



















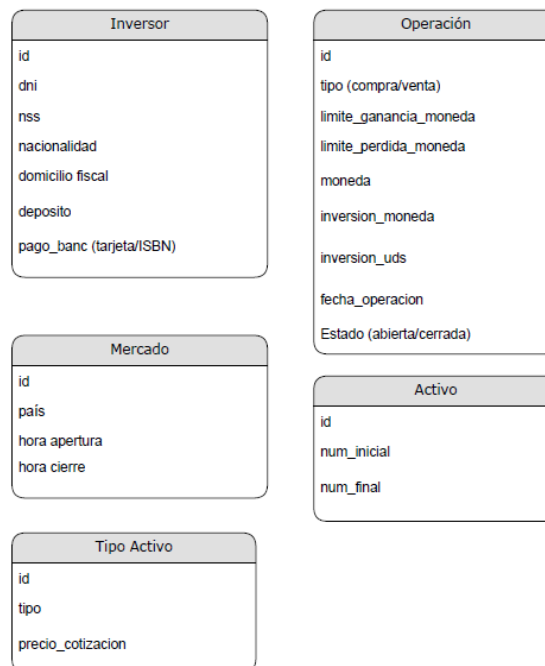
## EXAMEN III 2ª EVALUACIÓN

Fecha: 05/02/2025

Nombre y Apellidos \_\_\_\_\_

	GBPUSD	0.37% (0.0046)		S	1.2480	B	1.2484	88% BUYING
	NSDQ100	-0.13% (-10.05)		S	7839.54	B	7841.94	62% SELLING
	LLOY.L Lloyd's	-0.57% (-0.33)		S	57.20	B	57.32	95% BUYING
	EURGBP	-0.27% (-0.0024)		S	0.9002	B	0.9006	90% SELLING
	AAPL Apple	-0.56% (-1.15)		S	203.29	B	203.66	96% BUYING
	BTC Bitcoin	-1.03% (-100.72)		S	9698.22	B	9771.13	97% BUYING
	OIL	0.12% (0.07)		S	56.75	B	56.80	77% BUYING
	ETHEREUM Ethereum	-0.11% (-0.2328)		S	211.8140	B	215.8769	99% BUYING
	GOLD	-0.39%		S	1422.08	B	1422.53	52% SELLING

Crear una aplicación web que haga de broker o intermediario entre compradores y vendedores en los mercados financieros de diferentes activos tales como: divisas, acciones, criptomonedas, materias primas e índices bursátiles. Se necesita registrar al menos la siguiente información:



CENTRO DE ENSEÑANZA CONCERTADA  
**"Gregorio Fernández"**

Hay que tener en cuenta que los inversores pueden realizar operaciones financieras en cualquiera de los mercados financieros de los países.

Un inversor puede realizar operaciones de compra o venta de varios activos. En una operación pueden participar varios inversores. El activo pertenece de forma única a cada operación. Cada activo pertenece a un tipo de activo. A cada tipo de activo pertenecen muchos lotes de activos. Un tipo de activo es de un único mercado financiero, pero este puede ofrecer muchos tipos de activos.

1. La aplicación se desarrollará como proyecto web en Spring Boot siguiendo la implementando indicada:

- Configuración de dependencias necesarias en el startup "tunombre" perteneciente a la organización com.examen (0, 5ptos)
- Configuración propiedades del proyecto. (0, 5 ptos)

URL datasource: jdbc:mysql://localhost:3306/database  
Driver MySQL : com.mysql.cj.jdbc.Driver

```
spring.datasource.url  
spring.datasource.username  
spring.datasource.password  
spring.jpa.hibernate.ddl-auto  
spring.jpa.show-sql  
spring.datasource.driver-class-name
```

- Proyecto desarrollado utilizando el patrón MVC según el siguiente modelo de empaquetado: (4ptos)

```
└─ tunombre [boot] [devtools]  
  └─ src/main/java  
    ├── com.examen.tunombre  
    ├── com.examen.tunombre.controllers  
    ├── com.examen.tunombre.entities  
    ├── com.examen.tunombre.repositories  
    └── com.examen.tunombre.services
```

2. La persistencia de datos se realizará sobre una base de datos en MySQL llamada "examen" :

- Generación automática y persistente de la base de datos relacional utilizando JPA sobre el API Hibernate de Java en el proyecto creado. (2ptos)
- Testeo de cuatro servicios web distintos (GET, POST, PUT, DELETE) a través de Postman capturando el resultado de los mismos incluyendo el obtenido en la base de datos creada. Se debe de ver en la ventana de Postman la imagen incluida la URL y el resultado de la ejecución del servicio en la base de datos. (3ptos)