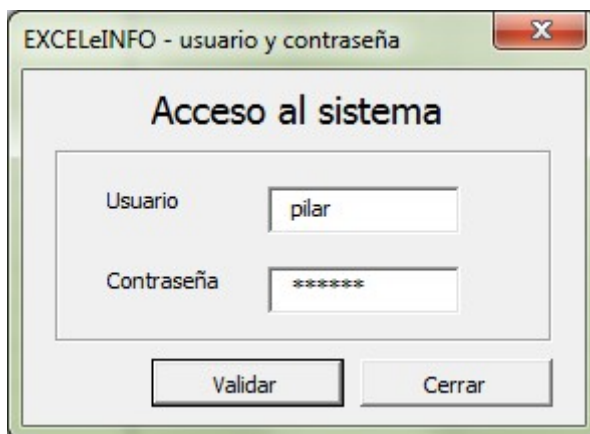


ACTIVIDAD 15. Controles de Acceso

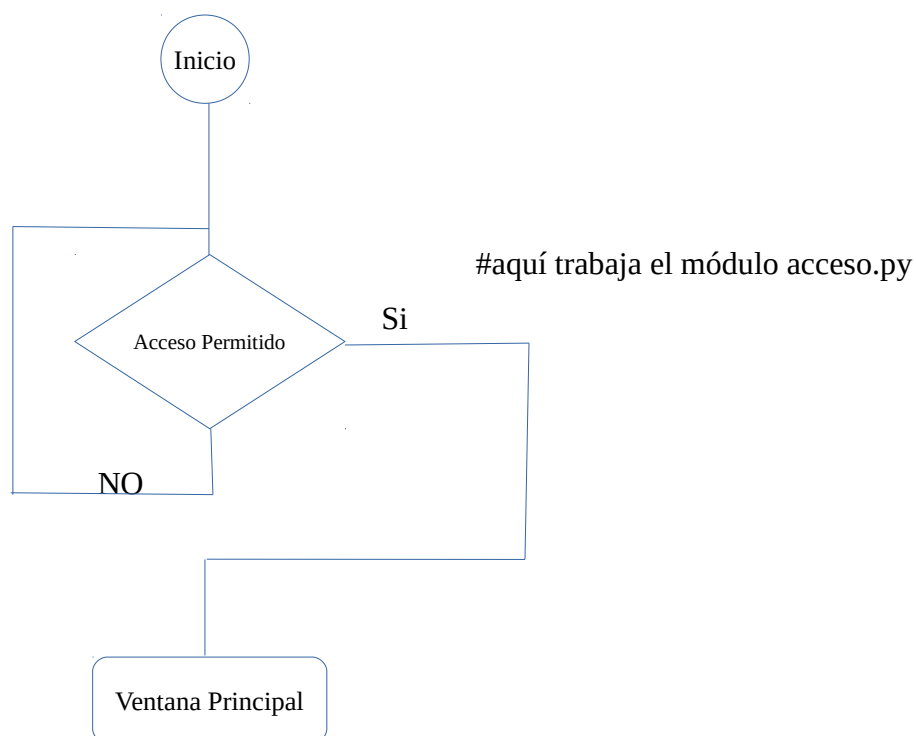
En esta actividad vamos a establecer un control de acceso a la aplicación que estamos desarrollando.

La idea es que al lanzar la aplicación lo primero que veamos es la típica pantalla de control de acceso. Algo así como esto:



Dicha ventana, además de los dos botones de uso obvio. Ahora vamos a desarrollar lo que nos piden en un módulo que podremos llamar **acceso.py**

1. En primer lugar debemos de añadir a la BBDD del sistema una tabla que nos guarde usuario y contraseña.
2. El DFD sería algo así:



3. Códigos de ayuda y algo de **encriptación**. Si vamos a distribuir la aplicación es lógico que guardemos encriptada la contraseña. El sistema más fiable en Python es el uso de la librería **PyCripto**¹. Los cifrados que soporta son *AES, BLOWFISH, CAST, DES, DES3, IDEA...* entre otros. Además permite la generación de números aleatorios.

A continuación algunas sentencias útiles para la resolución de la actividad:

```
from Crypto.Cipher import [Tipo_Cifrador]

#por su sencillez usaremos DES cuyos bloques son de 8 caracteres

...

usuario = self.entUsuario.get_text()

password = self.entPass.get_text()

# creación del cifrador New([clave],[modo],[vector IV])

cipher = DES.new('12345678')
```

Sólo la clave es obligatoria, y habrá que tener en cuenta si el tipo de cifrado exige que tenga un tamaño concreto, en este caso, ocho caracteres.

Los modos posibles son *MODE_ECB, MODE_CBC, MODE_CFB, MODE_PGP, MODE_OFB, MODE_CTR, MODE_OPENPGP*. Si se utilizan los modos *MODE_CBC* o *MODE_CFB*, hay que inicializar el tercer parámetro (Vector IV) que permite dar un valor inicial al cifrador.

```
c_usuario = cipher.encrypt(usuario) #realmente no tiene sentido encriptarlo
c_password = cipher.encrypt(password)
```

A continuación insertaríamos los valores encriptados en la tabla de acceso. Posteriormente cada vez que deseásemos acceder al sistema tendríamos que comparar el valor de usuario y password para permite el acceso al sistema. Para ello necesitamos desencriptar:

```
d_usuario = cipher.decrypt(d_usuario).strip() # ni desencriptarlo
d_password = cipher.decrypt(d_password).strip()
```

Actividades

1. Para empezar tenemos que crear un usuario inicial para poder acceder al sistema.
2. En la barra de menús, una vez dentro del sistema, podemos crear nuevos usuarios con sus respectivos password.

¹ <https://pypi.python.org/pypi/pycrypto>