Ingeniería en Computación
Ingeniería en Tecnologías de la Información

Sistemas Basados en el Conocimiento
Grupo de Investigación

How To:

Desarrollo en Python

Primeros pasos con PyCharm

René Rolando Elizalde Solano
Departamento de Ciencias de la Computación y Electrónica
Sistemas Basados en el Conocimiento
rrelizalde@utpl.edu.ec
@reroes

¿Qué es Python?

- Lenguaje de programación potente y fácil de aprender.
- Contiene estructuras de datos de alto nivel eficientes.
- Sintaxis elegante y la tipificación dinámica.
- Desarrollo rápido de aplicaciones en muchas áreas en la mayoría de las plataformas.

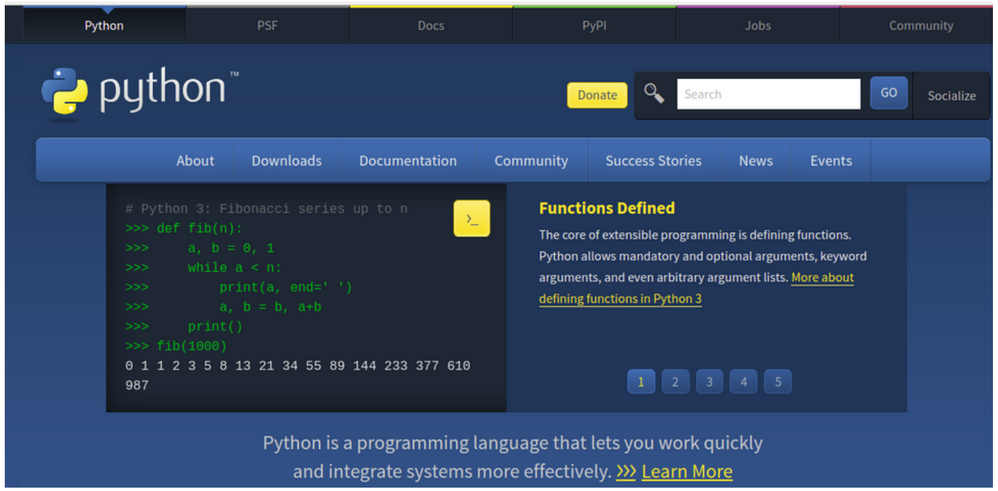
Tomado:
https://docs.python.org/3/tutorial/index.html

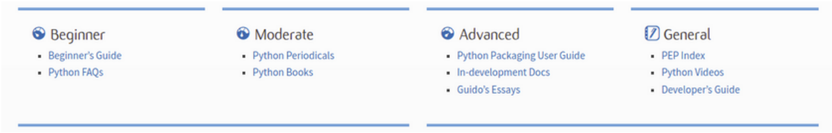
Guido van Rossum

python™

Documentación - Comunidad

- https://docs.python.org/3/tutorial/index.html





python™

Instalación

Download Python

The official home of the Python Programming Language

Python.org / /static/humans.txt

https://www.python.org/downloads/

python™

Instalación

Download Python

The official home of the Python Programming Language

Python.org / /static/humans.txt

https://www.python.org/downloads/

Versiones para trabajar

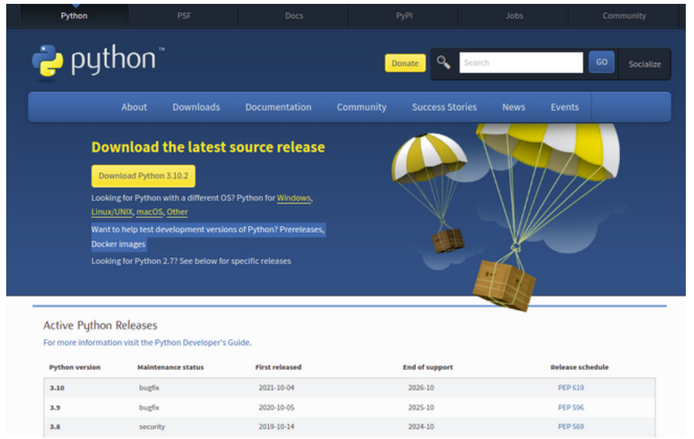
- Python 3.6
- Python 3.7
- Python 3.8
- Python 3.9
- Python 3.10

Instalación

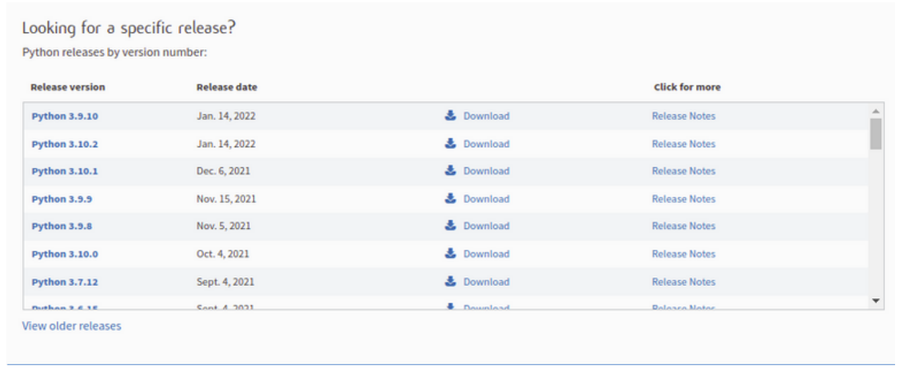
Download Python

The official home of the Python Programming Language

Python.org / /static/humans.txt



https://www.python.org/downloads/



python™

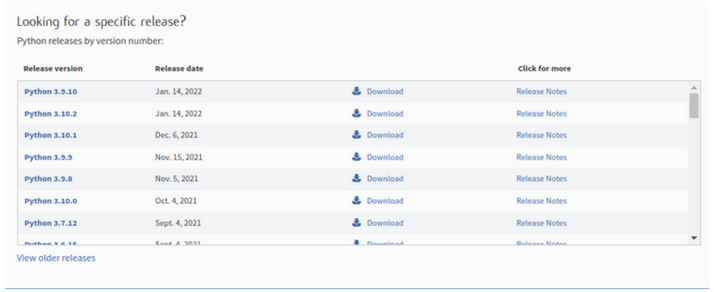
Instalación

Download Python

The official home of the Python Programming Language

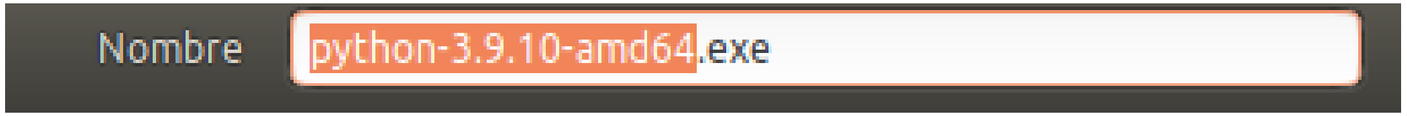
Python.org / /static/humans.txt

https://www.python.org/downloads/



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		1440acb71471e2394befdb30b1a958d1	25800844	SIG
XZ compressed source tarball	Source release		e754c4b2276750fd5b4785a1b443683a	19154136	SIG
macOS 64-bit Intel-only installer	macOS	for macOS 10.9 and later, deprecated	2714cb9e6241cf7e2f9022714a55d27a	30395760	SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	c2393ab11a423d817501b8566ab5da9f	38217233	SIG
Windows embeddable package (32-bit)	Windows		c1d2af96d9f3564f57f35cfc3c1006eb	7671509	SIG
Windows embeddable package (64-bit)	Windows		b8e8bfba8e56edcd654d15e3bdc2e29a	8509821	SIG
Windows help file	Windows		784020441c1a25289483d3d8771a8215	9284044	SIG
Windows installer (32-bit)	Windows		457d648dc8a71b6bc32da30a7805c55b	27767040	SIG
Windows installer (64-bit)	Windows	Recommended	747ac35ae667f4ec1ee3b001e9b7dbc6	28909456	SIG



