

1. Roles y Responsabilidades

Primera Iteración:

- **Darío Palma:**
 - Diseño e implementación de artefactos relacionados al diseño.
 - Garantizar que la interfaz gráfica sea funcional y fácil de usar.
- **Atik Tuquerrez:**
 - Desarrollo de la lógica del controlador para gestionar las interacciones entre la vista y el modelo.
 - Asegurar que las solicitudes del usuario sean eficientes.
- **Angel Cabezas:**
 - Implementación del modelo para gestionar la lógica de datos.
 - Responsable de la conexión y gestión de la base de datos MySQL.

Segunda Iteración (Mejoras Implementadas):

- **Darío Palma:**
 - Mejora de la disposición de los elementos para una experiencia de usuario más fluida.
 - Ordenamiento de acciones compradas.
- **Atik Tuquerrez:**
 - Resumen de las acciones compradas.
 - Guardar en un archivo Excel los valores comprados en la bolsa de valores.
- **Angel Cabezas:**
 - Implementación de la pantalla login y registro de usuario.
 - Optimización de consultas SQL y reducción de código redundante.

2. Herramientas y Tecnologías

Entorno de Desarrollo:

- **NetBeans:** IDE principal utilizado para el desarrollo del proyecto.

Modelo Vista Controlador (MVC):

- **Base de Datos:**
 - Sentencias SQL.
 - Conexiones a la base de datos mejoradas con try-with-resources.

- Script para la creación de la base de datos.

- **Modelo:**

- Gestión de eventos provenientes de la vista.
- Refactorización para reducir redundancias.

- **Vista:**

- Interfaces gráficas creada con Java
- Personalización de JTable para mejor visualización.

- **Controlador:**

- Manejo de datos del usuario con acceso a la base de datos.
- Realización de validaciones y cálculos como ganancias o pérdidas de acciones.

Base de Datos:

- **MySQL:**

- Base de datos relacionales para almacenar datos financieros y de usuarios.

Integración con APIs Externas:

- **API "Alpha Vantage":**

- API para obtener precios de acciones en tiempo real, se uso "Alpha Vantage".

3. Estructura del Repositorio

Directorio del Proyecto:

- **lib:**

- Almacena las librerías necesarias para el proyecto.
- Se incluyen "json-20240303.jar" , "mysql-connector-j-9.1.0", "poi-5.3.2.jar".

- **nbproject:**

- Archivos de configuración y metadatos de NetBeans.

- **src:**

Carpeta principal que alberga el código fuente organizado en subdirectores.

- **BD:** Conexión y operaciones con la base de datos.
- **Controlador:** Lógica de negocio y validaciones.
- **Modelo:** Define la estructura de datos y las reglas del sistema

- **Vista:** Interfaces graficas desarrolladas.

4. Resumen del Flujo de Trabajo

Primera Iteración:

a) Planificación:

- Definición de Historias de Usuario y Escenarios:
 - Basado en las historias de usuario que identificaste:
 - Inicio de sesión (E01).
 - Registro de compras de acciones (E02).
 - Consulta de precios en tiempo real (E03).
 - Consulta de estado de acciones (E04).

b) Diseño:

- Prototipo de Interfaz Gráfica:
 - Diseñar pantallas específicas para cada historia de usuario:
 - Pantalla de inicio de sesión (E01).
 - Formulario de registro de compras (E02).
 - Panel de consulta de precios en tiempo real (E03).
- Definición de la Estructura MVC en NetBeans:
 - Crear las carpetas y clases base para cada módulo relacionado con las historias de usuario:

c) Desarrollo:

- Modelo:
 - Crear clases para representar las entidades identificadas:
 - E01: Clase Usuario para manejar las credenciales de acceso.
 - E02: Clase Compra con métodos para registrar la información (fecha, cantidad, precio).
 - E03: Métodos en el modelo para conectar con la API y obtener precios en tiempo real.
 - E04: Métodos para calcular ganancias/pérdidas basadas en los precios actuales y datos históricos.
- Vista:
 - Diseñar interfaces gráficas alineadas con las historias:
 - E01: Formulario de login con validaciones de entrada.
 - E02: Formulario de registro con campos para fecha, cantidad y precio.
 - E03: Pantalla para mostrar precios actuales.
 - E04: Panel con una tabla que incluya información detallada sobre ganancias/pérdidas.
- Controlador:
 - Manejar la interacción entre la vista y el modelo para cada historia:
 - E01: Validar credenciales y mostrar mensajes en caso de error.
 - E02: Verificar los datos antes de guardarlos en la base de datos.

- E03: Consultar la API y actualizar la vista con los datos recibidos.
- E04: Consultar el modelo y mostrar resultados en la vista.

d) Pruebas:

Pruebas Funcionales:

- Simular escenarios específicos para cada historia:
 - E01: Inicio de sesión exitoso/fallido.
 - E02: Registro de datos válidos/inválidos.
 - E03: Consulta exitosa/error al obtener precios.
 - E04: Consulta de acciones registradas/sin acciones disponibles.

Segunda Iteración:

a) Planificación:

- Nuevas funcionalidades agregadas:
 - Ordenamiento de acciones compradas (E05).
 - Resumen de las acciones compradas (E06).
 - Exportación de datos a un archivo Excel (E07).
 - Implementación de mejoras en las pantallas de login y registro de usuario (E08).

b) Diseño:

- Actualización de la Interfaz Gráfica:
 - Inclusión de botones para ordenar acciones compradas.
 - Nueva pantalla de resumen de acciones (JFResumen).
 - Funcionalidad para exportar datos a Excel desde JFAcciones.
 - Optimización de la UI en login y registro de usuario.

c) Desarrollo:

- Modelo:
 - Métodos para ordenar acciones según diferentes criterios.
 - Cálculo y almacenamiento de resumen de compras.
 - Generación de archivos Excel con datos de compras registradas.
- Vista:
 - Adición de botón de ordenamiento en JFAcciones.
 - Nueva pantalla de resumen de compras (JFResumen).
 - Botón y funcionalidad para exportar a Excel.
 - Mejora en la interfaz de login y registro de usuario.
- Controlador:
 - Implementación de ordenamiento en ActionController.
 - Generación de resumen en ActionController y actualización de JFResumen.
 - Manejo de exportación en ExportarExcel.

d) Pruebas Funcionales:

- E05: Verificación del ordenamiento de acciones.
- E06: Validación de cálculo y visualización del resumen de compras.

- E07: Exportación correcta de datos a Excel.
- E08: Comprobación de mejoras en la usabilidad del login y registro.