

Boss Budget		
<b>Código:</b> BB-DEA-001	<b>Versión:</b> 1.0	<b>Fecha:</b> Septiembre 2025

---

## **Documento de Especificación de Arquitectura**

---

Realizado por:  
**David Alejandro Jáuregui Merchán**  
**Jorge Andrés Ballén Ramírez**  
**Dayana Andrea Escobar Medina**  
**Brayan Steven López**

Boss Budget		
Código: BB-DEA-001	Versión: 1.0	Fecha: Septiembre 2025

***HISTORIAL DE REVISIONES***

Fecha	Versión	Autor	Descripción	Revisado Por
-	-	-	-	

Boss Budget		
Código:	Versión:	Fecha:
BB-DEA-001	1.0	Septiembre 2025

## Contenido

### Tabla de contenido

1.	Documento de Arquitectura de Software.....	4
1.1.	Introducción	4
1.2.	Propósito	4
1.3.	Alcance	4
1.4.	Referencias	5
1.5.	Definiciones acrónimos y abreviaciones	5
2.	Generalidades del Proyecto .....	6
2.1.	Problema a Resolver	6
2.2.	Descripción General del Sistema a Desarrollar	6
2.2.1.	Plataformas y Diseño.....	7
2.2.2.	Funcionalidades clave .....	7
2.2.3.	Componentes fuera del alcance .....	7
2.2.4.	Arquitectura y tecnologías .....	7
2.2.5.	Características generales de calidad.....	7
2.2.6.	Alcance de implementación .....	8
2.3.	Identificación de los Stakeholders y sus responsabilidades	8
3.	Vistas de la arquitectura .....	9
3.1.	Vista de Casos de Uso	9
3.2.	Vista de Procesos	10
3.2.1.	Diagrama de Actividades.....	10
	Caso de uso (Registrarse): .....	10
3.2.2.	Caso de uso (Creación presupuesto): .....	10
3.2.3.	Caso de uso (Reportes usuario): .....	11
3.2.4.	Caso de uso (Pago de Suscripción): .....	11
3.3.	Vista Lógica	12
3.3.1.	Diagramas – Clases .....	12
3.4.	Vista de Implementación	13
3.4.1.	Diagrama de Paquetes .....	13
3.5.	Vista de Despliegue	13
3.5.1.	Diagrama de despliegue .....	13
4.	Arquitectura en capas.....	14
4.1.	Capa de presentación	14
4.2.	Patrones	14
4.3.	Plataforma	14
5.	Vista de Datos .....	15
5.1.	Modelo Relacional	15
6.	Definición de Interfaces de Usuario.....	16
6.1.	Mapa de navegación	16
7.	Características Generales de Calidad .....	19
8.	Stack Tecnológico .....	20

Boss Budget		
Código:	Versión:	Fecha:
BB-DEA-001	1.0	Septiembre 2025

## 1. Documento de Arquitectura de Software

### 1.1. Introducción

El propósito principal de este DEA es definir el alcance del proyecto *Boss-Budget*, especificando los objetivos, limitaciones y entregables esperados de la aplicación móvil. Este documento establece los lineamientos generales que guiarán al equipo de desarrollo, asegurando que el sistema responda a los requisitos funcionales y no funcionales identificados en la ERS. Asimismo, se busca garantizar que la solución propuesta sea robusta, escalable y segura, ofreciendo una experiencia de usuario óptima y cumpliendo con los estándares de calidad definidos.

### 1.2. Propósito

El propósito principal de este documento de arquitectura de software(DAS) es plasmar y definir la estructura técnica de la aplicación móvil Boss Budget, asegurando que el diseño propuesto soporte de manera eficaz los requisitos funcionales y no funcionales previamente establecidos en la Especificación de requisitos de software (ERS).

Este documento servirá como guía fundamental para el equipo de desarrollo, facilitando la implementación de una solución robusta, escalable, segura e intuitiva. Asimismo, se busca garantizar que la plataforma no solo cumpla con las expectativas en cuanto a usabilidad, rendimiento y seguridad, sino que también ofrezca una experiencia óptima para el usuario en todo momento.

### 1.3. Alcance

La aplicación Boss Budget se concibe como una herramienta móvil multiplataforma, para dispositivos IOS y Android. Su alcance se centra en asistir a los usuarios en la mejora de su vida financiera mediante funcionalidades que promueven el control, la organización y la toma de decisiones informadas.

Las funcionalidades y la forma de esta versión incluyen:

- Registro manual de transacciones: Ingreso de gatos, ingresos y ahorros.
- Recordatorios de pago recurrentes: Gestión de facturas, prestamos o tarjetas de crédito, ayudando a los usuarios a mantenerse al día y evitar cargos por retraso.
- Visualización y análisis de datos: Generación de gráficos y estadísticas sobre patrones e ingresos, ofreciendo una visión clara de la situación financiera.
- Extracción automática de datos financieros(opcional): Lectura de notificaciones de mensajes de texto (ej. pagos de nequi) para registrar transacciones automáticamente, optimizando el seguimiento en tiempo

Boss Budget		
Código:	Versión:	Fecha:
BB-DEA-001	1.0	Septiembre 2025

real. Cabe resaltar que la aplicación nunca operará directamente sobre las finanzas del usuario, sino únicamente sobre la información registrada.

- Personalización del perfil financiero: Visualización del progreso hacia metas de ahorro, alertas sobre límites presupuestarios y recomendaciones personalizadas.

El proyecto se enfocará inicialmente en un despliegue a nivel nacional, con la proyección de una futura internacionalización en función de la acogida del producto y su valor innovador.

**Fuera del alcance del proyecto:**

- Integración directa con APIs de entidades bancarias.
- Asesoría financiera personalizada por parte de expertos humanos.
- Gestión de criptomonedas u otros activos digitales.

#### 1.4. Referencias

1. Documento de Especificación de Requerimientos no funcionales.
2. Documento de Visión del Proyecto.
3. Plan de Proyecto del Sistema

#### 1.5. Definiciones acrónimos y abreviaciones

**ARQUITECTURA DE SOFTWARE:** conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al

Comportamiento del software en tiempo de ejecución. Naturalmente este diseño

Arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

**DESCRIPCION DE ARQUITECTURA:** colección de productos de documentación.

**VISTAS:** es una representación de un área de interés o perspectiva del sistema en alto nivel.

**TIPOS DE VISTAS:** especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

**STAKEHOLDER:** Individuo, equipo u organización con intereses relativos al sistema.

Boss Budget		
Código:	Versión:	Fecha:
BB-DEA-001	1.0	Septiembre 2025

## 2. Generalidades del Proyecto

### 2.1. Problema a Resolver

La aplicación de Boss Budget buscar dar solución a la gestión ineficaz y superficial de las finanzas personales, un problema común en la vida cotidiana. En muchos casos, las finanzas son tratadas de manera transitoria y sin la importancia necesaria, lo que deriva en una administración deficiente de los ingresos.

Las consecuencias de esta situación incluyen:

- Solicitud constante de préstamos.
- Enfoque exclusivo en la subsistencia económica.
- Dificultad para generar ahorros sostenibles.
- Incapacidad para cumplir metas financieras personales.

Este problema no solo afecta a nivel individual, sino que también tiene repercusiones colectivas. Ya que una población con baja cultura financiera limita el desarrollo económico sostenible y la estabilidad social.

En este contexto, el objetivo de Boss Budget es ofrecer una solución práctica que permita a los usuarios mejorar su situación financiera mediante un control ordenado y completo de sus recursos, contribuyendo a la adopción de mejores hábitos financieros y a un mayor bienestar económico.

