Logotipo

Descripción generada automáticamente





Contenido

**[1. Introducción 3](#_heading=h.gjdgxs)**

**[2. Propósito 3](#_heading=h.30j0zll)**

**[3. Alcance 3](#_heading=h.1fob9te)**

**[4. Equipo de Trabajo-Actores del Desarrollo 4](#_heading=h.3znysh7)**

**[5. Recomendaciones de conformidad con esta práctica. 4](#_heading=h.2et92p0)**

**[6. Referencias y estándares aplicables a este documento: 5](#_heading=h.2s8eyo1)**

**[7. Definiciones, acrónimos y abreviaciones. 5](#_heading=h.17dp8vu)**

**[8. Descripción arquitectura: 6](#_heading=h.3rdcrjn)**

[**8.1 VISTAS: 6**](#_heading=h.26in1rg)

[**8.2 TIPOS DE VISTAS: 6**](#_heading=h.lnxbz9)

[**8.3 Framework Conceptual 7**](#_heading=h.35nkun2)

[**8.3.1 Descripción de la arquitectura en contexto 7**](#_heading=h.1ksv4uv)

[**8.3.1 Actores Usuarios y sus roles: 8**](#_heading=h.44sinio)

[**8.3.2 Actividades de arquitectura en el ciclo de vida 9**](#_heading=h.2jxsxqh)

[**8.3.3 Descripciones prácticas de arquitectura. 9**](#_heading=h.z337ya)

[**8.4 Documentación de la arquitectura 10**](#_heading=h.3j2qqm3)

[**8.5 Selección de puntos de vista de la arquitectura 10**](#_heading=h.1y810tw)

[**8.6 Vistas de la arquitectura 11**](#_heading=h.4i7ojhp)

[**UML-VISTA DE ESCENARIO 11**](#_heading=h.2xcytpi)

[**UML-VISTA LÓGICA-DIAGRAMA DE CLASES 12**](#_heading=h.1ci93xb)

[**UML-VISTA LÓGICA-DIAGRAMA DE BASE DE DATOS 14**](#_heading=h.qsh70q)

[**UML-VISTA DE PROCESO-DIAGRAMAS DE ACTIVIDAD 16**](#_heading=h.1pxezwc)

[**UML-VISTA DE DESPLIEGUE 18**](#_heading=h.49x2ik5)

[**UML-VISTA FÍSICA – DIAGRAMA INFRAESTRUCTURA 19**](#_heading=h.2p2csry)

[**Usabilidad y Calidad de software- Front-end -Vistas Principales 21**](#_heading=h.147n2zr)

[**Usabilidad y Calidad de software – Vistas específicas de usuario 22**](#_heading=h.3o7alnk)

[**8.7 Consistencia en la cantidad de vistas de la arquitectura. 23**](#_heading=h.23ckvvd)

[**8.7.1 DESCRIPCION DE MODULOS 23**](#_heading=h.ihv636)

[**8.7.2 DESCRIPCIÓN DE COMPONENTES 23**](#_heading=h.32hioqz)

[**8.7.3 DESCRIPCIÓN DE CONECTORES 23**](#_heading=h.1hmsyys)

[**9.1 Acceso a fuentes de información del proyecto 25**](#_heading=h.41mghml)

**Introducción**

El presente informe tiene como objetivo describir en detalle la arquitectura de SalvaPOS, un sistema integral de cajas diseñado para optimizar y automatizar la gestión de ventas, inventarios en pequeños y medianos establecimientos comerciales. SalvaPOS se posiciona como una solución eficiente y accesible para farmacias que buscan acelerar sus procesos operativos, y obtener visibilidad completa sobre sus ventas y el estado de sus inventarios en tiempo real.

Este sistema permitirá a los farmacéuticos realizar las operaciones necesarias relacionadas a la venta de un producto de manera rápida. También permite que los clientes puedan ver el catálogo de productos que están con stock incluso antes de visitar la tienda de manera presencial. Y permitirá a los administradores de sus farmacias ver estadísticas y reportes relacionadas a su negocio.

# **Propósito**

El propósito de este documento es proporcionar una guía técnica exhaustiva para el diseño, implementación y despliegue de SalvaPOS, un sistema de punto de venta (POS) orientado a mejorar la gestión de ventas, inventarios y procesos de pago en establecimientos comerciales. El documento tiene como objetivo detallar las decisiones arquitectónicas y técnicas que permitirán que SalvaPOS sea una solución robusta, escalable y adaptable a diversas necesidades comerciales.

Este documento es crucial para los desarrolladores, arquitectos de software, y stakeholders del proyecto, ya que ofrece una visión clara y estructurada de los componentes clave del sistema, incluyendo la lógica de negocio, la interacción entre los diferentes módulos, y los puntos de integración con otros sistemas externos. A través de este documento, se establecen las bases para el desarrollo del sistema bajo metodologías ágiles como Scrum, con entregas iterativas y mejoras continuas.

