



MATERIA: PROGRAMACIÓN

BLOQUE : I.

ASESOR:

M.T.E. JOSÉ LUIS PÉREZ GARCÍA



Correo electrónico del asesor
joseluisperez@sealapiedad.edu.mx
Página web para los materiales
joseluisperez.sealapiedad.edu.mx

SIMBOLOGÍA

Icono	Descripción
 REFLEXIONA	Este icono te invita a que <u>prestes especial atención</u> al tema tratado. Sé cuidadoso en estos puntos, debido a que, por su relevancia, debes enfocarte un poco más en ellos.
 LEER	Cuando este icono aparezca en tus guías de estudio significa que tu maestro de asignatura te está haciendo una invitación a que, por medio de <u>una lectura detallada</u> , logres tus objetivos de aprendizaje.
 EVALUACIÓN	Mediante este icono te podrás dar cuenta de los espacios en que se implementarán <u>actividades donde se valoren los conocimientos</u> que has construido.
 ACTIVIDAD	Bajo este símbolo encontrarás diversos elementos que te permitirán <u>practicar lo aprendido</u> , incluidos, por ejemplo, resúmenes, mapas mentales, ejercicios, tareas, reportes de lecturas, etc. ENTREGABLES.
 RECUERDA	Es fundamental que este icono lo tengas en consideración, debido a que te indicará aspectos de tu guía que debes <u>analizar cuidadosamente</u> , gracias a su importancia en el desarrollo adecuado de los temas estudiados.
 ADVERTENCIA	Este símbolo hace énfasis en aquellos aspectos de la guía que <u>no debes dejar pasar por alto</u> , debido a la enorme importancia que implican para el avance de tu guía. Estos conceptos o contenido seguro vendrán en tu examen.
 PORTAFOLIO DE EVIDENCIAS	Este ícono hace referencia a aquellas actividades que has realizado y que deben ser “almacenadas” como evidencia de tu avance académico. <u>Es importante que guardes estos elementos</u> , debido a que podrán ser requeridos durante toda tu estancia en la asignatura.
 OBLIGATORIO	Al enviar un mensaje de <u>correo electrónico</u> al asesor debes asegurarte de que lo envíes con la siguiente información en el apartado Asunto o subject: Nombre de la actividad, grupo al que perteneces , matrícula, <i>nombre de la asignatura y tu nombre completo</i> .

CONTENIDO

1

Lógica de programación.

2

Lenguajes de programación.

3

Programación.

4

Lenguaje de alto nivel.

Evaluación Diagnóstica

Contesta con cuidado, conscientemente y de manera reflexiva.



Evaluación
Diagnóstica

1. ¿Cómo se hacen los aplicación o programas o softwares o apps?
2. ¿Qué nombre se le da a la herramienta especial para hacer programas?
3. ¿Elementos para resolver problemas mediante algoritmos?
4. ¿Cómo estructurarías una solución de un problema dado para su posterior programación?

Sistema de Evaluación

Que se te **calificará**:

1. Diagramas de flujo en PesInt 30 %.
2. Programación de dos ejercicios en Visual Estudio 40 %.
3. Examen escrito 30 %.

Sistema de
Evaluación

Lógica de programación

¡El primer paso para aprender a programar!

¿Te interesa aprender sobre programación y no sabes por dónde comenzar?

Sin duda una respuesta para esta pregunta es: *Lógica de programación*. Y bueno... una cierta afinidad con las matemáticas y el inglés te vendrían bien como aliados.

La lógica es la base de todo el conocimiento en programación, ya que es posible aprender a escribir un código para que la computadora interprete correctamente. Es decir que ***aprendes a comunicarte con la máquina a partir de un lenguaje***.

La lógica de la programación es la organización coherente de las instrucciones para que su objetivo sea alcanzado.

¿Qué es un **Algoritmo** en la lógica de programación?

Conjunto de instrucciones ordenadas y en secuencia para resolver un problema.

Para darnos una idea, podemos identificar a una receta de cocina como un algoritmo. Es decir, la receta de cocina cumple con ser un conjunto de instrucciones ordenadas para resolver una situación problema.

Como se aprecia los algoritmos están presentes en la vida cotidiana, aunque no se note a simple vista.

Las características del algoritmo son: *preciso* (indica el orden de las cosas), *definido* (los pasos deben ser claros y exactos) y *finito* (determina el número de pasos para finalizar el proceso).

El método

Ya que se entendió lo que es el algoritmo, se puede pensar que el logro de un buen algoritmo debe seguir una **metodología**, todo inicia con la identificación del problema, continua con el planteamiento de alternativas de solución, después la elección de una alternativa continúa el desarrollo de la solución y termina con la evaluación.



Después de aplicar la metodología a un problema es momento de simplificarlo, aquí entran los diagramas.

