

#1 Contexto

Una empresa de gestión de viñedos necesita implementar un sistema que maneje la información de los viñedos, gerentes, variedades de uva y registros de parcelas. Los datos se manejan en una base de datos, y la API está diseñada en ASP.NET Core. Los candidatos deberán crear un servicio que exponga una serie de endpoints para gestionar y obtener información de los viñedos, gerentes y parcelas.

#2 Datos Iniciales

-- Tablas de la Base de Datos

Managers (Gerentes)

Id (int): Llave primaria.

TaxNumber (string): Número fiscal del gerente.

Name (string): Nombre del gerente.

Grapes (Uvas)

Id (int): Llave primaria.

Name (string): Nombre de la variedad de uva.

Vineyards (Viñedos)

Id (int): Llave primaria.

Name (string): Nombre del viñedo.

Parcels (Parcelas)

Id (int): Llave primaria.

ManagerId (int): Llave foránea a Managers.

VineyardId (int): Llave foránea a Vineyards.

GrapelId (int): Llave foránea a Grapes.

YearPlanted (int): Año en el que fue plantada la uva.

Area (int): Área total de la parcela en metros cuadrados.

-- Data

Managers

- (1, "132254524", "Miguel Torres")
- (2, "143618668", "Ana Martín")
- (3, "78903228", "Carlos Ruiz")

Grapes

- (1, "Tempranillo")
- (2, "Albariño")
- (3, "Garnacha")

Vineyards

- (1, "Viña Esmeralda")
- (2, "Bodegas Bilbaínas")

Parcels

- (1, 1, 1, 1, 2019, 1500)
- (2, 2, 2, 2, 2021, 9000)
- (3, 3, 1, 3, 2020, 3000)
- (4, 1, 2, 1, 2020, 2000)
- (5, 3, 2, 3, 2021, 1000)

#4 Desarrollo

1. **Lista de Identificadores de Gerentes:** Crea un método que devuelva una lista de los identificadores de todos los gerentes.
Endpoint: *GET /managers/ids*
2. **Gerentes Ordenados por Nombre:** Devuelve los números fiscales de los gerentes ordenados alfabéticamente por su nombre.
Endpoint: *GET /managers/taxnumbers?sorted=true*
3. **Área Total por Variedad de Uva:** Crea un endpoint que retorne un diccionario donde las claves sean los nombres de las uvas y los valores sean la suma total del área plantada para cada variedad de uva.
Endpoint: *POST /grapes/area*

4. **Área Total Administrada por Gerente:** Implementa un servicio que devuelva un diccionario donde cada clave es el nombre de un gerente y el valor es la suma total del área que administra.

Endpoint: *POST /managers/totalarea*

5. **Viñedos y Sus Gerentes:** Implementa un endpoint que devuelva un diccionario donde las claves sean los nombres de los viñedos y los valores sean listas de nombres de gerentes, ordenados alfabéticamente.

Endpoint: *GET /vineyards/managers*

#5 Instrucciones

1. Un proyecto ASP.NET Core funcional.
2. Pruebas unitarias (No hace falta testear todos los endpoints, 2 son suficientes).
3. Instrucciones para ejecutar el proyecto.
4. Código limpio y documentado.
5. No copiar de foros, Stack Overflow o repositorios públicos.
6. **Docker:** El proyecto debe ejecutarse en un entorno de Docker utilizando docker-compose.

#6 Consideraciones

1. **Patrones de Diseño:** Implementa el /los patrones de diseño que más convenga a nivel de desarrollo.
2. **Base de Datos:** Se debe utilizar una base de datos relacional (por ejemplo, SQL Server o PostgreSQL). Se proporcionan las tablas y los datos iniciales a ser utilizados. Se debe manejar el acceso a la base de datos de manera eficiente.
3. **Pruebas Unitarias y/o de Integración:** Es necesario implementar pruebas unitarias utilizando un framework como xUnit, MSTest o NUnit para validar la funcionalidad de los servicios.