

# Proyecto Integrador: ARGBroker Demo

La empresa tecnológica ISPC Cba se ha inscripto como broker de bolsa para ser intermediario entre los inversores y la **Bolsa de Valores de Buenos Aires Merval**.

Es por ello que le ha solicitado la creación de una aplicación para realizar las transacciones entre los inversores (personas físicas, empresas o instituciones) y otros inversores dentro del Mercado de Valores de Buenos Aires (Merval).

## ¿Qué es una acción de una empresa?

Una acción representa una participación en la propiedad de una empresa, que incluye un derecho sobre las ganancias y los activos de la empresa. Como tal, los accionistas son propietarios parciales de la empresa. Cuando el valor de la empresa aumenta o disminuye, también sube el valor de sus acciones. Además, este valor aumenta o disminuye de acuerdo a la oferta y demanda. Es una forma que tienen las empresas para poder capitalizarse y poder crecer.

## ¿Qué es un broker?

Un bróker es un intermediario que ejecuta órdenes de compra y venta de activos en los mercados financieros. Bróker puede denominarse tanto la empresa que ofrece estos servicios como el corredor de bolsa particular que realiza las operaciones.

## ¿Cuáles son los distintos perfiles de los inversores?

El perfil del inversionista se refiere a las características de una persona que guían la manera en que debiera tomar sus decisiones de inversión, incluido su nivel de tolerancia al riesgo, en relación a los diversos instrumentos de inversión que existen en el mercado. Tiempo de duración de la inversión. Algunos de ellos pueden ser conservador, medio o agresivos.

## Diferencia entre renta fija y variable:

La renta fija y variable se diferencian en la rentabilidad conseguida con la inversión y el riesgo asumido. La renta fija garantiza una rentabilidad limitada con un riesgo mínimo, mientras que la renta variable ofrece mayor rentabilidad pero con un riesgo elevado.

Ejemplo de renta fija: plazo fijo tradicional luego de cierto tiempo estipulado se que me va a dar una compensación por el dinero que deje inmovilizado en un banco con una tasa fijada o en su defecto una tasa fijada más un monto variable de acuerdo a la inflación (plazo fijo UVA).

Ejemplo de renta variable: la acción de una empresa es una renta variable porque no se cuanto puedo ganar o perder al momento de venderla.

## Requerimientos Funcionales:


La aplicación, en su primer etapa, contará con una versión demo (simulación de compra y venta en la bolsa) que tendrá las siguientes funcionalidades:

1- Partir de un saldo inicial de \$1.000.000 (que se asigna al momento de crear un nuevo usuario/inversor).

2- Brindar un panel para consultar las cotizaciones de las **acciones de empresas Argentinas**. Dicho panel deberá mostrar los siguientes datos para cada símbolo/título de acción:

- Símbolo y nombre de la empresa de la acción. Ej.: (ALUA) Aluminio Argentino S.A
- Último Operado
- Cantidad Compra diaria
- Precio de Compra actual
- Precio Venta actual
- Cantidad Venta diaria
- Apertura (precio de la acción al momento de empezar la operación diaria en la bolsa)
- Mínimo diario
- Máximo diario
- Último Cierre.

Ejemplo:

Símbolo	Último Operado	Variación Diaria	Cantidad Compra	Precio Compra	Precio Venta	Cantidad Venta	Apertura	Mínimo	Máximo	Último Cierre	Monto Operado	
 ALUA Aluar	381,00	▲ 0,39%	1,001	379,50	381,00	11,584	380,00	375,00	382,00	379,50	32.679.578	<a href="#">Comprar</a> ☆

3- Permitir **comprar y vender acciones a precio del mercado (en la imagen, precio venta y precio de compra)**. Es decir que en esta etapa no se va a poder poner un precio inferior o superior al que se está comercializando en este momento. Por lo general se pide un precio superior para nuestra venta (hay un porcentaje máximo que se puede pedir) y un precio inferior para nuestras compras intentando comprar al menor precio y vender al mayor precio.

