

NOMBRE:

Puntos:

FECHA:

### Falso o verdadero 2pts respuesta correcta

- La mejor forma de crear un índice es que sea sobre una columna identity
- Es bueno tener un alto acoplamiento para tener un buen rendimiento
- Es correcto esta definición de estilo<p>{background-color:red;}
- la diferencia entre una clase abstracta y una interfaz es que una clase abstracta puede definir constructores, métodos, propiedades, tanto públicas como privadas, en cambio una interfaz únicamente puede tener métodos y propiedades públicas.
- Para Usar Using es necesario que el objeto tenga IDisposable implementado.
- En Scrum la estimación de Historias Usuario se hace en la reunión con el cliente
- En LINQ se pueden extraer datos de dos listas en una misma consulta y ordenarlos de acuerdo a un solo criterio
- a. ViewData es el mecanismo para que un controlador proporcione datos a la vista que presenta. Específicamente, ViewData es un diccionario que está disponible tanto en los métodos de acción MVC como en las vistas. Puede usar ViewData para transferir algunos datos de su método de acción a la vista devuelta por el método de acción.
- b. ViewBag es el equivalente dinámico del diccionario ViewData sin tipo. Aprovecha el tipo dynamic.
- c. TempData se puede usar cuando desea conservar datos entre una solicitud http y la siguiente solicitud HTTP solamente. La vida útil de los datos almacenados en TempDataDictionary finaliza después de la segunda solicitud. Por lo tanto, TempData es útil en situaciones en las que estás siguiendo el patrón PRG.  
*una prueba*
- De acuerdo a la teoría no es posible unitaria a un método cuya única función es insertar datos en una Base de Datos

2 puntos respuesta correcta

1. ¿Qué imprime el siguiente query

```
DECLARE @fecha DATETIME = NULL  
SELECT COUNT(1) FROM FActuraCom  
WHERE fecha = @fecha  
GROUP BY fecha  
HAVING COUNT(1)
```

- a. Todos los datos FacturaCom
- b. 1
- c. No devuelve nada
- d. Error

2. ¿Cuáles son verdaderas?

- a. ViewData es el mecanismo para que un controlador proporcione datos a la vista que presenta. Específicamente, ViewData es un diccionario que está disponible tanto en los métodos de acción MVC como en las vistas. Puede usar ViewData para transferir algunos datos de su método de acción a la vista devuelta por el método de acción.
- b. ViewBag es el equivalente dinámico del diccionario ViewData sin tipo. Aprovecha el tipo dynamic
- c. Una variable de sesión persiste durante todos los request, es decir no se destruye al final de un request y puede ser accedida desde cualquier lugar.

1. a y b

2. c y b

3. a y c

Todas

3. ¿Qué ocurre al ejecutar la siguiente consulta?

Create Table1 ( int id)

Create Table2 ( int id)

Insert into Table1 values (2)

Insert into Table2 values ('a')

- a) No se inserta el valor en Table2 y se hace rollback del insert en Table1
- b) No se inserta el valor en el Table2 y no se hace rollback del insert en Table1
- c) No se inserta el valor en el Table2 y se hace rollback de toda la consulta
- d) Ninguna de las anteriores.

try

```
{  
    // Read in nonexistent file.  
    using (StreamReader reader = new StreamReader("not-there.txt"))  
    {  
        reader.ReadToEnd();  
    }  
}  
catch(Exception ex)  
{  
    Console.WriteLine("Error General");  
}  
catch (FileNotFoundException fex)  
{  
    Console.WriteLine("Archivo no encontrado");  
}
```

4. ¿Qué retorna el código anterior en caso de que la ubicación del archivo sea incorrecta

- a. Retorna "Archivo no encontrado"
- b. Retorna "Error General" y "Archivo no encontrado"
- c) Retorna "Error General"
- d. Ninguna de las Anteriores

5. Dentro de qué etiqueta se agrega el trace:

- a) <system.Diagnostic>
- b. <system.Tracing>
- c. <system.model>
- d. <system.logging>

6. Seleccione la solución correcta, para eliminar las tuplas de la tabla Persona donde el campo FirstName sea igual a "Laura"

- a. Delete ROW Persona where FirsName = "Laura"

- b. Drop ROW Persona where FirsName = "Laura"
- c. Truncate Table Persona where FirsName = "Laura"
- d. Delete FROM Persona where FirsName = "Laura"

7. ¿Cuál nivel de acoplamiento y cohesión es preferible en una solución?