Además, se describen los aspectos críticos de seguridad, rendimiento, y usabilidad del sistema, asegurando que SalvaPOS no solo satisfaga las necesidades actuales de los usuarios, sino que también esté preparado para futuras expansiones.

# **Alcance**

Los alcances del proyecto se concentran en cinco puntos diferentes, estos son:

* A-Control de inventario
  + La aplicación permite poder editar la información del inventario respecto al stock de un producto, es decir, el farmacéutico al momento de hacer constancia en el inventario de manera física podrá buscar el producto en el sistema y editar su disponibilidad.
  + La aplicación permite a los farmacéuticos registrar productos nuevos.
  + Mostrar el listado de productos por medio de una vista (Vista inventario / Productos).
  + La aplicación permite a los farmacéuticos registrar nuevas categorías.
  + Mostrar el listado de categorías por medio de una vista (Vista inventario / categorías.)
* B-Gestión de ventas.
  + La venta se realizará por medio de la vista principal Ventas.
  + El sistema trabaja con cuatro métodos de pago:
    - Débito.
    - Crédito.
    - Efectivo.
    - Mixto (Efectivo y Debito).
  + Los clientes pueden acceder a la página de la farmacia y ver sus productos, pero, no podrán comprar de manera digital, ya que, se trata de un sistema de caja.
  + Al momento de realizarse alguna actualización dentro del stock de un producto o la categoría, la aplicación notificará por correo un listado de los elementos modificados.
  + La aplicación permite hacer devoluciones.
* D-Sistema de Reportes.
  + Este sistema permitirá a los administradores acceder a un dashboard que le permite recibir información relevante para el negocio por medio de una vista (Vista Reporte).
  + Puede filtrar por la categoría del producto.
  + Mostrar cantidad de ejemplares vendidos por producto.
  + Mostrar el pódium de productos más vendidos.
  + Mostrar las categorías disponibles y las que fueron dadas de bajas.
  + El sistema no debe ser visible tanto para el farmacéutico como para el cliente.
* E-Sistema de seguridad y usuarios.
  + Los clientes no deben poder acceder a otras secciones de la aplicación que no sean la que le corresponde (Home).
  + El administrador podrá acceder a todo el contenido y funcionalidades que otorga la aplicación.
    - Home.
    - Inventario.
    - Reportes.
    - Registro.
    - Inicio de sesión.
  + El administrador es el encargado de crear la cuenta con la que el farmacéutico opera dentro de la aplicación.
  + Los farmacéuticos tienen acceso a las siguientes vistas:
    - Home.
    - Inventario.
    - Iniciar Sesión.

# **Equipo de Trabajo-Actores del Desarrollo**

El desarrollo del sistema requiere la participación de un equipo multidisciplinario con roles claramente definidos para asegurar que el proyecto se lleve a cabo de manera eficiente. A continuación, se presenta una tabla con los principales roles y sus responsabilidades:

| Rol | Responsabilidades |
| --- | --- |
| Cliente-Stakeholder | Christian Lazcano |
| Product Owner | Christian Lazcano |
| Desarrolladores Back-end | Pablo Ávila – Matias Pérez |
| Desarrolladores Front-end | Richard Araneda – Matias Perez |
| DBA-Especialista en Base de Datos | Pablo Ávila |
| Tester/QA | Matias Perez |
| Analista de Requisitos | Richard Araneda |
| Administrador Plataformas | Pablo Ávila |

# **Recomendaciones de conformidad con esta práctica.**

# Para asegurar que la arquitectura del sistema cumpla con los requisitos y estándares establecidos, se recomiendan las siguientes prácticas:

# Implementar revisiones de arquitectura cada 15 días para asegurar que se ajusten a los cambios de requisitos.

# Utilizar patrones de diseño desacoplados como MVC para modularidad y escalabilidad o por capas.

# Adoptar metodologías ágiles como Scrum para la entrega continua de mejoras y funcionalidades.

# **Referencias y estándares aplicables a este documento:**

El documento sigue los siguientes estándares de calidad para asegurar la robustez y escalabilidad del sistema:

* **IEEE 830-1998 ST**: Estandarización de requisitos de software.
* **ISO 9126-2001**: Métricas de calidad del software.
* **Modelo 4+1 de Kruchten**: Modelo estructural basado en múltiples vistas de escenarios.
* **ATAM (Architecture Tradeoff Analysis Method)**: Metodología para evaluar y documentar decisiones arquitectónicas.

