



Programación para Niños

*¡Aprende a escribir programas
de computadora!*

Peter Armstrong
Debi y Pablo Bontti
Obie Fernandez

Programación para Niños

¡Aprende a escribir programas de computadora!

Peter Armstrong, Obie Fernandez y Debi & Pablo Bontti

Este libro está a la venta en

<http://leanpub.com/programacionparaninos>

Esta versión se publicó en 2014-03-29



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

©2014 Peter Armstrong, Obie Fernandez y Debi & Pablo Bontti

Índice general

Introducción	i
Capítulo 1: ¡Hola línea de comandos!	1
Ejercicios	12
Capítulo 2: ¡Hola Mundo!	13
Ejercicios	15
Esto es una muestra	16
Para los Padres	17

Introducción

¡Este libro te enseñará a escribir programas de computadora!

Tendrás que usar un Mac para seguir los capítulos. Los programas son cortos, así que podrás escribirlos tú solo, inclusive si todavía no sabes escribir bien en el teclado.

Este libro tiene un montón de capítulos chiquitos y cada capítulo trata de una idea diferente.

Al final de cada capítulo, habrá ejercicios para que hagas. ¡Es muy importante que hagas todos! Hacer los ejercicios asegúrate de saber todo el material del capítulo. Las respuestas para todos los ejercicios están al final del libro, y hay enlaces para las respuestas a todos los ejercicios.

Si tus padres quieren saber más sobre este libro, pueden leerlo contigo. O, pueden simplemente leer la sección [Para los Padres] (#paralospadres) al final del libro.

Si vas a usar su computadora, no estaría de más que se sienten contigo. ¡De seguro aprenderán algo también!

Si estas siguiendo con tu Mac, vas a querer leer el archivo PDF que descargaron tus padres. Si haces doble-clic en el archivo PDF, abrirá en un programa llamado “Vista Previa”. Vas a estar cambiando de ida y vuelta entre leer este libro en “Vista Previa” y escribiendo cosas en “Terminal”. Si hiciste clic en “Vista Previa” para cambiar de página, tendrás que hacer clic en la ventana de “Terminal” para enfocararlo antes de escribir cualquier comando.

Si te irrita estar cambiando entre Vista Previa y Terminal, hay un par de cosas que puedes hacer. Si tus padres tienen impresora, pueden imprimir el libro. O, si tienen un iPad, pueden descargar el archivo EPUB, añadirlo a iTunes y luego sincronizarlo al iPad.

¡Comencemos!

Capítulo 1: ¡Hola línea de comandos!

¡Este capítulo es acerca de la línea de comandos!

Este libro es para niños como tu que usan una computadora Mac, en casa o en la escuela.

Normalmente utilizas un Mac haciendo clic en las cosas con un ratón. En este capítulo aprenderás una manera diferente de usar el Mac. Se llama la línea de comandos, y solo usarás el teclado.

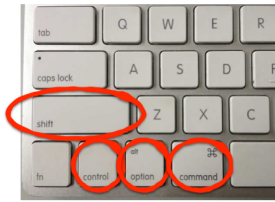
Había un vez, hace unos 30 años, lo único que tenían las computadoras era la línea de comandos.

Aunque la línea de comandos es sencilla, es muy poderosa. Si sabes usarla, serás como un mago que puede teclear hechizos extraños y hacer que tu computadora haga cosas asombrosas.

Tranquilízate, no tiene porque asustarte. Si te equivocas en algo, ¡no vas a romper nada!

Primero vamos a aprender los principios básicos.

Ya sabes lo que es un teclado. Pero hay algunas teclas que quizás no hayas usado antes. Estas son las teclas Comando, Opción, Control, y Shift.

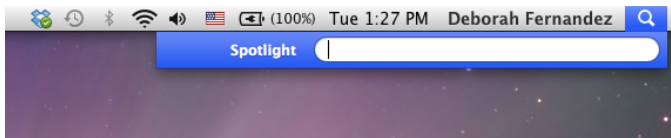


Las teclas Comando, Opción, Control, y Shift

Vamos a iniciar un programa que se llama Terminal. Terminal es lo que te permite usar la línea de comandos en tu Mac.

