Guía de preguntas entrevista de trabajo: .NET

TEST 1 - PREGUNTAS .NET

1. ¿QUÉ ES .NET?

Es una plataforma abierta de desarrollo creada por Microsoft para el desarrollo de múltiples aplicaciones.

Fuente: https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet

2. ¿VENTAJAS DE USAR .NET?

Tienes soporte al desarrollo de distintas aplicaciones como desktop, web, rest, mobile, etc, sobre una misma plataforma de desarrollo.

3. ¿CÓMO HARÍAS PARA REUTILIZAR CÓDIGO ENTRE .NET FRAMEWORK CON .NET CORE?

A través de .NET Standard, ya que este permite crear librerías que puedan ser usados entre las distintas aplicaciones de .NET.

Fuente: https://dotnet.microsoft.com/platform/dotnet-standard

4. ¿QUÉ ES LINQ?

LinQ nos permite manipular distintas fuentes de datos a través de un lenguaje de query que propone este.

Fuente: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq-queries

5. OVERLOADING VS OVERRIDING

- **Overloading**: crear varios métodos con el mismo nombre pero diferentes parámetros.
- **Overriding**: sobrescribir el comportamiento de un método padre.

Fuente: https://dotnet.microsoft.com/platform/dotnet-standard

6. ¿PARA QUE SE USA LA PALABRA RESERVADA ASYNC EN C#?

Para hacer uso de la programación asíncrona dentro de C# y que el método que encapsula pueda hacer uso de await.

7. ¿QUÉ ES UN DELEGADO?

Un delegado es similar a una interface, pero a nivel de métodos en el cual podemos definir los parámetros de entrada y salida de este. La implementación ya lo resolverá quien haga uso del delegado.

8. ¿READONLY VS CONST?

- **const**: indica que una propiedad o variable una vez definida no podrá ser modificada (se inicializa en el tiempo de compilación).
- **readonly**: similar al anterior pero la diferencia que esta se puede definir en tiempo de compilación o ejecución. Por ejemplo, podemos definir el valor de un readonly **a través del constructor**.

Fuente: https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/delegates

9. VALUE TYPE VS REFERENCE TYPE

- **value**: ciertas variables almacenan su valor en su propio espacio de la memoria y **si les preguntan mencione algunas**: *bool, byte, char, decimal, double, enum, float, int, long, sbyte, short, struct,uint, ulong, ushort*.
- **reference**: algunas variables al sobrescribir su valor generan otro punto de almacenamiento en la memoria. Ejemplo, las cadenas.

Fuente: https://www.tutorialsteacher.com/csharp/csharp-value-type-and-reference-type

10. ¿QUE ES UNA TUPLA (TUPLE)?

Una tupla permite agrupar múltiples tipos de datos en una sola estructura simple.

```
(double, int) t1 = (4.5, 3);
```

Fuente: https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/value-tuples

11. ¿CUÁL ES LA UTILIDAD DE LA PALABRA RESERVADA USING?

Automatiza la liberación de los objetos que implementen la interfaz IDisposable en vez de hacerlo a mano.

```
using (var reader = new StringReader(manyLines))

{

    string? item;

    do {

        item = reader.ReadLine();

        Console.WriteLine(item);

    } while(item != null);

}
```

Fuente: https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/using-statement

12. ¿QUÉ ES COMMON LANGUAGE RUNTIME?

Todos los lenguajes soportados en .NET como C# o VB se convierten al final en CLR para que pueda ser interpretado por la máquina.

13. ¿DIFERENCIA ENTRE UNA CLASE ABSTRACTA Y UNA INTERFACE?

- Una clase abstracta puede tener métodos abstractos (sin lógica) o no abstráctos (lógica).
- Una clase abstracta se hereda, una interface se implementa.
- Una interface solo puede hacer uso de métodos públicos.
- Una interface solo puede extender o heredar otras interfaces, mientras que una clase abstracta puede heredar cualquier otra clase o implementar interfaces.

Fuente: https://es.wikipedia.org/wiki/Common_Language_Runtime

14. ¿QUÉ ES INYECCIÓN DE DEPENDENCIA?

Es un patrón que permite desacoplar los diversos componentes (clases) de nuestro proyecto a través de un nivel de abstracción. De esta manera, cambiar una dependencia de nuestro proyecto no debe afectar el código de nuestro proyecto.

Fuente: https://anexsoft.com/ejemplo-de-inyeccion-de-dependencias-con-c

TEST 2 - PREGUNTAS .NET

Pregunta 1: ¿Qué es .NET?

Básicamente, tus entrevistadores querrán que expliques estas cosas con tus propias palabras. Te brindare unas de las más simples y *directo al grano* definiciones para que te hagas una idea de cómo formular tus respuestas.

"**.NET**" realmente no significa nada (Ni siquiera es una abreviación). El término completo a utilizar será "**Microsoft .NET**" ya que Microsoft es el creador de la estructura.

Entonces, ¿Qué es .NET?

.NET *es una estructura de desarrollo de software y sitios web.* Está considerada como una de las mejores y más confiables estructuras. Sin embargo, es realmente difícil aprenderlo. Muchas personas aman NET framework por la "*interoperabilidad del lenguaje,*" cualquier lenguaje en .NET puede utilizar el código escrito en otro lenguaje.

