

PRÁCTICAS CODE

1. Crea un clon enlazado da máquina virtual de Windows 10. O nome da máquina debe ser as iniciais do voso nome e apelidos_WIN10. P.e: JMG_WIN10



General

Nombre: ADLF_WIN10
Sistema operativo: Windows 10 (64-bit)
Ubicación de archivo de preferencias: D:\damd104\MV\ADLF_WIN10

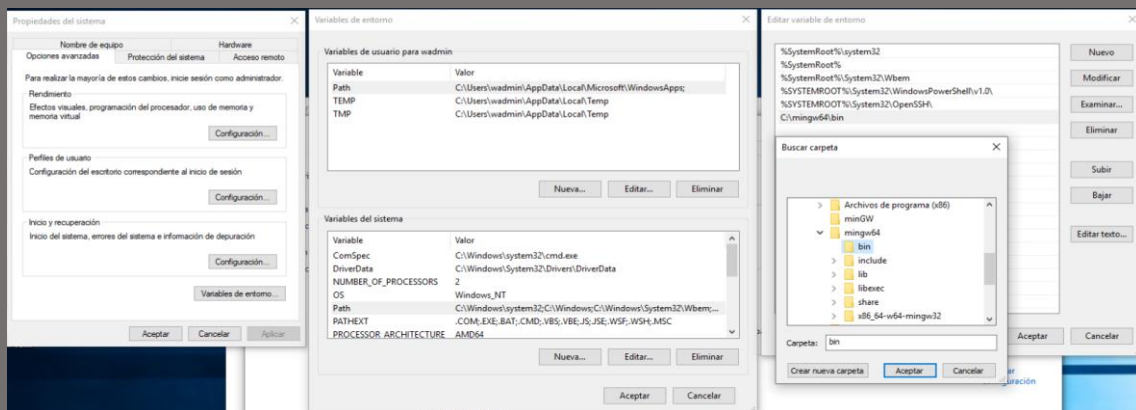
2. Na máquina de Windows 10 instala e configura a ruta de sistema para poder traballar coas seguintes linguaxes de programación

C/C++

Primero descargamos MinGW para poder compilar nuestros programas



Ahora lo extraemos en C: en una carpeta para el y lo añadimos a la ruta del sistema



Para comprobar que funciona abrimos un CMD y vemos que si hacemos “gcc – versión” nos debería salir el programa.

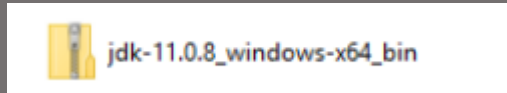
```
Símbolo del sistema
Microsoft Windows [Versión 10.0.17763.107]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\wadmin>gcc --version
gcc (GCC) 9.1.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

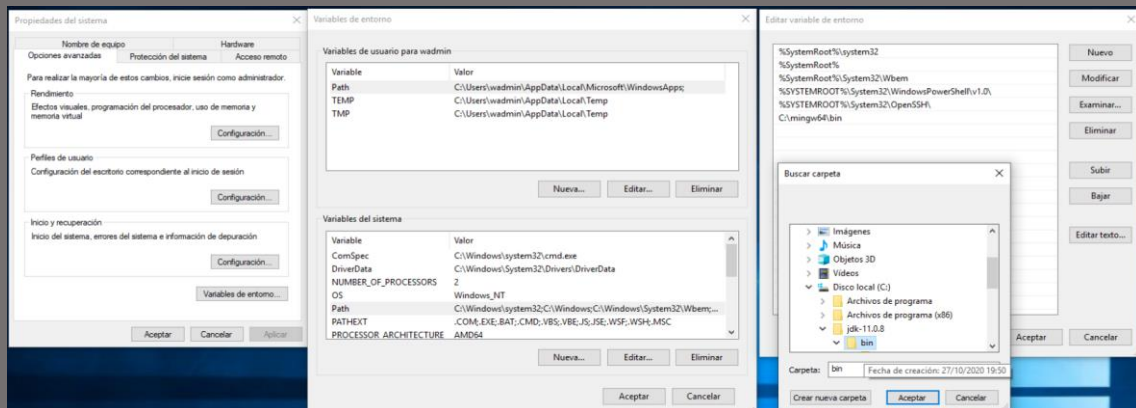
C:\Users\wadmin>
```