Para iniciar Terminal, mantén oprimida la tecla Comando y oprime la barra espaciadora. (Una manera mas corta de decir esto es decir “teclea **Comando + Barra Espaciadora**”. Así que eso diré de ahora en adelante.)

Teclear **Comando + Barra Espaciadora** abre un programa llamado Spotlight en la esquina derecha superior de tu pantalla.

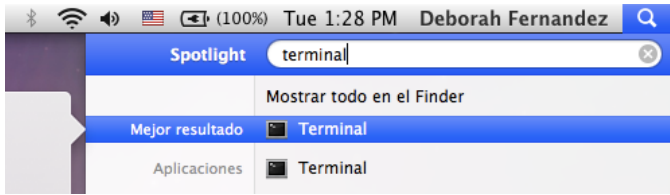


Spotlight

Spotlight te permite teclear los nombres de aplicaciones que quieres ejecutar.

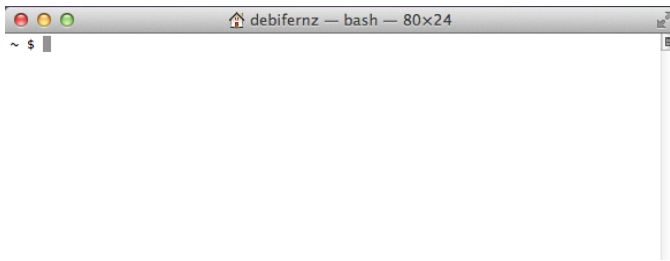
Vamos a iniciar Terminal. Así que teclea **terminal** en Spotlight y oprime la tecla Intro (↵).

Posiblemente lo veas aparecer en una lista desplegable después de teclear **term**. Si así es, puedes hacerle clic a esa opción en vez de terminar de teclear **terminal**.



Iniciando Terminal

Verás Terminal, que luce así.



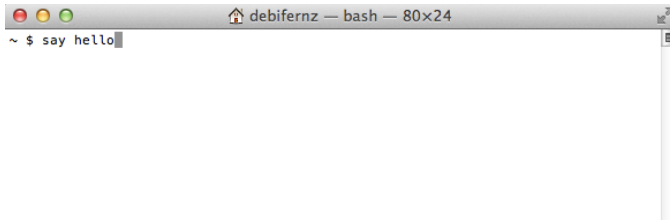
Terminal

Terminal es la línea de comandos del Mac.

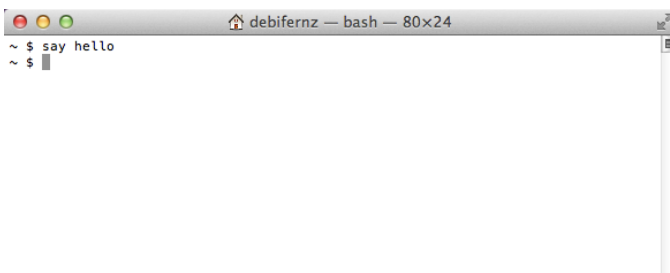
(No te preocupes por toda la ~[]\$ al principio de la línea. Eso es solamente como luce la "señal de inicio" en mi computadora, y es posible personalizar tu señal de inicio. Tu señal de inicio puede lucir así > ó así ~.)

Antes de comenzar, vamos a hacer que la línea de comandos nos diga hello. Para hacer esto vamos a ejecutar el programa **say** (say significa decir en inglés).

Teclea **say hello** en Terminal.



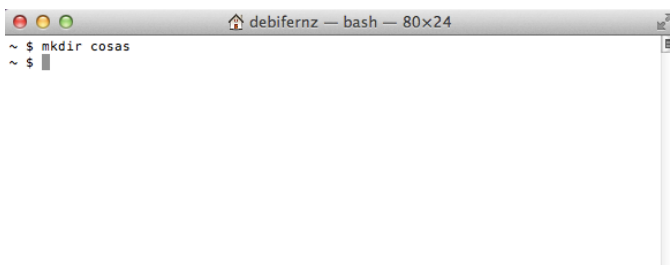
Entonces, oprime la tecla Intro para ejecutar el comando.