- a. Alto acoplamiento y Alta Cohesión
- b. Alto acoplamiento y Baja Cohesión
- c. Bajo acoplamiento y Alta Cohesión
- d. Bajo acoplamiento y Baja Cohesión

8. Las siguientes sentencias están relacionadas con SOLID. Marque solo las verdaderas.

- a. Son patrones de diseño de software
- b. Su uso permite que el software sea más fácil de probar
- c. Facilita la creación de pruebas unitarias
- d. Uno de los patrones es: segregación de dependencias
- e. Todas las anteriores
- f. Ninguna de las anteriores

9. Cuál de las siguientes instrucciones LINQ Válidas

- a. Dim result = From a in animals where a= "gato" Order by Descending
- b. Paralell.For(0, animales.Length Sub(it As Integer) Console.WriteLine(animales(it)) End Sub)
- c. Dim result2 = animales.Select(Function(x)x).Where(Function(y) y = "gato")
- d. Dim result4 = animales.AsParalell().ToLinq()
- e. Dim result3 addHandler animales, new EventHandler(AddressOf result)

10. En la clase StartUp se encuentra el método(2 pts):

- a. Main
- b. ConfigureGlobalization
- c. Create
- d. Configure

11. ¿Qué es Nuget? (3 pts)

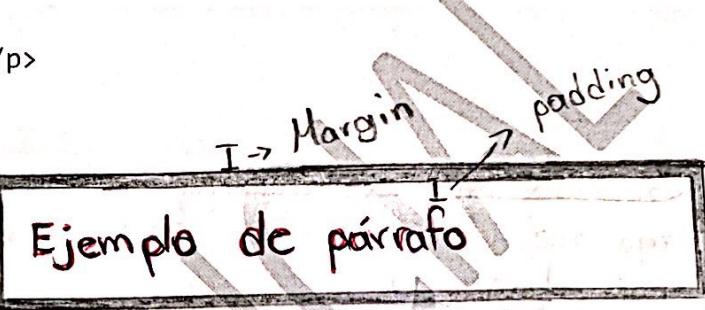
Es un paquete que se puede instalar en los proyectos con los que estemos trabajando, sirve para implementar funcionalidades y herramientas ya realizadas.

12. Escriba en código de JQuery cómo seleccionar el valor del siguiente select

```
<select name="cars" id="cars">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="mercedes">Mercedes</option>  
  <option value="audi">Audi</option>  
</select>  
$.('#cars').val();
```

13. Indique que se imprime en el siguiente código de html y css

```
<div class="viejo">  
  <p id="parrafo">Ejemplo de párrafo.</p>  
</div>  
.viejo {  
  border: 3px solid;  
  Margin: 2px;  
  Padding: 2px;  
}  
.viejo p {  
  color: black;  
}  
p {  
  Color: red;  
}"
```



14. Indique cual o cuales elementos se seleccionan en el siguiente código de JQuery

"\$('div#producto p.tiposProductos')"  
Todos los elementos 'p' con clase "tiposProductos" que estén dentro de un "div" con id "producto"

15. Que es programación paralela y en que escenarios se utilizan

Es la programación de 2 o varios procesos para que se ejecuten al mismo tiempo. Por ejemplo cuando se necesita cargar información de una persona pero esta de consultarse en servicios diferentes.

16. Que es reflection en programación y para que se utiliza

es la capacidad de determinar los objetos o datos en tiempo de ejecución. Se utiliza cuando no sabemos en concreto el resultado de una variable o respuesta.

17. Mencione 2 diferencias entre SOAP y REST (2pts)

Rest solo protocolo Http, soap permite Http y TCP.

Rest- es más eficiente y dinámico que soap

Tiene menos overhead que soap, más ligeros y veloces

18. Mencione y esquile 3 categorías de un binding

- ~~Code~~ Transporte: determina el protocolo de transporte que se utiliza.
- Encoding: determina la codificación del mensaje.
- ~~Seguridad~~.

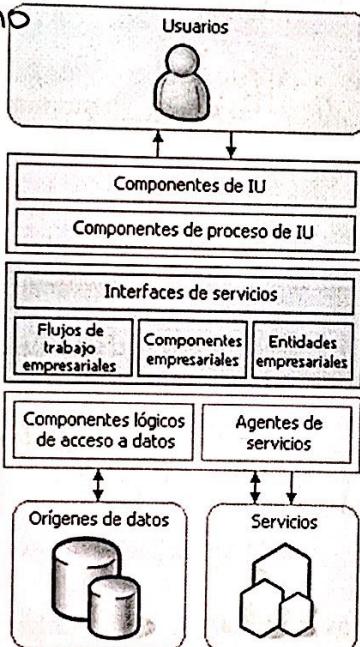
19. Implemente un endpoint en WCF con 2 bindings de tipo tcp y http

Respuesta en hojas aparte.

