**A.) Teoria .NET**

**1. ¿Que es Microsoft Visual Studio .Net?**

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades en línea bajo Windows Azure en forma del editor Monaco.

Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno compatible con la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y videoconsolas, entre otros.

**2. ¿Qué es el Framework .Net?**

La plataforma .NET de Microsoft es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución es el producto principal en la oferta de Microsoft, y pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma Windows.

**3. ¿Qué es el CLR?**

Es el Common Language Runtime es el verdadero núcleo del framework de .NET, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios del sistema operativo (W2k y W2003). Permite integrar proyectos en distintos lenguajes soportados por la plataforma .Net, como C++, Visual Basic, C#, entre otros.

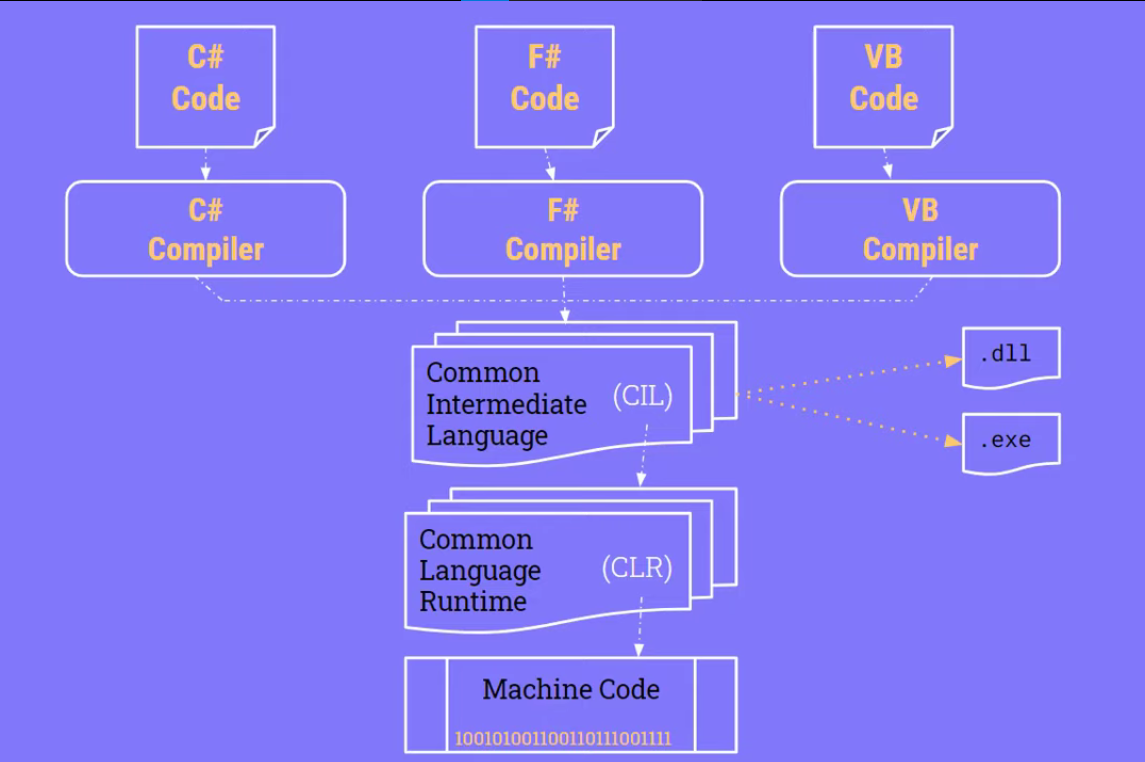
La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .NET en un código intermedio, el CIL (Common Intermediate Language) antes conocido como MSIL (Microsoft Intermediate Language), similar al BYTECODE de Java. Para generarlo, el compilador se basa en la especificación CLS (Common Language Specification) que determina las reglas necesarias para crear el código MSIL compatible con el CLR.

Para ejecutarse se necesita un segundo paso, un compilador JIT (Just-In-Time) es el que genera el código máquina real que se ejecuta en la plataforma del cliente. De esta forma se consigue con .NET independencia de la plataforma de hardware. La compilación JIT la realiza el CLR a medida que el programa invoca métodos. El código ejecutable obtenido se almacena en la memoria caché del ordenador, siendo recompilado de nuevo sólo en el caso de producirse algún cambio en el código fuente.

**4. ¿Qué es la BCL?**

La Base Class Library (BCL) Almacena todas las clases de la librería que requiere el . Net Framework para su ejecución. Permite crear librerías de clases personalizadas para facilitar la programación de los desarrolladores.

**5. Indique y explique el orden que se sigue en el proceso de compilación y ejecución en .net**

****

**6. ¿Cuál es la signatura de Main?**

**La signatura de main es la clase program**

**7. ¿Qué es un espacio de nombre (namespace)?**

Los espacios de nombres son un modo sencillo y muy eficaz de tener absolutamente todas las clases perfectamente organizadas, tanto las que proporciona el .NET Framework como las que podamos escribir nosotros.

**8. ¿Cómo se incluye una librería en C#?**

Una vez que tu proyecto está abierto en Visual Studio, para usar desde él una librería, se hace click derecho en la rama "Referencias" que aparece en el Explorador de Soluciones y se selecciona "Agregar Referencia". Entonces aparece una ventana que te permite buscar la librería y agregarla. Esto funciona para librerías de .Net y librerías COM. No vale para librerías binarias con puntos de entrada clásicos. Si necesitas usar una de estas, hay que usar "DllImport" desde el código.

**9. ¿Qué indica la palabra clave params?**

La palabra reservada params se usa en la declaración de los parámetros de un método, se antepone al tipo de dato que vamos a declarar y tiene una peculiaridad, solamente se puede anteponer a parámetros que sean arreglos.

Esto es porque lo que nos ayuda a escribir es un método que recibe una cantidad variable de argumentos del mismo tipo y los “introduce” un un arreglo sin tener que declararlo explícitamente.

**D) OBJETOS**:

**1. ¿Qué es una clase?**

Las clases son los tipos más fundamentales de C#. Una clase es una estructura de datos que combina estados (campos) y acciones (métodos y otros miembros de función) en una sola unidad. Una clase proporciona una definición para instancias de la clase, también conocidas como objetos.

**2. ¿Qué es un objeto?**

Una clase que se ha cargado en la memoria se denomina objeto o instancia. Se crea una instancia de una clase utilizando la palabra clave de C# new.

**3. Porque se caracterizan los objetos?**

Un objeto es una instancia de una clase Los objetos se caracterizan por:

* Identidad: Los objetos se distinguen unos de otros
* Comportamiento: Los objetos pueden realizar tareas
* Estado: Los objetos contienen información

**4. ¿Cómo se llama la táctica de obtener la forma mínima y esencial de un objeto?**

Mediante el operador new