¡Espero que tu Mac te haya saludado con un “hello”! Si no, asegúrate de que el volumen este encendido e inténtalo otra vez.

Por cierto, luego de ingresar cualquier comando en Terminal necesitas oprimir Intro para ejecutarlo.

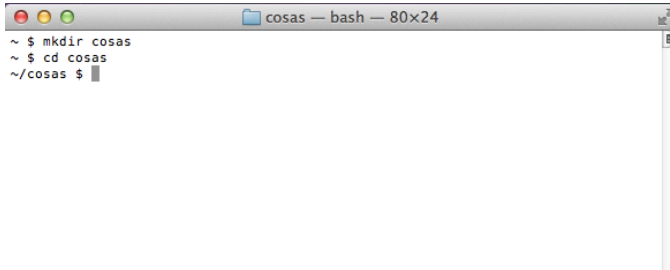
Ahora, comencemos por crear una carpeta para todas las cosas que vamos a hacer en este libro. Teclea **mkdir cosas** en Terminal y oprime Intro.



El comando **mkdir** se lee “make dir” y quiere decir “crear directorio”. Directorio es otra palabra para carpeta. Así que acabas de crear una

carpeta con el nombre **cosas**.

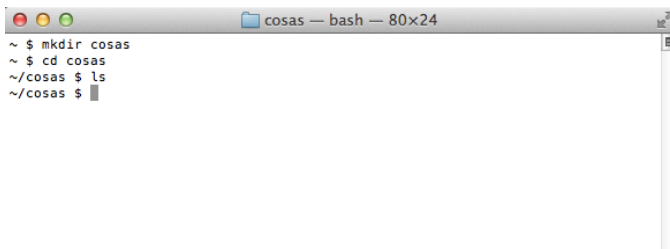
Lo siguiente que haremos es entrar en esa carpeta. Teclea **cd cosas** en Terminal y oprime Intro.



```
cosas — bash — 80x24
~ $ mkdir cosas
~ $ cd cosas
~/cosas $
```

El comando **cd** quiere decir “cambiar directorio”. Así que estás cambiando al directorio llamado cosas que acabas de crear.

Miremos adentro de este directorio. En Terminal, teclea **ls** (L minúscula y S minúscula) y oprime Intro.



```
cosas — bash — 80x24
~ $ mkdir cosas
~ $ cd cosas
~/cosas $ ls
~/cosas $
```

¡Teclear **ls** no muestra nada!

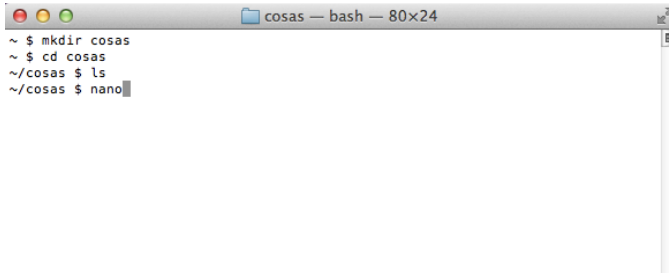
El comando **ls** significa “lista”. Quiere decir que muestre los archivos y las carpetas adentro de esta carpeta. No hay nada adentro de esta carpeta ya que lo acabamos de crear. Así que por eso teclear **ls** no muestra nada.

Ahora, cambiemos eso al crear un archivo.

Vamos a usar un programa que se llama un editor de texto para editar el archivo. Un editor de texto te permite teclear palabras en un

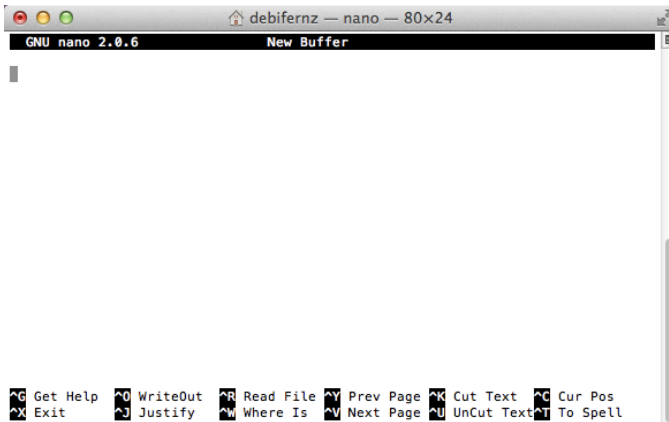
archivo, y guardarlas como guardarías un juego. Solamente vamos a usar un editor de texto pequeño que se llama **nano**. Nano quiere decir muy pequeño, y es un buen nombre porque **nano** es un editor de texto muy pequeño y sencillo.

