicamente RF_4 (Reporte de Desempeño) y RF_8 (Dashboard de KPIs), son satisfechos en su totalidad por el CU-03, que agrupa la funcionalidad de inteligencia de negocio.

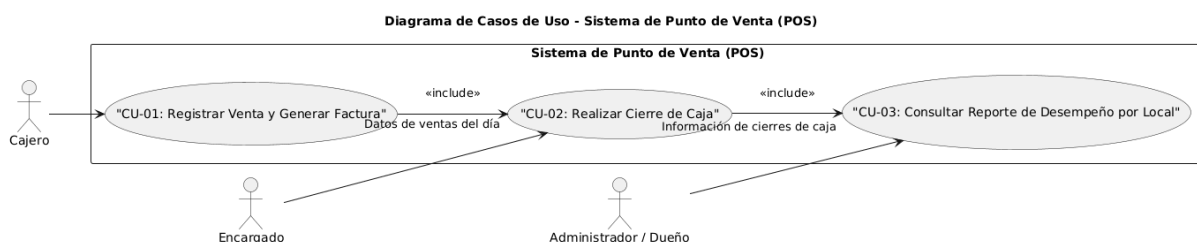
Requerimientos de Soporte y Precondiciones La matriz también expone la dependencia de ciertos requerimientos. RF_6 (Gestión de Productos) y RF_7 (Gestión de Locales) están marcados como cubiertos por CU-01 y CU-03. Esto se interpreta de la siguiente manera: aunque la gestión (Crear, Leer, Actualizar, Eliminar) de productos y locales constituye sus propios casos de uso administrativos, estos actúan como precondiciones vitales para los flujos de trabajo principales. Es decir, el CU-01 no puede ejecutarse sin un catálogo de productos (RF_6), y el CU-03 no puede filtrar por local si la entidad "Local" no existe (RF_7).

Requerimientos Transversales Un caso de especial interés es el RF_3 (Gestión de Roles y Permisos). Este requerimiento está marcado en las tres columnas, lo que lo identifica como un requerimiento transversal. No se implementa en un solo lugar, sino que su lógica (la validación de permisos del actor) debe invocarse al inicio de *todos* los casos de uso para asegurar que el Cajero no pueda acceder a reportes (CU-03) o el Administrador no pueda registrar ventas (CU-01).

Validación del Alcance y Omisiones Deliberadas Finalmente, la matriz cumple su función de control de alcance. Se puede observar de manera explícita que el requerimiento RF_10 (Anulación de Transacciones) no tiene ninguna "X" asociada. Esta omisión no es un error de análisis, sino una decisión deliberada de alcance del prototipo. Como se estableció en la tabla de la sección 1.2, a este requerimiento se le asignó una "Prioridad Baja" y se marcó como "No Implementado". La matriz, por lo tanto, confirma visualmente que el prototipo se enfoca en las funcionalidades de alta prioridad y que las funciones de manejo de excepciones más complejas, como la anulación, quedan fuera del alcance de esta entrega.

