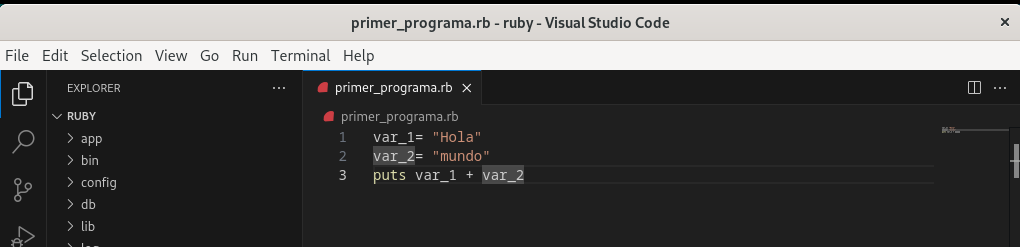
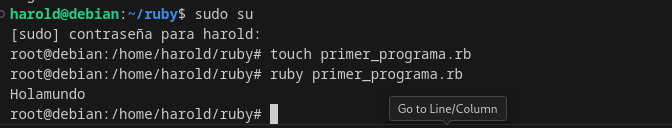
**1.1. Crear un directorio llamado Ruby, donde se almacenarán los ejercicios que se llevarán a cabo a lo largo de esta guía.**

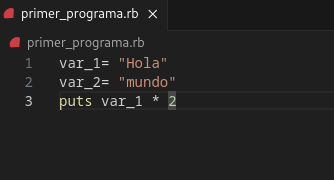
**1.2. Crear un programa primer\_programa.rb, se puede hacer desde el terminal o desde el editor de texto, asignar 2 variables de tipo String para luego imprimir por pantalla las 2 variables concatenadas.**

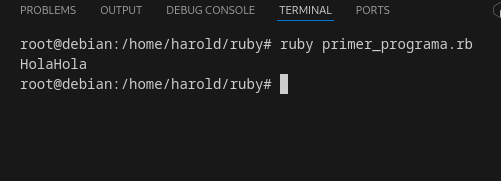
****

**Ejecutar el programa en el terminal.**

****

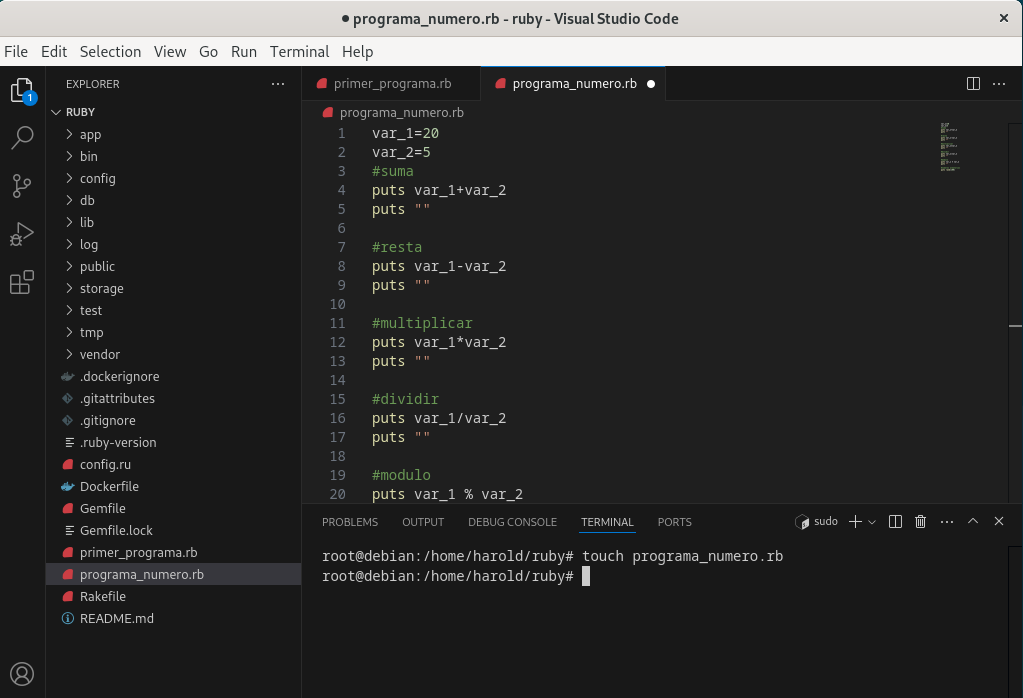
**Editar el archivo creado anteriormente, agregar el siguiente código y ver lo que se muestra por pantalla.**

