**PREGUNTAS DE RESUMEN BIG BANG**

**ENTITY FRAMEWORK**

1. ¿Cuáles son los 3 tipos de entity framework?

Code First: Genera la base de datos a partir del código

Database First: Genera el código a partir de la base de datos

Model First: Genera el código y la base de datos a partir de un asistente visual

1. Mencione tres ventajas de entity framework

Fácil de utilizar

Se instala mediante nuget

Flexible y fácil de modificar

1. ¿Cuáles son los 3 componentes de entity framework?

Conexión: Maneja excepciones y se encarga de abrir y cerrar la conexión

Command: Se encarga de ejecutar consultas

DataReader: Carga e itera los datos como un cursor, lee los registros uno a uno

1. ¿Cuáles son las 4 reglas de entity framework?

Busca el nombre del String de Conexión según en nombre del contexto.

Un atributo con Id al final de la palabra va a ser tomado como la llave primaria.

Se debe usar tracking cuando se va a modificar o eliminar un registro.

Cuando se sabe de antemano que no se va a modificar un registro se debe usar AsNoTraquing

1. ¿Cuáles son los 3 componentes de entity framework?

EntityFramework se compone de:

String de Conexión: se encuentra en el archivo de configuración (web.config, app.config, machine.config) define la conexión a las base de datos

Clases de entidad: son las que van a representar tablas de base de datos, está compuesta de constructor público y una entidad que representa la llave primaria.

La clase Contexto: Es donde se agregan las entidades como dbset, une las piezas (la clase entidad con una tabla en base de datos) y utiliza el Connection String

**JAVASCRIPT**

1. ¿JavaScript es compilado o interpretado?

Es interpretado que se encarga de agregar interactividad a las páginas HTML

1. Cuatro de las principales funciones de JavaScript son:

Capturar eventos

Validar datos

Agregar texto dinámico a las páginas

Crear Cookies

1. ¿Qué pasa si se añade un número y una cadena?

Un número más una cadena será igual a una cadena

1. ¿Qué hace ´confirm´ y ´prompt´?

Son tipos de mensajes de alerta

confirm agrega los botones aceptar y cancelar

prompt para agregar texto e incluye los botones agregar y cancelar

1. Mencione y explique 5 eventos de JScript

OnLoad se dispara cada vez que el usuario entra a la página

OnUnLoad se dispara cada vez que el usuario sale de la página

OnSubmit valida los campos del formulario antes de enviar los datos

OnChange validación del estado de entrada

OnMouseOver para el comportamiento relativo del cursor, usado en animaciones.

**JQUERY**

1. Defina Jquery

Es una librería de javascript más liviana para escribir código javascript en menos líneas

1. ¿En qué se diferencia de JavaScript?

Jquery puede hacer lo mismo que Javascript pero con menos líneas de código

1. ¿Qué función reemplaza el símbolo $()?

Reemplaza la función JQuery()

1. ¿Cómo agrega color al fonde de una página?

$ (‘body’) .css (‘background’, ‘ ’);

1. ¿Cómo oculta un elemento?

$(“elemento”).hide()

1. ¿Tres ventajas de Jquery son?

Flexible y rápido

Es open source

Integración con Ajax

1. ¿Qué es el DOM?

Modelo de Objetos de Documento. Es un protocolo que permite la comunicación entre Javascript y Html

1. Seis métodos de ejemplo de Jquery con los que se puede manipular CSS son:

.css ()

.attr ()

.addClass ()

.removeClass ()

toggleClass ()

removeAttr ()

**MVC**

1. ¿Qué es MVC?

Un patrón de diseño que separa la lógica en tres capas, Modelo, Vista y Controlador

1. Explique sus partes

**Modelo:** Contiene cada una de las entidades que representan la lógica que se utilizará

**Vista:** Es la interfaz gráfica que comunica con el usuario, muestra visualmente el modelo

**Controlador:** Es la capa que se encarga de comunicar la vista con el modelo mediante eventos activados por el usuario

1. Cuatro ventajas de MVC son:

Bien organizado y distribuido

Menor acoplamiento

Facilidad para realizar pruebas unitarias

Mejor separación de tareas al tener por un lado la vista y por otro parte la lógica

1. Dos desventajas de MVC son:

En caso de que no se tenga conocimiento va a extender los tiempos del desarrollo por la curva de aprendizaje

Más costosa la implementación para lenguajes que no soportan este paradigma

1. ¿Con MVC se tiene un control total sobre como las URLs serán formateadas o mostradas en el navegador? Explique su respuesta.

**Con MVC se tiene un control total sobre las URLs ya que se puede configurar desde el Route.Config**

**ORDENAMIENTOS**

1. ¿Qué es un ordenamiento?

Consiste en agrupar registros en un orden definido

1. ¿Cuál es su propósito principal?

Facilitar búsquedas, reduciendo el tiempo

1. ¿Cuándo un ordenamiento es eficiente?

Es cuando se realiza la menor cantidad de iteraciones posibles

1. ¿Cuándo un ordenamiento es interno? Dé 2 ejemplos

Son aquellos en los que los valores a ordenar se encuentran en memoria principal.

Ejemplo: QuickSort, Burbuja

**PATRONES DE DISEÑO**

1. ¿Qué son patrones de diseño?

Es una solución general y de eficacia comprobada que se aplica a un problema en específico

1. ¿Son exclusivos de la programación orientada a objetos?

Son Exclusivos de POO

1. ¿Cuáles son los 3 tipos de Patrones de diseño que hay?

Estructurales

Creacionales

Comportamiento

1. Mencione y explique 7 de las más importantes

**Creacionales**

Abstract Factory: Es un modelo creacional donde se define una interfaz para la creación de objetos relacionados

Singleton: Modelo creacional para asegura de que se cree una única instancia global. Tiene el constructor privado y una propiedad de acceso estática.

**Estructurales**

MVC: Es un modelo estructural que separa la lógica en capas Modelo, Vista y Controlador

Adaptador: Permite que 2 clases con interfaces incompatibles trabajen juntas

Puente desacopla una abstracción de su implementación permitiendo modificarlas independientemente

Decorador: Añade funcionalidad a un objeto sin modificarlo.

**Comportamiento**

Observer: Permite suscribirse a una serie de eventos que otro objeto va a emitir.

Memento: Almacena el estado del objeto y lo restaura posteriormente.

1. ¿Cuáles son los 4 elementos de un Patrón de diseño?

Nombre

Problema

Solución

Consecuencia

**PROGRAMACION ORIENTADA A OBJETOS**

1. ¿Qué es la POO?

Es un paradigma de programación que permite abstraer objetos de la vida real en estructuras que llamaremos entidades o clases que cuentan con propiedades y atributos describen su comportamiento.

1. **¿Cuáles son los 7 componentes principales de POO? Explíquelos (Herencia, acoplamiento y cohesión, ocultamiento y encapsulamiento, abstracción, y polimorfismo)**

Herencia: Capacidad que tiene una clase de heredar el comportamiento (atributos y métodos) de otra.

Acoplamiento: Es el grado de dependencias entre objetos.

Cohesión: Es el grado de coherencia de una clase, en otras palabras que la clase haga solamente lo que le corresponda y tenga una única responsabilidad.

Ocultamiento: la capacidad de ocultar atributos o métodos a otros componentes mediante los modificadores de acceso (public, private, protected, etc)

Encapsulamiento: consiste en encapsular los atributos para que estos no sean modificados ni accedidos directamente sino solamente por métodos get y set.

Abstracción: Conocer o entender las funciones, comportamientos y características de una clase sin conocer como las realiza.

Polimorfismo: La capacidad que tiene un objeto de comportarse de diferentes formas (tener métodos con el mismo nombre, misma cantidad y tipo de parámetros pero con comportamiento diferente) mediante la herencia

1. **¿Qué es Clase y Objeto?**

Una Clase es la definición de una entidad donde se declaran sus características y su comportamiento. Es una estructura de una entidad.

Un objeto es una instancia en memoria de esa clase

1. ¿Qué son atributos Estáticos (o Shared en VB)?

Son los que se pueden acceder sin necesidad de instanciar un objeto

1. **¿Qué es Garbage Collector? ¿Cuándo se ejecuta? ¿Se puede ejecutar manualmente? ¿Cómo?**

**El Garbage Collector es el que se encarga de administrar y liberar la memoria que no se está utilizando.**

**Se ejecuta mediante el motor de optimización que es el que decide cuando liberar la memoria**

**Se puede ejecutar manualmente System.GC.Collect**

1. Respecto a ocultamiento: mencione y explique 5 modificadores de acceso.

Friend: El elemento es accesible por cualquier otro elemento solo dentro del compilado

Public: El elemento es visible cualquier otro elemento

Private: El elemento es accesible solamente dentro de la clase

Protected: El elemento es accesible solo dentro de la clase y dentro de las clases derivadas

Protected Friend: El elemento es accesible solo dentro de la clase y las clases deribadas tanto fuera como dentro del compilado

1. Respecto a polimorfismo, mencione y explique 3 tipos.

Polimorfismo de Sobrecarga de métodos: Es cuando 2 funciones de diferentes clases con el mismo nombre y misma cantidad de parámetros tienen diferente comportamiento

Polimorfismo paramétrico: Es cuando 2 métodos con el mismo nombre reciben parámetros diferentes

Polimorfismo Subtipado: Es cuando se llama a un método sin tener que conocer su tipo directamente.

1. ¿Qué es herencia simple?

Que una clase derivada solo puede heredar de una sola clase padre

1. Respecto a herencia, ¿qué es herencia con sobre escritura? ¿Cómo se implementa en VB.NET?

Herencia con sobre escritura es cuando se declara un método en la clase padre y se indica que este se podrá sobre escribir, en otras palabras, quien herede de esta clase podrá implementar el mismo método con diferente comportamiento.