Diagrama de flujo

Un diagrama es la representación gráfica del **algoritmo** (conjunto de instrucciones ordenadas para resolver un problema).

El objetivo de los diagramas ya sea de flujo, de procesos u otros es: pasar un problema de la vida real, que los seres humanos entendemos, a un lenguaje que la computadora entienda. La computadora solo entiende, conoce o trabaja con 0 (ceros) y 1 (unos); mientras que los humanos trabajamos con letras, números y otros símbolos.



La idea sería:

1	2	3	4	5
Realidad	Algoritmo	Diagrama de flujo	Programación por Experto (instrucciones o pseudocódigo)	Usuario
Problema de la vida real: sumar dos números cualesquiera	1. Obtener los dos números a sumar	Inicio	Inicio	Programa o software que al abrirlo al usuario se le solicitan dos números, se realiza la operación de la suma de los dos números e imprimir en la pantalla el resultado
		A, B	Variable A, B, R	
	2. Realizar la suma de los dos números	R = A + B	Leer A, B	
		R	R = A + B Escribir R	
3. Mostrar el resultado		Fin	Fin	

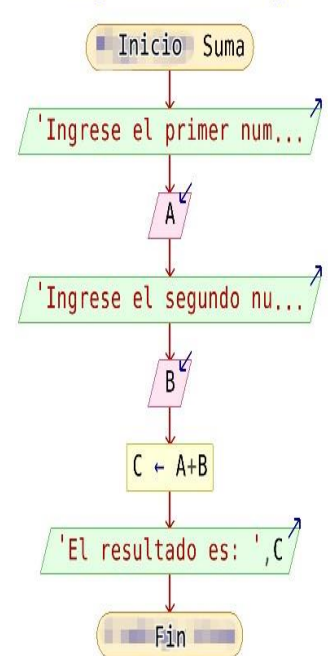
Realidad  **PC**

Ya que entendimos el problema y organizamos la solución llega el momento de hacer que la computadora lo entienda.

Lenguajes de programación

También llamado pseudocódigo, es el lenguaje que sólo los expertos y principiantes en el lenguaje de las computadoras entienden, es decir, es un lenguaje intermedio entre el lenguaje humano y el lenguaje de la computadora.

Para darnos una idea a continuación tenemos el pseudocódigo (en la columna izquierda) para hacer una suma de dos números, y su equivalente en diagrama (columna derecha):

<pre> 1 2 // este es el ejemplo más simple, 3 // toma dos numeros, los suma y muestra el resultado 4 5 Algoritmo Suma 6 7 // para cargar un dato, se le muestra un mensaje al usuario 8 // con la instrucción Escribir, y luego se lee el dato en 9 // una variable (A para el primero, B para el segundo) con 10 // la instrucción Leer 11 12 Escribir "Ingrese el primer numero:" 13 Leer A 14 15 Escribir "Ingrese el segundo numero:" 16 Leer B 17 18 19 // ahora se calcula la suma y se guarda el resultado en la 20 // variable C mediante la asignación (<-) 21 22 C <- A+B 23 24 25 // finalmente, se muestra el resultado, precedido de un 26 // mensaje para avisar al usuario, todo en una sola 27 // instrucción Escribir 28 29 Escribir "El resultado es: ",C 30 31 FinAlgoritmo 32 </pre>	<h3>Diagrama de flujo</h3>  <pre> graph TD Start([Inicio Suma]) --> Input1[/'Ingrese el primer num...'/] Input1 --> A[/A/] A --> Input2[/'Ingrese el segundo nu...'/] Input2 --> B[/B/] B --> Process[C <- A+B] Process --> Output[/'El resultado es: ',C'/] Output --> End([Fin]) </pre>
--	---



RECUERDA

Estructuras de control

También, ocupamos de **estructuras de control** para dirigir o cambiar el curso del flujo en el pseudocódigo/programa.

Secuencial o imperativa		Condicionales o de selección o de decisión		Iterativas o de repetición	
<i>Tienen como característica esencial que todas sus instrucciones se ordenan en una sucesión, ya que después de ejecutarse una de las instrucciones se debe continuar con la ejecución de la instrucción que está colocada inmediatamente después, y así se continúa hasta llegar a la final del algoritmo.</i>		Se usa para permitir la selección o ejecución de un grupo de instrucciones que se selecciona de entre uno, dos o varios grupos posibles; de acuerdo con el valor falso o verdadero (o sea: si o no) de una condición o una expresión lógica.		<i>En algunos casos se requiere que el algoritmo repita ciertas acciones hasta que alguna condición se cumpla y entonces se detiene el proceso.</i>	
Simple	Con resultados de transición	Simple	Doble	Ciclo definido en el algoritmo	Ciclo no definido en el algoritmo
<i>Se sigue el orden: datos, proceso y resultados</i>	Algoritmos que requieren resultados previos o resultados parciales (ejemplo: velocidad)	<i>La condición puede ser falsa o verdadera (ejemplo: F o V)</i>	La condición puede cumplirse o no cumplirse (ejemplo: SI o NO)	<i>Cuando se conoce el número de veces que se debe repetir el bloque (ejemplo: 64)</i>	<i>Cuando no se conoce desde antes el número de veces que se debe repetir el bloque (ejemplo: C > N)</i>

Hasta aquí, no es de preocuparse si las ideas aún no están del todo claras, es momento de practicar para aclararlas mejor.