Otro punto por el que los desarrolladores aprecian Microsoft NET Framework es por su gran variedad de funciones y ajustes preestablecidos.

Pregunta 2: ¿Cuántos lenguajes son compatibles con Microsoft NET Framework?

Al momento de escribir este tutorial, NET Framework es compatible con 44 lenguajes diferentes.

Pregunta 3: ¿Cuál es la vida útil de los elementos en ViewState?

Los elementos en ViewState permanecerán activos siempre y cuando la página en la que residen no sea eliminada.

Pregunta 4: ¿Qué es "CTS"?

CTS acrónimo de *Common-Type System*. Básicamente, este sistema determina el tipo de datos que son utilizados dentro del mismo.

Debes prestar mucha atención a las preguntas entrevista de trabajo relacionadas con CTS, son muy importantes.

Pregunta 5: Defina "encapsulación"

Encapsulación, *es una función que incluye varios métodos y datos dentro de un proyecto*. Esto es realizado para que el objeto del programa puede realizar sus tareas de manera eficaz y sin ningún tipo de error.

Pregunta 6: ¿Cuál es la diferencia entre una "clase" y un "objeto"?

Otra pregunta entrevista de trabajo basada en definiciones, esta puede resultar sencilla si estás familiarizado con lenguajes de programación.

Para simplificarlo, un "**objeto**" *es algo que ocurre o está ubicado dentro de una* "**clase**" Las clases definen como se verá un objeto, como funcionaran y qué tipo de propiedades poseen. A su vez, los objetos de acción similares forman dichas clases.

Pregunta 7: ¿Existe alguna diferencia entre "debug" y "trace"?

Si, la clase **Trace** se puede utilizar para depurar y liberar ciertas compilaciones mientras que **Debug** se utiliza exclusivamente para, *lo has adivinado*, la depuración.

Esta puede ser clasificada entre las más complicadas preguntas entrevista de trabajo, ya que es muy fácil olvidarse de las funciones adicionales de la clase **Trace.**

Pregunta 8: ¿Cuál es la diferencia entre "in-process" y "out-of-process"?

Ambas son las encargadas de administrar las sesiones de memoria. **In-process,** almacena todos los datos de una sesión de desarrollo en un servidor web. A su vez, **out-of-process,** almacena los mismos datos en un área externa de administración de memoria. Probablemente una de las áreas de almacenamiento de datos externa más populares serían los servidores SQL.

Pregunta 9: ¿Qué es MSIL?

MISL, acrónimo de *Microsoft Intermediate Language.* Es utilizado para definir valores, almacenar memoria y otras tareas similares de alta prioridad. Todos y cada uno de los códigos utilizados en Microsoft NET Framework deben primero pasar por MSIL.

Pregunta 10: ¿Qué es "inheritance"?

Aunque esta no es necesariamente una de las preguntas principales en una entrevista de trabajo, usualmente deberás responderla debido a su relación con NET Framework.

Inheritance, (Herencia) ocurre cuando una clase pequeña toma las características y parámetros de una mucho más grande. Esta clase más grande es luego vista como la "clase principal" a la pequeña.

Esta es una excelente pregunta para ampliar durante una entrevista de trabajo. Microsoft NET Framework solo es compatible con herencia simple. Lo que significa que la clase pequeña solo se puede beneficiar de (Heredar) una clase principal.

Pregunta 11: ¿Cuál es la diferencia entre "managed code" y "unmanaged code"?

Managed code, es un tipo de código que fue creado y compilado dentro de Microsoft NET Framework. A su vez, **unmanaged code,** proviene de otro software de creación y trae consigo toda la estructura, características y ajustes del anterior.

Pregunta 12: ¿Existe alguna diferencia entre “int” y “System?Int32"

No existe absolutamente ningún tipo de diferencia entre "**int**" y "**System.Int32**." "**Int**" simplemente es una abreviatura del mismo nombre.

Pregunta 13: Defina "caching"

Otro término el cual estarás familiarizado gracias a otros lenguajes de programación, "**caching**" sigue siendo unas de las preguntas entrevistas de trabajo más populares.

Caching, *es un proceso en el que mantienes los archivos y datos más utilizados en una memoria separada.* Esta ubicación aparte, un *caché*, es donde podrás acceder a todos tus archivos designados. Caching, les permite a los desarrolladores ahorrar mucho tiempo e incrementar la administración de memoria.

Pregunta 14: ¿Qué es un "ensamblaje"?

Un **ensamblaje,** *es un lugar en donde se encuentran ubicadas todas las herramientas Microsoft NET Framework necesarias para crear programadas y sitios web.* Un desarrollador puede tener un ensamblaje **privado** y/o **compartido.**

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

OOP, acrónimo de *Object-Oriented Programming,* mientras que **AOP** es la abreviación de *Aspect-Oriented Programming.* Por lo tanto, ¿Cuál de los dos utiliza NET Framework?

La respuesta usualmente se encuentra escondida en las anteriores preguntas entrevista de trabajo. Microsoft NET Framework, **es exclusivamente una estructura OPP**, conceptos previamente mencionados como
la **Encapsulación** e **Inheritance** *son algunas de las características clave* de una estructura de programación orientada a objetos.

Este es probablemente una de las preguntas entrevista de trabajo más interesantes con la que te toparas. Sin embargo, la respuesta es bastante obvia, especialmente si conoces la diferencia entre OOP y AOP.