En VB.NET se implementa declarando el método en la clase padre como overridable y en la clase deribada como overrides o overload

1. ¿Qué es shadowing?

**Cuando una clase derivada declara un método del mismo nombre que el de la clase padre como overrides, esto ocultará el método del padre y se ejecutará el del hijo**

1. ¿En qué se diferencia shadowing de herencia? Dé un ejemplo\*\* <https://www.c-sharpcorner.com/article/difference-between-shadowing-and-overriding-in-oop/>

**Shadowing oculta un miembro del padre para que se implemente el del hijo, en herencia se hereda el método del padre y si no se declara como overridable se implementa la funcionalidad del padre**

44. Respecto a herencia, si un método no se declara como overridable (en clase abstracta o no, es indiferente), ¿se puede heredar? ¿qué diferencia hace declararlo como overridable?\*\* <https://www.thoughtco.com/overrides-in-vbnet-3424372>

Si hereda el método de la clase padre, la diferencia es que no se va a poder sobrescribir sino que implementará el método que tiene definido el padre

1. ¿Cómo hago una referencia explícita a un método o propiedad de la clase base?

Utilizando la palabra reservada MyBase

1. Respecto a una clase abstracta: defínala; ¿se puede instanciar? ¿Qué es lo único que puede hacerse con ella?

**Una clase abstracta es aquella que no se puede instanciar directamente sino que define métodos y atributos para que otras puedan heredar de esta.**

**Una clase abstracta puede definir un constructor pero solo se puede invocar desde sus clases derivadas**

**No se puede instanciar, solo heredarse**

**Puede tener métodos y propiedades abstractos que deben ser obligatoriamente sobrescritos**

1. ¿Cómo se declara en C# y como en VB.NET; pueden tener constructor? ¿Desde dónde se puede invocar? ¿Puede tener propiedades y métodos? deben ser obligatoriamente sobrescritos? Explique.

En C# se declara como Abstrac y en VB.NET como MustInherit

La clase abstracta puede tener constructor pero solo se puede invocar desde la clase derivada

Puede tener propiedades y métodos pero estos deben ser obligatoriamente sobrescritos

1. ¿En qué se diferencia la clase abstracta de las interfaces?

En que una clase puede heredar de una sola clase abstracta, en cambio una clase puede implementar varias interfaces.

1. Si deseamos que una clase no pueda ser heredada, ¿Cómo debe declararse?

**NotInheritable**

1. ¿Puede una clase abstracta tener métodos implementados (no solo nombre, parámetros y tipo de retorno, sino funcionalidad)?\*\*   
   <https://www.dotnetperls.com/mustinherit-vbnet>

**Una clase abstracta puede tener métodos implementados**

1. ¿Qué definen los Campos o Miembros de clase?

**Definen el estado de una clase**

1. ¿Qué es una Propiedad?

**Es la que define métodos de acceso para un atributo de una clase. Lectura/Escritura, Solo lectura o solo escritura**

1. ¿Qué es el Constructor? Como se declara en VB?

**Es el que inicializa los miembres de una clase**

**Se declara como Public Sub New()**

1. ¿Puede una clase tener varios constructores? Explique.

**Efectivamente, se puede aplicar sobrecarga en los constructores de una clase.**

**Puede haber más de un constructor pero con diferentes parámetros**

1. ¿Qué pasa con el compilador si declaramos un constructor con parámetros?

**Ya no creará un constructor por defecto**

1. ¿Cómo se define un método que devuelve parámetros y el que no lo hace?

Sub para un método que no devuelve parámetros y Function para los que devuelven parámetros

1. ¿Qué es un método abstracto? ¿Cómo se declaran en C# y en Visual Basic? ¿Se pueden declarar fuera de clases abstractas?

**Son métodos que solo se pueden declarar dentro de clases abstractas y estos obligatoriamente se deben implementar en las clase derivada.**

**En C# se declaran como Abstract y en VB.NET como MustOveride**

1. ¿Los atributos privados de una clase son accesibles por sus clases derivadas? Si no es así, ¿cómo son accesibles?

**Los atributos privados no son accesibles por su clase derivada, en ese caso se deben declarar como Protected para que solo sean accesibles para la clase derivada y no visibles en las demás clases**

1. ¿Qué es sobrecarga y que permite respecto a constructores?

**En constructores la sobrecarga permite instanciar un objeto de diferentes formas teniendo varios constructores con diferentes parámetros**

1. ¿Qué se necesita para que una sobrecarga sea válida respecto a número o tipo de parámetros?

**El número y tipo de parámetros no puede ser igual**

1. ¿Se pueden declarar dos métodos con el mismo nombre y además con el mismo tipo y cantidad de parámetros?

**No**

1. ¿Cuál es la diferencia entre ByVal y ByRef?

**ByVal recibe una copia del valor y ByRef recibe la referencia en memoria**

**Para objetos ByVal pasa una copia del puntero pero no se hace una copia del puntero en memoria y ByRef pasa una referencia del puntero lo cual es ineficiente.**

1. ¿Qué modificadores de acceso para controlar la visibilidad tienen las clases y sus elementos por defecto?

**Friend por lo cual los elementos estarán visibles dentro del compilado**

1. Describa las 5 partes de SOLID con ejemplos para cada una.

**Responsabilidad Única:** una clase debe tener una única responsabilidad, en caso de que tenga más de una responsabilidad se debe separar en otras clases.

Ejemplo: una clase rectángulo que tenga los métodos CalcularSumaAreas y CalcularSumaPerímetros. Estos 2 métodos deberían separarse en una clase aparte Operaciones que se encargue de realizar los cálculos.

**Abierto/Cerrado:** Las clases deben estar abiertas a la extensión pero no a la modificación. Deben poder extender la funcionalidad sin ser modificadas

Ejemplo si en un método tengo varios If o un case que realice una operación según el tipo, en caso de que quiera agregar otro tipo, debo modificar el método para incluir el nuevo cálculo por lo cual estaría incumpliendo con el principio ya que una clase debe estar cerrada a modificación. Lo que se debería hacer es pasar los tipos a objetos con un método que realice el cálculo, e implementar una interfaz para que los tipos hereden de esta. Así en el método general llamaría solamente al objeto.calcular y mediante polimorfismo podría agregar más tipos sin que modificar la clase, quedando la funcionalidad abierta a extensión.

**Sustitución de Liskov:** Los objetos podrían reemplazarse por las instancias de sus subtipos sin generar errores.

Ejemplo: No se debería implementar una interfaz con un método donde una clase lo implemente y en otra no se defina su implementación. En este caso debería separar la funcionalidad en 2 interfaces e implementar solamente los métodos que realmente se requieren en una clase heredando así de las interfaces que se debe implementar su funcionalidad. Con esto podrías reemplazar las instancias por sus subtipos sin generar errores.

**Segregación de Interfaces:** Una interfaz no debería tener muchas responsabilidades, si tiene muchas funcionalidades debería redefinirlas, separándolas en más interfaces. Por lo cual va ligado al principio de responsabilidad única. Además si se requiere agregar funcionalidad se crear una nueva interfaz.

Ejemplo: una interfaz que realice 2 cálculos que no tengan relación el uno con el otro, sería mejor separar los métodos en 2 interfaces.

**Inversión de dependencia:** La abstracción no debe depender de los detalles sino los detalles deben depender de la abstracción.

Ejemplo: Una clase no debería ser la encargada de crear o instanciar objetos para luego aplicar alguna operación sino que esta debería recibirlos por parámetros.

1. Describa cuales serían 7 buenas prácticas de código mantenible.

**No escribir métodos de más de 20 líneas**

**Las funciones deben tener más de 2 niveles de anidamiento**

**No usar valores directamente que no especifican que representan, mejor pasar ese valor a una constante que describa que representa ese valor.**

**Evitar el scroll tanto horizontal como vertical**

**Utilizar nombres significativos**

**Evitar declaración de variables de un solo caracter**

**No implementar lo que no se va a utilizar**

**SCRUM**

1. ¿Qué es SCRUM? ¿Es un proceso o técnica? Sino... ¿Qué es?

Scrum es un proceso en el que se aplican de manera regular [un conjunto de buenas prácticas](https://proyectosagiles.org/fundamentos-de-scrum) para **trabajar colaborativamente, en equipo**, y obtener [el mejor resultado posible](https://proyectosagiles.org/beneficios-de-scrum) de un proyecto

1. ¿En qué consiste el marco de trabajo de Scrum? Mencione las 4 partes.

**Está compuesta por los equipos de trabajo, los eventos (son las reuniones), los artefactos y las reglas**

1. ¿Es SCRUM una metodología de gestión de proyectos o reemplazar a las mismas?

Scrum es una técnica de trabajo, esta no es una metodología de gestión de proyectos ni tampoco las reemplaza

1. ¿Cuáles son las 4 tareas de cada iteración?

**Cada iteración está compuesta por la planeación, diseño, desarrollo y pruebas**

1. Mencione y detalle los 3 roles de Scrum.

Product Owner: Es el que gestiona la lista del producto, se asegura de la lista muestre claramente lo que el equipo debe hacer y entiende los elementos de la lista

Scrum Master: Es un facilitador de los eventos de Scrum y además es el encargado de que Scrum sea entendido y adoptado.

Equipo de Desarrollo: Son los únicos encargados del desarrollo del producto y llevan a cabo los incrementos y el tamaño óptimo es de 4 a 7 personas

1. ¿Qué es un Sprint? ¿Cuánto dura? ¿Se inicia el sgte de inmediato? ¿Se puede considerar como un proyecto?

