



GUÍA DE APRENDIZAJE N° 7

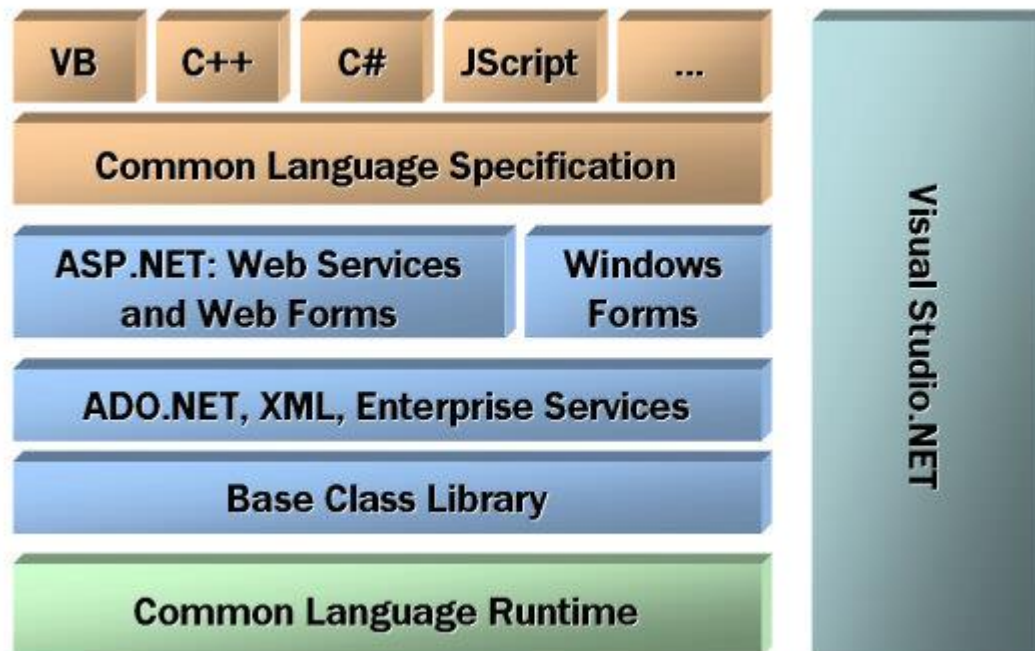
1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

<b>Programa de Formación:</b> Técnico en programación de software.	<b>Código:</b> <b>Versión:</b>	228120 100		
<b>Nombre del Proyecto:</b> Sistema de información para la gestión empresarial	<b>Código:</b>	471143		
<b>Fase del proyecto:</b>	FASE 1 Identificación del problema (Análisis y Diseño)			
<b>Actividad (es) del Proyecto:</b> Recolección de información y análisis de requerimientos.	<b>Actividad (es) de Aprendizaje:</b> Diseñar la arquitectura del software utilizando la programación orientada a objetos (POO)	Ambiente de formación ESCENARIO	<b>MATERIALES DE FORMACIÓN</b>	
			<b>DEVOLUTIVO</b>	<b>CONSUMIBLE</b>
<b>Resultados de Aprendizaje:</b>  22050103201 Identificar cada uno de los conceptos y principios que constituye la programación orientada a objetos para interpretar el diseño.	<b>Competencia:</b>  220501032 Analizar los requerimientos del cliente para construir el sistema de información.	Aula de Informática.  Computadores Actualizados  Conexión a internet  Ventilación adecuada  Sillas y escritorios ergonómicos  Extintor	-Portátiles  -Video Beam	-Papel
<b>Duración de la guía ( en horas):</b>	40			



## 2. INTRODUCCIÓN

### 2.1. PLATAFORMA .NET



La plataforma .NET es un componente de software el cual provee un conjunto extenso de soluciones predefinidas para satisfacer las necesidades generales de la programación de aplicaciones.

Ofrece un entorno gestionado de ejecución de aplicaciones, nuevos lenguajes de programación y compiladores. Permite el desarrollo de todo tipo de funcionalidades: desde programas de consola, servicios Windows hasta aplicaciones de escritorio, internet y dispositivos móviles.

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.

