



# Trabajo Práctico

Materia: Prácticas Profesionales 3 (PP3)

Profesora: Patricia Litovicius

Alumnos: Alancay Sergio y Pulvet Angeles

Grupo: 8

Año: 2025

## **Índice:**

3.....	Brief
6.....	FODA
8.....	En el marco del proyecto grupal (...)
10.....	Casos de Uso y User Persona
13.....	Diseño de Mockups / Prototipos de Interfaz
15.....	Diseño y arquitectura de sistemas
18.....	Actividad: Elaboración del Diagrama de Clases UML del Sistema
23.....	Diagrama de secuencia
29.....	Diagrama de componentes y Despliegue del sistema
30.....	Conclusiones finales

# Brief

## Definición del Caso – Aplicación de Gestión de emprendimientos para emprendedores

### 1. Contexto

- **Dónde sucede el caso:**  
El caso ocurre en el entorno de **emprendedores individuales, microemprendimientos y freelancers**, principalmente en ciudades de Latinoamérica.
- **Características principales del entorno:**
  - Emprendimientos pequeños, con 1 a 5 personas en promedio.
  - Recursos económicos limitados, por lo que no pueden pagar software costoso.
  - Gestión manual de ventas, gastos y tareas, usando Excel, cuadernos o apps aisladas.
  - Alta necesidad de herramientas móviles y simples, ya que muchos trabajan solos y sin conocimientos contables avanzados.

### 2. Problema o necesidad detectada

Los emprendedores enfrentan **desorden y falta de control** en la administración de su negocio:

- Pierden tiempo organizando ventas y gastos manualmente.
- Carecen de reportes claros sobre ganancias y pérdidas.
- Dificultad para planificar y priorizar tareas del emprendimiento.
- Uso de múltiples herramientas separadas que no se integran entre sí.

Se requiere una **aplicación centralizada** que permita organizar el negocio de manera simple, rápida y accesible.

### 3. Usuarios involucrados

- **Usuario principal:**

- **Emprendedores / dueños del proyecto** → necesitan controlar ingresos, egresos, clientes y tareas de manera práctica.
- **Otros usuarios relevantes:**
  - **Colaboradores o socios** → que participan en las ventas o producción y requieren acceso parcial a la información.
  - **Contadores externos** → que pueden necesitar reportes básicos para declaraciones fiscales.
  - **Clientes indirectos** → podrían recibir facturas o comprobantes generados por la app.

## 4. Objetivo del sistema

- Proporcionar a los emprendedores una **herramienta integral y fácil de usar** que:
  - Centralice ventas, gastos y clientes.
  - Permite generar reportes claros de rentabilidad.
  - Ayuda a planificar tareas y organizar el flujo de trabajo.
  - Reduzca errores y el tiempo dedicado a la administración.
  - Sea accesible desde cualquier dispositivo (desktop y móvil).

## 5. Alcance esperado

- **Qué sí resolverá el sistema:**
  - Registro de ventas e ingresos.
  - Control de gastos y generación de reportes básicos de ganancias/pérdidas.
  - Gestión de clientes y proveedores.
  - Organización de tareas y actividades (to-do list y recordatorios).
  - Generación de comprobantes simples (ej.: facturas básicas).

- Acceso multiusuario para socios/colaboradores con roles limitados.
- **Qué no resolverá el sistema (fuera de alcance inicial):**
  - Gestión compleja de nómina y recursos humanos.
  - Integración avanzada con sistemas contables de terceros.
  - Automatización de campañas de marketing (será fase futura).
  - Inteligencia de negocio predictiva (análisis avanzado de datos).

# Análisis FODA – Aplicación de Gestión de emprendimientos para emprendedores

## Fortalezas (Internas – Positivas)

- Interfaz **simple e intuitiva**, diseñada para personas sin experiencia técnica.
- **Accesible desde dispositivos móviles**, ideal para emprendedores que trabajan en movimiento.
- Precio competitivo o modelo freemium, lo que facilita la adopción.
- Centraliza en un solo lugar la **gestión de ventas, gastos, clientes y tareas**.
- Posibilidad de escalar con funciones adicionales (modularidad).
- Soporte en español y adaptado al contexto latinoamericano.

## Oportunidades (Externas – Positivas)

- Creciente **digitalización de los emprendedores** post-pandemia.
- Mercado en expansión: cada vez más personas crean microemprendimientos.
- Falta de soluciones simples y unificadas para este nicho.
- Potencial de alianzas con **incubadoras, coworkings y programas de emprendedores**.
- Tendencia al uso de apps móviles para controlar finanzas personales y de negocio.

## Debilidades (Internas – Negativas)

- Presupuesto limitado para marketing y adquisición de usuarios.
- Dependencia de un equipo pequeño para mantenimiento y soporte.
- Producto nuevo: falta de reputación y testimonios de usuarios.
- Funcionalidad inicial limitada (no cubre necesidades de emprendimientos más complejos).

- Posible curva de aprendizaje para usuarios poco familiarizados con apps digitales.

#### Amenazas (Externas – Negativas)

- Competencia creciente: herramientas como Notion, Trello, Alegra y apps de contabilidad.
- Riesgo de baja adopción si los usuarios perciben que es complejo o innecesario.
- Inestabilidad económica que puede hacer que los emprendedores prioricen otros gastos.
- Cambios en normativas fiscales que requieran actualizaciones constantes.
- Aparición de competidores con más inversión y marketing.

# En el marco del proyecto grupal que ya están desarrollando, deberán:

## 1. Contexto: Objetivo y Actores

**Objetivo del proyecto:** El objetivo principal de nuestro proyecto es desarrollar una herramienta digital integral y fácil de usar que actúe como un "organizador" para los emprendedores. La plataforma está diseñada para ayudarlos a gestionar y ordenar su negocio de manera eficiente, simplificando tareas clave como la administración de clientes, el seguimiento de ventas, la gestión de inventario y la planificación financiera.

**Actores principales:** El único actor principal que interactúa directamente con el sistema es el emprendedor. Él es el usuario directo de la herramienta, usándola para organizar, controlar y hacer crecer su negocio.

## 2. Requerimientos Funcionales

1) **Gestión de clientes:** El sistema debe permitir al emprendedor registrar, editar y eliminar información de sus clientes, incluyendo datos de contacto e historial de compras.

2) **Seguimiento de ventas:** El sistema debe registrar las ventas realizadas, asociándolas a un cliente y producto, y generar reportes de ingresos por período.

3) **Gestión de productos/inventario:** El sistema debe permitir al emprendedor crear un catálogo de productos o servicios, registrar su stock y alertar cuando un producto esté por agotarse.

4) **Generación de reportes:** El sistema debe generar reportes visuales (gráficos y tablas) sobre ventas, ingresos y productos más vendidos.

## 3. Requerimientos No Funcionales

**Rendimiento:** El sistema debe cargar cualquier página o reporte en menos de 3 segundos, incluso con un gran volumen de datos (más de 1000 registros).

**Seguridad:** El sistema debe proteger la información sensible de los usuarios (emprendedores y sus clientes) mediante encriptación de contraseñas y uso de protocolos de seguridad HTTPS.



Usabilidad: La interfaz de usuario debe ser intuitiva y fácil de navegar, permitiendo a un usuario nuevo utilizar las funciones principales (como registrar una venta) sin necesidad de consultar un manual.

#### 4. Historias de Usuario

Ejemplos:

Historia de Usuario 1: "Como emprendedor, quiero poder registrar cada venta que hago en mi negocio para poder llevar un control exacto de mis ingresos y no depender de apuntes en papel."

Historia de Usuario 2: "Como emprendedor, quiero recibir una notificación cuando un producto de mi inventario esté a punto de agotarse para poder hacer un nuevo pedido a tiempo y no perder ventas."

