

1- Identificar las clases del sistema, atributos y métodos.

Clase: Inversor

Atributos:

- id de tipo de documento
- cuit
- número de documento
- nombre
- apellido
- Tipo de documento, id de tipo de documento
- Portfolio, id portafolio
- User, user
- Tipo de inversor, id de tipo de inversor

Métodos:

- crear inversor
- modificar inversor

Clase: Tipo de Inversor

Atributos:

- id de tipo de inversor
- nombre

Métodos:

- editar nombre

Clase: Tipo de Documento

Atributos:

- id de tipo documento
- nombre

Métodos:

- editar nombre

Clase: Transaccion

Atributos:

- id de transaccion
- fecha y hora
- cantidad acciones
- precio
- comision de broker
- transaccion
- Accion, id de accion
- Inversor cuit
- Tipo de transaccion, id de Tipo de transaccion

Métodos:

- crear transaccion

Clase: Tipo de Transaccion

Atributos:

- id de tipo de transaccion
- nombre

Métodos:

- editar nombre

Clase: User

Atributos:

- user
- contraseña
- perfil, id de perfil

Métodos:

- crear usuario
- login
- logout
- recuperar contraseña

Clase: Perfil

Atributos:

- id de perfil
- nombre
- permiso, id de permiso

Métodos:

- editar nombre

Clase: Permiso

Atributos:

- id de permiso
- nombre

Métodos:

- editar nombre

Clase: Portafolio

Atributos:

- id de portafolio
- ganancia total
- saldo de cuenta
- perdida total

Métodos:

- Calcular total invertido

- Listar cantidad de acciones compradas
- Calcular ganancia
- Calcular perdida

Clase: Detalle de Portafolio

Atributos:

- id de detalle portafolio
- cantidad de acciones compradas
- precio por accion
- monto invertido
- perdida
- ganancia
- portafolio, id de portafolio
- accion, id de accion

Métodos:

- calcular ganancia
- calcular perdida
- cargar datos de accion

Clase: Accion

Atributos:

- id de accion
- simbolo
- nombre
- cantidad de compra diaria
- precio de venta actual
- cantidad de venta diaria
- apertura
- minimo diario
- maximo diario
- ultimo cierre
- cuit de la empresa

Métodos:

- Listar acciones

Clase: Empresa

Atributos:

- cuit
- nombre
- razon social

2- Nombrar las clases, atributos y métodos en base a la nomenclatura acordada con el equipo.

Nuestro equipo definió utilizar las siguientes nomenclaturas:

- Clases y entidades -> Upper Camel Case.
- Metodos -> Lower Camel Case.
- Atributos -> Lower Camel Case.

Clase: Inversor

Atributos:

- cuit : int
- numeroDocumento : int
- nombre :String
- apellido :String
- tipoDocumento:TipoDocumento
- portafolio :PortaFolio
- user : User
- tipoInversor : TipoInversor

Métodos:

- conoceTipoDocumento()
- conocePortafolio()
- conoceUser()

- conoceTipoInversor()
- obtenerCuit()
- modificarCuit()
- obtenerNumeroDocumento()
- modificarNumeroDocumento()
- obtenerNombre()
- modificarNombre()
- modificarApellido()
- crearInversor()
- modificarInversor()
- mostrarInversor()

Clase: TipoInversor

Atributos:

- idTipoInversor: int
- nombre :String

Métodos:

- obtenerTipoInversor()
- modificarTipoInversor()
- obtenerNombre()
- modificarNombre()
- mostarTipoInversor()
- crearTipoInversor()

Clase: TipoDocumento

Atributos:

- idTipoDocumento : int
- nombre :String

Métodos:

- obtenerTipoDocumento()
- modificarTipoDocumento()
- obtenerNombre()
- modificarNombre()

- crearTipoDocumento()
- mostrarTipoDocumento()

Clase: Transaccion

Atributos:

- idTransaccion: int
- fechaHora : date
- cantidadAcciones: int
- precio : float
- comisionBroker: float
- transaccion : Transaccion
- accion: Accion
- inversor: Inversor
- tipoTransaccion: TipoTransaccion