- ✓ Hagamos un viaje de programación a un videojuego muy popular “Minecraft”.
- ✓ Entra a esta página web y practica un poco: <https://code.org/minecraft>.
- ✓ Escoge un tutorial o atrévete a recorrer los cuatro (o más) para aprender los conceptos básicos de la programación:

Disponible en Code.org

All tutorials are available in many languages and compatible with modern browsers and tablets. Don't have internet?

[Download the offline version of Minecraft Adventurer!](#)

GRADOS: 2-12

Minecraft Voyage Aquatic



Build and explore with Minecraft! Use your creativity and problem solving skills to explore and build underwater worlds with code.

Comenzar

Ver plan de lección

GRADOS: 2-12

Minecraft Hero's Journey



Build and explore with Minecraft! Players write code that instructs the Agent to execute their commands and overcome in-game obstacles.

Comenzar

Ver plan de lección

GRADOS: 2-12

Minecraft Adventurer



Learn the basics of computer science by programming Alex or Steve to move through a simulated piece of a Minecraft world.

Comenzar

Ver plan de lección

<https://code.org/minecraft>

Es hora de programar nuestros primeros intentos:

1. Busca en internet el programa **PseInt**, es gratuito, bájalo e instálalo en tu computadora.
2. Ve los siguientes videos en YouTube para que aprendas el uso básico del programa y realiza los siguientes ejercicios, te recomiendo estos dos videos:
 - A) Ventana de trabajo (<https://youtu.be/A6jtcWHErb0>).
 - B) Uso básico de PSeInt (<https://www.youtube.com/watch?v=Urf3s3Wz5Eo>).



ACTIVIDAD

✓ **Diagramas** de flujo a realizar en el **PseInt**:

1. La suma/resta/multiplicación/división de dos números cualesquiera e imprimir en pantalla el resultado.
2. Imprimir en pantalla si un alumno aprobó o reprobó sumando 3 calificaciones distintas de 3 bloques de la asignatura de Informática 1.

Programación

De manera general la programación es un proceso que se utiliza para idear y ordenar las acciones que se realizarán en el marco de un proyecto.

En informática es el arte de **indicarle a una computadora lo que tiene que hacer** mediante un conjunto de instrucciones.

Para llegar a tener una secuencia de instrucciones que den solución a un problema es necesario ejecutar varias etapas.

- Etapa de análisis.
- Etapa de solución general.
- Etapa de prueba.
- Etapa de uso.

Un **programador** es aquella persona que escribe, depura y mantiene el código fuente de un programa informático, es decir, el conjunto de instrucciones que ejecuta el hardware de una computadora, para realizar una tarea determinada.

Este programador, programa informáticamente (escribe) las líneas de código en programas llamados **lenguajes de programación** (programas para hacer programas).

Son lenguajes creados por el ser humano para poder comunicarse con las computadoras.

Lenguaje de programación es un conjunto de reglas o normas, símbolos y palabras especiales utilizadas para construir un programa y con él, darle solución a un problema determinado.

El lenguaje de programación es el encargado de que la computadora realice paso a paso las tareas que el programador ha diseñado en el algoritmo. Se puede decir que un lenguaje de programación es el intermediario entre la máquina y el usuario para que este último pueda resolver problemas a través de la computadora haciendo uso de palabras (funciones) que le traducen dicho programa a la computadora para la realización de dicho trabajo.

Niveles de los Lenguajes de Programación

Desde que se desarrollaron las máquinas programables se han desarrollado lenguajes con los cuales las personas puedan dar órdenes a éstas. En su orden los lenguajes de programación se pueden clasificar así:

- **Lenguaje de máquina:** Las primeras computadoras se programaban en código de máquina. Se puede decir que los programas eran diseñados en código binario. Eran difíciles de leer, difíciles de entender y por su puesto difíciles de corregir. Los programas se caracterizaban por ser pequeños.

