



Trabajo Práctico 1
Sistema de Planificación Financiero

Solución informática diseñada para ayudar a individuos a gestionar sus finanzas personales de manera eficiente

Materia: Seminario de Práctica de Informática

Alumno: Francisco Vozzi

Legajo: VINF012173

DNI: 41032001

Profesores: Pablo Alejandro Virgolini, Ana Carolina Ferreyra

Tabla de contenidos

| | |
|---|----------|
| Introducción | 2 |
| Justificación | 3 |
| Definiciones del proyecto y del sistema..... | 4 |
| Definición del proyecto..... | 4 |
| Objetivos..... | 4 |
| Elicitación | 5 |
| Requerimientos Funcionales | 5 |
| Requerimientos No Funcionales | 5 |
| Restricciones | 6 |
| Conocimiento del negocio | 6 |
| Modelo de Negocio..... | 6 |
| Procesos del Negocio | 6 |
| Problemas Identificados | 7 |
| Oportunidades | 7 |
| <input type="checkbox"/> Adaptación cultural: Soporte para plazos fijos, FCIs y contexto económico local..... | 7 |
| <input type="checkbox"/> Educación: Incluir sugerencias de hábitos para mejorar la literacy financiera. | 7 |
| <input type="checkbox"/> Escalabilidad: Posibilidad de integrar datos en tiempo real en el futuro. | 7 |
| Propuesta de Solución | 7 |
| Inicio del Análisis: Casos de Uso | 8 |
| Aplicación del Proceso Unificado de Desarrollo (PUD) | 9 |
| Próximos Pasos..... | 10 |

Introducción

El Sistema de Planificación Financiero es una solución informática diseñada para ayudar a individuos a gestionar sus finanzas personales de manera eficiente, abordando una problemática común: la dificultad para planificar ingresos, gastos e inversiones en contextos económicos variables. Este sistema permitirá a los usuarios registrar ingresos (diarios, semanales o mensuales), categorizar gastos (incluyendo impuestos y obligaciones), seguir el rendimiento de inversiones (acciones, criptomonedas, fondos comunes de inversión, plazos fijos) y recibir recomendaciones personalizadas para

optimizar sus finanzas. Además, ofrecerá sugerencias de hábitos financieros para fomentar el ahorro y la planificación a largo plazo.

Este proyecto se desarrollará bajo el Proceso Unificado de Desarrollo (PUD), garantizando calidad, escalabilidad y eficiencia. Se utilizará el lenguaje Java para la lógica del sistema y MySQL para la persistencia de datos, con un prototipo inicial que implemente módulos operativos clave. Este sistema está dirigido a individuos de ingresos variables y asalariados que buscan herramientas accesibles y locales para gestionar sus finanzas sin depender de aplicaciones extranjeras o complejas.

Justificación

La gestión de finanzas personales es una problemática crítica en muchos países, especialmente en economías con inflación, como Argentina. Según estudios (BCRA, 2023), más del 60% de los hogares argentinos no cuentan con herramientas digitales para planificar sus finanzas, lo que lleva a decisiones improvisadas, deudas y falta de ahorros.

Las soluciones actuales, como aplicaciones internacionales (YNAB, Mint) o software empresarial, suelen ser costosas, **no están adaptadas a contextos locales** (impuestos, inversiones específicas como plazos fijos o FCIs) y requieren conectividad constante, lo que no siempre es viable.

Este Sistema de Planificación aborda esta problemática al ofrecer:

1. Adaptación local: Soporte para impuestos argentinos (IVA, Ganancias), inversiones locales (FCIs, plazos fijos) y monedas variables.
2. Accesibilidad: Funcionamiento offline con persistencia local en MySQL, ideal para usuarios con conectividad limitada.
3. Simplicidad: Interfaz en consola (prototipo inicial) que no requiere conocimientos técnicos avanzados.
4. Personalización: Recomendaciones basadas en los datos del usuario, promoviendo educación financiera.

El proyecto es viable porque utiliza tecnologías estándar (Java, MySQL), reduce costos al evitar dependencias externas y responde a una necesidad real de planificación financiera.

Además, su escalabilidad permite futuras integraciones con interfaces gráficas o APIs de datos financieros.

Definiciones del proyecto y del sistema

Definición del proyecto

El proyecto consiste en el diseño, desarrollo e implementación de un sistema informático que permita a los usuarios gestionar sus finanzas personales, incluyendo ingresos, gastos, inversiones y recomendaciones. El objetivo es entregar un prototipo operativo que incluya módulos clave (registro de transacciones, seguimiento de inversiones, reportes) y sienta las bases para una versión completa.

Definición del Sistema

El SIFIP es un sistema basado en Java que interactúa con una base de datos MySQL para:

- Registrar y categorizar ingresos y gastos.
- Seguimiento manual de inversiones (acciones, criptos, FCIs, plazos fijos) con cálculo de rendimientos.
- Generar reportes de balance y gastos por categoría.
- Ofrecer recomendaciones de inversión y hábitos financieros basadas en reglas predefinidas.
- Persistir datos localmente para uso offline.

