

Introducción a la programación con Python



Taller de programación #1

Descripción

Este documento contiene información relacionada a las actividades de prácticas del primer taller del curso Introducción a la Programación con Python. En este taller empezaremos con una pequeña revisión de las herramientas que usaremos en este curso. A continuación mostraremos cómo correr un programa básico en Python para luego continuar la explicación de cómo recibir e imprimir información. Por favor note que el taller se ha ejecutado en una computadora con el sistema operativo Windows 10.

1. Revision

1.1 Command Prompt (Simbolo del sistema)

Permite al usuario interactuar directamente con el sistema operativo, los comandos son en forma de línea de textos, antiguamente se usaba para controlar el sistema puesto que no existía mouse, control de manejo de ventanas o GUI.

1.2 Editor de código fuente

Es un programa que permite crear o modificar archivos de forma digital. Pueden decidir usar el editor de código fuente de su preferencia para este y los próximos talleres. En caso de no haber usado un editor de código anteriormente les recomendamos Visual Studio Code. Si desean descargarlo pueden hacerlo usando el siguiente enlace: <https://code.visualstudio.com/>.

Nota: Visual Studio Code contiene una terminal integrada, Puedes usar esta terminal si lo deseas una vez hayas instalado Python.

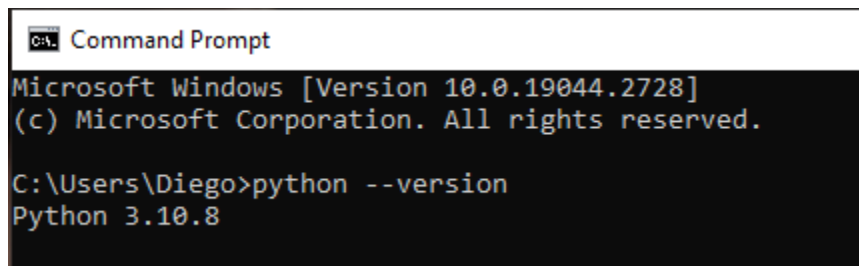
1.3 Entorno virtual

Los entornos virtuales son herramientas que permiten a los desarrolladores crear un espacio aislado en el que se pueden instalar las dependencias (a menudo bibliotecas y paquetes) requeridas para un proyecto específico sin interferir con otras instalaciones de software.

2. Python

2.1 Confirmar que Python está instalado

Recomendamos que confirmes si tienes una versión de Python mayor a 3, abriendo una terminal o símbolo de sistema e ingresando el comando "python --version" o si hay múltiples versiones de Python instaladas "python3 --version". Si tienes una versión mayor a la 3.x, puedes ir directamente a la sección 3.

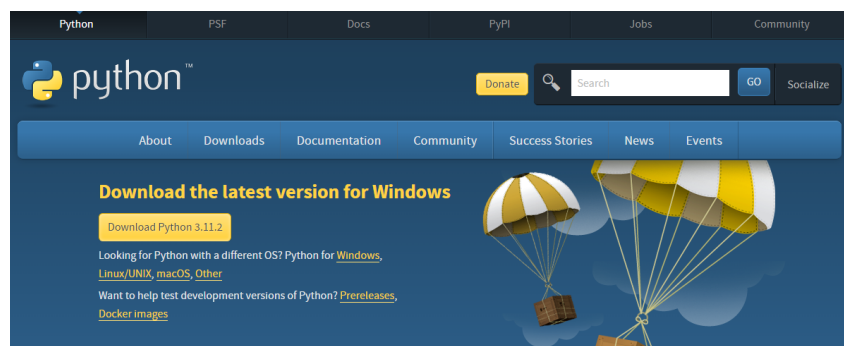


```
Command Prompt
Microsoft Windows [Version 10.0.19044.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Diego>python --version
Python 3.10.8
```

2.2 Instalar Python

Para obtener la última versión de Python ingresa a <https://www.python.org/downloads/> y hacer click en el botón de descargar.



Dependiendo del sistema operativo, deberás descargar diferentes archivos. En el caso que uses un sistema diferente a Windows. Por Favor, envíanos un mensaje para ayudarte con el proceso de instalación.