- **Lenguajes de Bajo Nivel:** Para dar solución a lo difícil que era programar en código máquina, se desarrolló un lenguaje conocido como lenguaje ensamblador. Este lenguaje era encargado de tomar algunas palabras comunes a una persona y traducirlas al código máquina. Lo anterior facilitaría un poco la escritura de programas.
- **Lenguajes de Alto Nivel:** Como las personas resuelven problemas y se comunican en lenguajes naturales (español, inglés, francés, etc.), se desarrollaron lenguajes de programación que estuvieran más cerca de esta manera de resolver problemas. De los lenguajes de alto nivel se puede citar el Basic, Cobol, Fortran, Pascal, Turbo Pascal, C, Modula, Ada. Como se hace necesario traducir el programa a lenguaje de máquina, en los lenguajes de alto nivel esa operación la realiza algo que se conoce con el nombre de Compilador.

Tipos de Programación

Dependiendo del lenguaje de programación que se elija, se puede hablar del tipo de programación que se va a realizar.

- **Secuencial:** Se considera programación secuencial a los programas que se diseñan con instrucciones que van unas detrás de otras. Las líneas se ejecutan una a una en secuencia. Ejemplos tales como Basic, Cobol.
- **Estructurada:** Se considera programación estructurada a la programación que se hace por módulos. Cada módulo realiza alguna tarea específica y cuando se necesite esa tarea simplemente se hace el llamado a ese módulo independiente de que se tengan que ejecutar los demás. Ejemplos tales como: Turbo PASCAL, C, Modula, Ada.
- **Orientada a Objetos:** Se considera programación orientada a objetos aquellos lenguajes que permiten la utilización de objetos dentro del diseño del programa y el usuario pueden pegar a cada objeto código de programa. Ejemplos de estos lenguajes se pueden mencionar el Visual Basic de la Microsoft, C Builder de la Borland Internacional, Java, Xml, entre otros.
- Lógica o de **lenguaje natural:** son aquellos programas que se diseñan con interfaces tal que la persona o usuario puede ordenar a la máquina tareas en un lenguaje natural. Pueden interactuar como una persona, pero nunca llegan a producir conocimiento. Ejemplo como Prolog (Programming Logic). Estos lenguajes se desarrollaron con base en las estructuras de sus antecesores. Recorren o navegan las bases de datos obedeciendo las reglas.
- **Inteligencia Artificial:** Los programas de inteligencia artificial son programas que se acercan a la inteligencia humana. Estos programas son capaces de desarrollar conocimiento. Este tipo de lenguajes trabajan similar a la mente humana.

Programación utilizando un lenguaje de alto nivel

Microsoft Visual Studio Express Edition es un programa de desarrollo en entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas



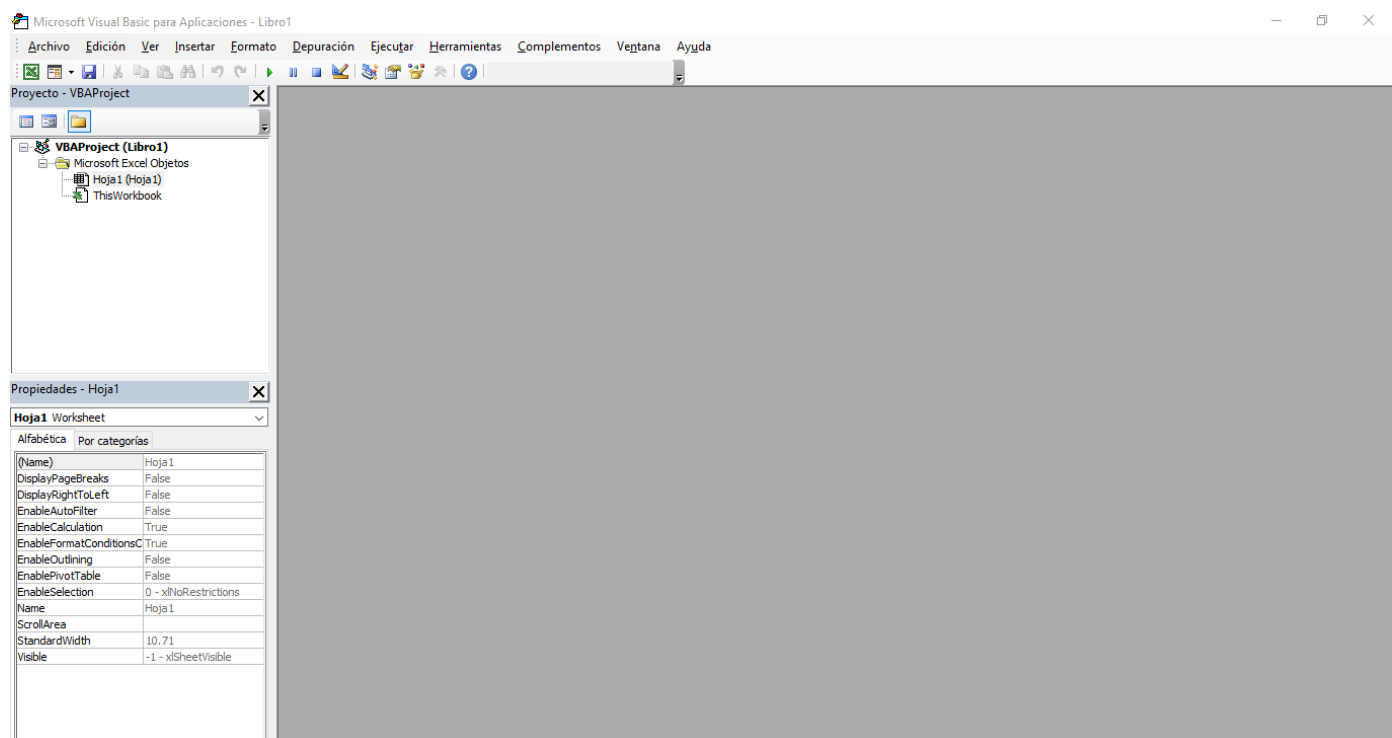
operativos Windows desarrollado y distribuido por Microsoft Corporation. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros. Es de carácter gratuito y es proporcionado por la compañía Microsoft Corporation orientándose a principiantes, estudiantes y aficionados de la programación web y de aplicaciones, ofreciéndose dicha aplicación a partir de la versión 2005 de Microsoft Visual Studio.