python™

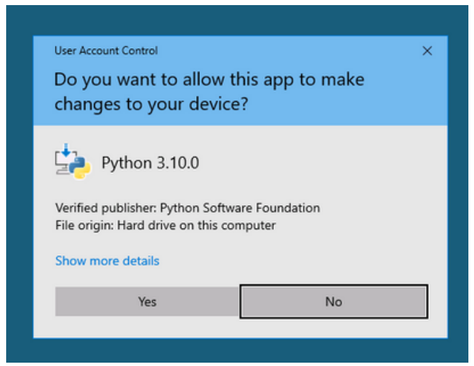
Entornos de Programación

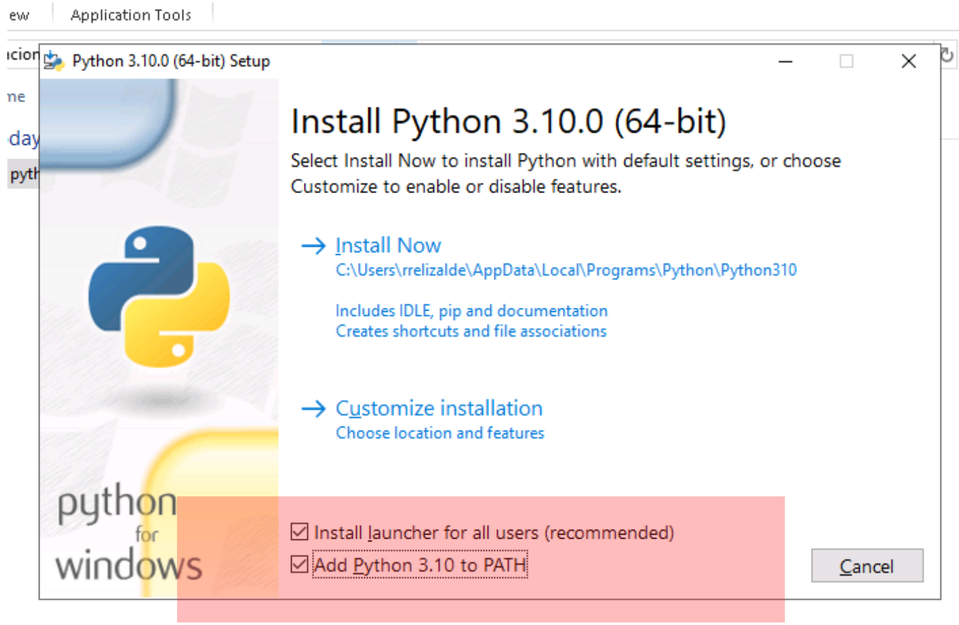
Instalación

Download Python

The official home of the Python Programming Language

Python.org / /static/humans.txt

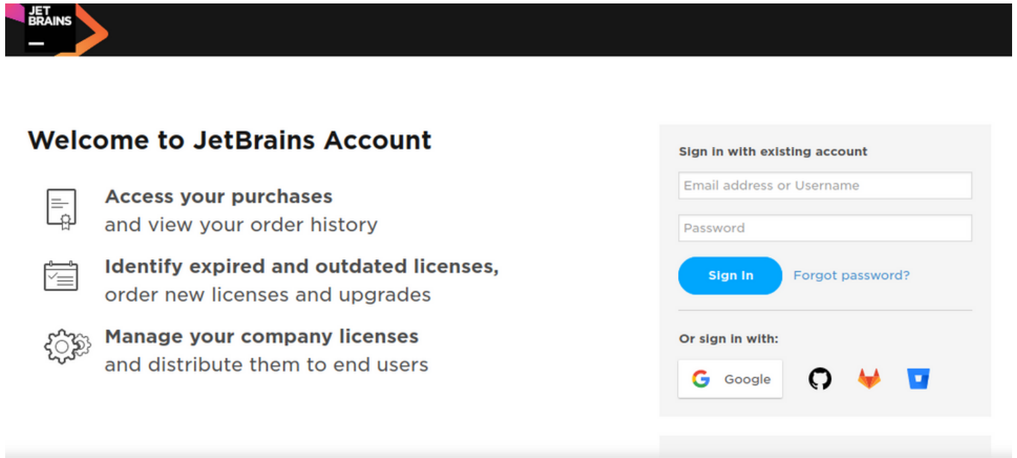




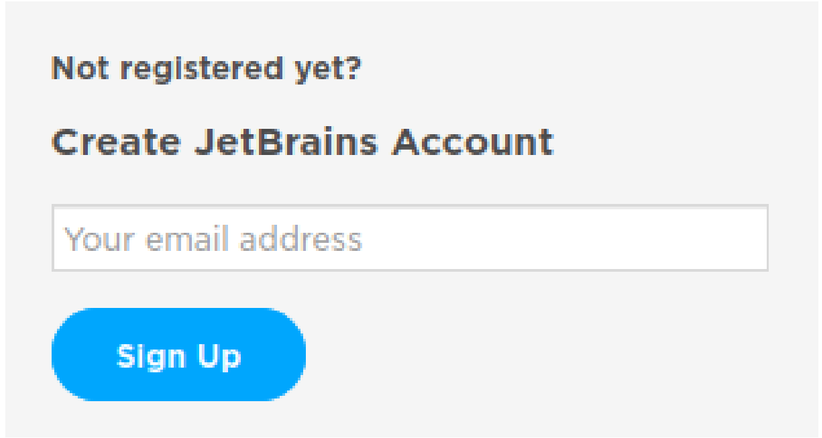
https://www.python.org/downloads/

Instalación - PyCharm

1. Ingresar a: https://account.jetbrains.com/login

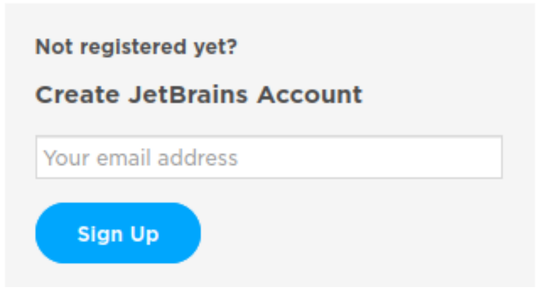


2. Registrarse



Instalación - PyCharm

3. Ingresar el correo en Crear Cuenta



Thank you for registering your JetBrains Account!

Please follow the instructions we just emailed to you.