# **Definiciones, acrónimos y abreviaciones.**

| Término | Definición |
| --- | --- |
| DAS | Documento de Arquitectura de Software |
| MVC | Modelo-Vista-Controlador, patrón de diseño utilizado para separar la lógica de negocio de la interfaz de usuario |
| UML | Unified Modeling Language, lenguaje gráfico para representar la arquitectura del sistema |
| CRUD | Operaciones básicas de Create, Read, Update, Delete para la gestión de datos. |

# **DESCRIPCIÓN DE ARQUITECTURA:** La arquitectura del sistema se presenta mediante el uso de vistas basadas en el Modelo 4+1 de Kruchten, con el fin de proporcionar una visión completa y estructurada del sistema desde diferentes perspectivas. La arquitectura del sistema se presenta mediante el uso de vistas basadas en el Modelo 4+1 de Kruchten, con el fin de proporcionar una visión completa y estructurada del sistema desde diferentes perspectivas.

## **VISTAS:**

Cada vista se representará en un diagrama UML para describir la estructura y comportamiento del sistema.

### **TIPOS DE VISTAS:**

1. **Vista de Escenarios (Diagrama de Caso de Uso)**: Describe las funcionalidades del sistema y cómo interactúan los diferentes actores con él.
2. **Vista Lógica (Diagrama de Clases)**: Detalla la estructura de datos y las relaciones entre las principales entidades del sistema.
3. **Vista de Procesos (Diagrama de Actividad)**: Muestra el flujo de actividades dentro del sistema y cómo se coordinan los diferentes procesos.
4. **Vista de Despliegue (Diagrama de Componentes)**: Representa la distribución de componentes en la infraestructura física.
5. **Vista Física (Diagrama Topológico)**: Representa la infraestructura de red y la comunicación entre los diferentes servicios del sistema

## **Framework Conceptual**

### **Descripción de la arquitectura en contexto**

El sistema será diseñado y desarrollado utilizando un enfoque ágil con metodologías como Scrum, lo que permitirá la entrega incremental de funcionalidades, la retroalimentación constante y la capacidad de adaptarse a cambios en los requisitos a lo largo del ciclo de vida del proyecto. Esto asegurará que se puedan abordar necesidades emergentes y maximizar el valor del producto para los usuarios.

El diseño del sistema se basará en la separación de vistas, que ofrecerán diferentes perspectivas de la arquitectura para una comprensión integral. A continuación, se describen las principales vistas:

* **Vista de Escenarios**: Representada mediante diagramas de Casos de Uso, que describen las interacciones del usuario con el sistema y los flujos principales.
* **Vista Lógica**: Incluye el Modelo de Clases, que muestra las entidades y relaciones del sistema, así como el Modelo de Datos que detalla la estructura de la base de datos.
* **Vista de Procesos**: Utiliza Diagramas de Actividad para ilustrar los flujos internos del sistema y las interacciones entre los distintos módulos.
* **Vista de Despliegue**: Se implementará a través de Diagramas de Componentes o Modelos de Capas para reflejar la distribución de la lógica de negocio, la interfaz de usuario y la capa de acceso a datos.
* **Vista Física**: Utiliza Diagramas Topológicos o de Servicios para mostrar la distribución del sistema en el entorno físico, como servidores, redes y otros componentes de infraestructura.

Cada una de estas vistas está interconectada para garantizar la coherencia y la alineación con los requisitos de negocio. No hay ninguna vista separada de una implementación específica, sino que todas contribuyen a una visión holística de la arquitectura del sistema descrita en este documento.

Estas vistas se han creado utilizando el Lenguaje de Modelado Unificado (UML) en su versión 2.0 y han sido desarrolladas empleando herramientas de modelado estándar como **StarUML** y Lucidchart y en front-end **Canva, Miro, Figma u otro** para facilitar la creación de diagramas visuales que respalden la documentación arquitectónica.

El estilo arquitectónico se centra en una combinación de patrones de diseño orientados a servicios y arquitectura por capas, en donde cada módulo del sistema se corresponde con un conjunto específico de casos de uso que resuelven los procesos definidos para la operación.

Estos procesos corresponden a:

* **Gestión del inventario:** Corresponde al control preciso de la disponibilidad de los Productos.
* **Sistema de reportes**: Módulo de reportes que genere información para el negocio a través de los datos obtenidos de las ventas y el inventario.
* **Gestión de usuarios:** Corresponde a la capacidad de poder permitir o no quienes serán los usuarios que tendrán acceso a las diferentes funcionalidades. También abarca por donde deben acceder y donde se crean los usuarios .
* **Sistema de seguridad:** Corresponde al control del acceso a la información según roles.
* **Gestión de ventas:** Corresponde al proceso en que los clientes escogen sus productos y el farmacéutico procede a inicializar la venta.