20. ¿A la siguiente función se le puede hacer una prueba unitaria? Justifique su respuesta(3 pts)

```
Private Function CantidadDiasTranscurridos(fecha As Date) As Integer
    Return DateDiff(Day, fecha, DateAndTime.Today)
End Function
```

No. Toda prueba unitaria debería probar una función  
~~que la respuesta sea la misma siempre cuando se~~  
~~envíe el mismo~~  
~~parámetro.~~



21. Explique cada uno de los componentes de la anterior arquitectura(3 pts)

Tenemos una arquitectura de n capas y se divide en:

- 1 - Interfaz o capa de presentación: acá tenemos todos los componentes visuales y de control.
- 2 - Interfaces de servicios, nos dice que métodos y funciones podemos acceder.
- 3 - Acceso a datos y servicios. Por medio de esta podemos acceder a los datos que están en base

22. Mencione una ventaja y una desventaja de la arquitectura anterior(3 pts) de datos y servicios

Ventaja: permite mantener un orden lógico del acceso a los datos y el tratamiento que se le dan a los datos.

Desventaja: se crean dependencias entre las diferentes capas

23. ¿Qué hace el Content Type y DataType en Ajax? (3 pts)

Nos define el tipo de datos que se envían en la petición "Content Type" y el tipo de datos que retorna la petición. "DataType"

24. Explique, ¿cuál es la diferencia entre Throw y Throw ex? (3 pts)

La diferencia es que el Throw mantiene el traz de la excepción, mientras que el Throw ex reinicia el Traz de la excepción.

25. Defina los siguientes aspectos o ceremonias de scrum: refinamiento de historias de usuario, retrospectiva, reunión diaria, Reunión de Planeamiento. (4 pts)

Refinamiento de Historias:

Definición de los historias de usuarios, se definen la más que se pueda los detalles y puntos de cada historia de usuario.

Retrospectiva:

Reunión para hablar sobre lo que se hizo en el sprint, puntos de mejora, que se hizo bien.

Reunión Diaria:

Reunión de sincronización, ¿qué hice ayer, ¿qué voy a hacer hoy? ¿tengo impedimentos? Se responden esas preguntas por cada miembro del equipo.

Reunión de Planeamiento:

Esta reunión es para definir cuales historias de usuario se van a desarrollar en la iteración. Se define el alcance del sprint.

26. ¿Qué es Binding? (3 pts)

Es la configuración de un servicio para definir el tipo de forma de comunicación del servicio.

27. Explicar los tipos de Binding(3 pts)

WsHttpBinding:

seguro, implementa seguridad en la comunicación. Transaccionales

NetTcpBinding:

utiliza el protocolo TCP y funciona muy bien en intranets.

BasicHttpBinding:

Trae por defecto la seguridad apagada. Utiliza Http y Soap

28. ¿Qué es un WCF? (3 pts)

Windows Communication Foundation, son aplicaciones orientadas a los servicios.

29. En cual etiqueta del binding se agrega el certificado. ~~<security>~~

30. Usando Linq traer la lista de los inquilinos donde el NumLocal sea igual a 1 en base a las siguientes clases:

```
class Edificio
{
    public string Nombre { get; set; }
    public List<Inquilino> Inquilino { get; set; }
}

class Inquilino
{
    public string Nombre { get; set; }
    public int NumeroLocal { get; set; }
}
```

~~var listaInquilinos = listaEdificios.Inquilino~~  
~~.Where(x => x.NumeroLocal == 1)~~  
~~.ToList();~~

31. ¿Qué es razor en MVC?

Es una ~~implementación~~ motor en .Net que nos permite utilizar código del servidor en la vista.