Java

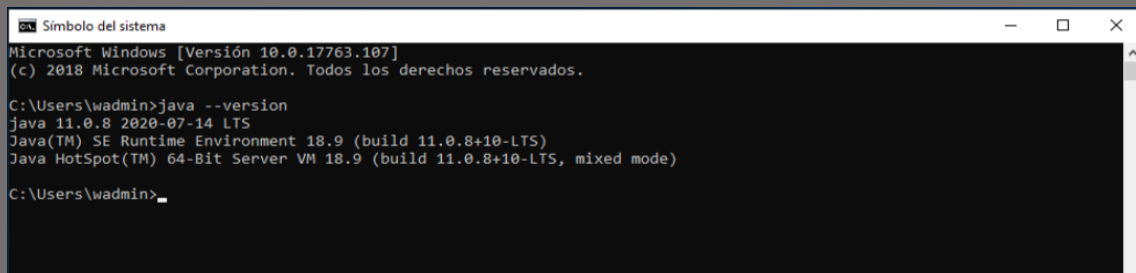
Primero nos descargamos el JDK 11 para poder compilar los programas en JAVA.



Ahora lo extraemos en su ruta correspondiente en C: y lo vamos a añadir al path del sistema

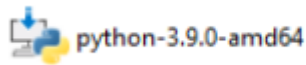


Comprobamos desde el CMD que este configurado correctamente

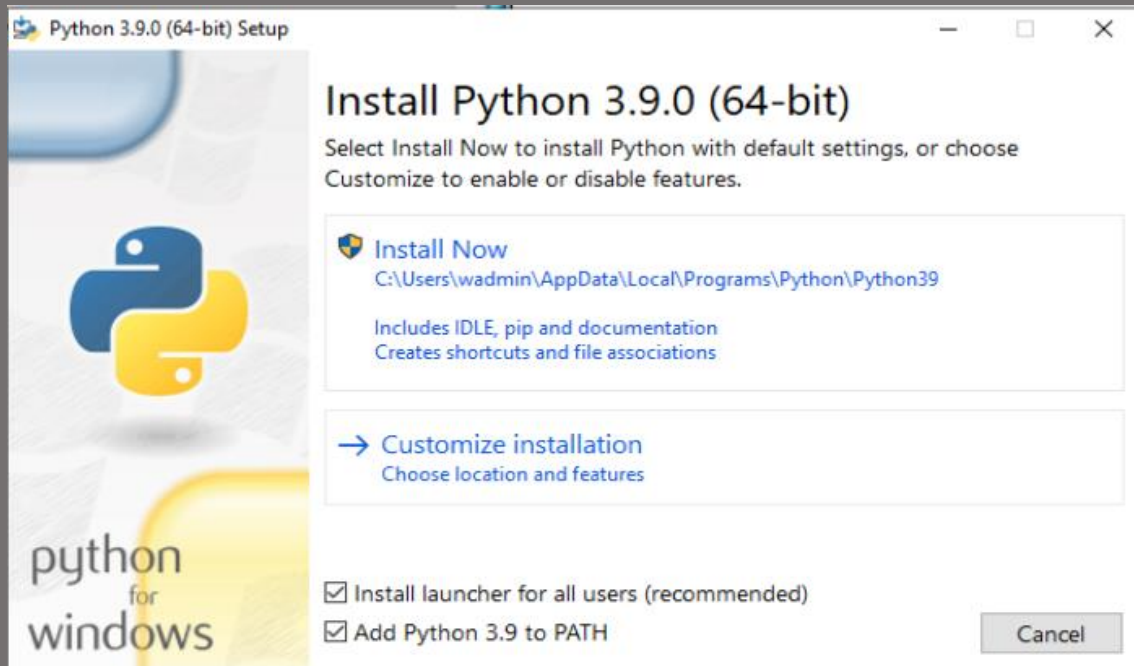


Python

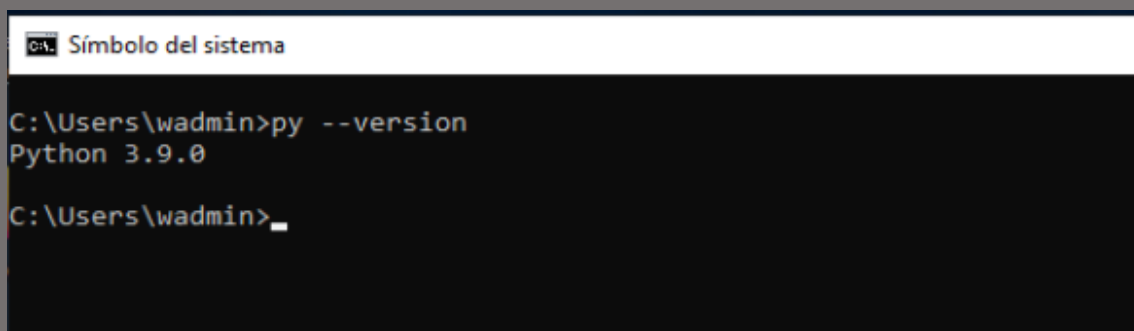