Así que, en Terminal, teclea **nano** y oprime la tecla Intro.

A terminal window titled 'cosas — bash — 80x24'. The prompt is '~'. The user enters '\$ mkdir cosas', '\$ cd cosas', and '/cosas \$ ls'. Then the user enters '/cosas \$ nano' and the cursor is at the end of the line.

```
~ $ mkdir cosas
~ $ cd cosas
~/cosas $ ls
~/cosas $ nano
```

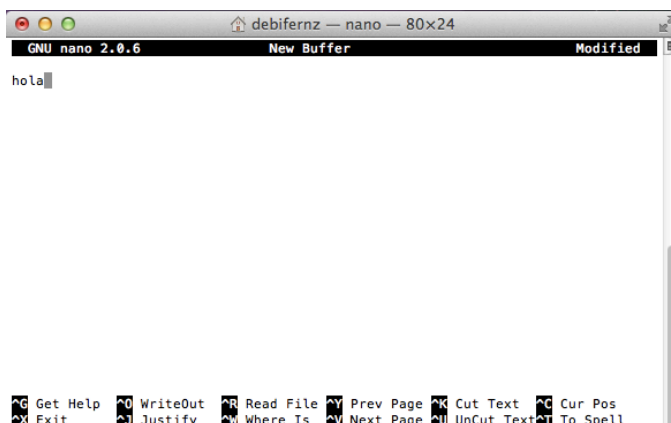
Esto inicia nano. La pantalla luce así.

The GNU nano 2.0.6 editor interface. The title bar shows 'debifernz — nano — 80x24'. Below the title bar is a black bar with 'GNU nano 2.0.6' and 'New Buffer' in white. The main area is empty with a cursor at the top left. At the bottom is a status bar with various shortcuts: ^G Get Help, ^O WriteOut, ^R Read File, ^V Prev Page, ^K Cut Text, ^C Cur Pos, ^X Exit, ^J Justify, ^W Where Is, ^N Next Page, ^U UnCut Text, ^T To Spell.

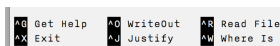
```
GNU nano 2.0.6 New Buffer

^G Get Help  ^O WriteOut  ^R Read File  ^V Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^N Next Page  ^U UnCut Text ^T To Spell
```

En nano, teclea **ho1a**. No necesitas oprimir la tecla Intro.



Puedes ver que nano te ofrece ayuda al mostrarte la lista de comandos que puedes teclear al mantener oprimida la tecla `control` y teclear una letra. (Nano te muestra la tecla `control` como el signo de intercalación (^), pero no es `shift + 6`, es la tecla `control`.)

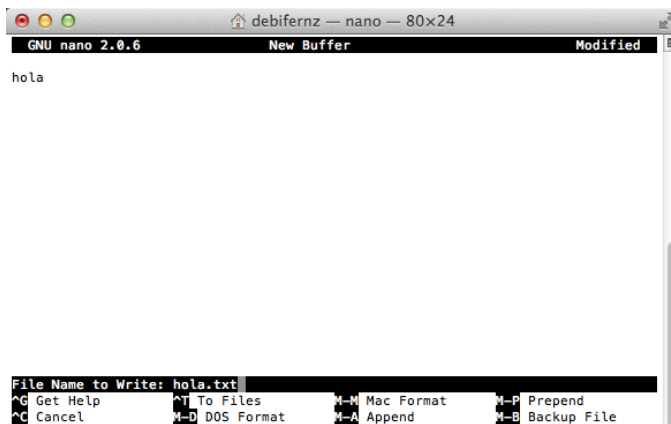


Así que para guardar el archivo, mantén oprimida la tecla **Control** y teclea la **o**.