### 2.2. Descripción General del Sistema a Desarrollar

El sistema a desarrollar, Boss Budget es una aplicación móvil multiplataforma y una herramienta web que tiene como propósito principal plasmar y definir la estructura técnica para la gestión eficaz y ordenada de las finanzas personales.

Los objetivos fundamentales son:

- Definir la estructura técnica de la aplicación, garantizando el soporte de los requisitos funcionales y no funcionales definidos en la ERS.
- Proveer a los usuarios de herramientas que faciliten la organización financiera, la toma de decisiones informadas y la mejora de hábitos económicos.
- Servir como guía de referencia para el equipo de desarrollo, facilitando la implementación de una solución robusta, escalable, segura e intuitiva, con altos estándares de usabilidad, rendimiento y seguridad.

Boss Budget		
Código:	Versión:	Fecha:
BB-DEA-001	1.0	Septiembre 2025

### 2.2.1. Plataformas y Diseño

- **Plataformas:** Aplicación móvil multiplataforma para dispositivos IOS y Android, y acceso vía navegador web.
- **Diseño:** Interfaz responsive, accesible y dinámica, implementación de JavaScript, Node.js y MySQL Workbench.

### 2.2.2. Funcionalidades clave

- **Registro manual de transacciones:** Ingreso de gastos, ingresos y ahorros.
- **Recordatorios de pagos recurrentes:** Gestión de facturas, préstamos p tarjetas de crédito para evitar cargos por retraso.
- **Visualización y análisis de datos:** Gráficos y estadísticas sobre patrones de ingresos y gastos.
- **Extracción automática de datos financieros:** Lectura de notificaciones con autorización del usuario.
- **Personalización del perfil financiero:** Metas de ahorro, alertas sobre límites presupuestarios y recomendaciones personalizadas.

### 2.2.3. Componentes fuera del alcance

No se incluye en esta versión el manejo de criptomonedas u otros activos digitales, no se tendrá integración de APIs de entidades bancarias y tampoco se aplicará asesoría personalizada financiera por expertos humanos.

### 2.2.4. Arquitectura y tecnologías

#### Arquitectura en capas

- **Presentación:** Javascript, React.
- **Lógica de aplicación:** Node.js con Express.
- **Datos:** MySQL Worchbeanch.

#### Patrones de diseño

- MVC (Modelo Vista Controlador).
- DAO (Data Access Object).
- Plantillas dinámicas (EJS).

#### Entorno de desarrollo

Visual Studio Code, MySQL Workbench, Node.js.

#### Despliegue

Inicialmente en entornos locales.

### 2.2.5. Características generales de calidad

El sistema se diseñará considerando atributos de calidad fundamentales:

- **Accesibilidad:** Compatibilidad con lectores de pantalla y diseño responsive.
- **Adaptabilidad:** Arquitectura modular y extensible.
- **Disponibilidad:** Acceso continuo con recuperación ante fallos.

Boss Budget		
Código: BB-DEA-001	Versión: 1.0	Fecha: Septiembre 2025

- **Compatibilidad:** Funcionamiento en navegadores y SO modernos.
- **Confidencialidad:** controles de acceso, cifrado y autenticación segura.
- **Eficiencia:** Optimación de recursos y consultas a base de datos.
- **Operabilidad:** Interfaz intuitiva y retroalimentación.
- **Redundancia:** Respaldo de datos y duplicación de componentes críticos.
- **Integridad:** Validación de reglas y transacciones.
- **Robustez:** Estabilidad en condiciones adversas.

#### 2.2.6. Alcance de implementación

El proyecto se desplegará inicialmente a nivel nacional, con proyección futura a expansión internacional de acuerdo con la acogida del producto y la escabilidad de la arquitectura diseñada.

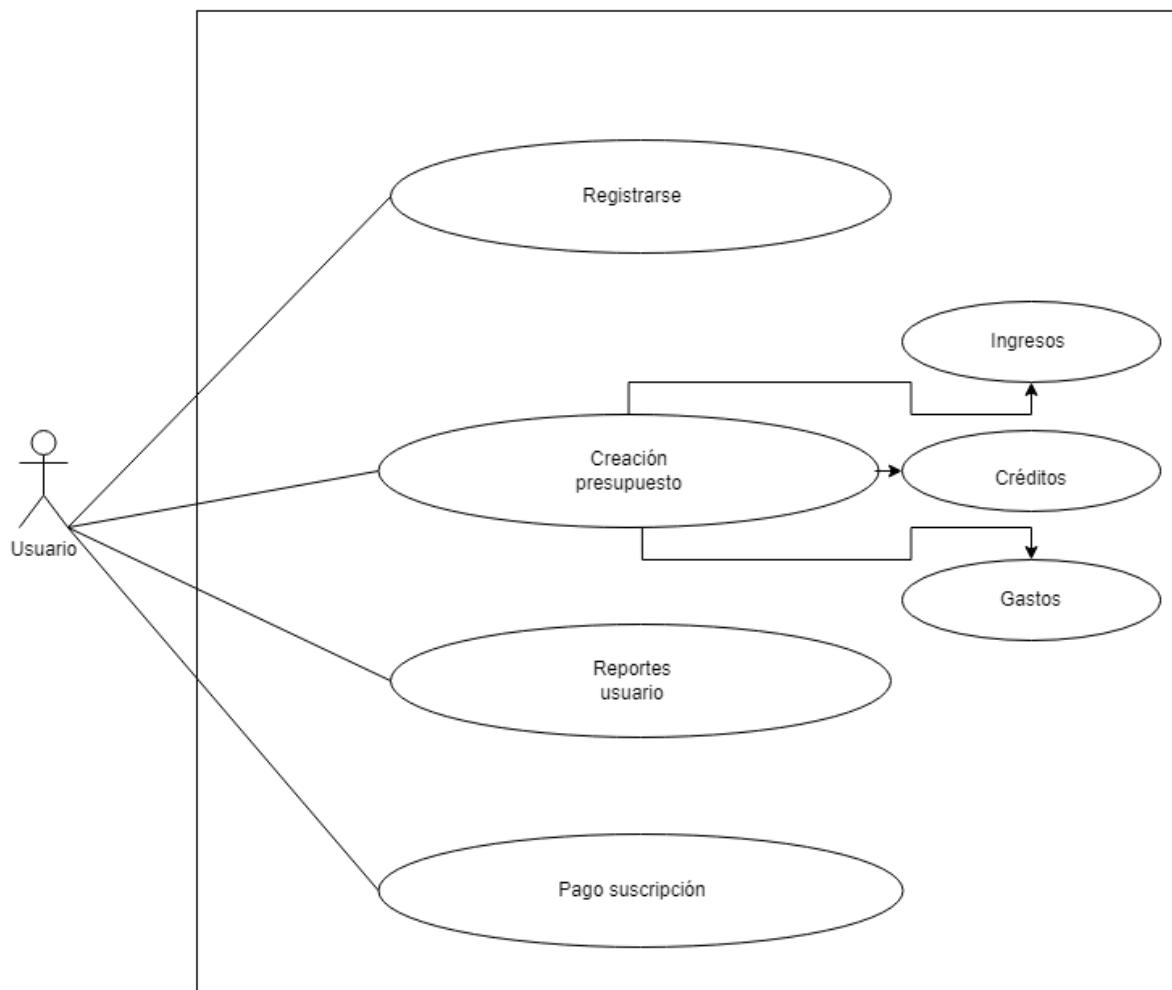
### 2.3. Identificación de los Stakeholders y sus responsabilidades

