

PROGRAMACIÓN I

TP 2

# GUÍA DE REVISIÓN CONCEPTUAL

INTRODUCCIÓN A LA PROGRAMACIÓN VISUAL #2

## UNIDAD 1

**INTRODUCCIÓN A LA PROGRAMACIÓN VISUAL**

**Autor de contenidos:**

Nicolás Battaglia

## OBJETIVOS

Formalizar conceptos teóricos sobre la programación visual

## ENUNCIADO

**RESPONDA LAS SIGUIENTES PREGUNTAS**

1. ¿Qué característica posee la programación visual
2. ¿Qué es un control de usuario?
3. ¿Cuáles son los tres elementos constitutivos que un control de usuario debe poseer?
4. ¿Para qué se utiliza la ventana denominada Solution Explorer?
5. ¿Para qué se utiliza la ventana denominada Properties?
6. ¿Para qué se utiliza la ventana denominada Toolbox?
7. ¿Qué es y que puede contener una Solución en el entorno de trabajo visto en clase?
8. ¿Qué es y que puede contener un Proyecto en el entorno de trabajo visto en clase?
9. ¿Qué es un formulario?
10. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Button.
11. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control CheckBox.
12. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Combobox.
13. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control DateTimePicker.
14. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Timer.
15. Seleccione 10 controles adicionales y de cada uno de ellos describa 5 métodos, 5 propiedades y 5 eventos.

## Parte 2

1. ¿Cómo se declara un procedimiento? Ejemplo.
2. ¿Cómo se declara una función? Ejemplo
3. ¿Qué diferencia existe entre una función y un procedimiento?
4. ¿Qué significa que el parámetro de un procedimiento se pasa por valor?
5. ¿Qué significa que el parámetro de una función se pasa por referencia?

## Parte 3

1. Enumere las estructuras de decisión. Ejemplifique cada una de ellas y explique en que se diferencian.
2. Enumere las estructuras de repetición. Ejemplifique cada una de ellas y explique en que se diferencian.
3. ¿Cómo se crea una estructura en el entorno visto en clase?
4. ¿Qué es una propiedad?
5. ¿Qué es un método?
6. ¿Qué es un evento?
7. ¿Cuál es el uso ideal de una variable de tipo Boolean?
8. ¿Para que se usa el método Parse cuando se lo utiliza por ejemplo como Decimal.Parse?
9. ¿Cómo se declara un vector de 10 posiciones de tipo Decimal?
10. ¿Cuándo hablamos de boxing y unboxing a que nos referimos?

Desarrollo

1. ¿Qué característica posee la programación visual?

Las características que posee la programación visual son:

Visual: los elementos se arrastran y sueltan en el flujo del programa para integrarlos.

Orientado a acontecimientos: cada paso del programa comienza cuando ocurre un acontecimiento previamente definido.

Imperativo: la programación consiste en una secuencia de comandos.

Orientado a objetos: hay objetos individuales con tareas definidas asignadas.

1. ¿Qué es un control de usuario?

Un control de usuario es un tipo de control compuesto que funciona de forma similar a la de una página Web ASP.NET: se pueden agregar controles de servidor Web y marcado a un control de usuario, así como definir propiedades y métodos para el control.

1. ¿Cuáles son los tres elementos constitutivos que un control de usuario debe poseer?

-Propiedades: “Variables” que permiten definir el estado de un control, puede ser por ejemplo el color, el tamaño, entre otras

-Método: Subrutina cuyo código es definido en una clase y puede pertenecer tanto a una clase, como es el caso de los métodos de clase o estáticos, como a un objeto, como es el caso de los métodos de instancia.

- Eventos: Reacciones que tienen los controles ante distintos contextos, por ejemplo, una interacción con el usuario por medio de un clic

1. ¿Para qué se utiliza la ventana denominada Solution Explorer?

La ventana denominada Solution Explorer se utiliza para explorar y administrar los proyectos y soluciones en las que estamos trabajando en Visual Studio.

1. ¿Para qué se utiliza la ventana denominada Properties?

La ventana denominada Properties se utiliza para visualizar y modificar las propiedades de los controles de usuario.

1. ¿Para qué se utiliza la ventana denominada Toolbox?

La ventana denominada Toolbox se utiliza para observar e ingresar los controles de usuario disponibles en los formularios de Windows.