**Un Sprint es una iteración que va de 2 a 4 semanas, al finalizar una iteración se inicia de inmediato otra y se puede considerar como un proyecto**

1. Describa las 4 distintas reuniones de Sprint y sus características.

**Reunión de planificación:**

Es donde se define el trabajo a realizar durante el sprint, qué se puede entregar, cómo se conseguirá hacer el trabajo necesario, tiene un máximo 8 horas para un sprint de un mes, el Scrum Master asegura que la reunión se lleve a cabo durante el tiempo asignado

**Scrum Diario**

Reunión diaria de 15 minutos donde se sincronizan las tareas y se planea lo que se realizará durante el día, el Scrum Master asegura que la reunión se lleve a cabo durante el tiempo asignado.

-Lo que hizo – Lo que haré, impedimentos.

**Revisión de Sprint**

Reunión informal para revisar el resultado del sprint con una duración de 4 horas para un sprint de un mes, se discute qué hacer a continuación y se revisa que cambios pueden afectar la lista del producto.

**Retrospectiva**

Se realiza luego de la revisión y antes de la reunión de planificación para un máximo de 3 horas para un sprint de un mes.

1. Describa una Tarjeta de Product Backlog.

Es la tarjeta donde se escribe la historia del usuario que son los requerimientos del producto la cual se le agrega un puntaje que representa el esfuerzo. Se redacta Como el usuario, quiere, con el fin de.

**INVEST**

1. ¿Qué es INVEST? Describa cada letra.

Es una técnica para el análisis de las historias

Independiente: Cada historia es completamente independiente

Negociable: Se puede reemplazar por otra de diferente prioridad

Valor: Le agrega valor significativo al producto

Estimable: El equipo se siente seguro al estimar los tiempos

Small (Pequeña): Hace algo simple y funcional, se puede hacer en 2 semanas

Testeable: Se le pueden hacer pruebas

1. Si una historia de usuario no cumple alguno de los puntos de INVEST, ¿qué debería hacerse?

Se debe replantear la historia

1. ¿Qué es Disponibilidad del equipo?

Es la suma de horas dedicadas por cada miembro del equipo al proyecto

77. ¿Qué es Velocidad del equipo?

Es el promedio en que desarrollan con éxito una operación

1. Cuatro diferencias entre desarrollo incremental y desarrollo iterativo son:

**Incremental**

En los incrementos se tiene una idea completa del producto final.

Al comenzar hay certeza absoluta sobre el resultado final deseado.

**Iterativo**

En las iteraciones se va construyendo un borrador, se valida, y luego se sigue agregando calidad al producto.

Al comenzar no hay certeza absoluta sobre el resultado deseado, sino que se va construyendo a medida que se avanza y se va viendo el producto

**TDD y Refactoring**

1. Mencione los 2 niveles de TDD.

**ATDD: es el que se escribe solo una prueba de aceptación y luego el código para esta**

**TDD: se escribe una prueba unitaria y luego el código**

1. De 3 características de ATDD y 2 para TDD.

**ATDD:**

Se escribe una sola prueba de aceptación

Describe requerimientos ejecutables detallados

**TDD:**

Al inicio las pruebas fallan

Luego se implementa el código y pasan a color verde

Luego se refactoriza y se inicia el ciclo

1. Cuatro aspectos a considerar sobre TDD son:

Se debe emplear desde el inicio de un proyecto

Se deben tener bien definidos los requisitos de la funcionalidad a realizar

Los criterios de aceptación deben estar bien definidos contemplando casos de éxito como de error

Se deben aplicar los principios de SOLID

1. Dé cinco características sobre refactorización.

Modificar código: consiste en eliminar redundancias, malas prácticas y aplicar todo cambio para tener un código mantenible.

Iterativo: Se requiere refactorizar varias veces.

Pruebas Unitarias: las mismas deben existir antes de refactorizar para garantizar el nuevo cambio no afectará la funcionalidad.

SOLID: Se deben emplear los principos de SOLID

Funcionalidad: No afectar la funcionalidad en la refactorización

**CASOS DE USO**

1. Mencione 3 características de Casos de Uso.

Describen el flujo de un proceso entre la interacción de usuario con el sistema

El actor inicia la interacción con el sistema.

En el caso de uso se puede escribir los diferentes escenarios según las acciones del actor, además todo caso de uso debe tener un escenario principal de éxito.

Los casos de uso definen lo que el usuario debe hacer.

**SQL**

-Entidad relación

84. ¿Para qué se usa el modelo ER?

Se usa para el diseño conceptual de un sistema

1. Explique sus 3 partes fundamentales.

Entidades: son las que representan elementos físicos o abstractos. Se representan en una caja rectangular.

Relaciones: Enlazan las entidades, tienen un nombre y una cardinalidad. Se representan con un rombo.

Atributos: Son las propiedades de una entidad. Se representan con óvalos.

1. ¿Qué es la cardinalidad?

La cardinalidad representa el número de entidades que se pueden relacionar con otras entidades.

87. Dé ejemplos de 3 formas de representar la cardinalidad.

1. Cuatro maneras de optimización de consultas SQL son:

Usar Exists en lugar de Count() para validar si hay registros

No utilizar Not Exists

Usar joins en lugar de selects anidador

Especificar siempre los nombre de las columnas en lugar de Select \*

1. ¿Qué es un Plan de Ejecución. Dé dos características.

Es el resultado de la forma que el optimizador de consultas planea ejecutar el query solicitado

Muestra el orden de las acciones, los costos y las operaciones intermedias

1. ¿Qué debe revisarse al Analizar el plan de ejecución?

Validar y analizar los table Scan, si hay más tablas asociadas

1. ¿Que son joins? Explique con ejemplos los 3 tipos.

Son instrucciones que permiten combinar registros entre varias tablas especificando los criterios que la asocian.

Inner Join: Realiza una intersección entre 2 tablas mientras cumplan la condición especificada

Cross Join: Realiza un producto cartesiano entre 2 tablas n\*m

Outer Join: Trae los registros entre una tabla y otra donde se especifica cuál de las 2 debe cumplir la condición mientras que para el otro caso si no cumple la condición trae los registros como null

-Transacciones en SQL Server

91. ¿Qué es una transacción?

Una unidad única de trabajo.

1. ¿A qué se refiere el termino Commit?

Confirma los cambios realizados en una transacción

1. ¿A qué se refiere el termino Rollback?

Devuelve los cambios realizados hasta el inicio de donde se declaró la transacción

1. Detalle cuatro términos de transacciones

Begin Transaction: Indica el inicio de una transacción

Commit Transaction: Confirma los cambios de una transacción

Rollback Transaction: Deshase los cambios realizados dentro de una transacción

Set Xact\_AbortOn: Hace rollback automático de la transacción si ocurre algún error.

-ACID

95. Defina qué significa cada letra en ACID

Atomic: Se ejecutan todas las acciones o ninguna

Consistency: Pasa de un estado válido a otro estado valido esto quiere decir que en caso de error se realiza un rollback a todo lo aplicado, de lo contrario todo se mantiene

Isolation (Aislamiento): Una transacción no afecta a otras transacciones concurrentes

Durability: Hacer que persistan los cambios solo en caso de que no hayan errores

1. ¿Qué controlan los niveles de aislamiento?

El comportamiento del bloqueo de los datos usado en las transacciones.

1. ¿Qué se puede hacer para asegurar la integridad de las operaciones?

Bloquear los registros leídos o recién modificados durante el tiempo que se ejecuta la transacción.

1. ¿Qué puede causar un bloqueo excesivo de datos?

Que el sistema sea sumamente lento o que se generen deadlocks

1. ¿Qué es un deadlock? Dé dos maneras de evitarlos.

Es cuando 2 o más procesos quedan detenidos esperando el uno por el otro.

Evitar mantener las transacciones abiertas durante mucho tiempo

1. ¿Qué son blocklocks? ¿Qué relación tienen con los Bloqueos de datos?

Cuando 2 o más procesos quedan esperando el uno por el otro a que terminen pero MicrosoftmSQL no detiene ningún proceso solo elige cual debe terminar primero

1. Explique los dos tipos de Bloqueo de datos.

Bloqueo compartido: Es el que ocurre cuando se ejecuta un select

Bloqueo exclusivo: Es el requerido para cualquier instrucción de modificación, un update, insert o delete

1. Explique los cinco niveles de aislamiento.

Read UnCommited: Se pueden leer datos que aún no han sido confirmados.

Read Commited: No se pueden leer datos que aún no se han confirmado.

Repeatable: No se pueden leer datos que no se han confirmado y además bloque los datos leídos para que estos no puedan se modificados.

Snapshot: Genera una instantánea de los datos. No hace bloqueos

Serializable: Bloquea los datos que lee y además los rangos entre los datos para que no puedan ser modificados.

1. ¿Qué se puede hacer con un GroupBy?

Se pueden agrupar datos

Se puede utilizar funciones de agregación con los datos agrupados

Se pueden ordenar los datos

Se pueden filtrar los datos mediante las funciones de agregación con la función having

1. ¿Cuáles son cinco aspectos a considerar respecto a GroupBy?

La cláusula Where se ejecuta antes que el agrupamiento

Las funciones de agregación se ejecutan luego de que se ha realizado el agrupamiento de los datos

La cláusula having se aplica luego del agrupamiento por lo que se pueden ejecutar condiciones para filtrar datos con funciones de agregación desde el having

Se declara de la siguiente forma Select, From, Where, Group By, Having

Se ejecuta de la siguiente forma From, Where, Group By, Having, Select

1. ¿Qué es un índice?

Es una estructura de datos que hace referencia a una columna de una tabla permitiendo un rápido acceso a los registros.

1. Explique los 2 tipos de Índice

Clustered: Son los que realizan una agrupación de los datos y un ordenamiento por la columna que se definió. Solo puede existir un índice clúster por tabla.

No Clusteres: No se encuentran agrupados sino que se crea una estructura separada que apunta a las filas de los datos

**VISUAL BASIC**

-Expresiones lambda

107. Defina expresiones lambda

Son funciones sin nombre que son insertadas en la sección de código donde se llaman

1. ¿En lugar de qué pueden usarse? ¿Por qué?\*\*

Se pueden usar en lugar d un delegado

1. ¿Qué ahorra usar expresiones lambda? ¿Qué se evita con ellas?

