Primera Parte.

1. ¿Qué es .NET Framework?

Es una plataforma o software que brinda soluciones de código, utilizados por diferentes programas, provee librerías de clases y el entorno de ejecución y de programación para el desarrollo de múltiples lenguajes.

2. ¿Qué es LINQ?

Es el lenguaje de consultas integrado o sus siglas en ingles lenguaje integrated query, que permite guardar, recuperar y administrar datos a través de consultas que representan a las consultas que se harian a una base datos con un lenguaje como sql.

3. ¿Cómo funciona LINQ?

Este extiende el lenguaje de consultas en sentencias similares pero escritas con sintaxis de programación como C# que asemeja la sintaxis sql y se usa para extraer y procesar datos en diferentes formas ya sea arrays, listas, clases enumeradas.

4. ¿Qué es la ejecución diferida y la ejecución inmediate en LINQ?

En la ejecución diferida la ejecución de la consulta se realiza al momento de iterar sobre el objeto que almacena la definición de la consulta, cuando en una variable solo almacenamos la consulta definida, asi permitiendo que nuevos elementos puedan aparecer en la consulta cada vez que se itera sobre el objeto, en la ejecución inminente no se almacena la definición de la consulta si no la consulta misma al convertir esta definición a lista a través de un método ToList() o ToArray(), asi la ejecución de la consulta se realiza al momento de inicializar la variable de tipo lista o estructura de datos donde se almacena la consulta y cada vez que se itere sobre esta simplemente se iterara pero no se ejecutara la consulta.

5. ¿Qué es un objeto y una clase?

Un objeto es la instanciación de una clase, es la representación de una clase a través de su instancia, el cual se usa para acceder a los métodos y valores o compartamientos y estados de una clase, una clase es una plantilla o bosquejo donde se almacenan variables y métodos que representan comportamientos y estados y que puede ser instanciada en otras clases y métodos para acceder a sus valores.

6. ¿Qué es herencia, polimorfismo y encapsulación?

Herencia es la capacidad de extender los valores y métodos de una clase o interfaz a otro, forzando asi comportamientos o extendiendo comportamientos y estados de sus clases/interfaces padres, polimorfismo es la capacidad de de respuesta distinta de diferentes objetos al mismo mensaje, asi al responder a la ejecución de un método heredado de una clase padre dos clases hijos pueden tener diferente respuesta o comportamiento, encapsulación es la capacidad de definir los accesos y visibilidad a las variables y métodos de una clase haciendolas accesibles solo dentro de la misma clase o dentro de otra clase que importe la clase que encapsula estos valores.

7. ¿Cuál es la diferencia entre una clase abstracta y una interfaz?

La clase abstracta permite declarar métodos con y sin implementación y permite tener valores no definidos o variables y permite extender estos a otras clases hijos, mientras que una interfaz solo permite declarar métodos sin implementación y variables finales que no pueden ser modificadas y extender estas métodos o comportamientos a otras clases o interfaces.

8. ¿Cual es la diferencia entre Override y Overload en un método?

Override es una anotación que permite sobre escribir el funcionamiento de un método y cambiar su funcionalidad manteniendo el mismo nombre que fue declara en la clase de la que se extiende, overload permite compartir el mismo nombre a diferentes métodos siempre y cuandos sus parámetros sean diferentes.

9. ¿Cuál es la diferencia entre struct y class?

La única diferencia es en un struct por defectos sus miembros son públicos a diferencia de una clase donde sus miembros por defecto son privados.

10. ¿Cuál es la diferencia entre encriptar un password y aplicarle un hashing?

El cifrado es la capacidad de encriptar un texto atraves de una clave privada si se tiene la clave privada se puede recuperar el texto y su longitud puede variar dependiendo del algoritmo utilizado para cifrar, un hash es cambio es un cifrado en base a un conjunto alfa numerico es y una longitud finita especifica de 32 caracteres sin importar la longitud del texto que se ha cifrado.

11. ¿Que es una patrón de diseño y para que sirve? Da algunos ejemplos

Los patrones de diseño permite gestionar la estructura de una aplicación y su desarrollo permitiéndonos tener un orden estructural en su implementación y desarrollo, la mas conocida como la MVC que permite estructurar un projecto en base a 3 ejes, vista controlador y modelo, haciendo que la funcionalidad gire entorno a estos parámetros, para poder recurar y proporciona valores de una vista y hacia un controlador y del controlador conectar a un modelo o vista paralelamente, también existen muchos otros patrones como AOP yOOP ect.

12. ¿Para que utilizamos la declaración “using”?

Para exportar librerías o clases en una clase y poder acceder a sus funciones y valores.

13. ¿Cómo ayuda .NET a gestionar las DLLs en un sistema?

Creando emsamblados que permiten gestionar la información guardada en los metadatos y poder guardar y ejecutar varias versiones de cada uno en el emsamblaje global del cache.

14. ¿Existe alguna diferencia entre "int" y "System.Int32"?

System.int32 es una estructura del namespace system o biblioteca, mientras que int es una palabra clave usada para representar a un valor primitivo entero.

15. ¿Microsoft NET Framework se basa en OOP o AOP?

Se basa en OOP siendo este un componente fundamental de .net permitiendo usar funciones como herencia, encapsulamiento, polimorfismo, abstracción.