#### 2.1.1. COMPONENTES DE LA PLATAFORMA .NET

##### 2.1.1.1. Lenguajes de programación:

La plataforma .NET no está atada a un determinado lenguaje de programación ni favorece a uno determinado frente a otros. En la actualidad existen implementaciones para varias decenas de lenguajes que permiten escribir aplicaciones para la plataforma .NET. Los más conocidos son Visual Basic .NET, C# y J#.

##### 2.1.1.2. CommonLanguageSpecification:

La especificación común de los lenguajes son unas reglas que definen un conjunto de características que



todos los lenguajes deben incorporar. Una de las partes más importantes de la CLS es la que se refiere a los tipos de datos.

**2.1.1.3. Asp .Net:**

Permite desarrollar aplicaciones dinámicas orientadas a la Web.

**2.1.1.4. Windows Form:**

Nos permite desarrollar aplicaciones de escritorio las cuales se basan en ventanas y controles comunes de Windows que se ejecutan localmente.

Windows Forms está constituido por múltiples clases especializadas que ofrecen funcionalidades para el trabajo con ventanas, botones, rejillas, campos de texto y todo este tipo de controles habituales en las aplicaciones de escritorio.

**2.1.1.5. ADO.Net:**

El acceso a fuentes de datos es algo indispensable en cualquier lenguaje o plataforma de desarrollo. La parte que se especializa en el acceso a datos se denomina de forma genérica como ADO.NET.

**2.1.1.6. Base Class Library:**

La plataforma .NET nos ofrece infinidad de funcionalidades "de fábrica" que se utilizan como punto de partida para crear las aplicaciones. Existen funcionalidades básicas (por ejemplo todo lo relacionado con la E/S de datos o la seguridad) y funcionalidades avanzadas (acceso a datos, creación de aplicaciones Web, etc).

Toda esta funcionalidad está implementada en forma de bibliotecas de funciones que físicamente se encuentran en diversas DLL (bibliotecas de enlazado dinámico). A su conjunto se le denomina Base Class Library (Biblioteca de clases base o BCL) y forman parte integral de la plataforma .NET, es decir, no se trata de añadidos que se deban obtener o adquirir aparte.

**2.1.1.7. CommonLanguageRuntime:**

Es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma .NET. El CLR es el encargado de compilar el código intermedio llamada CLI (CommonIntermediateLanguage) al código de maquina nativo, mediante un compilador en tiempo de ejecución.

Los desarrolladores que usan CLR escriben el código fuente en un lenguaje compatible con .NET, como C# o Visual Basic .NET. En tiempo de compilación, un compilador .NET convierte el código a CIL. En tiempo de ejecución, convierte el código CIL en código nativo para el sistema operativo. Alternativamente, el código CIL es compilado a código nativo en un proceso separado anterior a la ejecución. Esto acelera las posteriores ejecuciones del software debido a que la compilación de MSIL a nativo ya no es necesaria.

El CLR también permite otros servicios importantes, incluyendo los siguientes:



- Administración de la memoria
- Administración de hilos
- Manejo de excepciones
- Recolección de basura
- Seguridad