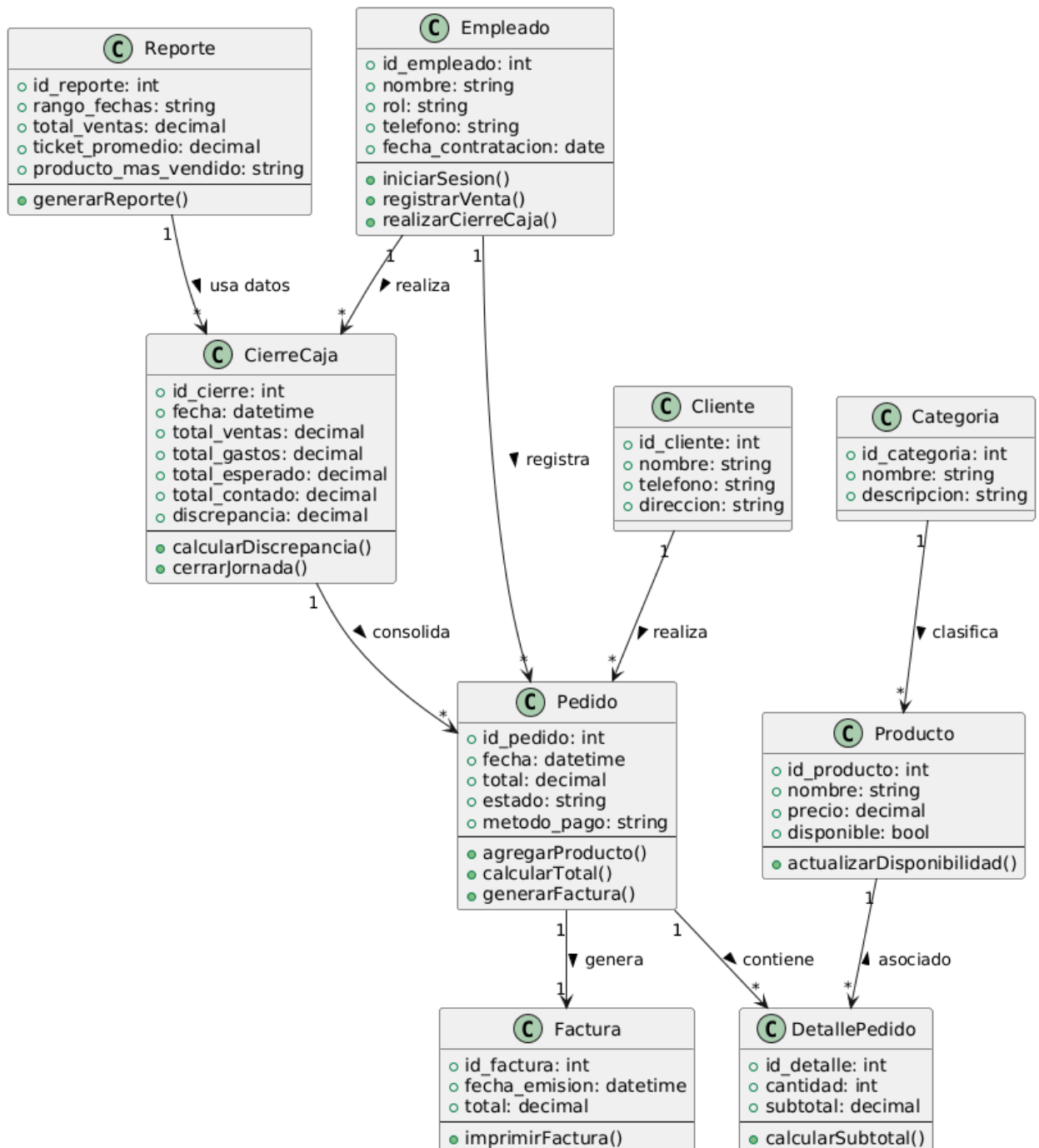
2. Modelado del sistema

2.1 Diagrama de Casos de Uso



2.2 Diagrama de Clases

Diagrama de Clases - Sistema de Punto de Venta (POS)