****

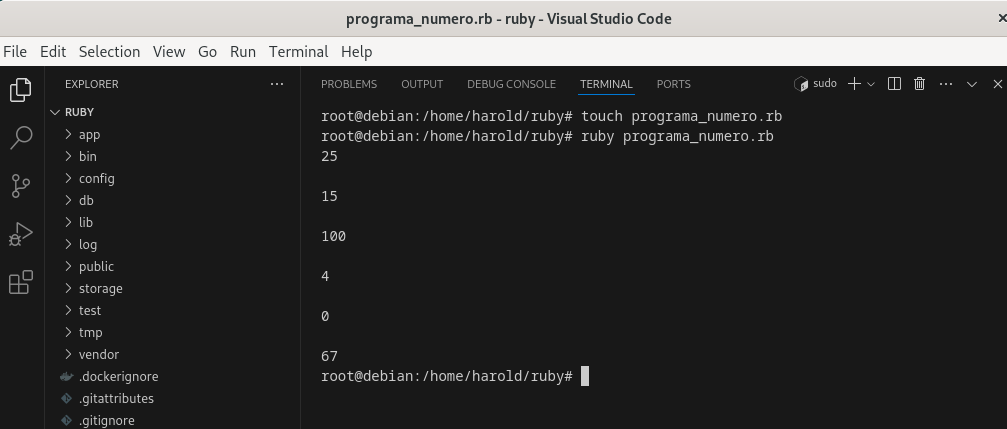
****

**2.Números**

**2.1.Crear un programa nuevo llamado programa\_numero.rb, en el que se asignarán 2 variables enteras para realizar operaciones de aritmética básica.**

