

## Programando con Ruby

<b>Programando con Ruby</b>	<b>1</b>
¿Qué aprenderás?	2
Formas de trabajar con Ruby	3
Trabajando con Ruby a través de IRB	3
Ingresando a IRB (Ruby interactivo)	4
Probando IRB	4
¿Cuáles son los otros números que aparecen en la pantalla?	5
Realizando nuestras primeras pruebas	5
Saliendo de IRB	5
Trabajando con Ruby desde el editor de texto	5
Extensión de los archivos .rb	6
Nuestro primer programa en el editor	6
Ejecutando el programa	6
Mostrando el resultado	6
Ejecutando el programa	7



**¡Comencemos!**

## ¿Qué aprenderás?

- Conocer y utilizar IRB para ejecutar código Ruby.
- Conocer limitaciones de IRB para ejecutar código Ruby.
- Ejecutar scripts de Ruby desde línea de comandos.
- Leer la salida de un script ejecutado.

**¡Vamos con todo!**



## Formas de trabajar con Ruby

Existen distintas formas de utilizar Ruby.

1. A través de un programa llamado Ruby Interactivo o [IRB](#).
2. Utilizando nuestro editor de texto y ejecutándolo con Ruby.

Con IRB podemos escribir código Ruby directamente en nuestro terminal e ir evaluando instrucciones, lo cual es muy útil al momento de realizar pruebas, pero es ineficiente para escribir un programa completo.

Con el editor de texto podemos escribir el programa en un archivo .rb y luego ejecutarlo con Ruby. Esta es la forma tradicional de trabajar.

En esta unidad aprenderemos a trabajar de ambas formas.

## Trabajando con Ruby a través de IRB

Para ingresar a IRB debemos entrar a la terminal.

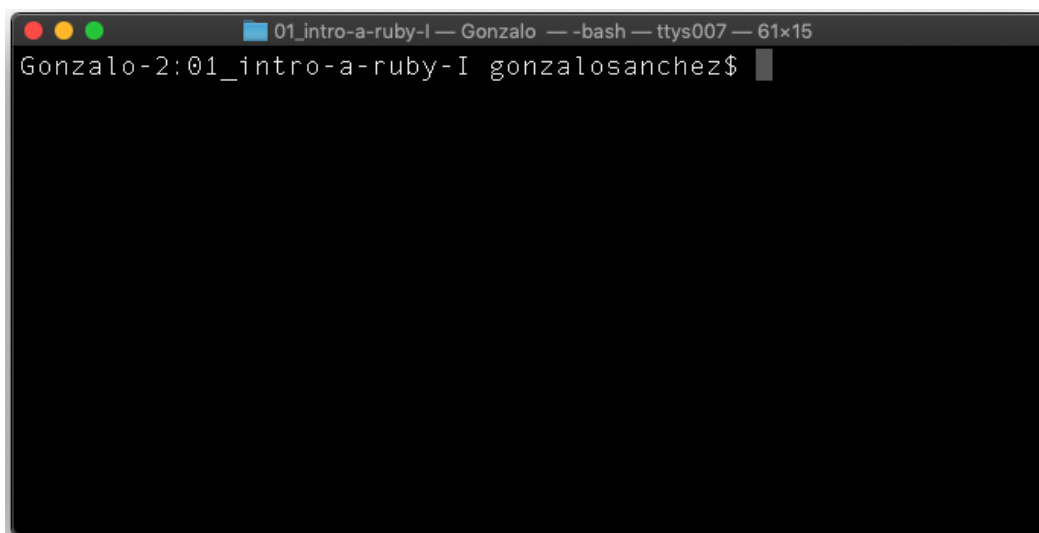


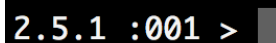
Imagen 1. Terminal.  
Fuente: Desafío Latam

Para eso abriremos la aplicación Command Prompt en Windows o terminal en Linux y OSX. Al abrirla veremos un signo peso \$ o en algunas ocasiones un signo > ó >>, símbolo que corresponde al prompt y corresponde al texto que indica que la terminal está lista para recibir una instrucción.

## Ingresando a IRB (Ruby interactivo)

Desde la terminal podemos ingresar a Ruby interactivo. Para ello escribiremos `irb` y luego enter.

Si todo funciona de manera correcta, deberíamos ver que el prompt cambió. Esto es porque ahora estamos ejecutando otro programa: IRB.