#### 5. Caso de Uso

Seleccionaremos la funcionalidad de registro de una nueva venta como caso de uso clave.

Nombre del Caso de Uso: Registrar una Venta

Actor: Emprendedor

Objetivo: Registrar una venta de forma completa y precisa en el sistema.

1. Flujo principal: El emprendedor inicia sesión en el sistema y navega a la sección de "ventas".

Descripción: Selecciona la opción "Nueva venta".

2. Flujo principal: El sistema muestra un formulario para registrar la venta.

Descripción: El emprendedor ingresa los datos de la venta: selecciona un cliente de la lista, elige los productos comprados y ajusta las cantidades.

3. Flujo principal: El sistema calcula automáticamente el total de la venta.

Descripción: El emprendedor confirma los datos y hace clic en "Guardar venta".

4. Flujo principal: El sistema guarda la nueva venta, actualiza el inventario de los productos vendidos y muestra un mensaje de confirmación.

# Actividad: Casos de Uso y User Persona

## 1. Diagrama de Casos de Uso – Aplicación de Gestión de Emprendimientos

Actores identificados:

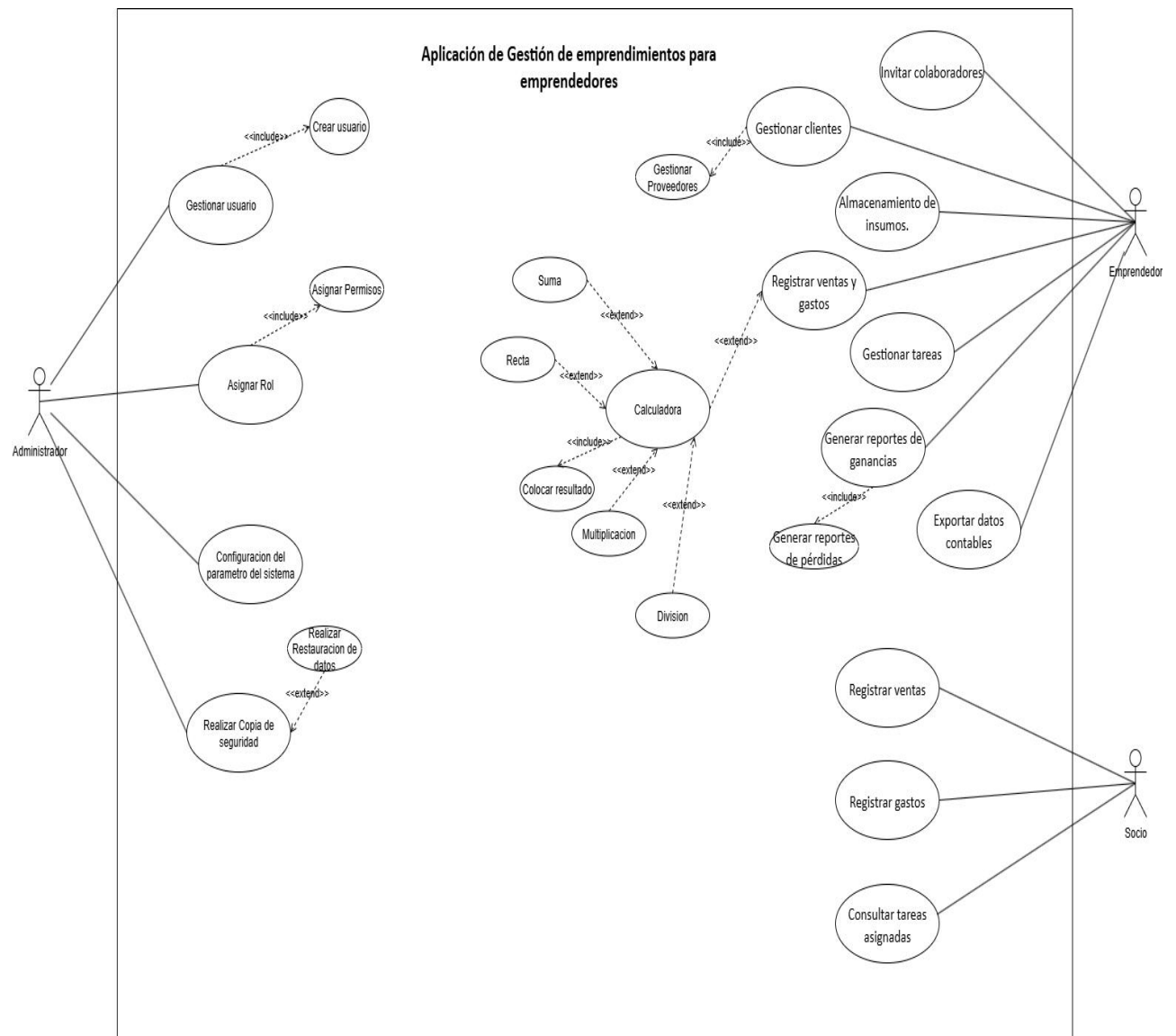
Administrador: gestiona usuarios, roles, permisos y configuraciones generales.

Emprendedor: gestiona el emprendimiento en sí (ventas, gastos, reportes, clientes).

Colaborador/Socio: registra operaciones simples (ventas, gastos, tareas).

Casos de Uso principales por actor:

- Administrador:
  - Crear/gestionar usuarios
  - Asignar roles y permisos
  - Configurar parámetros del sistema (moneda, impuestos, categorías)
  - Realizar copias de seguridad y restauración de datos
- Emprendedor:
  - Registrar ventas y gastos
  - Gestionar clientes y proveedores
  - Gestionar tareas
  - Generar reportes de ganancias/pérdidas
  - Invitar colaboradores
  - Exportar datos contables
  - Almacenamiento de insumos.
- Colaborador/Socio:
  - Registrar ventas
  - Registrar gastos
  - Consultar tareas asignadas



## 2. User Persona (para actores humanos):

### User Persona 1 – Emprendedor

- Nombre: Laura Ríos
- Edad y ocupación: 32 años, dueña de un emprendimiento de pastelería.
- Objetivos al usar el sistema:
  - Llevar control de ingresos y egresos en un solo lugar.
  - Saber su rentabilidad mes a mes.
  - Ahorrar tiempo en tareas administrativas.
- Necesidades y frustraciones:
  - Necesita una herramienta simple, que no requiera conocimientos contables.
  - Se frustra cuando tiene que revisar planillas de Excel y no entiende los números.
- Contexto de uso:
  - Dispositivo: principalmente smartphone, a veces laptop.

- Frecuencia: uso diario para registrar ventas y gastos.
- Entorno: desde su casa/taller o mientras entrega pedidos.

#### User Persona 2 – Colaborador/Socio

- Nombre: Martín Gómez
- Edad y ocupación: 25 años, asistente de ventas del emprendimiento.
- Objetivos al usar el sistema:
  - Registrar ventas en tiempo real.
  - Marcar tareas como completadas.
- Necesidades y frustraciones:
  - Necesita que el sistema sea rápido, fácil de usar en el móvil.
  - Se frustra si debe pedir acceso al emprendedor para cada registro.
- Contexto de uso:
  - Dispositivo: smartphone 100%.
  - Frecuencia: varias veces al día, mientras atiende clientes.
  - Entorno: local físico o ferias/emprendimientos itinerantes.

#### User Persona 3 – Administrador

- Nombre: Diego Fernández
- Edad y ocupación: 38 años, administrador de sistemas dentro de la empresa desarrolladora.
- Objetivos al usar el sistema:
  - Mantener la plataforma estable y segura.
  - Controlar accesos y permisos de los usuarios.
  - Realizar configuraciones iniciales para nuevos clientes.
- Necesidades y frustraciones:
  - Necesita herramientas claras para administrar usuarios y roles sin complicaciones.
  - Se frustra si hay errores que interrumpen el acceso de los emprendedores.
- Contexto de uso:
  - Dispositivo: computadora de escritorio.
  - Frecuencia: esporádica (cuando hay nuevas altas o mantenimiento del sistema).
  - Entorno: oficina técnica o área de soporte de la aplicación.

