

Universidad Nacional de San Agustín

Facultad de Producción y Servicios

Escuela Profesional de Ingeniería de Sistemas



Introducción a la Computación – A

Grupo 8

Lenguaje de Programación:

Visual Basic

Integrantes:

Gabriela Martell Villanueva 20%

Gustavo Ordoño Poma 20%

Michael Hurtado Bejarano 20%

Leonardo Ccory Guerra 20%

Brandon Quinteros Oblitas 20%

Profesor:

Mg. Richart Escobedo Quispe

Fecha:

27 de julio del 2021

Arequipa – Perú

Epígrafe:

“ Para nuestras mentes humanas, las computadoras se comportan menos como rocas y árboles que como los humanos, por lo que inconscientemente los tratamos como personas ... En otras palabras, los humanos tienen instintos especiales que cuentan cómo comportarse con otros seres sintientes, y tan pronto como el objeto exhibe suficiente función cognitiva, dicen los instintos y reaccionamos como si estuviéramos interactuando con otro ser humano sensible ”.

Alan Cooper. (Diseñador de software y
programador)

Introducción:

A continuación les mostraremos y explicaremos el tema más a fondo sobre el lenguaje de programación Visual Basic, para comenzar nos remontaremos en su origen este lenguaje de programación es un dialecto de BASIC, con importantes agregados, su primera versión fue presentada en 1991 con el fin de simplificar la programación utilizando un medio ambiente de desarrollo, el diseñador de este software es Alan Cooper, también se dará a conocer su historia, paradigma, sus aplicaciones, las ventajas, desventajas del programa, y lo más importante a qué queremos llegar con este tema.

Visual Basic al igual que los otros lenguajes de programación como: Python, COBOL, "C", Pascal, ADA, Java, entre otros comparten el mismo fin ya que al ser herramientas de alto nivel permiten que los programadores puedan realizar su labor que es crear programas de computadoras, por supuesto es un concepto abstracto y variable según cada lenguaje de programación.

Lo interesante de este tema es que tiene algo muy particular ya que siendo un lenguaje orientado a eventos creo que en su tiempo este paradigma intentaba organizar las funcionalidades de una aplicación de forma más sencilla que la que se tenía en su momento que el de la programación estructurada.

De ahí viene el nombre "Visual", teniendo la facultad de gestionar el comportamiento de los componentes de la aplicación de forma visual programando eventos en particular de forma separada.

Esperando que este trabajo de investigación formativa sea de su agrado, le recomendamos que lo lea detenidamente para una mejor comprensión del tema.

ÍNDICE

1. Introducción al lenguaje	
1.1 Historia.....	4
1.2 Paradigmas.....	4
2. Aplicaciones del lenguaje de programación	
2.1 Ventajas.....	5
2.2 Desventajas.....	5
3. Curva de aprendizaje.....	6
4. Ejemplos con respecto al lenguaje de programación.....	7
5. Ejercicios con respecto al lenguaje de programación.....	8
6. Aplicación.....	12
7. Recomendaciones.....	14
8. Conclusiones.....	14
9. Referencias.....	15

Introducción al Lenguaje de Programación:

Historia:

Visual Basic es un lenguaje de programación desarrollado por los estadounidenses John Kemeny y Tomas Kurtz en el Dartmouth College. Su versión original fue hecha en 1964, teniendo una gran popularidad gracias sobre todo a dos implementaciones, Tiny BASIC y Microsoft BASIC, que ayudaron a este lenguaje a ser la primera lengua franca de los microordenadores. Otras importantes implementaciones han sido C BASIC (BASIC Compilado), Integer y Applesoft BASIC (para el Apple II), GW-BASIC (para computadoras personales), Turbo BASIC (de Borland) y Microsoft QuickBASIC. El lenguaje ha tenido cambios en el paso de los años pues fue creado con el objetivo de servir como lenguaje para estas personas que deseaban aprender por primera vez el arte de la programación, y luego fue teniendo modificaciones, esto hasta que en 1978 se estableció el Basic estándar.

