



Documentación del diseño del diagrama de clases

Título: Sistema de gestión de inversiones.

Descripción: El Sistema de Gestión de Inversiones es una aplicación diseñada para facilitar y administrar las transacciones de compra y venta de acciones entre inversores y empresas. El sistema permite a los inversores gestionar sus portafolios, realizar transacciones y monitorear sus inversiones. Estas acciones son emitidas por empresas, las cuales los inversores pueden adquirir, vender y mantener en sus propios portafolios. El sistema proporciona una plataforma integrada para la gestión eficiente de las actividades financieras relacionadas con el mercado de valores.

Nomenclatura:

Nombres de clases:

- **Empresa:** Representa una empresa que emite acciones.
- **Inversor:** Representa un usuario que invierte en acciones.
- **Acción:** Representa una acción emitida por una empresa.
- **Portafolio:** Representa el conjunto de inversiones de un inversor.
- **Transaccion:** Representa una operación de compra o venta de acciones.
- **TipoTransaccion:** Enumeración que define los tipos de transacciones (COMPRA, VENTA).
- **TipoInversor:** Enumeración que define los tipos de inversores según su perfil de riesgo (BAJO_RIESGO, MEDIO_RIESGO, ALTO_RIESGO)

Atributos:

- Para la clase **Empresa**:
 - "empresa_id: int"
 - "nombre: string"
 - "simbolo: string"
- Para la clase **Inversor**:
 - "inversor_id: int"
 - "nombre: string"
 - "correo_electronico: string"
 - "contraseña: string"
 - "tipo_inversor: TipoInversor"
 - "fecha_alta: date"
- Para la clase **Accion**:
 - "accion_id: int"

- “empresa: Empresa”
- “ultimo_operado: float”
- “cant_compra_diaria: int”
- “cant_venta_diaria: int”
- “precio_venta_actual: float”
- “precio_compra_actual: float”
- “apertura: float”
- “min_diario: float”
- “max_diario: float”
- “ultimo_cierre: float”
- Para la clase **Portafolio**:
 - “portafolio_id: int”
 - “inversor: Inversor”
 - “saldo: float”
 - “total_invertido: float”
 - “ganancia_perdida: float”
 - “acciones: List[accion]”
- Para la clase **Transaccion**:
 - “transacción_id: int”
 - “inversor: Inversor”
 - “tipo_transaccion: TipoTransaccion”
 - “accion: List[Accion]”
 - “fecha: date”
 - “cantidad: int”
 - “monto: float”

Métodos:

- **Empresa:**
 - “crear_empresa()”
 - “modificar_empresa()”
 - “borrar_empresa()”
 - “obtener_empresa()”
- **Inversor:**
 - “crear_inversor()”
 - “modificar_inversor()”
 - “borrar_inversor()”
 - “obtener_inversor()”
- **Accion:**
 - “crear_accion()”
 - “modificar_accion()”
 - “borrar_accion()”
 - “obtener_accion()”
- **Portafolio:**
 - “calcular_ganancia_perdida()”
 - “actualizar_saldo()”
 - “crear_portafolio()”
 - “modificar_portafolio()”

- “obtener_portafolio()”
 - “borrar_portafolio()”
- **Transaccion:**
 - “comprar()”
 - “vender()”
 - “obtener_transaccion()”

Comentarios y notas:

1. **Decisión de diseño:** Se decidió utilizar enumeraciones (“enum”) para los tipos de transacción y tipos de inversor para garantizar la consistencia y reducir errores en la gestión de estos atributos.
2. **Gestión de relaciones:** Las relaciones entre las clases están claramente definidas en el diagrama, por ejemplo, un “Portafolio” está asociado a un único “Inversor” pero también puede contener múltiples “Acciones”.
3. **Extensibilidad:** Este diseño puede permitir futuras extensiones, como la inclusión de nuevas funcionalidades en las clases o la adición de nuevos tipos de transacciones sin afectar la estructura básica del sistema.
4. **Optimización:** Los métodos de las clases están diseñados para manejar las operaciones básicas de creación, modificación, borrado y obtención de datos, lo que facilita la implementación de servicios CRUD (Crear, Leer, Actualizar, Eliminar) en la aplicación.