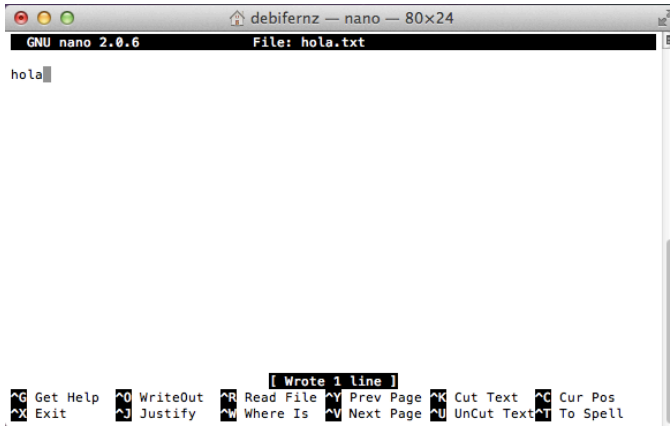
Nano te preguntará como quieres que se llame el archivo que estás guardando, al mostrar texto que dice "File Name to Write".



Teclea `hola.txt` y oprime la tecla Intro.



Nano guardará el archivo y te dirá que era una línea al decir "Wrote 1 line" (Eso es: "Escribieron 1 línea" en inglés).



```
debifernz — nano — 80x24
GNU nano 2.0.6      File: hola.txt
hola
```

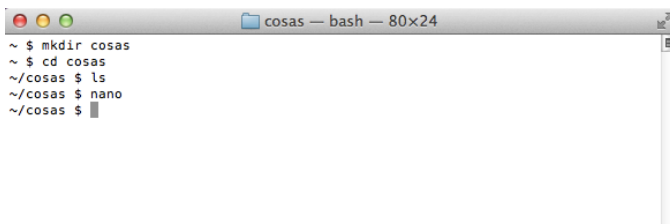
[Wrote 1 line]

Get Help	WriteOut	Read File	Prev Page	Cut Text	Cur Pos
Exit	Justify	Where Is	Next Page	UnCut Text	To Spell

(Si habías oprimido Intro después de teclear “hola” antes, dirá “Wrote 2 lines”. Eso está bien, no te preocupes.)


Ahora que hemos guardado nuestro archivo, salgamos de nano al mantener oprimido la tecla Control y tecleando **x**. (Una manera más corta de decir esto es decir “teclea **Control** + **x**”. Así que eso diré de ahora en adelante.)

Estarás de nuevo en la línea de comandos adentro de Terminal.



```
cosas — bash — 80x24
~ $ mkdir cosas
~ $ cd cosas
~/cosas $ ls
~/cosas $ nano
~/cosas $
```

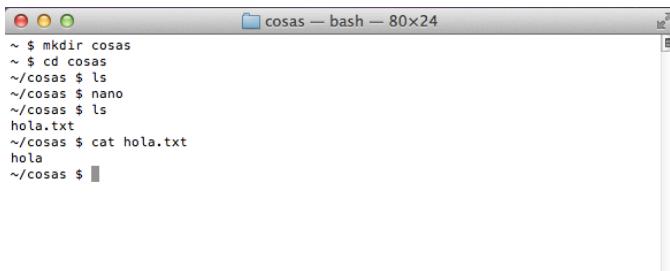
Miremos otra vez adentro de nuestra carpeta **cosas**. Recuerda que la ultima vez que tecleamos **ls** vimos que estaba vacía. Teclea **ls** en Terminal otra vez. Verás el archivo **hola.txt** que creaste.



```
cosas — bash — 80x24
~ $ mkdir cosas
~ $ cd cosas
~/cosas $ ls
~/cosas $ nano
~/cosas $ ls
hola.txt
~/cosas $
```

Miremos adentro de este archivo. Teclea **cat hola.txt** en Terminal y oprime la tecla Intro.

Aquí, **cat** no es un gato. En verdad, es un comando que te muestra lo que hay adentro de tu archivo. Ya que tecleamos la palabra hola, esto es lo que verás.