### **Actores Usuarios y sus roles:**

Este documento representa la identificación de Actores/Usuarios Stakeholders y sus roles a partir de la interpretación de los casos de uso del Negocio asociados.

| ACTOR | ROL / CARGO | FUNCION |
| --- | --- | --- |
| Cliente. | Cliente de la farmacia. | Consultar productos, comprar productos y solicitar devolución de ser necesaria. |
| Farmacéutico. | Trabajador de la farmacia. | Atender a los clientes, realizar ventas, control de inventario y realizar devoluciones. |
| Administrador. | Administrador. Encargado o dueño del negocio. | Gestionar disponibilidad de categorías, usuarios. Tomar decisiones respecto al negocio en base a las estadísticas proyectadas en la vista reporte. |

### **8.3.2 Actividades de arquitectura en el ciclo de vida**

El desarrollo de la arquitectura de la aplicación web SalvaPOS se abordó utilizando un ciclo de vida **iterativo incremental** basado en **Scrum**. Esta metodología fue seleccionada porque permite realizar entregas parciales y frecuentes de software funcional, facilitando la validación temprana de los requisitos por parte de los Stakeholders y el Product Owner con la adaptación a cambios en el entorno de negocio.

El ciclo de vida se organizó en sprints de 2 a 3 semanas, con cada iteración centrada en la construcción y mejora de un módulo específico del sistema (por ejemplo, el módulo de gestión de gastos comunes o el módulo de reservas). Al final de cada sprint, se realiza una revisión detallada de los resultados para asegurar la alineación con las expectativas del cliente y se priorizan las tareas para el siguiente sprint.

### **8.3.3 Descripciones prácticas de arquitectura.**

Durante el proceso de desarrollo, la arquitectura del sistema fue escalada desde un **escenario principal de caso de uso** hasta un diseño modular a través de los siguientes pasos:

1. **Análisis de Requerimientos**: Se analizaron los requisitos iniciales del sistema para identificar los casos de uso principales, tales como:
   1. **Ver lista de productos**.
   2. **Crear Productos**.
   3. **Crear Categorías**.
   4. **Realizar Venta.**
2. **Definición de la Vista Lógica**: A partir del escenario principal, se identificaron las entidades y relaciones claves en un diagrama de clases UML, representando el núcleo del sistema.
3. **Modelo de Datos**: Se desarrolló un diagrama de base de datos para definir la estructura de la información, manteniendo la coherencia con la vista lógica.
4. **Vista de Procesos**: Se modelaron los flujos de las actividades y transacciones principales del sistema mediante diagramas de actividad.
5. **Despliegue**: Se definió la infraestructura física y lógica para alojar el sistema en la nube, detallando los servicios necesarios, como balanceadores de carga, servidores de aplicaciones y bases de datos distribuidas.
6. **Definición de Planes de pruebas y técnicas de pruebas testing**.

## **Documentación de la arquitectura**

* **ISO/IEC 25010**: Se utilizó para evaluar la calidad del sistema en términos de funcionalidad, seguridad y eficiencia.
* **IEEE 1471-2000**: Proporcionó la guía para la representación de vistas arquitectónicas y la identificación de stakeholders.
* **Modelo 4+1 de Kruchten**: Estructura la arquitectura en vistas lógicas, de desarrollo, de procesos y física, facilitando la visualización desde múltiples perspectivas.
* **Planes de prueba**: (Declarar los tipos de pruebas a realizar)

## **Selección de puntos de vista de la arquitectura**

| VISTAS | UML | Cantidad |
| --- | --- | --- |
| Escenario Principal | Diagrama de Caso de uso | 1 (2-3 los necesarios) |
| Vista Lógica | Diagrama de Clases  MER | 1  1 |
| Vista de Proceso | Diagrama de Actividad:  Nombre 1.  Nombre 2  Nombre 3. | 1 (2-3, los necesarios) |
| Vista Física | Diagrama Topológico de Servicios | 1 |
| Vista de Despliegue | Diagrama de componentes  Diagrama de Capas (MVC) | 1  1 |

## Vistas de la arquitectura

Ejemplos:

| UML-VISTA DE ESCENARIO |
| --- |
| Diagrama de Caso de usos General Extendido. |
| Diagrama |

**Gestión de Inventario Farmacéuticos:**Permite al administrador gestionar el inventario de productos farmacéuticos en la farmacia. Incluye la visualización de productos, la modificación de precios y detalles, y la actualización de inventarios, como añadir o eliminar productos según la necesidad.

* **Ver Inventario:**Muestra todos los productos disponibles en el sistema, permitiendo a los administradores y farmacéuticos visualizar detalles importantes como cantidades en stock, precios y fechas de ingreso.
* **Modificar Inventario:**Incluye funcionalidades para agregar nuevos productos, modificar detalles existentes o eliminar productos obsoletos.

**Ingresar:** Añadir nuevos productos al inventario.

**Modificar:** Actualizar los detalles del producto como precio o cantidad.

**Eliminar:** Retirar productos del inventario que ya no están disponibles o que fueron descontinuados.

**Dar baja categoría:** Poder dar de baja una categoría y que no se muestren en el catálogo.

**Gestión de Ventas y Pagos:**Permite a los farmacéuticos y cajeros procesar ventas de productos, asegurando un flujo de trabajo eficiente al momento de atender a los clientes en la farmacia.

