

Lockifi



Manual de instalación de Lockifi



Manual de instalación de Lockifi

Este manual está hecho sobre Linux pero en todo momento se especificará las diferencias para la instalación de lockifi-python sobre un entorno Windows.

1. Nos aseguramos que todos los paquetes necesarios están instalados en nuestro sistema:

python	→	sudo apt-get install python
git	→	sudo apt-get install git
pip	→	sudo apt-get install python-setuptools

```
pi@raspberrypi:~/lockifi-dir/lockifi-python $ sudo apt-get install python
Reading package lists... Done
Building dependency tree
Reading state information... Done
python is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi:~/lockifi-dir/lockifi-python $ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi:~/lockifi-dir/lockifi-python $ sudo apt-get install python-setuptools
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-setuptools is already the newest version.
python-setuptools set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi:~/lockifi-dir/lockifi-python $
```

Windows: Las páginas de descarga de los programas son las siguientes:

[python](#) [git](#) [pip](#)

2. Instalamos el módulo “latchsdk” y el módulo “texttable”:

latchsdk	→	sudo pip install latchsdk
texttable	→	sudo pip install texttable

```
pi@raspberrypi:~/lockifi-dir/lockifi-python $ sudo pip install latchsdk
Downloading/unpacking latchsdk
  Downloading latchsdk-1.0.0.zip
  Running setup.py (path:/tmp/pip-build-yOZOWc/latchsdk/setup.py) egg_info for package latchsdk

Installing collected packages: latchsdk
  Running setup.py install for latchsdk

Successfully installed latchsdk
Cleaning up...
pi@raspberrypi:~/lockifi-dir/lockifi-python $ sudo pip install texttable
Downloading/unpacking texttable
  Downloading texttable-0.8.4.tar.gz
  Running setup.py (path:/tmp/pip-build-n3n__b/texttable/setup.py) egg_info for package texttable
e

Installing collected packages: texttable
  Running setup.py install for texttable

Successfully installed texttable
Cleaning up...
pi@raspberrypi:~/lockifi-dir/lockifi-python $
```



3. Creamos un directorio en el directorio de inicio (`mkdir ~/lockifi-dir`) y accedemos a el (`cd ~/lockifi-dir`).

```
pi@raspberrypi:~ $ mkdir ~/lockifi-dir
pi@raspberrypi:~ $ cd ~/lockifi-dir
pi@raspberrypi:~/lockifi-dir $
```

Windows: en esta plataforma no podemos usar “~” por lo tanto podemos crear el directorio y acceder a el con una ruta absoluta. Para crear la carpeta (`mkdir C:\Users\[NOMBRE DE USUARIO]\lockifi-dir`) y para posicionarnos (`cd C:\Users\[NOMBRE DE USUARIO]\lockifi-dir`) substituyendo en ambos casos por nuestro nombre de usuario.

4. Descargamos Lockifi de su repositorio en GitHub y entramos dentro de directorio de descarga:

Descargar Lockifi	→	<code>git clone git://github.com/srborines/lockifi-python</code>
Acceder al directorio	→	<code>cd lockifi-python</code>

```
pi@raspberrypi:~/lockifi-dir $ git clone git://github.com/srborines/lockifi-python
Cloning into 'lockifi-python'...
remote: Counting objects: 36, done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 36 (delta 6), reused 32 (delta 5), pack-reused 0
Receiving objects: 100% (36/36), 23.54 KiB | 0 bytes/s, done.
Resolving deltas: 100% (6/6), done.
Checking connectivity... done.
pi@raspberrypi:~/lockifi-dir $ cd lockifi-python
pi@raspberrypi:~/lockifi-dir/lockifi-python $
```

5. Instalamos el paquete “screen” que nos permitirá poder cerrar la terminal en cualquier momento y que Lockifi se siga ejecutando. Una vez instalado creamos una nueva screen llamada “lockifi”:

Instalar screen	→	<code>sudo apt-get install screen</code>
Crear nueva screen	→	<code>screen -R lockifi</code>

```
pi@raspberrypi:~/lockifi-dir/lockifi-python $ sudo apt-get install screen
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  iselect screenie byobu
The following NEW packages will be installed:
  screen
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/522 kB of archives.
After this operation, 854 kB of additional disk space will be used.
Selecting previously unselected package screen.
(Reading database ... 123538 files and directories currently installed.)
Preparing to unpack .../screen_4.2.1-3+deb8u1_armhf.deb ...
Unpacking screen (4.2.1-3+deb8u1) ...
Processing triggers for systemd (215-17+deb8u2) ...
Processing triggers for man-db (2.7.0.2-5) ...
Processing triggers for install-info (5.2.0.dfsg.1-6) ...
Setting up screen (4.2.1-3+deb8u1) ...
pi@raspberrypi:~/lockifi-dir/lockifi-python $ screen -R lockifi
```



