

## **Kit de programación Código Pi**

**Hola Raspberry Pi**



## Autoridades

### **Presidente de la Nación**

Mauricio Macri

### **Jefe de Gabinete de Ministros**

Marcos Peña

### **Ministro de Educación, Cultura, Ciencia y Tecnología**

Alejandro Finocchiaro

### **Secretario de Gobierno de Cultura**

Pablo Avelluto

### **Secretario de Gobierno de Ciencia, Tecnología e Innovación Productiva**

Lino Barañao

### **Titular de la Unidad de Coordinación General del Ministerio de Educación, Cultura, Ciencia y Tecnología**

Manuel Vidal

### **Secretaria de Innovación y Calidad Educativa**

Mercedes Miguel

### **Subsecretario de Coordinación Administrativa**

Javier Mezzamico

### **Directora Nacional de Innovación Educativa**

María Florencia Ripani

ISBN en trámite



Este material fue producido por el Ministerio de Educación, Cultura, Ciencia y Tecnología en base a contenidos provistos sin cargo por la Fundación Raspberry Pi mediante licencias Creative Commons y han sido desarrollados en función de los Núcleos de Aprendizajes Prioritarios de educación digital, programación y robótica y los recursos tecnológicos propuestos en el marco del Plan Aprender Conectados.

# Índice

¿Qué vamos a hacer?.....	5
¿Qué vamos a aprender?.....	5
¿Qué es una línea de comandos?.....	5
Hola mundo.....	7
Hello video .....	8
Hello triangle .....	9
Hello teapot .....	11
Hello audio .....	13
¿Qué sigue?.....	14

## Hola Raspberry Pi

### ¿Qué vamos a hacer?

¿Qué te parece si ejecutamos algunos programas de ejemplo que muestren las capacidades de tu Raspi?

Raspbian trae una serie de programas de ejemplo que podés compilar y ejecutar. Desde hola mundo en sólo texto, hasta reproducción de videos 1080p HD, teteras en tres dimensiones, y fractales animadas en tiempo real.

Estas son excelentes maneras de descubrir las posibilidades de tu Raspi, y de ganar familiaridad navegando el sistema, y ejecutando aplicaciones en la línea de comandos.

### ¿Qué vamos a aprender?

Siguiendo esta guía vas a aprender:

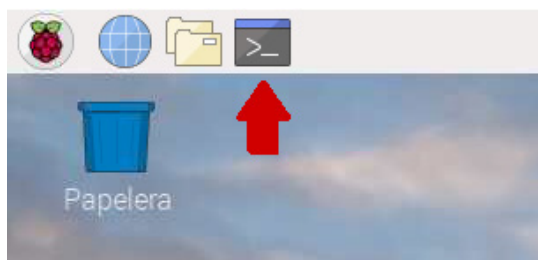
- Cómo usar la línea de comandos
- Cómo navegar el sistema de archivos
- Cómo compilar código
- Cómo ejecutar código

### Lo que necesitás

- Solo necesitás tu Raspberry Pi con la última versión de Raspbian instalada.

### ¿Qué es una línea de comandos?

Iniciá tu Raspberry Pi y abrí una ventana de terminal:



```
pi@raspberrypi ~ $ _
```

El texto de acá arriba es un símbolo de sistema. Una CLI (interfaz de línea de comandos) es verdaderamente una manera muy rápida y eficiente de usar una computadora.

Para iniciar, ingresá el comando de más abajo para navegar a la carpeta `hello_pi`, donde se encuentran los programas de ejemplo. **PISTA:** Podés usar la tecla `TAB` para auto-completar comandos a medida que los vas ingresando.

```
cd /opt/vc/src/hello_pi
```

El símbolo de sistema debería verse así, azul muestra dónde te encontrás en el sistema operativo de tu Raspi.

```
pi@raspberrypi /opt/vc/src/hello_pi $ _
```

Si ingresás `ls` y presionás `enter`, verás una lista de carpetas. Hay una por cada programa. Antes de que las puedas ejecutar, deberás compilarlas. No te preocupes si no entendés por qué es necesario hacer esto; solo seguí las instrucciones por ahora, y aprenderemos más acerca de esto más adelante.

Hay un pequeño script en la carpeta `hello_pi` llamado `rebuild.sh`, que se encargará de hacer la compilación por vos. Ingresá el siguiente comando para ejecutarlo. ¡Podés ignorar lo que aparezca en pantalla por el momento!

```
./rebuild.sh
```

Ahora aparecerá un montón de texto en pantalla, pero en este ejercicio podés ignorarlo. Es solamente la salida del compilador a medida que procesa el código del programa de ejemplo. Esperá a que vuelva el símbolo de sistema antes de continuar.