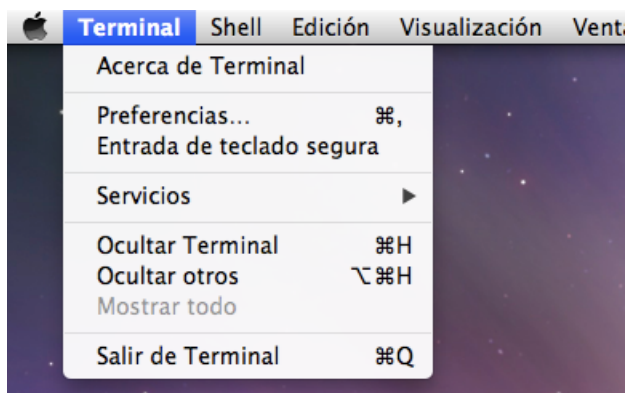
```
cosas — bash — 80x24
~ $ mkdir cosas
~ $ cd cosas
~/cosas $ ls
~/cosas $ nano
~/cosas $ ls
hola.txt
~/cosas $ cat hola.txt
hola
~/cosas $
```

Así que el archivo **hola.txt** tenía “hola” adentro, por eso cuando tecleaste **cat hola.txt** viste “hola”.

¡Ahora sabes usar algunos de los comandos básicos de la línea de comandos! Aprendiste un montón de comandos extraños, cosas como **mkdir**, **cd**, **ls** y **cat**. Lo genial es que ¡estos comandos en realidad simplemente son programas escritos por otras personas!

In the next chapter, **you** will create **your** first program, which **you** will also be able to run from the command line!

Por último, vamos a salir de Terminal. Teclea **Comando + Q** o selecciona Salir de Terminal del menú Terminal.



Por cierto, si alguna vez de equivocas mucho en la línea de comandos y no sabes como seguir, siempre puedes simplemente salir de Terminal y abrirlo nuevamente.

¡Asegúrate de hacer los ejercicios en la próxima página antes de continuar!

Ejercicios

1. Inicia Terminal y `cd` al directorio `cosas`.
2. Usa `nano` para crear un archivo con el nombre `bravo.txt` que contiene el texto "bravo". Sal de `nano` cuando termines.
3. Usa la aplicación `cat` para mirar el contenido de tu archivo `bravo.txt`.

1. Sal de Terminal

Puedes ver las [respuestas a los ejercicios](#), o simplemente continuar al siguiente capítulo.

Capítulo 2: ¡Hola Mundo!

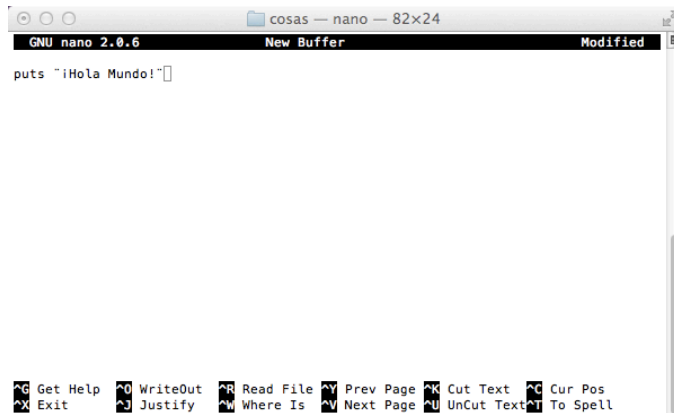
¡Escribamos nuestro primer programa de computadora!

Inicia Terminal, `cd` al directorio `cosas` y después inicia `nano`. (Si esto no tiene sentido, mira los [Ejercicios Capítulo 1](#).)

Vamos a crear un programa muy sencillo. Teclea `puts "¡Hola Mundo!"` en `nano`.

No necesitas oprimir la tecla `Intro`. Luego hablaré mas del significado de `puts`.