STAKEHOLDER	DESCRIPCIÓN	ESCENARIO	Caso de Uso
Administrador	Es el responsable de la gestión del sistema y las cuentas de usuarios.	<ul style="list-style-type: none"> <li>• Escenario de operación</li> <li>• Escenario de Soporte</li> <li>• Escenario de análisis</li> </ul>	<ul style="list-style-type: none"> <li>• Crear y editar cuentas de usuarios</li> <li>• Ajustar parámetros de seguridad</li> <li>• Gestionar accesos</li> </ul>
Cliente	Es la persona que Toma decisiones basadas en los datos presentados por la aplicación	<ul style="list-style-type: none"> <li>• Escenario de escalabilidad</li> <li>• Escenario de Toma de decisiones</li> </ul>	<ul style="list-style-type: none"> <li>• Visualización de reportes ejecutivos</li> <li>• Revisión de estadísticas de ahorros y/o gastos</li> </ul>
Usuario final	Es la persona que utilizará la aplicación para manejar su presupuesto.	<ul style="list-style-type: none"> <li>• Escenario de registro</li> <li>• Escenario de usabilidad</li> <li>• Escenario de personalización</li> </ul>	<ul style="list-style-type: none"> <li>• Añadir un gasto</li> <li>• Registrar un ingreso</li> <li>• Consultar el balance mensual</li> </ul>

Boss Budget		
Código:	Versión:	Fecha:
BB-DEA-001	1.0	Septiembre 2025

### 3. Vistas de la arquitectura

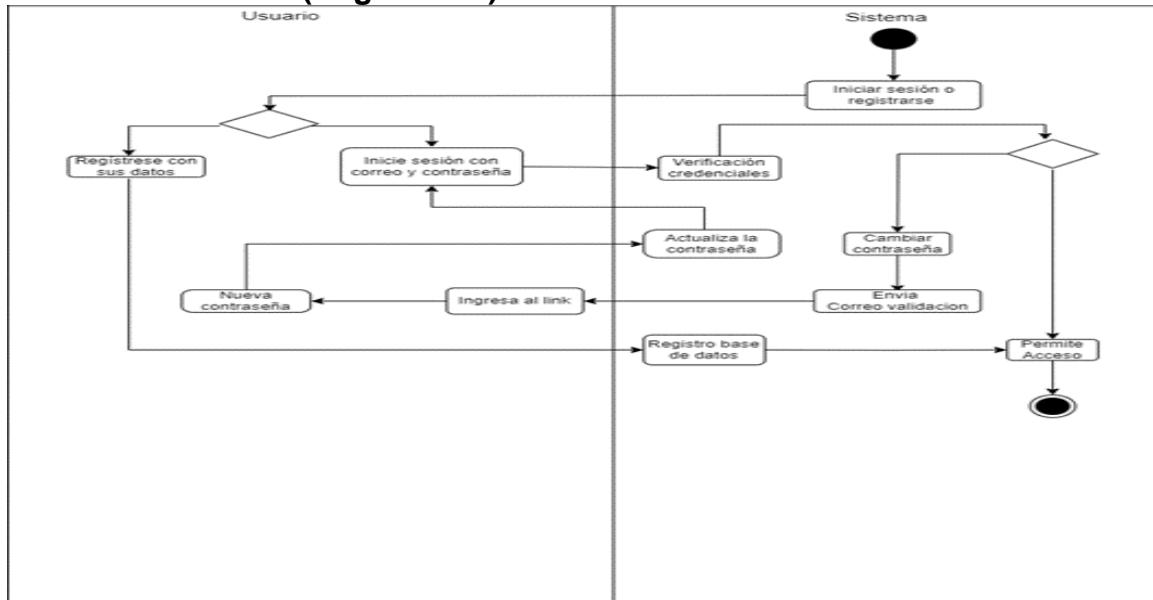
#### 3.1. Vista de Casos de Uso



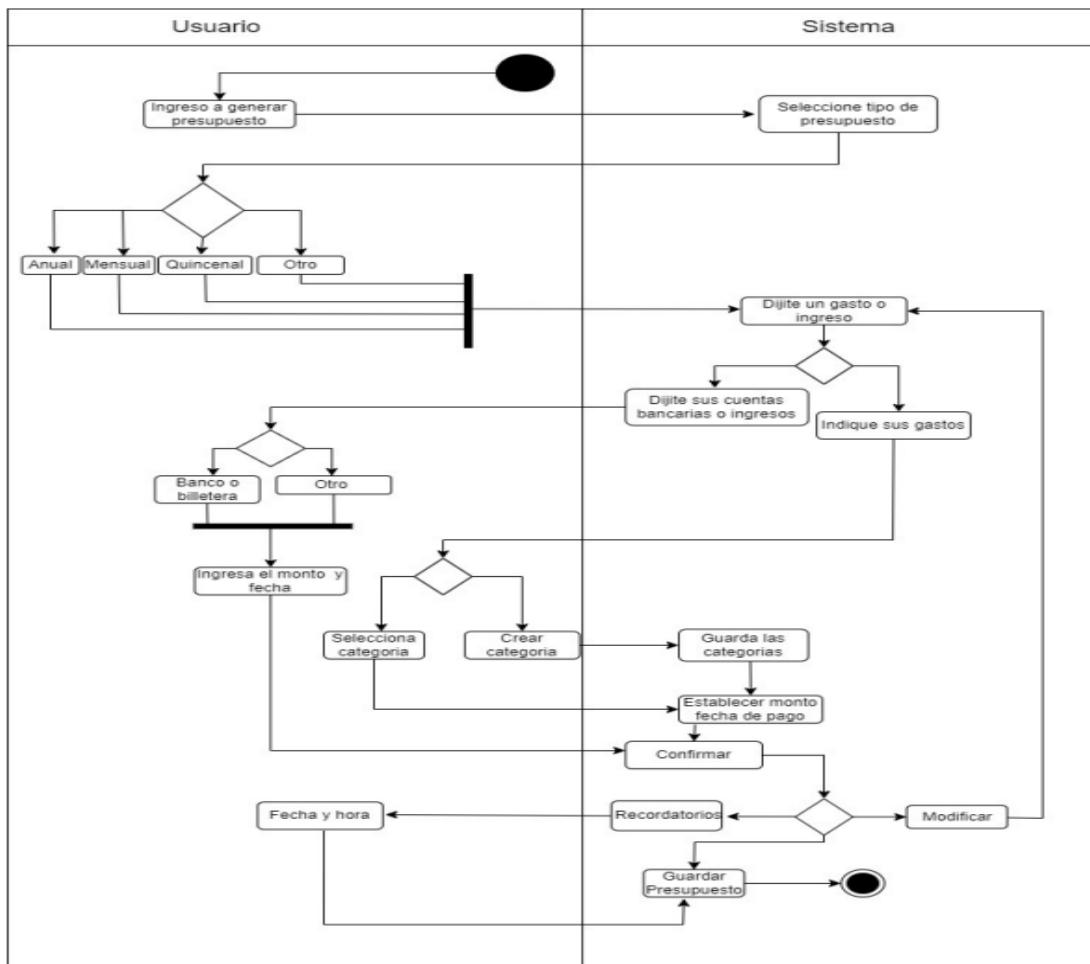
Boss Budget		
Código: BB-DEA-001	Versión: 1.0	Fecha: Septiembre 2025