El sistema será accesible inicialmente a través de una interfaz en consola, con posibilidad de escalar a una interfaz gráfica (Swing o JavaFX) en el futuro.

Objetivos

- Funcionales: Permitir el registro y consulta de transacciones financieras, seguimiento de inversiones, generación de reportes y recomendaciones.
- No funcionales: Garantizar escalabilidad (arquitectura modular), usabilidad (interfaz clara), seguridad (datos locales protegidos) y eficiencia (respuesta rápida en consultas).
- Resultados esperados: Reducir el tiempo de planificación financiera en un 50%, mejorar la toma de decisiones financieras y fomentar hábitos de ahorro en al menos el 70% de los usuarios.

Elicitación

La elicilación de requerimientos se realizó mediante:

- Entrevistas con usuarios potenciales: 5 personas (freelancers, asalariados, inversores amateur) identificaron necesidades como categorización de gastos, seguimiento de inversiones locales y alertas de gastos excesivos.
- Análisis de contexto: Estudio de la situación económica local (inflación, impuestos) y herramientas existentes, que reveló la falta de soluciones accesibles y adaptadas.
- Observación: Revisión de procesos manuales (planillas Excel, notas) usados por individuos para gestionar finanzas, destacando la necesidad de automatización.

Requerimientos Funcionales

- RF1: El sistema debe permitir registrar ingresos con monto, descripción, periodicidad (diario, semanal, mensual) y fecha.
- RF2: El sistema debe permitir registrar gastos con monto, descripción, categoría (impuestos, servicios, ocio, etc.) y fecha.
- RF3: El sistema debe permitir registrar activos financieros (acciones, criptos, FCIs, plazos fijos) con nombre, tipo, cantidad, precio inicial y fecha.
- RF4: El sistema debe calcular el balance financiero (ingresos - gastos).
- RF5: El sistema debe generar reportes de gastos por categoría.
- RF6: El sistema debe permitir actualizar precios de activos y calcular rendimientos.
- RF7: El sistema debe recomendar inversiones basadas en rendimientos históricos.
- RF8: El sistema debe sugerir hábitos financieros según el balance.

Requerimientos No Funcionales

- RNF1: El sistema debe ser escalable para soportar al menos 10,000 transacciones.
- RNF2: El sistema debe responder consultas en menos de 2 segundos.
- RNF3: Los datos deben persistir en MySQL con integridad garantizada.
- RNF4: El sistema debe ser usable por personas sin conocimientos técnicos.
- RNF5: Los datos deben estar protegidos contra accesos no autorizados (almacenamiento local).

Restricciones

- Uso exclusivo de Java estándar (JDK) y MySQL, sin frameworks externos.
- Prototipo inicial en consola, sin interfaz gráfica.
- Sin acceso a datos en tiempo real (precios de activos ingresados manualmente).

Conocimiento del negocio

Modelo de Negocio

El Sistema de Planificación Financiero se desarrolla para una organización ficticia, Finanzas Fáciles, una startup que ofrece herramientas de educación y planificación financiera para individuos en Argentina. Su modelo de negocio se basa en:

- Propuesta de valor: Herramientas accesibles y personalizadas para la gestión financiera, adaptadas al contexto local.
- Segmento de clientes: Individuos de 25-45 años, con ingresos variables o fijos, interesados en ahorrar e invertir.
- Canales: Distribución gratuita del prototipo, con planes futuros de suscripción para versiones premium (con interfaz gráfica o integraciones).
- Fuentes de ingresos: Versión gratuita inicial; en el futuro, suscripciones y servicios de consultoría financiera.
- Actividades clave: Desarrollo del sistema, educación financiera, recolección de *feedback*.
- Recursos clave: Equipo de desarrollo, base de datos MySQL, servidores locales.
- Costos: Desarrollo (salarios, infraestructura), mantenimiento de servidores.

Procesos del Negocio

- Captura de datos financieros: Los usuarios ingresan ingresos, gastos e inversiones.
- Análisis financiero: El sistema calcula balances, rendimientos y genera reportes.
- Recomendaciones: Basadas en datos ingresados, se sugieren acciones financieras y hábitos.
- Persistencia: Los datos se almacenan en MySQL para consultas futuras.

Problemas Identificados

- Falta de herramientas locales: Las soluciones existentes no consideran impuestos argentinos ni inversiones específicas.
- Complejidad: Muchas aplicaciones son difíciles de usar para no expertos.
- Conectividad: Dependencia de internet limita el acceso en zonas rurales.
- Educación financiera: Los usuarios carecen de orientación para ahorrar o invertir.

Oportunidades

- Adaptación cultural: Soporte para plazos fijos, FCIs y contexto económico local.
- Educación: Incluir sugerencias de hábitos para mejorar la literacy financiera.
- Escalabilidad: Posibilidad de integrar datos en tiempo real en el futuro.

