**UNIVERSIDAD NACIONAL AUNTONOMA DE NICARAGUA**

**UNAN-León.**

**Facultad de Ciencias y Tecnología**

**Carrera:** Ingeniería en Telemática.

**Componente:** Software como un Servicio.

**Docente:** Ervin Montes

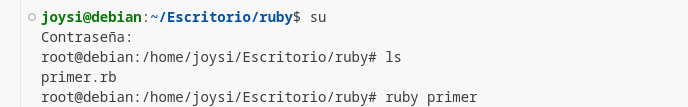
**Nombre:**

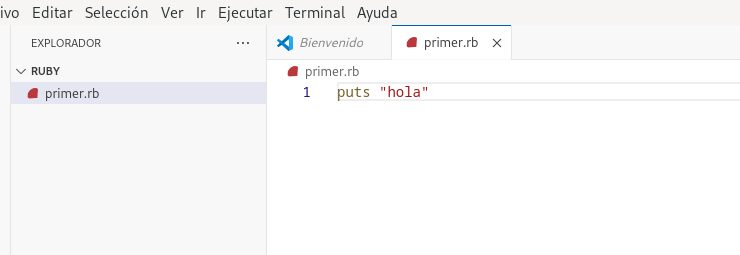
* Joysi Julissa García Soza.
* Rosa Arellys Orozco Lira

**Fecha:** 06/08/2024.

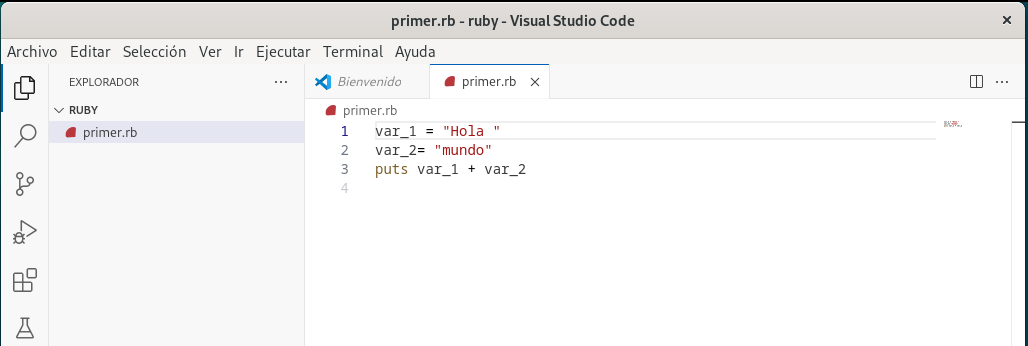
**¡A la libertad por la universidad!**

**1.Crear un directorio llamado ruby, donde se almacenarán los ejercicios que se llevarán a cabo a lo largo de esta guía.**

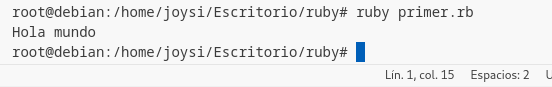




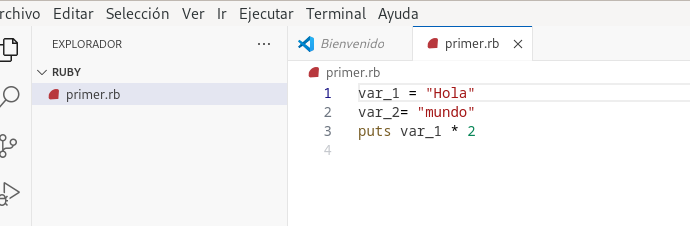
**2. Crear un programa primer\_programa.rb, se puede hacer desde el terminal o desde el editor de texto, asignar 2 variables de tipo String para luego imprimir por pantalla las 2 variables concatenadas.**



**3.Ejecutar el programa en el terminal.**



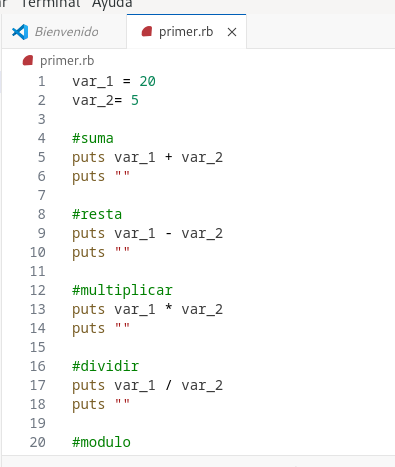
**.Editar el archivo creado anteriormente, agregar el siguiente código y ver lo que se muestra por pantalla.**