# Diseño de Mockups / Prototipos de Interfaz

## 1. Pantallas principales del sistema

Las pantallas básicas para un MVP (producto mínimo viable) serían:

### 1) Pantalla de Inicio de Sesión / Registro

- Campos: Email, contraseña, botón “Ingresar” y “Crear cuenta”.
- Navegación: acceso a Recuperar contraseña.

### 2) Dashboard (Inicio)

- Resumen de ventas, gastos y balance del día/mes.
- Atajos rápidos: “+ Venta”, “+ Gasto”, “+ Cliente”.
- Menú inferior con accesos a: Inicio, Reportes, Tareas, Configuración.

### 3) Gestión de Ventas

- Lista de ventas con búsqueda y filtro por fecha.
- Botón flotante **+** para registrar nueva venta.

### 4) Gestión de Gastos

- Similar a ventas, pero orientado a egresos.

### 5) Clientes / Proveedores

- Lista con tarjetas (nombre, contacto, historial).
- Botón **+** para agregar cliente/proveedor.

### 6) Reportes

- Gráficos simples (barras, circulares) sobre ingresos, egresos y balance.

### 7) Gestión de Tareas

- Lista estilo checklist con recordatorios.

### 8) Configuración (Administrador/Usuario)

- Perfil, idioma, roles de usuario, seguridad.

## 2. Principios de la Gestalt aplicados

- Proximidad: agrupar botones relacionados (ej. “+ Venta” y “+ Gastó” cerca).
- Semejanza: iconos consistentes para ventas, gastos, clientes.

- Figura-fondo: uso de colores contrastantes (ej. fondo claro, botones destacados en azul o verde).
- Continuidad: navegación lineal y consistente con barra inferior fija.
- Simetría: formularios alineados y con jerarquía visual clara.
- Cierre: tarjetas con bordes redondeados que delimitan secciones.

### **3. Heurísticas de Nielsen aplicadas**

- Visibilidad del estado: confirmaciones después de registrar una venta (“✓ Venta guardada”).
- Consistencia: mismos colores, tipografías y patrones en todas las pantallas.
- Prevención de errores: validación en formularios (ej. no permitir campo vacío en monto).
- Control y libertad: opción de cancelar al registrar una venta/gasto.
- Reconocimiento sobre recuerdo: iconos + etiquetas (“Clientes”, “Reportes”).

### **4. Interacciones básicas**

- El usuario entra al Dashboard, ve un resumen y accede rápido a registrar una venta/gasto.
- Desde la barra inferior, puede cambiar entre Clientes, Reportes, Tareas.
- Al pulsar “+ Venta”, aparece un formulario emergente → al confirmar, regresa al Dashboard actualizado.
- En Reportes, puede tocar un gráfico y filtrar por fechas.
- En Tareas, puede deslizar a la izquierda/derecha para marcar completada o eliminar.

### **5. Enfoque UX/UI**

- Diseño minimalista: menos clics, más claridad.
- Botones grandes y con colores de acción (verde = confirmar, rojo = eliminar).
- Íconos universales (ej. ventas, reportes).
- Uso de gráficos simples y visuales para reportes.
- Todo accesible en máx. 2–3 toques desde el Dashboard.

**Link del mockup de figma:**

<https://www.figma.com/design/Mfrt7rSj1AYt5PodiUQaF2/Sin-t%C3%ADtulo?node-id=0-1&t=MaHZ9QN6PnOvWGwu-0>

# Diseño y arquitectura de sistemas

## Parte 1: Cuadro de trazabilidad

### Cuadro del proyecto:

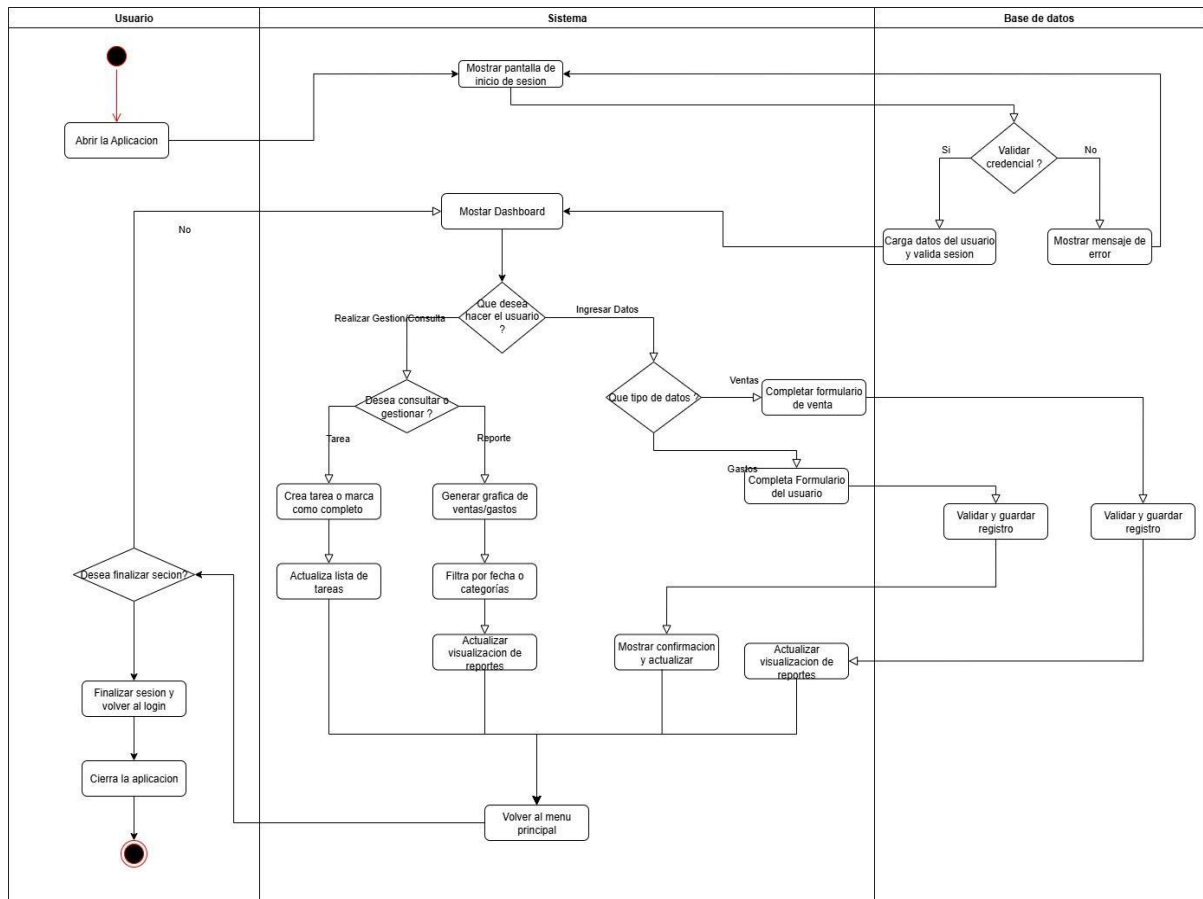
Caso de Uso	Actor Involucrado	Actividad del Sistema	Pantalla UX Relacionada
Iniciar sesión	Emprendedor / Administrador	Validar credenciales, mostrar inicio personalizado según el rol	Pantalla de <i>Inicio de Sesión</i>
Registrar nueva venta	Emprendedor	Solicitar datos del cliente, monto, fecha → Guardar registro en base de datos.	Pantalla <i>Nueva Venta</i>
Registrar nuevo gasto	Emprendedor	Solicitar datos del gasto → Registrar monto y categoría.	Pantalla <i>Nuevo Gasto</i>
Consultar balance general	Emprendedor / Administrador	Calcular ingresos – egresos → Mostrar resumen del mes y gráfico.	Pantalla <i>Dashboard / Reportes</i>
Visualizar reportes financieros	Emprendedor / Administrador	Filtrar por fechas y categorías → Generar gráficos dinámicos.	Pantalla <i>Reportes</i>

