



BOOTCAMP BACKEND C#

GUÍA EJERCITACIÓN - DESIGN PATTERNS

1 - Supongamos que tenemos un servicio que guarda en memoria el listado de países y el listado de provincias de cada país. Realizar el Patrón de diseño singleton para que las clases externas puedan acceder a los métodos `getCountries()`, `getProvince(int idCountry)`.

2 - Supongamos que tenemos un servicio que trabaja con 3 bases de datos distintas, y tiene una clase que realiza la conexión para cada una de las bases de datos (`ConexionSQL.cs`, `ConexionMysql.cs`, `ConexionOracle.cs`).

Realizar el método **connect()** de la clase `ConexionFactory.cs`, utilizando el Patrón de diseño Factory Method, para que pueda realizar la conexión a cualquier de estas bases de datos de manera transparente. Cada uno de los métodos **getConnection()** de las 3 clases debe devolver el mensaje "conectando con la clase [nombre clase]".

3 - Supongamos que un banco ofrece distintos productos a sus clientes (`CuentaCorriente.cs`, `CajaAhorro.cs`, `SeguroDeVida.cs`, `SeguroHogar.cs`).

Realizar el método `Clonar()`, utilizando el patrón de diseño Prototype, para realizar la copia de cualquiera de estos productos.

4 - Supongamos que hay dos servicios (`Informante.cs`, `Receptor.cs`) que envían la lista de productos vendidos en el día con los siguientes datos:

fecha, idCliente, idProducto, cantidad, precio, nroRemito.

El problema es que el que envía los datos lo hace a través de un archivo xml, y el que los recibe necesita recibirlos en formato json. Utilizar el patrón de diseño Adapter para realizar la adaptación entre estos dos servicios.