EASYFINANCE

E F

Autor: Héctor Santana Camacho

Fecha: 21/12/2025

Versión 1.0

PLAN DE PROYECTO

EasyFinance

DAW. Desarrollo de Aplicaciones Web
PRW. Proyecto de Desarrollo de Aplicaciones Web

Nombre del fichero:	DAW_PRW_EF_UT01 Plan de proyecto.odt
Fecha de esta versión:	21/12/2025

Historial de revisiones

Fecha	Descripción	Autor
21/12/2025	Creación del documento	Héctor Santana Camacho

ÍNDICE

1 RESUMEN EJECUTIVO

EasyFinance es una aplicación web de gestión de finanzas personales que ayuda a usuarios particulares y familias a controlar de manera sencilla y efectiva sus ingresos, gastos y objetivos de ahorro. La plataforma proporciona un entorno centralizado donde el usuario puede registrar movimientos diarios, visualizar el estado de su economía mediante gráficos por categorías, definir metas de ahorro personales y monitorizar el progreso hacia cada objetivo.

El objetivo principal es ofrecer una herramienta intuitiva y segura que permita mejorar la educación financiera personal, facilitando la toma de decisiones sobre el gasto y el ahorro de forma accesible a cualquier usuario sin conocimientos técnicos profundos.

2 ALCANCE DEL PROYECTO

El proyecto EasyFinance consiste en el desarrollo de una aplicación web de gestión de finanzas personales orientada a usuarios particulares que desean controlar de forma sencilla sus ingresos, gastos y metas de ahorro. El sistema incluirá un front-end para el usuario final y un panel de administración básico, así como los servicios de back-end y la base de datos necesarios para registrar movimientos económicos, visualizar resúmenes gráficos y gestionar objetivos de ahorro personales.

2.1 Objetivo general

Diseñar y desarrollar una aplicación web que permita a una persona o familia gestionar su economía doméstica de manera centralizada, facilitando el registro de movimientos, el control del nivel de gasto y el seguimiento del ahorro hacia metas concretas.

2.2 Objetivos específicos

- Permitir el registro, modificación y eliminación de ingresos y gastos diarios mediante formularios sencillos.
- Ofrecer un dashboard con resúmenes y gráficos que muestren la distribución del gasto y el saldo mensual.
- Proporcionar herramientas para definir metas de ahorro personales y visualizar el progreso acumulado hacia cada una de ellas.
- Facilitar la consulta de movimientos mediante filtros por fecha, categoría y tipo.
- Disponer de un panel de administración para la gestión de usuarios y de las categorías financieras base del sistema.

2.3 Límites del proyecto

Quedan fuera del alcance del proyecto las integraciones en tiempo real con entidades bancarias, pasarelas de pago u otros servicios financieros externos. Tampoco se contempla el desarrollo de aplicaciones móviles nativas ni funcionalidades avanzadas de inversión, asesoramiento financiero automático o análisis fiscal. La solución se centrará exclusivamente en la versión web, en la gestión manual de movimientos por parte del usuario y en un conjunto reducido de opciones de administración interna.

3 REQUISITOS DEL PROYECTO

3.1 Requisitos funcionales

El sistema deberá permitir al usuario:

- Registrarse, iniciar sesión y cerrar sesión de forma segura.
- Registrar, modificar y eliminar ingresos.
- Registrar, modificar y eliminar gastos.

- Mostrar un dashboard con resumen mensual de ingresos, gastos y saldo, incluyendo gráficos por categoría.
- Consultar la lista completa de movimientos aplicando filtros por fecha, categoría y tipo (ingreso/gasto).
- Crear, modificar y eliminar metas de ahorro personales.
- Mostrar el progreso de cada meta de ahorro (importe ahorrado, importe objetivo y porcentaje alcanzado).
- Consultar y modificar los datos básicos de su perfil y cambiar la contraseña.

El sistema deberá ofrecer al administrador:

- La posibilidad de visualizar el listado de usuarios registrados sin acceder a sus detalles financieros.
- Crear, modificar y eliminar las categorías financieras base (ingresos y gastos).

3.2 Requisitos no funcionales

- El sistema deberá ser accesible desde un navegador web moderno sin necesidad de instalar software adicional en el cliente.
- La interfaz deberá ser clara y usable, adaptándose correctamente a diferentes resoluciones de pantalla (diseño responsive).
- Las operaciones básicas de consulta (dashboard y listado de movimientos) deberán obtener respuesta en un tiempo razonable para el usuario en condiciones normales de carga.
- Los datos de acceso y la información sensible deberán gestionarse de forma segura, utilizando prácticas estándar de autenticación y protección de contraseñas.
- El sistema deberá permitir trabajar con varios usuarios de forma concurrente sin interferencias entre sus datos.

3.3 Requisitos técnicos

- La aplicación deberá implementarse con una arquitectura web cliente-servidor.
- El front-end deberá desarrollarse utilizando HTML5, CSS y JavaScript, pudiendo apoyarse en un framework de estilos para el diseño responsive.
- El back-end deberá implementarse con una tecnología de servidor basada en Java (por ejemplo, Spring Boot) o equivalente que permita exponer una API REST.
- La información persistente (usuarios, movimientos, metas de ahorro y categorías) deberá almacenarse en una base de datos relacional (por ejemplo, MySQL).
- El proyecto deberá gestionarse mediante control de versiones y podrá desplegarse en un entorno de servidor estándar (local o en la nube) compatible con la pila tecnológica elegida.