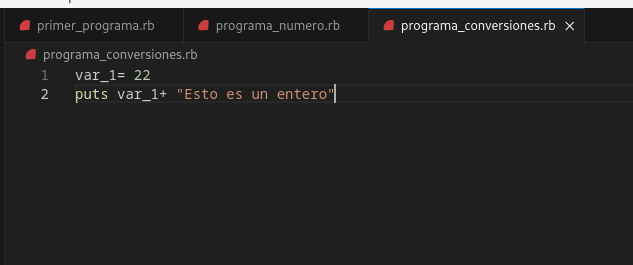
****

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

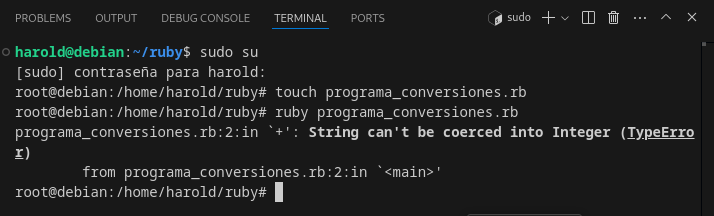
****

**3.Conversiones**

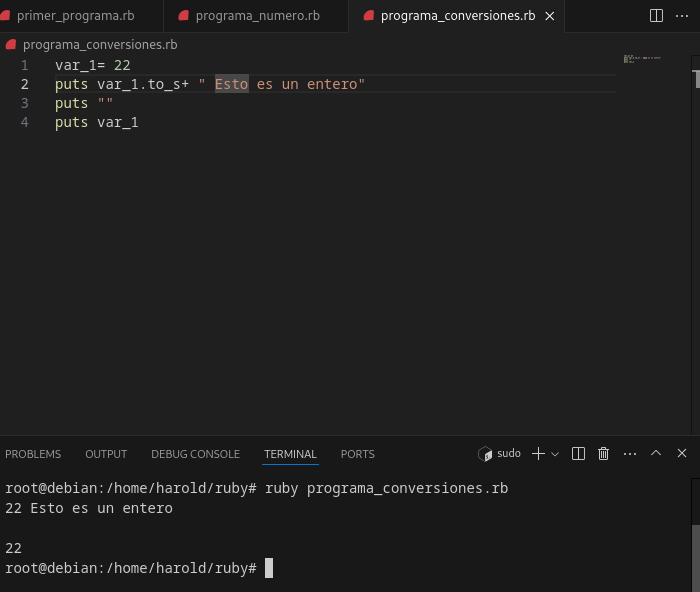
**3.1 En Ruby existen distintos métodos que se aplican a objetos como los String, números enteros, etc. Existen métodos especiales de conversiones que se utilizan en diferentes formas o casos, para observar el funcionamiento de estos, crear un archivo programa\_conversiones.rb, declarar una variable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

****

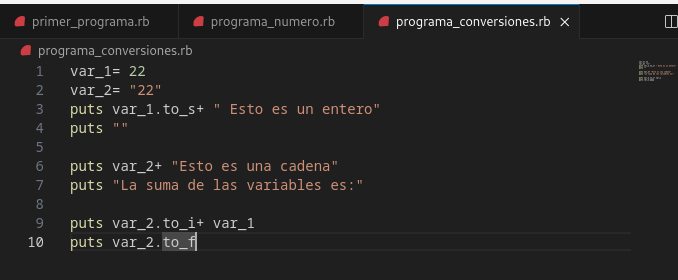
**3.2 Ejecutar el programa en el terminal**

****

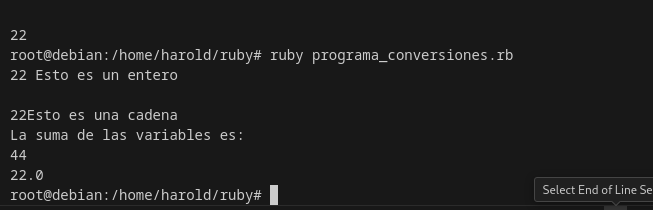
**3.3 Para solucionar ese error, hacer uso del método to\_s, editar el programa y agregar:**

****

**3.4 Editar nuevamente el programa para hacer uso de los métodos to\_i, el cual convierte una variable a entero y to\_f, el cual convierte una variable a flotante.**

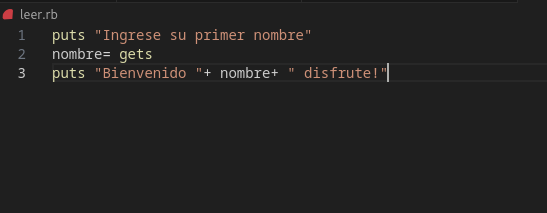
****

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

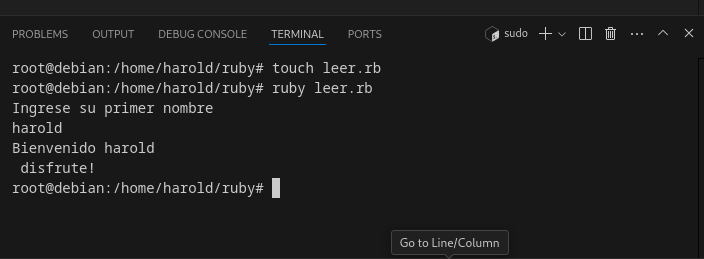
****

**4.Métodos gets y chomp. Se ha visto que el método puts se utiliza para imprimir en la pantalla; por el contrario, para leer existe el método gets que trabaja junto con el método chomp, lo que hace este último es eliminar el carácter “enter” al momento de que el método gets lee un dato del teclado.**