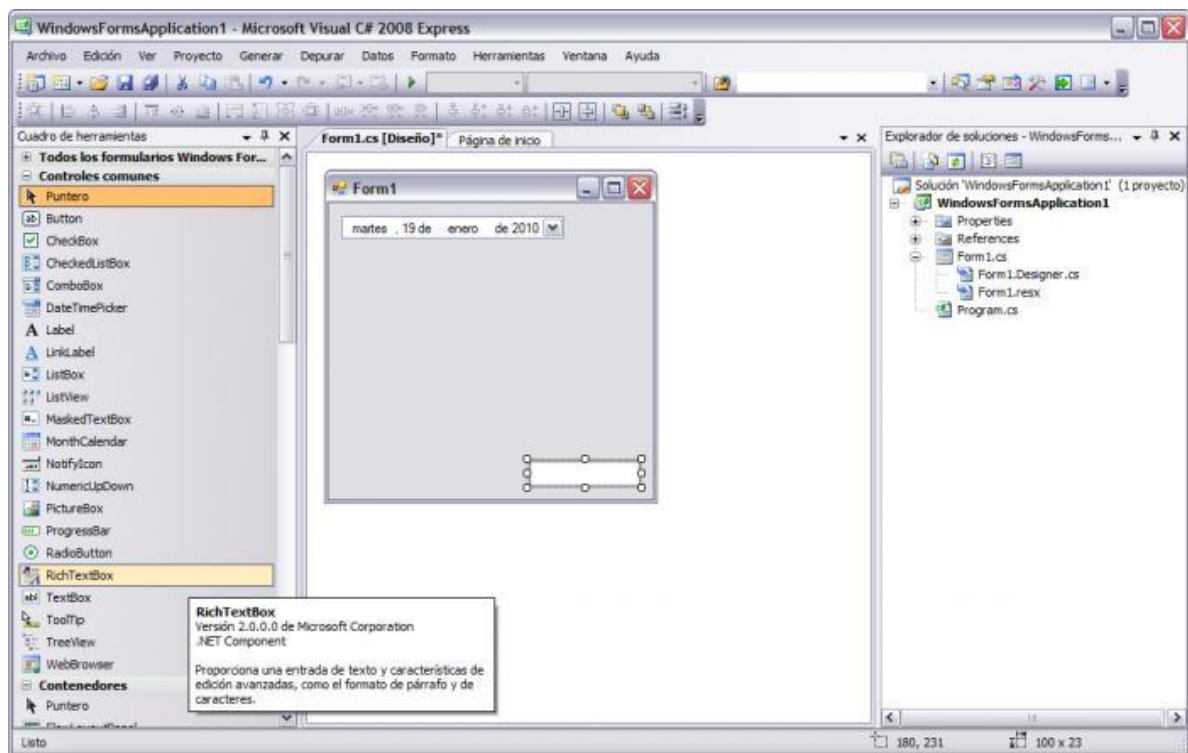
### 2.1.1.8. Visual Studio .NET:

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para la creación de aplicaciones de Consola, Windows Form, Web, Móviles, Servicios Windows y Servicios Web.

Este software brinda las herramientas necesarias para crear, distribuir, administrar y dar mantenimiento a las aplicaciones .NET.

Visual Studio .NET permite también la integración y el uso cruzado de lenguajes de programación: Visual Basic .NET, Visual C# .NET, Visual C++ .NET y JScript .NET.

Es importante aclarar que en el desarrollo de nuestro programa de formación estará basada en la plataforma .NET, pero utilizaremos los lenguajes de programación C# y ASP.NET.





## 2.2. VARIABLES

Una variable puede verse simplemente como un cajón donde se puede almacenar información de un determinado tipo de dato, como un número, un carácter, una palabra, un conjunto de palabras, entre otras. Una variable se debe declarar asignándole un nombre y el tipo de dato que podrá almacenar. Para poderla utilizar solo se debe llamar a la variable con su respectivo nombre. El valor que la variable almacena puede cambiar, pero el nombre sigue siendo el mismo.

### 2.2.1. TIPOS DE DATOS EN C#

Nombre corto	Clase .NET	Tipo	Ancho	Intervalo (bits)
byte	Byte	Entero sin signo	8	0 a 255
sbyte	SByte	Entero con signo	8	-128 a 127
int	Int32	Entero con signo	32	-2.147.483.648 a 2.147.483.647
uint	UInt32	Entero sin signo	32	0 a 4294967295
short	Int16	Entero con signo	16	-32.768 a 32.767
ushort	UInt16	Entero sin signo	16	0 a 65535
long	Int64	Entero con signo	64	-922337203685477508 a 922337203685477507
ulong	UInt64	Entero sin signo	64	0 a 18446744073709551615
float	Single	Tipo de punto flotante de precisión simple	32	-3,402823e38 a 3,402823e38
double	Double	Tipo de punto flotante de precisión doble	64	-1,79769313486232e308 a 1,79769313486232e308
char	Char	Un carácter Unicode	16	Símbolos Unicode utilizados en el texto
bool	Boolean	Tipo Boolean lógico	8	True o false
object	Object	Tipo base de todos los otros tipos		
string	String	Una secuencia de caracteres		
decimal	Decimal	Tipo preciso fraccionario o integral, que puede representar números decimales con 29 dígitos significativos	128	$\pm 1.0 \times 10e-28$ a $\pm 7.9 \times 10e28$



### 2.2.2. DECLARACION DE VARIABLES

Para declarar una variable en C# se debe escribir el tipo de dato que podrá almacenar la variables, seguido del nombre de la variables y por ultimo un punto y coma, como se muestra en la siguiente ilustración:

```
<tipoVariable> <nombreVariable>;
```

Ejemplo: Declarar una variable de nombre miCedula donde se podrá almacenar un numero entero:

```
Int miCedula;
```

Ejemplo: Declara una variable de nombre miApellido donde se podrá almacenar un conjunto de caracteres:

```
String miApellido;
```

### 2.2.3. ASIGNAR VALORES A UNA VARIABLE

Después de declarar la variable se puede asignar un valor a esta, para asignar un valor escriba el nombre de la variable seguido de un signo = (signo de asignación) y posteriormente el valor que se va a asignar a la variable, terminando con un punto y coma.