Gestionar clientes/proveedores	Emprendedor	Consultar lista, agregar o editar información de contacto.	Pantalla <i>Clientes / Proveedores</i>
Administrar tareas o recordatorios	Emprendedor	Crear, marcar o eliminar tareas.	Pantalla <i>Gestión de Tareas</i>
Gestionar usuarios y roles	Administrador	Agregar, editar o eliminar usuarios → Asignar permisos.	Pantalla <i>Configuración (Admin)</i>
Configurar parámetros generales del sistema	Administrador	Actualizar preferencias (idioma, notificaciones, seguridad).	Pantalla <i>Configuración</i>

## Parte 2: Diagrama de Actividades

El diagrama de actividades de proyecto:





# Actividad: Elaboración del Diagrama de Clases UML del Sistema

## 1. Clases identificadas y justificación.

Basado en los casos de uso y en las pantallas UX, las clases principales del sistema son:

Clase	Descripción	Justificación
<b>Usuario</b>	Clase base para cualquier persona que acceda al sistema.	Permite autenticación y manejo de roles.
<b>Emprendedor</b>	Usuario principal que gestiona las operaciones del emprendimiento.	Actor principal del sistema.
<b>Socio</b>	Usuario colaborador que asiste al emprendedor en tareas o registros de operaciones.	Permite trabajo en equipo y delegación.
<b>Administrador</b>	Usuario con permisos técnicos de configuración y mantenimiento.	Supervisa el sistema.
<b>Venta</b>	Representa una transacción de ingreso registrada por el emprendimiento.	Caso de uso: "Registrar venta".
<b>Gasto</b>	Representa un egreso o salida de dinero.	Caso de uso: "Registrar gasto".
<b>Reporte</b>	Genera informes financieros basados en ventas y gastos.	Caso de uso: "Visualizar reportes".
<b>Tarea</b>	Permite crear y gestionar pendientes del emprendimiento.	Caso de uso: "Gestionar tareas".
<b>Producto</b>	Representa los bienes o servicios ofrecidos por el emprendimiento.	Permite registrar, actualizar y consultar productos.

## 2. Atributos y métodos principales

<b>Clase</b>	<b>Atributos (nombre : tipo)</b>	<b>Métodos principales</b>
<b>Usuario</b>	<ul style="list-style-type: none"> <li>- idUsuario: int</li> <li>- nombre: String</li> <li>- email: String</li> <li>- contraseña: String</li> <li>- rol: String</li> </ul>	<ul style="list-style-type: none"> <li>+ iniciarSesion(): boolean</li> <li>+ cerrarSesion(): void</li> </ul>
<b>Emprendedor</b>	<ul style="list-style-type: none"> <li>- nombreEmprendimiento: String</li> </ul>	<ul style="list-style-type: none"> <li>+ registrarVenta(v: Venta): void</li> <li>+ registrarGasto(g: Gasto): void</li> <li>+ generarReporte(): Reporte</li> <li>+ gestionarTareas(): void</li> <li>+ invitarSocio(s: Socio): void</li> </ul>
<b>Socio</b>	<ul style="list-style-type: none"> <li>- rolAsignado: String</li> </ul>	<ul style="list-style-type: none"> <li>+ registrarVenta(v: Venta): void</li> <li>+ registrarGasto(g: Gasto): void</li> <li>+ actualizarTarea(t: Tarea): void</li> </ul>
<b>Administrador</b>		<ul style="list-style-type: none"> <li>+ gestionarUsuarios(): void</li> <li>+ configurarSistema(): void</li> </ul>
<b>Venta</b>	<ul style="list-style-type: none"> <li>- idVenta: int</li> <li>- fecha: Date</li> <li>- monto: float</li> </ul>	<ul style="list-style-type: none"> <li>+ guardarVenta(): void</li> <li>+ calcularTotal(): float</li> </ul>
<b>Gasto</b>	<ul style="list-style-type: none"> <li>- idGasto: int</li> <li>- descripcion: String</li> <li>- monto: float</li> <li>- fecha: Date</li> </ul>	<ul style="list-style-type: none"> <li>+ calcularTotal(): float</li> <li>+ guardarGasto(): void</li> </ul>
<b>Reporte</b>	<ul style="list-style-type: none"> <li>- idReporte: int</li> </ul>	<ul style="list-style-type: none"> <li>+ generarBalance(): float</li> </ul>

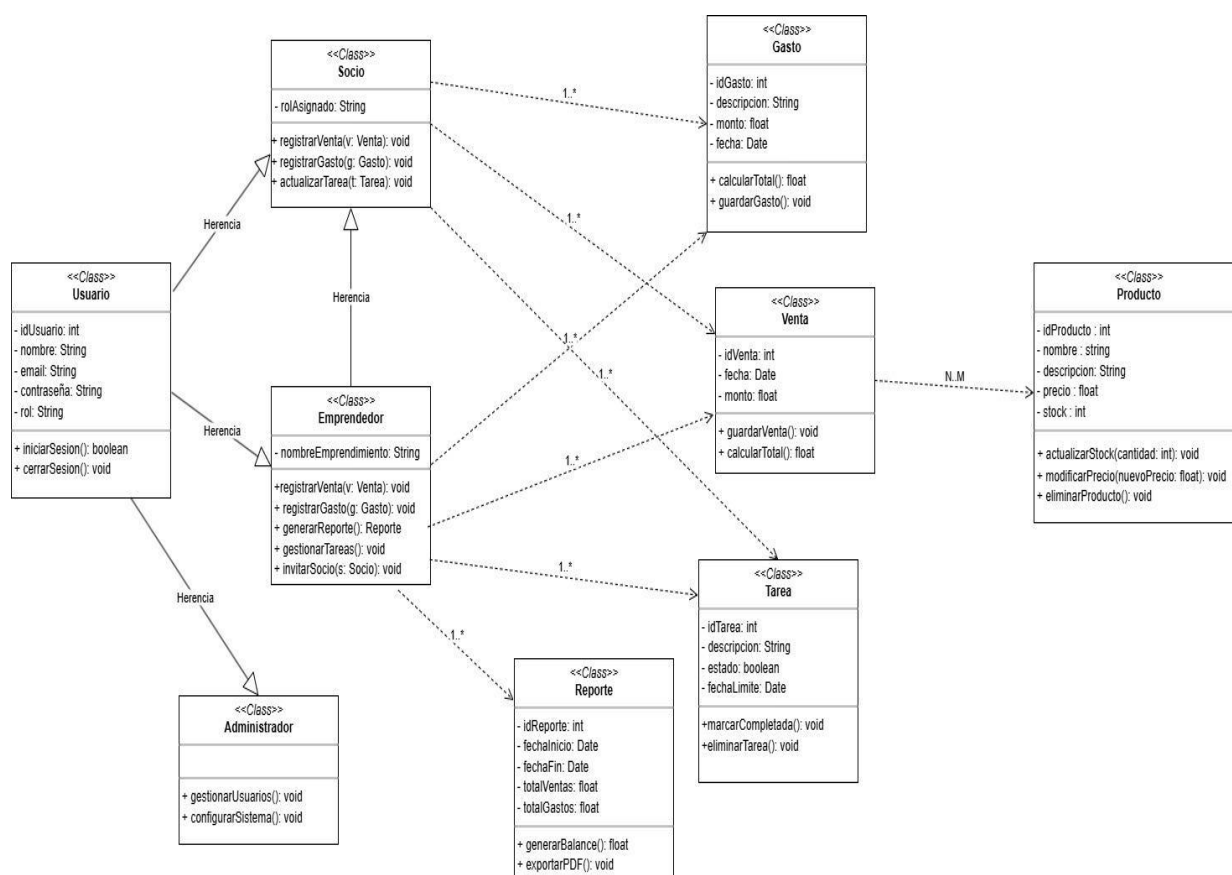
	<ul style="list-style-type: none"> <li>- fechaInicio: Date</li> <li>- fechaFin: Date</li> <li>- totalVentas: float</li> <li>- totalGastos: float</li> </ul>	+ exportarPDF(): void
<b>Tarea</b>	<ul style="list-style-type: none"> <li>- idTarea: int</li> <li>- descripcion: String</li> <li>- estado: boolean</li> <li>- fechaLimite: Date</li> </ul>	+marcarCompletada(): void + eliminarTarea(): void
<b>Producto</b>	<ul style="list-style-type: none"> <li>- idProducto : int</li> <li>- nombre : string</li> <li>- descripcion: String</li> <li>- precio : float</li> <li>- stock : int</li> </ul>	+ actualizarStock(cantidad: int): void + modificarPrecio(nuevoPrecio: float): void + eliminarProducto(): void