No se tiene que escribir una función con nombre. Utilizada para cuando una función se va a usar solo en esa parte del código y no requiere reutilizarce

1. ¿Cuáles son las dos maneras de declararlas y por qué? ¿Qué se puede omitir y bajo qué condiciones?

Se declaran Function cuando retornan un valor y Sub cuando no debe retornar. Si se declara en una sola línea se puede omitir el End al final. Si son inline tampoco se usa el return

1. ¿Qué parámetros no acepta?

No acepta parámetros Optional o Paramarray

1. ¿Cuándo un delegado hace match con cualquier método?

Cuando reciban la misma cantidad de parámetros, orden y tipo y además retornen el mismo tipo de Dato

1. ¿Qué debe coincidir entre una función lambda y un delegado? ¿Qué no es necesario que coincida y porque?

Misma cantidad de argumentos y mismo tipo de retorno

-Declaración inline de delegados

113. ¿Cómo se declaran cuando retorna un valor? ¿Cómo cuando no lo hace? Escriba un ejemplo para cada caso. ¿Qué sucede con los nombres y tipos de argumentos?

Cuando retorna un valor se declara con la palabra Func y cuando no retorna se declara con la palabra Action.

Si es Func el último argumento es el tipo de retorno

En los delegados no se declara el nombre de los argumentos, solo su tipo

Ejemplo

Action(Of String)

Func(Of Integer, Boolean)

1. ¿Acepta parámetros genéricos?

Los delegados aceptan parámetros genéricos

-Delegados en Colecciones de datos

115. Explique cuatro ejemplos de uso de delegados en colecciones de datos.

Comparison(Of T): Para comparar elemento

Predicate(Of T): Verificar que se cumplan ciertas condiciones

Converter(Of TInput, TOutput): Convierte un objeto de un tipo a otro

Progress(Of T)

1. Dé la relación entre funciones lambda y entity framework.

Se utilizan para realizar consultas con la cláusula Where

-Lambda en MVC

117. ¿Cuál es la relación entre lambda y MVC? Dé dos ejemplos.

Se utiliza en métodos helpers de Razor LabelFor y TextboxFor.

-Programación Concurrente en VB.NET

118. Defina TPL. ¿Qué hace? ¿En cuáles APIs están sus ensamblados?

Sirve para ejecutar proceso en paralelo, multi-hilo y asíncronos

System.Threading System.Threading.Task

119. Describa cuatro características de TPL.

Usa procesadores de forma eficiente

Usa un pool de hilos

Tiene soporte para cancelar procesos

Maneja el particionamiento de trabajo

120. ¿Qué es Paralelismo de Datos?

Ejecución de acciones concurrentes sobre colecciones de datos

121. ¿Qué clase y métodos usa?

Clase System.Threading.Task.Parallel

Y los métodos Parallel.For y Parallel.ForEach

122. ¿Qué hace TPL al invocar estos métodos?

Particiona los datos permitiendo que los métodos puedan trabajar sobre ellos de manera concurrente

123. ¿Qué hace el calendarizador de tareas?

Distribuir el trabajo en hilos y controlar las cargas

124. Describa los métodos For, ForEach e Invoke para la clase Parallel, con sus semejanzas y diferencias.

Parallel.For: Misma función que el For solamente que recibe un índice inicial y un índice final y la acción a ejecutar

Parallel.ForEach: Realizar tareas en paralelo. Recibe la estructura de datos y luego la acción

Parallel.Invoke: Recibe una o más funciones a ejecutar en paralelo

125. Defina Task. ¿Cuáles son las dos formas de declararlo? Explique las tres maneras de iniciar tareas asíncronas con la clase Task.

Task representan tareas que se ejecutan de forma asíncrona.

Uno se declara como Task que no retorna ningún valor y el otro es Task(Of T) que retorna un valor

Task.Run: Crea, ejecuta y retorna un task

Task.Factory.StartNew: Retorna un task que puede calendarizarce

Task.Start: Inicia la ejecución de un task previamente creado

126. ¿Cuáles son las dos clases que son la base de la Programación Asíncrona?

Async y Await

127. ¿Qué pasa con la UI y el flujo de la aplicación cuando se utilizan métodos asíncronos?

No se bloquea el UI mientras se ejecuta la tarea y el flujo tampoco se interrumpo

128. ¿Con qué palabra se definen los métodos asíncronos?

Async

129. ¿Qué sucede al usar Await dentro de un método Asíncrono? ¿Qué pasa si no existe ningún Await en un método Asíncrono?

Hace que se suspenda la ejecución del método hasta que la operación finalice

* Dentro de un método Asíncrono, ¿se puede llamar un método NO Asíncrono?   
  Si
* Dentro de un método Asíncrono, ¿se puede llamar otro método Asíncrono?   
  Si
* Dentro de un método Asíncrono, ¿cuál es la diferencia entre llamar un metido asíncrono y otro que no lo es? \*\*  
  Si hay un método que no es asíncrono dentro de uno Asíncrono, el asíncrono espera a la ejecución del otro método para continuar  
    
    
   https://blog.stephencleary.com/2012/02/async-and-await.html

**130. ¿Qué es Task y Task(Of T)? ¿En qué se diferencian?**

Representan tareas que se ejecutan de forma asíncrona.

El task es void y task(of T) tiene retorno de un objeto de tipo T.

**131. ¿Se puede usar Await dentro de un método que no está definido como Asíncrono?\*\***

No, el await únicamente se puede utilizar dentro de métodos asíncronos y hace que se suspenda la ejecución del método async, hasta que la operación finalice.

132. Dentro de un método Asíncrono, al llamar otro método (ya sea Asíncrono o no), cual es la diferencia de llamarlo usando Task vs Await? De un ejemplo para cada caso.\*\* (Para responder esta pregunta se puede correr el código en la presentación de BigBang "Introducción a la Programación Concurrente en VB", y llamar al método LongMethodAsync() de las dos maneras:

Dim myTask As Task(Of DateTime) = LongMethodAsync()

Dim myTask = Await LongMethodAsync()

)

Que con task(of) no espera lo que dure el método asíncrono que se esté llamando.

Con await espera lo que dure el método asíncrono que se esté llamando, por ejemplo si hay un sleep de 4 segundos esté esperará esos 4 segundo, de la otra manera no, es inmediato.

-Manejo de excepciones

133. ¿Cómo se manejan las excepciones en métodos invocados con Await?

Se manejan con try, catch. Si en un task ocurre un error await va devolver ese error, pero debe ir dentro del bloque try.

134. ¿Hay alguna diferencia en la manera de manejar excepciones entre métodos Síncronos y Asíncronos?

Siempre y cuando se utilice con await van a funcionar de la misma manera.

135. ¿Puede ir Await dentro de un bloque Catch o Finally en VB.NET?

136. Mencione y explique tres malas prácticas y errores comunes.

* No esperar el valor de retorno de los métodos asíncronos.
* Tener muchos hilos en paralelo en procesos pequeños.
* Paralelizar en desorden el parallel.for es más eficiente cuando se ejecuta de manera incremental.

**INYECCION DE DEPENDENCIAS**

137. ¿Qué es (IoC Inversion de control)? Explique. ¿Quién crea los objetos consumidos?

Es cuando se delega el control o flujo de trabajo a otro componente.

Los objetos consumidores no crean los objetos consumidos, estos son creados por un contenedor.

**138. ¿Qué es Inyección de Dependencias?**

Una clase encargada de inyectar las dependencias de otra. Es una forma de IoC

139. ¿Para qué se utiliza? ¿Cómo se logra en .NET?

Se implementa por medio de interfaces para desacoplar componentes.

140. ¿Cuál es la relación entre IoC y DI?

Cuando se utiliza inyección de dependencias se puede utilizar también un contenedor de IoC para que se encargue de inicializar las dependencias de las clases. De esta manera la clase consumidora no necesita saber ni inicializar las dependencias de la clase consumida, puesto que esto se le delega al contenedor de IoC

141. ¿Qué debe tener una clase para ser inyectable? ¿Cuál puede ser un problema? ¿Como se resuelve?

Debe tener la implementación de una interfaz.

Que donde se vaya a inyectar no implemente dicha interfaz.

Personalizando el constructor que por default es vacío.

142. ¿Qué es un Contenedor de IoC? ¿Qué administra? ¿En qué principios se basa?

Componente especializado en la creación de instancias de objetos. En los principios ID IoC

143. ¿Qué inicializa un IoC? ¿Qué gana la clase consumidora utilizando un Contenedor de IoC?

Un contenedor Ioc si se utiliza con Unity, se configura en el archivo UnityConfig ubicado en la carpeta App\_start del proyecto.

No necesita saber ni inicializar las dependencias de la clase consumida, puesto que esto se le delega al contenedor de IoC.

**UNITY**

144. ¿Qué es Unity?

Es un contenedor Ioc creado por Microsoft.

145. ¿Cuáles son las dos maneras de especificar las dependencias en Unity?

* En tiempo de ejecución
* En el archivo de configuración

146. ¿Cómo se agrega Unity a MVC? ¿En qué archivo se configura? ¿En qué ruta queda dicho archivo?

Se agrega desde el Nudget.

Se configura en UnityConfig

Dentro de la carpeta App\_start.

147. ¿Qué acción adicional debe hacerse al proyecto de pruebas al hacer pruebas unitarias usando Unity?

Instalar Unity\_MVC en el proyecto de pruebas.

148. ¿Qué se puede lograr con configuraciones avanzadas de Unity?

Que cada instancia se resuelva como singleton.

149. ¿Qué permite Unity para crear instancias que requieran de argumentos adicionales en su constructor?

Crear métodos explícitos, para crear instancias que requieran argumentos adicionales en su constructor.