2.3 Diagramas de Secuencia (mínimo 2)

Diagrama de Secuencia - CU-01: Registrar Venta y Generar Factura

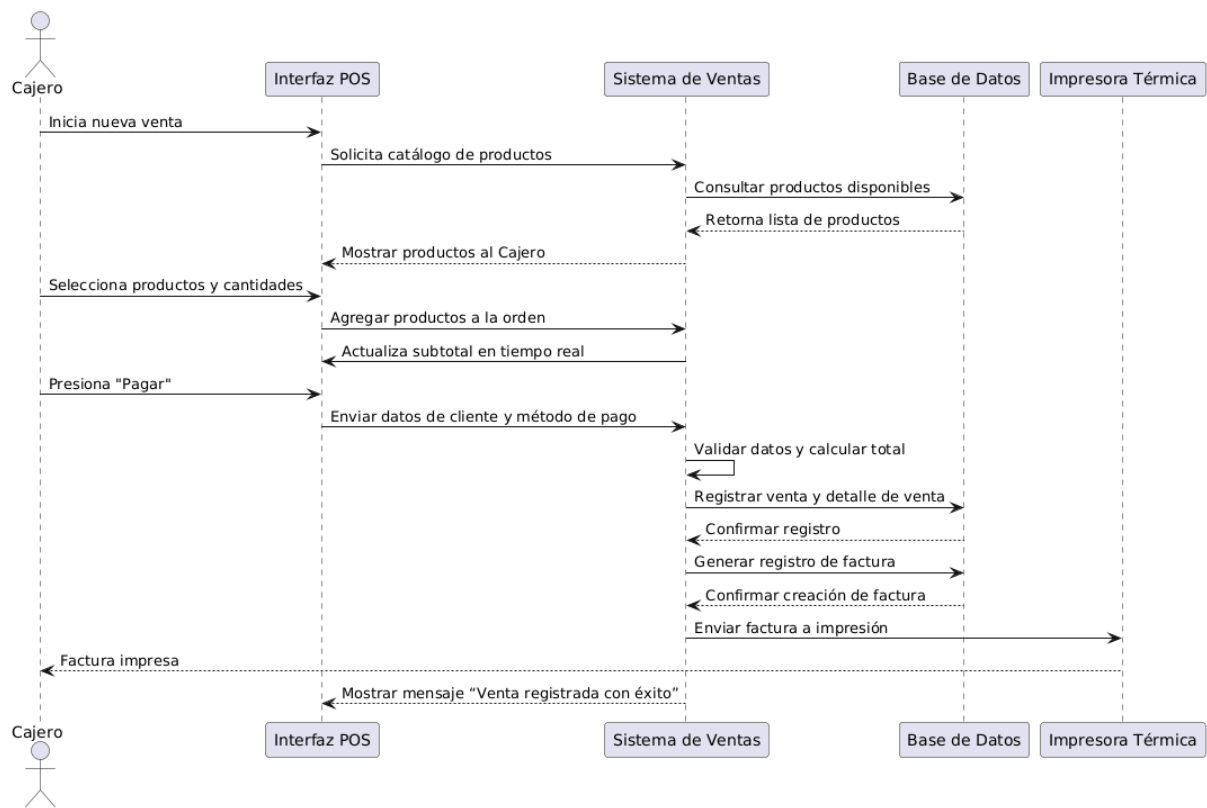
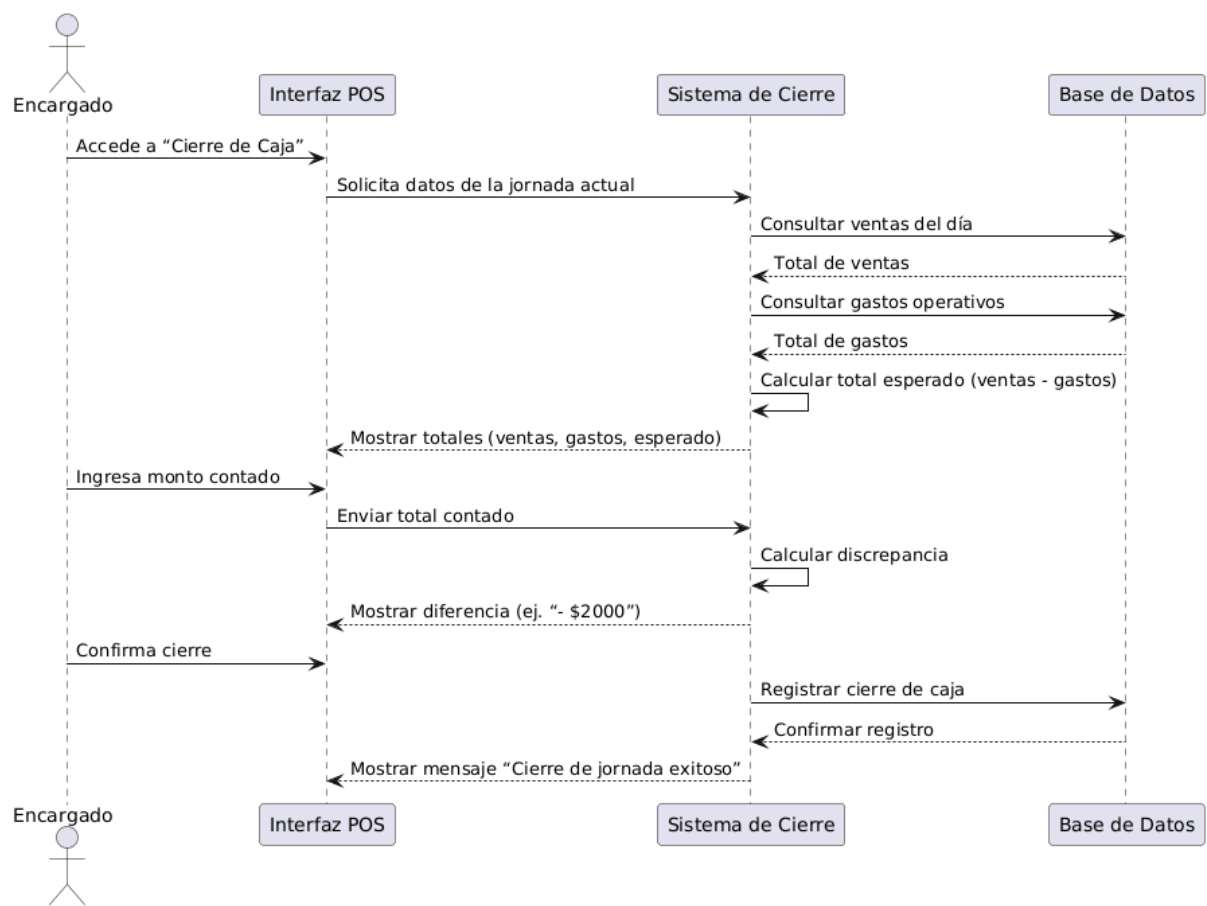