4- Brindar un panel (Mi Portafolio) que muestre el total invertido, saldo de la cuenta demo, la cantidad de acciones compradas con su respectivo símbolo/título, el valor comprometido y, la Ganancia o Pérdida.

5- La aplicación debe estar en español y operar en \$ (pesos) con 2 decimales.

6- Calcular la comisión del broker que es de 1.5% por operación de compra o venta.

**Puesta en marcha:**

Esta actividad formativa, tiene por objetivo integrar los saberes adquiridos en el módulo Programador.

Una vez conformados los equipos, la organización en github y, definido el repositorio de github, los estudiantes deberán:

1. Definir roles,
2. Crear issues (tareas), distribuir y asignar tareas de manera que todos los integrantes del equipo tengan trabajo para hacer. Tip. Distribuyan de manera que las tareas sean independientes, con esto pueden trabajar en paralelo.
3. Configurar el archivo readme especificando el propósito, el contexto, autores.

## Programación I

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

1. Identificar las clases del sistema, atributos y métodos.
2. Nombrar las clases, atributos y métodos en base a la nomenclatura acordada con el equipo.
3. Identificar las relaciones entre clases.
4. Realizar el diagrama de clases mediante herramientas como draw.io, lucidchart, creately, miro u otro.
5. Documentar el diseño especificando:
  - a. Un título significativo. Debe reflejar el propósito del sistema.
  - b. Una breve descripción que explique el contexto, objetivo y alcance del diagrama.
  - c. Nomenclatura elegida para nombres de clases, atributos y métodos.
  - d. Comentarios y/o notas que se considere necesario para proporcionar información adicional o explicar decisiones de diseño.

### Notas:

En esta etapa, no es necesario programar las clases en Python, ni conectar con la base de datos. El enfoque está en el diseño de clases del sistema y las relaciones entre ellas.

### Entregable:

Documento que contenga el diagrama de clases y la documentación del diseño.

## Base de datos

Basándose en los requerimientos funcionales del proyecto, se solicita realizar el diseño de la base de datos para la aplicación ARGBroker Demo.

### En esta etapa, se deberá:

1. Identificar las entidades principales del sistema.
2. Definir los atributos para cada entidad, considerando los datos necesarios para cumplir con las funcionalidades especificadas.
3. Establecer las relaciones entre las entidades identificadas.
4. Aplicar el proceso de normalización a las entidades y sus atributos para eliminar redundancias y dependencias funcionales. Se debe alcanzar la Tercera Forma Normal (3FN).

5. Crear el modelo relacional resultante de la normalización, representando las entidades como tablas con sus respectivos atributos y relaciones. Definir las claves primarias y foráneas necesarias para establecer las relaciones entre las tablas.
6. Documentar el diseño de la base de datos, incluyendo una pequeña descripción de cada tabla. Y las aclaraciones que creas necesarias en cada una de ellas. Por ejemplo si se realizó una suposición de algo que no estaba en el enunciado y eso determinó que una relación es uno a muchos, cuando se podría haber interpretado por una ambigüedad que podía ser muchos a muchos, dejar asentada tal suposición.

#### Notas:

En esta etapa, no es necesario desarrollar consultas SQL. El enfoque está en el diseño y estructura de la base de datos.

Se recomienda utilizar una herramienta de modelado de bases de datos, como draw.io, para crear el diagrama del modelo relacional.

#### Entregable:

Documento que contenga el modelo relacional resultante y la documentación del diseño de la base de datos.

### **Ética**

Basándose en los requerimientos funcionales del proyecto y aplicando la legislación revisada en la cursada:

- Describir cómo el grupo se conforma como instrumento legal (Empresa, Socios, CEO, Empleados) y si tendrán un contrato con que cláusulas.
- Describir quiénes y ante quien deben matricularse a nivel provincial para ejercer la Profesión.
- Describir que requerimientos deben cumplir el código programado y las bases de datos según la legislación vigente
- Enumerar los pasos para que ARGBroker sea parte del Registro Nacional de software.
- Si un integrante del grupo no realiza el trámite de matriculación, que pena le corresponde a nivel provincial.
- Si el código es replicado, describir como la Ley de Propiedad Intelectual puede salvaguardar a ARGBroker.
- Si un integrante del grupo divulga los datos de la base de datos interna, describir como legalmente deberían accionar los demás.

