6.3 MARCO REFEENCIAL

A continuación se van a describir las características mas importantes de la plataforma de trabajo .NET sobre la que se va a desarrollar nuestro software.

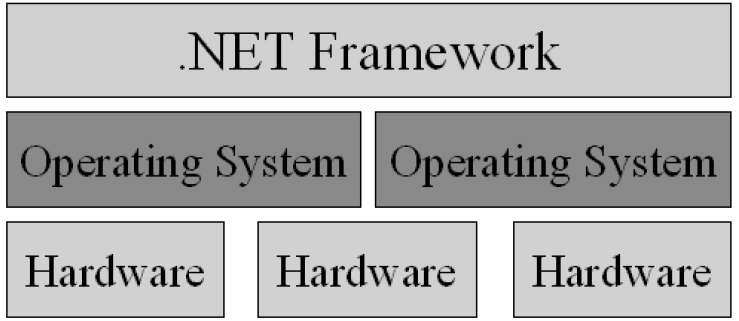
6.3.1 .NET framework y el entorno común de ejecución para lenguajes (CLR)

Uno de los ideales de la ciencia de la computación a través de los últimos años, ha sido el concepto de que un programa de computadora puede ser escrito para que sea ejecutado sin importar el Hardware con el que este compuesto.

Hasta cierto punto esta ideal se ha logrado al poder abstraer el acceso al Hardware utilizando un sistema operativo, desde este punto de vista la función básica de un sistema operativo es encapsular la capa de Hardware de un sistema, por lo que un programador no tenga la necesidad de preocuparse por registros, memoria, y otros problemas que ocurren a nivel de Hardware.

En los últimos años, este problema volvió a surgir, pero esta vez ya no a un nivel de Hardware, sino a un nivel de un sistema operativo, en el mercado actual se encuentran diferentes sistemas operativos Win32, Solaris, OSs, inclusive de un mismo Proveedor como por ejemplo Microsoft contiene una serie de sistemas operativos diferentes (Win98, WinXp, WinVista). El problema radica ahora en darle al programador, otro nivel de abstracción en donde este encapsulado el sistema operativo cuando se esta desarrollando una aplicación. En muchos sentidos esto es exactamente lo que realizo Microsoft con .NET.

El Entorno Común de Ejecución para Lenguajes o CLR (Common Language Runtime) por sus siglas en ingles, el cual es uno de los pilares del Framework o marco de trabajo de .Net , forma un ambiente que abstrae el sistema operativo.



.NET abstrae los sistemas operativos, los cuales abstraen el acceso a Hardware, esto significa que al desarrollar software, se esta escribiendo código encamidao al CLR y no directamente al sistema operativo, por lo que este software puede correr sobre diferentes sistemas operativos o cualquier plataforma que implemente el CLR. El CLR esta presente en el nucleo del Framework de .NET. El CLR provee un ambiente en donde las aplicaciones son ejecutadas, esto incluye conceptos como compilación, registración, y hasta problemas de desarrollo.

Como su nombre lo implica el CLR esta diseñado para soportar diferentes lenguajes de programación en una forma en común entre ellos. Esto es muy interesante ya que los desarrolladores de software no tienen que aprender un nuevo lenguaje de programación, para trabajar sobre .NET.

Microsoft creo los siguientes lenguages de programación encamidos para trabajar sobre .NET, los principales son: VB,C#,C++, JScript. Otras firmas estan trabajando para desarrollar lenguages de programación encamidados a .NET como COBOL, Phyton, y posiblemente JAVA.

* + 1. El proceso de compilación

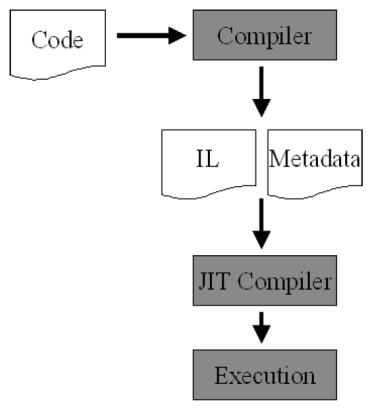
Una de las partes mas importantes del CLR es el proceso de compilación. Este proceso provee gran flexibilidad y es la raíz de muchas de las capacidades que provee .NET.

Todo el código en .NET es compilado a lenguaje de maquina, pero esta compilación típicamente ocurre en tiempo de ejecución, esta es una forma diferente de cómo se compilaban las aplicaciones anteriormente, en donde el código era compilado a lenguaje de maquina antes de la ejecución de la aplicación. .Net introdujo un nuevo lenguaje de programación el Lenguage Intermedio de Programacion de Microsoft o (MSIL) por sus siglas en ingles, Todos los lenguages que trabajan sobre .NET como VB, C#, C++, tienen compiladores que generan código MSIL, en vez de lenguaje de maquina como se hacia anteriormente. Eventualmente el MSIL ira hacer compilado en lenguaje de maquina para su ejecución.

Un lenguaje de programación no solo genera MSIL, también genera metadatos, estos metadatos son la clave que .NET provee para dar una descripción exacta de los componentes que fueron compilados a IL.

Antiguamente en COM los componentes creados eran descritos por un lenguaje de definición de interfaces de Microsoft MIDL por sus siglas en ingles pero este no daba una descripción suficiente y exacta de los componentes que se generaba este Framework, esto cuasaba problemas de interoperación entre componentes que eran escritos en C++, al ser llamados por código escrito en VB y viceversa.

Los metadatos usados en .NET son mucho mas robustos, los datos son almacenados en un formato XML, por lo que es accesido fácilmente y es bastante descriptivo. Estos metadatos describen de una forma muy exacta las interfaces de los componentes, el resultado de esto es que componentes creados en un lenguaje de programación, pueden ser llamados de una forma muy sencilla desde otros lenguajes de programación



Como el MSIL y los metadatos están disponibles por el CLR, la llamada a métodos de componentes creados entre diferentes lenguajes de programación es transparente, La herencia de clases entre diferentes lenguajes de programación es posible, el código puede ser verificado para detectar algún manejo inseguro, antes de que este entre a ejecución, .Net puede serializar automáticamente los objetos.

Adicionalmente, los metadatos eliminan la necesidad de escribir sobre el registro de Windows, como se realizaba anteriormente, toda la información de las interfaces de los componentes esta descrita en los metadatos.