En 1990, el hecho de hacer programas de ordenador requería un alto conocimiento para un lenguaje de programación, como C/C++, lo cual necesitaba mucho esfuerzo para crear cualquier programa de MS-DOS o de Windows, o también utilizar Microsoft Quick BASIC para la creación de programas en MS-DOS, pero llegaban ser de mala calidad. Por ello, el equipo de Microsoft decidió inventar un concepto de programación sencilla para todos. Y el resultado fue Microsoft Visual Basic, que permite generar programas con interfaz gráfica programando de forma sencilla, aunque haciendo programas potentes y estables. En 1991 fue presentado Microsoft Visual Basic 1.0. Esta versión era considerada escasa, pero era capaz usar su interfaz gráfica para producir ventanas, botones, cuadros de texto entre varias aplicaciones y vincularlos entre sí. En años siguientes se estrenaron próximas versiones de Microsoft Visual Basic, y en 1992 se creó una versión para MS-DOS, que producía interfaces gráficas de texto y que utilizaba el mismo estilo que la versión hecha para Windows.

Paradigma:

Contiene una filosofía que se asemeja al paradigma de programación de diseño gráfico mezclado con un paradigma orientado a eventos

Para propósitos de comparación, este lenguaje utiliza el paradigma orientado a objetos y tiene grandes semejanzas con Java.

Aplicaciones:

VBA permite a usuarios y programadores ampliar la funcionalidad de programas de la suite Microsoft Office. VBA para Aplicaciones es un subconjunto casi completo de Visual Basic 5.0 y 6.0.

Microsoft VBA viene integrado en aplicaciones de Microsoft Office, como Outlook, Word, Excel, Access y PowerPoint. Prácticamente cualquier cosa que se pueda idear en Visual Basic 5.0 o 6.0 se puede llevar a cabo asimismo internamente de un apunte de Office, con

la sola limitación que el artículo final no se puede compilar separado del apunte, hoja o base de datos en que fue creado; es decir, se redime en una macro (o más aceptablemente súper macro). Esta macro puede ocupar o distribuirse con nada más datar la reseña, estreno o pulvínulo de datos.

Ventajas:

Entre las ventajas del Lenguaje de Programación Visual Basic se encuentran las siguientes:

- ❖ Posee una curva de aprendizaje muy rápida, es decir, es muy fácil aprender a programar con Visual Basic, he de ahí su popularidad.
- ❖ Es uno de los lenguajes de uso más extendido, por lo que resulta fácil encontrar información, documentación y fuentes para los proyectos.
- ❖ Permite usar con facilidad la plataforma de los sistemas Windows, dado que tiene acceso prácticamente total a la API de Windows, incluidas librerías actuales.
- ❖ Adhiere soporte para ejecutar scripts en las aplicaciones con VBScript o JScript, gracias a Windows Script Control 7.
- ❖ Fácilmente extensible mediante librerías DLL y componentes ActiveX de otros lenguajes.
- ❖ Tiene acceso a la API multimedia de DirectX, también está disponible, de forma no oficial, un componente para trabajar con OpenGL 1.1.
- ❖ Si bien es cierto que permite desarrollar grandes y complejas aplicaciones, también provee un entorno adecuado para realizar pequeños prototipos rápidos.
- ❖ Programación con la gran mayoría de bases de datos, entre ellas SQL Server y Oracle.

(Todo sobre el lenguaje Visual Basic .NET, s. f.)

Desventajas:

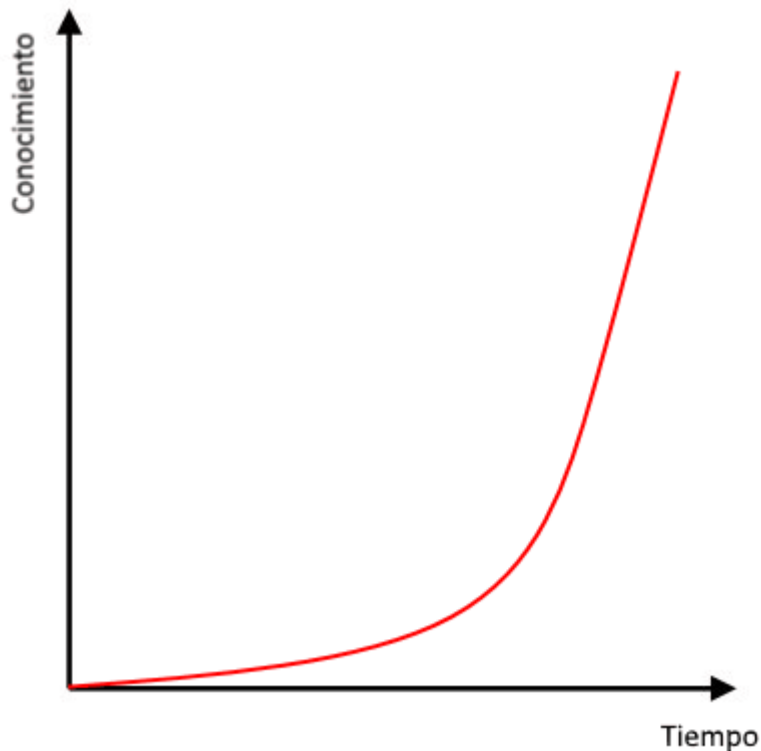
Entre las desventajas del Lenguaje de Programación Visual Basic se encuentran las siguientes:

- ❖ Problema de versionado asociado con varias librerías runtime DLL's, conocido como DLL Hell.
- ❖ Pobre soporte para programación orientada a objetos.
- ❖ Incapacidad para crear aplicaciones multihilo, sin tener que recurrir a llamadas de la API de Windows.
- ❖ Dependencia de complejas y frágiles entradas de registro COM.
- ❖ Solo Microsoft puede decidir la evolución de este lenguaje.
- ❖ Solo genera ejecutables para Windows.
- ❖ Los ejecutables generados son relativamente lentos.
- ❖ La sintaxis es demasiado inflexible.
- ❖ Solo existe un compilador, llamado igual que el lenguaje.

(ventajas y desventajas de programar en VISUAL BASIC, 2016)

Curva de aprendizaje:

Aprender Visual Basic bien supone conocer primero una serie de fundamentos, comunes a toda la plataforma, y que te servirán para cualquier lenguaje que uses en el futuro. La curva de aprendizaje se parece a algo así:



Interpretación: Es decir, al principio se va lento, pero luego, de repente, se avanza muy rápido. El motivo es que son muchos conceptos y técnicas de base que es necesario dominar, pero una vez lo consigas, especializarse y aprender cualquiera de las múltiples bibliotecas especializadas es mucho más fácil. Es tentador empezar la casa por el tejado y aprender antes alguna de éstas, pero a medio plazo es un error. Sin comprender bien antes los fundamentos, construiremos una casa sin cimientos, y ante la primera dificultad se puede caer (*Qué es la plataforma .NET y cuáles son sus principales partes* - campusMVP.es, s. f.).

Ejemplos:

1. *Hola Mundo*

The screenshot shows a Visual Studio window titled 'HOLAMUNDO' with a 'Program' dropdown. The code editor displays the following code:

```

1  Module Program
2      Sub Main(args As String())
3          Console.WriteLine("hola mundo")
4      End Sub
5  End Module

```

Below the code editor, the 'Consola de depuración de Microsoft Visual Studio' (Microsoft Visual Studio Debug Console) is open, showing the output:

```

hola mundo

```

2. *Variables*

The screenshot shows a Visual Studio window titled 'VARIABLES' with a 'Program' dropdown. The code editor displays the following code:

```

1  Module Program
2      Sub Main(args As String())
3          Dim a As Integer = 2147483647 '#máximo (4 bytes)
4          Dim b As String = "visual"
5          Dim c As Char = "a" 'Para un caracter
6          Dim d As Long = 9223372036854775807 '# máximo (8 bytes)
7          Dim e As Single = 34028235 '#máximo (4 bytes)
8          Dim f As Double = 1.7976931348623157 '#máximo (8 bytes)
9          Dim g As Boolean = True
10         Console.WriteLine(a)
11         Console.WriteLine(b)
12         Console.WriteLine(c)
13         Console.WriteLine(d)
14         Console.WriteLine(e)
15         Console.WriteLine(f)
16         Console.WriteLine(g)
17     End Sub
18 End Module

```

Below the code editor, the 'Consola de depuración de Micro' (Microsoft Visual Studio Debug Console) is open, showing the output of the program:

```

2147483647
visual
a
9223372036854775807
34028236
1,7976931348623157
True

```

3. Estructuras de control

```

1  Module Program
    0 referencias
2  Sub Main(args As String())
3      For num As Integer = 1 To 10
4          Console.WriteLine("Visual Basic")
5      Next
6  End Sub
7  End Module