32. Considera la siguiente base de Datos (6 pts)

Empleado		
Nombre	DNI	Sueldo
Juan Mora	2-2222-2222	1.000.000
Ana Rojas	3-3333-3333	256.934
Luis Ramirez	4-4444-4444	1.900.983

Codigo Postal		
Provincia	Poblacion	Codigo Postal
Barcelona	Barajas	123
Madrid	Madrid	100

Telefono	
Numero	DNI
1122-1122	2-2222-2222
1123-1123	2-2222-2222
2222-1212	4-4444-4444

Domicilio		
Direccion	DNI	Codigo Postal
San Adrian	2-2222-2222	123
Plaza Mayor	3-3333-3333	123
Torrejon	4-4444-4444	100

- a. Construya la siguiente consulta: Listado de sueldo medio y número de empleados por población ordenado por población.
- b. Construya la siguiente consulta: Listado de Salario Máximo, Salario Mínimo por provincia, tenga en cuenta la cantidad de personas por provincia.

33. ¿Qué son bloqueos en base de datos?. Mencione dos tipos de bloqueos en base de datos y una forma de evitarlos. (3 pts)

- Es la reservación de recursos para la ejecución de procesos y transacciones.
- Tipos:
  - Exclusivo
  - Compartido
- Para evitarlos es bueno ponertos solo si es necesario y de forma responsable.

34. Explique ¿qué es TempData, ViewData y ViewBag. (3 pts)

~~ViewBag~~: es un tipo de variable que nos permite enviar datos del controlador a la vista, solo se mantiene en el request actual.

~~TempData~~: es un tipo de variable que está disponible por hasta 2 request, nos permite enviar datos a la vista a otro action del controlador.

~~ViewData~~: tipo de variable con el que podemos enviar datos a la vista, desde el controlador y se mantiene solo en el request actual.

35. Explique ¿porqué son importantes los patrones de diseño y explique uno? (3 pts)

Porque nos permiten solucionar problemas con patrones que ya se han probado y depurado.

MVC: el Model, View, Controller es ~~un~~ patrón de diseño que nos da una arquitectura con un orden lógico para aplicaciones web.

36. ¿Cuál sería la salida en consola al ejecutar el siguiente método, simulando que el valor ingresado es

20? (3 pts)

```
Public Sub Metodo()
    Try
        Dim posicion As Integer
        Dim vector(9) As Integer
        Dim numero = 100
        Console.WriteLine("Ingrese una posición")
        posicion = Convert.ToInt32(Console.ReadLine())
        vector(posicion) = numero
    Catch ex As FormatException
        Console.Write("Formato incorrecto")
    Catch ex As IndexOutOfRangeException
        Console.Write("Posición fuera de lugar")
    End Try
    Console.Write("Programa")
    Console.ReadKey()
End Sub
```

~~Formato incorrecto~~

~~Programa. Ingrese una posición.~~

~~Posición fuera de lugar~~

~~Programa.~~

37. Mencione 3 ventajas de la arquitectura de microservicios. (6pts)

- \* Se distribuye el procesamiento.
- \* Escalable
- \* Bajo acoplamiento

38. Describa la diferencia entre una base de datos monolítica y una con microservicios. (6 pts)

Una base de datos monolítico tiene todas sus tablas, sp, funciones, todo en una misma base de datos.

Mientras que la ~~de~~ de microservicios tiene todo separado en distintas bases de datos y servidores.

39. Genere una consulta en Linq que devuelva la lista de nombres de los productos, agrupados por Nombre donde el nombre del Producto no sea "Maquinaria". (6 pts)

Var lista = From p in productos where p.<sup>nombre</sup> != "Maquinaria"  
group p by p.nombre lista select lista.ToList();

40. Realice una sentencia en SQL que determine el conteo de los registros repetidos en la tabla mostrada a continuación. (6 pts)

```
CREATE TABLE VIAJES
(
    idViaje int Primary Key identity(1,1),
    descripcion varchar(max),
    idViajePersona int not null
    Constraint FK_Viaje_Persona FOREIGN KEY (idViajePersona) REFERENCES VIAJE_PERSONA(idViajePersona)
);
```

Select count(idViaje) AS repetidos from viajes having count(idViajePersona) > 1

41. Que es un deadlock y como los resuelve el motor de base de datos de SQLServer? Como los evitaría? (3pts)

Es cuando 2 procesos o transacciones bloquean recursos que ambos necesitan para terminar, entonces la base de datos no responde. El sql usa varios motivos para terminar alguno de los procesos, entre ellos es el tiempo de la transacción, termina el que tiene menos tiempo para concluir el otro proceso. Los evitaría usando los bloqueos solo cuando son necesarios. <sup>Niveles de aislamiento.</sup> 10