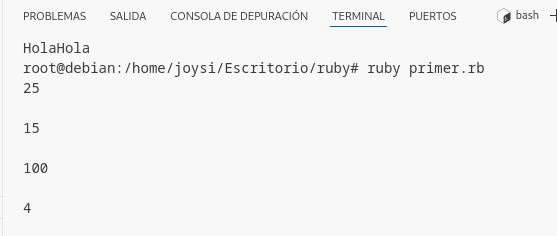
**.Ejecutar el programa en el terminal y observar la salida.**



**En Ruby existen distintos métodos que se aplican a objetos como los String, números enteros, etc. Existen métodos especiales de conversiones que se utilizan en diferentes formas o casos, para observar el funcionamiento de estos, crear un archivo programa\_conversiones.rb, declarar una variable entera y concatenar con un texto.**



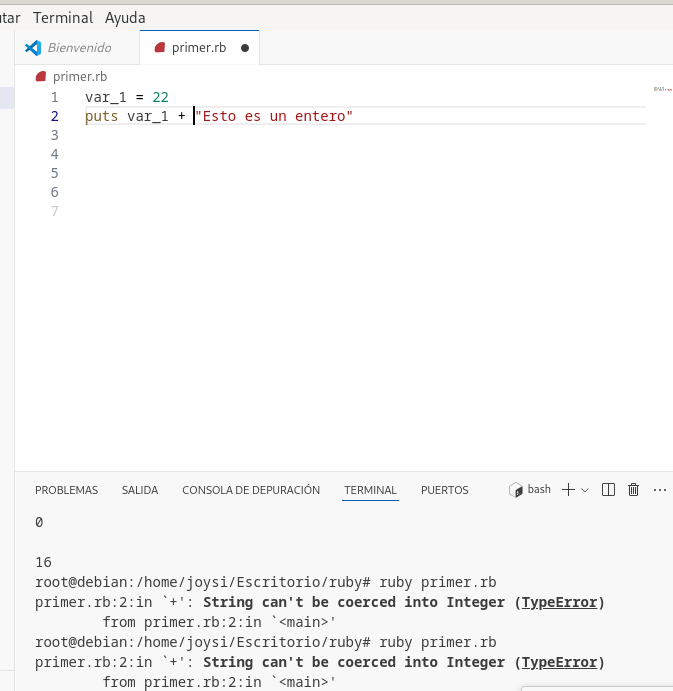
**8.Ejecutar el programa en el terminal**



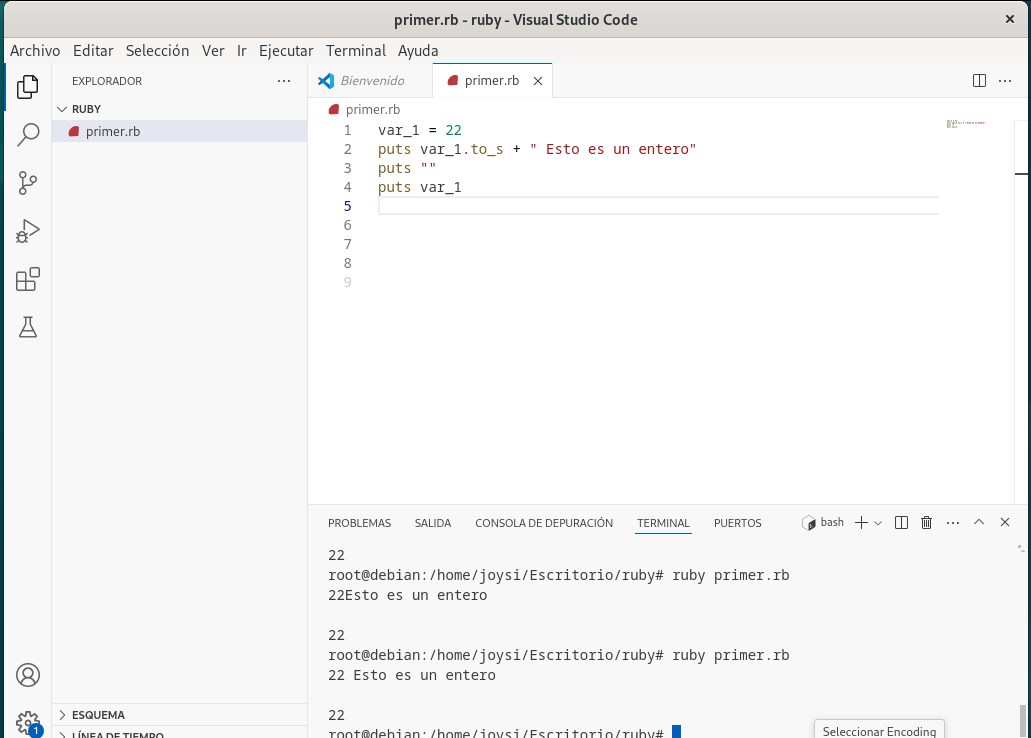


**7. En Ruby existen distintos métodos que se aplican a objetos como los String, números enteros, etc. Existen métodos especiales de conversiones que se utilizan en diferentes formas o casos, para observar el funcionamiento de estos, crear un archivo programa\_conversiones.rb, declarar una variable entera y concatenar con un texto.**