* **Seleccionar Producto:**Los cajeros o farmacéuticos seleccionan los productos que el cliente desea comprar antes de proceder con el pago.
* **Realizar Compra:**Gestiona la transacción completa de la venta, incluyendo la selección del método de pago y la generación del comprobante de compra.
  + - **Procesa la Forma de Pago:**  
      Permite al cliente elegir su método de pago preferido (efectivo, tarjeta o mixto) para completar la compra.
    - **Efectivo:** Procesa el pago con dinero en efectivo.
    - **Tarjeta:** Procesa el pago utilizando tarjetas de crédito o débito.
    - **Pago Mixto:** Combina diferentes métodos de pago si el cliente decide pagar parcialmente con efectivo y el resto con tarjeta**.**
    - **Recibir Comprobante:**Genera un recibo de compra que el cliente puede llevarse como confirmación de la transacción completada.

**Consultas de Productos y Disponibilidad:**Facilita a los clientes y farmacéuticos revisar el inventario para asegurarse de que los productos requeridos están disponibles.

* **Revisar si Existe Tal Medicamento:**Permite al farmacéutico verificar si un medicamento específico está disponible en la farmacia. Si el producto no está en stock, el sistema puede sugerir alternativas o proporcionar enlaces para buscar en otras farmacias.

**Gestión Administrativa:**Facilita al administrador la gestión de usuarios y otras tareas administrativas dentro del sistema.

* **Administrar Factura:**Permite gestionar la facturación y visualizar reportes de ventas. Incluye la opción de ver la facturación por fechas específicas (día, mes, año).
* **Administrar Inventario Farmacéuticos:**Permite al administrador tener control total sobre la gestión del inventario, incluyendo la visualización, modificación y actualización de los productos.
* **Por Fecha:**Organiza la facturación y reportes de acuerdo con las necesidades específicas del administrador, permitiendo visualizar datos por día, mes o año.

| UML-VISTA LÓGICA-DIAGRAMA DE CLASES |
| --- |
| [Diagrama de Clases] |
|  |

El **Diagrama de Clases** para el sistema de gestión de condominios muestra las clases involucradas, sus atributos, métodos y las relaciones entre ellas. Este diagrama ayuda a modelar la estructura y el comportamiento del sistema de una manera orientada a objetos.

#### **2. Elementos Principales**

* **Clases**:
  + **Cliente**
    - **Atributos**: id\_cliente, nombre, apellido, teléfono, dirección, y email.
    - **Métodos**: iniciarSesion(), consultarStock().
  + **Producto**:
    - **Atributos**: id producto, precio\_compra, precio\_venta, categoría, image, description..
    - **Métodos**: crearProducto(), editarProducto()
  + **Categoría:**
    - **Atributos**: id\_categoria, nombre\_categoria, status.
    - **Métodos**: obtenerUsuarios(), agregarGastoComún(), notificarEstado()
  + **Farmacéutico**:
    - **Atributos**: id\_farmaceutico, nombre, apellido, cargo, teléfono, email, username.
    - **Métodos**: iniciarSesion(), generarVenta(), generarDevolucion().
  + **Administrador**:
    - **Atributos**: id\_admin, nombre, apellido, dirección, teléfono, email.
    - **Métodos**: iniciarSesion(), crearUsuario(), deshabilitarUsuario(), editarEstadoCategoria()
  + **Venta**:
    - **Atributos**: id\_venta, fecha, precio\_total, descripción.
    - **Métodos**: generarDetalleVenta(), generarReporteVentas().
  + **Reporte**:
    - **Atributos**: id\_reporte, fecha.
    - **Métodos**: podiumProductos(), filtrarCategoria(), menosVendidos()

#### **3. Relaciones:**

* **Cliente** tiene una relación de asociación con **Venta** (Un cliente puede realizar múltiples compras).
* **Farmacéutico** tiene una relación de asociación con **Venta** (Un farmacéutico puede realizar múltiples ventas).
* **Reporte** tiene una relación de **composición** con **Venta** (Un reporte se compone de múltiples ventas).
* **Administrado** tiene una relación de asociación con **Reporte** (Un administrador puede ver múltiples reportes.)
* **Producto** tiene una relación de **composición** con **Categoría** (Un producto se compone de una categoría).
* **Historial de Actividades** tiene una relación de **asociación** con **Usuario** (registrar acciones del usuario).
* **Venta** tiene una relación de **composición** con **Producto** (Una Venta se Compone de múltiples productos).

| UML-VISTA LÓGICA-DIAGRAMA DE BASE DE DATOS |
| --- |
| SalvaPOS-BBDD. |
|  |

El **Modelo Entidad-Relación (MER)** para el sistema de gestión de condominios representa las entidades involucradas, sus atributos y las relaciones que existen entre ellas. Este modelo permite visualizar cómo se organiza la información relacionada con la administración del condominio.