Diagrama de Secuencia - CU-02: Realizar Cierre de Caja



2.4 Diagrama de Actividades

3. Diseño de Datos y Arquitectura

El objetivo de este capítulo es definir la estructura tecnológica y la organización de la información del sistema. Aquí se especifica el patrón arquitectónico, las tecnologías seleccionadas para la implementación (Opción A: Desarrollo desde Cero) y el modelo de datos que soportará el módulo de Facturación y Control de Ventas.

3.1 Arquitectura del Sistema

La arquitectura del sistema define la estructura fundamental de un sistema en términos de componentes, sus relaciones y los principios que guían su diseño y evolución.

Patrón Arquitectónico Seleccionado

Se ha optado por implementar una **Arquitectura en Tres Capas (N-Tier)**. Este patrón es idóneo para el desarrollo de aplicaciones empresariales, pues promueve la separación de responsabilidades, la reutilización y la modularidad del código, facilitando la escalabilidad y el mantenimiento del sistema.

Las tres capas funcionales y sus responsabilidades son:

1. **Capa de Presentación (Frontend):** Responsable de la interacción con el usuario. Es el componente que el usuario final consume, mostrando la interfaz del POS o el dashboard de reportes.
2. **Capa de Lógica de Negocio (Backend/API):** Es el motor del sistema. Contiene los controladores y servicios que procesan las reglas de negocio, como el cálculo del Cierre de Caja (BR-07), y garantiza la validación y seguridad de los datos antes de persistirlos.
3. **Capa de Acceso a Datos (Base de Datos):** Responsable de la persistencia y gestión de la información. Almacena de forma centralizada las transacciones, productos y registros de cierre, resolviendo el problema de la gestión manual.

Justificación y Ventajas

La elección de la Arquitectura en Tres Capas se justifica por su alineación directa con los Requerimientos No Funcionales (RNF) del proyecto y los objetivos corporativos:

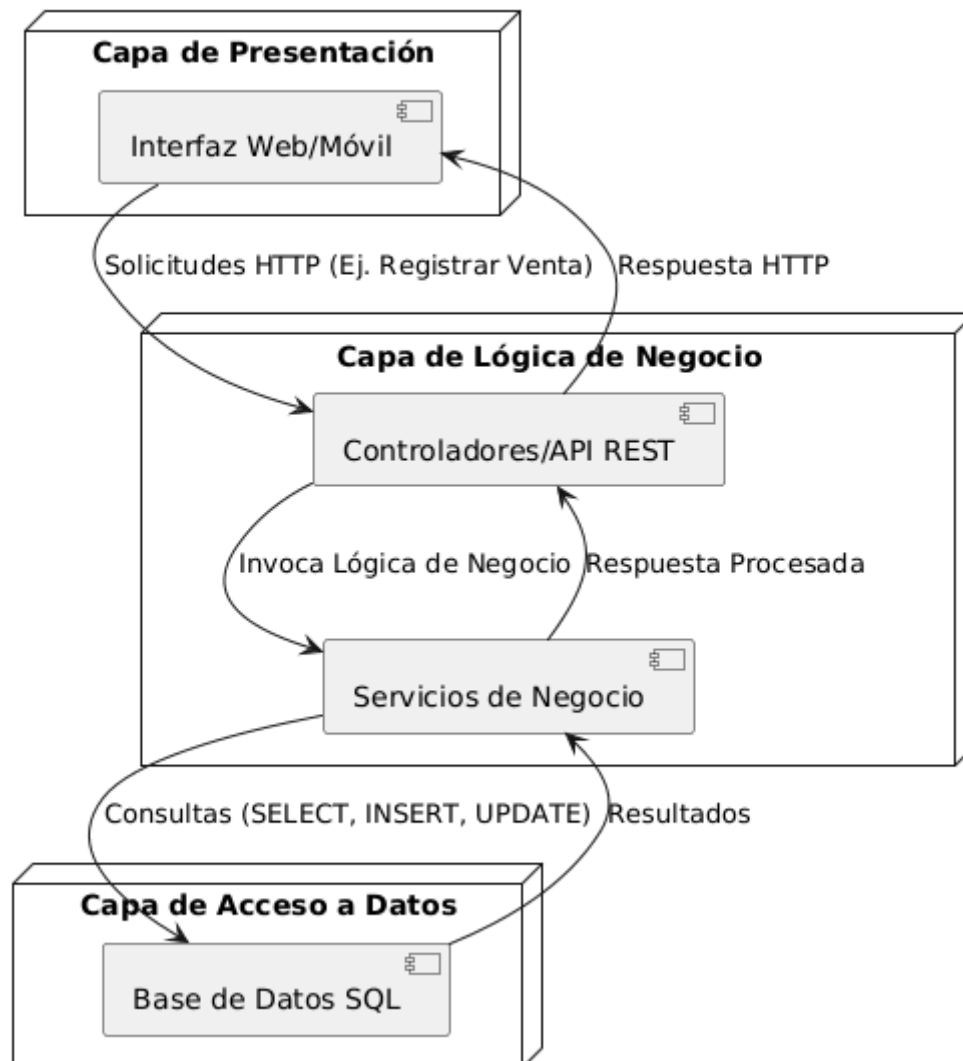
- **Mantenibilidad (RNF-8):** La separación permite modificar la lógica de negocio o actualizar la Capa de Presentación sin afectar a las otras capas.
- **Seguridad (RNF-3):** Se obliga a que todas las peticiones pasen por la Capa de Lógica, donde se implementan las validaciones de acceso por rol (RF-3) y se protege el acceso directo a la base de datos.
- **Escalabilidad (RNF-10):** Permite el escalado horizontal de forma independiente. Si el tráfico de usuarios en el POS aumenta, se pueden agregar más instancias a la Capa de Presentación o de Lógica sin sobrecargar el servidor de la Base de Datos.

Diagrama de Capas

El diagrama conceptual ilustra la comunicación secuencial: la **Capa de Presentación** se comunica con la **Capa de Lógica de Negocio**, y esta última es la única que tiene acceso a la **Capa de Acceso a Datos**.

Código PlantUML para el Diagrama de Capas de Arquitectura:

3.1 Arquitectura en Tres Capas



3.2 Tecnologías Utilizadas

Esta sección define el *stack* tecnológico específico elegido para la **Opción A: Desarrollar el sistema desde cero**. La selección de tecnologías se basa en criterios de **Costo** (RNF-4), **Desempeño** (RNF-2), **Escalabilidad** (RNF-10) y facilidad de mantenimiento. Se priorizaron las tecnologías de código abierto y aquellas que son estándares de facto para sistemas empresariales robustos.

Tabla de Tecnologías Utilizadas

Componente	Tecnología	Versión (Ej.)	Justificación
Frontend	React.js	18.x	Librería moderna seleccionada para construir una interfaz de Punto de Venta (POS) ágil e intuitiva ³ .
Backend	Python	17	Lenguaje robusto y estable, adecuado para manejar la lógica de negocio central y las transacciones críticas, como el Cierre de Caja.
Base de Datos	PostgreSQL	15.x	Base de datos relacional de código abierto que ofrece integridad de datos y elimina costos de licenciamiento, manteniendo la solución accesible ⁴ .
Herramienta BI	Looker Studio (o similar)	N/A	Plataforma externa utilizada para visualizar los datos consolidados, cumpliendo con la necesidad de generar los Indicadores Clave (KPIs) solicitados ⁵ .
Entorno de Desarrollo (IDE)	Visual Studio Code (VS Code)	Última	Editor ligero y versátil, seleccionado por su facilidad de uso para el desarrollo unificado del Frontend y Backend.