4 ESTRUCTURA DE TRABAJO Y CRONOGRAMA

El proyecto se organizará en varias fases de trabajo consecutivas, con solapamiento parcial cuando sea necesario:

Fase 1 – Análisis y definición (Semana 1–2)
Recopilación de requisitos, definición del alcance, elaboración de casos de uso y modelo de datos inicial. Responsable: desarrollador del proyecto.

Fase 2 – Diseño funcional y técnico (Semana 3–4)
Diseño de la arquitectura de la aplicación, especificación de la API, diseño de la base de datos y maquetación básica de las pantallas principales. Responsable: desarrollador del proyecto.

Fase 3 – Implementación del back-end (Semana 5–7)
Desarrollo de los servicios de servidor, lógica de negocio para movimientos y metas de ahorro, integración con la base de datos y pruebas unitarias básicas. Responsable: desarrollador del proyecto.

Fase 4 – Implementación del front-end usuario (Semana 8–9)
Desarrollo de las vistas del usuario (login, registro, dashboard, movimientos, metas de ahorro y perfil) y consumo de la API del back-end. Responsable: desarrollador del proyecto.

Fase 5 – Implementación del front-end administrador (Semana 10)
Desarrollo del panel de administración para gestión de usuarios y categorías base. Responsable: desarrollador del proyecto.

Fase 6 – Pruebas integradas y ajustes finales (Semana 11–12)
Pruebas funcionales completas, corrección de errores, ajustes de usabilidad y preparación de la versión final para entrega.

Responsable: desarrollador del proyecto.

Para la planificación y seguimiento de tareas se utilizará un tablero tipo Kanban en una herramienta de gestión de tareas (por ejemplo, Trello o equivalente), donde se registrarán las tareas detalladas y su estado (pendiente, en curso, completada).

5 ESTRUCTURA DE TRABAJO Y CRONOGRAMA

Para la ejecución del proyecto se utilizarán los siguientes recursos.

5.1 Hardware

Equipo personal con capacidad suficiente para ejecutar entorno de desarrollo, servidor de aplicaciones y base de datos de forma local.

5.2 Software

- Sistema operativo de escritorio (Windows, Linux o equivalente).
- Entorno de desarrollo integrado (IDE) para Java y desarrollo web.
- Servidor de aplicaciones / framework para back-end (por ejemplo, Spring Boot).
- Navegador web moderno para pruebas del front-end.
- Gestor de base de datos relacional (por ejemplo, MySQL) y herramienta de administración asociada.
- Sistema de control de versiones (por ejemplo, Git) y repositorio remoto para el código fuente.
- Herramientas auxiliares para diseño y prueba (postman o similar para pruebas de API, editor de diagramas UML, herramienta de gestión de tareas).

5.3 Otros recursos

Conexión a internet para acceso a documentación técnica, repositorios y despliegue en entornos remotos si fuese necesario.

6 GESTIÓN DE RIESGOS

6.1 Posibles problemas o retrasos

- Riesgo de retrasos por una estimación insuficiente del tiempo necesario para el desarrollo y pruebas de todas las funcionalidades previstas.
- Riesgo técnico asociado al uso de tecnologías de servidor y base de datos (errores de configuración, problemas de integración entre front-end y back-end).
- Riesgo de incremento del alcance del proyecto (añadir nuevas funcionalidades no previstas inicialmente) que afecte a los plazos de entrega.
- Riesgo de pérdida de información o corrupción de datos durante el desarrollo por errores en la base de datos o en el código.

6.2 Plan de contingencia o prevención

- Definir un calendario realista, priorizar las funcionalidades esenciales y revisar periódicamente el avance para ajustar tareas si es necesario.
- Realizar pruebas incrementales desde las primeras fases, mantener entornos de desarrollo bien configurados y documentar la arquitectura técnica para facilitar la resolución de incidencias.
- Controlar el alcance del proyecto mediante una lista clara de requisitos cerrados y posponer posibles mejoras o ampliaciones a fases posteriores, una vez cumplidos los objetivos mínimos.
- Utilizar control de versiones de forma sistemática, realizar copias de seguridad periódicas de la base de datos y aplicar buenas prácticas de desarrollo para reducir el riesgo de pérdida o corrupción de datos.

7 CONCLUSIÓN

El proyecto EasyFinance se considera técnicamente viable y abordable dentro del marco de un proyecto final de ciclo, ya que se apoya en una arquitectura web conocida, en tecnologías consolidadas y en un alcance funcional bien acotado (gestión de movimientos, metas de ahorro y administración básica). Su desarrollo permite aplicar de forma integrada los conocimientos adquiridos en análisis, diseño, programación web, bases de datos y seguridad, a la vez que mantiene un nivel de complejidad razonable para ser completado en los plazos previstos.

Desde el punto de vista del valor, la aplicación aporta una utilidad directa al usuario final, facilitando el control del día a día financiero, la visualización de la información mediante resúmenes claros y la planificación del ahorro hacia objetivos personales. Además, genera un producto demostrable que puede servir como carta de presentación profesional, al mostrar experiencia real en el diseño y construcción de una solución web completa orientada a necesidades cotidianas de gestión económica.