Instalación - PyCharm

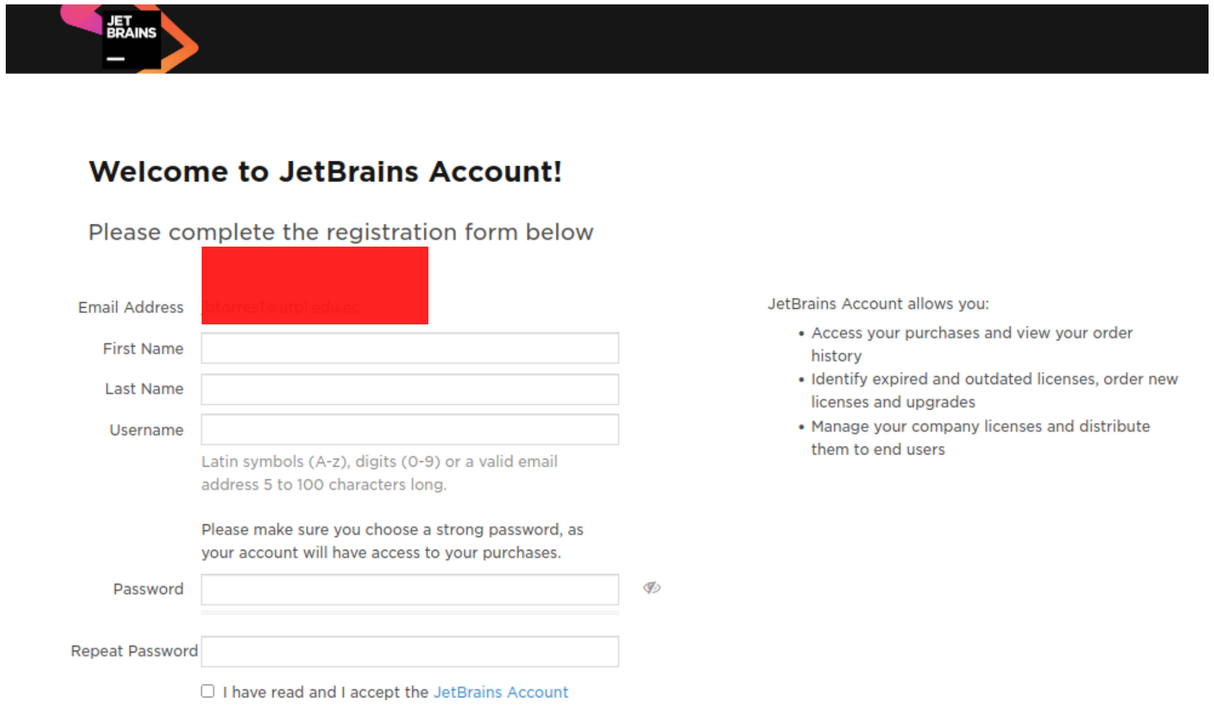
3. Esperar correo institucional

Hello!

Thank you for creating your JetBrains Account.

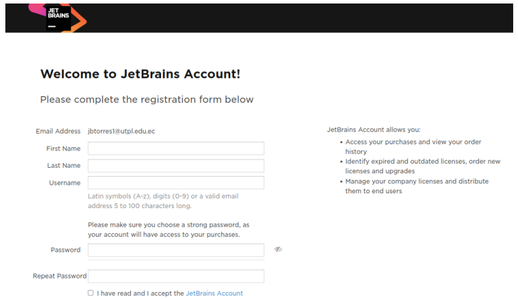
To complete your registration, click the link below:
Confirm your account

