

Spendn't

Por:

**Luis Felipe Franco Giraldo
Steven David Osorio Araque**

Materia:

**Aplicaciones y Servicios Web
Grupo:580202009-9**

Profesor:

William Díaz Villegas

Fecha:

13/09/2025

2025

Spendn't

Spend'nt es una plataforma en línea creada para cambiar la forma en que las personas administran su dinero personal. Su propósito fundamental es proporcionar una plataforma completa, fácil de usar y segura que habilite a los usuarios a gestionar plenamente sus recursos financieros a través del registro, análisis y planificación de sus operaciones monetarias. La aplicación enfrenta problemas habituales como la desorganización en la gestión de finanzas, la dificultad para identificar patrones de consumo, la imposibilidad de fijar objetivos de ahorro alcanzables y la falta de alertas eficaces para gastos recurrentes o por venir. En el flujo actual de tareas, los usuarios generalmente recurren a métodos manuales como hojas de cálculo, notas físicas o diversas aplicaciones no conectadas, lo que provoca dispersión de datos, errores humanos y una perspectiva fragmentada de su situación financiera. Spend'nt agrupa todas estas actividades en un solo entorno, automatizando cálculos, ofreciendo alertas en el momento adecuado y produciendo perspectivas integradas mediante herramientas como el calendario financiero y el registro de transacciones.

Objetivo General

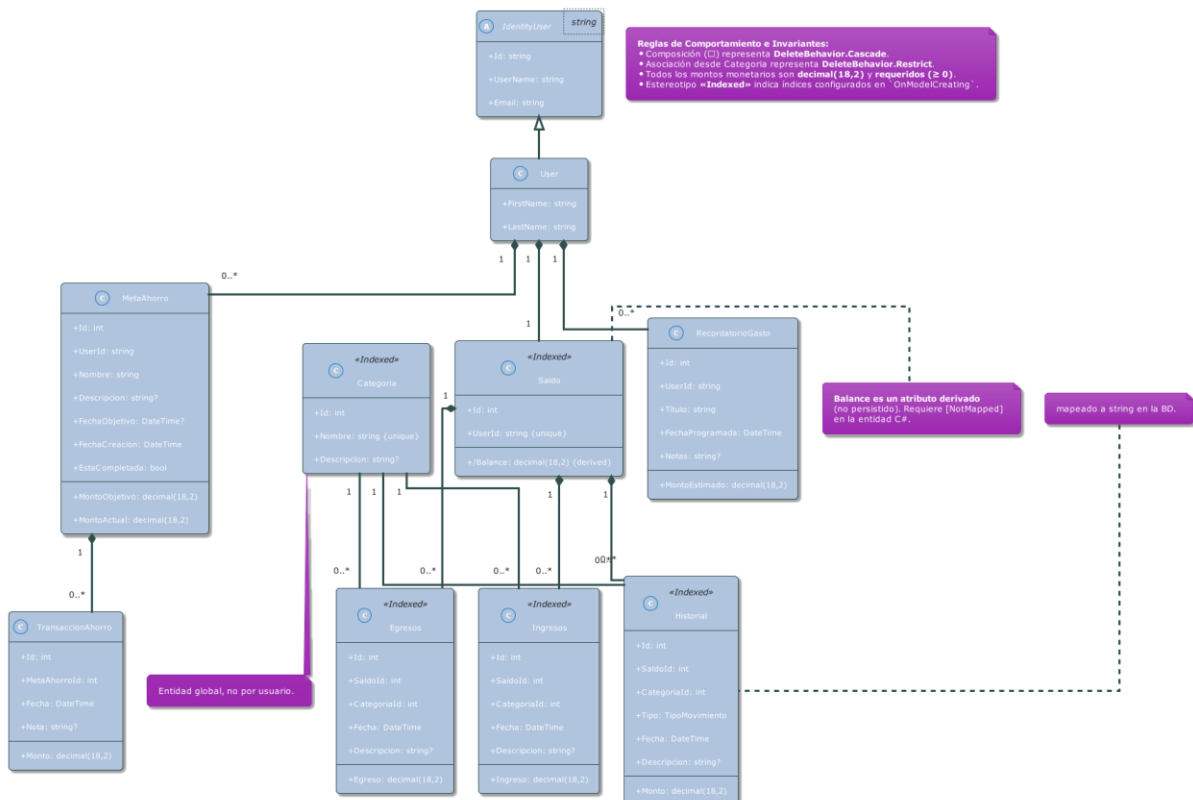
Desarrollar una plataforma en línea integral, segura e intuitiva que centralice la gestión financiera personal, permitiendo a los usuarios tomar el control de sus recursos, identificar patrones de consumo y planificar eficazmente su futuro económico, superando así la fragmentación y los errores de los métodos manuales.

