

Documentación del Proyecto: Gestión Financiera en JavaFX

Indice

Documentación del Proyecto: Gestión Financiera en JavaFX.....	1
1. Contextualización.....	1
Herramientas utilizadas:.....	2
Elementos destacables:.....	2
2. Estudio de Mercado.....	2
Competencia:.....	2
Valor diferenciador:.....	2
Clientes potenciales:.....	2
3. Presupuesto.....	2
Recursos Humanos:.....	2
Software y Licencias:.....	2
Infraestructura:.....	3
Presupuesto Total Estimado:.....	3
4. Documentación Técnica.....	3
Diagrama de Clases.....	3
Diagrama de Casos de Uso.....	5
Actores:.....	5
Casos de uso principales:.....	5
5. Manual de Usuario.....	5
Requisitos del Sistema:.....	5
Pasos para usar la aplicación:.....	5
6. Estructura del Proyecto.....	6
Paquete com.carballeira.model.....	6
Clase: FinanceEntry.....	6
Paquete com.carballeira.controller.....	7
Clase: FinanceController.....	7
Clase: LoginController.....	7
Clase: AlertaController.....	7
Paquete com.carballeira.view.....	8
Clase: AlertaView.....	8
7. Flujo del Proyecto.....	8

1. Contextualización

Este proyecto consiste en una aplicación de escritorio desarrollada en JavaFX para la gestión de entradas financieras (ingresos y gastos). El objetivo principal es proporcionar una herramienta sencilla e intuitiva para registrar, validar y analizar transacciones financieras.

El sistema permite:

- Agregar y validar datos financieros.
- Gestionar tipos de transacciones (ingreso o gasto).
- Mostrar mensajes de error en caso de validaciones incorrectas.

- Un sistema básico de inicio de sesión.

Herramientas utilizadas:

- **Java** : Lenguaje de programación utilizado.
- **JavaFX**: Framework para el desarrollo de interfaces gráficas.
- **Eclipse IDE/IntelliJ IDEA**: Entornos de desarrollo integrados.

Elementos destacables:

- Implementación del patrón Modelo-Vista-Controlador (MVC) para una clara separación de responsabilidades.
 - Validaciones integradas para evitar entradas erróneas o inconsistentes.
 - Sistema básico de inicio de sesión.
 - Mensajes de alerta mediante la API de JavaFX.
-

2. Estudio de Mercado

Competencia:

- Aplicaciones como **Microsoft Excel**, **Google Sheets** y herramientas específicas de contabilidad (por ejemplo, QuickBooks, Wave).
- Estas herramientas suelen ser más complejas y con una curva de aprendizaje más alta, lo que puede desanimar a usuarios sin experiencia técnica.

Valor diferenciador:

- Interfaz simplificada para usuarios no técnicos.
- Validación inmediata de los datos introducidos.
- Integración visual con gráficos circulares para representar los datos.

Clientes potenciales:

- Estudiantes o profesionales que deseen gestionar sus finanzas personales.
 - Pequeñas empresas o autónomos que requieran un sistema básico para seguimiento de ingresos y gastos.
-

3. Presupuesto

Recursos Humanos:

- **Desarrolladores**: 2 personas (tiempo estimado: 1 semanas, 40 horas).
 - Coste: 25 €/hora → Total: 2.000 €.

Software y Licencias:

- **JavaFX**: Gratuito.
- **Scene Builder**: Gratuito.

- **IDE (Eclipse/IntelliJ IDEA):** Gratuito.
- **Sistema Operativo:** Windows/Linux/Mac (sin coste adicional).

Infraestructura:

- Uso de un ordenador estándar para desarrollo (~700 € si no se cuenta con hardware previo).

Presupuesto Total Estimado:

2.700 €.

4. Documentación Técnica

Diagrama de Clases

Incluye las clases principales de la aplicación, organizadas por paquetes:

1. `com.carballeira.model`
 - **FinanceEntry:** Modelo que representa una entrada financiera.
2. `com.carballeira.controller`
 - **FinanceController:** Controlador para manejar operaciones relacionadas con las entradas financieras.
 - **LoginController:** Controlador para gestionar el inicio de sesión.
 - **AlertaController:** Controlador para mostrar mensajes de error.
3. `com.carballeira.view`
 - **AlertaView:** Vista para generar mensajes de alerta.

