

Prueba técnica Junior Software Engineer

Para evaluar tus conocimientos de programación básicos deberás completar las 3 pruebas que describiremos a continuación.

A tener en cuenta:

- Nos vamos a centrar sobre todo en el proceso que has seguido para resolver cada una de las pruebas y ver cómo piensas, por lo que deberás incluir comentarios, ya sea en el propio código o aparte, explicando el porqué de las cosas. Por favor, intenta detallarlo todo, incluso aquello que pueda parecer muy obvio o básico.
- Para las 2 primeras pruebas utiliza el lenguaje de programación orientado a objetos con el que te sientas más cómodo. No importa que nos pases un programa funcionando, evaluaremos el código sin ejecutarlo.
- Puedes entregar la respuesta de toda la prueba en el formato que mejor te parezca (rar, zip, txt, word, pdf...), pero intenta que sean ficheros adjuntos del mail.

1. Programación

Implementación de un método que indique si una palabra es un **pálindromo**.

Una palabra es un palíndromo si se lee igual de izquierda a derecha y de derecha a izquierda. *Por ejemplo: "radar", "reconocer" y "lol".*

Un ejemplo de cómo programaríamos este método de forma rápida sería:

```
public bool IsPalindrome(string word)
{
    string reversed = word.Reverse(); // Devuelve el mismo string invertido recorriendo todas las letras 1 sola vez
    return word == reversed;
}
```

Contesta a los siguientes puntos:

- a) Escribe tu propia implementación del método `IsPalindrome` utilizando **bucles** y de la manera más **EFICIENTE** posible.
- b) Justifica si tu versión es más eficiente o no que la del ejemplo y por qué (eficiente en cuanto a tiempo de procesamiento y memoria).



NO es necesario tener en cuenta posibles espacios en blanco o nulls ni distinguir entre mayúsculas y minúsculas.

2. Programación orientada a objetos

Se quieren simular con programación orientada a objetos **vehículos** de diferentes tipos. De un vehículo cualquiera, se podrá:

- **Obtener el número de ruedas.** Es un entero fijo para cada tipo de vehículo.
- **Obtener la velocidad actual.** Es un entero. Todos los vehículos parten del reposo.
- **Acelerar.** Incrementa en 1 punto la velocidad.

- a) Implementa el programa de simulación de vehículos que pueda trabajar con dos tipos de vehículo: **coche** y **moto**.
- b) Se quiere ahora añadir a los vehículos la posibilidad de **Frenar**. Añade dicha funcionalidad al programa.
- c) Implementa un nuevo tipo de vehículo: **bicicleta**.
- d) Implementa una nueva funcionalidad, **Repostar**, aplicable a aquellos vehículos que tengan esta necesidad en la realidad.

Justifica en cada apartado por separado todas las implementaciones que realices.

3. Gestión de bases de datos

Disponemos de una base de datos en la que gestionamos los clientes de la compañía y las reservas que éstos han realizado:

Customers		
CustomerID	integer	Primary Key
CustomerName	varchar(50)	
Address	varchar(100)	
PostalCode	varchar(10)	

Orders		
OrderID	integer	Primary Key
CustomerID	integer	Foreign Key (Customers.CustomerID)
OrderDate	DateTime	

Contesta a los siguientes puntos:

a) Realiza una query que devuelva los datos de los clientes que tengan un código postal que empiece por "10" (p.e: "101010", "1045", "1000") [CustomerID, CustomerName, Address y PostalCode]

b) Realiza una query que devuelva los datos de todos los clientes y el número de reservas que ha realizado [CustomerID, CustomerName, Address, PostalCode y NumOrders]