Propuesta de Solución

El Sistema de Planificación Financiero será un sistema modular desarrollado en Java, con persistencia en MySQL, que incluye:

- Módulo de Transacciones: Registro y consulta de ingresos y gastos.
- Módulo de Inversiones: Seguimiento y actualización de activos financieros.
- Módulo de Reportes: Balance, gastos por categoría, rendimientos.
- Módulo de Recomendaciones: Sugerencias de inversiones y hábitos financieros.
- Persistencia: Base de datos MySQL con tablas para ingresos, gastos, activos y configuraciones.

El prototipo inicial implementará:

- Registro de ingresos y gastos.
- Cálculo de balance.
- Registro y actualización de activos con cálculo de rendimientos.
- Reporte básico de gastos por categoría.
- Persistencia en MySQL.

Arquitectura del Prototipo

- Capa de presentación: Interfaz en consola (Java Scanner).
- Capa de lógica: Clases Java para gestionar transacciones, inversiones y reportes.
- Capa de datos: Conexión a MySQL usando JDBC (parte del JDK).

- Base de datos: Tablas ingresos, gastos, activos con relaciones simples.

Inicio del Análisis: Casos de Uso

Diagrama de Casos de Uso

Actores:

- Usuario: Individuo que gestiona sus finanzas.
- Sistema: SIFIP, que procesa datos y genera reportes/recomendaciones.

Casos de Uso:

- **Registrar Ingreso:**
 - **Actor:** Usuario.
 - **Descripción:** El usuario ingresa un ingreso con monto, descripción, periodicidad y fecha.
 - **Precondición:** El sistema está activo.
 - **Flujo principal:**
 - El usuario selecciona "Agregar ingreso".
 - Ingresa monto, descripción, periodicidad y fecha.
 - El sistema guarda el ingreso en MySQL.
 - Confirma la operación.
 - **Postcondición:** El ingreso se registra correctamente.
- **Registrar Gasto:**
 - **Actor:** Usuario.
 - **Descripción:** El usuario ingresa un gasto con monto, descripción, categoría y fecha.
 - **Precondición:** El sistema está activo.
 - **Flujo principal:**
 - El usuario selecciona "Agregar gasto".
 - Ingresa monto, descripción, categoría y fecha.
 - El sistema guarda el gasto en MySQL.

- Confirma la operación.
- **Postcondición:** El gasto se registra correctamente.
- **Consultar Balance:**
 - **Actor:** Usuario.
 - **Descripción:** El usuario solicita el balance financiero.
 - **Precondición:** Hay ingresos o gastos registrados.
 - **Flujo principal:**
 - El usuario selecciona "Ver balance".
 - El sistema calcula ingresos - gastos desde MySQL.
 - Muestra el balance.
 - **Postcondición:** El balance se muestra correctamente.
- **Registrar Activo:**
 - **Actor:** Usuario.
 - **Descripción:** El usuario registra un activo financiero.
 - **Precondición:** El sistema está activo.
 - **Flujo principal:**
 - El usuario selecciona "Agregar activo".
 - Ingresa nombre, tipo, cantidad, precio inicial y fecha.
 - El sistema guarda el activo en MySQL.
 - Confirma la operación.
 - **Postcondición:** El activo se registra correctamente.

Aplicación del Proceso Unificado de Desarrollo (PUD)

El desarrollo seguirá las fases del PUD:

- **Incepción** (actual):
 - Definición del alcance, objetivos y requerimientos.
 - Justificación del proyecto mediante análisis del modelo de negocios.
 - Creación del prototipo inicial (código anterior).
 - Entregables: Documento de visión, casos de uso iniciales.

- **Elaboración** (próxima fase):
 - Diseño detallado de la arquitectura.
 - Modelado UML (diagramas de clases, secuencia).
 - Implementación de módulos adicionales (inversiones, reportes).
 - Pruebas unitarias en Java.

- **Construcción**
 - Desarrollo completo de módulos.
 - Integración con MySQL.
 - Pruebas de integración.
 -

- **Transición**
 - Entrega del prototipo operativo.
 - Capacitación de usuarios.
 - Recolección de feedback para iteraciones futuras.

Próximos Pasos del proyecto

Como líder de proyecto, las siguientes acciones serán clave para asegurar el progreso del proyecto:

- **Establecer un cronograma:** Planificar las siguientes fases: 2 semanas para inyección, 4 semanas para elaboración y 6 semanas para construcción.
- **Usar herramientas de colaboración:** Utilizar Git para gestionar el control de versiones.

Respecto al desarrollo del prototipo:

- **Completar el módulo de activos**, asegurándome de que incluya la actualización de precios y el cálculo de rendimientos.
- **Implementar reportes detallados de gastos**, organizados por categorías.
- **Agregar recomendaciones básicas** al sistema para mejorar la experiencia del usuario.

Finalmente, en cuanto a la participación activa:

- **Organizar reuniones semanales** para revisar el progreso y asegurar que todo esté en marcha.
- **Participar en foros** para compartir avances, resolver dudas y recibir retroalimentación.
- **Documentar todos los cambios y actualizaciones** en espacios comunes, de modo que todo quede registrado para asegurar la trazabilidad del proyecto.