### 3.2. Vista de Procesos

#### 3.2.1. Diagrama de Actividades Caso de uso (Registrarse):

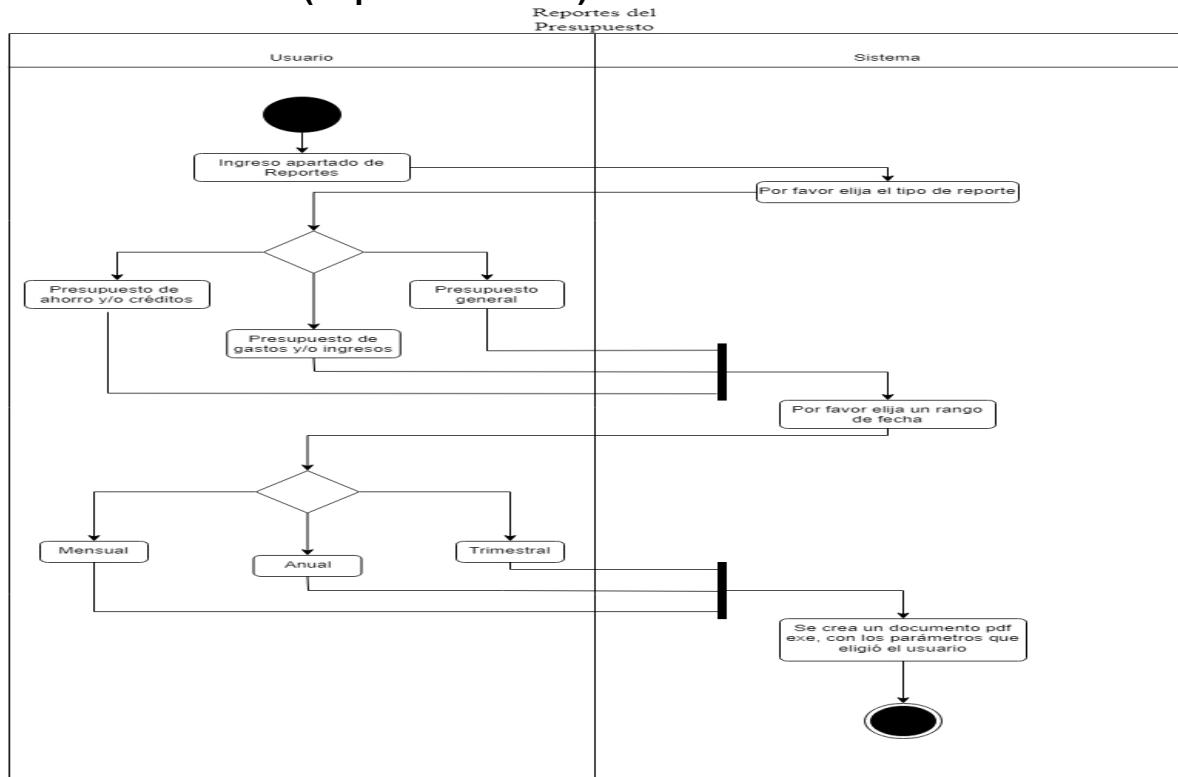


#### 3.2.2. Caso de uso (Creación presupuesto):

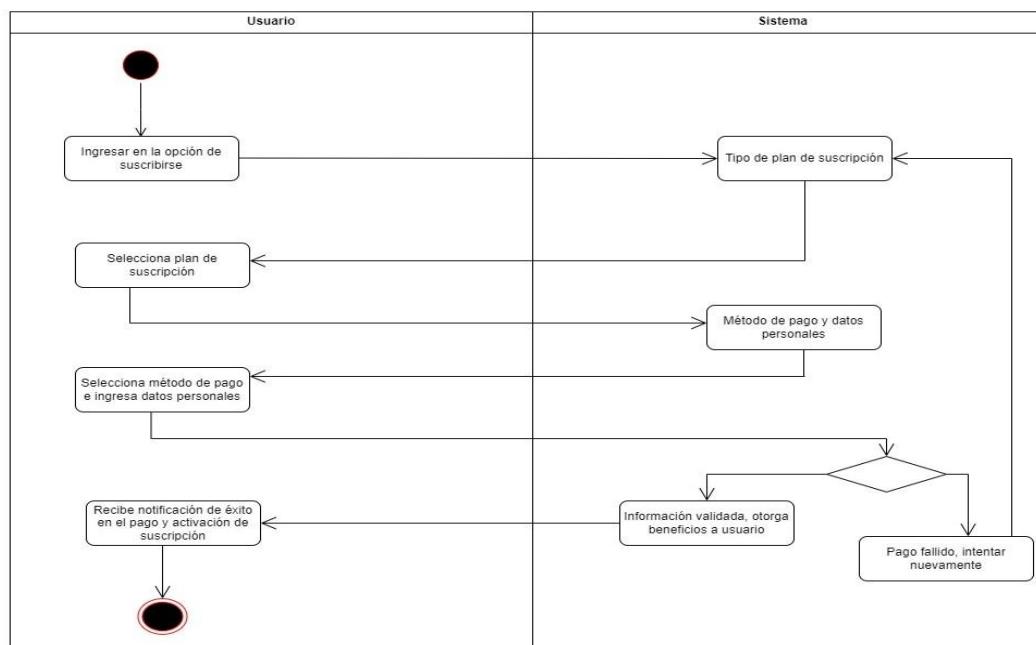


Boss Budget		
Código: BB-DEA-001	Versión: 1.0	Fecha: Septiembre 2025

### 3.2.3. Caso de uso (Reportes usuario):



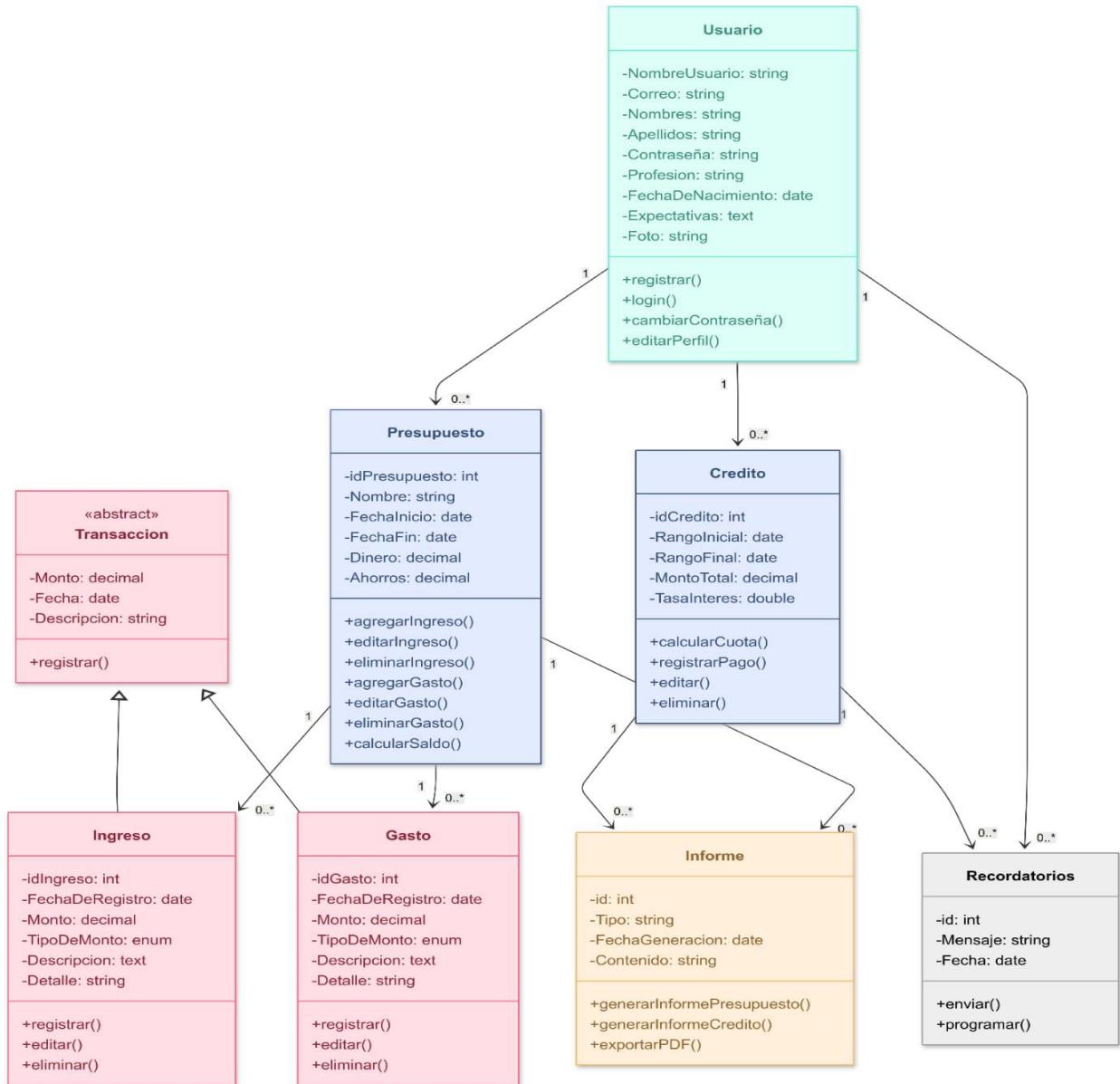
### 3.2.4. Caso de uso (Pago de Suscripción):



