

## Usando QR para mostrar contraseñas largas para nuestro WiFi

Hace ya tiempo [hablamos de los códigos QR](#) pero sinceramente habíamos olvidado darle un uso práctico al software. Ahora con los ataques a la encriptación WPA2 basados en la esperanza que usemos contraseñas cortas (y usando tablas *rainbow*) ha tomado auge una nueva solución, veamos cómo.

### Repaso a la ingeniería social

Pues también [escribimos un artículo sobre introducción de contraseñas](#) a las cuales al final podíamos agregar todos los caracteres que queramos para así alargarlas y cifrarlas antes de enviarlas, lo que dificulta mucho su descifrado si en el camino le sacan copia (ataque de hombre en el medio). El chiste del asunto es que nuestro servidor sabrá de antemano cuántos primeros caracteres tomar y descartar los demás y luego crear *hash* y comparar contraseñas.

Sí, han adivinado: la solución para evitar que descifren las contraseñas de nuestras redes inalámbricas es... **hacerlas lo más largas posibles**. En el caso de la norma WPA2 son 63 caracteres. ¿El problema es cómo metemos nosotros una contraseña tan larga?

### Copiar y pegar

Como hacen todos los estudiantes de bachillerato ahora: copiar y pegar *pero con estilo*. Ahora los teléfonos con sistema operativo **Android 10** cuentan con la norma y función de leer el SSID, el tipo de cifrado, la contraseña y si está oculta la red. Veamos cómo generar códigos QR para ello y como generar contraseñas aleatorias (no, no estamos en los años 1990 cuando tanto discutimos sobre cómo generar números verdaderamente aleatorios con nuestras computadoras).

### Instalando pwgen

Poco tenemos que decir: instalamos con **sudo apt install pwgen** y luego usamos **pwgen -s 63** para generar contraseñas de 63 letras y/o números de largo. Dos cosas curiosas:

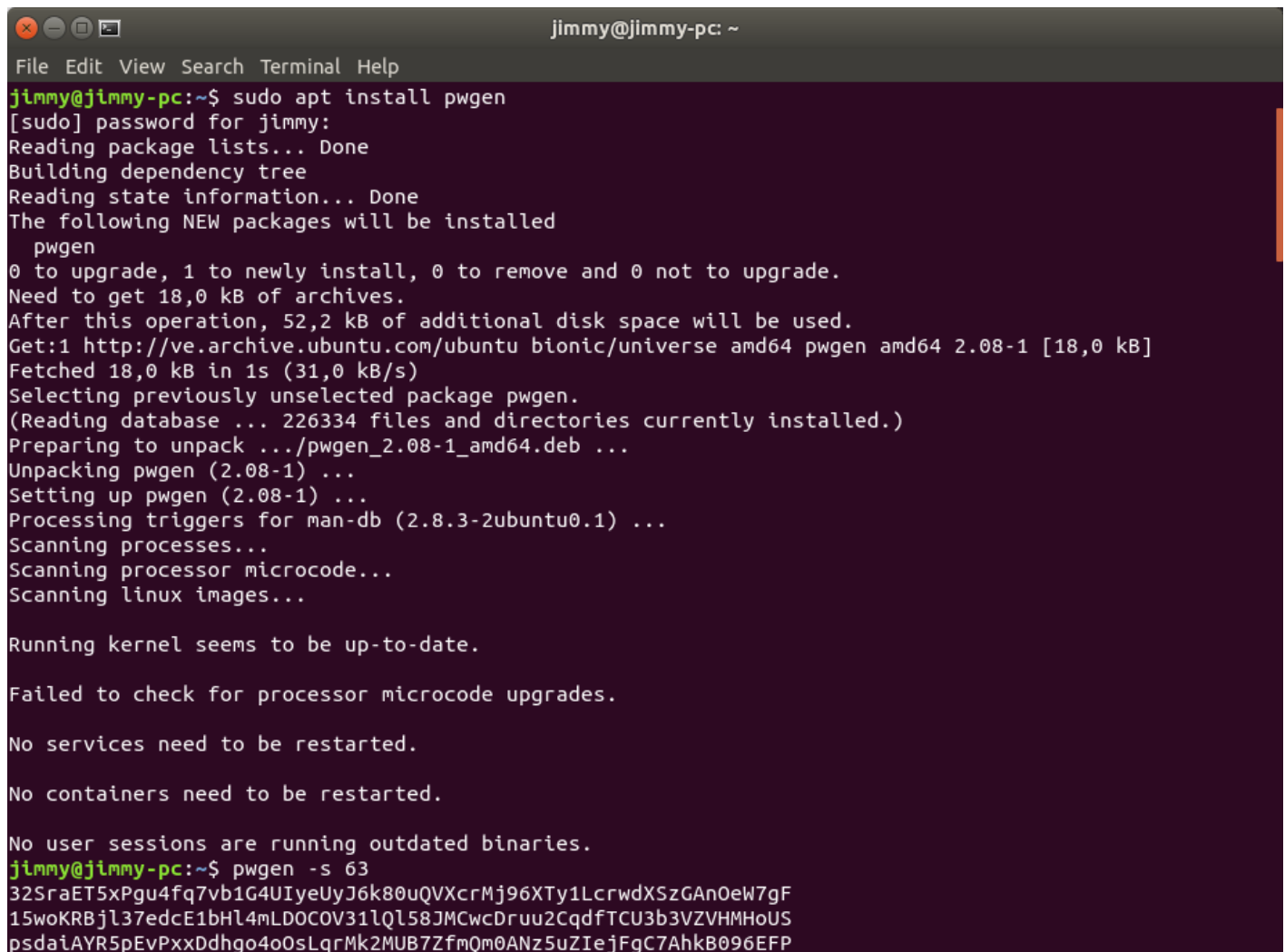
## KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