1. ¿Qué es y que puede contener una Solución en el entorno de trabajo visto en clase?

Una solución es un tipo de archivo contenedor con uno o más proyectos relacionados, junto con información de compilación, la configuración de ventanas de Visual Studio y archivos varios que no estén asociados a un proyecto determinado.

1. ¿Qué es y que puede contener un Proyecto en el entorno de trabajo visto en clase?

Un proyecto es un tipo de archivo que contiene todos los elementos que se compilan en un ejecutable, biblioteca o sitio web. Estos elementos pueden incluir código fuente, iconos, imágenes, etc. Un proyecto también contiene la configuración del compilador y otros archivos de configuración que podrían ser necesarios en diversos servicios o componentes con los que el programa se comunica.

1. ¿Qué es un formulario?

Un formulario es una superficie visual en la que se muestra información al usuario suelen ir acompañados de controles de usuario con respuestas a las eventos que ocurren, como clics del mouse o presiones de teclas.

1. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Button.
2. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control CheckBox.
3. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Combobox.
4. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control DateTimePicker.
5. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Timer.
6. Seleccione 10 controles adicionales y de cada uno de ellos describa 5 métodos, 5 propiedades y 5 eventos.

## Parte 2

1. ¿Cómo se declara un procedimiento? Ejemplo.

public void VerResultado() {

Console.WriteLine("¡¡¡Ganador!!!");

}

Un procedimiento es un bloque de instrucciones delimitadas por una instrucción de declaración ( Function , Sub , Operator , Get , Set ) y una End declaración coincidente.

1. ¿Cómo se declara una función? Ejemplo

int calculo () {

resultado=dato1+dato2;

return resultado;

}

La función devuelve un resultado al código invocante. La ejecución de return provoca la salida de la función.

1. ¿Qué diferencia existe entre una función y un procedimiento?

La diferencia es que las funciones devuelven un resultado al código que las llamo, mientras que los procedimientos no devuelven ningún resultado

1. ¿Qué significa que el parámetro de un procedimiento se pasa por valor?

Que el parámetro de un procedimiento se pase por valor significa que el procedimiento recibe sólo una copia del valor que tiene la variable, o sea que no la puede modificar.

1. ¿Qué significa que el parámetro de una función se pasa por referencia?

Que el parámetro de un procedimiento se pase por referencia significa el procedimiento recibe la posición de memoria donde esta guardada la variable, por lo que la función puede saber cuánto vale, pero además puede modificarla de cualquier manera.

## Parte 3

1. Enumere las estructuras de decisión. Ejemplifique cada una de ellas y explique en qué se diferencian.

- Estructura if: La estructura IF permite controlar la ejecución o no de una porción de código según se cumpla una función establecida. Si la condición es verdadera, entonces la instrucción se ejecuta; en este caso, «condición» debe ser una expresión que, una vez evaluada, debe devolver una booleana true o false. Con esta sintaxis, sólo la instrucción colocada después del if, se ejecutará si la condición es verdadera.

Ejemplo:

if (condición) {

condición verdadera;

} else {

condición falsa;

}

- Estructura switch: La estructura SWITCH permite un funcionamiento equivalente a la estructura IF, pero con más posibilidades de ejecución de código según el valor de una variable. El valor de la variable se evalúa al principio de la estructura (por el switch). Luego el valor obtenido se compara con el valor especificado en el primer case (valor1). Si los dos valores son iguales, entonces el bloque de código 1 se ejecuta. Si no, el valor obtenido se compara con el valor del case siguiente; si hay correspondencia, el bloque de código se ejecuta y así sucesivamente hasta el último case. Si ningún valor concordante se encuentra en los diferentes case, entonces el bloque de código especificado en el default se ejecuta. Cada uno de los bloques debe terminarse con la instrucción break. El valor que hay que probar puede estar contenido en una variable, pero también puede ser el resultado de un cálculo.

Ejemplo:

switch (variable)

{

case valor1:

Bloque de código 1

case valor2:

Bloque de código 2

case valor3:

Bloque de código 3

default:

Bloque de código 4

}

En este caso, el cálculo sólo se efectúa una vez al principio del switch. El tipo del valor probado puede ser numérico o cadena de caracteres. El tipo de la variable probada debe corresponder por supuesto al tipo de los valores en los diferentes case.

