



# .NET Challenge

## Introducción

El objetivo de este desafío es evaluar la capacidad del desarrollador para diseñar, desarrollar y desplegar un sistema completo de gestión de órdenes de inversión en el mercado financiero.

El sistema debe permitir a los usuarios realizar operaciones completas de CRUD (Crear, Leer, Actualizar, Eliminar) sobre órdenes de inversión, considerando las particularidades de diferentes tipos de activos financieros.

## Requerimientos técnicos

### API REST

Diseñe y desarrolle una API REST utilizando C# y ASP.NET Core que gestione órdenes de inversión con las siguientes características:

- Implementación de operaciones CRUD completas
- Uso correcto de códigos de estado HTTP
- Documentación de endpoints
- Manejo adecuado de errores y validaciones

### Persistencia de la información

Utilice el método de persistencia que le resulte conveniente como la base de datos principal para almacenar órdenes de inversión, considerando la posibilidad de que el mismo pueda ser reemplazado por otro, y este proceso debe generar el menor impacto posible.

Diseñe un esquema de base de datos adecuado que incluya información relevante para el mercado financiero.

Campo	Tipo de dato	Requerido	Validación
ID de la orden	int	SI	N/A
ID de la cuenta	int	SI	N/A
Nombre del activo	string	SI	Largo máx 32 caracteres
Cantidad	int	SI	mayor que 0
Precio	decimal	SI	mayor que 0
Operación	char	SI	'C' para indicar Compra, 'V' para indicar Venta
Estado	int	NO	Ver Adicionales
Monto Total	decimal	NO	mayor que 0



## Reglas de negocio

### **Cálculo y asignación de valores:**

El campo "Monto total" será automáticamente calculado por la API al crear una nueva orden.

El campo "Estado" también será asignado por la API al crear una orden, iniciando con el estado **"En proceso"**. Este es el único campo que puede ser modificado una vez creada la orden.

### **Vinculación con activos financieros:**

Las órdenes se vincularán exclusivamente a un tipo de activo financiero. Para este ejercicio, se considerarán tres tipos de activos, cada uno con un impacto específico en el cálculo de la orden:

#### **FCI**

El precio recibido es unitario.

El "Monto total" se calcula multiplicando el precio unitario por la cantidad.

No se aplican comisiones ni impuestos.

#### **Acción**

No se debe recibir el precio, ya que este se obtendrá de la base de datos.

Para calcular el "Monto total", se multiplica el precio (obtenido de la BBDD) por la cantidad.

Se deben discriminar comisiones e impuestos:

- Comisiones: 0.6% sobre el "Monto Total".
- Impuestos: 21% sobre el valor de las comisiones.

#### **Bono**

Se reciben tanto el precio como la cantidad.

Se deben discriminar comisiones e impuestos:

- Comisiones: 0.2% sobre el "Monto Total".
- Impuestos: 21% sobre el valor de las comisiones.



## Datos de referencia

### Activos disponibles

```
[
  {
    "id": 1,
    "ticker": "AAPL",
    "nombre": "Apple",
    "tipoActivo": 1,
    "precioUnitario": 177.97
  },
  {
    "id": 2,
    "ticker": "GOOGL",
    "nombre": "Alphabet Inc",
    "tipoActivo": 1,
    "precioUnitario": 138.21
  },
  {
    "id": 3,
    "ticker": "MSFT",
    "nombre": "Microsoft",
    "tipoActivo": 1,
    "precioUnitario": 329.04
  },
  {
    "id": 4,
    "ticker": "KO",
    "nombre": "Coca Cola",
    "tipoActivo": 1,
    "precioUnitario": 58.3
  },
  {
    "id": 5,
    "ticker": "WMT",
    "nombre": "Walmart",
    "tipoActivo": 1,
    "precioUnitario": 163.42
  },
  {
    "id": 6,
    "ticker": "AL30",
    "nombre": "BONOS ARGENTINA USD 2030 L.A",
  },
]
```

```

        "tipoActivo": 2,
        "precioUnitario": 307.4
    },
    {
        "id": 7,
        "ticker": "GD30",
        "nombre": "Bonos Globales Argentina USD Step Up 2030",
        "tipoActivo": 2,
        "precioUnitario": 336.1
    },
    {
        "id": 8,
        "ticker": "Delta.Pesos",
        "nombre": "Delta Pesos Clase A",
        "tipoActivo": 3,
        "precioUnitario": 0.0181
    },
    {
        "id": 9,
        "ticker": "Fima.Premium",
        "nombre": "Fima Premium Clase A",
        "tipoActivo": 3,
        "precioUnitario": 0.0317
    }
]

```

## Tipos de activos

```

[
    {
        "id": 1,
        "descripcion": "Acción"
    },
    {
        "id": 2,
        "descripcion": "Bono"
    },
    {
        "id": 3,
        "descripcion": "FCI"
    }
]

```

## Estados de órdenes

```
[
  {
    "id":0,
    "descripcionEstado": "En proceso"
  },
  {
    "id":1,
    "descripcionEstado": "Ejecutada"
  },
  {
    "id":3,
    "descripcionEstado": "Cancelada"
  }
]
```

## Requerimientos opcionales

### Arquitectura y deployment

Docker: Configuración para deployment en Docker Desktop  
Dockerfiles y docker-compose apropiados

### Seguridad

Implemente medidas de seguridad adecuadas para el uso de la API

### Testing

Pruebas unitarias para validar la funcionalidad crítica  
Cobertura de reglas de negocio principales  
Testing de cálculos por tipo de activo

## Criterios de evaluación

Sistema completo y funcional de órdenes de inversión  
Cumplimiento de las mejores prácticas de desarrollo en C# y ASP.NET Core  
Diseño apropiado y normalizado del modelo de datos  
Claridad y organización de la documentación dentro del código  
Los contenidos optativos no son requeridos, aunque cualquier agregado por fuera de lo propuesto que genere valor agregado será valorado.  
Esperamos que tenga en cuenta el correcto uso de los códigos de estado HTTP



## Entrega

### **Formato de entrega**

- Repositorio Git (preferible) o archivo comprimido
- README.md con instrucciones de instalación y ejecución
- Documentación de API (Swagger recomendado)

### **Plazo de entrega**

Definido en el email donde se adjunta este documento