#### **2. Elementos Principales**

* **Entidades**:
  + **DetalleVenta:**
    - **Atributos**: Id\_detalle(PK), id\_venta(FK), id\_producto(FK), cantidad, precio\_unitario, subtotal, metodoPago.
  + **Venta:**
    - **Atributos**: id\_venta(PK), fecha, total, id\_cliente(FK), id\_empleado(FK).
  + **Cliente:**
    - **Atributos**: id\_cliente(PK), nombre, apellido, teléfono, dirección, email.
  + **Empleado:**
    - **Atributos**: id\_empleado(PK), nombre, apellido, cargo, teléfono, email.
  + **Producto:**
    - **Atributos**: id\_producto(PK), nombre, descripción, precio, cantidad-stock, fecha\_vencimiento, id\_categoria.(FK)
  + **Categoría:** 
    - **Atributos**: id\_categoria(PK), nombre\_categoria.
* **Relaciones**:
  + **Farmacéutico** puede estar asociado en múltiples boletas y a un solo usuario dentro del sistema.
  + **DetalleVenta** puede estar asociado a un solo cliente y a una sola venta. Un mismo cliente y un farmacéutico pueden estar en muchas boletas, pero, nunca dos clientes y dos farmacéuticos en una misma boleta.
  + **Venta** está asociada a un cliente, un farmacéutico y una boleta.
  + **Producto** está asociada a una categoría, un stock, un precio. Puede haber más de un ejemplar dentro de la misma boleta, el mismo producto puede estar en múltiples boletas y ventas.
  + **Categoría** está asociada a muchos productos.

| UML-VISTA DE PROCESO-DIAGRAMAS DE ACTIVIDAD |
| --- |
| [Diagramas de Actividad] |
| Diagrama, Esquemático |

**Diagrama de Actividad** representa el flujo de actividades que se llevan a cabo en el proceso de realizar una compra en SalvaPOS. Este diagrama es útil para visualizar los pasos secuenciales y las decisiones que los diferentes actores (cajero/farmacéutico, sistema, cliente) deben tomar al realizar un proceso de pago o gestionar transacciones.

**Componentes**:

1. **Inicio**: El **cajero/farmacéutico** inicia sesión en el sistema SalvaPOS
2. **Gestión de Usuarios (Opcional para Super Administrador)**: Si el usuario es un **Super Administrador**, tiene acceso para gestionar usuarios y registrar nuevos cajeros o farmacéuticos.
3. **Recuperación de contraseña (Opcional)**: Si el cajero ha olvidado su contraseña, puede iniciar el proceso de recuperación, el sistema envía un enlace de recuperación al correo proporcionado.
4. **Selección de Productos**: El cajero/farmacéutico selecciona los productos que el cliente desea comprar.
5. **Realizar Pago**:

El **cajero/farmacéutico** pregunta al cliente por su método de pago preferido:

* **Pago en Efectivo**: El sistema registra el pago directamente en efectivo.
* **Pago con Tarjeta**: El sistema procesa el pago a través de un método de tarjeta.
* **Pago Mixto**: El cliente puede elegir pagar con una combinación de efectivo y tarjeta.

1. **Verificación de Fondos**:

El sistema verifica si los fondos proporcionados son suficientes para completar la compra.

* **Fondos Suficientes**: Se procede a registrar la venta.
* **Fondos Insuficientes**: El sistema muestra un mensaje de error y la transacción se cancela

1. **Generación de Comprobante**: Si la compra se completa exitosamente, el sistema genera un comprobante o voucher que se entrega al cliente.

El cliente recibe el producto y el comprobante antes de retirarse.

**Proceso para Compras en Línea:**

Si la compra es realizada en línea, el cliente recibe un comprobante que indica que el producto está listo para ser recogido en la farmacia.

El cliente se dirige a la farmacia para recoger el producto.

**Revisión de Productos en Stock (Opcional)**:

* Si el cliente está interesado en revisar otros productos, puede acceder al catálogo dentro de la vista principal, acá puede buscar los productos que son de su interés y ver su disponibilidad.
* Si un producto está disponible, pero, no en todas las sucursales, el sistema mostrará la sucursal en donde se encuentra disponible.

**Fin**: El proceso finaliza cuando el cliente ha recibido su producto o si decide retirarse sin comprar debido a fondos insuficientes o la falta de productos en stock.

| UML-VISTA DE DESPLIEGUE |
| --- |
| Sistema de Gestión de Condominios |
| Diagrama, Dibujo de ingeniería |

**Servidor de Aplicaciones**:

**SalvaPOS App**: Proporciona una interfaz amigable para todos los actores del sistema (superadministrador, cajeros/farmacéuticos, clientes). Permite gestionar ventas, inventario, usuarios, y generar reportes. Esta aplicación facilita la consulta y gestión de productos, el procesamiento de transacciones, y el acceso a funcionalidades administrativas para mantener el sistema operativo.