### 3. Relaciones entre clases

Relación	Tipo	Descripción
<b>Usuario <del>→</del> Emprendedor / Socio / Administrador</b>	Herencia	Diferencia los tipos de usuario.
<b>Emprendedor <del>→</del> Socio</b>	Asociación 1..*	Un emprendedor puede tener varios socios colaboradores.
<b>Emprendedor → Producto</b>	Composición 1..*	Los productos pertenecen al emprendimiento.
<b>Emprendedor <del>→</del> Venta</b>	Asociación 1..*	El emprendedor o sus socios pueden registrar ventas.
<b>Venta <del>→</del> Producto</b>	Asociación N..M	Una venta puede incluir varios productos.
<b>Emprendedor <del>→</del> Gasto</b>	Asociación 1..*	Registra los egresos.
<b>Emprendedor <del>→</del> Reporte</b>	Asociación 1..*	Cada emprendimiento puede generar varios

		reportes.
<b>Emprendedor → Tarea</b>	Composición 1..*	Las tareas son creadas y gestionadas por el emprendedor.
<b>Socio → Producto / Tarea / Venta</b>	Asociación	Colabora registrando o modificando información.

#### 4. Elaborar el Diagrama de Clases UML utilizando una herramienta gratuita



#### 5. Descripción breve del diagrama

Este **diagrama de clases UML** representa el núcleo estructural del sistema de **Gestión de Emprendimientos para Emprendedores y Socios**.

El **Emprendedor** es el usuario principal, encargado de la gestión económica y operativa del negocio, mientras que el **Socio** colabora en tareas específicas.

Ambos interactúan con entidades de negocio como **Producto**, **Venta**, **Gasto**, **Reporte** y **Tarea**, que modelan las principales operaciones del emprendimiento.

La inclusión de la clase **Producto** permite controlar el inventario, actualizar precios y relacionar cada venta con los productos comercializados, lo cual amplía el alcance funcional del sistema.

La estructura mantiene coherencia con los **casos de uso, mockups y diagramas previos**, asegurando consistencia entre la lógica de negocio y la experiencia del usuario.

# Diagrama de Secuencia

## 1. Caso de uso seleccionado

**Caso de uso:** “Registrar Venta”

**Descripción breve:**

El **Emprendedor** o **Socio** registra una nueva venta desde la aplicación. El sistema valida la información, actualiza desde la base de datos, guarda la transacción y genera un resumen de confirmación.

## 2. Actores y objetos participantes

Elemento	Tipo	Descripción
Emprendedor / Socio	Actor	Inicia el proceso de registro de venta
InterfazUsuario	Objeto	Representa la pantalla de registro de venta (móvil)
ControladorVenta	Objeto	Gestiona la lógica de registro, validación y comunicación con las clases del dominio
BaseDeDatos	Objeto	Almacena los datos de la venta y actualizaciones

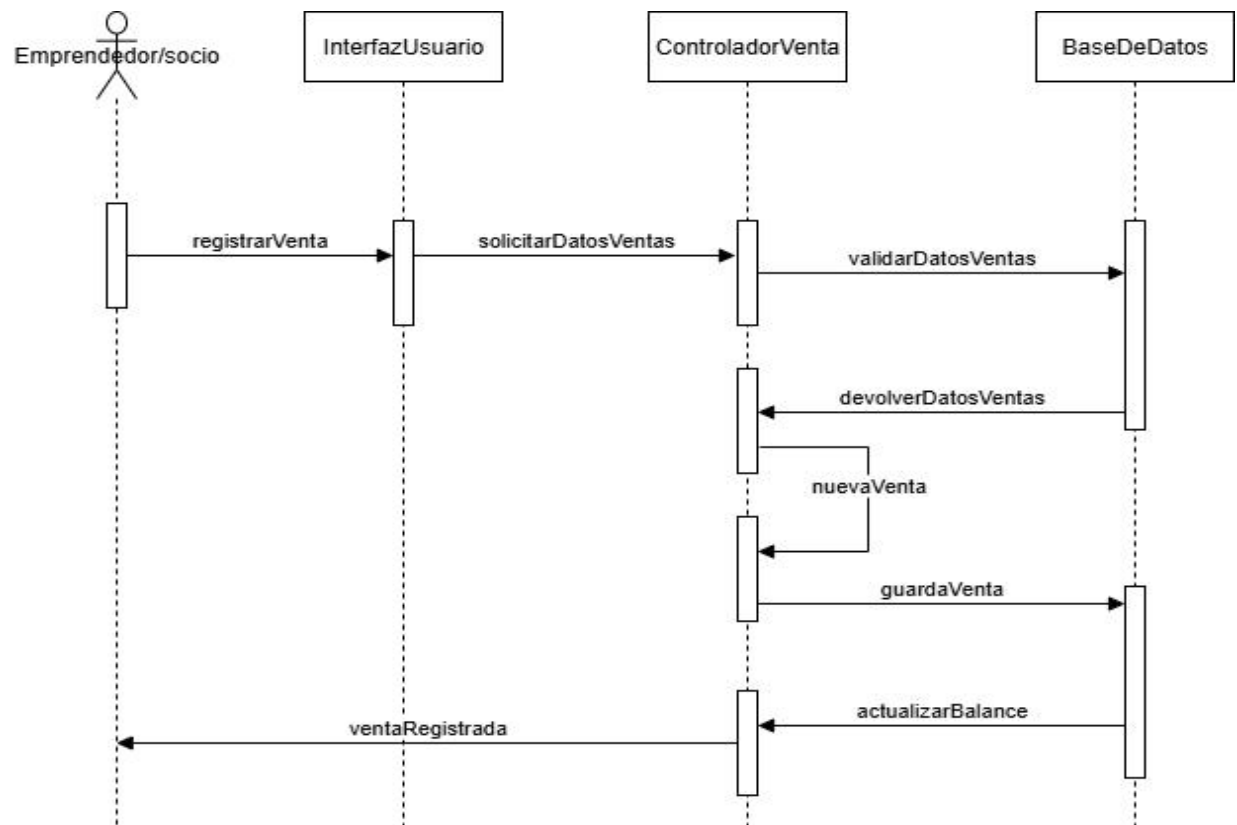
## 3. Flujo del diagrama (mensajes)

1. **Emprendedor/Socio** selecciona la opción “*Registrar Venta*” desde la **InterfazUsuario**.
2. La **InterfazUsuario** solicita los datos de la venta (producto, cantidad, monto, fecha).
3. El usuario completa los campos y presiona “*Guardar*”.
4. La **InterfazUsuario** envía los datos al **ControladorVenta** (*SolicitudDatosVenta()*).