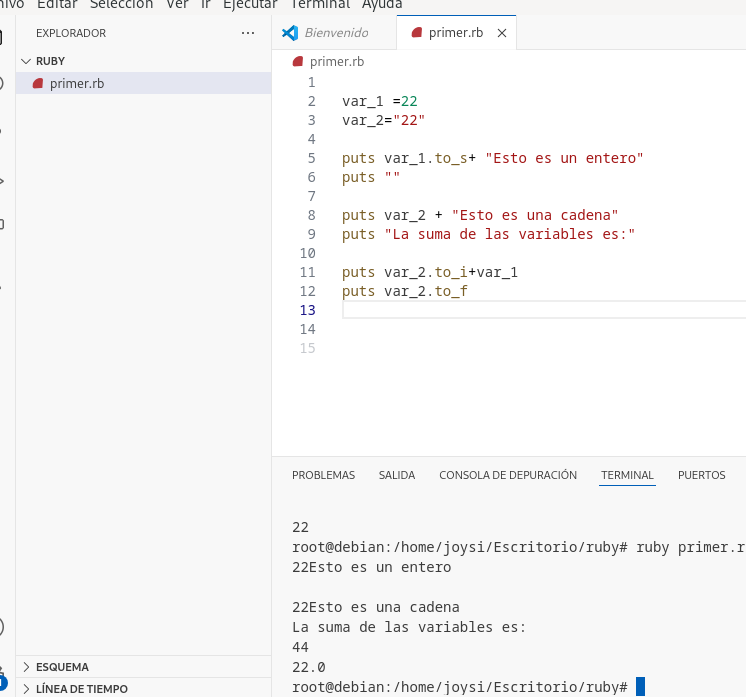
**8.Ejecutar el programa en el terminal**



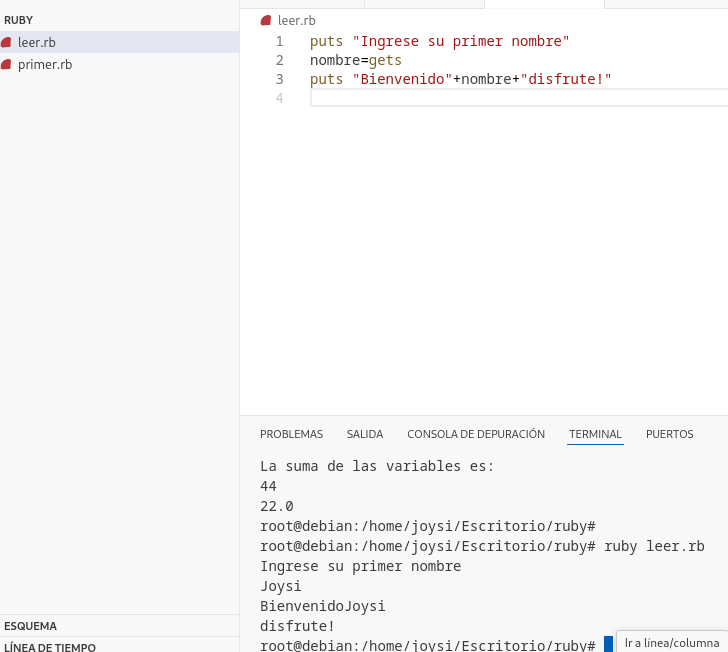
**9.Para solucionar ese error, hacer uso del método to\_s, editar el programa y agregar:**



**11.Guarda los cambios y ejecutar el programa en el terminal.**



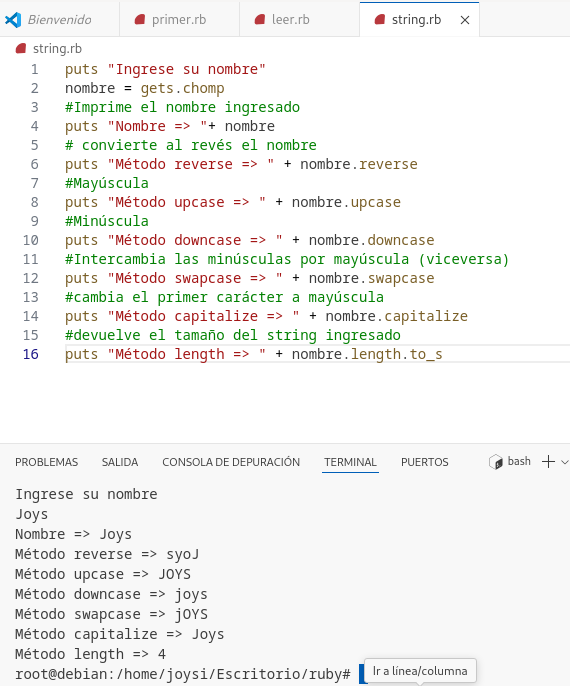
**12 Crear un programa leer.rb y agregar el siguiente código.**