**Servidor de Base de Datos**:

* **Base de Datos de SalvaPOS**: Almacena toda la información crítica del sistema, incluyendo detalles de inventario, transacciones, usuarios y reportes. Este servidor es fundamental para que la aplicación funcione correctamente, ya que procesa consultas y actualizaciones de datos en tiempo real. Se organiza en distintas secciones:
* **Inventario**: Registra todos los productos disponibles y su cantidad en stock.
* **Transacciones**: Guarda los registros de todas las ventas realizadas, asegurando un seguimiento completo de cada operación.
* **Usuarios**: Contiene la información de los usuarios registrados, incluyendo permisos y roles.
* **Reportes**: Gestiona la generación de reportes de inventario y ventas realizadas para ayudar en el análisis y la toma de decisiones.

**Servidor de Correo**:

* **Sistema de Notificación**: Se encarga de enviar correos electrónicos automáticos a los usuarios. Esto incluye notificaciones de recuperación de contraseñas, recibos de compras y confirmaciones de pedidos en línea. Este sistema garantiza que los usuarios estén informados de sus transacciones y estatus dentro del sistema.

**Servidor de Autenticación**:

* **Autenticación y Recuperación de Contraseña**: Gestiona la autenticación segura de los usuarios al sistema, permitiendo el acceso solo a usuarios autorizados. También facilita la recuperación de contraseñas, enviando enlaces de restablecimiento a través del sistema de correo, asegurando que los usuarios puedan recuperar el acceso de manera rápida y segura.

| UML-VISTA FÍSICA – DIAGRAMA INFRAESTRUCTURA |
| --- |
| Sistema Gestión De SalvaPos |
| Diagrama |

**Centro de Datos Principal**: Es el núcleo del sistema, donde se alojan los principales componentes de la infraestructura. Incluye servidores dedicados para la aplicación web, la base de datos, sistemas de correo y de backup. Todos estos elementos están protegidos por un firewall para garantizar la seguridad y privacidad de la información.

**Router Principal:** Gestiona la conexión y distribución del tráfico de red entre los servidores y otros dispositivos de la infraestructura del centro de datos.

**Firewall de Seguridad:** Protege el sistema contra accesos no autorizados, amenazas y ataques externos, asegurando que solo el tráfico permitido pueda acceder a los servicios del sistema.

**Servicio de Aplicaciones:** Ejecuta la aplicación principal de SalvaPOS, gestionando las operaciones de ventas, inventario, usuarios, y más. Se conecta directamente con la base de datos para almacenar y recuperar información.

**Servidor de Correo**: Responsable de manejar la comunicación y notificaciones enviadas a los usuarios del sistema, incluyendo correos de recuperación de contraseñas y recibos de compras.

**Servidor de BDD (Base de Datos)**: Almacena datos cruciales como inventario, transacciones, usuarios, y reportes, asegurando el acceso rápido y seguro a la información.

**Servidor de Backup**: Realiza copias de seguridad regulares de la base de datos y otros datos importantes para prevenir pérdidas de información en caso de fallas.

**Red del Comercio**:  
Esta red incluye todos los dispositivos de acceso dentro de la tienda o farmacia, como las estaciones de trabajo de los cajeros, la red WiFi para clientes, cámaras de seguridad y el sistema de control de acceso. Se conecta al centro de datos a través del router principal para una comunicación directa y eficiente.

**Estación de Trabajo: Cajero**: Terminales utilizadas por los cajeros para procesar ventas, gestionar inventario, y realizar otras operaciones diarias dentro de SalvaPOS.

**Red WiFi Cliente**: Permite a los clientes conectarse a la red local para acceder a servicios adicionales como verificar disponibilidad de productos o realizar compras en línea.

**Cámara de Seguridad**: Monitorea la seguridad en las instalaciones, permitiendo la vigilancia de las áreas clave de la tienda.

**Sistema de Control de Acceso**: Gestiona la entrada y salida de personal y clientes en zonas restringidas de la tienda, asegurando que solo personas autorizadas tengan acceso.

| Usabilidad y Calidad de software- Front-end -Vistas Principales |
| --- |
| Vistas principales: SalvaPOS / HOME / Inventario / Reportes |
|  |

La **Vista Front-End** de la aplicación web SalvaPOS ha sido diseñada para ofrecer una experiencia de usuario intuitiva, moderna y minimalista. La interfaz está dividida en un body y un sidebar. Por medio de los enlaces que conforman el sidebar podemos cambiar el contenido del body y de esta manera poder acceder a las principales secciones del sistema.