Manual de instalación de Lockifi

6. En esta screen ejecutamos Lockifi (`python lockifi.py`), de esta forma aunque cerremos la terminal Lockifi se seguirá ejecutando y podremos recuperar la vista de usuario en todo momento:

```
Lockifi beta - Plugin Latch

Menu:

1. Show/edit users
2. Show/edit routers
3. Start listen
4. Config Latch
5. Config Lockifi
6. Show log
7. Exit

Option: █
```

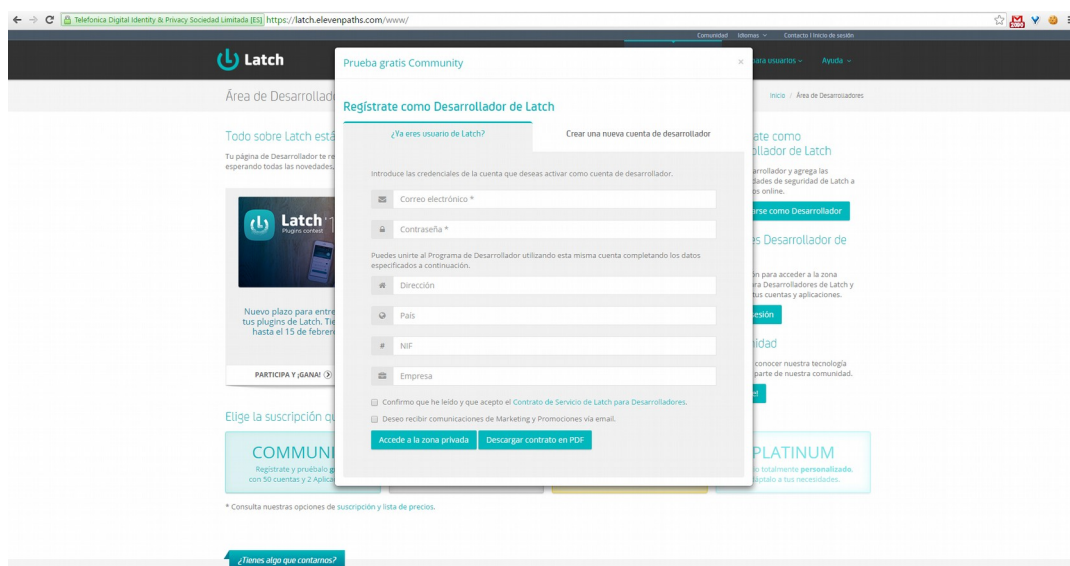
7. Si cerramos la terminal y queremos volver a ver el estado de Lockifi o interaccionar con él simplemente tenemos que recuperar la screen (`screen -r lockifi`). Nota: para ver las screen que tenemos abiertas debemos introducir (`screen -ls`)

```
pi@raspberrypi:~ $ screen -ls
There is a screen on:
      18205.lockifi      (12/02/16 11:26:36)      (Detached)
1 Socket in /var/run/screen/S-pi.

pi@raspberrypi:~ $ screen -r lockifi █
```

