**TP 2 PROGRAMACION I**

**Alumno: Bentos Leandro**

**Dni: 31989478 comision 1A**

**Parte 1**

1. ¿Qué característica posee la programación visual

la programación visual es la que se utiliza más de una dimensión para expresar la semántica.1​ Los lenguajes de programación visual permiten a los usuarios crear programas mediante la manipulación de elementos gráficos, en lugar de especificarlos exclusivamente de manera textual.

2. ¿Qué es un control de usuario?

Los controles de usuario son componentes creados por un desarrollador para ubicarlos en formularios Windows Forms o Web Forms. Cada control de usuario tiene su propio conjunto de propiedades, métodos y eventos que lo hacen adecuado para un determinado uso.

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

Hay tres categorías generales de controles en Visual Basic: Controles intrínseco, Controles ActiveX, y los Objetos insertables,

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

Es la manera en cómo organizar los archivos de tu proyecto, lo cual nos sirve para poder entender cómo está organizado todo nuestro código y a la vez poder ubicarlo de una manera sencilla

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

Las propiedades permiten que una clase exponga una manera pública de obtener y establecer valores, a la vez que se oculta el código de implementación o verificación.

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

El **cuadro de herramientas**es un elemento importante en el **IDE,**pues es allí donde están los controles que usaremos en nuestros proyectos. Por default, está localizado en el lado izquierdo. Si no lo ve, actívelo en el menú **Ver**-----à**Otras Ventanas**-------à**Cuadro de Herramientas**. O mediante **Control + Alt + X**

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

Una solución es un contenedor que se usa para organizar uno o más proyectos de código relacionados. Cuando se abre una solución en Visual Studio, esta carga automáticamente todos los proyectos que la solución contiene.

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

Los proyectos son los contenedores de segundo nivel en una aplicación. Se utilizan para organizar lógicamente, gestionar, generar y depurar los componentes de una aplicación.

9. ¿Qué es un formulario?

El formulario es el primer objeto o control que se visualiza en Visual Basic y constituye la pantalla o ventana sobre la que se colocan otros objetos o controles como etiquetas, controles de texto, botones, etc. y por supuesto el código necesario de nuestros programas, por lo tanto constituirán la Interfaz de usuario.

10. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Button.

**control Button:**

|  |  |  |
| --- | --- | --- |
| **propiedades** | **metodos** | **eventos** |
| Autosize | true | Autosizechanged |
| Blackcolor | control | Blackcolorchanged |
| Cursor | arrow | Cursorchanged |
| Flatstyle | popup | enter |
| Image | ninguno | keydown |
| Locked | False | layuot |
| Size | 75.23 | leave |
| Tapstop | true | mouseclick |
| Text | Button1 | move |
| Visible | true | paint |

11. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control CheckBox.

**control CheckBox:**

|  |  |  |
| --- | --- | --- |
| **propiedades** | **metodos** | **eventos** |
| backgroundimage | ninguno | Checkedchanged |
| Causesvalidation | true | click |
| Checkalign | middleleft | Clientsizechanged |
| Checked | false | Contextmenustripchanged |
| Checkedstate | unchecked | Controladded |
| Contextmenustrip | Ninguno | Controlremoved |
| cursor | Default | cursorchanged |
| Dock | None | Dockchanged |
| enabled | True | Dragdrop |
| forecolor | controltext | dragleave |

12. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Combobox.

**control Combobox:**

|  |  |  |
| --- | --- | --- |
| **propiedades** | **metodos** | **eventos** |
| Formatstring | Ninguno | Mouseup |
| Formattingenabled | True | Move |
| generatemember | True | Parentchanged |
| imemode | Nocontrol | Previewkeydown |
| integralheight | 20 | Querycontinuedrag |
| items | collection | Regionchanged |
| Location | 237.103 | Resize |
| Locked | False | Righttoleftchanged |
| minimumsize | 0.0 | Selectedindexchanged |
| Modifiers | private | sizechanged |

13. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control DateTimePicker.

**control DateTimePicker:**

|  |  |  |
| --- | --- | --- |
| **propiedades** | **metodos** | **eventos** |
| Name | Datetimepicker | Keyup |
| accessiblerole | Default | Layout |
| allowdrop | False | Leave |
| anchor | Top,left | Locationchanged |
| calendarfront | microsoftsanzserif | Marginchanged |
| calendarforecolor | Controltext | Mousecapturechanged |
| Calendarmonthblackground | window | Mousedown |
| calendartitlebackcolor | activecaption | mousenter |
| Cursor | default | Mousemove |
| Format | long | Mouseup |

14. Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Timer.

**control Timer:**

|  |  |  |
| --- | --- | --- |
| **propiedades** | **metodos** | **eventos** |
| name | Timer1 | tick |
| enabled | False |  |
| Generatemember | True |  |
| Interval | 100 |  |
| Modifiers | Private |  |
| tag | ninguno |  |

**Parte 2**

16. ¿Cómo se declara un procedimiento? Ejemplo.

Puede incluir un procedimiento entre una instrucción de declaración de inicio (Sub o Function) y una instrucción de declaración de finalización (End Sub o End Function). Todo el código del procedimiento se encuentra entre estas instrucciones.