Para Windows: Una vez que el ejecutable ha sido descargado, es posible iniciar el proceso de instalación haciendo doble clic izquierdo en el ejecutable. Durante la instalación, asegúrate de marcar la casilla "Add python.exe to PATH" y seleccionar "Install Now" para continuar con la instalación.



Una vez el instalador finalice, comprueba que Python está instalado ejecutando los comandos mencionados en la section 2.1

Nota: Es posible que necesites abrir una nueva terminal para comprobar que Python ha sido instalado, o que reinicies tu computador.

2.3 Organización de código

Con el fin de organizar nuestro código, crearemos una carpeta en el escritorio. Para esto ejecute en su símbolo de sistema > cd c:\Users\(\nombre_de_usuario)\Desktop > mkdir curso_de_python > cd curso_de_python

2.3 Configurar un entorno virtual (Opcional)

Crear un entorno virtual es considerado una buena práctica en el desarrollo de software, ya que evita conflictos de dependencias y problemas de compatibilidad. Python nos permite crear estos entornos usando el siguiente comando: **python -m venv intro_python**. Dentro de curso_de_python" Al hacer esto podrás observar una nueva carpeta en el directorio con el nombre de intro_python. Para iniciar el entorno desde la terminal en la carpeta curso_de_python usa el siguiente comando:

Windows: intro_python\Scripts\activate.bat

Unix or MacO: source intro_python/bin/activate

```
C:\Windows\system32\cmd.exe

(intro_python) c:\Users\Diego>cd c:\Users\Diego\Desktop

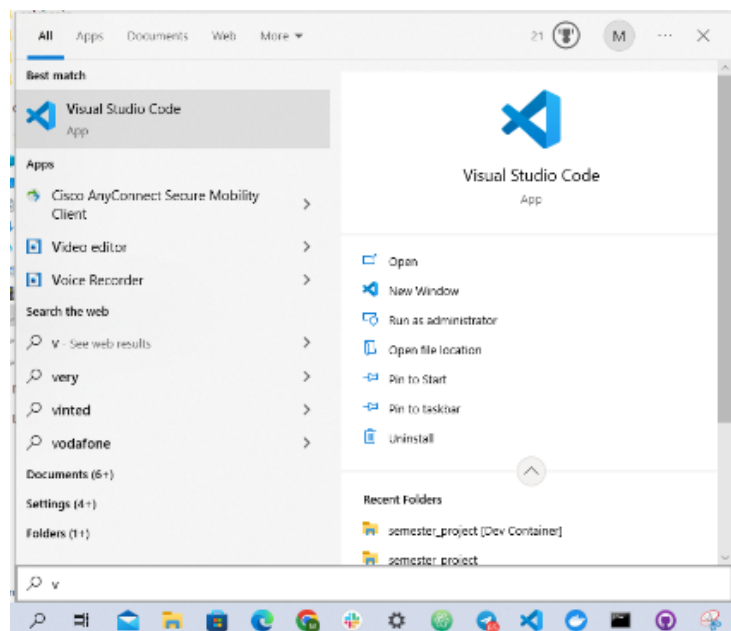
(intro_python) c:\Users\Diego\Desktop>mkdir curso_de_python

(intro_python) c:\Users\Diego\Desktop>cd curso_de_python

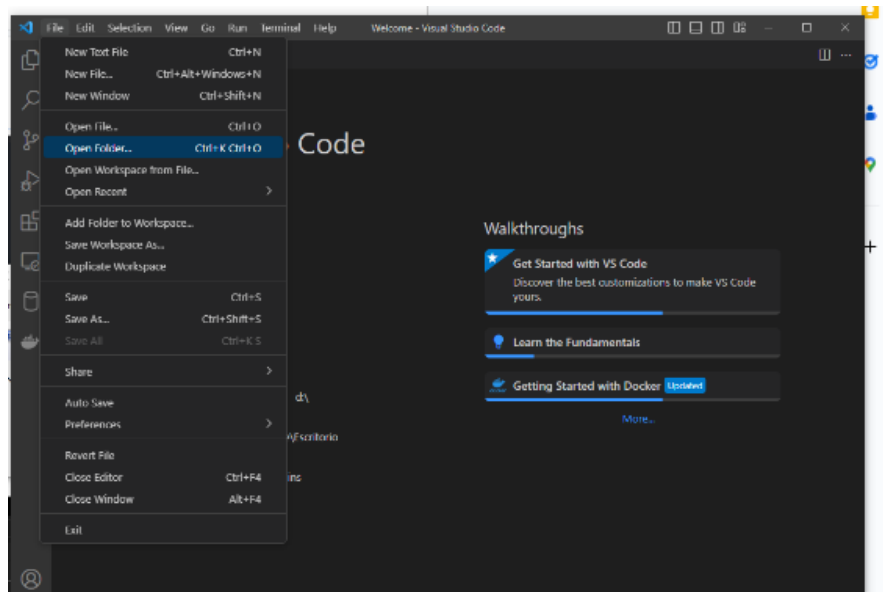
(intro_python) c:\Users\Diego\Desktop\curso_de_python>
```