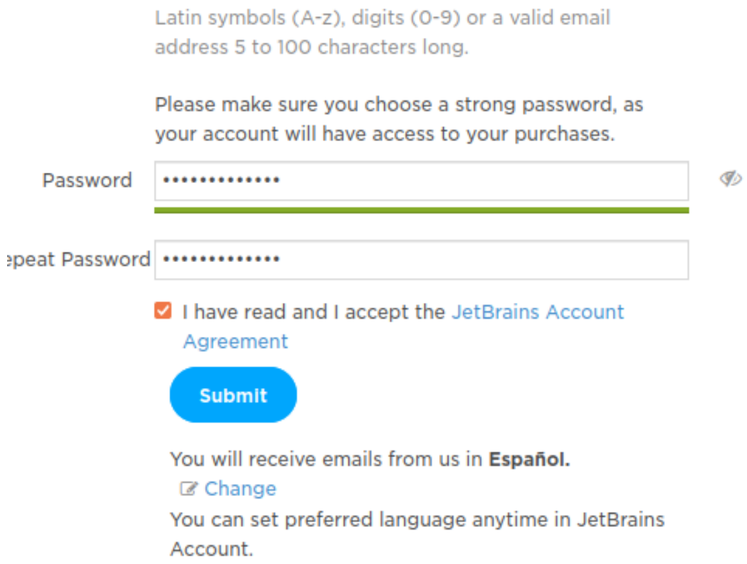
4. Ingresar los datos necesarios



Instalación - PyCharm

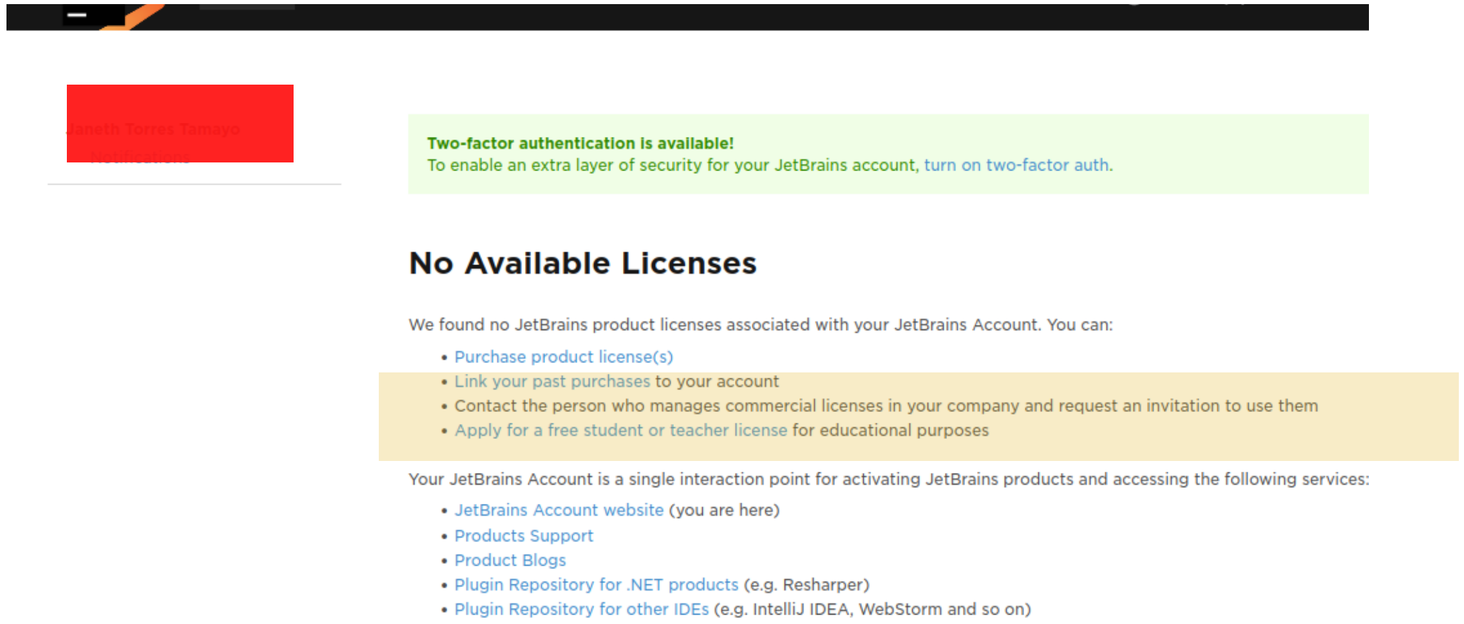
4. Ingresar los datos necesarios





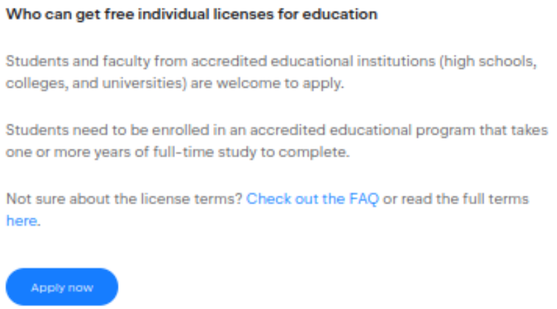
Instalación - PyCharm

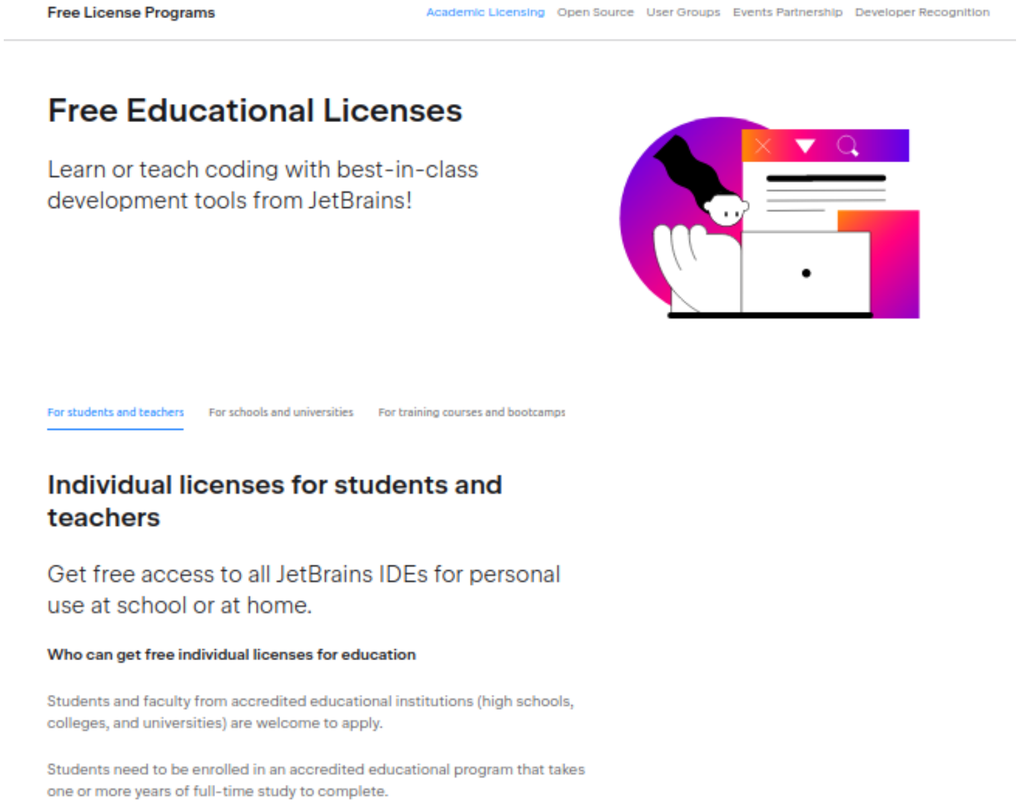
5. Revisar la información



Instalación - PyCharm

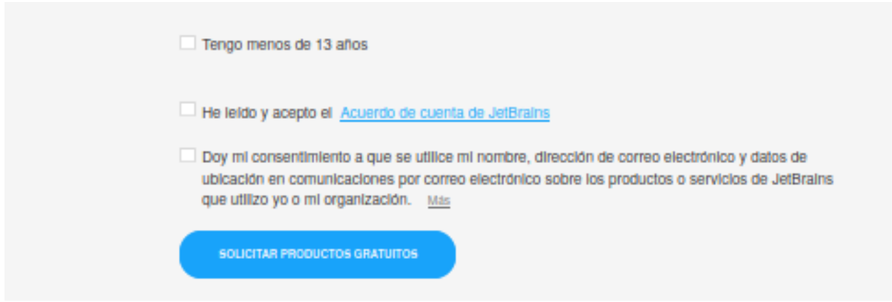
6. Ingresar la opción licencia educativa





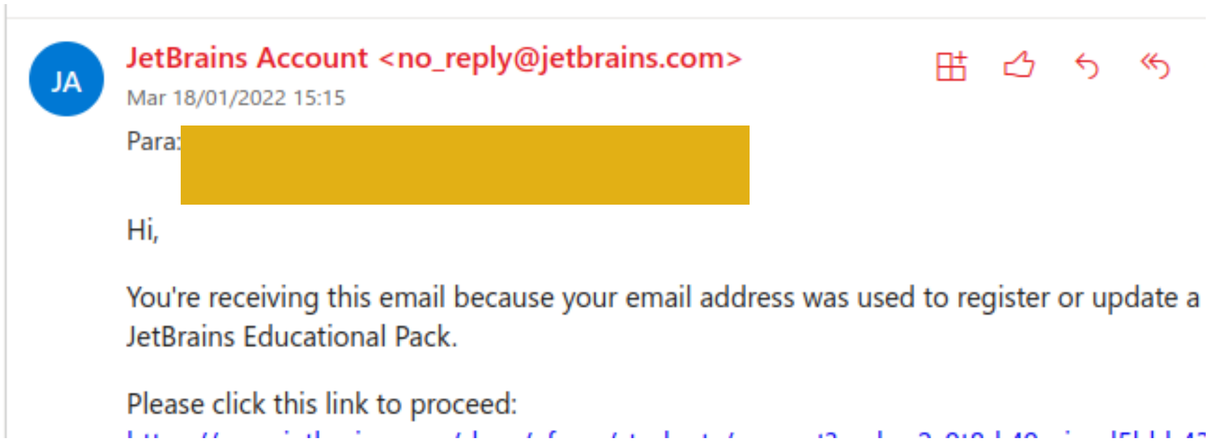
Instalación - PyCharm

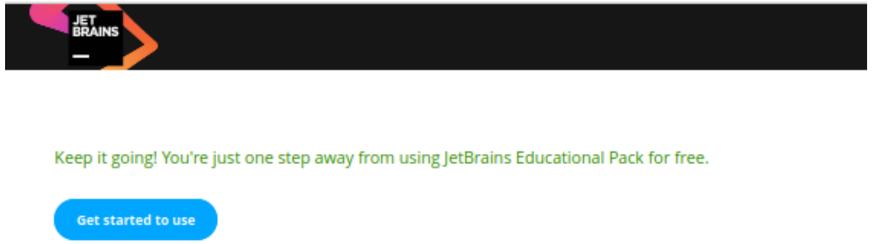
6. Ingresar la nueva información que se solicita