Primero nos descargamos Python para poder instalarlo



Lo ejecutamos y seguimos los pasos. Importante elegir que nos añada automáticamente a la path del sistema el programa

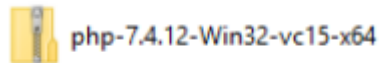


Comprobamos desde el CMD que funciona

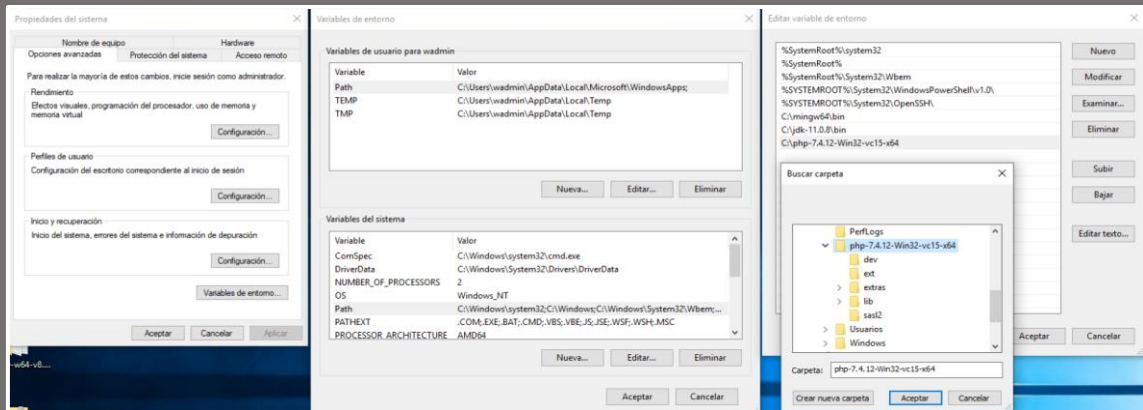


PHP

Para poder trabajar con PHP nos descargamos PHP 7. Una vez descargado lo vamos a preparar para poder trabajar con el



