

INSTITUTO TECNOLÓGICO DE BUENOS AIRES

Guía de PyQt5

Kammann, Lucas Agustín

Trozzo, Rafael Nicolás

3 de noviembre de 2019

Índice

Comentarios: ¿Qué es esto?	3
Requisitos: ¿Qué necesito para empezar?	3
Instalación: ¿Qué tengo que instalar?	3
Comandos para consola de windows	3
¿No te funcionan?	3
Compilar los diseños de QtDesigner	6
Compilar los recursos de QtDesigner	6
QtDesigner: ¿Cómo se usa?	7
PyQt	8
¿Cómo estructurar los proyectos?	8

Comentarios: ¿Qué es esto?

En la guía hay detalles sobre instalación y manejo de comandos, así como descripciones paso a paso y aclaraciones para comprender las cosas un poco más y no cometer errores al momento de usar PyQt5. Si sufre ansiedad de código, probablemente no debería empezar por aquí, sino por los ejemplos.

Se agradece toda colaboración o contribución para con el [Repositorio de GitHub](#), siguiendo el normal flujo de trabajo de Git con Fork y Pull-Request.

Requisitos: ¿Qué necesito para empezar?

- **Python 3:** Hay que instalar Python... fíjate [aquí](#).
- **PyCharm:** Hay otras opciones como VSCode, Atom, SublimeText, Eclipse, Notepad++, Bloc de notas, del color que quieras... Se recomienda PyCharm, fíjate [aquí](#).

Nota importante: según el proceso de instalación suele hacerse de forma automática o no, pero puede ser que necesites usar comandos en la consola de Windows ejecutando scripts de Python que son muy usados, como pip principalmente, o el propio intérprete de Python. Si no te funciona, es porque no lo tienes configurado en las variables de entorno del sistema, fíjate cómo [¿No te funcionan?](#).



```
bash: pip: command not found
```

Figura 1: Error en la ejecución desde consola con PIP

Instalación: ¿Qué tengo que instalar?

- **PyQt5:** Es necesario instalar este paquete. Básicamente es una adaptación del framework QtCreator original empleado en C++ para Python, puede ser que la documentación en este último lenguaje este un poco incompleta. Instalamos PyQt5 ejecutando en consola de comandos de Windows, o en la consola de Linux, el comando:

pip install PyQt5

- **PyQt5-tools/QtDesigner:** Las herramientas de diseño QtDesigner, entre otras, para usar con este framework hay que instalarlas con otro comando:

pip install pyqt5-tools

Comandos para consola de windows

¿No te funcionan?

Para que Windows pueda ejecutar un script, bash o algún ejecutable pasando argumentos al programa desde la consola simplemente utilizando una palabra clave, es necesario que esté registrado en las variables de entorno.

Nota importante: si bien abajo explico cómo agregar variables de entorno, es un tema interesante, sirve para otras cosas y tiene vínculo con el mecanismo de Python para importar un paquete o módulo en forma absoluta, podrías [mirarme](#), o [a mi](#), o también [a mi](#).