Ejemplo: Asignar el valor de una cedula a la variable de nombre miCedula:

```
miCedula = 80237864;
```

Ejemplo: Asignar el valor de un apellido a la variable de nombre miApellido:

```
miCedula = "Gutierrez";
```

Nota: Los valores de las variables de tipo de dato `string` van entre comillas dobles.



#### 2.2.4. OPERACIONES ENTRE VARIABLES

Se pueden ejecutar operaciones aritméticas entre variables.

Ejemplo:

Crear tres variables Numero1, numero2, resultado.

Asignar el número 7 a la variable Numero1.

Asignar el número 12 a la variable Numero2.

Sumar la variable Numero1 y Numero2.

Asignar la suma a la variable resultado.

```
int numero1;
```

```
int numero2;
```

```
int resultado;
```

```
numero1 = 7;
```

```
numero2 = 12;
```

```
resultado = numero1 + numero2;
```

“La variable resultado tendrá el valor de 19”



### 3. ESTRUCTURACION DIDACTICA DE LAS ACTIVIDADES DE APRENDIZAJE

#### 3.1 Actividades de Reflexión inicial.

Después de realizar la lectura participe en el **Foro “Plataforma .NET – Variables”**, dando su aporte a las dos preguntas planteadas a continuación :

¿Cómo la plataforma .NET mediante sus diferentes componentes nos permite desarrollar e integrar aplicaciones de software?

¿Cómo los diferentes tipos de datos y las variables influyen en la organización y estructura de la información para la construcción de las aplicaciones de software?

#### 3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje.)

3.2.1. En equipos de trabajo identificar y listar:

- Las características de los diferentes tipos de aplicaciones de software (programas) que utilizan comúnmente u observan utilizar.
- Los diferentes datos que usan a diario y clasifíquelos según sus categorías. Según estos datos defina qué información se podría relacionar con estos.

3.2.2. Socializar dicha evidencia desarrollada.

#### 3.3 Actividades de apropiación del conocimiento (Conceptualización y Teorización).

3.3.1. Realice la consulta de los siguientes ítems:

- Diferentes definiciones de la plataforma .NET.
- Cual es la estrategia y objetivo de la plataforma .NET.
- Describa todos sus componentes y de un ejemplo de cada uno de ellos.
- Existe una plataforma similar a esta, si es así describa sus similitudes y diferencias.
- Diferentes tipos de datos.
- Clasifique los tipos de datos según su categoría.
- Defina que es una variable.





- Describa las características que debe tener una variable.
- Defina las formas de asignar valores a una variable.

### **3.4 Actividades de transferencia del conocimiento.**

#### **3.4.1. Responda las siguientes preguntas:**

- Describa los lenguajes más utilizados en la plataforma .NET. ¿Qué diferencia hay entre ellos?
- Qué diferencia hay entre aplicaciones Windows Form y aplicaciones ASP.NET. De un ejemplo real de cada una de ellas.
- De qué trata la tecnología ADO.Net.
- Mencione y describa las librerías más utilizadas en la plataforma .Net.
- Describa los diferentes componentes del entorno de desarrollo Visual Studio .NET.
- Qué ventajas tiene utilizar la plataforma .NET para el desarrollo de su proyecto de Software.
- Realice un mapa conceptual donde se identifiquen los componentes de la plataforma .NET.

#### **3.4.2. Desarrolle los siguientes ítems:**

- Seleccione 6 tipos de datos. Defina por cada uno el intervalo y menciones 3 ejemplos de cada uno de las variables que podría utilizar.
- Cree un mapa conceptual donde muestre la estructura de los diferentes tipos de datos.
- Liste los tipos de datos y variables que utilizará para el desarrollo de su proyecto de Software.