1. Enumere las estructuras de repetición. Ejemplifique cada una de ellas y explique en que se diferencian.

- For: Esta estructura nos permite la posibilidad de generar un bucle que se repita una cantidad determinada de veces. En la cabecera se indica la variable de control, la condición de repetición y el incremento o decremento de la misma.

Ejemplo:

for (int i = 0 ; i < 10 ; i++){

Bloque de código

}

- While: Este tipo de iteración, se diferencia porque seguirá ejecutándose hasta que una condición determinada deje de cumplirse y no cuando se cumpla. Dentro de este tipo de bucles se pueden usar condiciones numéricas, pero también de tipo booleano, carácter o cadena de texto. La ejecución del ciclo while, no se realiza si la condición no se cumple al momento de ejecutar al código.

Ejemplo:

int i = 0;

while (i < 5) {

Console.WriteLine(i);

i++;

}

Ejemplo:

bool entra\_bucle = false;

while(entra\_bucle == true){

Bloque de código

}

- Do while: Esta estructura, es similar a la previa, con la particularidad de que la condición se comprobará al final de ejecución del código y no al principio, a diferencia de como ocurría tanto con los bucles For y While. El código se ejecutará al menos una vez, aunque la condición sea falsa. Resulta útil cuando se desea que la ejecución del código ocurra al menos una vez o cuando queremos que la condición de nuestra estructura de iteración se verifique al final del bloque de código.

Ejemplo:

do{

Console.WriteLine(i);

i++;

} while (i < 5)

- For each: Este tipo de bucle permite recorrer colecciones o arreglos completamente. No existe ningún tipo de condición, excepto que debemos completar toda la colección que hayamos querido ver. Este tipo de bucles, resulta útil para evitar las excepciones del tipo fuera del índice.

Ejemplo:

string[] arreglo = new string[5];

Foreach(string cad in arreglo){

Bloque de código

}

1. ¿Cómo se crea una estructura en el entorno visto en clase?

public struct Coordenadas

{

public Coordenadas(double x, double y)

{

X = x;

Y = y;

}

public double X { get; }

public double Y { get; }

public override string ToString() => $"({X}, {Y})";

}

1. ¿Qué es una propiedad?

Una propiedad es una especie de combinación entre lo que es una variable y un método, ya que se define parecido al segundo, pero se utiliza como el primero. Es un componente de la clase que combina un campo con los métodos que dan acceso a él.

Se caracterizan porque el acceso a ellas como si se tratara de un campo, permitir la asociación del código al momento de leer o escribir en el campo y utiliza datos como si los almacenara, pero no los almacena

1. ¿Qué es un método?

Un método es un bloque de código que contiene una serie de instrucciones. Un programa hace que se ejecuten las instrucciones al llamar al método y especificando los argumentos de método necesarios. En C#, todas las instrucciones ejecutadas se realizan en el contexto de un método. El método Main es el punto de entrada para cada aplicación de C#.

1. ¿Qué es un evento?

Un evento en C# es un mecanismo que permite la interacción entre el usuario y un control de usuario al ejecutar una acción lo que provoca la ejecución o no de una porción de código debido al cambio de estado del control de usuario.

1. ¿Cuál es el uso ideal de una variable de tipo Boolean?

El uso ideal de una variable de tipo Boolean es para realizar operaciones lógicas cuyos resultados pueden ser TRUE o FALSE. El tipo bool es el tipo de resultado de los operadores de comparación e igualdad. Una expresión bool puede ser una expresión condicional de control en las instrucciones if, do, while y for, así como en el operador condicional. Su valor predeterminado es false.

1. ¿Para qué se usa el método Parse cuando se lo utiliza por ejemplo como “DecimalParse”?

El método Parse se lo utiliza para la conversión de datos.

Ejemplo:

public static int Parse (string s);

1. ¿Cómo se declara un vector de 10 posiciones de tipo Decimal?

decimal [10] precios;

1. ¿Cuándo hablamos de boxing y unboxing a que nos referimos?

C# provee dos tipos de datos, por Valor y por Referencia. El proceso de conversión de tipos por Valor a tipos por Referencia es conocido como BOXING y la conversión de tipos de Referencia a tipos por Valor es conocido como UNBOXING.