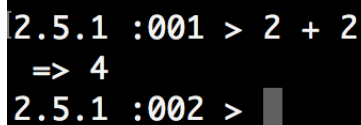


```
2.5.1 :001 > 
```

Imagen 2. IRB.  
Fuente: Desafío Latam

## Probando IRB

Dentro de este programa podemos escribir instrucciones válidas para Ruby. Por ejemplo, podemos escribir `2 + 2` y ver como resultado 4.



```
2.5.1 :001 > 2 + 2  
=> 4  
2.5.1 :002 > 
```

Imagen 3. Sumando en IRB.  
Fuente: Desafío Latam

¿Cuáles son los otros números que aparecen en la pantalla?

Antes de probar otras operaciones, revisemos algunos detalles que son importantes:

- **El número 2.5.1** corresponde a la versión de Ruby con la que estamos trabajando.
  - Si estás trabajando con la versión 2.5.3 se mostrará ese número.
- **El número 001** indica que es la primera línea que estamos escribiendo. Este número es autoincremental, es decir, irá aumentando a medida que vamos escribiendo y ejecutando instrucciones.
- El resultado de la línea ejecutada aparece acompañado de `=>`.

## Realizando nuestras primeras pruebas

Ejecuta, en IRB, las siguientes instrucciones:

- ¿Qué se obtiene al realizar la operación `5 / 2`?
- ¿Qué se obtiene al realizar la operación `'hola' + 'mundo'`?
- ¿Qué se obtiene al realizar la operación `[1,2,3].sum`?

Más adelante -en este módulo- profundizaremos en las instrucciones de Ruby.

## Saliendo de IRB

Para salir de IRB debemos escribir `exit`, y con eso volveremos al prompt del terminal.

## Trabajando con Ruby desde el editor de texto

Hay varios editores de texto gratuitos y muy buenos como [Atom](#), [Visual Studio Code](#) y [Sublime Text](#). Este último es de pago, pero ofrece versión gratuita.



Puedes utilizar el editor que más te guste, y si no sabes cuál escoger, utiliza Visual Studio Code, el cual es el que utilizaremos para los ejemplos de este módulo.

## Extensión de los archivos .rb

Para escribir un programa desde cero en Ruby en un editor de texto, necesitamos crear un archivo con extensión .rb.

## Nuestro primer programa en el editor

Para escribir nuestro primer programa en un editor de texto vamos a seguir las siguientes instrucciones:

1. Abrir nuestro editor de texto.
2. Escribir una instrucción sencilla.
3. Guardar el archivo con extensión .rb.

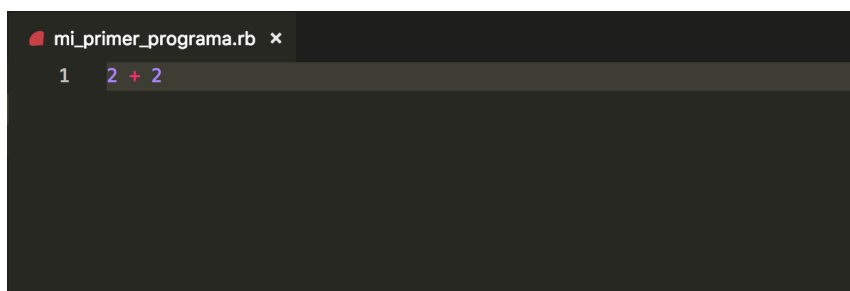


Imagen 4. mi\_primer\_programa.rb.  
Fuente: Desafío Latam

## Ejecutando el programa

A continuación, debemos abrir una terminal en la misma ubicación donde se encuentra nuestro archivo y ejecutar,

```
ruby nombre_archivo.rb
```

No obtendremos ningún resultado. Es normal.

## Mostrando el resultado

La razón por la que en IRB vimos un resultado, pero no cuando ejecutamos el script desde la terminal es que Ruby no muestra el resultado de cada línea. Si queremos ver este resultado,

debemos indicarle a Ruby de manera explícita que muestre el resultado en pantalla, lo cual logramos con la instrucción puts.



```
mi_primer_programa.rb x
1 puts 2 + 2
```

Imagen 5. Mostrando resultado con puts.  
Fuente: Desafío Latam

## Ejecutando el programa

Si ejecutamos nuevamente el script, veremos el resultado en pantalla:

4

Si utilizamos puts directamente en IRB veremos como respuesta # => nil. IRB siempre muestra el resultado de la ejecución de la última línea, sin embargo, algunas operaciones (como mostrar en pantalla) no tienen un resultado. nil corresponde al objeto nulo, la ausencia de algún valor.