150. ¿Qué puede hacer el repositorio Mock? Detalle dos opciones.

-Conectarnos a una base de datos de pruebas usando entity, ado .net u algún otro medio.

-Podemos cambiar la instancia del repositorio en tiempo de ejecución porque la configuración se realiza desde .configs

-Unity se puede usar en cualquier parte del código para obtener una instancia de la interfaz mediante el método resolve, de esta manera si no se quieren inicializar los componentes en el constructor, podemos usar el método resolve en cada método que utilicemos las dependencias.

151. ¿Cuál es la ventaja de tener un archivo de configuración? Mencione dos ventajas.

-No se necesita volver a recompilar la aplicación en caso de algún cambio

-Fácil de modificar

152. ¿Para qué sirve el método Resolve de la clase UniyContainer?

Sirve para obtener una instancia de la interfaz desde cualquier parte del código.

153. ¿Qué se hace si no se quiere inyectar dependencias en el constructor?

Usar el método resolve en cada método que utilicemos las dependencias.

**PILA**

154. ¿Cuándo se usa una Pila? ¿Qué significa LIFO? Dé un ejemplo práctico en el que se puede usar una pila.

- Cuando se quiere guardar una secuencia pero en su orden inverso

- Primero que entre el último que sale

-El recorrido de un árbol en profundidad

**COLA**

155. ¿Qué es una cola? ¿Cómo funciona? ¿Qué significa FIFO? De un ejemplo práctico de una cola

- Es una estructura que permite guardar elementos

- El primero que entra es el primero que sale

- La fila de un banco se puede manejar con una cola

**MICROSERVICIOS**

156. ¿Qué son Microservicios?

Es una arquitectura en qué la creación de sistemas se orienta a la creación de servicios pequeños.

157. ¿Qué relación tienen los distintos servicios en este estilo arquitectural?

En que no se relacionan unos con otros, son independientes.

158. Mencione cinco características de Microservicios.

-Bajo acomplamiento

-Endpoints inteligentes.

-Diseño tolerante al fallo

-Productos en lugar de proyectos

-Componentes vía servicios

159. ¿Qué es común al trabajar con Microservicios? ¿Es obligatorio?

Trabajar estrictamente usar apis basadas en Rest. No es obligatorio, pero facilita la comunicación.

**VERBOS DE HTTP**

160. Describa los cinco verbos existentes.

Delete: Eliminar

Get: Consultar

Put: Actualizar o crear recursos si el id no existe. Se envían todos los campos.

Post: Crear nuevos recursos

Patch Actualizar los campos enviados

161. Liste y describa ocho códigos de respuesta estandarizados.

200 ok

201 Created

204 No content

400 Solicitud invalida

404 Recurso no encontrado

405 Método no permitido

409 Conflicto

500 Error interno

**SERIALIZACION**

162. Defina Serialización.

Convertir un objeto a bits

**DEBUG REMOTO**

163. Describa los cinco pasos necesarios para hacer un debug remoto.

- instalar un componente adicional en visual studio

-Configurar el firewall

-Activar modo debug tanto en visual como el servidor remoto

- Configurar visual studio para conectarnos al servidor.

**BLOQUE USING**

164. ¿Para qué se utiliza el bloque Using?

Se utiliza para manejar recursos no manejados.

165. ¿Cuándo se usa Using en que momento se liberan los recursos?

Se fuerza de manera inmediata

166. ¿Qué interfaz es necesaria en quien consume los recursos? ¿Qué garantiza?

IDisposable, que tenga una implementación del método Dispose

167. Ante un error, ¿cómo se comporta el bloque Using? ¿Cómo se obtendría el mismo comportamiento usando Try y Catch?

No lo notifica.

Si lo que sea desea es capturar el error del using no se puede hacer sin un try catch.

168. ¿Cómo afecta el scope de Using a las variables declaradas en él?

Una vez finaliza el using desaparecen.

169. ¿Using debe tener poco código?

Sí, una vez finalizado el bloque using desaparece.

**GAC GLOBAL ASSEMBLY CACHE**

170. ¿Cuáles son 4 características del GAC?

-Es un cache de DLLS que se utiliza por la aplicación de .net.

- Las DDLS son compartidas.

-Puede tener muchas copias de dll pero con diferente versión.

171. Dos puntos respecto a Registrar DLLs en el GAC.

Gautil –i ruta

172. Tres puntos respecto a eliminar DLLs del GAC.

Gautil –u nombreDll

173. Dos puntos para ver contenido del GAC.

Gautil -l

174. ¿Qué debe hacerse para eliminar el caché de ASP.NET?

Reinciar el servidor IIS

**PRUEBAS DE ESTRES**

175. Dos Características de funciones testeables son:

- Una única responsabilidad

-Retornan algo.

176. ¿Cuáles son los dos tipos de prueba que existen? Descríbalas y explique porqué son diferentes entre ellas.

Prueba de carga: Prueba en la que se espera una determinada carga, con el de determinar tiempos de respuesta, tamaño de la respuesta.

Pruebas de estrés Lo que se prueba es que el sistema responda a cargas impensables en un determinado momento en escenarios concurrentes, para encontrar puntos de falla.

177. Dos maneras en que Load Test hace ambas pruebas:

Se puede hacer una prueba unitaria que representa la prueba de carga y un LoadTest con un usuario y una única iteración

O Se puede hacer una prueba unitaria que realice una carga y un LoadTest con muchos usuarios concurrentes realizando varias solicitudes con esa carga

178. ¿Qué es sample rate? ¿Qué valores se usan para pruebas cortas y que valores para pruebas largas? ¿Por qué?

Sample rate determina que tan seguido se recolectan los contadores de rendimiento.

Para pruebas cortas se utilizan valores pequeños, que indica que los contadores se deben recolectar más a menudo. Para pruebas largas utilizan valores más grandes, sino la base de datos crecería mucho.

179. Dos maneras de configurar una prueba son:

1. Por tiempo de duración

2. Cantidad de veces (iteraciones)

180. ¿Qué es el tiempo de calentamiento y para qué se usa?

El tiempo de calentamiento es un periodo en que no se recolectan los contadores, ya que pueden sesgar la prueba.

181. ¿Para qué se usa 'el perfil de tiempo para pensar'?

Para determinar los tiempos de espera o de reacción que utiliza un usuario antes de seleccionar una opción

182. ¿Qué es una carga constante y qué una carga en pasos?

Una carga constante es que la cantidad de usuarios es continua desde el incio, mientras que una carga a pasos es que poco a poco se van sumando más usuarios.

183. ¿Cuántas pruebas unitarias o web test pueden agregarse a una prueba de carga?

Una o varias pruebas unitarias.

184. ¿Cuál es la diferencia entre UNIT TESTs y Web Tests?

185. ¿Qué es un Text Mix Model? ¿Cuáles son las cuatro maneras de realizarlo? Explíquelas

Probabilidad de que un usuario virtual ejecute una prueba determinada en un escenario de prueba de carga.

1. Basado en el total de pruebas: Cada vez que un usuario se ejecuta una prueba es una iteración.

2. Basado en el número de usuarios: Hasta que todos los usuarios ejecuten la prueba se toma como una iteración.

3 Basado en ritmo de usuario. Cada usuario ejecuta n cantidad de veces una prueba.

4 Basado en orden secuencial: Un usuario ejecuta una prueba en el orden que fueron definidas y hasta que se agote el tiempo definido vuelve a ejecutar la prueba.

186. ¿Qué es Network Mix Model? Dé dos características

Podemos indicar como simular la conexión a internet y se puede usar un solo modelo o distribuir varios modelos según el escenario que se desea probar.

187. ¿Qué permiten los 'counter sets'? ¿Cuáles son los tres contadores a recolectar?

- Permiten simular la máquina que deseamos probar.

Los contadores a recolectar son:

1. ASP.NET
2. IIS
3. .NET

**EXCEPCIONES**

188. ¿Qué es una excepción?

Es cualquier condición de error.

189. ¿Qué hay en .NET para cada aplicación acerca de las excepciones?

Hay una tabla de manejo de información de excepciones.

190. ¿Cuál es la base de todas las expeciones? ¿De quién hereda esta base?

Exception y hereda de SystemException hereda de Exception

191. Mencione y explique las tres propiedades de la base de todas las excepciones.

1 Message. Contiene le mensaje de la excepción

2. StackTrace Contiene información de la excepción como: archivo, clase método donde ocurrió el error.

3. innerException Sirve para anidar excepciones. Por ejemplo cuando se envía una personalizada y se quiere mantener la original.

192. ¿Qué es 'Data'?

Diccionario que contiene datos adicionales de la excepción.

193. Explique las tres partes de una excepción

194. Mencione y explique las tres maneras de hacer un reenvío de excepciones.

Throw. Loguea una excepción, mantiene le stack trace original

Throw ex Loguea una excepción, sobreescribe el stack trace por el nuevo, es decir oculta la línea en la que ocurrió el error.

Throw new Reemplaza por completo la excepción original

195. Menciones seis Errores comunes al usar el manejo de excepciones.

1. Querer capturas todas las excepciones

2. Poner la excepción más general de primera.

3. Revelar demasiada información.

4. Poner un try catch en cada método tiene un costo en rendimiento

5. Dejar el catch vació

6. Anidar un try catch en un ciclo

**POSTBACK**

196. ¿Qué es Postback? Explíquelo con cuatro caracteristicas.

Valor que indica si la página se carga por primera vez o en respuesta de una a una devolución de datos.

Se valia con IF NOT isPostBack

**RAZOR**

197. ¿Qué es Razor? Explíquelo con cuatro caracteristicas.

Tipo de vista MVC, en la que se puede escribir código del servidor en el cliente.

**HERENCIA**

198. ¿Cómo se declara una clase que hereda de otra?

Public Class alumno

Inherits Persona

End Class

199. ¿Qué debe hacerse si se quiere evitar que una clase sea heredable?