Instalación - PyCharm

7. Seguir las instrucciones del correo electrónico



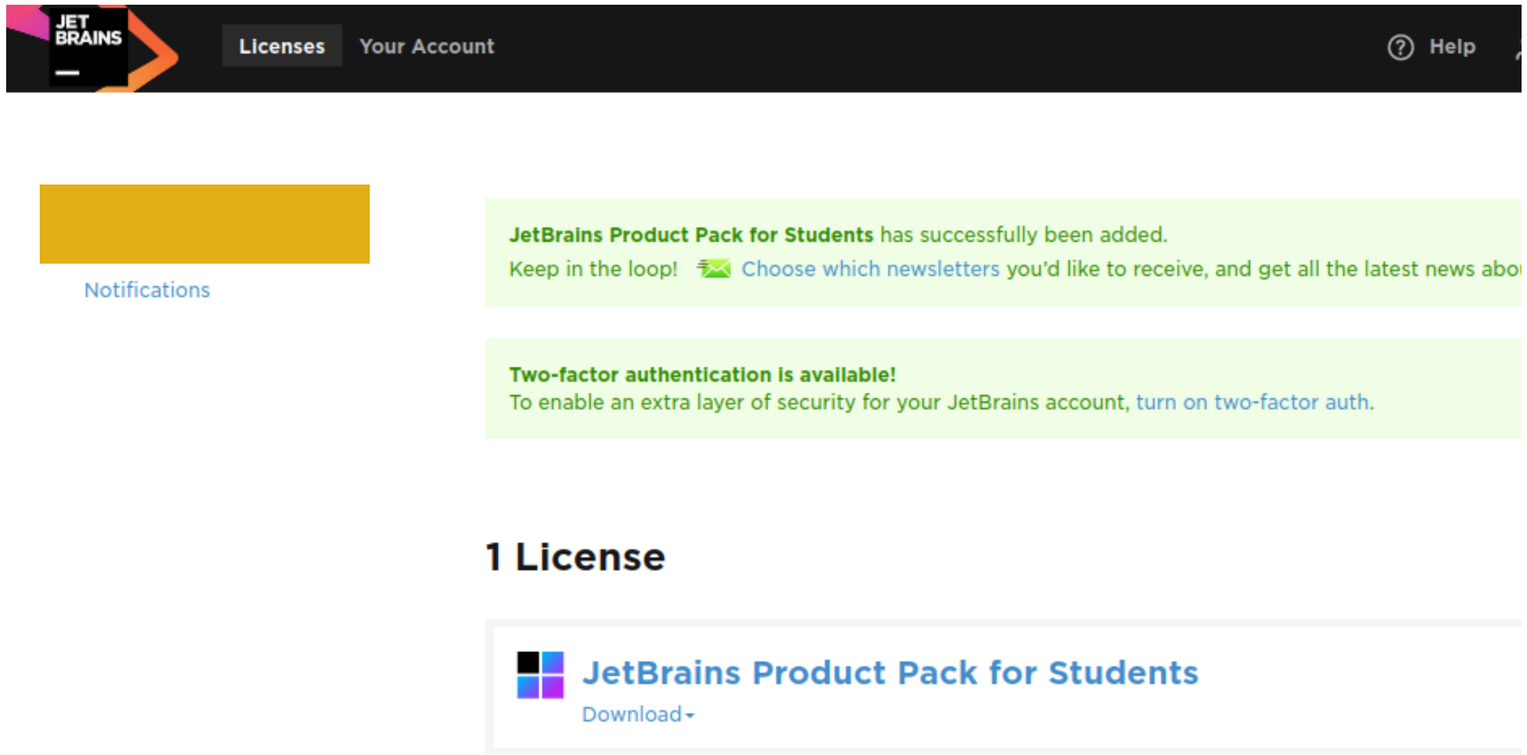


Instalación - PyCharm

8. Aceptar las condiciones

Instalación - PyCharm

9. Ingresar nuevamente a : https://account.jetbrains.com/login



Instalación - PyCharm

10. Ver listado de licencias dispobibles

License restriction: For educational use only

Valid through: January 17, 2023

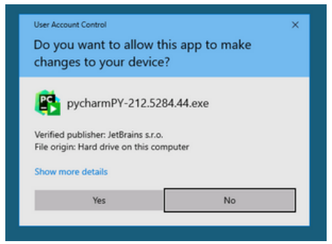
Following products included:

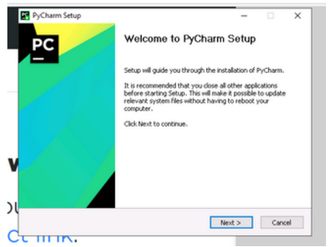
- AppCode
- CLion
- DataGrip
- DataSpell
- dotCover
- dotMemory
- dotTrace
- GoLand
- IntelliJ IDEA Ultimate
- PhpStorm
- PyCharm
- ReSharper
- ReSharper C++
- Rider
- RubyMine
- WebStorm

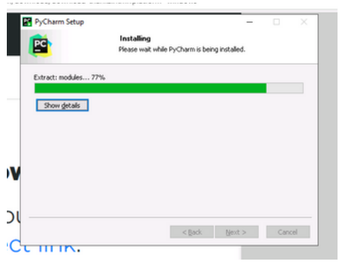
After downloading and installing the software, simply run it and follow the on-screen prompts to sign in with your JetBrains Account.