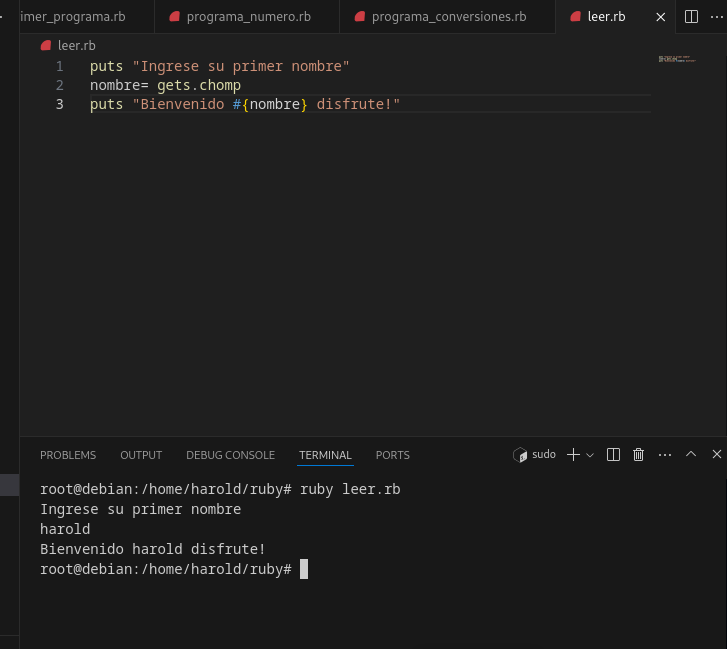
**4.1 Crear un programa leer.rb y agregar el siguiente código.**

****

**Como se puede observar, el método gets recibe el carácter “enter” como un carácter más de lectura, para solucionar eso es que se utiliza el método chomp.**

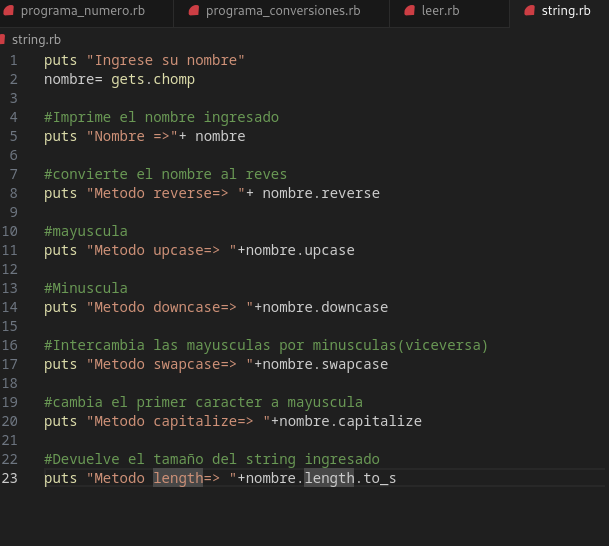
****

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

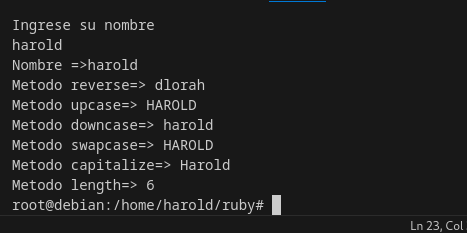
****

**5.Métodos de String Como se menciona anteriormente, en Ruby existen distintos métodos que se pueden aplicar a cada uno de los objetos del lenguaje, en esta sección se conocerá sobre los métodos relacionados a los String.**