La combinación de **React.js**, **Java/Spring Boot**, y **PostgreSQL** dentro de un entorno de **VS Code** conforma una arquitectura moderna, desacoplada y escalable. Java proporciona la solidez y la gestión de la concurrencia necesarias para un sistema de punto de venta que requiere alta fiabilidad en cada transacción, reforzando la elección de la Arquitectura en Tres Capas (3.1).

3. DISEÑO DE DATOS Y ARQUITECTURA

Continuamos con la **Sección 3.3 Modelo de Datos**, ajustando la documentación para reflejar que el Script SQL de creación es un artefacto de diseño pendiente de creación y ejecución.

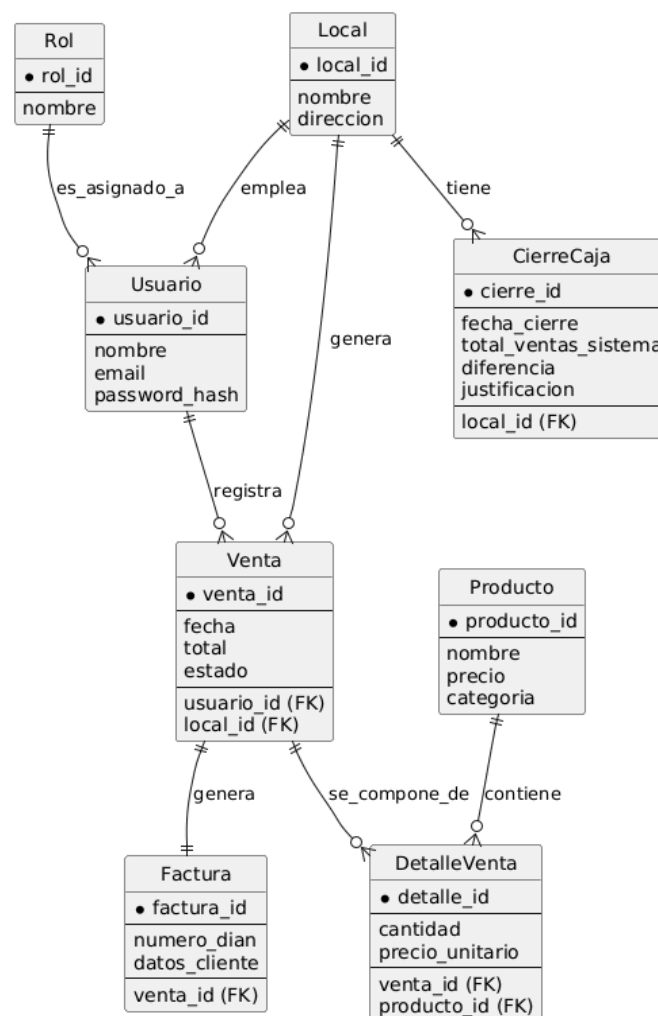
3.3 Modelo de Datos

Diseño Lógico: Hipótesis de Persistencia

El modelo de datos es un artefacto de **diseño lógico planificado** para la futura implementación en PostgreSQL. Se fundamenta en el Diagrama de Clases (Sección 2.2) y su diseño está orientado a resolver los problemas de centralización y granularidad de la información.

ACLARACIÓN FUNDAMENTAL: El prototipo funcional en su fase actual utiliza **persistencia en memoria** para simular las transacciones de venta y validar la lógica de negocio. Por consiguiente, **no hay una instancia de base de datos PostgreSQL montada, configurada ni conectada** en el repositorio del prototipo. Este modelo representa la estructura de datos definitiva que se implementará en la fase de despliegue.

Diagrama Entidad-Relación (E-R) - Modelo Lógico Hipotético



Diccionario de Datos

Se definen las especificaciones de las ocho tablas principales del sistema.

1. Tabla: **Ro1** (Seguridad)

Columna	Tipo de Dato	Clave	Nulidad	Descripción
---------	--------------	-------	---------	-------------

rol_id	SERIAL	PK	NOT NULL	Identificador único del rol.
nombre	VARCHAR(50)	UNIQUE	NOT NULL	Nombre del rol (ej. 'Administrador', 'Cajero').