Ahora estamos listos para ejecutar algunos ejemplos.

## Hola mundo

Primero, hagamos una prueba rápida que asegure que la compilación anterior haya funcionado correctamente. Este programa (bastante aburrido) sólo mostrará el texto `Hello world!`, pero si funciona correctamente entonces sabremos que los demás ejemplos funcionarán.

Ingresa los siguientes comandos para entrar a la carpeta `hello_world` y listar sus contenidos.

```
cd hello_world  
ls
```

Notarás que el archivo `.bin` aparece en verde. Esto es porque es un archivo ejecutable. Esto significa que ese es el archivo que debemos abrir para ejecutar nuestro programa.

Usá el siguiente comando para ejecutar el ejemplo. Tenés que especificar el directorio actual por medio del `./`, de otro modo el nombre de archivo que escribas sería buscado en las carpetas de sistema de Linux solamente.

```
./hello_world.bin
```

## Hello video

Esto mostrará un video clip de 15 segundos, en calidad full HD 1080p, sin sonido. La intención aquí es mostrar las capacidades de decodificación y reproducción de video. ¡Verás que la reproducción es muy fluida!



Ingresá los siguientes comandos para navegar a la carpeta `hello_video` y listar sus contenidos.

```
cd ..  
cd hello_video  
ls
```

Notarás que hay un archivo `.bin` nuevamente. Este programa necesita que le indiquemos qué video vamos a reproducir cuando lo ejecutemos: en este caso será el archivo `test.h264` (h264 es un tipo de codec de video). Necesitas especificar nuevamente el directorio actual por medio del `./`.

```
./hello_video.bin test.h264
```



## Hello triangle

Esto muestra un cubo girando, con distintas imágenes en cada lado. Esto se utiliza para demostrar gráficos Open GL ES (una librería de programación open-source para crear gráficos en 3 dimensiones).

Ingresa los siguientes comandos para navegar a la carpeta `hello_triangle` y listar sus contenidos.

```
cd ..  
cd hello_triangle  
ls
```

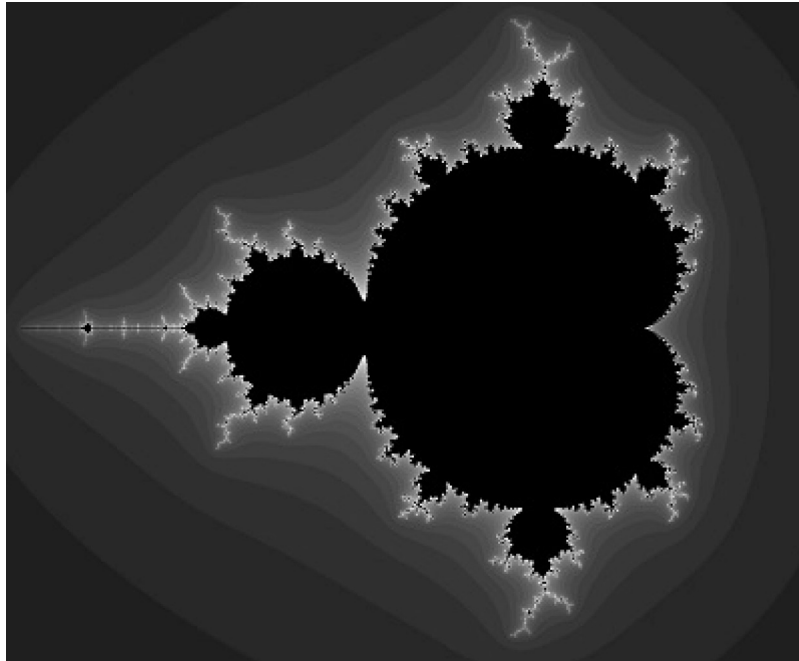
Notarás nuevamente que uno de los archivos es verde. Ese es el ejecutable. En este caso, no necesitamos ningún archivo adicional como en el programa anterior, así que podés ejecutar directamente el archivo `.bin`.

```
./hello_triangle.bin
```

Este ejemplo se ejecutará por siempre hasta que decidas cerrarlo. Podés hacer eso pulsando `Ctrl - C`.