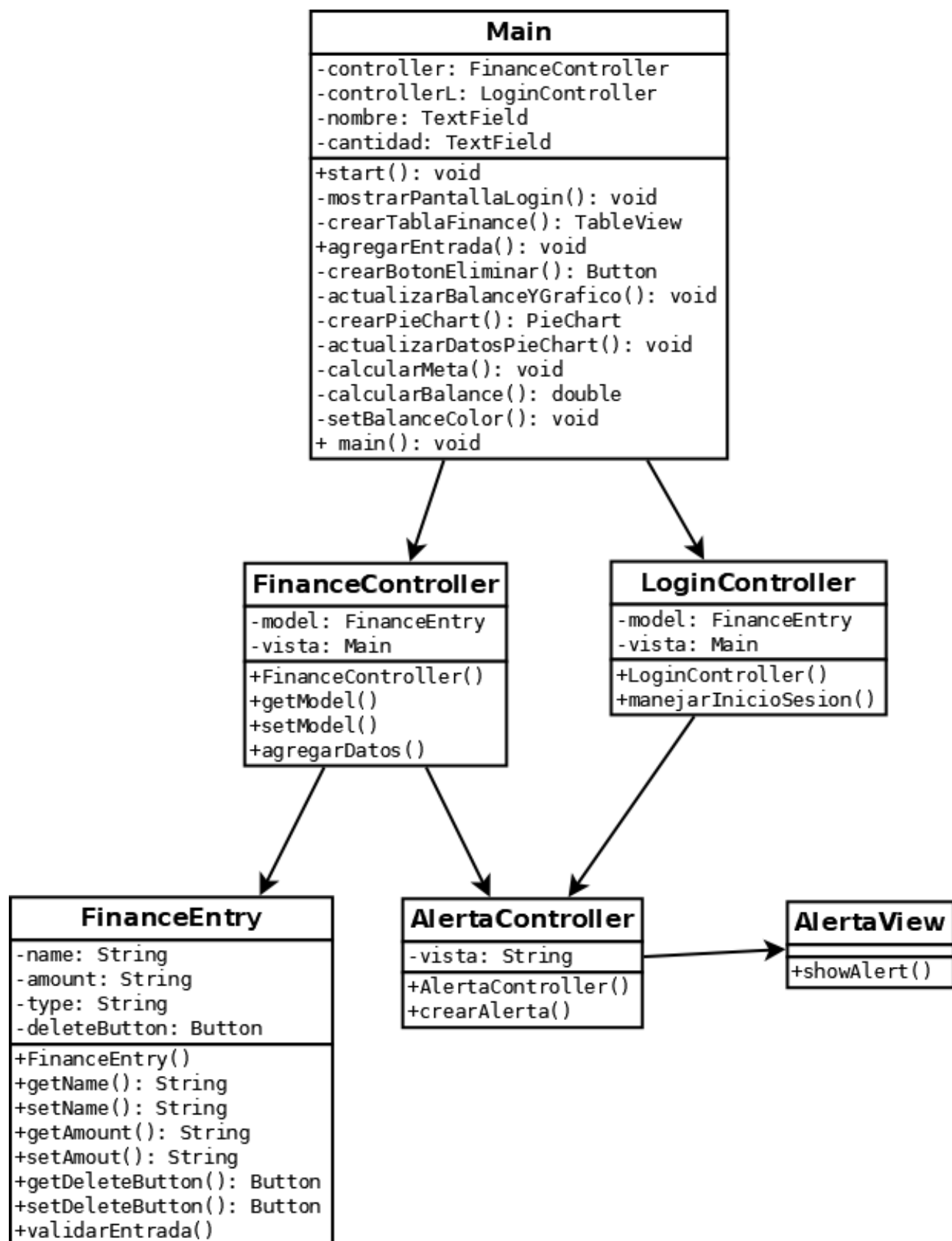


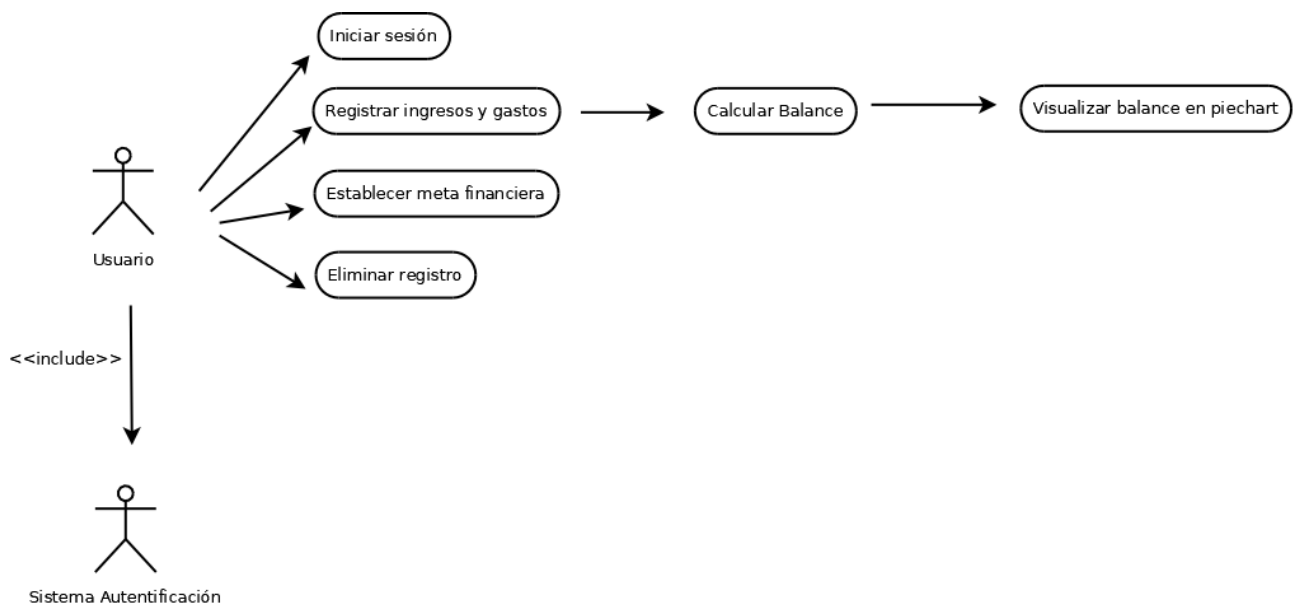
Diagrama de Casos de Uso

Actores:

- **Usuario:** Introduce datos financieros y visualiza el estado de sus finanzas.
- **Sistema:** Valida, almacena y presenta los datos.

Casos de uso principales:

1. **Iniciar sesión.**
2. **Registrar una nueva entrada financiera.**
3. **Establecer meta financiera.**
4. **Eliminar una entrada existente.**
5. **Calcular Balance.**
6. **Visualizar Balance en el gráfico.**



5. Manual de Usuario

Requisitos del Sistema:

- **Sistema Operativo:** Windows, MacOS o Linux.
- **Java Runtime Environment (JRE):** Versión 17 o superior.
- **RAM:** Mínimo 2 GB.

Pasos para usar la aplicación:

1. **Abrir la aplicación:**
 - Ejecute el archivo `.jar` proporcionado.

2. Inicio de sesión:

- Introduzca su nombre de usuario y contraseña.
- Si algún campo está vacío, aparecerá un mensaje de error.

3. Registrar una entrada financiera:

- Complete los campos de **Nombre**, **Cantidad** y seleccione el tipo (**Ingreso** o **Gasto**).
- Haga clic en el botón "Agregar".
- Si los datos son válidos, la entrada se añadirá al sistema y se actualizará el gráfico.
- En caso de error, aparecerá una alerta indicando el problema.

4. Eliminar una entrada financiera:

- Haga clic en el botón de eliminación correspondiente.

5. Cerrar la aplicación:

- Simplemente cierre la ventana principal

6. Estructura del Proyecto

Paquete `com.carballeira.model`

Clase: `FinanceEntry`

Esta clase actúa como el modelo para representar una entrada financiera en la aplicación.