Ahora lo extraemos en su ruta y lo añadimos a la path del sistema



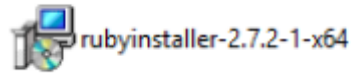
Desde el CMD comprobamos que este todo correctamente funcional

```
C:\Users\wadmin>php -v
PHP 7.4.11 (cli) (built: Sep 29 2020 13:18:06) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies

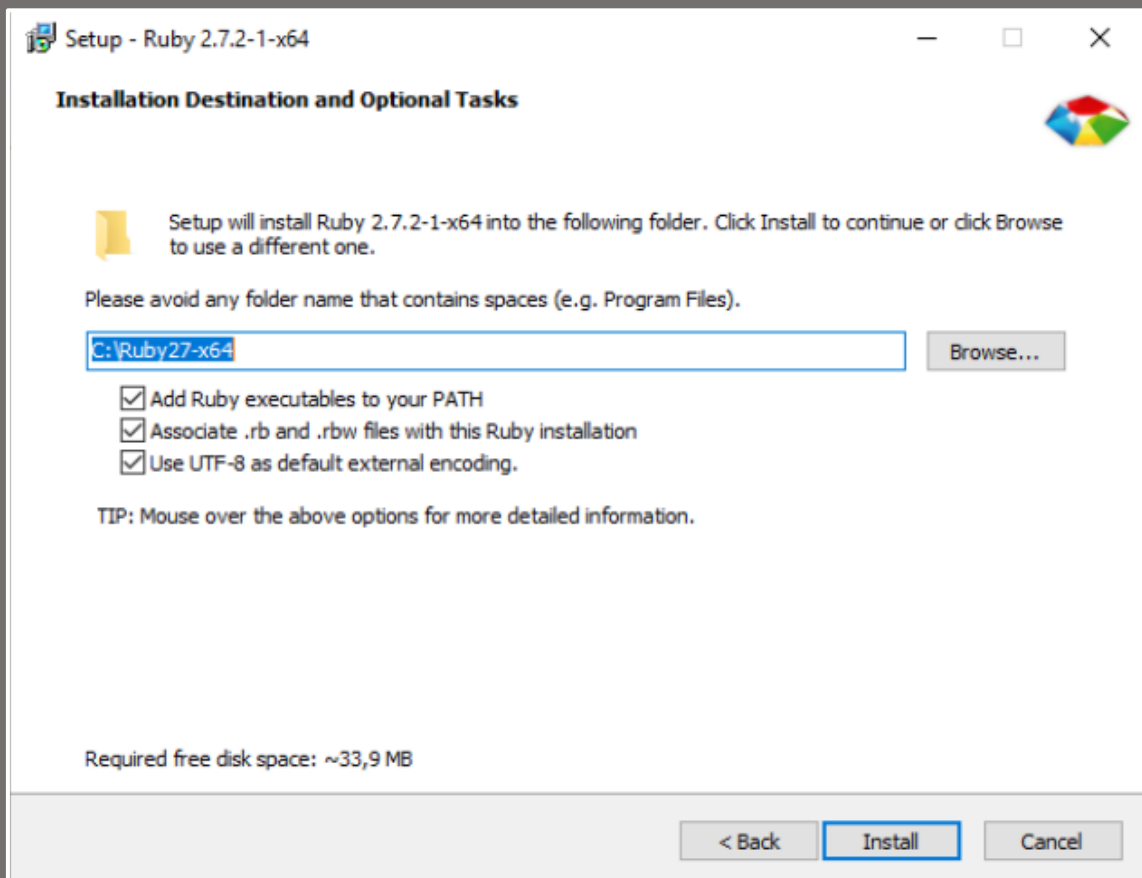
C:\Users\wadmin>
```

Ruby

Nos descargamos el instalador de Ruby para integrarlo en nuestro sistema y así poder trabajar con él



Ponemos todos los parámetros necesarios que creamos convenientes antes de instalar



Comprobamos en el CMD que está bien configurado

```
C:\Users\wadmin>ruby --version
ruby 2.7.2p137 (2020-10-01 revision 5445e04352) [x64-mingw32]
C:\Users\wadmin>_
```

Ahora crearemos un programa en cada uno de los lenguajes que nos permita leer dos números y realizar su división y nos de el resultado

C++

Creamos el código del programa en C++

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main(){
6     int n1,n2, division = 0;
7     cout<<"Escriba un numero: "; cin>>n1;
8     cout<<"Escriba un numero: "; cin>>n2;
9
10    division = n1 / n2;
11
12    cout<<"La division es: "<<division<<endl;
13
14    return 0;
15 }
```

Lo comprobamos

