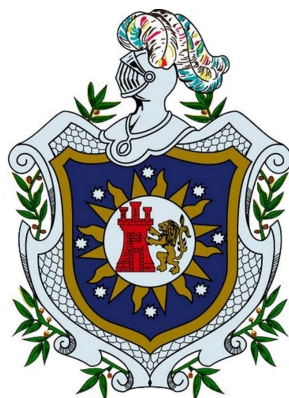


UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA
UNAN-LEÓN



UNAN - León

FACULTAD DE CIENCIAS Y TECNOLOGÍAS

COMPONENTE: SOFTWARE COMO UN SERVICIO

NOMBRE:
RONALD ALEXANDER CABALLERO

CARNET:
21-01270-0

DOC
ERVIN MONTES

FECHA:
6 DE AGOSTO DEL 2024

A LA LIBERTAD POR LA UNIVERSIDAD

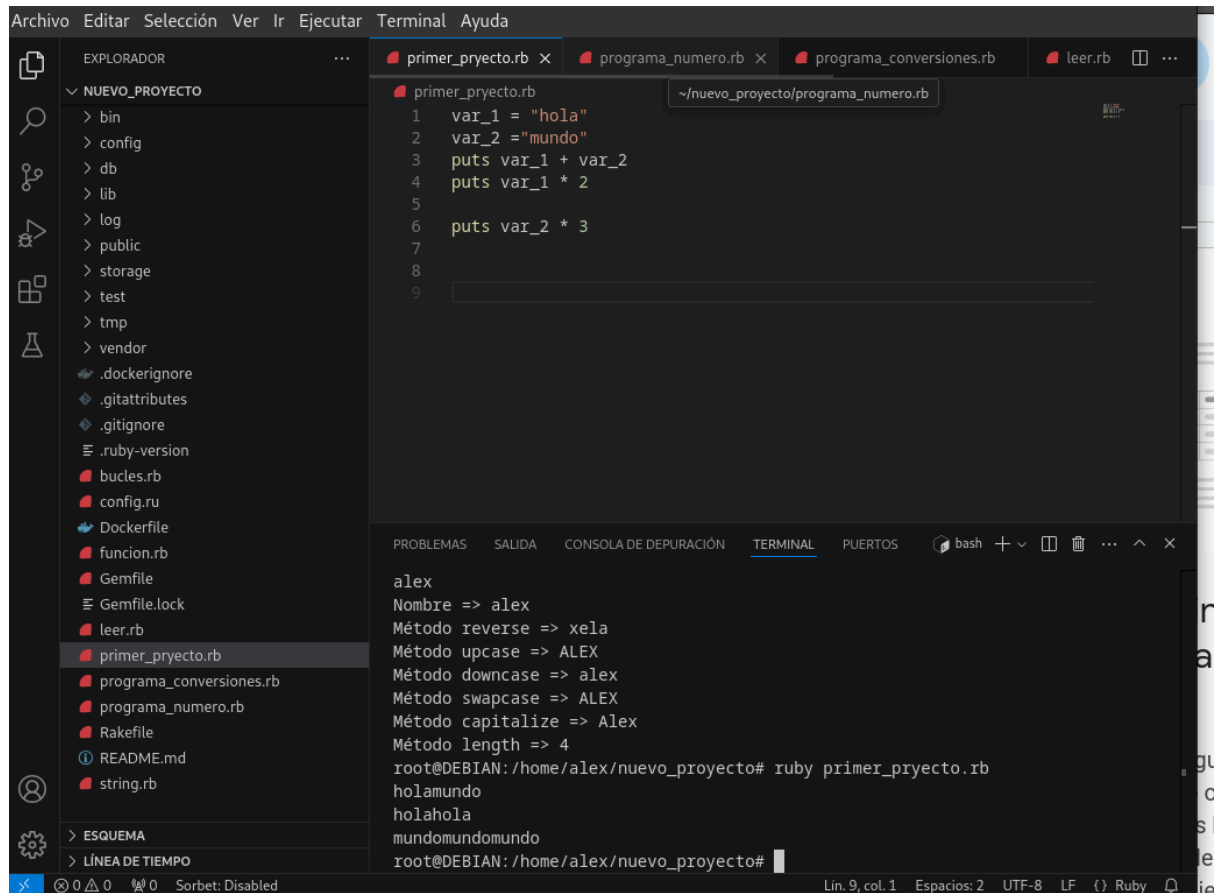
1. Realizar cada uno de los enunciados de la guía, probar su funcionamiento y analizar cada uno de los programas planteados.