Para teclear unas comillas como `"`, mantén oprimida la tecla `Shift` y teclea la `\`. No hay diferencia entre las del principio y las del final.



```
GNU nano 2.0.6      cosas — nano — 82x24
New Buffer      Modified
puts "¡Hola Mundo!"

Get Help  WriteOut  Read File  Prev Page  Cut Text  Cur Pos
Exit      Justify   Where Is  Next Page  UnCut Text To Spell
```

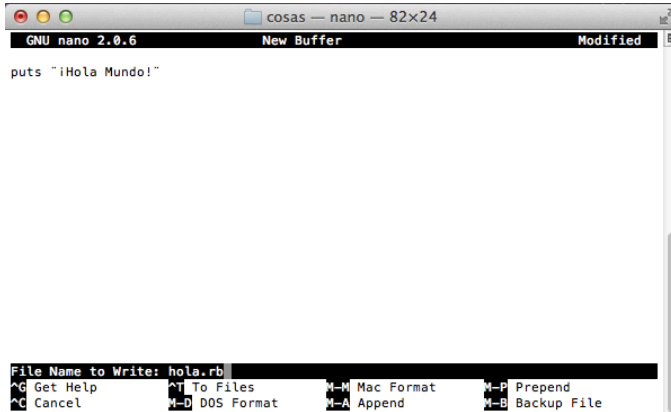
¡Eso es todo!

Teclea `Control + o` para guardar el archivo.

Nano te preguntará qué nombre darle al archivo que estás guardando al decir “File Name to Write:”

Teclea `hola.rb` y oprime la tecla `Intro`. Mientras escribes el nombre

del archivo lo verás al lado de “File Name to Write” en la parte de abajo de Terminal.

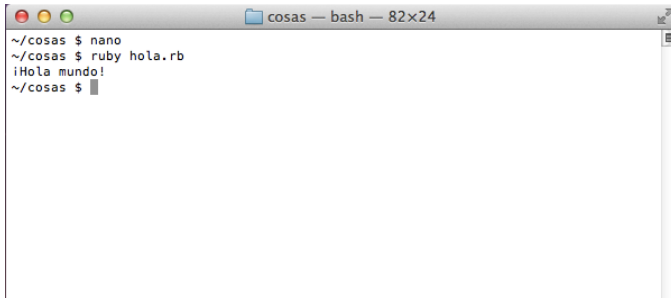


Nano guardará el archivo y te dirá que era una línea al decir “Wrote 1 line”. (Si habías oprimido Intro, dirá “Wrote 2 lines”. Eso está bien también.)

Ahora que hemos guardado nuestro archivo, teclea **Control + x** para salir de nano.

¡Ejecutemos nuestro programa!

En Terminal, teclea **ruby hola.rb** y oprime **Intro**. Necesitarás un espacio entre **ruby** y **hello.rb**.

A screenshot of a terminal window titled "cosas — bash — 82x24". The terminal shows a sequence of commands and their output: first, the user enters `nano` at the prompt `~/cosas $`; then, they enter `ruby hola.rb` at the prompt `~/cosas $`, which results in the output `¡Hola mundo!`; finally, the prompt returns to `~/cosas $` after the output.

¡Tu Primer Programa de Computadora!

¡Felicitaciones! Has escrito tu primer programa de computadora. Como acabas de ver, **puts** imprimio algo en la pantalla.

Pero, ¿que fue esa palabra extraña “ruby”?

Ruby es un lenguaje de programación. Tú hablas español, pero hay muchos otros lenguajes que personas hablan. Igualmente, hay muchos lenguajes que puedes usar para decirle a una computadora qué hacer. Ruby es uno de los más faciles de usar, y tu Mac viene con Ruby ya instalado.

¡El programa que escribiste fue un programa de Ruby! La extensión de archivo (lo que está detrás del `.` en el nombre del archivo) para programas de Ruby es `rb`, así que le dimos el nombre `hola.rb`.

En el próximo capítulo, ¡jugaremos con Ruby en su propia línea de comandos!

Ejercicios

1. Escribe y ejecuta un programa Ruby que se llame `oye.rb` que imprima `¡Oye!`.

Puedes ver las [respuestas a los ejercicios](#), o simplemente continuar al siguiente capítulo.

Esto es una muestra

Esta es una muestra de un libro más largo, que está disponible a la venta en Leanpub en [<https://leanpub.com/programmingforkids>] (<https://leanpub.com/programmingforkids>). El libro completo tiene 13 capítulos. La versión en Inglés tiene más de 170 páginas y más de 19.000 palabras. 30% de todos los derechos van a Watsi, que es una gran caridad!