---

1. En realidad **pwgen** fue programado para generar contraseñas que nos sean relativamente fácil de recordar (en idioma inglés) pero con el parámetro **-s** eliminamos tal comportamiento.
2. No da un solo resultado, sino varios. Si somos paranoicos nos viene de perlas: ¡**pwgen** no sabrá cuál de ellas utilizaremos!



```
jimmy@jimmy-pc: ~  
File Edit View Search Terminal Help  
jimmy@jimmy-pc:~$ sudo apt install pwgen  
[sudo] password for jimmy:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following NEW packages will be installed  
  pwgen  
0 to upgrade, 1 to newly install, 0 to remove and 0 not to upgrade.  
Need to get 18,0 kB of archives.  
After this operation, 52,2 kB of additional disk space will be used.  
Get:1 http://ve.archive.ubuntu.com/ubuntu bionic/universe amd64 pwgen amd64 2.08-1 [18,0 kB]  
Fetched 18,0 kB in 1s (31,0 kB/s)  
Selecting previously unselected package pwgen.  
(Reading database ... 226334 files and directories currently installed.)  
Preparing to unpack .../pwgen_2.08-1_amd64.deb ...  
Unpacking pwgen (2.08-1) ...  
Setting up pwgen (2.08-1) ...  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
Scanning processes...  
Scanning processor microcode...  
Scanning linux images...  
  
Running kernel seems to be up-to-date.  
  
Failed to check for processor microcode upgrades.  
  
No services need to be restarted.  
  
No containers need to be restarted.  
  
No user sessions are running outdated binaries.  
jimmy@jimmy-pc:~$ pwgen -s 63  
32SraET5xPgu4fq7vb1G4UIyeUyJ6k80uQVXcrMj96XTy1LcrwdXSzGAn0eW7gF  
15woKRBj137edcE1bH4mLD0COV31lQl58JMCwcDrUU2CqdfTCU3b3VZVHMH0US  
psdaiAYR5pEvPxxDdhgo4o0sLqrMk2MUB7ZfmQm0ANz5uZIEjFgC7AhkB096EFP
```

```
sudo apt install pwgen
```

## Instalando qrencode

El cual está programado en lenguaje C, solo colocamos **sudo apt install qrencode**, sin mayor

## KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.  
<https://www.ks7000.net.ve>

---

complicación.