Boss Budget		
Código: BB-DEA-001	Versión: 1.0	Fecha: Septiembre 2025

### 3.3. Vista Lógica

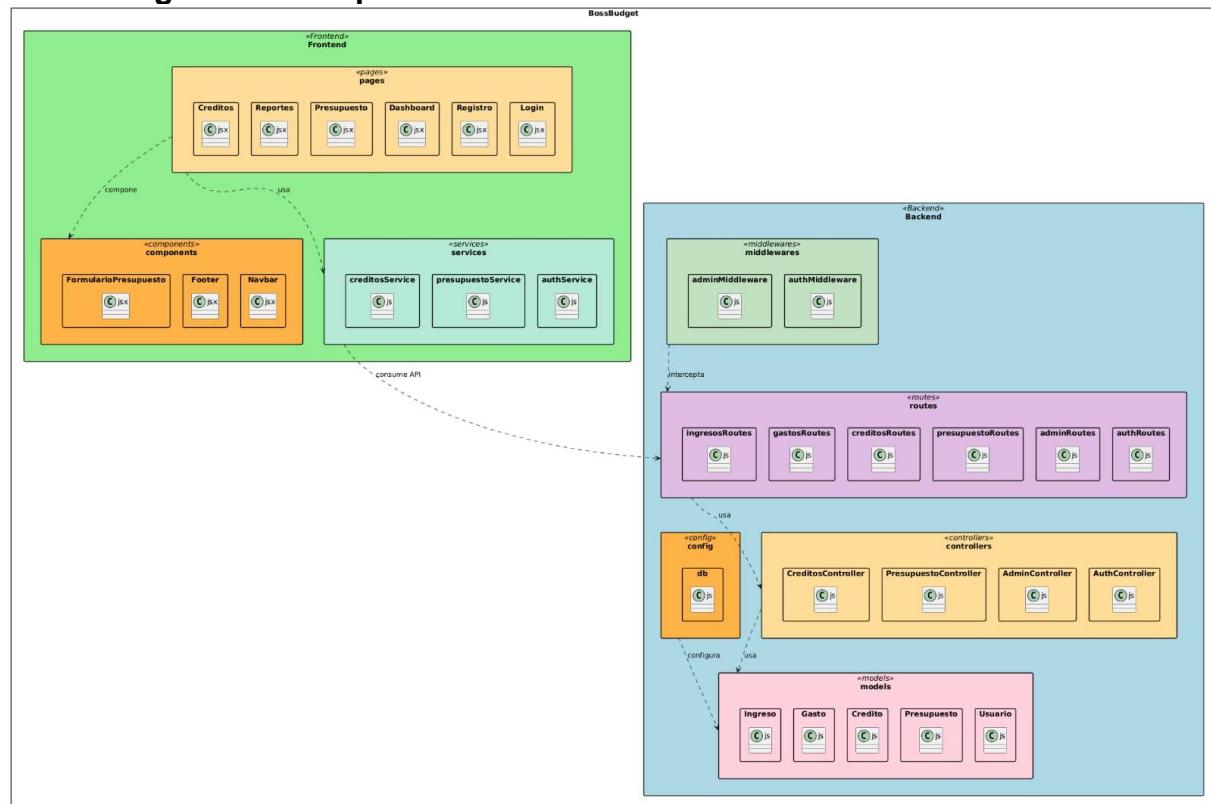
#### 3.3.1. Diagramas – Clases



Boss Budget		
Código: BB-DEA-001	Versión: 1.0	Fecha: Septiembre 2025

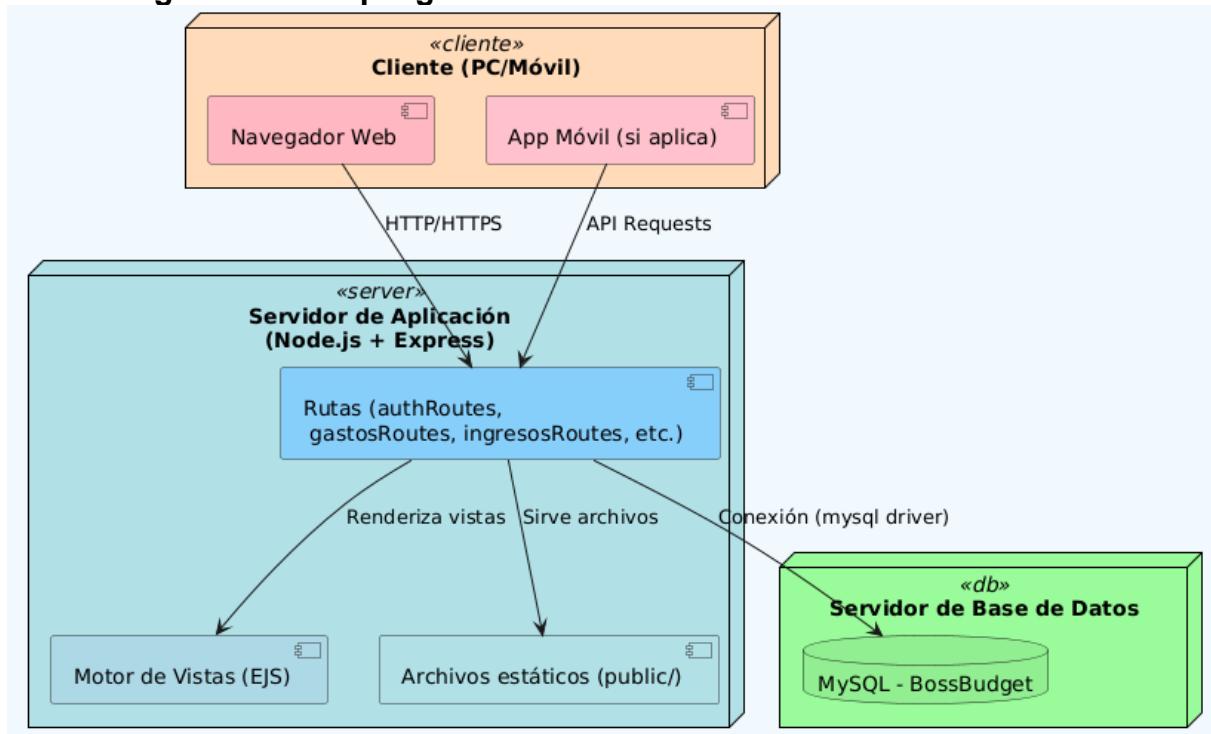
### 3.4. Vista de Implementación

#### 3.4.1. Diagrama de Paquetes



### 3.5. Vista de Despliegue

#### 3.5.1. Diagrama de despliegue



Boss Budget		
Código:	Versión:	Fecha:
BB-DEA-001	1.0	Septiembre 2025

## 4. Arquitectura en capas

La aplicación se desarrolla bajo una arquitectura en capas, lo que permite separar responsabilidades, mejorar la mantenibilidad del sistema y facilitar la escalabilidad futura.