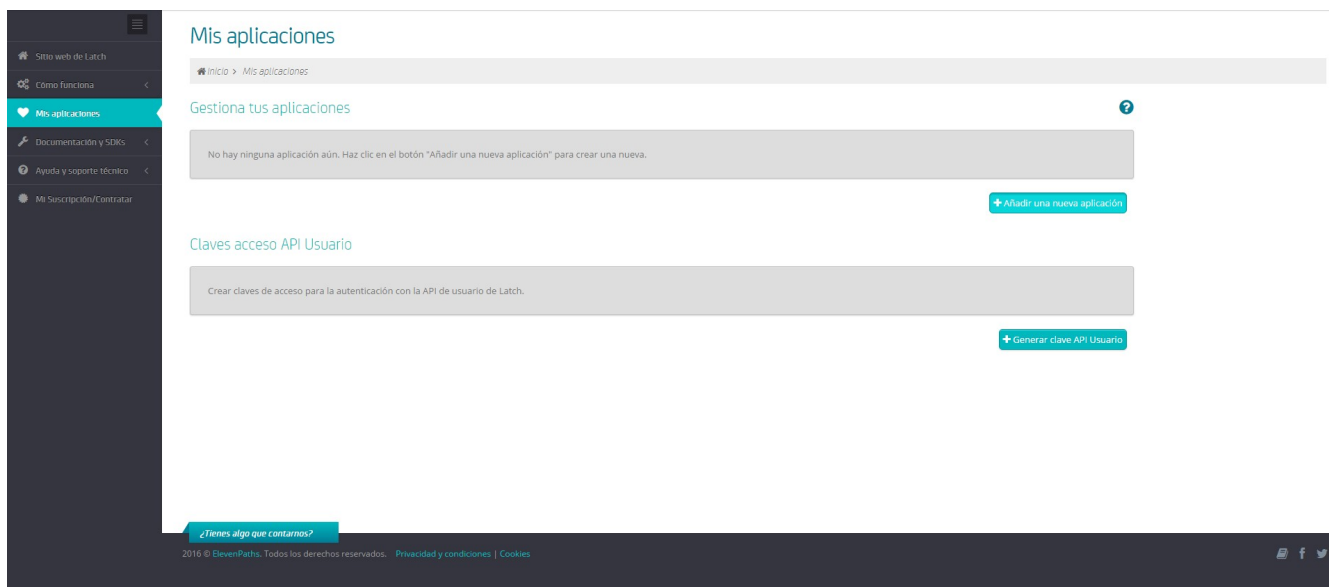
Windows: en Windows no es posible desanclar el proceso de la terminal por lo que deberá estar siempre abierta. Por lo tanto omitimos la instalación de “screen” y ejecutamos Lockifi de la misma forma (`python lockifi.py`).

8. Procedemos a configurar Latch en Lockifi. Nos registramos en [Latch](https://latch.elevenpaths.com/) , tendremos que confirmar nuestro correo, y una vez hecho esto ya podremos acceder.

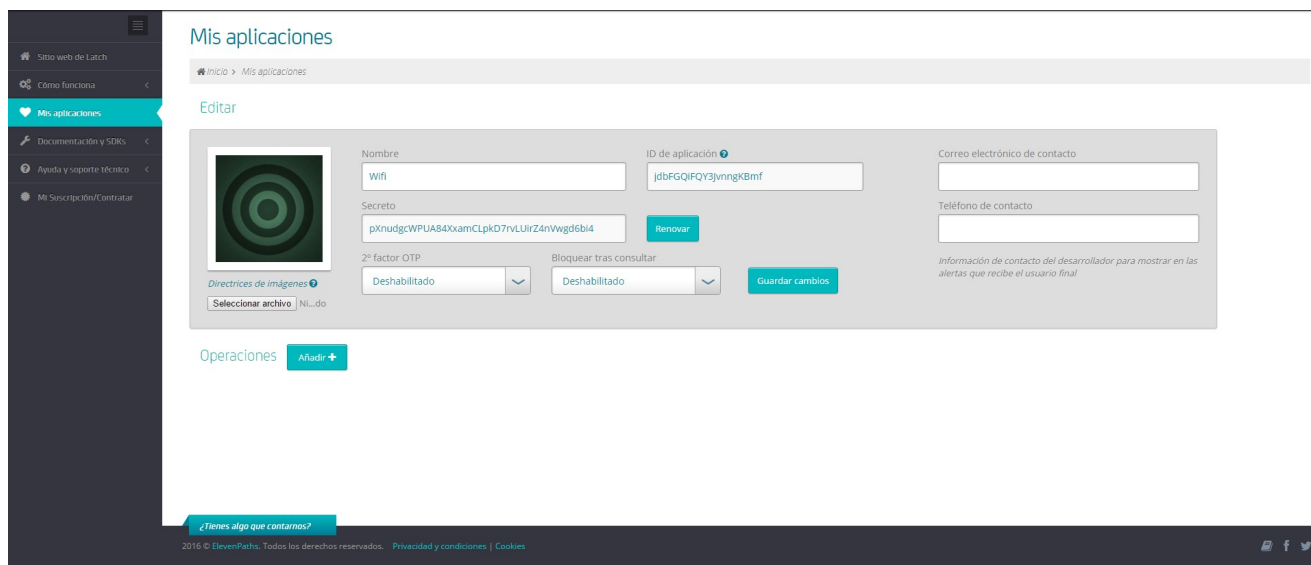




9. Ya dentro del área de desarrolladores de Latch seleccionamos la opción “añadir una nueva aplicación”



10. Introducimos el nombre con el que identificará nuestra aplicación y ya obtenemos la configuración de Latch para nuestra aplicación





