

Flujo de Trabajo para el Proyecto de Gestión de Acciones

1. Roles y Responsabilidades

- **Darío Palma**
 - Responsable del diseño e implementación de artefactos correspondientes al diseño.
 - Asegurará que la interfaz gráfica sea funcional y fácil de usar.
- **Atik Tuquerrez**
 - Desarrollará la lógica del controlador para manejar las interacciones entre la vista y el modelo.
 - Garantizará que las solicitudes del usuario se gestionen de manera eficiente.
- **Angel Cabezas**
 - Implementará el modelo para gestionar la lógica de datos.
 - Será responsable de la conexión y gestión de la base de datos MySQL.

2. Herramientas y Tecnologías

- **Entorno de Desarrollo:**
 - **NetBeans:** IDE utilizado para el desarrollo del proyecto.
- **Modelo Vista Controlador (MVC):**
 - **BD:**
 - Sentencias SQL.
 - Conexiones a la base de datos
 - Script para la creación de la base de datos.
 - **Modelo:**
 - Gestión de eventos provenientes de la vista.
 - Comunicación entre la vista y el modelo para cumplir con las solicitudes del usuario.
 - **Vista:**
 - Interfaz gráfica creada con Java.
 - **Controlador:**
 - Manejo de datos del usuario con acceso a la base de datos MySQL.
 - Realización de validaciones y cálculos, como ganancias/pérdidas de acciones.

- **Base de Datos:**
 - **MySQL:** Base de datos relacional para almacenar datos financieros y de usuarios.
- **Integración con APIs Externas:**
 - API para obtener precios de acciones en tiempo real, se usó “Alpha Vantage”.

3. Estructura del Repositorio

- **lib:**
 - Almacena las librerías necesarias para el proyecto.
 - Estas incluyen “[mysql-connector-j-9.1.0](#)” y “[json-20240303.jar](#)”.
- **nbproject:**
 - Contiene archivos de configuración y meta data de NetBeans.
 - Específico para facilitar la carga del proyecto en el IDE.
- **src:**
 - Carpeta principal que alberga el código fuente organizado en subdirectorios:
 - BD: Maneja la conexión y operaciones con la base de datos.
 - Controlador: Implementa la lógica del negocio y coordina las interacciones entre la vista y el modelo.
 - Modelo: Define las estructuras de datos y las reglas del sistema.
 - Vista: Contiene las interfaces gráficas desarrolladas con Swing.
 - Imágenes: Incluye recursos gráficos utilizados en la interfaz.

4. Resumen del Flujo de Trabajo

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:

3. 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.

4. 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.