Si otro integrante del grupo reutiliza el código para fines personales, describir como legalmente deberían accionar los demás según la Legislación de la Provincia de Córdoba y el Código Penal Argentino.

- Si otro integrante del grupo reutiliza el código para fines personales a nivel internacional, describir qué instrumento legal respalda a ARGBroker.
- Si la base de datos es adulterada de manera externa, a nivel nacional, describir qué instrumento legal respalda a ARGBroker.
- Si los datos de la base de datos son adulterados de manera externa, a nivel internacional, describir qué instrumento legal respalda a ARGBroker.
- Cuando los programadores de ARGBroker incurren en una negligencia, a que instrumento legal se acude y quien.
- Cómo ARGBroker debe implementar seguridad según la Ley de Protección de Datos Personales.
- Si el cliente decide reemplazar a ARGBroker por otro proveedor, describir cómo se debe actuar de manera ética ante el cliente y los colegas.
- Si un usuario denuncia a ARGBroker por divulgación de sus datos personales, a que legislación recurrió el mismo y como ARGBroker puede respaldarse jurídicamente.

**Todos los integrantes del equipo deberán colaborar con el proyecto mediante tareas (issues) asignadas.**

## Evidencia 6

### Programación I

Partiendo del diseño de clases previo, aplicando TDD (Test Driven Development) y las 4 reglas de diseño simple propuestas por Kent Beck crear las correspondientes clases y el programa de consola que permita al usuario manipular la aplicación.

1- Permitir el registro de nuevo inversor (nombre, apellido, cuil, email, contraseña) e inicio de sesión. Opcional: Recuperación de contraseña y bloqueo al tercer intento fallido.

2- Si el usuario inició sesión satisfactoriamente, permitir las siguientes acciones mediante un menú:

- Permitir mostrar los datos de la cuenta: saldo, el total invertido y rendimiento total.
- Permitir listar los activos del portafolio dónde conste: nombre del activo (empresa), cantidad de acciones (de dicha empresa), los valores de cotización (precio actual para la compra, precio actual para la venta) y, el rendimiento (comparando el precio de compra con el precio de cotización actual)
- Permitir realizar las operaciones de compra/venta de acciones teniendo en cuenta:
  - Validaciones pertinentes. Ej. Validación de existencias.
  - Registrar la correspondiente operación/transacción. No olvidar la comisión del broker.
  - Actualizar la lista de acciones del portafolio, el saldo y el total invertido.

Se valorará el respeto a los siguientes principios: Abstracción, Encapsulamiento, Modularidad, Bajo Acoplamiento y Alta Cohesión.

Se valorará el respeto al patrón de diseño DAO para separar la lógica de acceso a datos de la lógica de negocio.

Opcional, respetar los lineamientos propuestos por el PEP 8. Si se decide no se respetar, se debe escribir un documento que describa mínimamente las convenciones de nomenclatura que utilizarán en el proyecto.

Opcional, pruebas unitarias resultado de aplicar TDD.

Ejemplo sobre convenciones de nomenclatura:

- Archivos y Directorios
  - Archivos Python: Utilizar el formato snake\_case en minúsculas. Ejemplo: main.py, module1.py.
  - Directorios: Utilizar nombres en minúsculas sin espacios ni caracteres especiales. Ejemplo: docs, src.
- Clases
  - Utilizar el formato PascalCase para los nombres de las clases. Ejemplo: Usuario, GestorDeBaseDeDatos.
- Funciones y Métodos



- Utilizar el formato snake\_case en minúsculas comenzando la primera palabra con un verbo en infinitivo. Ejemplo: obtener\_usuario, actualizar\_datos.
- Variables
  - Utilizar el formato snake\_case. Ejemplo: contador, nombre\_usuario.
- Constantes
  - Utilizar el formato snake\_case en mayúsculas. Ejemplo: MAX\_INTENTOS.