## Generando el código QR

Ahora debemos utilizar el siguiente formato para el comando:

```
qrencode -o nombre_imagen.png "WIFI:T:WPA;S;;P:;;"
```

Como vemos arranca con una etiqueta identificadora «**WIFI**» luego dos puntos y luego cuatro pares de valores separados por punto y coma cada uno. Para la imagen de muestra escogimos los siguientes valores:

- T:**WPA** (tipo de cifrado).
- S:**KS7000\_BBC2** (identificador de la red inalámbrica).
- P:  
**eWw**  
**LZKoA3e**  
**WoWb5cTYtHeczq6zOFAVTJ2zqdU8scW0VyK2qkKJ0WYIkXA50Do0D** (contraseña).
- Y el último campo (no usado aquí, no mostrado) H:**true** si decidimos no anunciar el identificador de red o "red oculta".

```
qrencode -o wifi_bbc2.png "WIFI:T:WPA;S:KS7000_BBC2;P:eWwLZKoA3eWoWb5cTYtHeczq6zOFAVTJ2zqdU8scW0VyK2qkKJ0WYIkXA50Do0D;;"
```

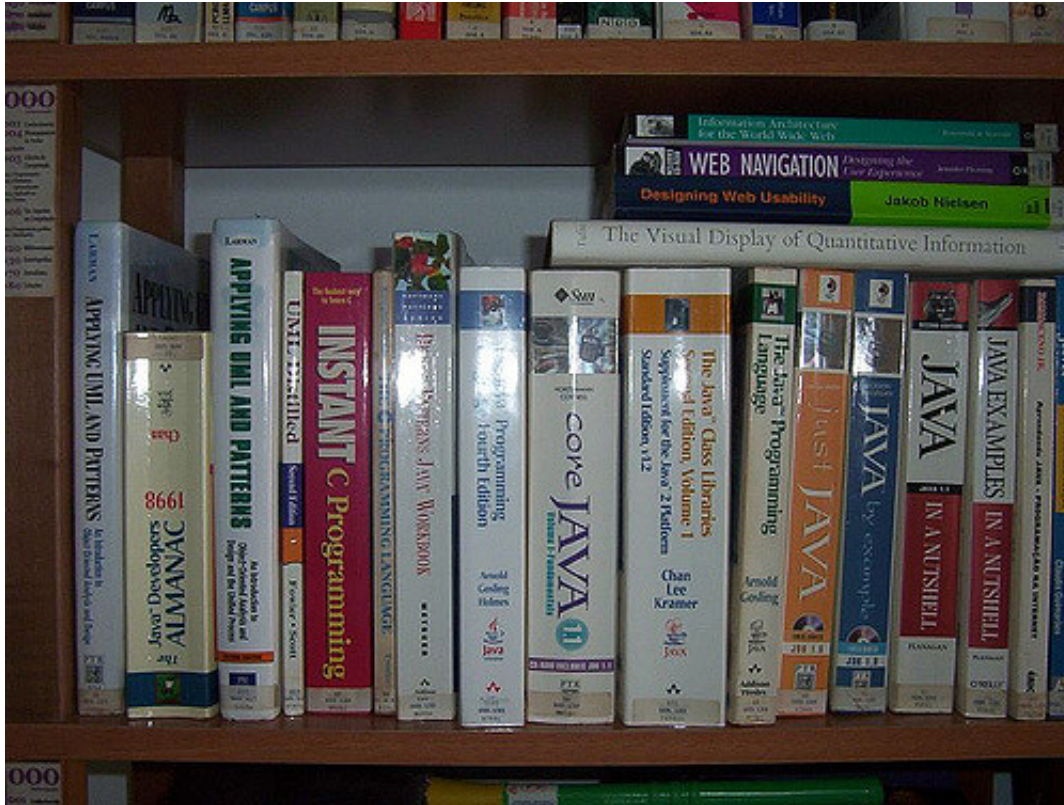
```
qrencode -o wifi_bbc2.png
```

## KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

---



## Fuentes consultadas

### En idioma castellano

- «».
- «».
- «».