11. En el menú de Lockifi seleccionamos la opción 4, y después la opción 2. Introducimos el AppID y la SecretKey que acabamos de obtener:

```
Lockifi beta - Plugin Latch
```

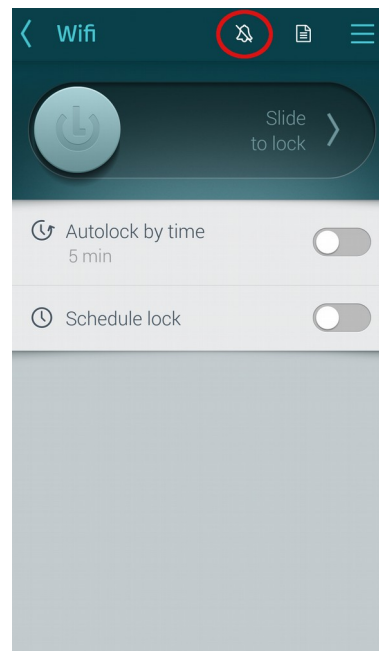
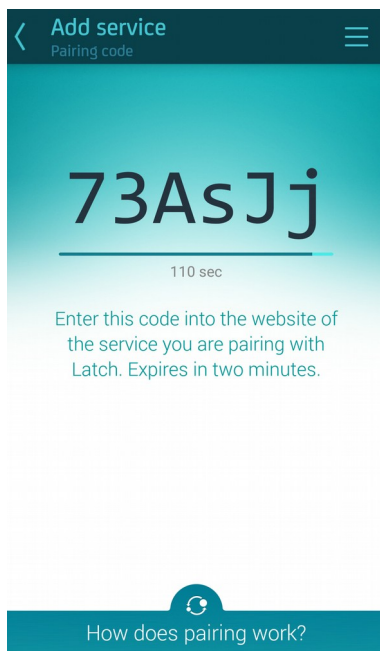
```
Latch configuration>:
```

```
AppID: <empty>
```

```
SecretKey: <empty>
```

```
Enter the new AppId (press enter to not change it): █
```

12. De nuevo en el menú principal seleccionamos la opción uno y en el menú de usuarios la opción 2 para añadir un usuario. Introducimos su nombre y lo pareamos con nuestra app de Latch. Importante desactivar el aviso de bloqueo de sesión.



```
Lockifi beta - Plugin Latch
```

```
Users>Add user:
```

```
Enter a username[enter 0 to cancel]: PabloRT
```

```
Enter pairing code: 73AsJj █
```



13. Configuramos el modo de escucha de Lockifi, desde el menú principal accedemos al apartado de Lockifi pulsando 5 y lo editamos pulsando 2.

Changes: El programa detecta cambios, si cualquiera de los usuarios cambia su Latch el sistema lo detecta y pone el Wifi a ese estado.

All open: El wifi solo está encendido si todos dan su consentimiento abriendo el Latch.

```
Lockifi beta - Plugin Latch

Lockifi configuration>:

    Listening mode: all_open

Enter the new mode [1:all_open/2:changes] (press 0 to cancel): █
```

14. Escogemos el modelo de nuestro Router accediendo a la opción 2 desde el menú principal, luego la opción 3, seleccionamos el router que queremos predeterminar, dejamos los dos primeros campos igual y por ultimo afirmamos que queremos predeterminarlo.

```
Lockifi beta - Plugin Latch

ROUTERS:
+-----+-----+-----+-----+
| ID | Selected | Name | file |
+-----+-----+-----+-----+
| 1 | *default* | HitronBVW3653 | HitronBVW3653.py |
+-----+-----+-----+-----+

Enter the new name for the router with name 'HitronBVW3653' (press enter to not change it):
Enter the new filename for the router with filename 'HitronBVW3653.py' (press enter to not change it):
Do you want do default this router?(y/n)y █
```

15. Ya podemos seleccionar la opción 3 del menú principal y empezar a controlar nuestro Wifi mediante Latch.

```
Lockifi beta - Plugin Latch

USERS:
+-----+-----+-----+-----+
| ID | Name | AccountId | Status |
+-----+-----+-----+-----+
| 1 | PabloRT | 8PfFFHCsdTsNtdWT | on |
| | | ffWBjD444q9tTuhF | |
| | | 6fgNgJNb3Hb3T7f2 | |
| | | jW7FB7jDZ62LxCtM | |
+-----+-----+-----+-----+

Listening...(Introduce 'quit' to stop): █
```