## Base de Datos

Como parte del proyecto integrador ARGBroker, se requiere diseñar e implementar una base de datos robusta que soporte las operaciones de la plataforma de intermediación bursátil. Este trabajo se enfocará en el diseño de la base de datos, considerando los requerimientos funcionales del sistema y las particularidades del mercado financiero argentino.

1. Diseñar el modelo relacional de base de datos que soporte las entidades principales del sistema. Pueden utilizar los modelos entregados en la primera parte haciendo las mejoras que crean correspondiente luego de las devoluciones. El diseño debe contemplar lo siguiente:
  - a. Mantener un registro histórico de las cotizaciones de las acciones.
  - b. Registrar todas las transacciones de compra y venta, incluyendo comisiones.
2. Escribe un script o conjunto de sentencias DDL que permita crear la base de datos con todas las tablas replicando el modelo relacional.
3. Escribe un conjunto de sentencias DML de tipo INSERT que inserten datos iniciales en la base de datos.
4. Escribe al menos 5 consultas de tipo UPDATE que permitan actualizar datos de los datos ya insertados.
5. Escribe al menos 5 consultas de tipo SELECT que permitan obtener datos de los datos ya insertados.
6. Escribe al menos 3 consultas multitabla que permitan obtener datos de interés para el caso de estudio. En este caso debe explicar en un pequeño párrafo que desea resolver y luego la consulta correspondiente.
7. Probar todas las consultas en un motor de MySQL online. Y debería funcionar correctamente. Sugiero la plataforma <https://onecompiler.com/mysql>, pero pueden usar la que deseen. Indicar en cual la probaron y la misma no debe requerir registrarse, de modo que a la hora de corregir pueda probar el código en la misma.

## Entregables

**Importante.** Sólo se corrige la rama main. Todos los integrantes deben participar de git mediante confirmaciones (commits). No se requiere participar en todo pero si en al menos una parte.

Repositorio público en github que incluya dos carpetas:

- Programación.
  - Diagrama de clases. Pueden utilizar el diagrama de clases entregado previamente. Tener en cuenta la retroalimentación anterior.

- Clases que representen los objetos de negocio y las operaciones de acceso a datos.
- Programa (por consola).
- Documento sobre convenciones de nomenclatura (si el equipo decide no aplicar PEP 8)
- (Opcional) Pruebas Unitarias resultado de aplicar TDD (Test Driven Development)
- Base de datos
  - Imagen del modelo relacional normalizado en 3 FN (Punto 1)
  - Archivo SQL con las consultas DDL (Punto 2)
  - Archivo SQL con las consultas DML (Punto 3, 4, 5, 6)
  - Archivo PDF que contenga:
    - Link al motor de base de datos online
    - Consultas SQL DDL y DML en formato adecuado para ser copiado y pegado en ese motor, ejecutarlo y comprobar su funcionamiento.

Y el archivo

- Readme (en github). El mismo debe especificar el propósito, contexto, alcance y autores. Pueden utilizar <https://readme.so/es> para su diseño.



## Aclaratorias en relación a Brokers de Argentina

A continuación se listan Brokers que operan con acciones Argentinas. Puedes para ver su funcionamiento creando una cuenta demo para simular operaciones en la bolsa:

- [Invertir online](#). Propiedad del Banco Supervielle. Este broker provee además una Api para desarrollar tus propias herramientas de análisis.  
Link: <https://www.invertironline.com/api>
- [Inviu](#). Propiedad del Banco Galicia.
- [Bull Markets](#).
- [Interactive Brokers](#).
- Entre otros.

Ente que regula el mercado de valores y sus actores:

- [Comisión Nacional de Valores \(CNV\)](#)

Para más información en relación a educación financiera, puedes acceder al siguiente link de la Bolsa de Comercio de Bs. As. <https://www.labolsa.com.ar/capacitacion/invertir/>