```

to

Consola de depuración de

```

Visual Basic
Visual Basic
Visual Basic
Visual Basic
Visual Basic
Visual Basic
Visual Basic
Visual Basic
Visual Basic
Visual Basic
Visual Basic

```

Ejercicios:

a) *Básicos:*

EjercicioBasico1

```

1  Module Program
    0 referencias
2  Sub Main(args As String())
3      Dim a, b, c As Integer
4      Console.WriteLine("SUMA")
5      Console.WriteLine("Primer numero")
6      a = Console.ReadLine
7      Console.WriteLine("segundo numero")
8      b = Console.ReadLine
9      c = a + b
10     Console.WriteLine("Resultado: " & c)
11 End Sub
12 End Module

```

Consola de depuración de Microsoft Visual Studio

```

SUMA
Primer numero
7
segundo numero
5
Resultado: 12

```


EjercicioBasico2

```

1  Module Program
2      Sub Main(args As String())
3          Dim num1, num2, rpt As Integer
4          Console.WriteLine("PRODUCTO")
5          Console.WriteLine("Primer numero")
6          num1 = Console.ReadLine
7          Console.WriteLine("segundo numero")
8          num2 = Console.ReadLine
9          rpt = num1 * num2
10         Console.WriteLine("Resultado: " & rpt)
11     End Sub
12 End Module

```

Consola de depuración de Microsoft Visual Studio

```

PRODUCTO
Primer numero
12
segundo numero
6
Resultado: 72

```

b) Intermedios:

EjercicioIntermedio1

```

1  Module Program
2      Sub Main(args As String())
3          Dim num1, num As Integer
4          Console.WriteLine("NUMEROS PARES")
5          Console.WriteLine("Desde 1 hasta")
6          num1 = Console.ReadLine
7          Console.WriteLine("")
8          For num = 1 To num1
9              If (num Mod 2 = 0) Then
10                 Console.WriteLine(num)
11             End If
12         Next
13     End Sub
14 End Module

```

Consola de depuración de Microsoft Visual Studio

```

NUMEROS PARES
Desde 1 hasta
20

2
4
6
8
10
12
14
16
18
20

```

EjercicioIntermedio2

```

1  Module Program
    0 referencias
2  Sub Main(args As String())
3      Dim mtp, num, a As Integer
4      Console.WriteLine("MULTIPLoS DE:")
5      mtp = Console.ReadLine
6      Console.WriteLine("Desde 1 hasta:")
7      num = Console.ReadLine
8      Console.WriteLine("")
9      For a = 1 To num
10         If (a Mod mtp = 0) Then
11             Console.WriteLine(a)
12         End If
13     Next
14 End Sub
15 End Module

```

Consola de depuración de Microsoft Visual Studio

```

MULTIPLoS DE:
3
Desde 1 hasta:
13

3
6
9
12