3. Mi primer programa

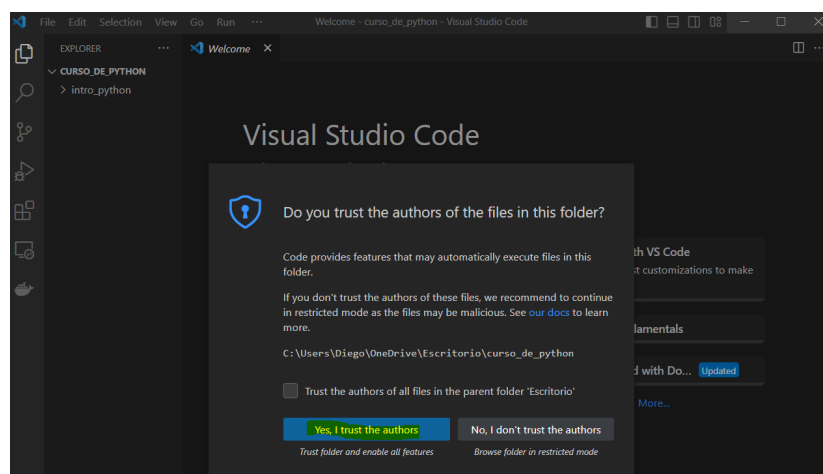
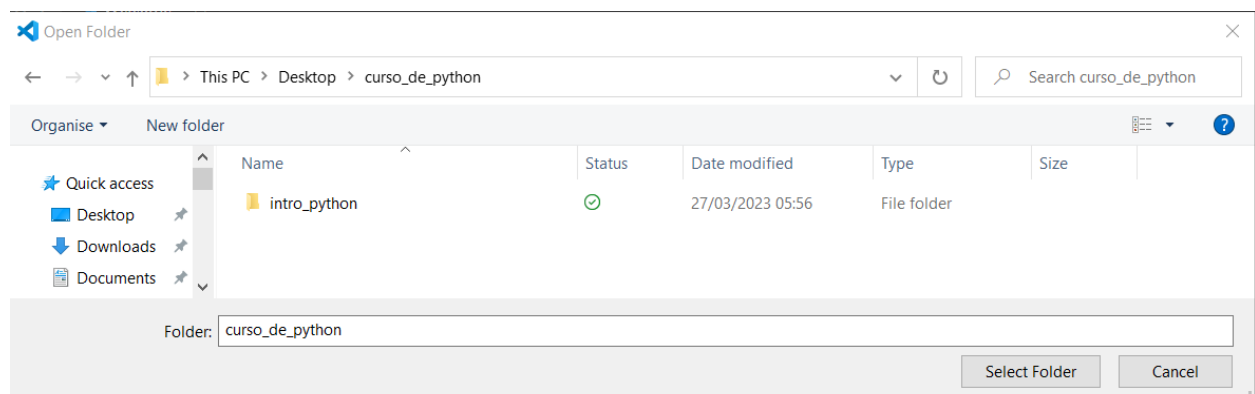
Empezaremos nuestro tutorial abriendo el editor de código fuente “Visual Studio Code” que nos permitirá crear nuestro primer programa. Una vez lo hayas descargado e instalado lo puedes abrir como cualquier otra aplicación de Windows.