La liga de **descarga** es: <https://visualstudio.microsoft.com/es/vs/express/?rr=https%3A%2F%2Fes.wikipedia.org%2F>.

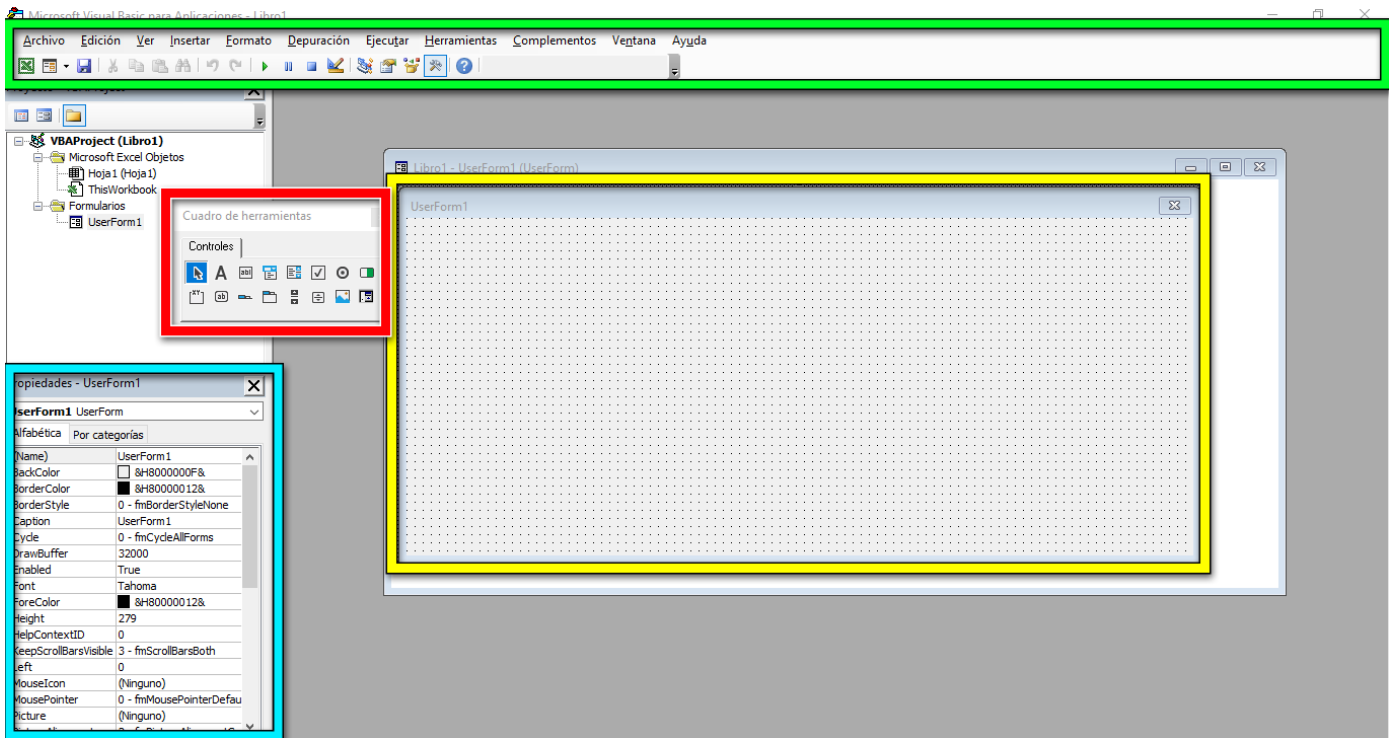
Sin embargo, para facilitarte la vida puedes utilizar a “Visual Basic” – considerado como el papá de “Visual Studio” – que viene integrado con Excel, solo debes presionar las teclas “**Alt**” + “**F11**” dentro de Excel y se abrirá la ventana de trabajo.