```

c) Avanzados

EjercicioAvanzado1

```

1  Module Program
    0 referencias
2  Sub Main(args As String())
3      Dim longitud, cantidad As Integer
4      Dim palabra As String
5      Console.WriteLine("Escribir palabra")
6      palabra = Console.ReadLine
7      longitud = Len(palabra)
8      For i = 1 To longitud
9          If Mid(palabra, i, 1) = "a" Then
10              cantidad = cantidad + 1
11          End If
12      Next
13      Console.WriteLine("Tiene " & cantidad & " a's")
14 End Sub
15 End Module

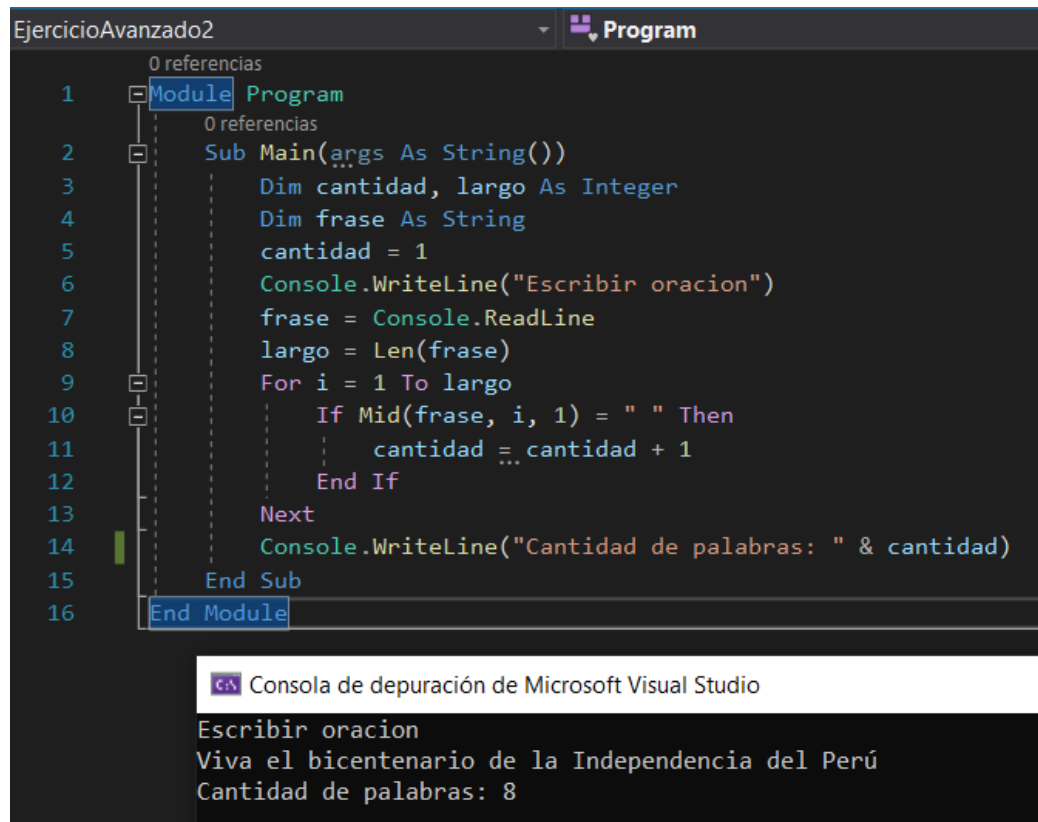
```

Consola de depuración de Microsoft Visual Studio

```

Escribir palabra
Parangaricutirimicuario
Tiene 4 a's

```



The image shows a screenshot of the Visual Studio IDE. The top window, titled 'EjercicioAvanzado2', displays a VB.NET program. The code defines a module named 'Program' containing a 'Main' subroutine. The 'Main' subroutine takes 'args' as a string array, declares 'cantidad' and 'largo' as integers, and 'frase' as a string. It initializes 'cantidad' to 1, prompts the user to 'Escribir oracion', reads the input into 'frase', and calculates 'largo' as the length of 'frase'. A 'For' loop iterates from 1 to 'largo', and an 'If' statement checks if each character is a space. If it is, 'cantidad' is incremented. After the loop, it prints 'Cantidad de palabras: ' followed by 'cantidad'. The bottom window, titled 'Consola de depuración de Microsoft Visual Studio', shows the program's output: 'Escribir oracion', 'Viva el bicentenario de la Independencia del Perú', and 'Cantidad de palabras: 8'.

```
1  Module Program
2      Sub Main(args As String())
3          Dim cantidad, largo As Integer
4          Dim frase As String
5          cantidad = 1
6          Console.WriteLine("Escribir oracion")
7          frase = Console.ReadLine
8          largo = Len(frase)
9          For i = 1 To largo
10             If Mid(frase, i, 1) = " " Then
11                 cantidad = cantidad + 1
12             End If
13         Next
14         Console.WriteLine("Cantidad de palabras: " & cantidad)
15     End Sub
16 End Module
```

Consola de depuración de Microsoft Visual Studio

Escribir oracion
Viva el bicentenario de la Independencia del Perú
Cantidad de palabras: 8

Aplicación:

Consigna: Usted debe definir un rango de números, la máquina extraerá un número aleatorio de este rango y usted debe adivinar cuál número es.