42. Analice el siguiente código:

```
9  @Using (Html.BeginForm("Guardar", "Home"))
10 @Html.AntiForgeryToken()
11 @div class="form-horizontal">
12     <div id="laHora"></div>
13     <h4>Agente</h4>
14     <br />
15     @Html.ValidationSummary(True, "", New With {.class = "text-danger"})
16     <div class="form-group">
17         @Html.LabelFor(Function(model) model.Nombre, htmlAttributes:=New With {.class = "control-label col-md-2"})
18         <div class="col-md-10">
19             @Html.EditorFor(Function(model) model.Nombre, New With {.htmlAttributes = New With {.class = "form-control"}})
20             @Html.ValidationMessageFor(Function(model) model.Nombre, "", New With {.class = "text-danger"})
21         </div>
22     </div>
23     <div class="form-group">
24         <div class="col-md-offset-2 col-md-10">
25             <input type="submit" value="Crear" class="btn btn-default" />
26         </div>
27     </div>
28 </div>
29 End Using
30 @Section Scripts
31 <script>
32     $document.ready(actualizar());
33     function actualizar(){
34         var div = $("#laHora");
35         var date = '@DateTime.Now.ToString("yyyy-MM-dd")'
36         div.innerHTML= "La hora es: " + date;
37         setTimeout(div, 1000)
38     }
39 </script>
40 @Scripts.Render("~/bundles/jqueryval")
41 End Section
```

De acuerdo al código anterior, conteste las siguientes cuatro preguntas:

a. ¿En dónde se ejecuta la línea 35 de código y que es lo que hace? (2pts)

~~"en el cliente y en el servidor, ya que le está asignando al "var date" la hora actual del servidor.~~

b. Según lo que se busca es que el programa muestre en pantalla la fecha y hora en un div y que este se actualiza cada segundo, cumple este propósito el ejemplo anterior, ¿explique porque lo cumple o porque no lo cumple? (2pts)

~~No lo cumple, en este caso en el setTimeout debería invocar a la función actualizar, también el "ToString("yyyy-MM-dd")" castea la fecha y hora a solo fecha, es decir, te quita el campo de tiempo.~~

#### Algoritmos

- Realice un algoritmo recursivo que reciba como parámetro un número entero y que devuelva el número invertido (no utilizar string) (10 pts)

Ejemplos de salida:

943 → 349

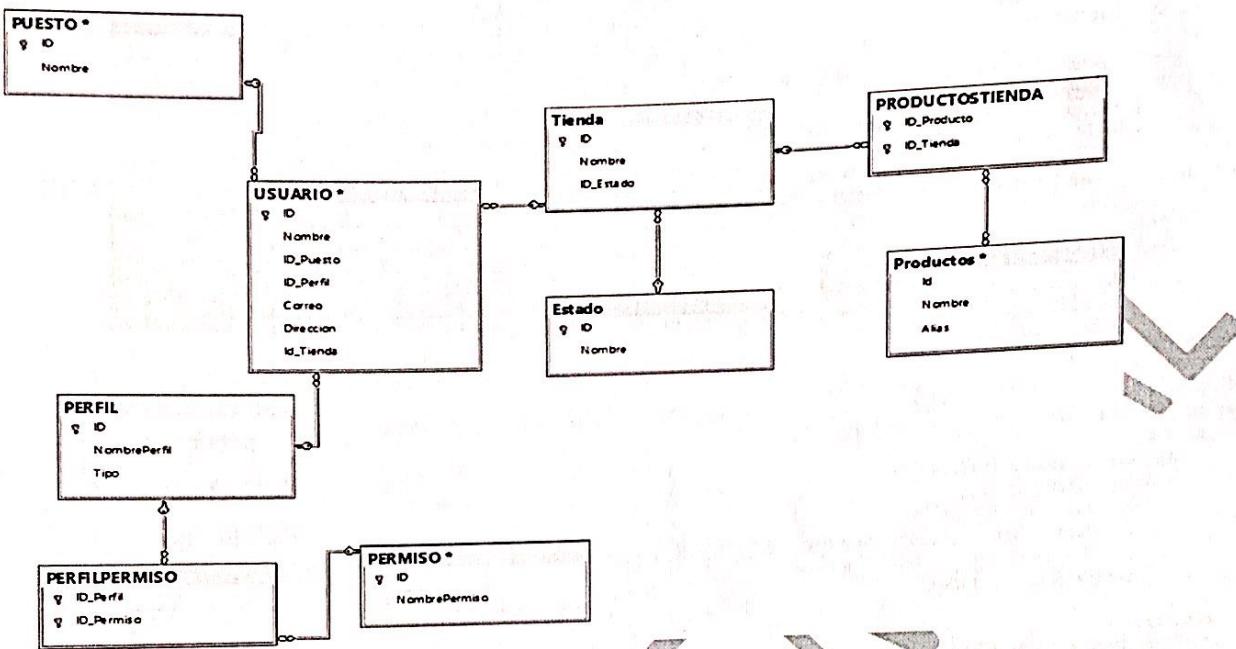
17 → 71

3 → 3

## Algoritmos 1 Recursivo.

```
public int invertirNumero (int num, int rev = 0) {  
    if (num < 10) {  
        return num;  
    }  
    else {  
        return invertirNumero (num/10, rev * 10 + num % 10);  
    }  
}
```

2. Realice las siguientes consultas en base al diagrama



a. Cree una vista que obtenga todos los productos que no están en una tienda con estado activo  
(3pts)

b. Obtener la cantidad de usuarios por puesto donde un perfil tenga más de un permiso (2pts)

43. A continuación, se le presenta unos fragmentos de código los cuales no aplican correctamente los principios solid. Para cada fragmento explique con cual o cuales principios no se está cumpliendo y escriba una posible solución en código. 7 pts

No aplica el principio de Open/closed y d de ~~g~~ Inversión de dependencias.

public interface IVehiculo public class Vehiculo : IVehiculo

```

    {
        void Acelerar();
        void Detener();
    }

    public void Acelerar() {
        console.WriteLine("Acelerar");
    }

    public void Detener() {
        console.WriteLine("Detener");
    }
}
  
```

## PRUEBA TÉCNICA

```

interface ICarro
{
    4 references
    void Acelerar();
    2 references
    void Detener();
}

2 references
public class Carro : ICarro
{
    4 references
    public void Acelerar()
    {
        Console.WriteLine("Acelerar");
    }

    2 references
    public void Detener()
    {
        Console.WriteLine("Detener");
    }
}

2 references
public class CarroElectrico : ICarro
{
    4 references
    public void Acelerar()
    {
        Console.WriteLine("Acelerar");
    }

    2 references
    public void Detener()
    {
        Console.WriteLine("Detener");
    }
}

```

```

public void imprimir_carro(string tipoCarro)
{
    if(tipoCarro == "carro")
    {
        Carro carro= new Carro();
        carro.Acelerar();
    }
    else if(tipoCarro == "carroElectrico")
    {
        CarroElectrico carroElectrico = new CarroElectrico();
        carroElectrico.Acelerar();
    }
    else
    {
        throw new Exception("Carro Invalido");
    }
}

```

~~public void imprimir\_vehiculo(Vehiculo miVehiculo)~~

~~{~~

~~miVehiculo ?? != null~~

~~miVehiculo.Acelerar();~~

~~throw new Exception ("Vehículo invalido");~~

~~}~~

44. Explique cada letra de Solid y mencione un ejemplo práctico(10pt)

~~Se responde en una hoja aparte.~~

45. Se debe desarrollar una función que reciba un string con un numero binario, y retorne un el numero en decimal integer EJEMPLO: `pasardecimal("1010") = 10`  
 Sin asumir que ya hay funciones que ya realizan esto en .NET. Utilizar solamente código C# o VB. (10pts)

1010  
8421

46. Realice un algoritmo recursivo que reciba como parámetro un número entero y que devuelva el número invertido (no utilizar string) (5pts)

44. respuesta corta.

Single responsibility: Métodos y funciones deberían hacer una única responsabilidad.

Open/Closed: el código debe estar manteniendo <sup>y abierto</sup> para nuevas funcionalidades, pero cerrado para modificaciones.

Liskov Substitution: Las clases hijas deben permitir diferente comportamiento sin afectar a sus clases padres.

Interface Segregation: Segregación de interfaces. Si una interface tiene mucha funcionalidad se debe considerar separarlas.

Dependency Inversion: Se deben evitar sentencias "new". La inyección de dependencias nos permite bajar el acoplamiento entre clases.

45 → Algoritmos.

```
public int pasarDecimal(string binario) {
```

```
    int numero = 1;
```

```
    int decimal = 0;
```

```
    for (int i = binario.Length - 1; i >= 0; i--) {
```

```
        if (binario[i] == 1) {
```

```
            decimal = decimal + numero;
```

```
}
```

```
        numero = numero * 2;
```

```
}
```

```
    return decimal;
```

```
}
```