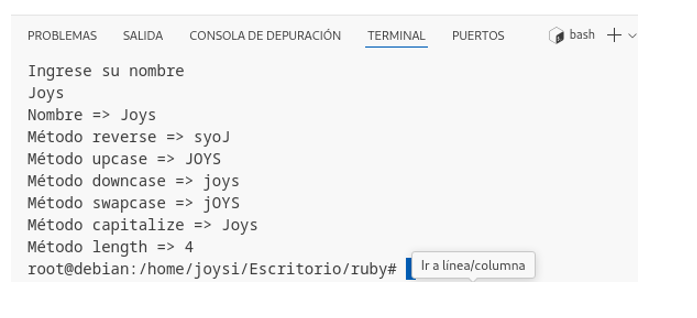
**13. Editar el programa anterior y utilizar el método chomp al momento de leer el nombre.**



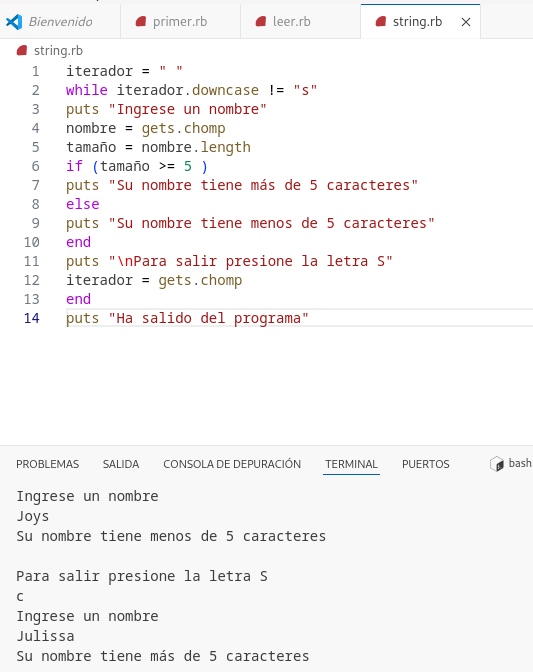
**14.Crear un nuevo programa string.rb y agregar el siguiente código:**



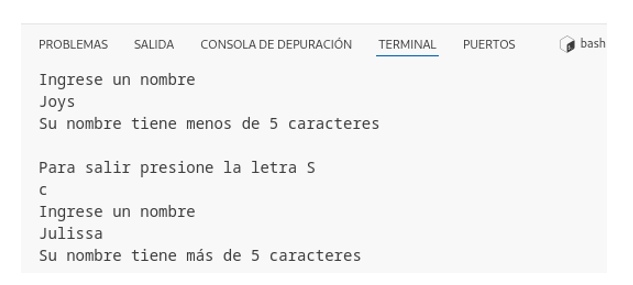
**15.Ejecute el programa en el terminal y observar el comportamiento de los métodos.**



**16.Los condicionales y los bucles en Ruby funcionan de la misma manera que en otros lenguajes de programación, para ver el funcionamiento, crear un programa nuevo y agregar el siguiente código.**



**17.Ejecutar e interactuar con el programa para ver su funcionamiento.**



**18.Complete el método/función para que convierta las palabras delimitadas por guiones/guiones bajos en mayúsculas y minúsculas. La primera palabra dentro de la salida debe estar en mayúsculas solo si la palabra original estaba en mayúsculas (conocido como Upper Camel Case, también conocido como caso Pascal). Las siguientes palabras deben estar siempre en mayúscula.**

def convert\_to\_camel\_case(s)

words = s.split(/\_|-/) # Dividir la cadena en palabras por guiones bajos o guiones

# Convertir la primera palabra a mayúscula si la original estaba en mayúscula

if words[0] =~ /^[A-Z]/

words[0] = words[0].capitalize

else

words[0] = words[0].downcase

end

# Convertir las palabras restantes a mayúscula

words = words.map(&:capitalize)[1..-1]

# Unir las palabras en una sola cadena

camel\_case\_string = words.join("")

camel\_case\_string

end

# Ejemplos de uso:

puts convert\_to\_camel\_case("hello\_world") # Salida: helloWorld

puts convert\_to\_camel\_case("HELLO\_WORLD") # Salida: HelloWorld

puts convert\_to\_camel\_case("my-python-example") # Salida: myPythonExample

puts convert\_to\_camel\_case("MY-PYTHON-EXAMPLE") # Salida: MyPythonExample