### En idioma italiano

- [«Package: qrencode \(4.0.2-1\) codificatore di QR Code in immagine PNG»](#).
- «».

## KS7000+WP

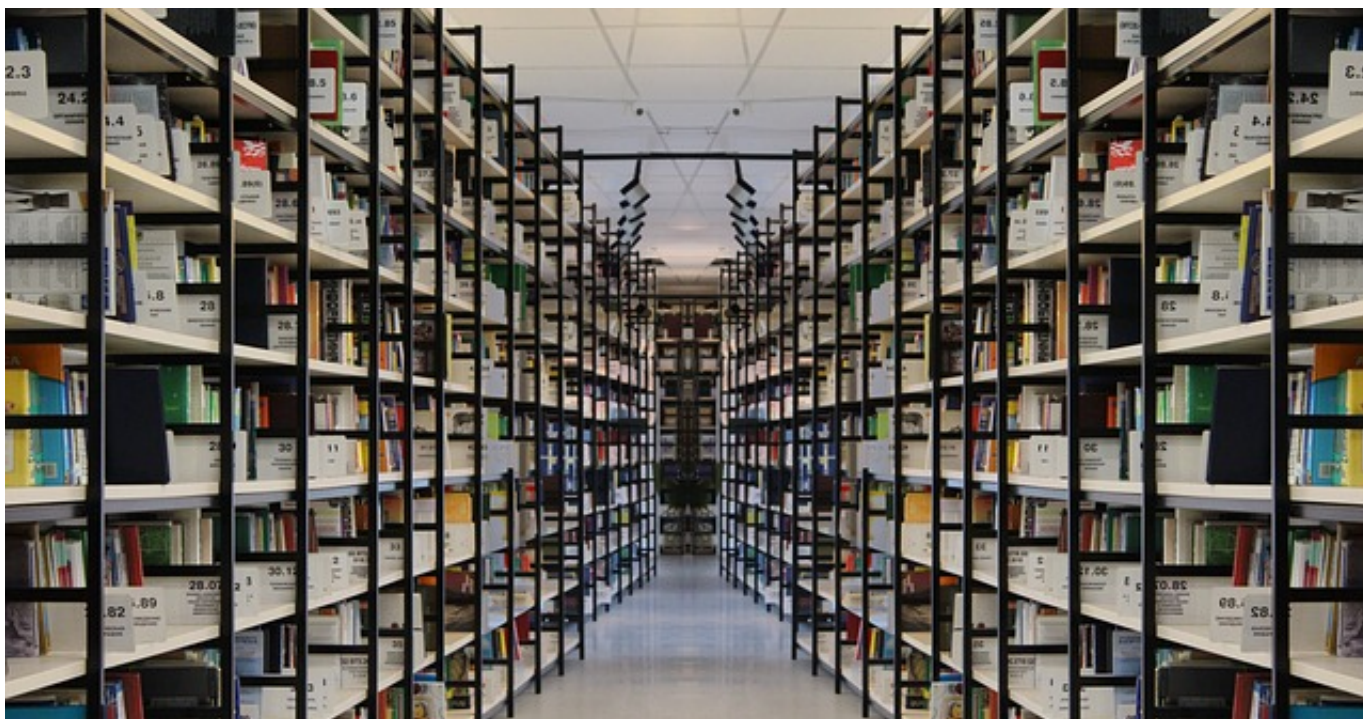
KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.  
<https://www.ks7000.net.ve>

---

- «».

## En idioma inglés

- [«Encoding your WiFi access point password into a QR code»](#).
- [«pure JS WiFi QR Code Generator»](#).
- [«Share Your Wi-Fi Password with a QR Code in Android 10»](#).



Crédito de la imagen [Gerd Altmann](#), [trabajo](#), licencia de uso: [Pixabay](#)

---