Public NotInheritable Class Persona

End Class

200. ¿Una clase que hereda a otra está obligada a sobrescribir todos sus métodos?

No, uno hereda de una clase para extender su funcionalidad, no para modificar la existente.

201. ¿Una clase que implementa varias interfaces está obligada a implementar todos los métodos de la interfaz?

Sí, es necesario implementar todos sus métodos.

202. ¿Una clase abstracta se puede instanciar?

No, únicamente se puede heredar.

203. ¿Para qué se utiliza ForEach, Next? ¿Puede usarse en colecciones que NO implementen la interfaz IEnumerable?\*\*

Para recorrer colecciones que implementan IEnumerable.

204. ¿Qué operador se usa para concatenar strings?

Se concatena con &

205. Si se concatena un string y un número, ¿qué sucede si usamos '+' y que sucede si usamos '&'? Dé un ejemplo para cada caso.\*\*

1+”2” = 3

1+”H”= ERROR

1 & “a”= “1a”

1 & “2”=”12”

206. ¿Qué pasa con el objeto original de un string cuando se modifica?

Se destruye y se reemplaza por uno nuevo.

207. ¿Cuándo se usa String y cuando StringBuilder?

StringBuilder cuando un string se modifica mucho o se hacen muchas concatenaciones.

208. Explique ocho métodos utilitarios del objeto string.

1. Trim quita espacios al inicio y el final

2. ToUpper

3. ToLower

4 Replace

5 Split

Join Uno los valores de un array en un string

IndexAt Devuelve el índice del valor específicado.

**CLASE GENERICA**

209. Defina Clase Genérica. ¿Cómo se declara? ¿Por qué es fuertemente tipada?

Clase que usa el polimorfismo para permitir la instanciación con varios tipos de datos.

(OF T) donde t es el tipo que le quiere dar y no puede cambiar.

210. ¿Por qué usar Clase Genérica en vez de Object?

Object no específica realmente el tipo de dato concreto, mientras que con la clase genérica sí.

**WCF**

211. ¿Qué significan las letras de WCF?

Windows Communication Foundation

212. ¿Qué es WCF?

Es un framework para crear aplicaciones orientadas a servicios

213. Mencione y explique tres patrones de mensajería que se soportan en WCF.

1. Request/Reply Se envía un único mensaje a la vez y se espera su respuesta.

2. Duplex Ambos lados envían y reciben n cantidad de mensajes mensajes como una llamada telefónica donde cada palabra es un mensaje.

3. One-way. El remitente envía un mensaje, el receptor no envía un mensaje de respuesta.

214. Menciones dos aspectos de seguridad existentes en WCF.

1. Encripta el canal de comunicación

2. Encripta el mensaje

215. ¿Qué tipo de transacciones soporta? Explique cada una \*\*

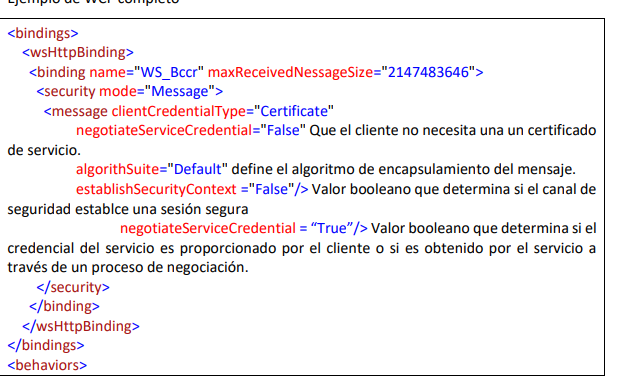
1. Atómicas Es un bloque de código que se ejecuta completamente o no se ejecuta.

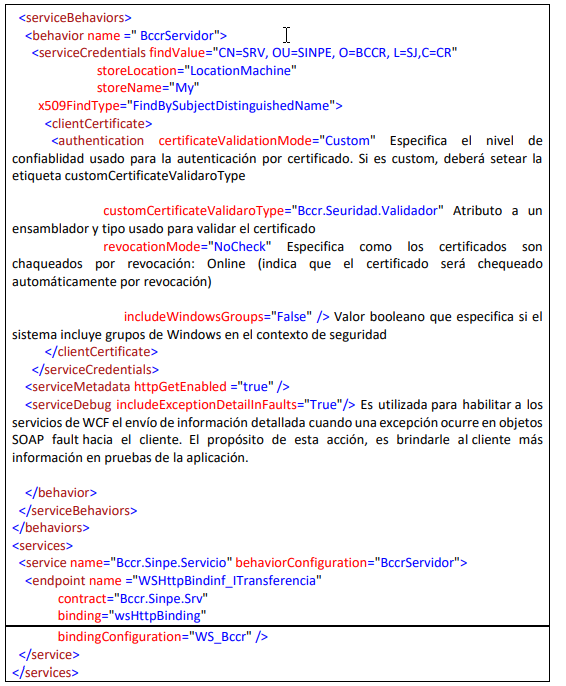
2. Distribuidas Es una transacción que confirma los datos en dos o más servidores de BD

216. ¿WCF soporta Ajax y REST?

Sí

217. Escriba un ejemplo de archivo de configuración de WCF completo, con todas sus partes.





218. ¿Qué muestra los mensajes de error detallados al cliente para conocer y depurar el error? ¿En qué tipo de ambientes se usa y porque?

<serviceDebug includeExceptionDetailInFaults="false"/>

Se usa solo en ambiente de desarrollo, porque en ambiente productivo no nos interesa que el cliente conozca información detallada de la aplicación.

219. Si un archivo de configuración escucha Critical y se lanza un Error, ¿qué sucede? ¿Cómo se resuelve?

No quedara registrado, porque critical no contiene los errores manejados, solo los errores no manejados. Cambiando en el archivo de configuración el switchValue por error.

220. ¿Por medio de qué se da la comunicación entre clientes y servicios?

Por medio de mensajes.

221. Defina EndPoint. Describa sus tres partes.

Un endpoint es un punto donde llegan y se envían los mensajes y se define que se requiere para intercambiar mensajes. Un servicio expone uno o más endpoint y el cliente debe generar uno compatible para comunciarse.

Address: url del servicio

Binding: El tipo de binding que utiliza

Contract define la interfaz que está implementando

Un endpoint necesita el ABC para ser completo.

222. Describa dos componentes adicionales de WCF.

Behaviors, define comportamientos adicionales del endpoint.

223. Explique en detalle cuatro tipos de Binding.

BasicHttpBinding. Por defecto tienen la seguridad apagada. Utiliza HTTP Y SOAP para comunicación con servicios asmx y WS-I BASIC PROFILE 1.1

WSHttpBinding Seguro, interoperable, soporta transacciones. Se comunica con servicios WS\*

WSDualHttpBinding. Seguro, interoperable permite comunicación dúplex.

NetTcpBinding. Seguro, especial para uso de intranet.

224. Dos posibles protocolos de transporte de WCF son:

1. HTTP

2 TCP

225. ¿Tres tipos de Encoding son? Explique cada uno

Texto El más interoperable pero el menos eficiente para los mensajes XML.

MTOM Es eficiente para la transmisión de datos binarios, MTOM intenta tener un equilibrio entre eficiencia e interoperabilidad.

Binario Es más eficaz de las opciones de codificación pero la menos interoperable porque solo es compatible con puntos de comunicación WCF.

226. ¿Qué permite configurar los Binding? 4 partes

1 El tamaño del mensaje

2 Los tiempos de espera

3 La seguridad

4 Formato del mensaje

227. Explique los tipos de Manejo de instancia en WCF

1. InstanceContextPer.Call : Una instancia por llamada, se utiliza cuando no es prioridad mantener el estado entre los mensajes.

2. InstanceContextPer.Session Una instancia del servicio por sesión para mantener el estado de los mensajes.

3. InstanceContextPer.Single Una instancia global, se utiliza cuando se necesita compartir información de manera global.

228. Explique los tipos de Manejo de concurrencia

ConcurrencyContextMode.Single Una instancia puede manejar un único hilo.

ConcurrencyContextMode.Multiple Una instancia puede manejar multiples hilos en paralelo.

229. Explique los tipos de sesión en el IIS

1. Modelo InProc : Almacena el estado de sesión en memoria del servidor web. Esta es la opción por defecto.

2. Modo StateServer Almacena el estado de sesión en “Servicio de estado ASP.NET”. Garantiza que el estado de sesión se mantenga si se reinicia el servidor y que esté disponible para varios servidores web (granja)

3. Modo SqlServer Almacena el estado en una bd SQL, garantiza que el estado de sesión se mantenga si se reinicia y esté disponible para varios servidores web ( granja)

230. ¿Es lo mismo los tipos de sesión en el IIS que tipos de sesión en ASP.NET? ¿Por qué? Explique su respuesta \*\*

231. ¿En qué archivo se configura el Behavior? ¿Para limitar qué se puede configurar un Behavior?

En el archivo de configuración, se configura información adicional. Como por ejemplo: numero concurrente de sesiones, llamadas e instancias.

En el webconfig en la sección de Behavior para limitar el acceso al servicio

<system.serviceModel>

<behaviors>

<serviceBehaviors>

<behavior name="myBehaviorTony">

<!-- Habilita metadata para que el cliente genere su proxy -->

<serviceMetadata httpGetEnabled="true" httpsGetEnabled="true"/>

<!-- Muestra informacion de los mensajes de error al cliente -->

<serviceDebug includeExceptionDetailInFaults="false"/>

<serviceThrottling maxConcurrentCalls="5" maxConcurrentInstances="5" maxConcurrentSessions="5"/>

</behavior>

</serviceBehaviors>

</behaviors>

</system.serviceModel>

232. ¿Qué sección no debe olvidarse al declarar un behavior? ¿Por qué? \*\*

La de servicesMetada para que el clienta pueda generar el proxy para conectarse al servicio.