1. String.

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

CODIGO COMPLETO



The screenshot shows an IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a directory structure for 'NUEVO_PROYECTO' with various files and folders. The code editor displays the content of 'primer_pryecto.rb' (note the typo in the filename). The terminal shows the execution of the program, displaying the concatenated strings 'holamundo' and 'holahola', and the product of the numbers 2 and 3, which is 6.

```
EXPLORADOR
NUEVO_PROYECTO
  bin
  config
  db
  lib
  log
  public
  storage
  test
  tmp
  vendor
  .dockerignore
  .gitattributes
  .gitignore
  .ruby-version
  bucles.rb
  config.ru
  Dockerfile
  funcion.rb
  Gemfile
  Gemfile.lock
  leer.rb
  primer_pryecto.rb
  programa_conversiones.rb
  programa_numero.rb
  Rakefile
  README.md
  string.rb
  ESQUEMA
  LÍNEA DE TIEMPO

primer_pryecto.rb
1 var_1 = "hola"
2 var_2 = "mundo"
3 puts var_1 + var_2
4 puts var_1 * 2
5
6 puts var_2 * 3
7
8
9

TERMINAL
alex
Nombre => alex
Método reverse => xela
Método upcase => ALEX
Método downcase => alex
Método swapcase => ALEX
Método capitalize => Alex
Método length => 4
root@DEBIAN: /home/alex/nuevo_proyecto# ruby primer_pryecto.rb
holamundo
holahola
mundomundomundo
root@DEBIAN: /home/alex/nuevo_proyecto#
```

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

The screenshot shows a VS Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'NUEVO_PROYECTO' with various files and folders. The terminal shows the execution of a Ruby script named 'programa_numero.rb'. The script contains the following code:

```
1 var_1 = 20
2 var_2 = 5
3 #suma
4 puts var_1 + var_2
5 puts ""
6 #resta
7 puts var_1 - var_2
8 puts ""
9 #multiplicar
10 puts var_1 * var_2
11 puts ""
12 #dividir
```

The terminal output shows the results of the script execution:

```
holahola
root@DEBIAN: /home/alex/nuevo_proyecto# ruby primer_pryecto.rb
holamundo
holahola
root@DEBIAN: /home/alex/nuevo_proyecto# ruby primer_pryecto.rb
holamundo
holahola
mundomundomundo
root@DEBIAN: /home/alex/nuevo_proyecto# touch programa_numero.rb
root@DEBIAN: /home/alex/nuevo_proyecto# ruby programa_numero.rb
25

15

100

4

0
```

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.

The screenshot shows a VS Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'NUEVO_PROYECTO' with various files and folders. The terminal shows the execution of a Ruby script named 'programa_conversiones.rb'. The script contains the following code:

```
1 var_1 = 22
2 puts var_1.to_s + " Esto es un entero"
3 puts ""
4 puts var_1
```

The terminal output shows the results of the script execution:

```
100

4

0

58
root@DEBIAN: /home/alex/nuevo_proyecto# touch programa_conversiones
.rb
root@DEBIAN: /home/alex/nuevo_proyecto# ruby programa_conversiones
.rb
programa_conversiones.rb:2:in '+': String can't be coerced into Integer (TypeError)
from programa_conversiones.rb:2:in '<main>'
root@DEBIAN: /home/alex/nuevo_proyecto# ruby programa_conversiones
.rb
22 Esto es un entero

22
root@DEBIAN: /home/alex/nuevo_proyecto#
```

```
EXPLORADOR
...
NUEVO_PROYECTO
  > app
  > bin
  > config
  > db
  > lib
  > log
  > public
  > storage
  > test
  > tmp
  > vendor
  > .dockerignore
  > .gitattributes
  > .gitignore
  > .ruby-version
  > config.ru
  > Dockerfile
  > Gemfile
  > Gemfile.lock
  > primer_pryecto.rb
  > programa_conversiones.rb
  > programa_numero.rb
  > Rakefile
  > README.md
  > ESQUEMA
  > LÍNEA DE TIEMPO
  > 0 / 0 0 % 0 Sorbet: Disabled

programa_conversiones.rb
1 var_1 = 22
2 var_2 = "22"
3 puts var_1.to_s + " Esto es un entero"
4 puts ""
5 puts var_2 + " Esto es una cadena"
6 puts "La suma de las variables es:"
7 puts var_2.to_i + var_1
8 puts var_2.to_f

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
.ruby
root@DEBIAN:/home/alex/nuevo_proyecto# ruby programa_conversiones
.ruby
programa_conversiones.rb:2:in `+': String can't be coerced into Integer (TypeError)
from programa_conversiones.rb:2:in `<main>'
root@DEBIAN:/home/alex/nuevo_proyecto# ruby programa_conversiones
.ruby
22 Esto es un entero

22
root@DEBIAN:/home/alex/nuevo_proyecto# ruby programa_conversiones
.ruby
22 Esto es un entero

22 Esto es una cadena
La suma de las variables es:
44
22.0
root@DEBIAN:/home/alex/nuevo_proyecto#
```