5. El **ControladorVenta** solicita a la **BaseDeDatos** validar los datos (*validarDatosVenta()*).
6. La **BaseDeDatos** valida la información recibida.
7. El **ControladorVenta** crea una nueva instancia de **Venta** (*nuevaVenta = new Venta(...)*).
8. El **ControladorVenta** solicita a la **BaseDeDatos** guardar la venta (*guardarVenta()*).
9. La **BaseDeDatos** confirma la persistencia.
10. La **BaseDeDatos** notifica al **ControladorVenta** para actualizar los totales (*actualizarBalance()*).
11. El **ControladorVenta** genera un nuevo balance.
12. Finalmente, la **InterfazUsuario** muestra el mensaje “Venta registrada exitosamente”.

## 5. Diagrama de Secuencia





## 6. Descripción textual (para incluir en el informe)

El diagrama de secuencia del caso de uso “**Registrar Venta**” describe la interacción entre los objetos del sistema para cumplir con la operación principal de registrar una venta.

El **Emprendedor o Socio** inicia el proceso desde la **InterfazUsuario**, que actúa como punto de entrada.

El **ControladorVenta** coordina la lógica del negocio, validando los datos y comunicándose con las clases **Producto**, **Venta**, **Reporte** y **BaseDeDatos**.

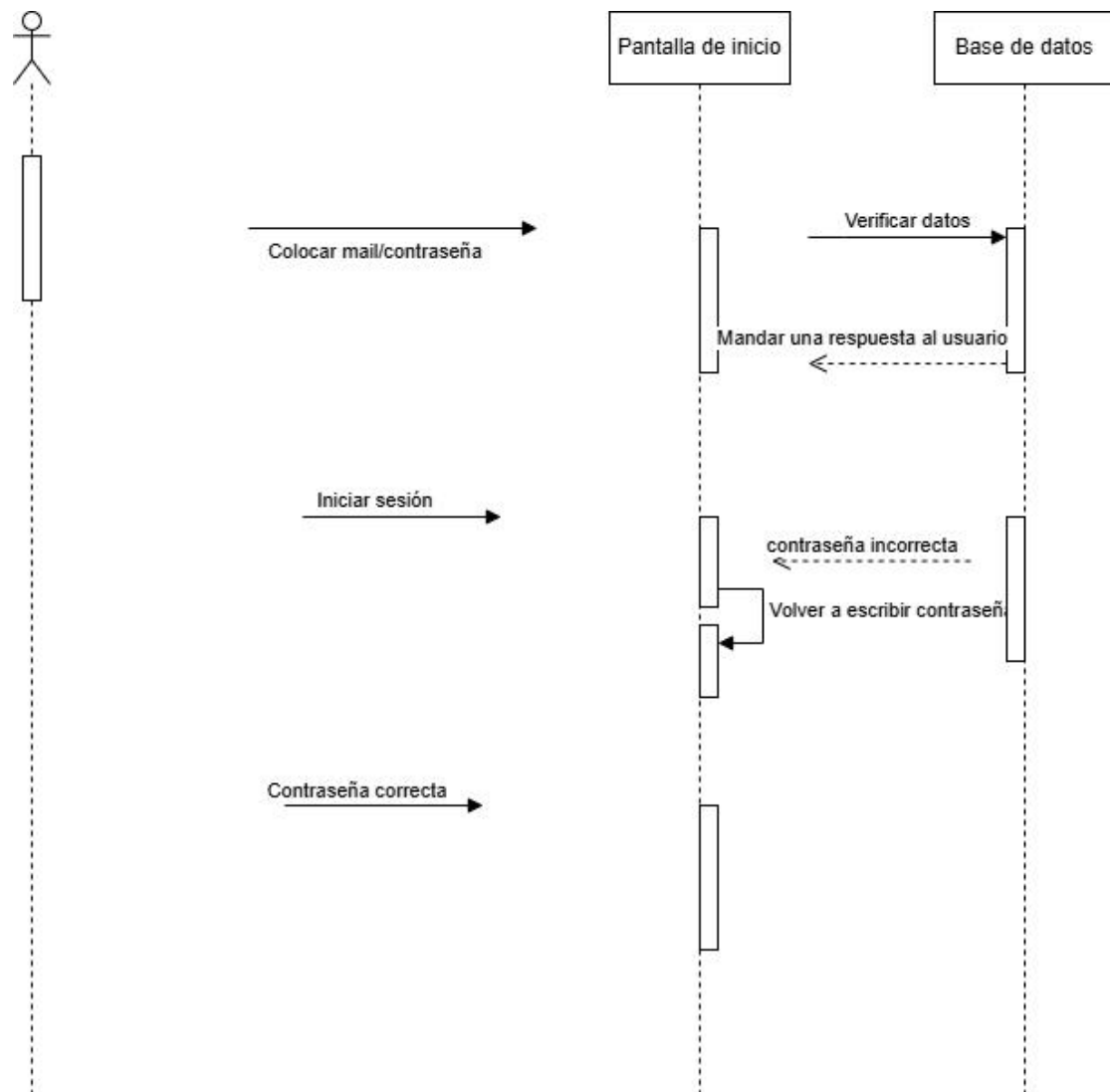
El sistema asegura la consistencia de los datos mediante la actualización del stock, el almacenamiento persistente de la venta y la generación automática del reporte financiero.

Finalmente, el usuario recibe una notificación de éxito.

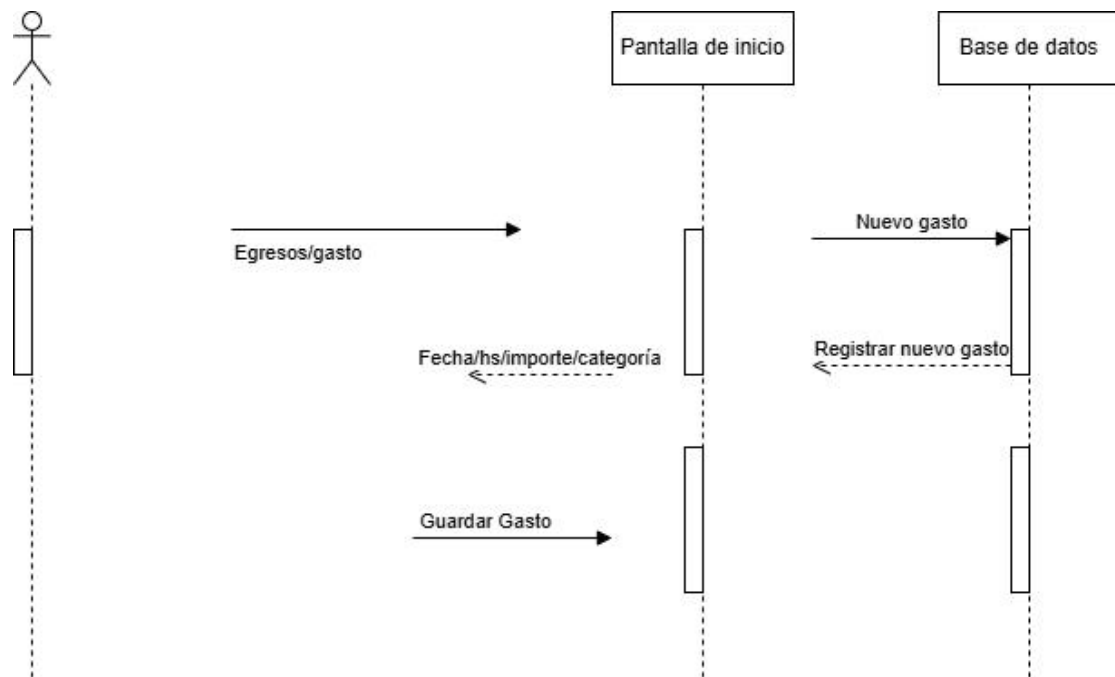
Este flujo refleja la **colaboración ordenada y sincrónica** entre los componentes del sistema y mantiene coherencia con los casos de uso, actividades y clases definidos anteriormente.