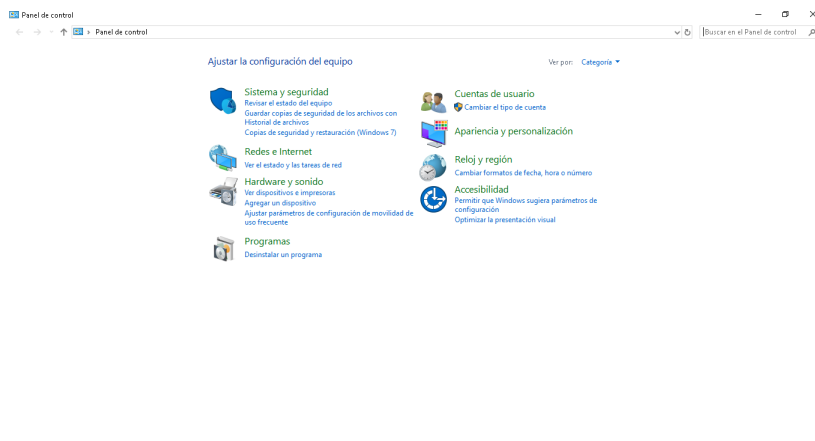


Figura 2: Abrimos Panel de Control de Windows y entramos a Sistema y Seguridad

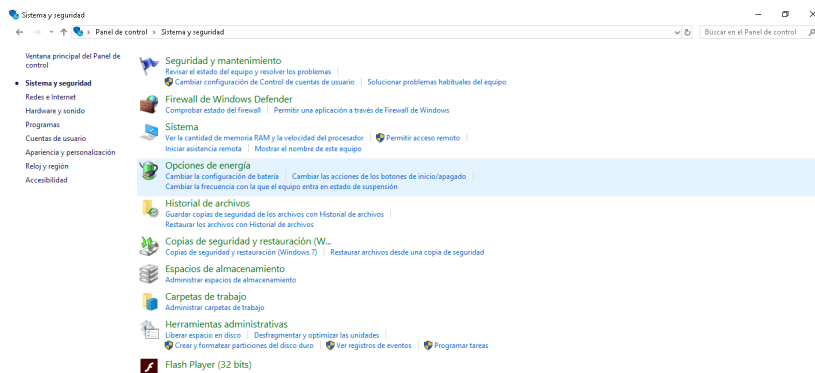


Figura 3: Entramos en Sistema

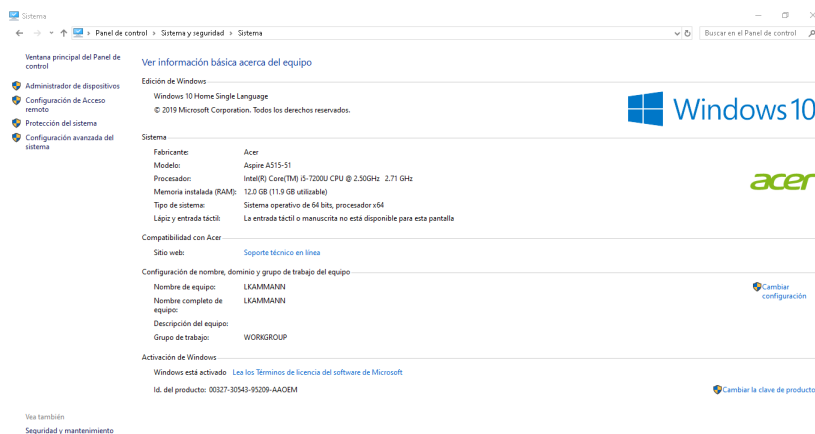


Figura 4: A la izquierda accedemos en Configuración avanzada del sistema

Compilar los diseños de QtDesigner

Con la consola de Windows/Linux abierta, posicionado en la carpeta donde tenemos el archivo "*.ui" que creamos y editamos con el QtDesigner, luego corremos la siguiente línea para compilarlo. Sobre las ventajas o desventajas de este método para utilizar los diseños que realizamos en QtDesigner, es necesario leer otra sección del documento.

```
pyuic5 -x filename.ui -o output.py
```

Compilar los recursos de QtDesigner

Con la consola de Windows/Linux abierta, posicionado en la carpeta donde tenemos el archivo "*.qrc" que creamos y editamos con el QtDesigner, luego corremos la siguiente línea para compilarlo.

```
pyrcc5 filename.qrc -o output.py
```

QtDesigner: ¿Cómo se usa?

PyQt

¿Cómo estructurar los proyectos?

Los proyectos que involucran el diseño de una GUI utilizando PyQt5 estarán compuestos por el Backend y el Frontend, conceptos que vienen del desarrollo web, y puedes ver [acá](#). Entonces, tu proyecto va a tener código en Python de la GUI, de la lógica de lo que hagas, y otros archivos como recursos, stylesheets, imagenes, etc... se propone la siguiente estructura de archivos muy utilizada en PyQt5, y Python en algunos casos:

```
/project
├── /designer
│   └── mainwindow.ui
├── /resources
│   └── logo.png
├── /tests
│   └── test_backend.py
├── /src
│   ├── __init__.py
│   ├── app.py
│   ├── mainwindow.py
│   └── /ui
│       └── mainwindow.py
├── main.py
├── test.py
├── requirements
└── readme
```