### 3.5 Actividades de evaluación.

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
<b>Actividad de aprendizaje 1 (Reflexión Inicial)</b> <b>Nombre Actividad:</b> Componentes plataforma .NET y Tipos de datos <b>Evidencia:</b> Conclusión del foro	Elabora algoritmos de acuerdo con un problema dado.	Foro
<b>Actividad de aprendizaje 2 (Contextualización)</b> <b>Nombre Actividad:</b> Características tipos de aplicaciones y aplicación de los tipos de datos. <b>Evidencia:</b> Archivo en Word		Lista de Chequeo
<b>Actividad de aprendizaje 3 (Conceptualización)</b> <b>Nombre Actividad:</b> Conceptos de la plataforma .NET y variables <b>Evidencia:</b> Evaluación y/o Prueba		Cuestionario de preguntas
<b>Actividad de aprendizaje 4 (Aplicación al Proyecto)</b> <b>Nombre Actividad:</b> Tipos de datos y variables. <b>Evidencia:</b> Presentación		Lista de chequeo



#### 4. RECURSOS PARA EL APRENDIZAJE

ACTIVIDADES DEL PROYECTO	DURACIÓN (Horas)	Materiales de formación devolutivos: (Equipos/Herramientas)		Materiales de formación (consumibles)		Talento Humano (Instructores)		AMBIENTES DE APRENDIZAJE TIPIFICADOS
		Descripción	Cantidad	Descripción	Cantidad	Especialidad	Cantidad	ESCENARIO (Aula, Laboratorio, taller, unidad productiva) y elementos y condiciones de seguridad industrial, salud ocupacional y medio ambiente
Recolección de información y análisis de requerimientos.	40	-Equipos de computo -Video Beam	30 1	Papel		Programación de software	1	- Aula de computo - Computadores - Conexión a internet -Ventilación adecuada Sillas y escritorios ergonómicos - Extinguidor



## 5. GLOSARIO DE TERMINOS

**.NET:** .NET es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

**Lenguaje de programación:** Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras.

**ASP.Net:** ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.

**Windows Forms:** Windows Forms (WinForms) es el nombre dado a la gráfica de la interfaz de programación de aplicaciones (API) que se incluye como parte de Microsoft .NET Framework, que proporciona acceso a los nativos de Microsoft Windows elementos de la interfaz envolviendo la existente API de Windows en código administrado.

**Visual Studio:** Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET. Aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

**Dato:** El dato es una representación simbólica (numérica, alfabética, algorítmica, entre otros) de un atributo o característica de una entidad. Los datos describen hechos empíricos, sucesos y entidades.

**Información:** La información es un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.

**Tipo de dato:** Es un atributo de una parte de los datos que indica al ordenador (y/o al programador) algo sobre la clase de datos sobre los que se va a procesar. Esto incluye imponer restricciones en los datos, como qué valores pueden tomar y qué operaciones se pueden realizar. Tipos de datos comunes son: enteros, números de coma flotante (decimales), cadenas alfanuméricas, fechas, horas, colores, etc.

**Variable:** Son espacios reservados en la memoria que, como su nombre indica, pueden cambiar de contenido a lo largo de la ejecución de un programa. Una variable corresponde a un área reservada en la memoria principal del ordenador.



## 6. REFERENTES BIBLIOGRÁFICOS

Plataforma.Net. En línea. Disponible en:

- <http://infoprogramacion1.blogspot.com/p/plataforma-net.html>  
(Consulta 02/02/2012)

La plataforma.Net. En línea. Disponible en:

- <http://people.cs.uchicago.edu/~borja/pubs/revistaeside2002.pdf>  
(Consulta 02/02/2012)

Constantes y variables. En línea. Disponible en:

- [http://msdn.microsoft.com/es-es/library/wew5ytx4\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/wew5ytx4(v=vs.90).aspx)  
(Consulta 02/02/2012)

Variables tipos de datos y operadores. En línea. Disponible en:

- <http://atc.ugr.es/pedro/tutoriales/cursos/csharp/variables.htm>  
(Consulta 02/02/2012)

## 7. CONTROL DEL DOCUMENTO (ELABORADA POR)

**ELABORADA POR: Paola Cárdenas Jiménez**

**REVISADA POR: Edwin Cañón Mozo**