### 4.1. Capa de presentación

Responsable de la interacción con el usuario final, se implementa utilizando HTML, JavaScript, EJS y Tailwind, garantizando un diseño responsivo, accesible y dinámico.

- **Capa de navegación:**

La lógica de la aplicación fue desarrollada en Node.js con Express, procesan las operaciones de registro de usuarios, ingresos y gatos, las validaciones, cálculos de balances y generación de reportes.

- **Capa de datos:**

Es la encargada de la persistencia y consulta de la información, es donde se definen las tablas, relaciones y restricciones necesarias para garantizar la integridad de los datos, esta capa responde a las operaciones CRUD solicitadas para la lógica de la aplicación.

### 4.2. Patrones

La arquitectura aplica el patrón por capas, asignando responsabilidades claras a cada nivel, adicionalmente se emplean:

- **MVC:**

Para separar la lógica de la aplicación, las vistas dinámicas y el control de flujo.

- **DAO:**

Para centralizar las operaciones de acceso a la base de datos y reducir la dependencia directa del SQL en la lógica.

- **Plantillas dinámicas (EJS):**

Permite reutilizar vistas y facilitar la generación de interfaces dinámicas.

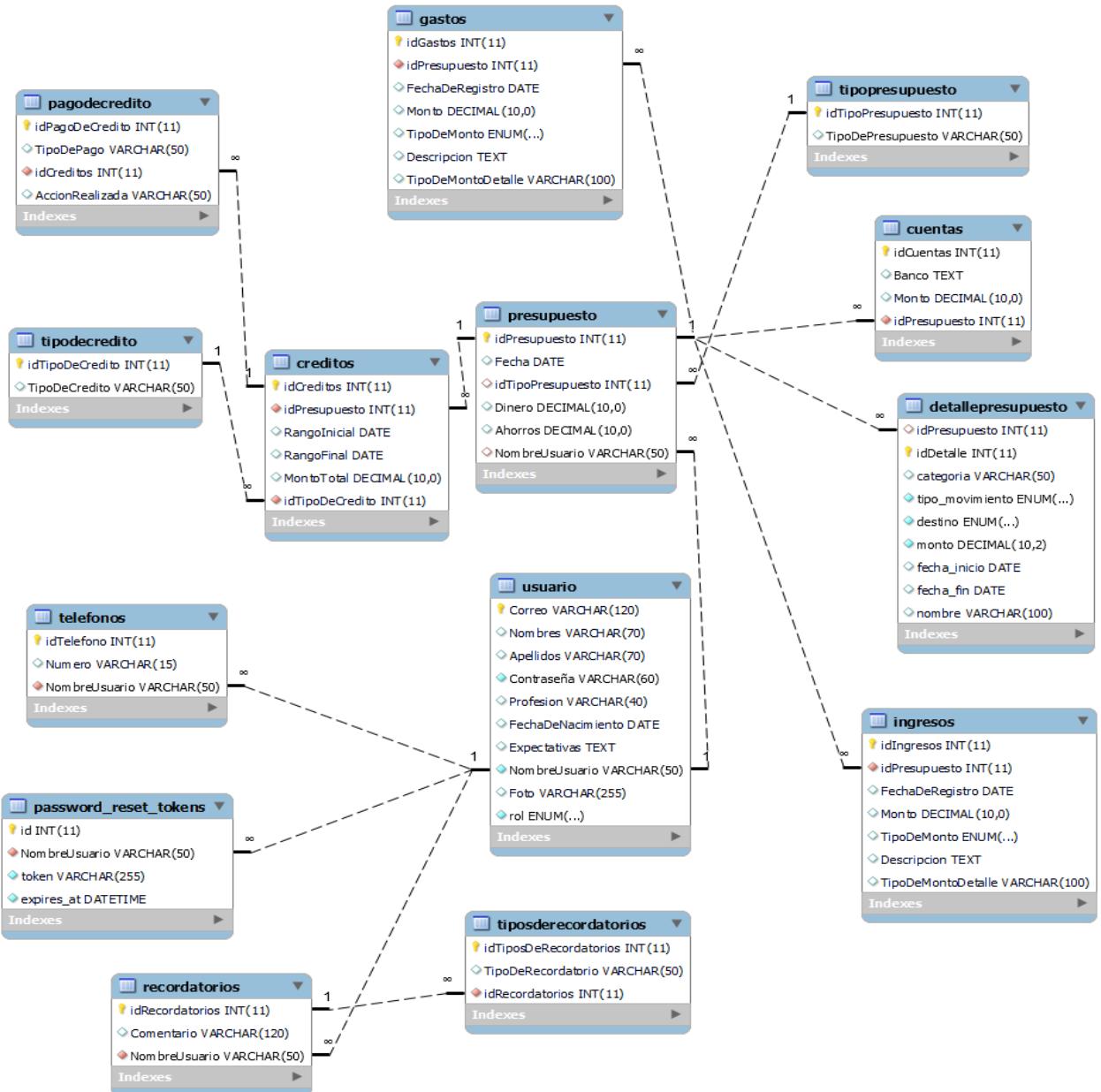
### 4.3. Plataforma

El desarrollo se lleva a cabo en Visual Studio Code como entorno principal, la aplicación se ejecuta sobre Node.js, mientras la base de datos se gestiona con MYSQL Workbench. La aplicación está diseñada para ser desplegada en entornos locales y escalable a plataformas en la nube en caso de requerirse. El acceso se realiza mediante navegadores modernos, asegurando compatibilidad y portabilidad.