Entorno de desarrollo

Este es el espacio de trabajo o entorno de trabajo, de esta manera luce la ventana en la que estarás trabajando el desarrollo de las líneas de programación para los dos ejercicios a resolver (ver apartado actividades). *Recuerda el lugar y estilo de las áreas puede cambiar.*



Principales elementos/áreas de la ventana de trabajo de Visual Estudio.

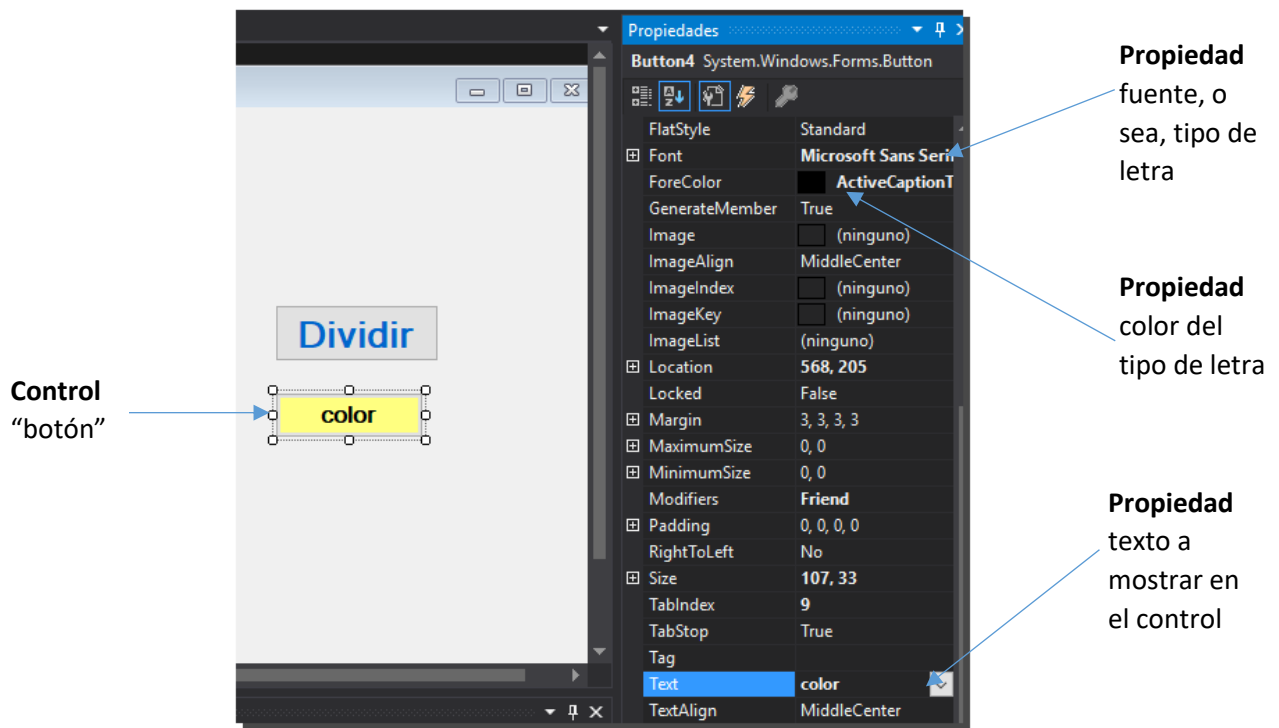


Cómo puedes ver en la imagen anterior las áreas a reconocer primordialmente son las coloreadas, de éstas debes saber algunos puntos:

- ✓ El área en amarillo, en este caso de programación, es el formulario y este será la interfaz gráfica que comunicará a la computadora con el usuario a través de controles.
- ✓ El área en azul, mostrara las propiedades del control que tengamos seleccionado en el formulario.
- ✓ El área en rojo, muestra los controles que se pueden agregar al formulario.
- ✓ Entonces combinadas las tres áreas podremos agregar controles a la interfaz del usuario (formulario) y modificar las características de dichos controles para poder personalizarlos a voluntad.

Por ejemplo: si tienes un cuadro de texto en el formulario que diga “hola mundo”, con letra “arial”, tamaño de letra “12” y color “negro” y deseo cambiar esas características que debo hacer.

1. El control que muestra el texto tiene un nombre, sino lo he cambiado, seguro se llamara “**Label1**”; desde su creación, por default, ese control contara con las **propiedades** -vistas en el área en **azul** - antes listadas (color, tamaño, etc.).
2. Para poder cambiar desde el código fuente dichas propiedades tendremos algo cómo: **Label1.Forecolor = Color.Blue**
3. Para cambiarlas o preestablecerlas desde antes sólo debo hacerlo en la caja de **propiedades** (área en azul).