**Estilo Moderno y Profesional**: Colores sobrios y elementos gráficos que se destacan sin distraer.

**Interactividad y Usabilidad**: Uso de gráficos animados, transiciones suaves y elementos interactivos que facilitan la navegación.

| Usabilidad y Calidad de software – Vistas específicas del Cliente. |
| --- |
| HOME |
|  |

| Usabilidad y Calidad de software – Vistas específicas del Farmacéutico. |
| --- |
| HOME / Inventario (Productos / Categorías) / Venta / Iniciar Sesión. |
| Interfaz de usuario gráfica  Descripción generada automáticamente |

| Usabilidad y Calidad de software – Vistas específicas de administrador |
| --- |
| HOME / Reporte / Registro. |
| Interfaz de usuario gráfica  Descripción generada automáticamente |

Cada tipo de usuario tiene un escritorio personalizado con acceso a las funcionalidades correspondientes a su rol:

* **Administrador**: Gestión de usuarios, dar de baja categorías, visualizar reportes.
* **Farmacéutico:** Gestión de inventarios, realizar ventas, realizar devoluciones.
* **Cliente**: Ver productos y solicitar venta.

## **Consistencia en la cantidad de vistas de la arquitectura.**

Se garantiza que cada vista esté alineada con los requisitos funcionales y no funcionales del sistema, evitando redundancias y manteniendo la coherencia en toda la arquitectura

### **Descripción de módulos.**

| Nombre del módulo | Descripción | Componentes incluidos |
| --- | --- | --- |
| Producto. | Listado de Productos. | Inventario |
| Categoría. | Listado de Categorías. | Inventario |
| Usuario. | Administración de cuentas y roles | Gestor usuario. |
| Venta. | Realizar la venta y operaciones. | Gestor de ventas. |
| Reportes. | Mostrar los reportes | Gestor de reportes. |

### DESCRIPCIÓN DE COMPONENTES

| Nombre del Componente | Descripción | Componentes Relaciones |
| --- | --- | --- |
| Inventario | Permitir la creación y edición de las categorías y productos. | Venta |
| Gestor de usuarios. | Gestionar el ingreso, registro y perfil de usuario de los farmacéuticos que acceden a la aplicación.  Administración de usuarios y control de acceso. | Inventario, Gestor de ventas. |
| Gestor de Ventas | Realización de ventas, anulación y devolución de ventas. Búsqueda de venta por Fecha u ID. Calculo de inventario cuando se compra algo. | Inventario, Gestor de usuarios. |
| Gestor de Reportes | Recopilación de las ventas y despliegue de información en un dashboard. | Venta. |

### DESCRIPCIÓN DE CONECTORES

| Nombre del Conector | Tipo | Propósito |
| --- | --- | --- |
| Rest API | Comunicación | Transferencia de datos de NestJS. |
| Servicios Angular | Comunicación | Cargar los datos de NestJS y poder mostrarlos en frontend. |

* + - 1. **Arquitectura lógica**
      2. Performances

El sistema está diseñado para manejar una media concurrencia, soportando más de 500 usuarios simultáneamente sin comprometer el rendimiento. Para lograr tiempos de respuesta inferiores a 2 segundos por transacción, se implementarán las siguientes estrategias:

* **Escalabilidad Horizontal**: Se utilizarán servicios de Angular para poder conectar con el Back-end de NestJS por medio de un API.
* **Optimización de Consultas**: Se aplicarán técnicas de optimización en las consultas a la base de datos MySQL, incluyendo índices y consultas asíncronas para reducir los tiempos de respuesta.
  + - 1. Criterios de Calidad

El sistema debe cumplir con los siguientes criterios de calidad:

* **Responsividad**: La aplicación debe adaptarse a todas las pantallas.
* **Respuesta Rápida**: Todas las transacciones deberán completarse en menos de 2 segundos, lo que se logrará mediante el uso de técnicas de caché y un diseño eficiente de la API.
  + - 1. Detalles de la implementación

La implementación del sistema se basará en una arquitectura de microservicios, utilizando las siguientes tecnologías:

* **Backend**: NestJS.
* **Frontend**: Angular – Bootstrap 5.
* **Base de Datos**: MySQL
  + - 1. Lenguajes y plataformas

Las tecnologías y versiones específicas que se utilizarán en el desarrollo del sistema incluyen:

* **Node.js**: v10.x
* **NestJS**: v4.x
* **Angular**: v17.x
* **MySQL**: v8.x
* **Visual Studio Code**: v1.50

## Acceso a fuentes de información del proyecto

Para acceder a la carpeta de evidencias acumuladas y otros documentos relevantes del proyecto, puede utilizar el siguiente enlace:  
  
Insertar link: https://drive.google.com/drive/u/3/folders/1oe5GxJtIK5mk5dxHXjKESdrX6-zQIwZ\_?dmr=1&ec=wgc-drive-globalnav-goto  
  
Esta carpeta incluye documentación técnica, diagramas, pruebas de rendimiento y cualquier otra evidencia relacionada con el desarrollo y la implementación del sistema