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.

```
EXPLORADOR
...
NUEVO_PROYECTO
  > app
  > bin
  > config
  > db
  > lib
  > log
  > public
  > storage
  > test
  > tmp
  > vendor
  > .dockerignore
  > .gitattributes
  > .gitignore
  > .ruby-version
  > config.ru
  > Dockerfile
  > Gemfile
  > Gemfile.lock
  > leer.rb
  > primer_pryecto.rb
  > programa_conversiones.rb
  > programa_numero.rb
  > Rakefile
  > README.md
  > ESQUEMA
  > LÍNEA DE TIEMPO
  > 0 / 0 0 % 0 Sorbet: Disabled

leer.rb
1 puts "Ingrese su primer nombre"
2 nombre = gets.chomp
3 puts "Bienvenido #{nombre} disfrute! "

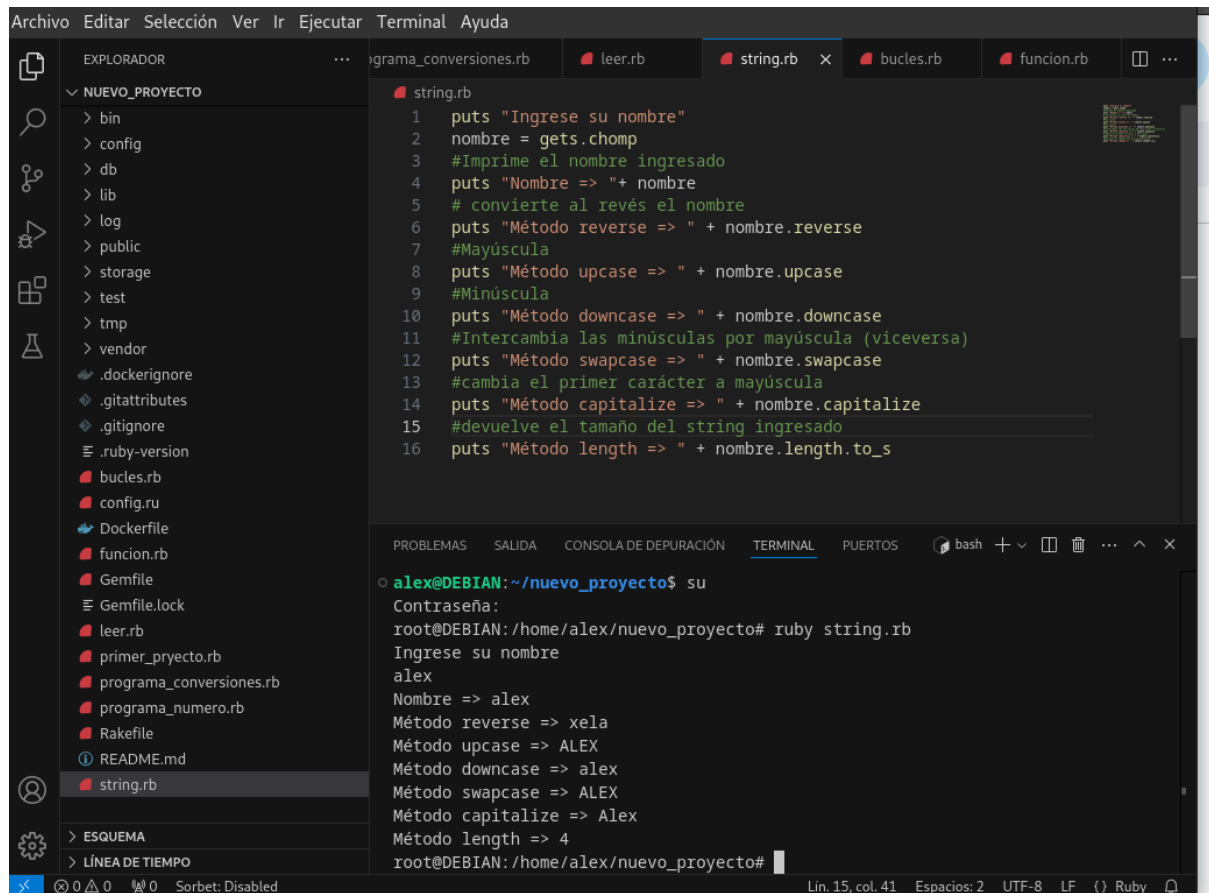
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
from programa_conversiones.rb:2:in `<main>'
root@DEBIAN:/home/alex/nuevo_proyecto# ruby programa_conversiones
.ruby
22 Esto es un entero

22
root@DEBIAN:/home/alex/nuevo_proyecto# ruby programa_conversiones
.ruby
22 Esto es un entero

22 Esto es una cadena
La suma de las variables es:
44
22.0
root@DEBIAN:/home/alex/nuevo_proyecto# touch leer.rb
root@DEBIAN:/home/alex/nuevo_proyecto# ruby leer.rb
Ingrese su primer nombre
alex
Bienvenido alex disfrute!
root@DEBIAN:/home/alex/nuevo_proyecto#
```

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.