Como podrás darte cuenta, existen muchos tipos de proyectos, muchos controles para cada proyecto y muchas propiedades para cada control.

El secreto está en ir conociendo para que se utiliza cada tipo de proyecto.

Saber que controla cada control y cuales controles tiene que proyecto.

Identificar las propiedades de cada control.

Y saber cambiar las propiedades de dichos controles desde antes de iniciar el programa y durante su ejecución.

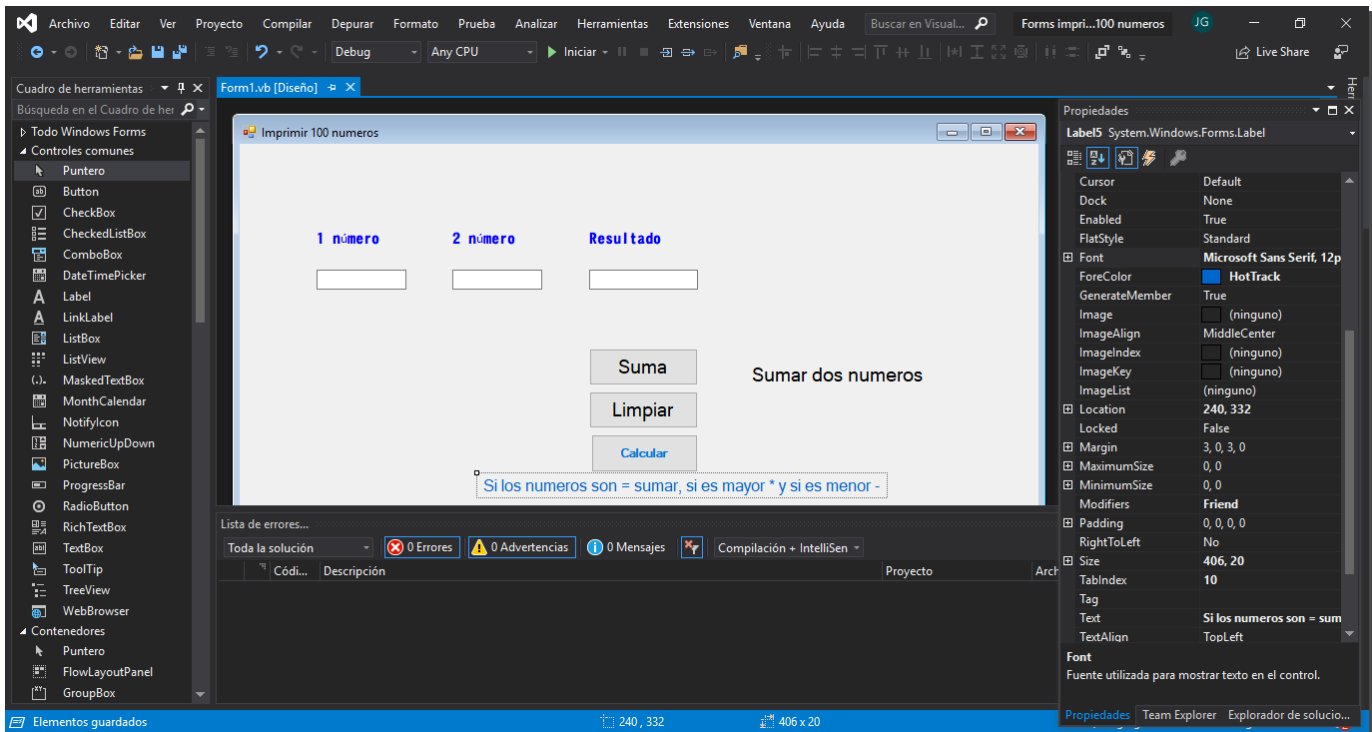
Seguramente te será un poco más difícil cambiar las propiedades cuando el programa ya este iniciado pues esto sólo se puede realizar desde las líneas de código.

“Lo olvidaba, el conocimiento que tengas del idioma inglés te ayudara un poco más ya que prácticamente todos los nombres de los controles y sus atributos están en ese idioma”

A continuación, veamos como programar en Visual Estudio (recuerda aún que estamos programando en Visual Basic ambos son casi idénticos).

Iniciemos con un problema: “la suma de dos números cualesquiera”.

1. Ya abierto el programa “Visual Studio o Visual Basic”, y habiendo seguidos los pasos dados anteriormente. Nos quedara un área en blanco en el centro de la pantalla.
2. Ahora intentaremos lograr algo similar a la pantalla de abajo:



Para lo anterior, sigue las instrucciones del video (el primer video no tiene sonido, solo sigue la imagen):

Primer programa (<https://youtu.be/kmNQKO6N2c?si=edNZ0e8vMLxIQ5tB>).