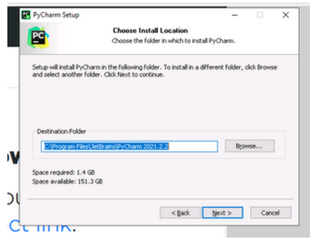
Instalación - PyCharm

11. Ingresar a PyCharm; descargar e instalar



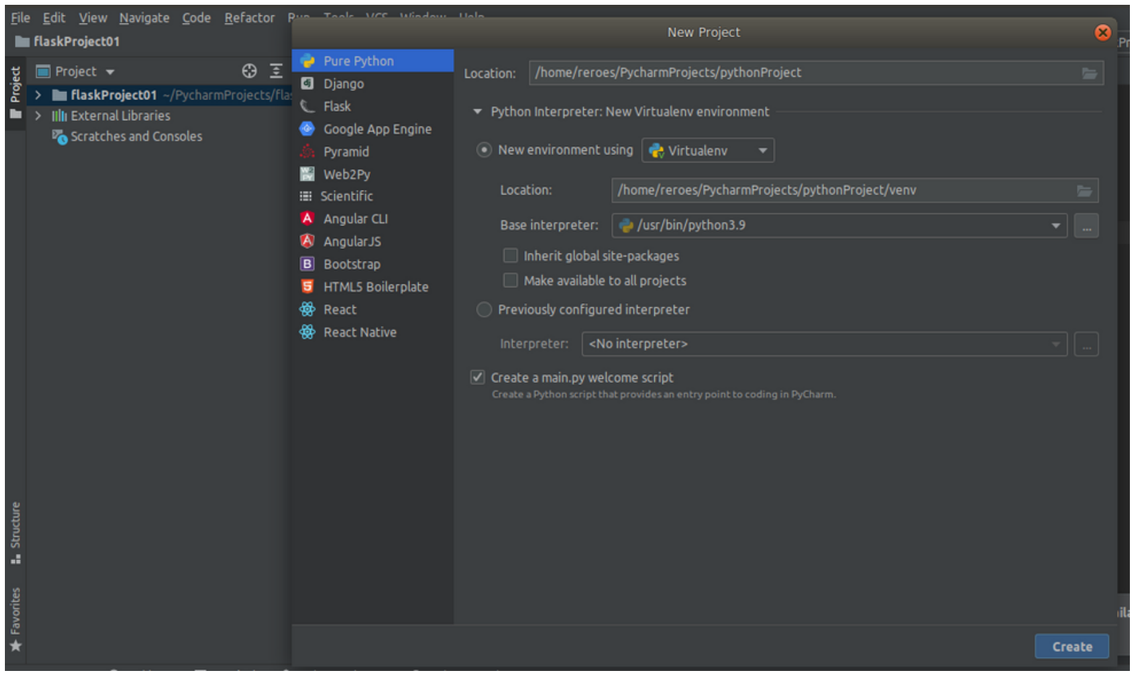






Instalación - PyCharm

11. Ingresar a PyCharm en la máquina local



Instalación

Download Python

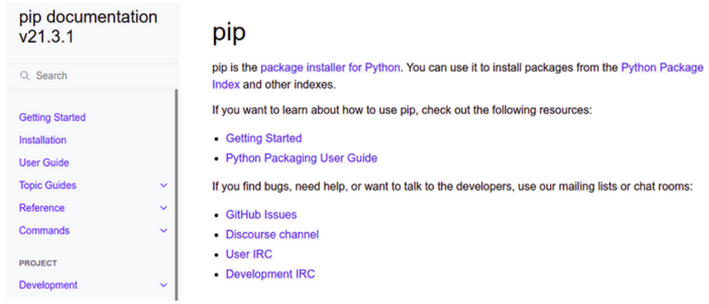
The official home of the Python Programming Language

Python.org / /static/humans.txt

https://www.python.org/downloads/

Usar pip, para instalar paquetes.

https://pip.pypa.io/en/stable/



Paquetes a instalar:

- Ipython: pip install ipython
- Virtualenv: pip install virtualenv

python™

Uso de virtualenv

Paquetes a instalar:

- Virtualenv: pip install virtualenv

> `virtualenv` is a tool to create isolated Python environments. Since Python `3.3`, a subset of it has been integrated into the standard library under the venv module. The `venv` module does not offer all features of this library, to name just a few more prominent:
>
> - is slower (by not having the `app-data` seed method),
> - is not as extendable,
> - cannot create virtual environments for arbitrarily installed python versions (and automatically discover these),
> - is not upgrade-able via pip,
> - does not have as rich programmatic API (describe virtual environments without creating them).

Comandos:

- Crear:
  - virtualenv ruta-a-carpeta-entorno
- Inciar entorno:
  - source ruta-a-carpeta-entorno/bin/activate
  - \ruta-a-carpeta-entorno\Scripts\activate.bat

Instalación

Usar pip, para instalar paquetes.

https://pip.pypa.io/en/stable/

IPython
Interactive Computing

pip install ipython

pandas

pip install pandas

Flask
web development,
one drop at a time

pip install flask

pip install jupyter
pip install jupyterlab

django

pip install django

bokeh

pip install bokeh

python™

Ejecutar código en Python

https://www.python.org/downloads/


```
ntroduccion-python/ejercicios$ python ejercicio5.py
Ingrese nombre del persona: René
Ingrese edad de persona: 30
Ingrese el sueldo de la persona: 500.2
Nombre:René
Edad:30
Sueldo:500.20
```

Opción 1:

- Abrir un editor de texto y guardar con la extensión .py
- Ingresar el código Python
- Guardar los cambios
- Abrir un terminal en su computador
- Ubicarse en la carpeta donde está el archivo guardado.
- Ejecutar el comando: python nombre-archivo.py

python™

Ejecutar código en Python

https://www.python.org/downloads/


```
ntroduccion-python/ejercicios$ python
Python 3.6.8 (default, Oct  7 2019, 12:59:55)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>>
>>>
>>>
```

Opción 2:

- Usar la consola de python que se instala por defecto

python™

Ejecutar código en Python

https://www.python.org/downloads/


```
ntroduccion-python/ejercicios$ ipython
Python 3.6.8 (default, Oct  7 2019, 12:59:55)
Type "copyright", "credits" or "license" for more information.

IPython 5.6.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]:
```

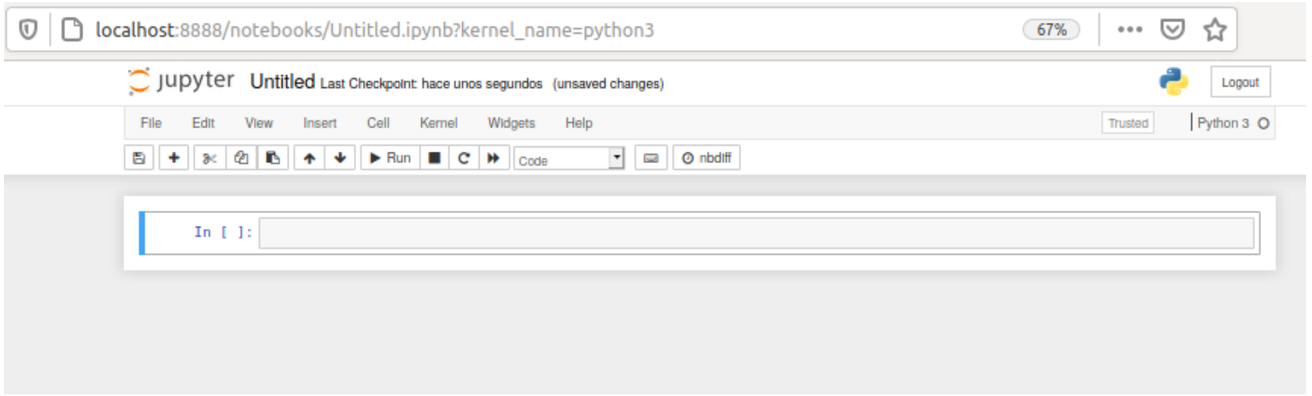
Opción 3:

- Usar la librería ipython

python™

Ejecutar código en Python

https://www.python.org/downloads/



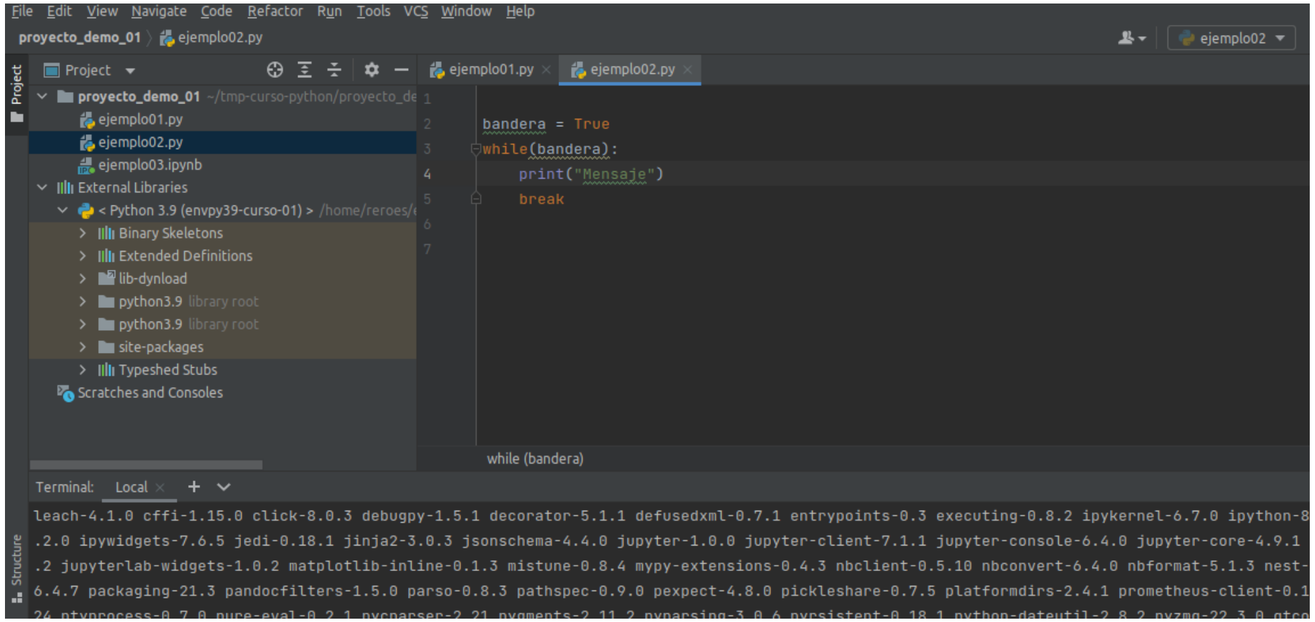
Opción 4:

- Usar la librería jupyter con el comando

jupyter notebook

Ejecutar código en Python

https://www.python.org/downloads/



Opción 5:

- Usar IDE
  - PyCharm

Tipos de Datos Básicos en Python

https://www.python.org/downloads/

- Cadenas (str)

c = “Hola Mundo”

- Booleano (bool)

bandera = True

bandera = False

- Entero (int)

valor = 100

- Decimal (float)

valor = 100.2

python™

Presentación - Imprimir información con Python

https://www.python.org/downloads/




```
mensaje = "Hola mundo"

print(mensaje)
```

```
# cadena - str
mensaje = "hola mundo"
print(mensaje)
print("%s" % (mensaje))
print(f"{mensaje}")
```

Ingreso de datos por teclado en Python

https://www.python.org/downloads/




```
nombre = input("Ingrese nombre del persona: ")

edad = int(input("Ingrese edad de persona: "))

sueldo = float(input("Ingrese el sueldo de la persona: "))

mensajeFinal = "Nombre:%s\nEdad:%d\nSueldo:%.2f\n" % (nombre,
        edad, sueldo)

print(mensajeFinal)
```

Sentencias - Condicionales

```
 9 sueldo = 100
10 if sueldo <= 100:
11     print("Correcto")
12 else:
13     if (sueldo >= 101) and (sueldo <= 110):
14         print("Sobresaliente")
15     else:
16         print("Incorrecto")
```

```
 4 sueldo = 110
 5 if sueldo <= 100:
 6     print("Correcto")
 7 elif (sueldo >= 101) and (sueldo <= 110):
 8     print("Sobresaliente")
 9 else:
10     print("Incorrecto")
```

Sentencias - Ciclos repetitivos

```
4 sueldo = 110
5
6 while sueldo <= 120:
7     print(f"{sueldo}")
8     sueldo = sueldo + 5
9
10 print("%.2f" % sueldo)
```

```
4 sueldo = 110
5
6 for i in range(0, sueldo):
7     print(i)
8
```

```
 9 sueldo = 110
10
11 for i in range(0, sueldo, 20):
12     print(i)
```

Sentencias - Estructuras de datos

- Listas
  - Estructura donde se puede almacenar variables de cualquier tipo de dato
  - Los valores ingresados a las listas se los separa con coma (,)
  - Declaración de lista vacía: mi_lista = [] (corchetes)

```
 5 lista1 = []
 6
 7 lista2 = ["a", 1, [], "b"]
 8
 9 lista3 = [1]
10 lista3.append("b")
11 lista3.append("c")
12 lista3.append(10)
13
```

Sentencias - Estructuras de datos

- Diccionarios
  - Conocidos como matrices asociativas.
  - Estructura que relaciona una llave con un valor.
  - Representamos un diccionario vacío con llaves así: diccionario = **{}**

```
 5 diccionario = {}
 6 diccionario["nombre"] = "René"
 7 diccionario["apellido"] = "Elizalde"
 8
 9 diccionario2 = {"nombre": "René", "apellido": "Elizalde", "edad": 30}
10 diccionario2["sueldo"] = 1000.2
```