## Ejemplos de otros diagramas de secuencia:

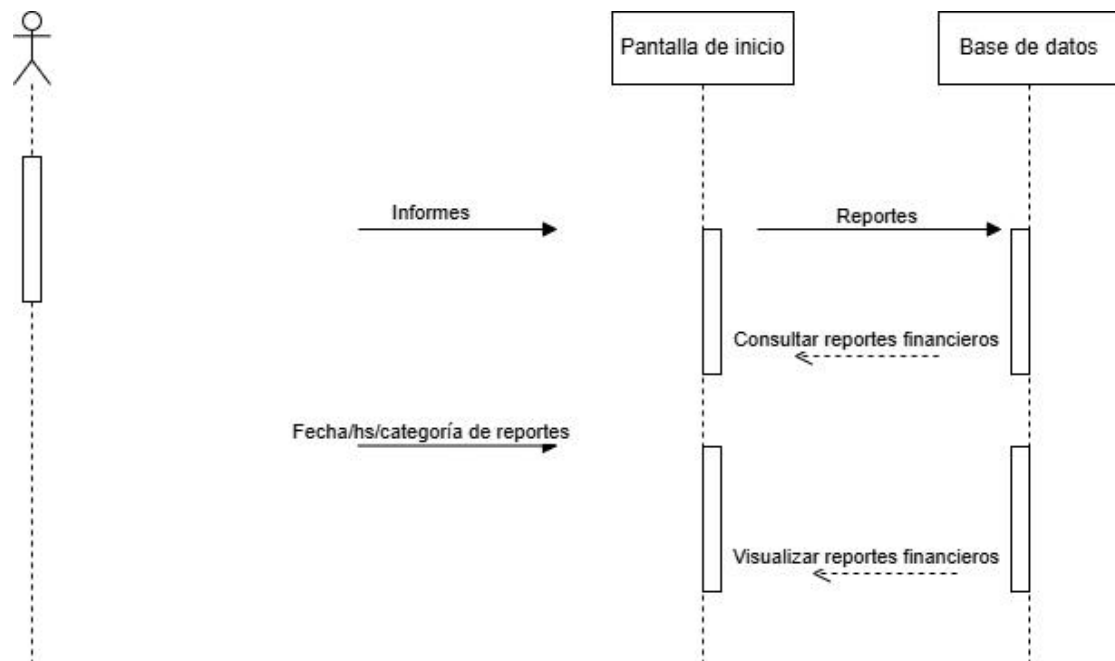
. Inicio de sesión:



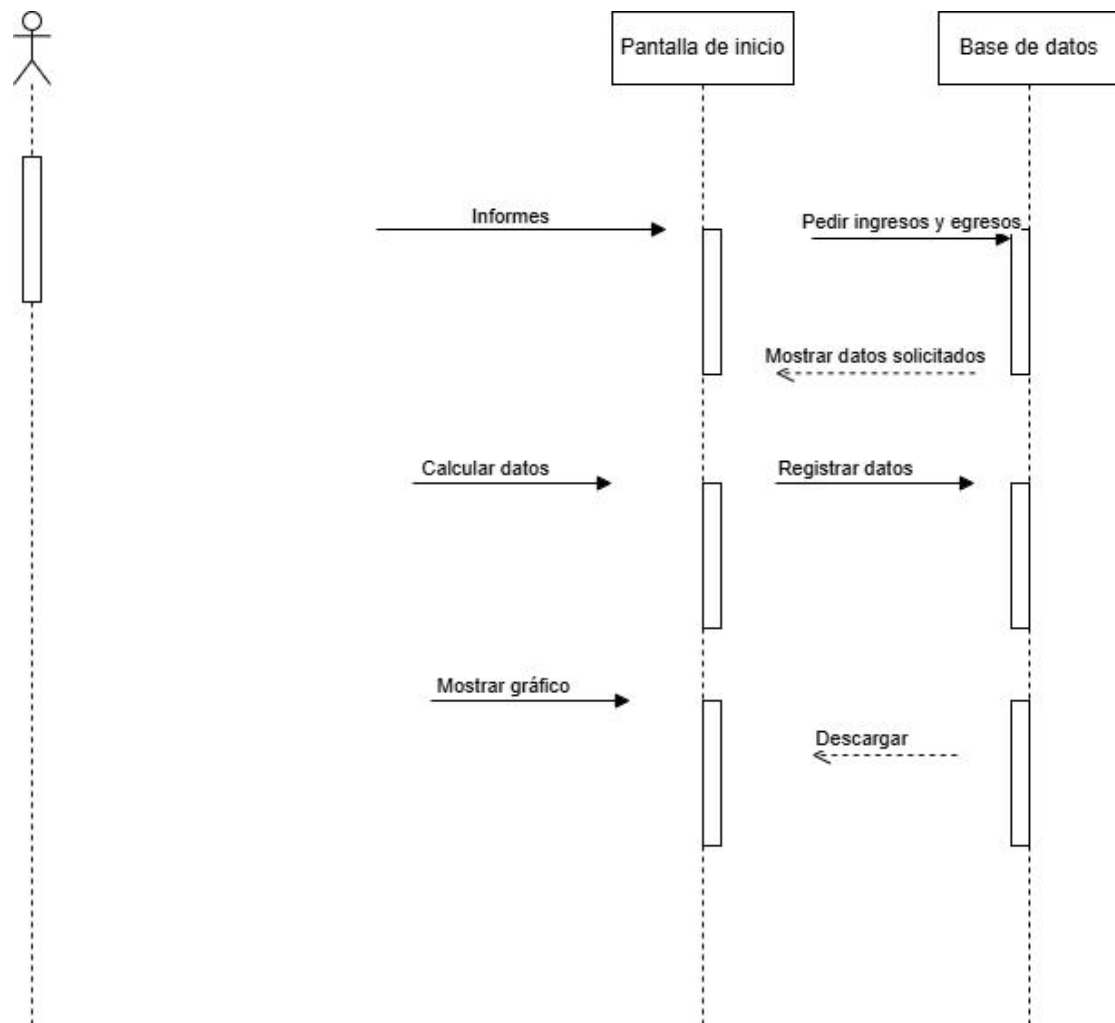
. Registro de Gastos:



. Visualizar Reportes:



. Consultar Balance:



### **Presentación:**

[https://www.canva.com/design/DAG2NnQKlg8/embo3WFwtf9i3241LQNmg/edit?utm\\_content=DAG2NnQKlg8&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAG2NnQKlg8/embo3WFwtf9i3241LQNmg/edit?utm_content=DAG2NnQKlg8&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

### **Enlaces del video:**

#### Video de la presentacion :

[https://drive.google.com/file/d/1ATQUdn3PjDYxMvjHG0SBJQpsmfvLG2V\\_/view?usp=drive\\_link](https://drive.google.com/file/d/1ATQUdn3PjDYxMvjHG0SBJQpsmfvLG2V_/view?usp=drive_link)