• Atributos:

- `name`: Nombre de la entrada financiera.
- `amount`: Cantidad asociada a la entrada financiera (String, luego validada como número).
- `type`: Tipo de entrada, puede ser "Ingreso" o "Gasto".
- `deleteButton`: Botón que permite eliminar la entrada financiera.

• Constantes de error:

- `ERROR_NOMBRE_VACIO`: Mensaje de error para nombres vacíos.
- `ERROR_CANTIDAD_VACIA`: Mensaje de error para cantidades vacías.
- `ERROR_TIPO_NO_SELECCIONADO`: Mensaje de error si no se selecciona un tipo.
- `ERROR_CANTIDAD_INVALIDA`: Mensaje de error si la cantidad no es válida.

• Métodos:

- **Constructor**: Inicializa una entrada financiera con o sin botón de eliminación.
 - **Getters y Setters**: Acceden y modifican los atributos, especialmente `deleteButton`.
 - `validarEntrada()`
 - Verifica que los datos de la entrada sean válidos:
 - Nombre no vacío.
 - Cantidad no vacía y numéricamente válida.
 - Tipo definido.
 - Retorna un mensaje de error si falla la validación o una cadena vacía si es válida.
-

Paquete com.carballeira.controller

Clase: FinanceController

Controlador que gestiona las operaciones relacionadas con las entradas financieras y conecta el modelo con la vista.

- **Atributos:**
 - `model`: Modelo `FinanceEntry` asociado.
 - `vista`: Vista principal `Main`.
- **Constantes:**
 - `ERROR`: Título genérico para las alertas de error.
 - `GASTO` y `INGRESO`: Tipos posibles de transacciones.
- **Métodos:**
 - **Constructor:** Permite inicializar el controlador con un modelo y una vista o solo con la vista.
 - `agregarDatos()`
 - Valida la entrada utilizando `validarEntrada()` del modelo.
 - Si la entrada es válida, agrega la entrada financiera a la lista y actualiza el gráfico circular (delegado en la vista).
 - En caso de error, muestra un mensaje de alerta utilizando `AlertaController`.

Clase: LoginController

Controlador para manejar el inicio de sesión.

- **Atributos:**
 - `model`: Modelo asociado.
 - `vista`: Vista principal `Main`.
- **Constantes:**
 - `ERROR`: Título genérico para las alertas de error.
 - `ERROR_LOGIN`: Mensaje de error si faltan datos en el inicio de sesión.
- **Métodos:**
 - **Constructor:** Inicializa el controlador con el modelo, la vista o sin parámetros.
 - `manejarInicioSesion()`
 - Verifica que el usuario y la contraseña no estén vacíos.
 - Si son válidos, cierra la ventana de inicio de sesión y muestra la pantalla principal.
 - En caso de error, muestra un mensaje de alerta.

Clase: AlertaController

Controlador para la gestión de alertas en la aplicación.

- **Atributos:**

- **vista:** Vista de tipo `AlertaView` para mostrar mensajes.
 - **Métodos:**
 - **Constructor:** Inicializa el controlador con una instancia de `AlertaView`.
 - `crearAlerta()`
 - Llama al método de la vista para mostrar una alerta con un título y un contenido específicos.
-

Paquete `com.carballeira.view`

Clase: `AlertaView`

Vista responsable de mostrar mensajes de alerta en pantalla.

- **Métodos:**
 - `showAlert()`
 - Crea una alerta de tipo `ERROR` con un título y un contenido proporcionados.
 - Utiliza la API de JavaFX para mostrar la alerta.
-

7. Flujo del Proyecto

1. Inicio de sesión:

- El usuario introduce sus credenciales.
- Si los campos están vacíos, se muestra un mensaje de error.
- Si las credenciales son válidas, se abre la pantalla principal.

2. Gestión de datos financieros:

- El usuario introduce una entrada con nombre, cantidad y tipo.
- El sistema valida los datos:
 - Si son válidos, la entrada se agrega a la lista y al gráfico circular.
 - Si no son válidos, se muestra un mensaje de error.

3. Alertas:

- Los errores son manejados por `AlertaController` y mostrados mediante `AlertaView`.
-