Boss Budget		
Código:	Versión:	Fecha:
BB-DEA-001	1.0	Septiembre 2025

## 5. Vista de Datos

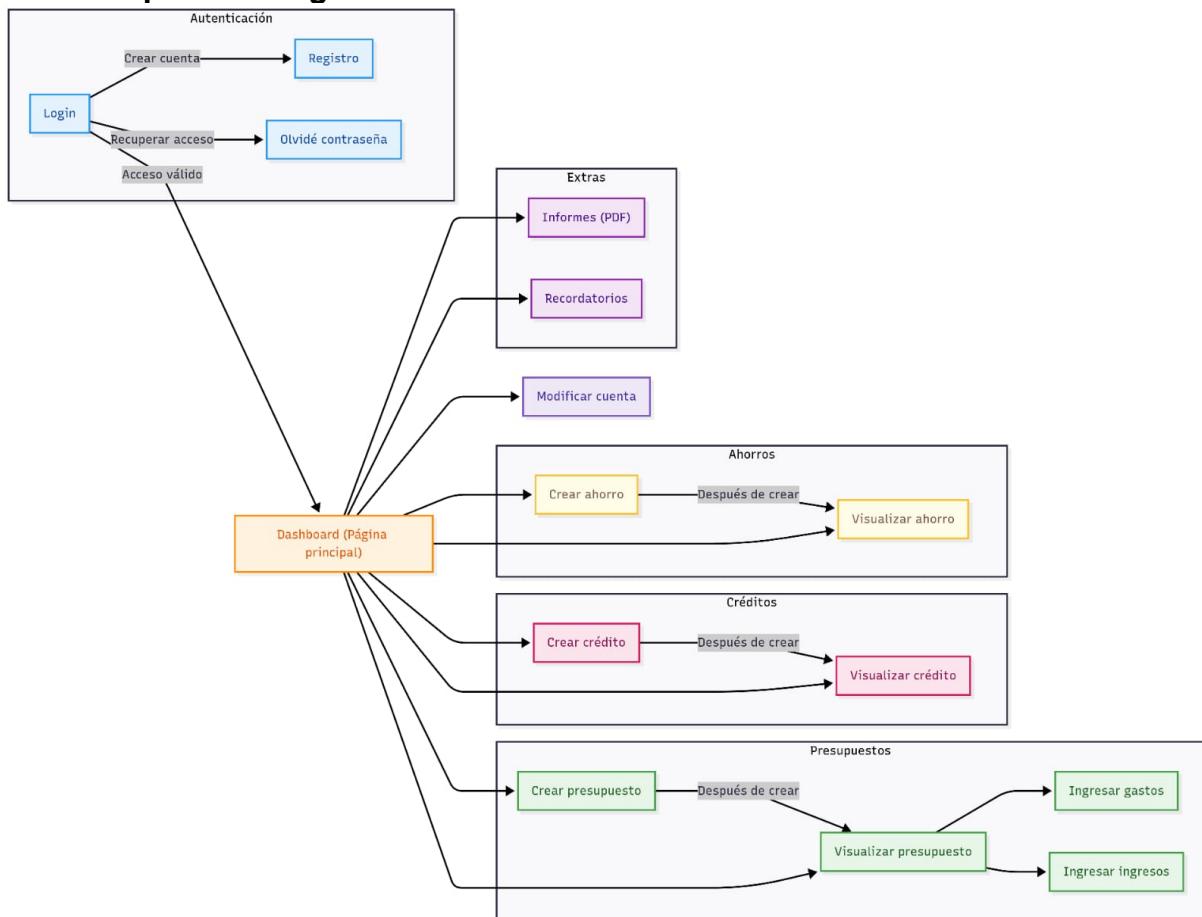
### 5.1. Modelo Relacional



Boss Budget		
Código: BB-DEA-001	Versión: 1.0	Fecha: Septiembre 2025

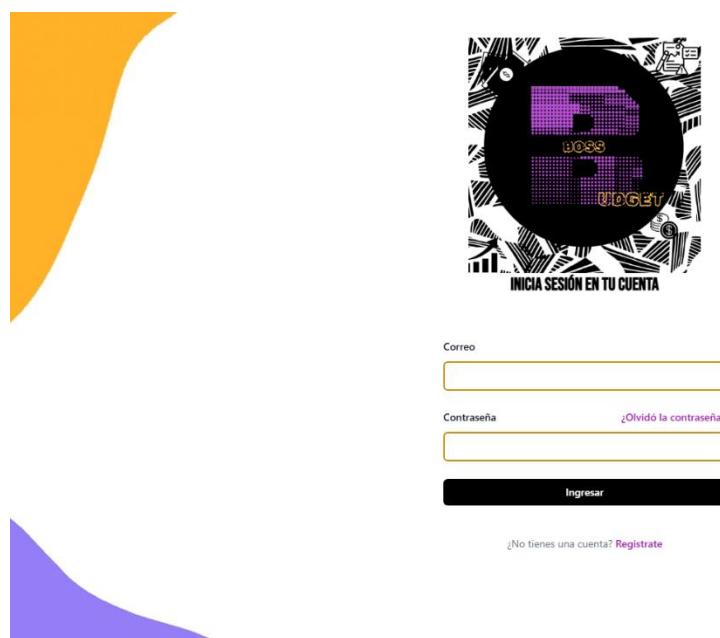
## 6. Definición de Interfaces de Usuario

### 6.1. Mapa de navegación



### 6.2 Demostración de interfaces

- Pantalla de inicio / Registro



Boss Budget		
Código:	Versión:	Fecha:
BB-DEA-001	1.0	Septiembre 2025

**Registro**

Únete a la plataforma de presupuestos más inteligente

Comienza tu viaje hacia la libertad financiera. Esta información se mostrará públicamente.

**Información Personal**

Nombres: Juan Carlos Pérez Gómez  
Apellido: Pérez Gómez  
Nombre de usuario: janesmith123  
Profesión: Ej. Analista Financiero  
Correo electrónico: ejemplo@tucorreo.com  
Contraseña: \*\*\*\*

**Foto de Perfil**

Subir imagen

**Metas Financieras**

Cuéntanos tus objetivos financieros..

**Notificaciones de Presupuesto**

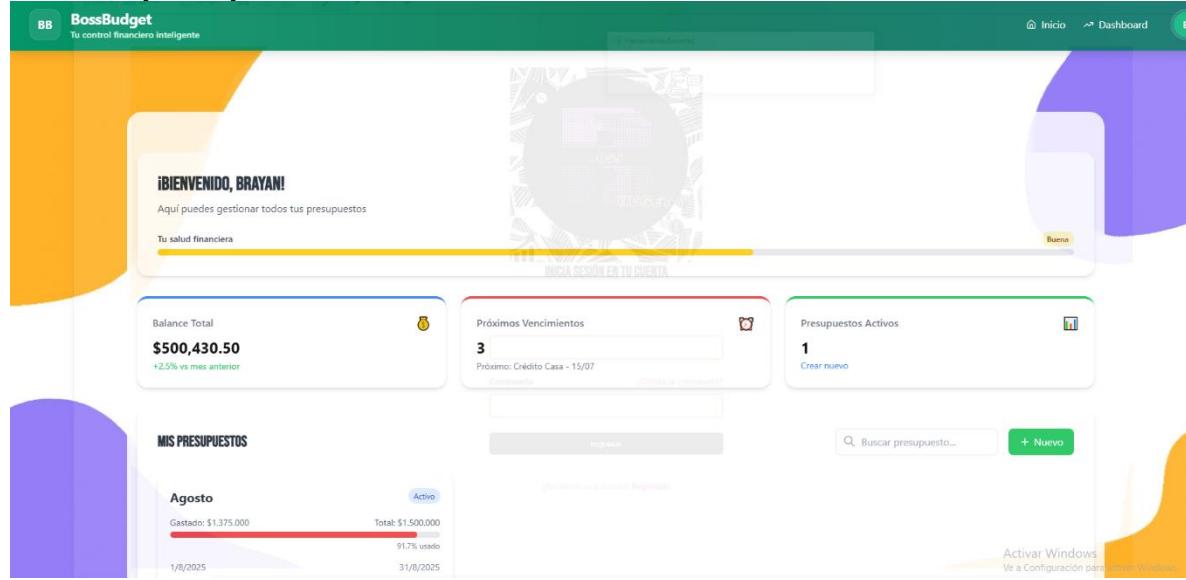
Eige cómo quieras recibir alertas sobre tus finanzas:

**Básico**: Resumen mensual de gastos e ingresos

**Moderado**: Resumen mensual + alertas de límites de gastos

**Completo**: Alertas diarias, semanales y recordatorios de pagos

- **Pantalla principal**

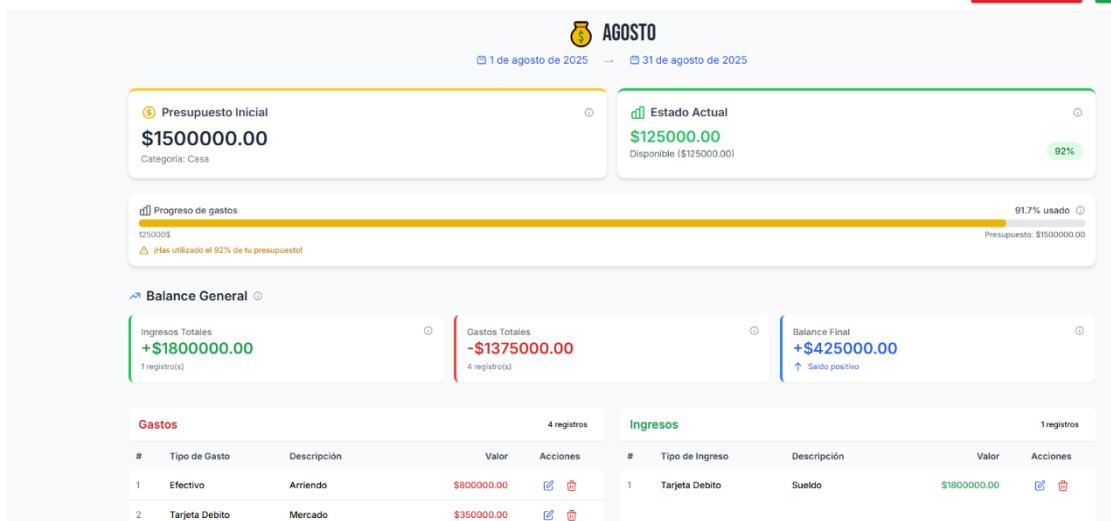


Boss Budget		
Código: BB-DEA-001	Versión: 1.0	Fecha: Septiembre 2025

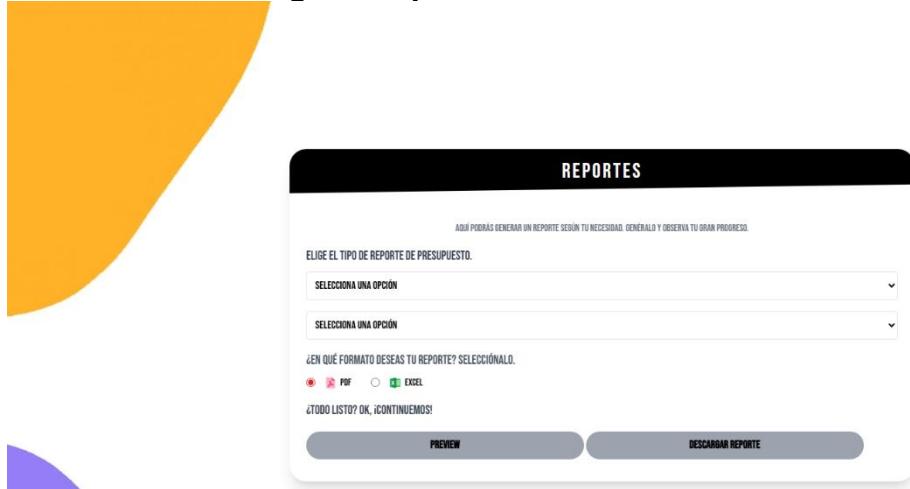
- Gestión de presupuesto

← Volver

– Registrar Gasto + Registrar Ingreso



- Pantalla de descarga de reportes



- Configuración de usuario

Configuración de Cuenta

Personaliza tu perfil y preferencias de presupuesto

Perfil

Nombre: Brayan Apellido: Lopez

Nombre de Usuario: Brayan

Correo Electrónico: brayan123@gmail.com

Profesión: Estudiante

Expectativas: Muchas

Preferencias de Presupuesto

Notificaciones

Activar Windows  
Ve a Configuración para activar Windows

Boss Budget		
Código:	Versión:	Fecha:
BB-DEA-001	1.0	Septiembre 2025

## 7. Características Generales de Calidad

- **Accesibilidad:**  
La aplicación maneja prácticas de accesibilidad que permiten su uso por cualquier tipo de usuarios, se emplean etiquetas semánticas en HTML para que sea compatible con los lectores de pantallas, con un diseño responsive para asegurar la correcta visualización en distintos dispositivos.
- **Adaptabilidad:**  
La aplicación está diseñada para ajustarse a diferentes entornos de uso, su arquitectura modular permite la incorporación de nuevas funcionalidades sin afectar la estabilidad del sistema, su flexibilidad permite su evolución y mantenimiento a lo largo del tiempo.
- **Disponibilidad:**  
La aplicación busca garantizar su acceso continuo y confiable a sus servicios, se implementan prácticas de gestión de errores y recuperación ante fallos que reducen los tiempos de inactividad, así como una arquitectura que permite la ejecución estable en distintos entornos.
- **Compatibilidad:**  
La aplicación está desarrollada siguiendo estándares web que aseguran su correcto funcionamiento en los principales navegadores y sistemas operativos, utilizando tecnologías como HTML y JavaScript altamente soportadas que garantizan una experiencia consistente.
- **Confidencialidad:**  
La aplicación protege la información sensible de los usuarios mediante controles de acceso y manejo seguro de los datos, implementando mecanismos de autenticación para garantizar que solo los usuarios propios accedan a su información.
- **Eficiencia:**  
La aplicación está optimizada para hacer uso adecuado de los recursos disponibles asegurando tiempos de respuesta agiles y un desempeño estable, aplicando buenas prácticas en el diseño de consultas a la base de datos, carga de recursos estáticos y en la ejecución de procesos internos con el fin de reducir el consumo innecesario de memoria y procesamiento.
- **Operabilidad:**  
La aplicación está diseñada para facilitar su uso y administración tanto por los usuarios finales como por el personal de soporte, se implementa una interfaz intuitiva que reduce su complejidad y se

Boss Budget		
Código:	Versión:	Fecha:
BB-DEA-001	1.0	Septiembre 2025

incluyen mensajes de retroalimentación claros ante errores o acciones críticas.

- **Redundancia:**  
La aplicación contempla mecanismos de redundancia para asegurar la continuidad del servicio ante posibles fallos, se prevé la duplicación de componentes críticos y la implementación de respaldos periódicos de la base de datos.
- **Integridad:**  
La aplicación garantiza que la información almacenada y procesada se mantenga completa, coherente y libre de alteraciones no autorizadas, aplicando reglas de validación en las entradas de datos, restricción a la base de datos y controles de transacciones para evitar inconsistencias.
- **Robustez:**  
La aplicación está diseñada para mantener un funcionamiento estable incluso en condiciones adversas, implementando controles de manejo de errores y mecanismos de recuperación ante fallos, lo que reduce la probabilidad de interrupciones críticas.

## 8. Stack Tecnológico

- **JavaScript:**  
Es el lenguaje de programación principal de la aplicación, utilizado en el frontend para que el usuario tenga una integración dinámica y en el backend para manejar la logia del programa y manejo de peticiones HTTP.
- **HTML:**  
Lenguaje de etiquetado utilizado para estructurar la semántica del programa y la integración de contenido que se le presentara al usuario.
- **Tailwind CSS:**  
Framework de CSS utilizado para agilizar el desarrollo de interfaces, contribuyendo con la construcción de diseños responsivos, ayudando a la estandarización de estilos en toda la aplicación.
- **EJS (Embedded JavaScript Templates):**  
Motor de plantillas que permite generar vistas dinámicas, permitiendo la inyección de datos desde el backend a las páginas HTML, reduciendo la complejidad en la generación de contenido dinámico.
- **SQL:**  
Lenguaje de consulta estructurado, utilizado para la gestión y manipulación de datos dentro del sistema, contribuye a las operaciones relacionadas con el CRUD sobre las tablas que manejan los datos de la aplicación asegurando la integridad y consistencia de la información.