Métodos:

- conoceAccion()
- conoceInversor()
- conoceTipoTransaccion()
- obtenerIdTransaccion()
- obtenerFechaHora()
- obtenerCantidadAcciones()
- obtenerPrecio()
- obtenerComisionBroker()
- conoceTransaccion()
- conoceAccion()
- conoceInversor()
- conoceTipoTransaccion()
- modificarIdTransaccion()
- modificarFechaHora()
- modificarCantidadAcciones()
- modificarPrecio()
- modificarComisionBroker()
- crearTransaccion()
- modificarTransaccion()
- mostrarTransaccion()

Clase: TipoTransaccion

Atributos:

- idTipoTransaccion: int
- nombre :String

Métodos:

- obtenerTipoTransaccion()
- modificarTipoTransaccion()
- obtenerNombre()
- modificarNombre()
- mostarTipoTransaccion()
- crearTipoTransaccion()

Clase: User

Atributos:

- usuario:String
- contraseña:String
- perfil : Perfil

Métodos:

- conocePerfil()
- obtenerUsuario()
- modificarUsuario()
- modificarContraseña()
- modificarUser()
- crearUser()
- mostrarContraseña()
- login()
- logout()
- recuperarContraseña()

Clase: Perfil

Atributos:

- idPerfil: int

- nombre :String
- permiso:Permiso

Métodos:

- obtenerIdPerfil()
- modificarIdPerfil()
- obtenerNombre()
- modificarNombre()
- mostrarPerfil()
- conocePermiso()
- crearPerfil()

Clase: Permiso

Atributos:

- idPermiso: int
- nombre :String

Métodos:

- mostarIdPermiso()
- mostarNombre()
- modificarIdPermiso()
- modificarNombre()
- modificarPermiso()
- mostarPermiso()
- crearPermiso()

Clase: Portafolio

Atributos:

- idPortafolio : int
- gananciaTotal : float
- saldoCuenta : float
- perdidaTotal : float

Métodos:

- modificarIdPortafolio()
- modificarGananciaTotal()
- modificarSaldoCuenta()
- modificarPerdidaTotal()
- obtenerIdPortafolio()
- obtenerGananciaTotal()
- obtenerSaldoCuenta()
- obtenerPerdidaTotal()
- modificarPortafolio()
- crearPortafolio()
- mostrarPortafolio()
- CalcularTotalInvertido()
- ListarCantidadAccionesCompradas()
- CalcularGanancia()
- Calcular_perdida()

Clase: DetallePortafolio

Atributos:

- idDetallePortafolio : int
- cantidadAccionesCompradas : int
- precioPorAccion : float
- montoInvertido: float
- perdida : float
- ganancia : float
- portafolio : PortaFolio
- accion : Accion

Métodos:

- conoceAccion()
- conocePortafolio()
- tomarIdDetallaPortafolio()
- modificarIdDetallePortafolio()
- tomarCantidadAccionesCompradas()
- modificarCantidadAccionesCompradas()
- tomarPrecioPorAccion()
- modificarPrecioPorAccion()

- tomarMontoInvertido()
- modificarMontoInvertido()
- tomarPerdida()
- modificarPerdida()
- tomarGanancia()
- modificarGanancia()
- crarDetallePortafolio()
- modificarDetallePortafolio()
- mostrarDetallePortafolio()
- calcularGanancia()
- calcularPerdida()
- cargarDatosAccion()

Clase: Accion

Atributos:

- idAccion : int
- simbolo : String
- nombre : String
- cantidadCompraDiaria : int
- precioVentaActual : float
- cantidadVentaDiaria : float
- apertura : date
- minimoDiario: int
- maximoDiario : int
- ultimoCierre : float
- empresa: Empresa

Métodos:

- modificarIdAccion()
- modificarSimbolo()
- modificarNombre()
- modificarCantidadCompraDiaria()
- modificarPrecioVentaActual()
- modificarCantidadVentaDiaria()
- modificarApertura()
- modificarMinimoDiario()
- modificarMaximoDiario()
- modificarUltimoCierre()