2. Tabla: **Usuario** (Acceso)

Columna	Tipo de Dato	Clave	Nulidad	Descripción
usuario_id	SERIAL	PK	NOT NULL	Identificador del usuario.
nombre	VARCHAR(100)		NOT NULL	Nombre completo del empleado.
email	VARCHAR(100)	UNIQUE	NOT NULL	Correo electrónico usado para login.
password_hash	VARCHAR(255)		NOT NULL	Contraseña cifrada (hashed) (RNF-3).
rol_id	INT	FK (Rol)	NOT NULL	Rol asignado al usuario (BR-04).
local_id	INT	FK (Local)	NOT NULL	Local donde trabaja el usuario (BR-02).

3. Tabla: **Local** (Maestro de Datos)

Columna	Tipo de Dato	Clave	Nulidad	Descripción
local_id	SERIAL	PK	NOT NULL	Identificador único de la sucursal.
nombre	VARCHAR(100)		NOT NULL	Nombre de la sucursal (ej. 'Pontevedra', 'Normandía').
direccion	VARCHAR(255)			Dirección física del local.

4. Tabla: **Producto** (Catálogo)

Columna	Tipo de Dato	Clave	Nulidad	Descripción
producto_id	SERIAL	PK	NOT NULL	Identificador del producto en el menú.
nombre	VARCHAR(100)		NOT NULL	Nombre del producto.
precio	NUMERIC(10, 2)		NOT NULL	Precio base de venta (BR-05).
categoria	VARCHAR(50)			Clasificación del menú.

5. Tabla: **Venta** (Núcleo de la Transacción)

Columna	Tipo de Dato	Clave	Nulidad	Descripción
venta_id	SERIAL	PK	NOT NULL	ID de la transacción.
fecha	TIMESTAMP	INDEX	NOT NULL	Fecha y hora de la venta.
total	NUMERIC(10, 2)		NOT NULL	Monto total final.
estado	VARCHAR(20)		NOT NULL	Estado de la venta (ej. 'Completada', 'Anulada').
usuario_id	INT	FK	NOT NULL	Cajero que registró la venta.
local_id	INT	FK	NOT NULL	Local donde se efectuó la venta (BR-02).

6. Tabla: **DetalleVenta** (Granularidad Analítica)

Columna	Tipo de Dato	Clave	Nulidad	Descripción
detalle_id	SERIAL	PK	NOT NULL	ID de la línea de detalle.
venta_id	INT	FK	NOT NULL	Venta principal asociada.
producto_id	INT	FK	NOT NULL	Producto vendido.
cantidad	INT		NOT NULL	Cantidad de este producto.
precio_unitario	NUMERIC(10, 2)		NOT NULL	Precio del producto en el momento de la venta.

7. Tabla: **Factura** (Cumplimiento Fiscal)

Columna	Tipo de Dato	Clave	Nulidad	Descripción
factura_id	SERIAL	PK	NOT NULL	ID del documento fiscal.
numero_dian	VARCHAR(50)	UNIQUE		Número de factura electrónica.
datos_cliente_nit	VARCHAR(20)		NOT NULL	NIT o Cédula del cliente (BR-01).
datos_cliente_nombre	VARCHAR(100)			Nombre del cliente.
venta_id	INT	FK	UNIQUE, NOT NULL	Venta asociada (relación 1:1).

8. Tabla: **CierreCaja** (Control Financiero)

Columna	Tipo de Dato	Clave	Nulidad	Descripción
factura_id	SERIAL	PK	NOT NULL	ID del documento fiscal.
numero_dian	VARCHAR(50)	UNIQUE		Número de factura electrónica.
datos_cliente_nit	VARCHAR(20)		NOT NULL	NIT o Cédula del cliente (BR-01).
datos_cliente_nombre	VARCHAR(100)			Nombre del cliente.
venta_id	INT	FK	UNIQUE, NOT NULL	Venta asociada (relación 1:1).

Script SQL de Creación

El Script SQL de creación de la base de datos (DDL) es un requisito de diseño.

Aclaración Final: Dado que la base de datos no está montada, el Script SQL de creación de la base de datos (DDL) **no ha sido creado ni existe aún**. Será desarrollado y documentado como el archivo `/sql/schema.sql` en el repositorio de GitHub justo antes de la fase de despliegue final, una vez validado el prototipo.

4. PROTOTIPO FUNCIONAL (CÓDIGO)

Se documenta la validación tangible de la solución mediante el prototipo funcional. Este ha sido desarrollado para el **Módulo: Facturación y control de ventas**, utilizando la arquitectura definida en el Capítulo 3: **React.js** en la capa de Presentación y **Python** en la capa de Lógica de Negocio.