Para los Padres

Éste es un libro para leer con su hijo , o para que su niño lea por sí mismo.

Lo escribí para enseñar a mi hijo de nueve años los fundamentos de la programación. Él ha estado jugando videojuegos por muchos años, y quiere aprender a programar para poder hacer sus propios videojuegos algún día. El propósito de este libro es ayudarlo a dar el primer paso. (No, no es que te enseñe cómo crear los tipos de videojuegos que le gustan a los niños de nueve años, ¡eso lleva mucho más conocimiento!) Para un niño que desea aprender a programar computadoras este es el mejor “primer libro”. Ese es el propósito de esta publicación. La programación es una buena habilidad para cultivar, independientemente de la ocupación que su hijo finalmente elija como adulto. (Opinaría que es inclusive mucho más importante que mucha de las matemáticas que se aprenden en la escuela secundaria, por ejemplo.) Es más, aprender a programar computadoras enseña un rigor y una disciplina de pensamiento que es útil en cualquier campo. Este libro existe con el objetivo de mostrarle al niño que si puede programar computadoras y ayudarlo a dar el primer paso.

A excepción de este apéndice, este libro está escrito para el nivel de un niño en la escuela primaria. Esta pensado para niños entre las edades de nueve y catorce años. En América del Norte, eso sería del 3º al 8º grado. Mientras escribo los capítulos del libro, contemplo a mi hijo avanzar por los capítulos. (Personalmente creo que no es un buen libro para niños de siete u ocho años: Comencé a escribir este libro cuando mi hijo tenía siete años de edad, pero él no estaba preparado todavía, así que pausé la escritura por dos años. Si su hijo tiene siete u ocho años, creo que un libro como Scratch es una mejor opción)

El libro comienza desde cero, sin asumir que el niño tenga algún conocimiento previo de programación. Los ejemplos son tan cortos como sea posible, pues supongo que el lector todavía no puede teclear bien. (Mi hijo todavía necesita mirar el teclado para escribir, por eso cuando escribía ejemplos muy largos, enseguida me presentaba las quejas).

Para un niño que desea aprender programación, como primer libro, este es el mejor. Luego de este libro, podrán seguir el rumbo que más les interese.

Este libro está escrito asumiendo que su niño está usando una Mac. Creo que una Mac es la mejor computadora para que los niños aprendan a programar. Al ser este un libro para niños, no puedo escribirlo genéricamente para abarcar Mac, Windows y Linux. Tuve que elegir un sistema operativo y hacer que el niño siga las instrucciones palabra por palabra.

Además de enseñar programación, el libro también enseña el uso básico de la línea de comandos en una Mac, a la que se accede a través del programa Terminal. Los programadores reales utilizan la línea de comandos todos los días. Si usted quiere aprender programación, deberá utilizar la terminal y los archivos. Sí, una actividad simpática e instructiva es jugar en un navegador web en sitios como Codecademy, sin embargo, esa actividad es muy diferente de lo que hacen los programadores reales. Además, aunque es mas fácil, es de alguna manera menos gratificante.

Si su niño estará usando **su** Mac, le recomiendo **enfáticamente** que se siente a su lado y lo acompañe en el proceso. Por ejemplo, no estoy planeando enseñar el comando para borrar archivos, ¡que por cierto es bastante corto!

Los ejemplos están en Ruby. Ruby es un lenguaje de programación muy simple. Si alguna vez has oído hablar de sitios web construidos en “Ruby on Rails”, usted ha oído hablar de Ruby: es el lenguaje de programación en el cual está escrito Rails. Sin embargo, este no es un

libro sobre cómo aprender Ruby. Los ejemplos podrían fácilmente haber sido escrito en JavaScript, CoffeeScript o Python.

Finalmente, ¡me interesa muchísimo su opinión! ¿Se trabó su hijo en alguna parte?, ¡Me encantará escuchar cualquier sobre el libro! Por favor, ¡escribame un correo electrónico a peter@leanpub.com!