

SIMULACRO PARCIAL

Ejercicio 1 Complejidad Cognitiva:

- A. Definir el concepto de complejidad cognitiva.
- B. ¿Cuáles son las reglas básicas para calcular la complejidad cognitiva? (Brindar ejemplos de su uso).
- C. ¿Existen otros aspectos a tener en cuenta (además de la complejidad cognitiva) para ayudarnos a lograr el mismo objetivo?.

2. Ejercicio 2 Complejidad Cognitiva/ Linq:

```
List<Alumno> alumnos = new List<Alumno>() {  
    new Alumno("Eva", 20, 6),  
    new Alumno("Ana", 18, 7),  
    new Alumno("Rosa", 22, 5),  
    new Alumno("Ricardo", 30, 9),  
    new Alumno("Felipe", 45, 2),  
    new Alumno("Pepe", 19, 3),  
    new Alumno("Laia", 26, 10),  
    new Alumno("Stephanie", 33, 6),  
    new Alumno("Agustin", 50, 7),  
    new Alumno("Mauricio", 31, 12)  
};  
  
var alumnosExonerados = new List<Alumno>();  
  
foreach (var alumno in alumnosExonerados)  
{  
    if (alumno.Nota > 6)  
    {  
        if (alumno.Nombre.Length > 4)  
        {  
            alumnosExonerados.Add(alumno);  
        }  
    }  
}  
  
for (var indice = 0; indice < alumnosExonerados.Count - 1; indice++)  
{  
    if (alumnosExonerados[indice].Edad > alumnosExonerados[indice].Edad + 1)  
    {  
        var alumnoAuxiliar = alumnosExonerados[indice];  
  
        alumnosExonerados[indice] = alumnosExonerados[indice + 1];  
        alumnosExonerados[indice + 1] = alumnoAuxiliar;  
  
        indice = -1;  
    }  
}  
  
foreach (var alumno in alumnosExonerados)  
{  
    Console.WriteLine($"Nombre: {alumno.Nombre} - Edad: {alumno.Edad} - Nota: {alumno.Nota}");  
}
```

Tomando como referencia el bloque de código brindado a continuación, realizar las siguientes tareas:

- A. Calcular la complejidad cognitiva del método.
- B. Reducir la complejidad cognitiva del método utilizando LinQ.

Ejercicio 3. Responde las siguientes preguntas y/o selecciona la/s opción/es correctas:

A. Los distintos acercamientos que podemos tomar al momento de trabajar con bases de datos relacionales y lenguajes orientados a objetos son

- a. Code First - Objects First - Connection First
- b. Table First - SQL First - Class First
- c. Code First - Database First - Model First

B. Si utilizamos Database First tenemos más control sobre la base de datos

- a. Verdadero
- b. Falso

C. Haciendo uso de LinQ, para obtener la suma de una colección de enteros, debo utilizar la operación

- a. All()
- b. Average()
- c. Sum()
- d. Last()

D. Una de las principales características que debe ofrecer un sistema de control de versiones es que sea seguro y el código este protegido.

- a. Verdadero
- b. Falso

E. Cuando hablamos de sistemas de control de versión. Una rama es:

- a. Una nueva funcionalidad en el código.
- b. Una nueva versión del código, a ser publicada en producción.
- c. Una bifurcación del estado del código.
- d. Ninguna de las anteriores.

F. Los frameworks de acceso a base de datos más populares en .NET son ADO.NET, Entity Framework y Dapper.

- a. Verdadero
- b. Falso

- G. El desfase de impedancia en la programación orientada a objetos es
- a. Una tecnología para acceder a bases de datos relacionales.
 - b. Un ORM.
 - c. Un problema que se presenta interconectar sistemas orientados a objetos y bases de datos relacionales.
 - d. Un micro ORM.
- H. Explica que es un ORM y en que nos ayuda. Enumera cuáles son las principales ventajas de utilizar uno y cuáles son algunas de las desventajas.
- I. Cuáles son las principales diferencias entre las herramientas de acceso a base de datos tradicionales como ADO.NET, los ORM (Como Entity Framework) y los Micro ORM (Como Dapper). Explica en que escenarios es preferible usar uno u otro

Ejercicio 4 - ADO Teórico

- A. ¿Cuáles son las ventajas y desventajas de utilizar ADO Desconectado?
- B. ¿Cuál es la función de los objetos creados a partir de las clases DataAdapter y DataReader en ADO.net?