233. ¿Dentro de qué sección va el Bindings y Behaviors en el web.config

<system.serviceModel> </system.serviceModel>

234. ¿Qué sección debe agregarse dentro de system.serviceModel para configurar el endpoint en web.config?(

<services>

<service name="WCFConfiguracion.Service1" behaviorConfiguration="myBehaviorTony" >

<endpoint address="/Service1.svc" binding="wsHttpBinding" bindingConfiguration="mybindingTony"

contract="WCFConfiguracion.IService1">

</endpoint>

</service>

</services>

235. ¿En que sección va el Behavior y en cual el Binding?

<bindings></bindings>

<behaviors></behaviors>

236. ¿Qué se usa para probar la accesibilidad de un endpoint? Describa los pasos para su uso\*\*

Usamos WCFTestClient

C:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise\Common7\IDE

Ejecuta WCFTestClient y pega la ruta del servicio.

237. ¿Cómo debe realizarse la transferencia de mensajes en un WCF?

De forma segura.

238. ¿Cuáles son los tres puntos a considerar para tener un WCF seguro?

Integridad. El mensaje no debe ser modificado

Autentificación mutua. El cliente se conecta al servicio y el servicio se encarga de verificar que el cliente es quien dice ser.

Privacidad. Nadie ajeno debe de ver el mensaje, es privado.

239. ¿En qué parte del ABC se configura el modo de seguridad?

En el binding

240. Explique los cinco tipos de seguridad de un WCF.

1. None Sin seguridad.

2 Transport Seguro punto a punto pero no cuando viaja por intermediarios, encripta el canal de comunicación. Se aprovecha de la aceleración de las tarjetas para no cargar en cpu encriptando y desencintando el mensaje.

3 Message Seguro a punto a punto incluso cuando viaja por intermediarios, encripta el mensaje.

4 Mixto Encripta el canal de comunicación y los credenciales

Both Encripta el canal de comunicación y el mensaje.

241. Explique en detalle cual es la diferencia entre Seguridad Mixta y Both? \*\*

Que mixta utiliza el modo message pero solo para los credenciales y both para todo el mensaje.

242. ¿Cuáles son los dos puntos en los que se basan los modos de seguridad de WCF?

1. Encriptar el canal de comunicación

2 Encriptar el mensaje.

243. Describa los cuatro modos de autentificación en WCF.

Sin autentificación. Los clientes son permitidos

Autentificación por Windows. Se utilizan los credenciales de Windows

Certificados El cliente se autentifica por un certificado que el servicio conoce de ante mano

244. ¿En qué consiste el Tracing?

Tracing consiste en el proceso de recopilar mensajes y generar diagnósticos sobre la ejecución de la aplicación.

245. ¿Qué hace el listener? Cuatro distintas maneras de guardar la información son:

Es el encargado de escuchar los distintos eventos que se desencadenan.

Listener:

DefaultTraceListener

EventLogListener

EventProviderTraceListener

TextWriterTraceListener

XmlWriterTraceListener

DelimitedListTraceListener

ConsoleTraceListener

EventSchemaTraceListener

1. Archivos de texto
2. Xml
3. Base de datos

246. ¿Para qué se usa la información obtenida?

Para rastrear el origen del error.

247. ¿En qué archivo y en qué sección se configura el tracing?

En el archivo de configuración. En <system.diagnostics>

248. ¿Qué diferencia al tracing del debug?

Que el debug se utiliza en modo desarrollo y el tracing es para ambiente productivo.

249. ¿Cuáles son los cinco distintos niveles de trazabilidad existentes? ¿Puede encontrarse la misma información en niveles distintos? ¿Por qué?

0. None

1. critical Errores no manejados

2. Error Errores manejados

3. Warning Posibles errores

4 Information Mensajes positivos por ejemplo los de finalización

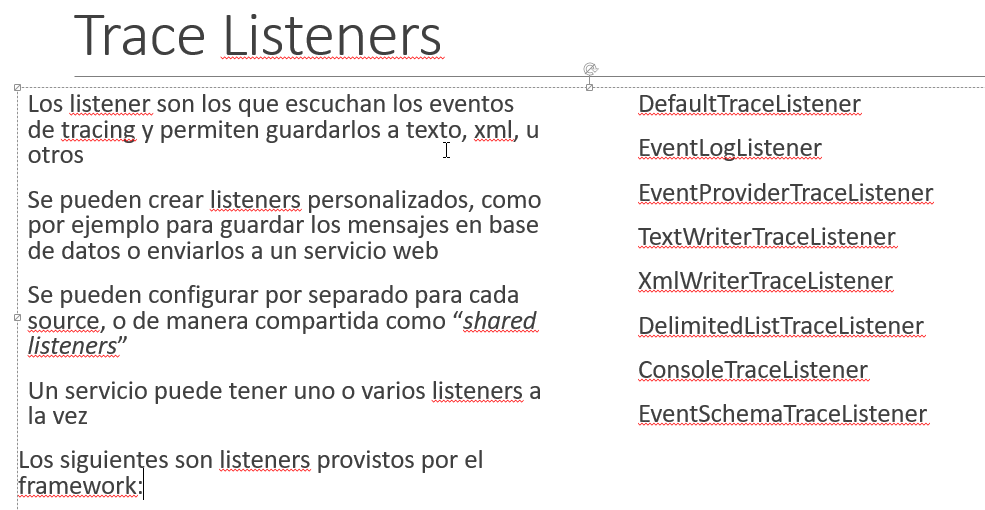
5 Verbose información de debugging

Estos errores contienen los anteriores

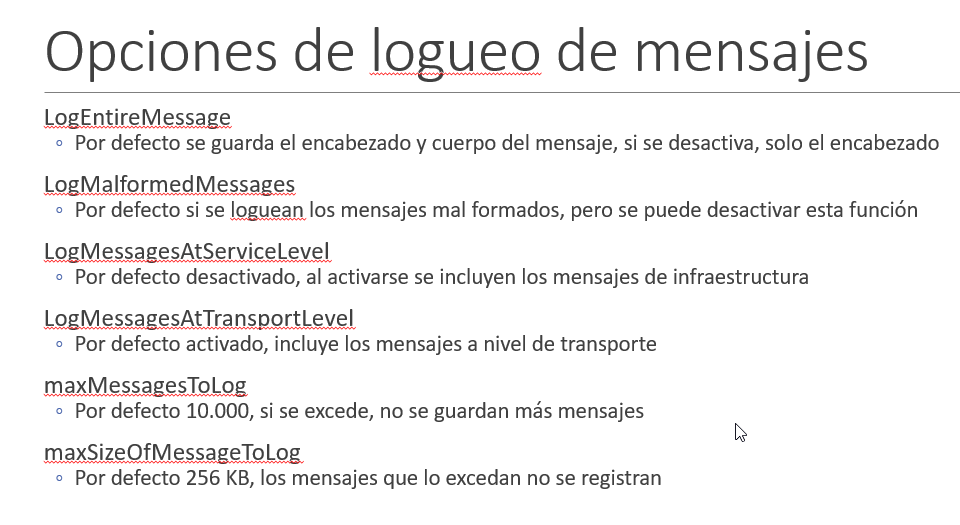
6 Activity Tracing Inicio y finalización de actividades, de flujo.

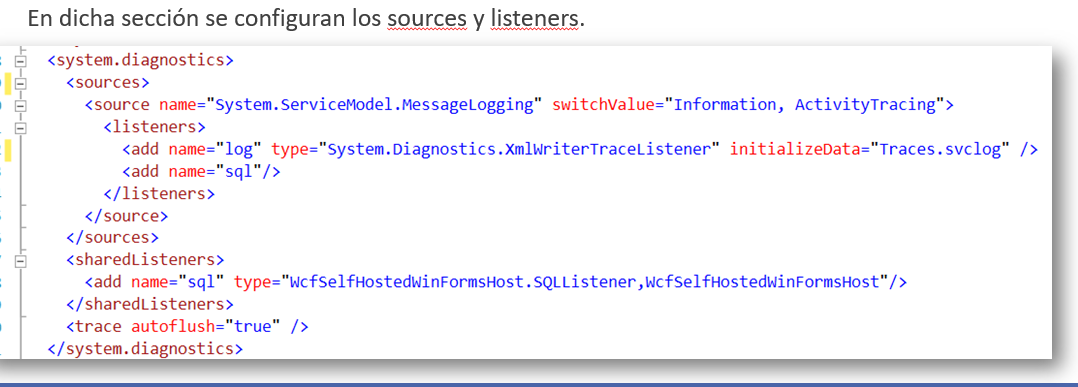
7 All todos los anteriores.

250. ¿Cuántos listener puede tener un servicio? ¿Se pueden personalizar?



251. Describa seis opciones de logueo de mensajes





252. ¿Se pueden guardar el tracing en una ruta específica? ¿Porque es necesario un formateador de XML?

Sí se puede guardar, se necesita un formateador de xml porque el mensaje se guarda en una sola línea.

SvcTraceViewer se puede utilizar.

253. ¿Para qué sirve el Autoflush?

Para limpiar el buffer de memoria automáticamente.

254. ¿Qué problema puede ocurrir si se usa un archivo para el log si existe mucha concurrencia?

Se pueden producir bloqueos.

255. ¿Qué es el web.config? ¿Dónde se ubica?

Archivo de configuración de toda aplicación, se ubica en la raíz del proyecto.

256. ¿Qué es el machine.config? ¿Puede haber más de uno en una sola máquina? ¿A cuáles aplicaciones se aplica su configuración? ¿Cuál es su diferencia respecto al Web.config?

Machine config es un archivo para todas las aplicaciones en un mismo servidor es compartido, puede haber uno por Framework instalado. El web.config es independiente de cada aplicación.