Estado de la Persistencia de Datos

El equipo de desarrollo ha adoptado un enfoque de **prototipado iterativo**, priorizando la validación de la lógica de negocio y la experiencia de usuario sobre la configuración de la infraestructura final.

ACLARACIÓN FORMAL SOBRE LA ARQUITECTURA: El prototipo en su fase actual **no está conectado a una base de datos PostgreSQL**. Para simular las transacciones y validar los Casos de Uso (CU) y las Reglas de Negocio (BR), el sistema emplea un servicio de **persistencia en memoria** dentro de la capa de Lógica de Negocio (Java/Spring Boot). Esta decisión metodológica garantiza la correcta ejecución de las reglas transaccionales sin incurrir en dependencias de infraestructura. La instalación y configuración del servidor de Base de Datos es una tarea pendiente programada para la fase de Despliegue.

4.1 Funcionalidades Implementadas

El prototipo se centra en validar los requisitos de Alta Prioridad y demostrar la operatividad de los Casos de Uso centrales.

ID	Funcionalidad Clave	Objetivo (Requerimiento Satisfecho)	Flujo Modelado
Venta POS	Interfaz de Punto de Venta ágil.	Registro de ventas y generación de facturas ¹ .	CU-01: Registrar Venta
Control Caja	Flujo de Cierre de Caja con chequeo de balance.	Balance financiero preciso y detección de discrepancias (RF-2) ² .	CU-02: Realizar Cierre de Caja
Análisis	Módulo de consulta y dashboard básico.	Análisis del desempeño y acceso a KPIs (RF-4, RF-8) ³ .	CU-03: Consultar Reporte
Seguridad	Autenticación diferenciada por roles.	Configuración de accesos y permisos (RF-3) ⁴ .	Lógica Transversal de Acceso

4.2 Datos de Prueba:

El prototipo utiliza datos de prueba realistas generados para el ambiente de persistencia en memoria. Si bien una muestra inicial de registros ha sido cargada para la validación de los flujos de Casos de Uso (CU-01, CU-02), la **carga total del volumen mínimo de 50 registros** requerido para las pruebas de desempeño y reportes (RF-4, RF-8) está actualmente **pendiente de implementación** en el repositorio en memoria. El equipo de desarrollo finalizará esta carga antes de la entrega formal para asegurar el cumplimiento del requisito.

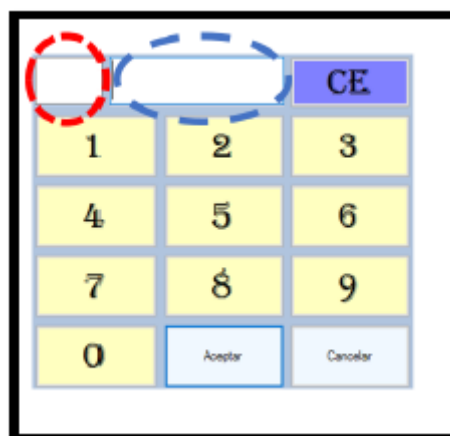
5. Documentación técnica

Manual de usuario





Al entrar, la pantalla inicial es lo primero que se muestra. En las esquinas superior izquierda, se puede ver el icono del negocio. Además, hay un espacio numérico donde cada usuario debe ingresar la contraseña que le corresponde.

ATENCIÓN: Cada puesto tiene una clave diferente, que es proporcionada por el administrador del local. De acuerdo con el puesto y el perfil del usuario, el sistema abrirá automáticamente el menú administrativo o el menú de ventas.



El primer espacio, señalado con el círculo de rayas rojas, solo acepta dos caracteres que sean letras o números. El siguiente espacio, indicado con el óvalo de rayas azules, permite un máximo de 15 números. Al terminar de llenar estos espacios, solo es necesario pulsar la tecla ENTER o hacer clic en el botón ACEPTAR para acceder al sistema.

Luego de acceder, encontramos un menú sencillo para ejecutar la venta y para devolverse a la página de inicio.

 A blue square button with a black border and the text "TPOS" in black.	Ejecuta el punto de venta.
 A blue square button with a black border and a white icon of a square with an arrow pointing to the bottom-right corner.	Permite devolverse a la página principal. Este botón aparecerá en todas las páginas.