



Apellido y Nombre: [REDACTED]

TEORÍA

- 1) Un artículo en redes sociales afirma que usar un ORM podría perjudicar la performance de una aplicación por lo tanto NO recomienda su uso. Ud. está de acuerdo? Tiene algo de veracidad la afirmación? Justifique en todos los casos.

Respuesta: es cierto que el rendimiento podría verse afectado pero aun así es recomendado en aplicaciones con modelos de datos complejos donde el rendimiento no es crítico. **BIEN**

- 2) Asumiendo que existe una variable Contexto EF llamada "context" y que existe una entidad Cliente correctamente creada con EF, se quiere Modificar la razón social de un cliente existente (cuyo id es recibido por parámetro), utilizando el siguiente código, sin lograr el resultado esperado:

```
public ActionResult ModificarRazonSocialCliente(int id)
{
    Cliente cliente = new Cliente();

    cliente.Id = id;

    cliente.RazonSocial = "Cliente Modificado";

    context.SaveChanges();

    return RedirectToAction(nameof(Index));
}
```

- a) Describa que está sucediendo, es decir identifique y describa el/los problema/s

Respuesta: se recibe un id por parámetro y se pretende modificar la razón social de un cliente en base de datos con ese id pero en ningún momento se busca el cliente en base de datos, en el código mostrado se crea un cliente, se le modifica en id y la razón social pero solamente en memoria; BIEN

- b) Corrija el código anterior para que finalmente cumpla el objetivo de modificación.

Respuesta:

```
public ActionResult ModificarRazonSocialCliente(int id)
{
    Cliente cliente = context.Clientes.Find(id);

    cliente.RazonSocial = "Cliente Modificado";

    context.SaveChanges();

    return RedirectToAction(nameof(Index));
} BIEN
```

- 3) A partir del siguiente código:

```
public string LINQ()
{
    int[] fibo = new int[] { 1, 2, 3, 5, 8, 13, 21, 34, 55 };
    var fiboOJO = fibo.Where(n => n >= 8);

    fibo[0] = 100; //Linea 3

    string salida = "";
    foreach (int num in fiboOJO)
    {
        salida += $"{num} - ";
    }
}
```

```
}  
return salida;  
}
```

- a) Identifique las partes o pasos de una consulta en LINQ. **MAL**
- b) Indique cual es la salida del método y justifique en relación a la Línea 3. **MAL**

4) Verdadero o falso EF. Justifique en los casos que sea falso:

- a) Luego de recuperar un objeto del contexto de EF, el mismo tiene estado “**recovered**”, ya que está vinculado al contexto. **V MAL**
- b) Una de las limitaciones de EF Core 5 es que NO existe forma de agregar validaciones DataAnnotations a las entidades. **V MAL**
- c) En EF Core 5 es mandatorio/obligatorio usar el enfoque DataBaseFirst, debido a que ningún otro enfoque tiene soporte por Microsoft. **F no es obligatorio dependerá de muchos factores si es recomendable o no MAL. Code First también está soportado.**
- d) LazyLoading es el patrón o mecanismo que permite postergar la inicialización de objetos relacionados hasta el momento que son utilizados y se encuentra activo por defecto en Net 5 / EF Core 5. **V MAL.**

5) Verdadero o falso Web Services. Justifique en los casos que sea falso:

- a) ASPNET Web API se basa en convenciones respecto al nombre de sus métodos y NO permite bindear parámetros complejos (clases) **F se permite bindear parámetros complejos BIEN**
- b) Un servicio ASPNET Web Api utiliza SOAP como protocolo de comunicación. **F puede usar otros protocolos MAL. Utiliza HTTP como protocolo de comunicación.**
- c) Un cliente que invoca a un servicio ASPNET MVC Web api debe usar serialización XML. **F también se usa JSON BIEN**
- d) ASPNET MVC Web Api NO mantiene el estado entre llamadas cliente y el servidor. **V BIEN**

PRÁCTICA (no entregada)

- 1) a) Crear una Tabla “Aula” en SQL Server con los siguientes campos (id integer, nombre varchar(20), CantidadAlumnos integer) b) Crear una aplicación asp.net que tenga una funcionalidad de crear nuevas Aulas, esta página o vista tendrá los controles Nombre Aula, Cantidad Alumnos, correspondientes a la entidad Aula, con un botón grabar Nueva Aula.
c) Utilizando EF Core 5, Enfoque DatabaseFirst, crear el modelo conceptual, en una carpeta llamada “ModeloEF”.
- 2) Generar el código necesario para almacenar una nueva Aula por medio del botón grabar.
- 3) Una vez grabada el Aula, se debe redirigir a una funcionalidad (página o vista) que muestre todas las Aulas grabadas cuya cantidad de Alumnos sea mayor o igual a 20. La forma de mostrar los Alumnos queda a consideración del-a alumna@ (Combo o Grilla o Tabla o similar).
- 4) Crear una nueva funcionalidad “Modificar Aula” (Pagina o Vista). Generar el código necesario a propósito de poder modificar una Aula (Nombre y Cantidad de Alumnos), por medio de un botón Modificar Aula. Luego de modificar el Aula se debe redirigir a la funcionalidad 3).

Aclaraciones:

- Se debe utilizar EF Core 5 y LINQ sin excepción.
- No es obligatorio controlar las validaciones de ingreso de información.

Nota Teoría 6(seis)

Nota Práctica 2(dos)

Matías Paz Wasuchnik