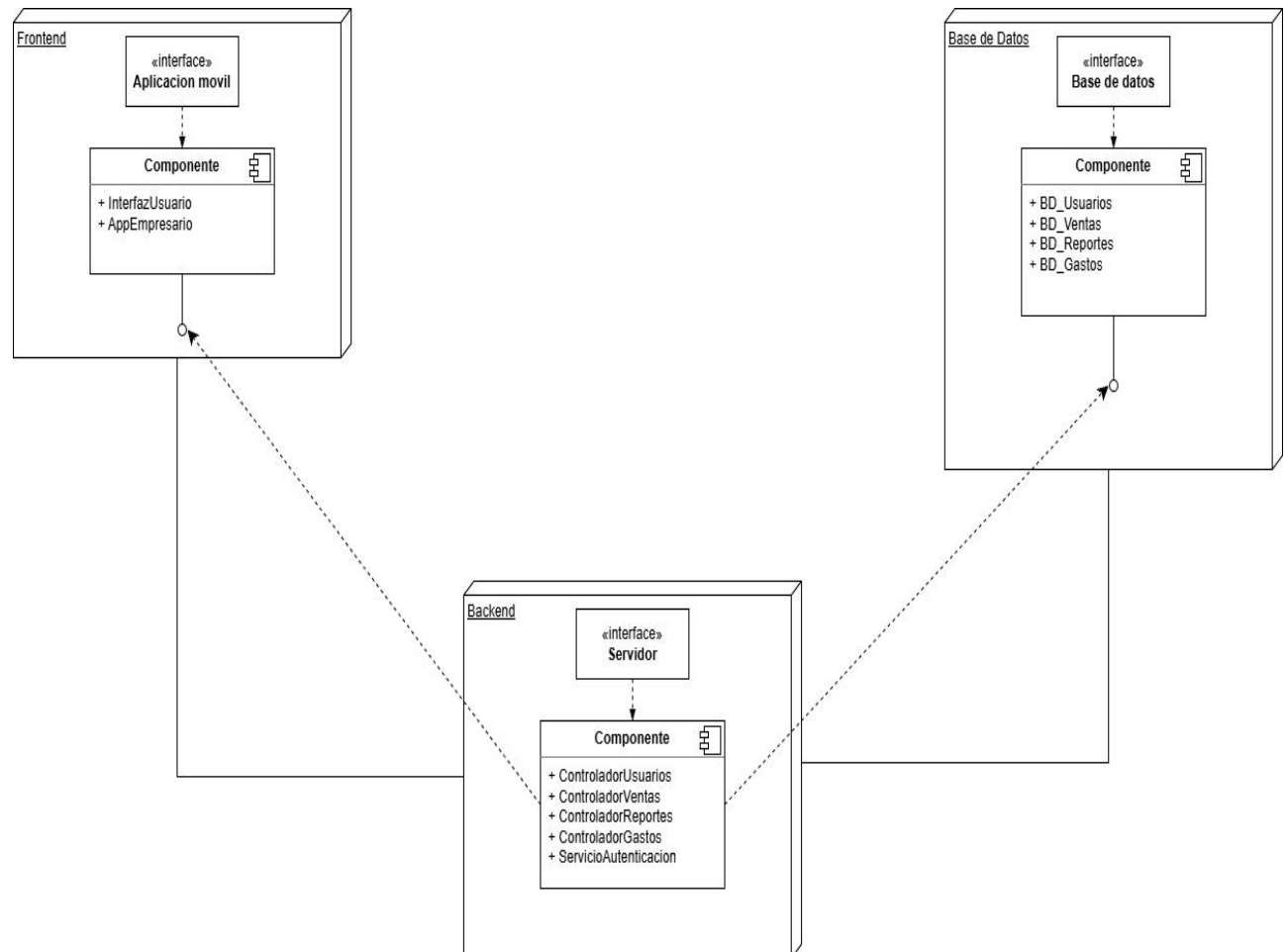
#### Video complementario:

Drive:

<https://drive.google.com/file/d/1dbtfnCWA3WUHYtnX1ggWxPTYbv9oyJOG/view?usp=sharing>

# Diagrama de Componentes y Despliegue del Sistema

## 1. Diagrama de Componentes:



## 3. Explicación:

Este diagrama representa la arquitectura de despliegue del sistema del proyecto. Muestra cómo se distribuyen los componentes en tres nodos principales: el **dispositivo móvil**, donde se ejecuta la interfaz del usuario; el **servidor backend**, donde se procesan las ventas, reportes, gasto, socios y autenticación; y el **servidor de base de datos**, que almacena la información persistente del sistema. Los componentes se comunican mediante solicitudes **HTTP/HTTPS** hacia la API y mediante **JDBC/ORM** hacia la base de datos. Esta arquitectura permite gestionar ventas, reportes y usuarios de manera segura, organizada y escalable.

# Conclusiones Finales

## Fortalezas a futuro

- La app puede convertirse en una herramienta clave para emprendedores que buscan simplicidad y centralización, algo que el mercado aún no cubre completamente.
- Su estructura modular permite agregar nuevas funciones (facturación, marketing, inventarios avanzados) sin necesidad de rehacer el sistema.
- La base de usuarios inicial puede crecer de forma orgánica gracias a su facilidad de uso y su enfoque práctico.
- Al estar pensada para dispositivos móviles, facilita una adopción masiva en un mercado donde el celular es la herramienta principal de trabajo.

## Oportunidades a futuro

- El crecimiento constante del ecosistema emprendedor abre la posibilidad de expandir la app a nuevos perfiles de usuarios (artesanos, servicios técnicos, feriantes, comercios pequeños).
- Integraciones con billeteras virtuales, marketplaces o sistemas de facturación electrónica aumentarían su valor.
- Posibilidad de alianzas con incubadoras, municipios, cámaras de comercio o cursos de oficios.
- El modelo freemium permitiría atraer usuarios y, con el tiempo, sumar funciones premium de valor real.

## Debilidades a futuro

- Requerirá inversión sostenida en mantenimiento, actualizaciones técnicas y soporte, algo que puede ser difícil para un equipo pequeño.
- Para escalar, será necesario profesionalizar la estructura técnica (servidores, seguridad, UX) lo que implica costos crecientes.

- Algunos emprendedores podrían abandonar la app si la perciben “una tarea más” o si no la integran en sus rutinas.
- La curva de aprendizaje puede seguir siendo un obstáculo para usuarios con poca familiaridad digital.

### **Amenazas a futuro**

- Competidores más grandes con más presupuesto podrían lanzar herramientas similares y captar rápidamente el mercado.
- Cambios en regulaciones fiscales podrían exigir actualizaciones rápidas y costosas.
- Crisis económicas o inestabilidad pueden reducir el interés de los emprendedores en pagar por herramientas digitales.
- La rápida evolución tecnológica puede hacer que ciertas funciones queden obsoletas si no se actualizan a tiempo (IA, automatización, integraciones).

### **Conclusión del equipo:**

Este proyecto nos permitió entender todo el proceso de crear una solución real para un problema que muchos emprendedores viven todos los días: la falta de organización y el desorden en la administración del negocio. A partir de esa necesidad, diseñamos una app simple y accesible que centraliza ventas, gastos, clientes y tareas en un solo lugar. Durante el trabajo fuimos armando requerimientos, casos de uso, diagramas y mockups que nos ayudaron a visualizar cómo funcionaría el sistema y qué necesitaba cada tipo de usuario. También analizamos riesgos, oportunidades y el contexto donde la app se usaría, lo que nos permitió tomar decisiones más realistas. En general, el proyecto nos sirvió para entender cómo se planifica y estructura una solución tecnológica desde cero, y cómo la gestión de proyectos ayuda a ordenar el proceso y transformar una idea en algo concreto y útil. Sentimos que lo desarrollado es coherente, aplicable y con potencial para seguir creciendo en el futuro.