



Desarrollo de Software

## Espacio curricular Modulo Programador

---

**Tema: ABP ArgBroker**

**Docente: Ivana Rojas Corsico; Martin Garlero, Maximiliano Giraudo**

**Fecha de Entrega: 28 de Octubre de 2024**

**Cohorte: 2024**

**Alumnos: Laura Zarate, Cristian Vargas, Franco Miranda, Karina Quinteros,  
Cristian Vellio, Nahir Ñañez**

**Grupo: Innova Coders**

**Comisión: 2**

## DOCUMENTACION DE PROGRAMACION ARG BROKER

Este diagrama de clases UML representa la estructura de un sistema de gestión de inversiones, modelando las entidades clave y sus relaciones. El sistema permite a los inversores gestionar sus portafolios, realizar transacciones, y monitorear el estado de sus inversiones.

### Convenciones de nomenclatura UML

- **Clases:** PascalCase, ej. Usuario, Portafolio.
- **Atributos:** snake\_case, comenzando con un verbo en infinitivo, ej. agregar\_accion.
- **Métodos:** snake\_case, comenzando con un verbo en infinitivo, ej. agregar\_accion.

### Descripción de las Clases

#### Entidades del Dominio

- **Inversor:** Representa a un usuario del sistema con información personal y financiera.
- **Accion:** Representa una acción en el mercado de valores.
- **Transaccion:** Representa una compra o venta de acciones.
- **Portafolio:** Representa una colección de acciones de un inversor.
- **HistorialSaldo:** Registra los cambios en el saldo de un inversor a lo largo del tiempo.
- **EstadoPortafolio:** Representa el estado actual de un portafolio en un momento dado.

- **CotizacionDiaria:** Representa la cotización de una acción en un día determinado.

Estas clases utilizan propiedades para controlar el acceso a sus atributos. Estas propiedades actúan como getters y setters, permitiendo encapsular la lógica de acceso y modificación de los atributos.

Por ejemplo, la clase Accion utiliza propiedades para gestionar sus atributos nombre\_accion, simbolo\_accion e id\_accion.

- **Getter:** La función @property en el método nombre\_accion permite acceder al valor de \_nombre\_accion sin necesidad de llamar a una función explícita, usando accion.nombre\_accion.
- **Setter:** La función @nombre\_accion.setter permite modificar el valor de \_nombre\_accion de forma controlada, verificando o procesando el valor si es necesario, mediante accion.nombre\_accion = "NewValue".

De esta forma, la encapsulación mantiene el control sobre cómo se accede y se cambia cada atributo de una instancia de Accion, mejorando la mantenibilidad y la seguridad del código.

## DAOs (Data Access Objects)

- **InversorDAO, AccionDAO, TransaccionDAO, HistorialSaldoDAO, CotizacionDAO, PortafolioDAO, EstadoPortafolioDAO:** Estas clases encapsulan la lógica de acceso a la base de datos para las respectivas entidades.
- **DAOInterface:** Clase abstracta que define una interfaz común para todas las clases DAO, proporcionando métodos para crear, leer, actualizar y eliminar registros.

## Servicios

- **ServicioDeRegistro, ServicioDelInicioSesion, ServicioDeAutenticacion:** Se encargan de las operaciones relacionadas con el registro, inicio de sesión y autenticación de usuarios.
- **CompraAccion, VentaAccion:** Representan las acciones de compra y venta de acciones, respectivamente.

## Otros

- **MySQLConnector:** Se encarga de la conexión a la base de datos MySQL.
- **Menu:** Representa la interfaz de usuario principal del sistema, proporcionando un menú de opciones para el usuario.

## Relaciones entre Clases

- **Asociación:** Representa una relación entre clases, como la relación entre Inversor y Portafolio.
- **Herencia:** Representa una relación "es un" entre clases, como la relación entre InversorDAO y DAOInterface.

## Notas Adicionales

- **DAO:** Las clases DAO encapsulan la lógica de acceso a la base de datos, separando la lógica de negocio de los detalles de implementación de la base de datos.
- **Servicios:** Las clases de servicio proporcionan una interfaz para las operaciones de negocio, como registrar un usuario o realizar una transacción.
- **Menu:** La clase Menu representa la interfaz de usuario principal, proporcionando un punto de entrada para el usuario.
- **MySQLConnector:** La clase MySQLConnector se encarga de la conexión a la base de datos MySQL y proporciona métodos para ejecutar consultas SQL.