- obtenerIdAccion()
- obtenerSimbolo()
- obtenerNombre()
- obtenerCantidadCompraDiaria()
- obtenerPrecioVentaActual()
- obtenerCantidadVentaDiaria()
- obtenerApertura()
- obtenerMinimoDiario()
- obtenerMaximoDiario()
- obtenerUltimoCierre()
- conoceEmpresa()
- mostrarEmpresa()
- modificarEmpresa()
- crearEmpresa()

Clase: Empresa

- obtenerCuit()
- modificarCuit()
- obtenerRazonSocial()
- modificarRazonSocial()
- obtenerNombre()
- modificarNombre()
- modificarEmpresa()
- crearEmpresa()
- MostrarEmpresa()

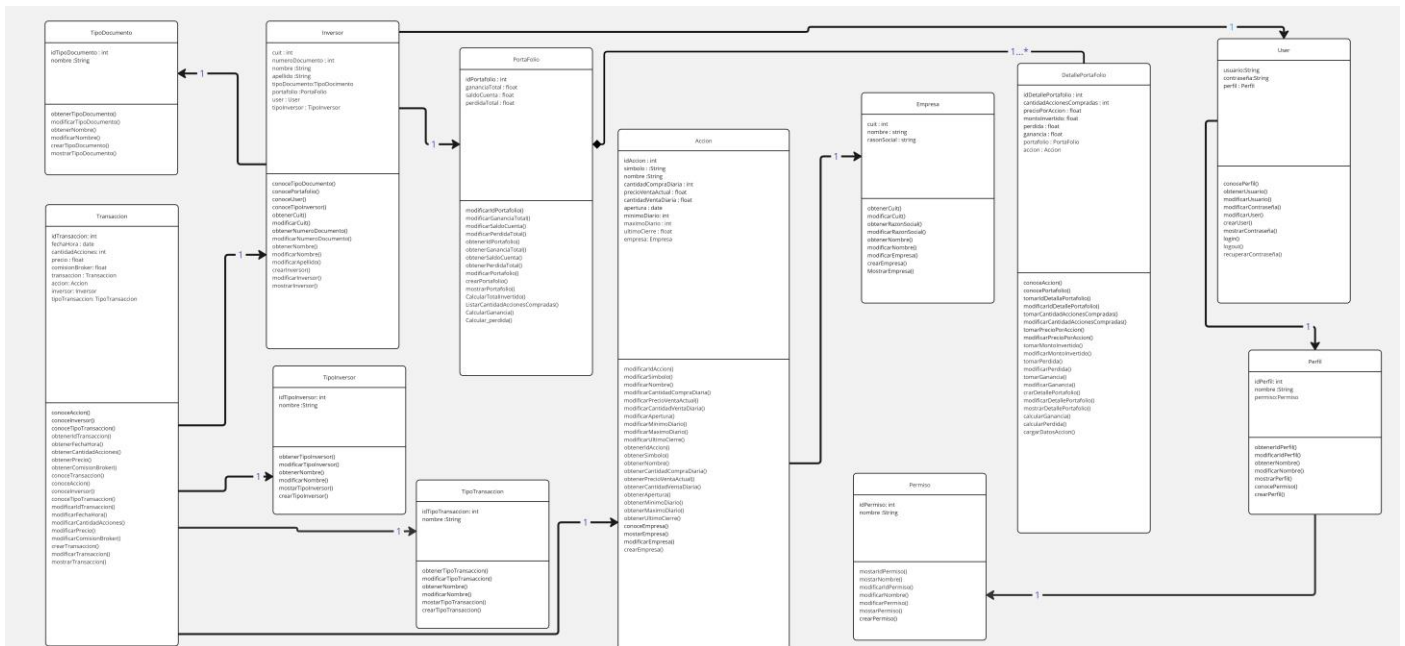
3- Identificar las relaciones entre clases.

Basándose en los requerimientos funcionales del proyecto ARGBroker Demo, se solicita: Identificar las relaciones entre clases.

