Plataforma de Gestión de Usuarios y Finanzas Personales

El diagrama presenta el diseño de una plataforma de broker de bolsa en el que el usuario podrá contar con herramientas para administrar sus finanzas de manera efectiva, ofreciendo funcionalidad tales como el registro de usuario,la gestión de cuentas y la supervisión de sus saldos financieros.

Objetivo: El objetivo de ARGBroker Demo es proporcionar una plataforma educativa que simule la experiencia de un broker financiero en la compra y venta de acciones en el mercado argentino (MERVAL). La plataforma permite a los usuarios gestionar su portafolio de inversiones de manera intuitiva, proporcionando herramientas para tomar decisiones informadas mediante la consulta de cotizaciones y el seguimiento de su capital virtual. La aplicación busca fomentar la comprensión de conceptos financieros y estrategias de inversión sin riesgo real.

Alcance: La plataforma está diseñada para cumplir con las siguientes funcionalidades:

- Registro y autenticación de usuarios, cada uno con una cuenta demo inicial.
- Gestión de cuentas para el seguimiento de saldo, ganancias y pérdidas.
- Simulación de operaciones de compra y venta de acciones del mercado MERVAL, manteniendo un registro detallado de cada operación.
- Visualización de cotizaciones en tiempo real y detalles relevantes de las acciones.
- Generación de reportes básicos de ganancia/pérdida y análisis de portafolio.

Es importante destacar que esta plataforma es solo una simulación educativa y no realiza operaciones reales en el mercado. Su uso está limitado a fines académicos y demostrativos.

Contexto: El proyecto se desarrolla en el contexto de una plataforma de aprendizaje financiero destinada a usuarios que deseen adquirir conocimientos prácticos sobre la gestión de un portafolio de inversiones en el mercado bursátil. ARGBroker Demo se enmarca como un proyecto educativo, pensado para ser usado por estudiantes, personas interesadas en el mundo financiero y programadores en proceso de aprendizaje que buscan explorar conceptos básicos de finanzas y programación aplicada en un entorno seguro y controlado.

1. Clase Usuario

- **Atributos**:
    - id: Identificador único del usuario.
    - email: Correo electrónico del usuario.
    - nombre: Nombre del usuario.
    - saldo: Saldo actual de la cuenta demo del usuario.
- **Métodos**:
    - agregar_accion(accion, cantidad): Agrega una acción al portafolio del usuario.
    - remover_accion(accion, cantidad): Remueve una cantidad específica de acciones del portafolio del usuario.

2. Clase Accion

- **Atributos**:
    - id: Identificador único de la acción.
    - simbolo: Símbolo de la acción (ej. ALUA).
    - nombre: Nombre de la empresa asociada a la acción (ej. Aluminio Argentino S.A.).
    - precio_actual: Último precio de la acción.
- **Métodos**:
    - consultar_cotizacion(): Retorna la cotización actual de la acción.

3. Clase Operacion

- **Atributos**:
    - usuario_id: Identificador del usuario que realiza la operación.
    - accion_id: Identificador de la acción involucrada en la operación.
    - tipo: Tipo de operación (compra o venta).
    - cantidad: Cantidad de acciones compradas o vendidas.
    - precio_unitario: Precio al cual se realiza la operación.
    - fecha: Fecha de la operación.
- **Métodos**:
    - calcular_total(): Calcula el monto total de la operación (precio_unitario *

cantidad).

- validar_operacion(): Verifica si la operación es válida (por ejemplo, si el usuario tiene saldo suficiente para comprar o cantidad suficiente para vender).

4. Clase Portafolio

- **Atributos**:
  - usuario: Usuario propietario del portafolio.
  - acciones: Lista de acciones en el portafolio.
- **Métodos**:
  - valor_total(): Calcula el valor total de todas las acciones en el portafolio.
  - mostrar_acciones(): Muestra las acciones que posee el usuario en su portafolio.
  - consultar_portafolio(): Devuelve una vista detallada del portafolio del usuario.

5. Clase ManejadorDB

- **Atributos**:
  - conexion: Conexión a la base de datos.
  - cursor: Cursor para ejecutar consultas SQL.
- **Métodos**:
  - conectar_a_base_datos(): Establece la conexión con la base de datos.
  - verificar_usuario_existente(id): Comprueba si un usuario ya está registrado en la base de datos.
  - verificar_accion_existente(simbolo): Verifica si una acción específica existe en la base de datos.
  - guardar_usuario(usuario): Guarda un nuevo usuario en la base de datos.
  - guardar_accion(accion): Almacena una nueva acción en la base de datos.
  - guardar_operacion(operacion): Guarda una operación (compra o venta) en la base de datos.
  - obtener_precio_accion(simbolo): Recupera el precio actual de una acción específica.
  - obtener_usuario(id): Obtiene la información de un usuario según su ID.
  - verificar_empresa_existente(nombre): Comprueba si una empresa ya está registrada en la base de datos.

- guardar_empresa(nombre, simbolo): Almacena los datos de una nueva empresa.
- consultar_portafolio(usuario_id): Consulta y devuelve el portafolio de un usuario específico.
- obtener_portafolio(usuario_id): Devuelve las acciones y cantidades en el portafolio de un usuario.
- generar_reporte_financiero(usuario_id): Genera un reporte financiero sobre el usuario.
- cerrar_conexion(): Cierra la conexión a la base de datos.

Relaciones entre Clases

1. **Usuario y Portafolio**: Relación de composición con cardinalidad 1 a 1.
2. **Portafolio y Accion**: Relación de agregación con cardinalidad 0..* (muchas).
3. **Operacion**: Asociación con Usuario y Accion con cardinalidad 1 a 0..* (muchas).
4. **ManejadorDB**: Interactúa con todas las clases mediante métodos, pero no tiene una relación directa o atributos de tipo de estas clases en el diagrama.

Nomenclatura: La nomenclatura utilizada para el diagrama es PascalCase.