python™

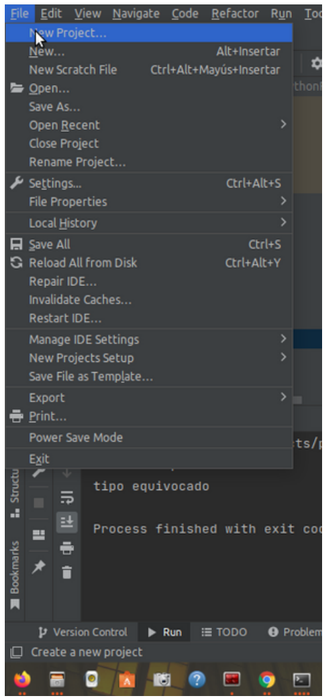
Funciones en Python

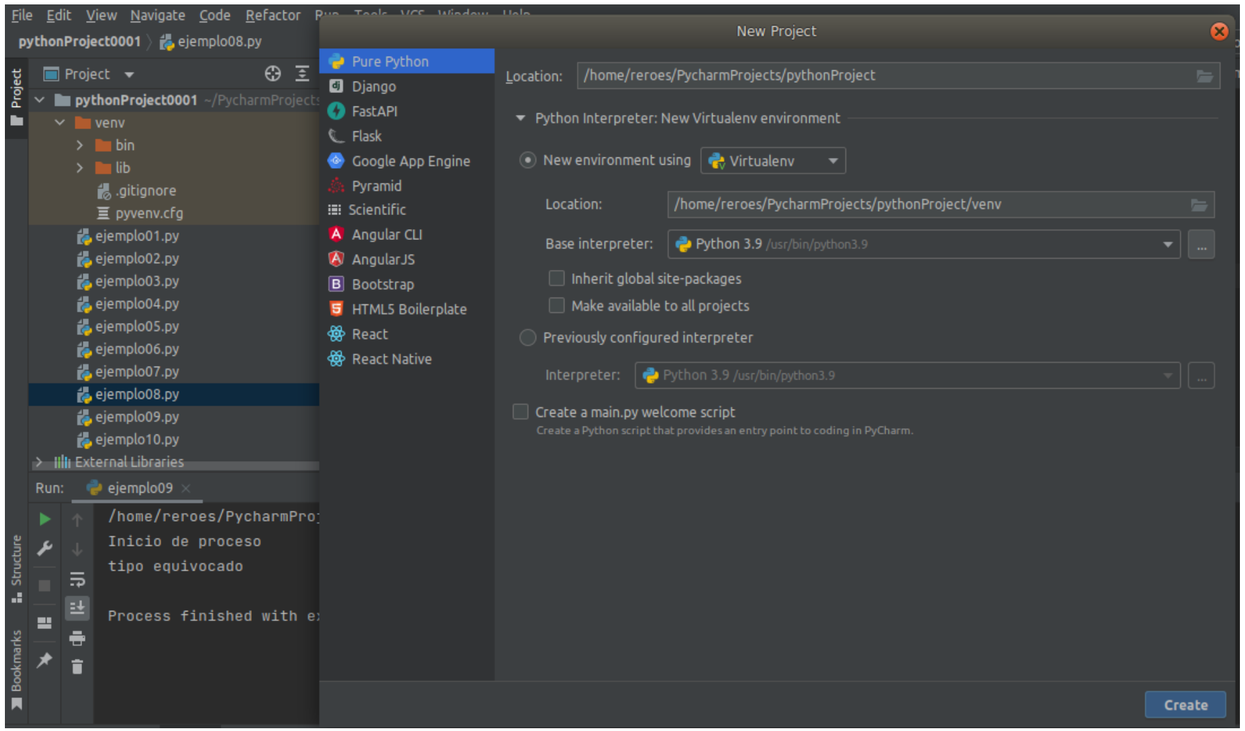
- Palabra reservada **def**
- Uso de dos puntos **:**, para dar inicio al cuerpo de la función

```
 4 def obtener_datos():
 5     """ """
 6     nombre = input("Ingrese nombre: ")
 7     apellido = input("Ingrese apellido: ")
 8     edad = input("Ingrese edad: ")
 9     edad = int(edad)
10     cadena = f"Datos Ingresados\n" \
11              f"Nombre: {nombre}\n" \
12              f"Apellido: {apellido}\n" \
13              f"edad: {edad}"
14     return cadena
15
16
17 if __name__ == "__main__":
18     print("Inicio de proceso")
19     mensaje = obtener_datos()
20     print(mensaje)
```

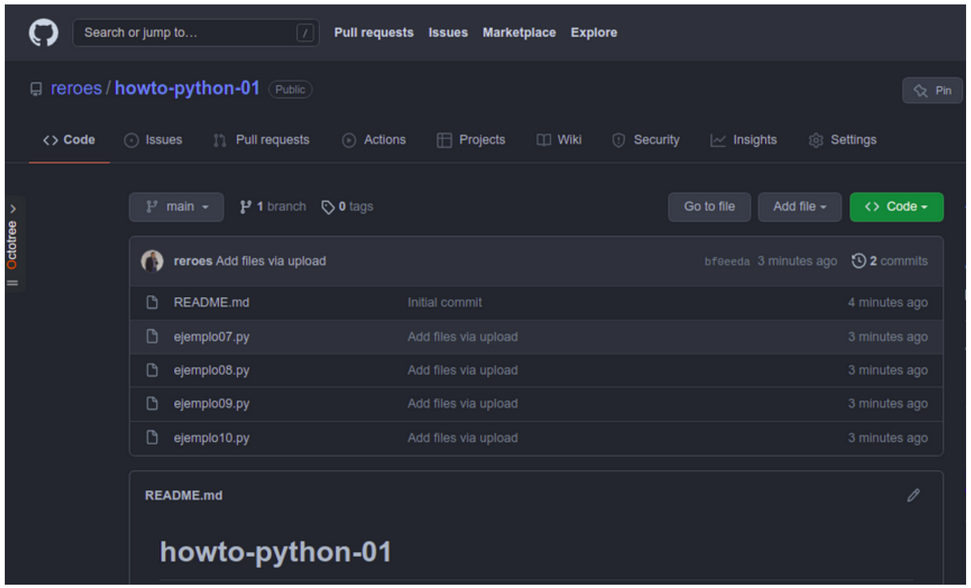
```
 6 def obtener_datos():
 7     """ """
 8     nombre = input("Ingrese nombre: ")
 9     apellido = input("Ingrese apellido: ")
10     edad = input("Ingrese edad: ")
11     edad = int(edad)
12     cadena = f"Datos Ingresados\n" \
13              f"Nombre: {nombre}\n" \
14              f"Apellido: {apellido}\n" \
15              f"edad: {edad}"
16     print(cadena)
17
18
19 if __name__ == "__main__":
20     print("Inicio de proceso")
21     obtener_datos()
```

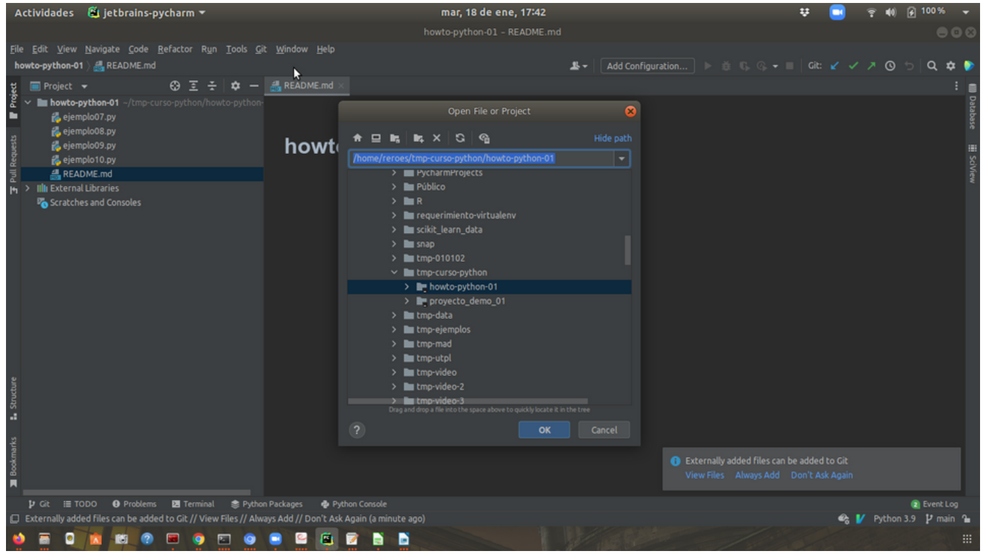
Caso 1: Creación de un proyecto den PyCharm





Caso 1: Clonar un repositorio - GitHub





KBS
Sistemas Basados en el Conocimiento
Grupo de Investigación

Futuro?

SciPy

NumPy

django The web framework for perfectionists with deadlines.

Flask
web development, one drop at a time

PyTorch

bokeh

jupyter

Project Jupyter

The Jupyter Notebook is a web-based interactive computing platform. The notebook combines live code, equations, narrative text, visualizations, interactive dashboards and oth...

jupyter.org

Pyramid
The Start Small, Finish Big Stay Finished Framework

Projects with ambition start small but finish big and must stay finished.
You need a Python web framework that supports your decisions, by artisans for artisans.

pandas

Tornado

matplotlib

django REST framework

Scrapy

spaCy

python™

Gracias

René Rolando Elizalde Solano
Departamento de Ciencias de la Computación y Electrónica
Sistemas Basados en el Conocimiento
rrelizalde@utpl.edu.ec
@reroes