- La clase **TipoDocumento** tiene una relación de asociación 1 a 1 con la clase **Inversor**
- La clase **Inversor** tiene una relación de asociación 1 a 1 con la clase **User**
- la clase **User** tiene una relación de asociación 1 a 1 con la clase **Perfil**
- La clase **Perfil** tiene una relación de asociación 1 a 1 con la clase **Permiso**
- La clase **Inversor** tiene una relación de asociación 1 a 1 con la clase **Portafolio**
- La clase **Portafolio** tiene una relación de agregación 0 a M con la clase **DetallePortafolio**

- la clase **Inversor** tiene una relación de asociación 1 a 1 con la clase **Transaccion**
- la clase **Transaccion** tiene una relación de asociación 1 a 1 con la clase **Accion**
- La clase **TipoTransaccion** tiene una relación de asociación 1 a 1 con la clase **Transacción**
- La clase **Empresa** tiene una relación de asociación 1 a 1 con la clase **Accion**
- La clase **TipoInversor** tiene una relación de asociación 1 a 1 con la clase **Inversor**

4- Realizar el diagrama de Clases:



5- Documentar Título, Descripción, nomenclatura y comentarios

Basándose en los requerimientos funcionales del proyecto ARGBroker Demo, se solicita:

Documentar el diseño especificando:

- **Un título significativo, debe reflejar el propósito del sistema.**

ARGBroker que representa un sistema de inversión de nacionalidad argentina.

- **Una breve descripción que explique el contexto, objetivo y alcance del diagrama.**

Contexto En el ámbito de las inversiones y el comercio de acciones, **ARGBroker** es un intermediario financiero que facilita la compra y venta de acciones, entre inversores y el mercado. **ARGBroker** puede ofrecer un servicio de ejecución de órdenes de compra y venta y la gestión del portafolio del inversor.

Objetivo El objetivo principal de **ARGBroker** es facilitar la compra y venta de acciones en el mercado y gestionar los portafolios de los inversores.

Alcance El diagrama abarcará los siguientes elementos y flujos: Inversores: Personas o entidades que desean comprar o vender acciones. Transacciones: Las solicitudes que los inversores envían para realizar ordenes de ventas de acciones. Empresas: Las cuales ponen a disposición sus acciones para las transacciones pertinentes de las mismas con información detallada de los datos de las acciones (precio, cantidades, límite de compra ,etc).

El diagrama detallará el flujo de información desde que un inversor coloca una orden, cómo el broker procesa esta orden, la interacción con el mercado de valores, y cómo se confirma la transacción al inversor. Además, puede incluir aspectos adicionales como las comisiones y tarifas que cobra el broker, y la gestión de riesgos y cumplimiento normativo.

Esta representación será útil tanto para nuevos inversores que buscan entender cómo funcionan sus transacciones, como para profesionales del sector que deseen optimizar y evaluar sus procesos operativos.

- **Nomenclatura elegida para nombres de clases, atributos y métodos.** La nomenclatura utilizada: Para el desarrollo del diagrama de clases utilizaremos la nomenclatura camelCase ya sea para los atributos, métodos y nombre de las clases. Para el desarrollo del diagrama de la base de dato utilizaremos la nomenclatura snakeCase ya sea para el nombre de las tablas como para el nombre de los atributos de la tabla.
- **Comentarios y/o notas que se considere necesario para proporcionar información adicional o explicar decisiones de diseño.**

El diseño planteado para la base de dato fue consensuado entre los integrantes de la empresa, verificando cada uno de los atributos como también el nombre de las tablas, las cuales sean representativas cuando uno observa el diagrama de la base de datos en su totalidad y facilite una rápida comprensión de los datos que se almacenan en la base de datos y como se relacionan las tablas entre sí.

Realizamos un trabajo del diseño del diagrama de clases teniendo en cuenta el diagrama de la base de datos, teniendo en cuenta los criterios del nombre de los atributos y de las tablas confeccionamos el diagrama de clases, con un trabajo en conjunto de los integrantes.