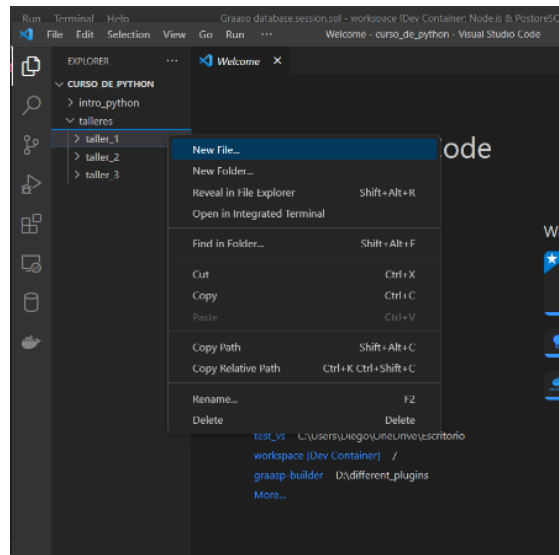
Una vez en el editor, podemos abrir la carpeta ‘curso_de_python’ con el material del curso haciendo click en File y luego Open Folder como se muestra en la siguiente imagen:



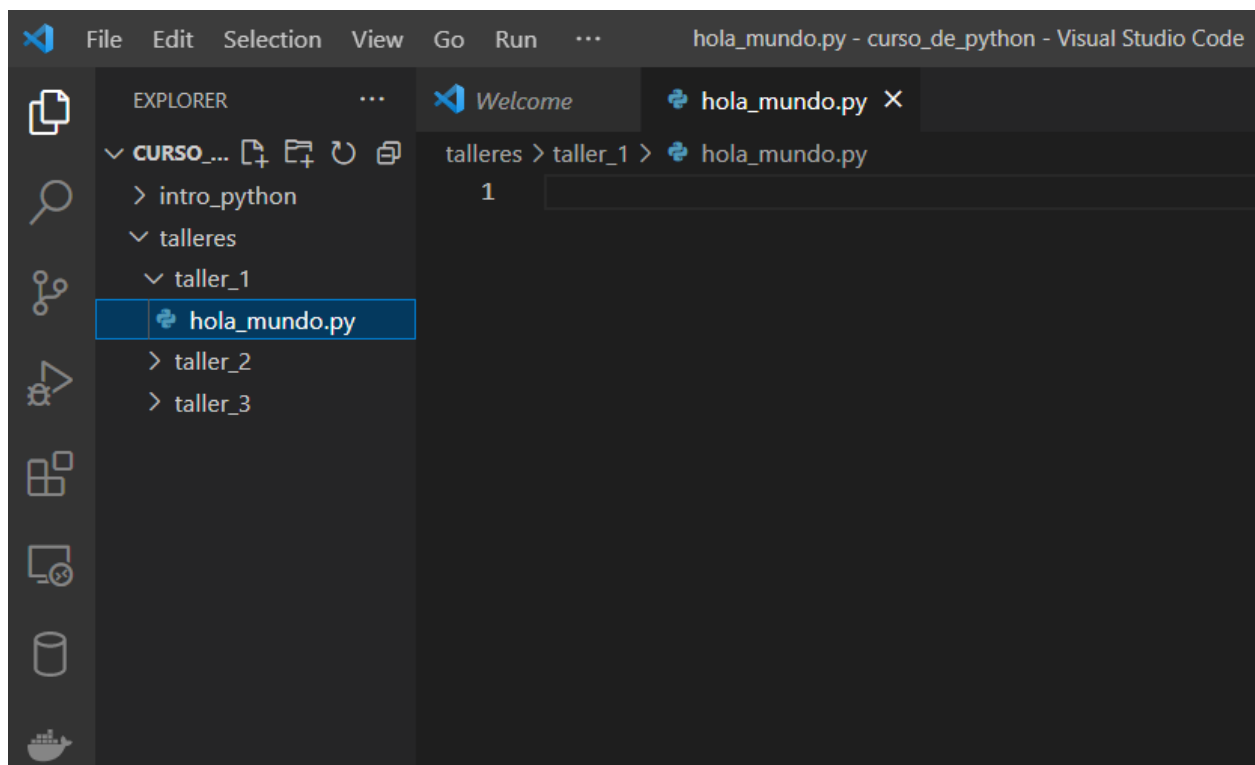
Una vez nos hayamos dirigido a la ruta de nuestra carpeta debemos hacer click en ‘seleccionar carpeta’ y aceptamos en confiar los autores de la carpeta. Esto nos permitirá visualizar y explorar sus contenidos como se muestra en las siguientes imágenes:



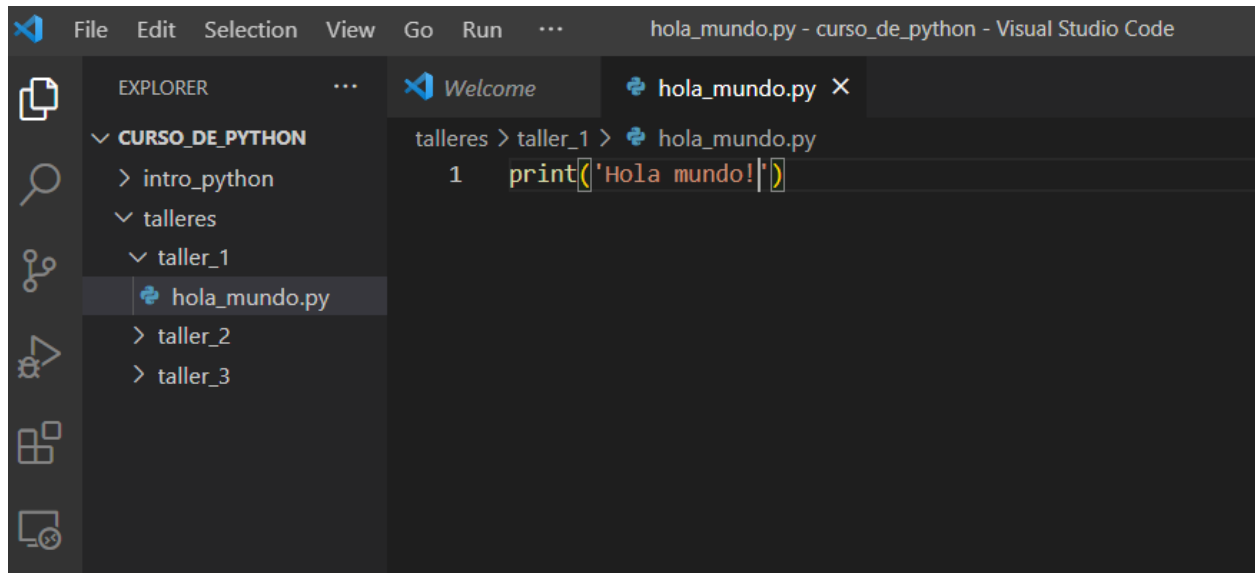
Se puede añadir un nuevo archivo en una carpeta en específico al hacer click derecho sobre esta y seguido seleccionamos 'New File'. En la siguiente imagen se muestra como creamos una nuevo archivo en la carpeta 'taller_1':



A continuación Visual Studio Code nos preguntara como nombrar al archivo, en nuestro caso lo llamaremos 'hola_mundo.py'. Recuerda seleccionar la terminación '.py' ya que esto permite que el editor identifique nuestro archivo como un archivo de Python.

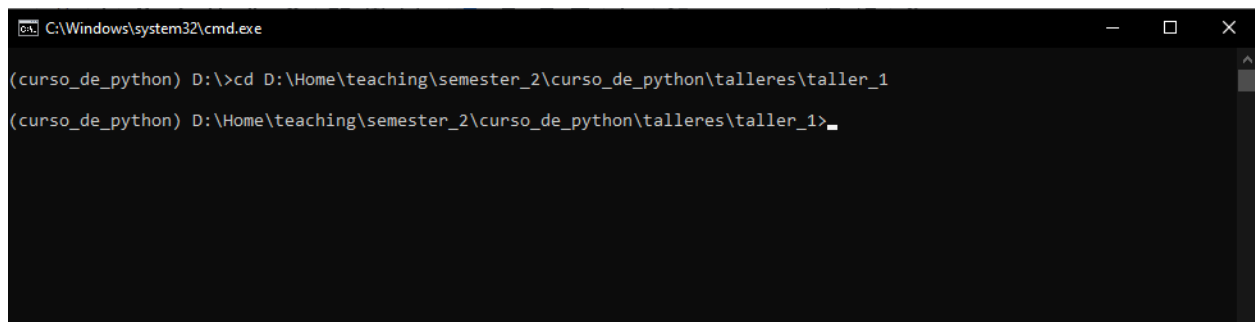


Esto hará que Python cree y abra el nuevo archivo donde podemos escribir sintaxis de Python. Para nuestro primer programa haremos que la consola muestra un mensaje de bienvenida, para esto usaremos una palabra especial en python llamada print (print nos permite enviar parámetros e imprimirlos línea por línea)



Para guardar los cambios hechos al documento puedes simplemente ocupar el atajo 'CTRL + S' o dirigirte a la pestaña 'File' y luego selecciona 'Save'.