257. Tres caracteristicas de WCF Restful son:

1 Utiliza los verbos http.

2 Utilizan el protocolo http.

3 Tienen menos overhead que soap porque son más ligeros y veloces.

258. Explique en detalle la diferencia entre WCF y WebServices.

1. WCF es conjunto de librerías que provee le framework de .net. WCF utiliza protocolos de comunicación como basicHttpBinding WSHttpBinding WSDualHttpBinding y NetTcpBinding.

2. WCF se expone al mundo a través de interfaces

3 Robusto y seguro

WebServices

1. Es una capa de negocio que se accede por medio del protocolo SOAP.

2. Se solicita o envían peticiones por el protocolo como get, put, post y delete.

3. Solo se comunica por htpp, donde la seguridad no se encuentra activa.

259. ¿Cuál es la diferencia entre SOA y SOAP?

SOAP es un protocolo de comunicación basado en XML

SOA es una arquitectura para crear aplicaciones orientadas a servicios.

260. ¿Cuál es la diferencia entre SOAP y REST?

REST opera únicamente por el protocolo htpp y tienen menos overhead que SOAP.

**HTML5**

261. ¿Qué implementa que antes dependía de algo más?

Css y javascript.

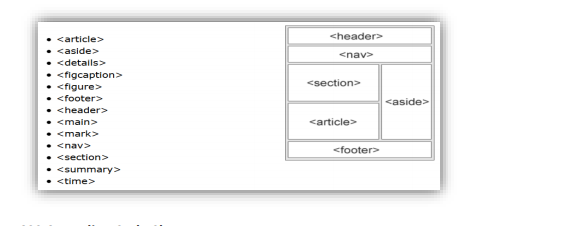
262. Explique cuatro caracteristicas que un sitio debe tener:

1 Perceptible 2 Operable 3 Entendible 4 Robusto.

263. ¿En que sección se puede declarar el <script>? ¿Dónde se recomienda hacerlo? ¿Cuáles son dos maneras de agregar código?

Se puede declarar tanto en el head como en el body. Por medio de un archivo externo <sciript src=”ruta”></script> o bien escribiendo código dentro de los tabs <scripts></scripts>

264. Mencione las 13 partes en las que puede dividirse un HTML? Haga un gráfico con la distribución de las 6 principales.



265. Defina CSS. ¿Qué hace?

Estilos en cascada que sirven para personalizar una página html

266. ¿Cuál es la característica más importante de CSS y porque?

El modelo de cajas boxModel, ya que condiciona el diseño de toda la página.

267. ¿Qué hace el modelo de cajas? ¿Cuáles son sus cuatro componentes?

Que todos los elementos de la página se representen mediante cajas rectangulares.

1 Margin

2 Border

3 Padding

4 Content

**XML**

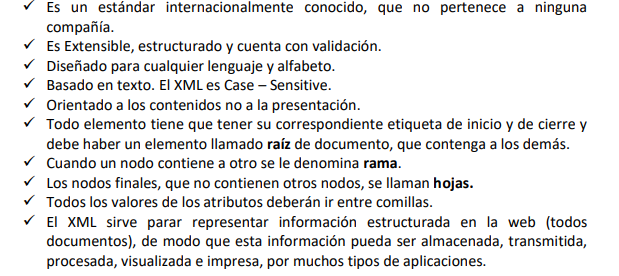
268. ¿Cuál es la relación entre XML y SGML? ¿Qué es SGML? \*\*

XML es un subconjunto de SGML. SGML es un estándar para definir lenguajes de marcado

269. ¿Qué es XML?

Es una especificación para diseñar lenguajes de marcado

270. Mencione y detalle diez características de XML



271. ¿Qué hace un DTD? ¿Qué se comprueba con él?

Define elementos, atributos entidades, notaciones, reglas de un documento. Por medio de este se define la validez de un documento.

272. ¿Qué es XSD? ¿A qué sustituye y por qué?

Formato para definir la estructura de un documento XML. Sustituye a DTD, debido a que detalla más la estructura de un XML.

273. ¿Qué es XSLT?

Convierte documentos XML a otro XML.

274. Dé cuatro ejemplos de conversiones posibles con XSLT.

Formatos finales:

1 : XHTML.

2: WML

274. ¿Qué es DTSX?

Es un formato de archivo basado en XML que almacena las instrucciones para el procesamiento de un flujo de datos.

**1. Métodos para publicar servicios web**

IIS: Basado en htpp, se ejecuta en el IIS. Se aprovecha de las características:

1. Apagado inactivo
2. Monitoreo de procesos
3. Reciclaje de procesos.

Windows services: Publicado en un servicio de Windows

Managed .net app: Requiere menos infraestructura menos robusto.

2. **Tipos de sesión de IIS**

**Modo In Proc:** Almacena el estado de sesión en el servidor web.

**Modo stateServer** Almacena el estado de sesión en un proceso llamado servicio de estado de ASP.NET

**ModoSqlServer:** Almacena el estado de sesión en una bd SQL, garantiza que se mantenga si se reinicia la aplicación.

**ModoCustom** Especifica un proveedor de almacenamiento especializado.

**Modo off** Por defecto apagado.

1. **Modos de autentificación de IIS**

**Anónima:** Predeterminada provee acceso a sitios públicos de una página.

**Básica:** Forma parte de la especificación http 1.0 utiliza las cuentas de los usuarios de Windows.

**Implícita:** Corrige los problemas de la autentificación básica entre ellos él envió de contraseñas sin formato.

**Autentificación de Windows.** Habilitada de forma predeterminada solo funciona en IE, utiliza kerberos.

**Certificado digital: único** método válido para validar a una persona en internet.

* **Qué ventajas da el uso de Generics en .NET sobre la utilización de colecciones que contienen objetos tipo Object?**

Creando una clase genérica, se puede crear una colección que garantiza la seguridad de tipos en tiempo de compilación.

* **Qué es una clase Genérica (Generics) en .NET? ¿Para qué se utilizan?**
  + Permite manejar información, manteniendo el tipo de dato una vez creada la clase. El ejemplo más común, es una lista de algún tipo (p.e. una lista de personas), por lo que a partir de ese momento, solo va a recibir personas.
* **Qué es HashTable?**

(También mapa hash, tabla de dispersión o tabla fragmentada) Estructura de datos que asocia claves con valores. La operación principal es la búsqueda.

* **Qué es ViewState en ASP.NET?**

Proporciona información de estado para una página específica ASP.NET.

* **¿En qué casos utilizaría String o StringBuilder?**

Cuando se utilizan métodos de la clase String se crea un nuevo objeto cuando se utiliza StringBuilder permite modificar el string sin necesidad de crear un nuevo elemento.

* **En un ensamblado, ¿qué significa el “Strong-Name”?**
  + Crea una identidad única para el ensamblado, por lo que la instalación puede evitar conflictos.
* **¿Qué es la impersonalización o suplantación, autenticación y autorización en ASP.NET? ¿Cómo se implementa?**
  + En ASP.NET la seguridad es un proceso de 2 capas, todas las solicitudes a nuestro servidor web se gestionan en primer lugar por IIS y luego por ASP.NET.
  + De este modo, inicialmente IIS podrá aceptar o rechazar la petición. Si IIS la acepta, entonces se pasará a ASP.NET (en forma de token de seguridad), donde se volverá a examinar, atendiendo ahora los criterios de seguridad de ASP.NET.
  + Los dos sistemas de seguridad, son independientes entre sí, pero trabajan en forma coordinada entre sí.
    - ***Autenticación***: se validan credenciales suministrados por el usuario, permitiendo confirmar que es quien realmente dice ser.
    - ***Autorización***: luego de la autenticación, se valida si el usuario tiene permiso o no para acceder al recurso solicitado.
    - ***Suplantación o impersonalización***: proceso por el cual, ASP.NET puede acceder a un recurso, con una cuenta distinta de la que arrancó el proceso de ASP.NET.
* **¿Qué es un ensamblado (Assembly) en .NET?**

Librería dinámica DLL o programa ejecutable en el cual pueden existir distintos espacios de nombres.

Proporcionan al common lenguaje rountine, información para estar al corriente de la implementación de tipos.

* **¿Qué es el Common Language Runtime (CLR)? ¿Para qué sirve?**
  + Núcleo de la plataforma .NET. Motor encargado de gestionar la ejecución de las aplicaciones desarrolladas y a las que ofrece numerosos servicios que simplifican su desarrollo, favorecen fiabilidad y seguridad.
* **¿Diferencia entre IDENT\_CURRENT, SCOPE\_IDENTITY y @@IDENTITY en SQL Server?**

IDENT\_CURRENT: no está limitado a una sesión o ámbito, puede obtener el valor de identidad de una tabla específica.

SCOPE\_IDENTITY: devuelve el último valor de identidad insertado en una tabla en la sesión actual, dentro del ámbito actual.

@@IDENTITY: devuelve el último valor de identidad insertado en una tabla en la sesión actual, diferenciándose del anterior que no está limitado a un ámbito específico.

# XML

* **¿Qué es XML?**
  + *Extensible Markup Language*: Lenguaje de estructura y orden de datos, se usa para tener datos legibles al almacenarlos y transformarlos.
* **¿Qué es XSDL?**
  + *XML Schema Definition Language*: lenguaje esquemático, en el cual indicamos la posible estructura y contenido que puede tener un fichero XML. ***Define estructura, tipos de datos de los nodos, si se pueden repetir, etc.***
* **¿Qué es XSLT?**
  + *XML Schema Language Translate*: lenguaje de programación declarativo. Mediante hojas de estilo XSLT, se puede transformar un XML en HTML, texto plano, o en otro XML distinto. ***Carga, transforma y estiliza los datos.***
* **¿Qué es WSDL?**
  + *Web Services Description Language*: formato del XML que se utiliza para describir servicios web, es el contrato. Nos permite conocer acerca de los métodos y parámetros disponibles.