## Hello triangle 2

Este programa muestra dos fractales superpuestas, una encima de la otra. Podés mover el mouse para cambiar la forma de la fractal en tiempo real. Este ejemplo también está pensado para demostrar gráficos de Open GL ES. Quizá puedas reconocer aquí la fractal de Mandelbrot.



```
cd ..  
cd hello_triangle2  
ls
```

¿Notás el archivo verde `.bin` ? ¡Ejecútalo!

```
./hello_triangle2.bin
```

Ahora mové el mouse, y vas a ver como cambia el fractal. Fijate si logras formar un círculo perfecto. Es un poco difícil, pero puede hacerse. Para salir del ejemplo, pulsá `Ctrl - C`.

## Hello teapot

Este programa muestra una tetera girando, con el videoclip del programa `hello_video` texturizado en su superficie. ¡Impresionante! Puede que reconozcas la tetera si estás familiarizado con un programa llamado Blender. Este programa muestra gráficos de Open GL ES y decodificación y reproducción de video simultáneamente.



```
cd ..  
cd hello_teapot  
ls
```

¿Notás el archivo verde `.bin`? ¡Ejecutalo!

```
./hello_teapot.bin
```

Puede que recibas el siguiente error cuando intentes ejecutar este programa:

```
Note: ensure you have sufficient gpu_mem configured
eglCreateImageKHR: failed to create image for buffer 0x1 target
12465 error 0x300c
eglCreateImageKHR failed.
```

No te preocupes: si ves este error, sólo tenés que cambiar una opción de configuración para que funcione.

El error significa que la GPU (unidad de procesamiento de gráficos) no tiene suficiente memoria para ejecutar el programa. El GPU es el encargado de dibujar las imágenes 3D en la pantalla (como una placa de video en una PC para juegos). El Raspberry Pi comparte su memoria RAM entre el CPU y el GPU, y por defecto está configurada para sólo dar 64MB de RAM al GPU. Si subimos este límite a 128, eso debería corregir el problema.

Para ello, ejecutá el siguiente programa:

```
sudo raspi-config
```

Esto mostrará un menú con un fondo azul. Realiza las siguientes acciones:

- Andá a Opciones Avanzadas.
- Andá a Memory Split.
- Borrá `64` y poné `128` . Presioná `enter` .
- Andá a Finalizar
- Hacé clic en Si para reiniciar.

Una vez que hayas iniciado sesión nuevamente, ingresá el siguiente comando para volver al programa `hello_teapot` :

```
cd /opt/vc/src/hello_pi/hello_teapot
```

Ahora intentá ejecutarlo de nuevo, verás que esta vez funciona.

```
./hello_teapot.bin
```

Este ejemplo se ejecutará por siempre hasta que decidas cerrarlo. Podés hacer eso pulsando `Ctrl + C`.

## Hello audio

Este programa solamente prueba la salida de audio. El mismo reproduce una onda senoidal, que hace un sonido similar a WOO WOO WOO.

```
cd ..  
cd hello_audio  
ls
```

¿Notás el archivo verde `.bin`? ¡Ejecutalo!

```
./hello_audio.bin
```

Esto reproducirá el sonido a través del conector de auriculares de tu Raspi. Si estás usando una pantalla HDMI, podés hacer que el sonido se reproduzca por HDMI agregando un `1` al comando.

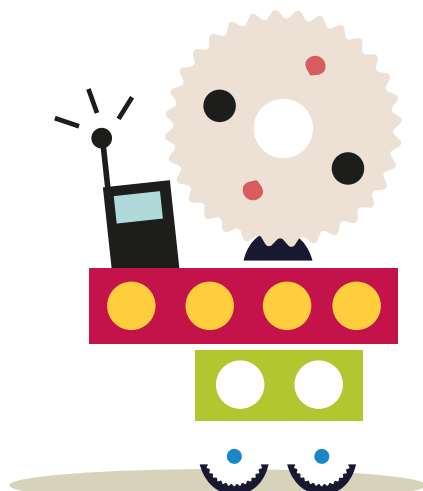
```
./hello_audio.bin 1
```

Este ejemplo se ejecutará por siempre hasta que decidas cerrarlo. Podés hacer eso pulsando `Ctrl - C`.

## ¿Qué sigue?

- A esta altura deberías poder navegar hasta la carpeta anterior a `hello_pi` folder (usando `cd ..`) y luego a la de alguna de los programas de muestra (usando `cd hello something`).
- Intentá probar los demás programa. `hello videocube` es un buen lugar para empezar.

La fundación Raspberry Pi suministra contenidos de aprendizaje de programación sin cargo. Encuentre más información en <https://projects.raspberrypi.org/en/> (inglés)



**APRENDER  
CONECTADOS**



Ministerio de Educación,  
Cultura, Ciencia y Tecnología  
**Presidencia de la Nación**