Ejemplo básico del uso de TextBox y Command en Visual Basic (https://youtu.be/-mANS4MX_mE?si=AbdAjLi6QJLOBNxs).

Continuemos practicando, para realizar la actividad anterior (imagen de arriba), ocuparas:

- ✓ El formulario (form: puedes crear en el menú “insertar” la opción “UserForm”).
- ✓ 3 controles de etiqueta (Label).
- ✓ 3 controles caja de texto (TextBox).
- ✓ 2 controles de botón (Button).

Tu programación o pseudocódigo para el botón “calcular” se verá más o menos así:

```
Form1.vb*  X
VB suma  Button1  Click
1 reference
Public Class Form1
  0 references
  2 Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
  3     If TextBox1.Text = TextBox2.Text Then
  4         TextBox3.Text = Val(TextBox1.Text) + Val(TextBox2.Text)
  5     End If
  6     If TextBox1.Text < TextBox2.Text Then
  7         TextBox3.Text = Val(TextBox1.Text) * Val(TextBox2.Text)
  8     End If
  9     If TextBox1.Text > TextBox2.Text Then
  10        TextBox3.Text = Val(TextBox1.Text) - Val(TextBox2.Text)
  11    End If
  12 End Sub
  13 End Class
  14
```

Finalmente, no te pierdas la explicación en clase para entender el detalle de los nombres y de los procedimientos para llegar a una buena programación. Lo necesitas ya que tu deberás realizar los dos ejercicios que son solicitados en el apartado “actividades”, más debajo de este texto.

Ahora, práctica colocando un botón más (el tercero) en el proyecto anterior, dale en nombre de “Dividir” y realiza en el mismo la programación para que divida el número 1 con el número 2 y que muestre el resultado (recuerda, para este ejercicio escribir el número 1 mayor que el número 2).

La idea es algo como esto:

The screenshot shows a standard Windows application window with a title bar containing the text 'Form1'. The main area of the window is light gray and contains three text boxes arranged horizontally. The first two are labeled 'Número 1' and 'Número 2' respectively. Below them are three buttons: 'Dividir' (with a blue background and white text), 'Calcular' (with a white background and blue border), and 'Limpiar' (with a white background and gray border).

¡NO LO HAS LOGRADO AÚN!

Se **honesto y realista contigo mismo**, sólo si ya practicaste y ya lo intentaste varias ocasiones, y sólo en ese caso, abre la carpeta donde el asesor te paso el material entra a la carpeta “Ejercicios” y “suma” y abra el proyecto “suma.vbproy”. <-- para Presencial.

Revisalo, pruébalo y aprende de él, después ciérralo e intenta crear el propio desde cero.

Recuerda, puedes apoyarte en videos de Youtube, tutoriales de Microsoft y otros, unos muy buenos son:

- ✓ Visual Basic, mi primer programa
(<https://youtu.be/OCW4ICAKnVQ?si=PQHHiWQsS5Azktlh>).
- ✓ Proyecto Control de Restaurante Parte 1 – excelente ejemplo par entender el uso real
– (https://youtu.be/1ftV_vNtmz0?si=ZGSpkK4JuJWijctp).



ACTIVIDAD

Es momento de trabajar en cómo realizar nuestra programación en Microsoft Visual Basic o Estudio u otra aplicación para programar, en el que gustes.

1. Programar (hacer el pseudocódigo) los dos ejercicios realizados en PseInt en Visual Basic.
2. Hacer el pseudocódigo/programa que imprima en pantalla los números del 1 al 10.
3. Crea el pseudocódigo/programa del proceso en el que se almacenen tres números en tres variables A, B, C. El programa debe decidir cuál de los tres números es el mayor y cuál es el número menor.

NOTAS:

- a) Envía por correo electrónico al asesor los archivos fuente de los dos ejercicios hechos en PseInt.
- b) Envía por correo electrónico al asesor los archivos fuente de los cuatro programas hechos en Visual Basic.



PORTAFOLIO DE EVIDENCIAS

Ya terminada tus cuatro programas **envía** los archivos/códigos fuente al asesor para su revisión y calificación.

No olvides guardar los códigos fuente en tu **Portafolio de Evidencias**.

Revisando tus conocimientos

Qué sabes hacer ahora...



Contesta con atención y honestamente. Si tienes serias dificultades para contestar estas preguntas es mejor que estudies un poco más, tu material de estudio, antes de avanzar.

1. ¿En qué consiste programar?
2. ¿Qué es la lógica de programación?
3. ¿Fases de la metodología de programación?
4. ¿Qué es un lenguaje de programación?
5. ¿Niveles de los lenguajes de programación?
6. ¿Tipos de programación?
7. ¿Principales áreas de la ventana de trabajo de Visual Studio?