```
c:\dev\c++>g++ programa.cpp
c:\dev\c++>a.exe
Escriba un numero: 10
Escriba un numero: 2
La division es: 5

c:\dev\c++>a.exe
Escriba un numero: 5
Escriba un numero: 0

c:\dev\c++>a.exe
Escriba un numero: 5
Escriba un numero: 2
La division es: 2
```

C

Creamos un programa en C

Creamos el código

```
1  #include <stdio.h>
2  int main()
3  {
4      float a, b;
5      float division;
6      printf("Introduce el primer valor: ");
7      scanf("%f", &a);
8      printf("Introduce el segundo valor: ");
9      scanf("%f", &b);
10     division = (a / b);
11     printf("La division es: %f", division);
12     return 0;
13 }
```

Comprobamos el código

```
c:\dev\c>gcc programa.c
c:\dev\c>a
Introduce el primer valor: 10
Introduce el segundo valor: 5
La division es: 2.000000
c:\dev\c>a
Introduce el primer valor: 10
Introduce el segundo valor: 2
La division es: 5.000000
c:\dev\c>_
```


Java

Creamos el código de java

```
1  import java.util.Scanner;
2  public class division {
3      public static void main(String[] args) {
4
5          Scanner teclado= new Scanner(System.in);
6          dividir();
7      }
8
9      public static void dividir(){
10         Scanner teclado = new Scanner(System.in);
11         float num1,num2,divide;
12
13         System.out.println("Ingrese Primer Número:");
14         num1= teclado.nextInt();
15         System.out.println("Ingrese Segundo Número:");
16         num2= teclado.nextInt();
17         divide= num1/num2;
18
19         System.out.println("La División es:" + divide);
20     }
21 }
```

Y lo probamos ejecutándolo desde el CMD

```
C:\dev\java>javac division.java
C:\dev\java>java division
Ingrese Primer Numero:
10
Ingrese Segundo Numero:
5
La Division es:2.0
C:\dev\java>java division
Ingrese Primer Numero:
10
Ingrese Segundo Numero:
2
La Division es:5.0
C:\dev\java>_
```

Python

Primero creamos el código

```
1 print ("Introduce un numero")
2 primerNumero = int(input())
3 print ("Introduce otro numero")
4 segundoNumero = int(input())
5 print ("La division es")
6 print (primerNumero / segundoNumero)
```

Despues lo probamos desde el CMD

```
C:\dev\python>py division.py
Introduce un numero
10
Introduce otro numero
5
La division es
2.0

C:\dev\python>py division.py
Introduce un numero
10
Introduce otro numero
2
La division es
5.0

C:\dev\python>_
```

PHP

Primero creamos el código

```
1 <?php
2
3 echo "Introduzca un numero";
4 $numero1 = stream_get_line(STDIN, 1024, PHP_EOL);
5 echo "Introduzca otro numero";
6 $numero2 = stream_get_line(STDIN, 1024, PHP_EOL);
7 echo "La division es: " . $numero1 . "/" . $numero2 . "=" . ($numero1/$numero2);
8 ?>
```

Y después lo probamos desde el CMD

```
C:\dev\php>php division.php
Introduzca un numero10
Introduzca otro numero5
La division es: 10/5=2
C:\dev\php>php division.php
Introduzca un numero10
Introduzca otro numero2
La division es: 10/2=5
C:\dev\php>php division.php
Introduzca un numero20
Introduzca otro numero3
La division es: 20/3=6.6666666666667
C:\dev\php>_
```

Ruby

Primero creamos el código

```
1 puts "Introduce el primer numero"
2 n1 = gets.to_i
3 puts "Introduce el segundo numero"
4 n2 = gets.to_i
5 puts (n1 / n2)
```

Y después lo probamos desde el CMD

```
C:\dev\ruby>ruby division.rb
Introduce el primer numero
10
Introduce el segundo numero
5
2

C:\dev\ruby>ruby division.rb
Introduce el primer numero
10
Introduce el segundo numero
2
5

C:\dev\ruby>_
```