Para poder correr el archivo que hemos escrito debemos primero empezar nuestro entorno de desarrollo siguiendo los pasos explicados en la parte 2.1.2 de este taller. Una vez que se abra la terminal debemos dirigirnos a la ruta donde están nuestros archivos. Para eso podemos ocupar el comando 'cd' que nos permite movernos a través de directorios.



Desde el símbolo de sistema, llamaremos al interpretador de Python usando 'python hola_mundo.py'

```
C:\Windows\system32\cmd.exe

(curso_de_python) D:\Home\teaching\semester_2\curso_de_python\talleres\taller_1>python hola_mundo.py
Hola mundo!

(curso_de_python) D:\Home\teaching\semester_2\curso_de_python\talleres\taller_1>
```

4. Saludando desde el simbolo de sistema

Recibir datos desde el símbolo del sistema es posible en python, gracias a la ayuda de **input()**, de esta manera obtendremos un dato de tipo **string**, la cual puede ser guardada en un variable, finalmente modificaremos **print** para imprimir esta.

```
nombre = input("Ingresa tu nombre: ")
print("Hola ", nombre)
```

```
C:\Windows\system32\cmd.exe

(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Ingresa tu nombre: Amongus
Hola Amongus

(intro_python) c:\Users\Diego\Desktop\curso_de_python>_
```

5. Calculadora

Ahora te toca a ti, usando lo que has aprendido en este día desarrolla una calculadora que obtenga el siguiente resultado en el símbolo del sistema:

```
(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Ingrese operador: +
Ingrese primer numero: 10
Ingrese segundo numero: 5
15.0

(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Ingrese operador: -
Ingrese primer numero: 10
Ingrese segundo numero: 5
5.0

(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Ingrese operador: *
Ingrese primer numero: 10
Ingrese segundo numero: 5
50.0

(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Ingrese operador: /
Ingrese primer numero: 10
Ingrese segundo numero: 5
2.0

(intro_python) c:\Users\Diego\Desktop\curso_de_python>python amongus.py
Ingrese operador: LOL
Ingrese primer numero: 10
Ingrese segundo numero: 10
Operador invalido
```

Por ahora, no te preocupes de aquellos casos donde el usuario ingresa datos que no son válidos para la calculadora.

6. Número par o impar

Desarrolla un programa que permita a un usuario ingresar un número y detectar si este es par o impar **SIN** usar el operador de división /. El resultado esperado es el siguiente:

```
(intro_python) C:\Users\Diego\Desktop\sh>python3 script.py
Ingrese su numero: 1
impar

(intro_python) C:\Users\Diego\Desktop\sh>python3 script.py
Ingrese su numero: 7
impar

(intro_python) C:\Users\Diego\Desktop\sh>python3 script.py
Ingrese su numero: 2
par

(intro_python) C:\Users\Diego\Desktop\sh>python3 script.py
Ingrese su numero: 4
par
```

Ahora trata de hacerlo usando operadores a nivel de bits.

7. ¡Ayudanos con el código!

El siguiente extracto de código contiene errores. ¿Puedes encontrarlos y arreglarlos?

```
# correo_recibido = input("usuario ingrese su mensaje: ")
correo_recibido = 'Pablo se encuentra a 10km de distancia y quiere
conocererte.'

es_spam = "quiere conocer" in correo_recibido
#
if no es_spam:
    if "importante" in correo_recibido:
        print "Este es un correo importante"
        print(correo_recibido)
    else:
        print("Este es un correo no deseado")
```

Marcel Mauricio Moran Calderon
marcel.morancalderon@epfl.ch

Alexis Rodriguez
alexis.rodrigueznunez@epfl.ch