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

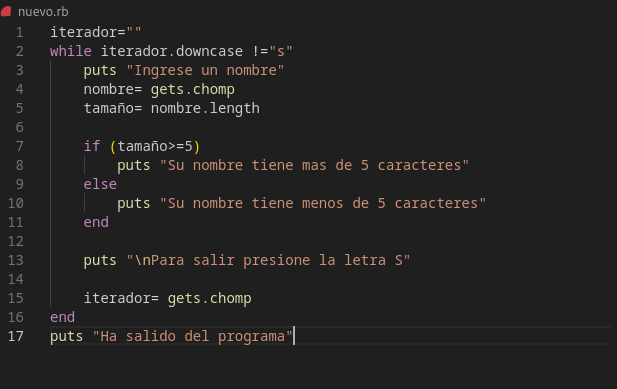
****

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

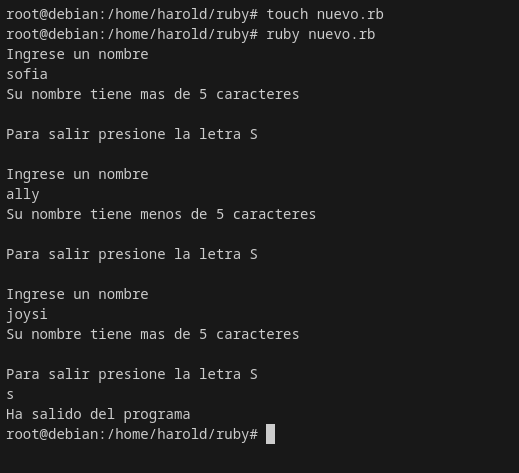
****

**6.Condicionales y bucles**

**6.1 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.**

****

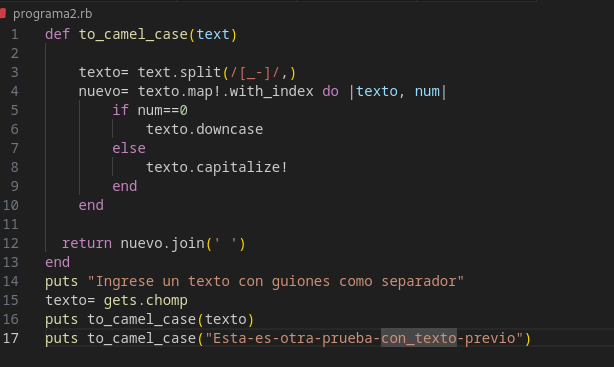
**6.2 Ejecutar e interactuar con el programa para ver su funcionamiento.**

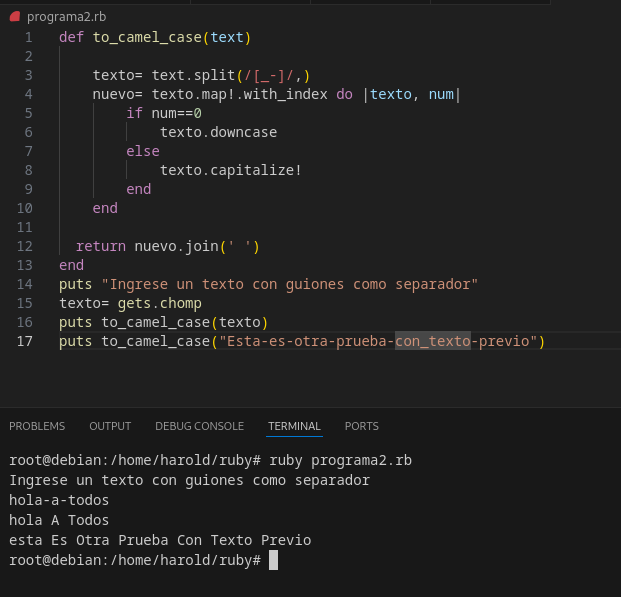
****

**Ejercicios propuestos.**

**1. Realizar cada uno de los enunciados de la guía, probar su funcionamiento y analizar cada uno de los programas planteados.**

**2. 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.**

****

****