Objetivos Específicos del Proyecto

1. Llevar a cabo un Análisis y Especificación Exhaustiva de RequisitosRealizar una investigación profunda y sistemática para comprender, definir y documentar con precisión todas las necesidades de los usuarios finales, así como las restricciones y reglas de negocio que gobernarán el sistema "Spend'nt". Este proceso tiene como finalidad establecer una base sólida y consensuada para todo el ciclo de desarrollo, asegurando que los requisitos funcionales y no funcionales reflejen fielmente los problemas a resolver y los objetivos a alcanzar, definiendo así el alcance completo y la visión del producto final.
2. Elaborar un Diseño Técnico y Visual Detallado y RobustoCrear un plano integral del sistema que traduzca los requisitos previamente definidos en una arquitectura de software coherente, escalable y segura. Este objetivo implica diseñar la estructura de la base de datos, modelar los componentes lógicos y sus interacciones, y desarrollar prototipos de alta fidelidad que definan una interfaz de usuario intuitiva y una experiencia de usuario (UX) fluida y atractiva. El diseño resultante servirá como una guía técnica precisa para el equipo de implementación, garantizando la calidad, mantenibilidad y eficiencia del producto a construir.
3. Materializar el Diseño en un Producto de Software Funcional y de Alta CalidadConstruir y desarrollar el sistema "Spend'nt" traduciendo las especificaciones de diseño en código fuente limpio, eficiente y mantenible. Este objetivo se centra en la codificación de cada uno de los módulos y funcionalidades, así como en su integración progresiva para conformar una aplicación cohesiva y operativa. El proceso de implementación se guiará por las mejores prácticas de la industria y los estándares de codificación establecidos, con el fin de entregar un producto tangible que sea fiel al diseño y cumpla con todos los requisitos funcionales.
4. Ejecutar un Proceso de Verificación y Validación Riguroso e IntegralAsegurar la calidad, fiabilidad y corrección del software desarrollado a través de un plan de pruebas sistemático que abarque todos los aspectos del sistema. El propósito es identificar, documentar y corregir defectos para garantizar no solo que el software está libre de errores críticos, sino también que cumple fielmente con todos los requisitos funcionales, de rendimiento, seguridad y usabilidad especificados en la fase de análisis. Esta validación exhaustiva es crucial para certificar que el producto está listo para ser entregado a los usuarios finales.

5. Gestionar la Implantación Exitosa del Sistema en el Entorno de ProducciónOrquestar el despliegue controlado y sistemático de la versión final y validada del software, poniéndolo a disposición de los usuarios en un entorno operativo real. Este objetivo incluye la configuración de la infraestructura necesaria, la migración de datos si fuera pertinente, y la ejecución de un plan de lanzamiento que asegure una transición fluida. La meta final es completar la puesta en marcha de la plataforma "Spend'nt", garantizando que sea estable, segura y completamente accesible para su audiencia objetivo.

Diagrama de clases



Entidades

IdentityUser

Usuario base de ASP.NET Identity.

Clave primaria: Id (string).

Aporta autenticación, roles y seguridad.

User

Extiende IdentityUser con datos de perfil: FirstName, LastName.

Relaciones:

1-1 con Saldo (borrado en cascada).

1-* con MetaAhorro y RecordatorioGasto (cascada).

Propósito: representar al titular de la información financiera.

Saldo

Resume el estado financiero del usuario.

Atributos: Id, UserId. /Balance es derivado (no persistido).

Relaciones:

1-* con Ingresos, Egresos y Historial (cascada).

Invariante: se calcula como $\Sigma \text{Ingresos} - \Sigma \text{Egresos}$.

Categoria

Catálogo global de clasificación (no por usuario).

Atributos: Id, Nombre, Descripcion.

Relaciones: 1-* con Ingresos, Egresos, Historial.

Borrado: Restrict para preservar historiales.

Sugerido: índice único en Nombre.

Ingresos

Entradas de dinero del usuario.

Atributos: Id, SaldoId, CategoriaId, Fecha, Descripcion?, Ingreso: decimal(18,2).

Relaciones: pertenece a Saldo y a Categoria.

Reglas: $\text{Ingreso} \geq 0$. Índice recomendado (SaldoId, Fecha).

Egresos

Salidas de dinero del usuario.

Atributos: Id, SaldoId, CategoriaId, Fecha, Descripcion?, Egreso: decimal(18,2).

Relaciones: pertenece a Saldo y a Categoria.

Reglas: Egreso ≥ 0 . Índice recomendado (SaldoId, Fecha).

Historial

Log desnormalizado para reportes rápidos de movimientos.

Atributos: Id, SaldoId, CategoriaId, Tipo (string o enum {Ingreso,Egreso}), Fecha, Descripcion?,
Monto: decimal(18,2).

Relaciones: pertenece a Saldo y Categoria.

Índice recomendado (SaldoId, Fecha).

MetaAhorro

Objetivo de ahorro del usuario.

Atributos: Id, UserId, Nombre, Descripcion?, FechaObjetivo?, FechaCreacion, EstaCompletada,
MontoObjetivo: decimal(18,2), MontoActual: decimal(18,2).

Relaciones: 1-* con TransaccionAhorro (cascada).

Uso: planificar y seguir metas personales.

TransaccionAhorro

Movimientos internos de una meta.

Atributos: Id, MetaAhorroId, Fecha, Nota?, Monto: decimal(18,2).

Relaciones: pertenece a MetaAhorro.

Reglas: Monto puede ser aporte (+) o retiro (-) según tu política.

RecordatorioGasto

Recordatorios programados de gastos futuros.

Atributos: Id, UserId, Titulo, FechaProgramada, Notas?, MontoEstimado: decimal(18,2).

Relaciones: pertenece a User (cascada).

Uso: alertas para pagos recurrentes o puntuales.

Reglas transversales

Monedas con precisión decimal(18,2).

Fechas en UTC o DateTimeOffset si habrá zonas horarias.

Integridad: FK obligatorias; cascada en agregados de usuario; Restrict en Categoria.

Nomenclatura PascalCase y un solo idioma en el dom