Link: <https://youtu.be/I7V9MtaNSz0>

```

Form1.vb  Form1.vb [Design]
VB3 Aplicacion  Form1  num3

1  Public Class Form1
2      Dim num3 As Integer 'num3 es un entero
3
4      'Al hacer click en el boton 2(Generador del aleatorio)
5
6      Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
7          num3 = Int((TextBox1.Text - TextBox2.Text) * Rnd() + TextBox2.Text) 'Numero random
8          Label3.Text = 0 '(Intentos)
9          Label7.Text = "?????????" '(Proximidad al numero)
10         Label6.Visible = False 'el label6 no es visible
11     End Sub
12
13     'Al hacer click en boton 1(Adivinar el aleatorio)
14
15     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
16         If (TextBox3.Text = num3) Then 'compara el numero ingresado con el random
17             Label6.Visible = True 'El texto6 se hace visible
18             Label7.Text = "Exacto" 'El texto7 cambia a exacto
19             Label6.Text = "ACERTÓ" 'El texto6 cambia a acertó
20             Label3.Text = Label3.Text + 1 'el numero aumenta con cada intento
21         ElseIf (TextBox3.Text < num3) Then 'Si el numero ingresado es menor que el random
22             Label7.Text = "Muy Pequeño"
23             Label3.Text = Label3.Text + 1 'el numero aumenta con cada intento
24         ElseIf (TextBox3.Text > num3) Then 'Si el numero ingresado es mayor que el random
25             Label7.Text = "Muy Grande"
26             Label3.Text = Label3.Text + 1 'el numero aumenta con cada intento
27         End If
28     End Sub
29 End Class
  
```

Form1

ADIVINA EL NUMERO

Rango 1(menor) Rango 2(mayor)

10 20

OK

12

Intentos

1

ENTER

Muy Pequeño

Form1

ADIVINA EL NUMERO

Rango 1(menor) Rango 2(mayor)

10 20

OK

17

Intentos

2

ENTER

Muy Grande

Form1

ADIVINA EL NUMERO

Rango 1(menor) Rango 2(mayor)

10 20

OK

14

Intentos

5

ENTER

Exacto

ACERTÓ

Recomendaciones:

- Si vamos a trabajar con bastantes componentes en la interfaz de nuestro programa lo recomendable sería modificar el nombre de los atributos por ejemplo los “Label”, “TextBox”, “Button” que vienen con nombre por defecto de manera ascendente de tal forma que así nos permite tener más claridad.
- A medida que se va avanzando en la programación, es mejor ir comprobando la efectividad de cada parte por separado, porque si se programa todo junto de una vez y al momento de probarlo salta un error, A pesar de que nos indique en qué línea está el error, es muy probable que tengamos que cambiar todo el código ya hecho

Conclusiones:

- En el presente trabajo de investigación formativa hemos apreciado que a pesar de la diferencia de otros lenguajes de programación este maneja la misma lógica e inclusive algunos comandos son iguales e investigando descubrimos cuales son y de qué manera aplicarlos línea por línea dando resultado así la ejecución limpia y sin errores de nuestro programa
- Llevando así la idea de que a partir del lenguaje que conozcamos podemos hacer una especie de traducción para mover nuestro programa a otros lenguajes y aprendiendo más de ellos.

Video de la Exposición: <https://www.youtube.com/watch?v=l7V9MtaNSz0>

Link del GitHub: <https://github.com/GustavoEdu/IC>

Referencias:

- [1] (2017, Abr 06) *VBMath.Randomize Method* [Online]. Recuperado de:
<https://docs.microsoft.com/en-us/dotnet/api/microsoft.visualbasic.vbmath.rnd?view=net-5.0>
- [2] (2015, Feb 18) *Bucles en visual basic.net* [Online]. Recuperado de :
<https://www.discoduroderoer.es/bucles-en-visual-basic-net/>
- [3] (2010, Abr 30) *Tipos de variables en visual basic* [Online]. Recuperado de:
https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=140:tipos-de-variables-en-visual-basic-integer-single-double-string-object-boolean-etc-ejem-plos-cu00308a&catid=37&Itemid=61
- [4] (2016, Jul 20). *Ventajas y desventajas de programar en Visual Basic* [Online].
Recuperado de:
https://adictoalcodigo.blogspot.com/2016/07/ventajas-y-desventajas-de-programar-en_19.html
- [5] (2017, May 16) *Qué es la plataforma .NET y cuáles son sus principales partes* [Online].
Recuperado de:
<https://www.campusmvp.es/recursos/post/que-es-la-plataforma-net-y-cuales-son-sus-principales-partes.aspx>
- [6] (2020, Nov 12) *Todo sobre el lenguaje Visual Basic .NET* (s. f.). Lenguajes de programación. Recuperado de: <https://lenguajesdeprogramacion.net/visual-basic/>