Un procedimiento no puede contener otro procedimiento, por lo que sus instrucciones de inicio y finalización deben estar fuera de cualquier otro procedimiento.

Sub tellOperator(ByVal task As String)

Dim stamp As Date

stamp = TimeOfDay()

MsgBox("Starting " & task & " at " & CStr(stamp))

End Sub

17. ¿Cómo se declara una función? Ejemplo

la sintaxis para declarar un Function procedimiento es la siguiente:

Function FunctionName [(ParameterList)] As ReturnType

' The following statement immediately transfers control back

' to the calling code and returns the value of Expression.

Return Expression

End Function

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

Las funciones solo pueden tener parámetros de entrada, mientras que los procedimientos pueden tener parámetros de entrada/salida. Para una función es obligatorio tomar un parámetro de entrada, pero un procedimiento almacenado puede tomar 0 a n parámetros de entrada.

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

**pasar por valor:** Cuando se utiliza este mecanismo de paso, Visual Basic copia el valor del elemento de programación subyacente en una variable local en el procedimiento. El código de procedimiento no tiene ningún acceso al elemento subyacente en el código de llamada.

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

**pasar por referencia**: Cuando se utiliza este mecanismo de paso, Visual Basic proporciona al procedimiento una referencia directa al elemento de programación subyacente en el código de llamada.

**Parte 3**

21. Enumere las estructuras de decisión. Ejemplifique cada una de ellas y explique en que se diferencian.

Las construcciones **If...Then...Else** le permiten probar una o más condiciones y ejecutar una o más instrucciones en función de cada condición. Puede probar las condiciones y tomar medidas de las maneras siguientes:

Ejecutar una o más instrucciones si una condición es True.

Ejecutar una o más instrucciones si una condición es False.

Ejecutar algunas instrucciones si una condición es True y otras si es False.

Probar una condición adicional si una condición anterior es False.

La construcción **Select...Case** le permite evaluar una expresión una vez y ejecutar distintos conjuntos de instrucciones basados en valores posibles diferentes.

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

**for**: este tipo de bucle, es la **condición**, incluye una inicialización de una variable y un incremento o decremento de esa variable, en principio, no es necesario que incluya las 3 partes. Si no indicamos el incremento por defecto es 1.

**Module Module1**

**Sub Main()**

**For num As Integer = 1 To 5 Step 1**

**Console.WriteLine(num)**

**Next**

**Console.ReadLine()**

**End Sub**

**End Module**

while:indicamos una **condición** que se ejecutara hasta que la condición sea falsa, es importante que la variable que este en la **condición**, varíe para que no se produzca un **bucle infinito**

**While num <= 5**

**Console.WriteLine(num)**

**num += 1**

**End While**

**do-while**: esta estructura repetitiva, tiene algo especial y es que como mínimo las instrucciones dentro de la estructura repetitiva al menos una vez, mientras que el **while** sus instrucciones pueden ejecutarse 0 o N veces. Se sale del bucle cuando la **condición** del **while** no se cumple, si esto pasa empieza desde el **do.** Se utiliza para la validación de datos.

**Module Module1**

**Sub Main()**

**Dim num As Integer = 0**

**Do**

**Console.WriteLine("Introduce un número entre 0 y 10")**

**num = Console.ReadLine()**

**Loop While num >= 10 Or num < 0**

**Console.WriteLine("El numero introducido con Loop While es  " & num)**

**Do**

**Console.WriteLine("Introduce un número entre 0 y 10")**

**num = Console.ReadLine()**

**Loop Until num <= 10 Or num > 0**

**Console.WriteLine("El numero introducido con Loop Until es  " & num)**

**Console.ReadLine()**

**End Sub**

**End Module**

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

24. ¿Qué es una propiedad?

Una propiedad es un miembro que proporciona un mecanismo flexible para leer, escribir o calcular el valor de un campo privado. Las propiedades se pueden usar como si fueran miembros de datos públicos, pero en realidad son métodos especiales denominados *descriptores de acceso*.

25. ¿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.

26. ¿Qué es un evento?

Un evento es una señal que comunica a una aplicación que ha sucedido algo importante. Por ejemplo, cuando un usuario hace clic en un control en un formulario, el formulario puede provocar un evento Click y llamar a un procedimiento que controla el evento

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

Una *expresión booleana* es una expresión que se evalúa como un valor del [tipo de datos Boolean](https://docs.microsoft.com/es-es/dotnet/visual-basic/language-reference/data-types/boolean-data-type): True o False

28. ¿Para que se usa el método Parse cuando se lo utiliza por ejemplo como Decimal.Parse?

convierte la representación de cadena de un número en su equivalente [decimal](https://docs.microsoft.com/en-us/dotnet/api/system.decimal?view=netcore-3.1) .

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

decimal dim = 10;

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

El boxeo es el proceso de convertir un [tipo de valor](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/value-types) en el tipo objecto en cualquier tipo de interfaz implementado por este tipo de valor. Unboxing extrae el tipo de valor del objeto. El boxeo está implícito; el unboxing es explícito.