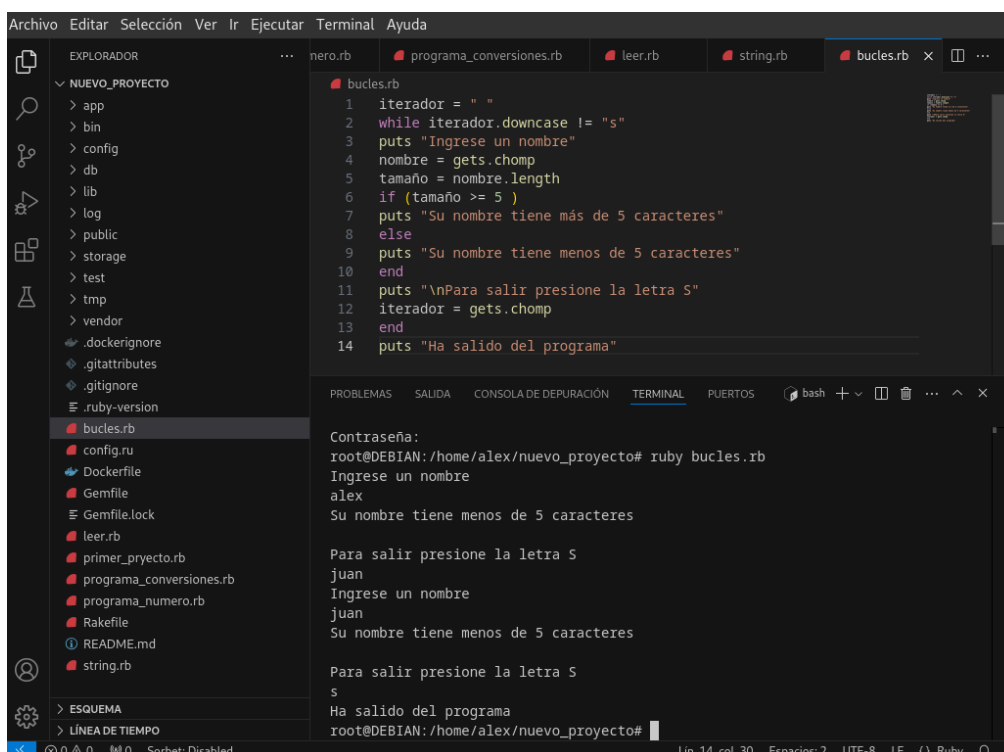


The screenshot shows a Ruby IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'NUEVO_PROYECTO' with various files and folders. The code editor displays the contents of 'string.rb', which contains a Ruby script that prompts the user for their name and then performs several string operations: reversing the name, converting it to uppercase, lowercase, and swapping the case of each character. The terminal at the bottom shows the execution of the script, with the user entering 'alex' and the program outputting the results of the operations.

```
1 puts "Ingrese su nombre"
2 nombre = gets.chomp
3 #Imprime el nombre ingresado
4 puts "Nombre => " + nombre
5 # convierte al revés el nombre
6 puts "Método reverse => " + nombre.reverse
7 #Mayúscula
8 puts "Método upcase => " + nombre.upcase
9 #Minúscula
10 puts "Método downcase => " + nombre.downcase
11 #Intercambia las minúsculas por mayúscula (viceversa)
12 puts "Método swapcase => " + nombre.swapcase
13 #cambia el primer carácter a mayúscula
14 puts "Método capitalize => " + nombre.capitalize
15 #devuelve el tamaño del string ingresado
16 puts "Método length => " + nombre.length.to_s
```

```
alex@DEBIAN:~/nuevo_proyecto$ su
Contraseña:
root@DEBIAN:/home/alex/nuevo_proyecto# ruby string.rb
Ingrese su nombre
alex
Nombre => alex
Método reverse => xela
Método upcase => ALEX
Método downcase => alex
Método swapcase => ALEX
Método capitalize => Alex
Método length => 4
root@DEBIAN:/home/alex/nuevo_proyecto#
```

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.



The screenshot shows a Ruby IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'NUEVO_PROYECTO' with various files and folders. The code editor displays the contents of 'bucles.rb', which contains a Ruby script that prompts the user for their name and then checks if the name is longer than 5 characters. If it is, it prints a message and asks the user to press 's' to continue. If it is not, it prints a message and asks the user to press 's' to continue. The terminal at the bottom shows the execution of the script, with the user entering 'alex' and the program outputting the results of the operations.

```
1 iterador = " "
2 while iterador.downcase != "s"
3   puts "Ingrese un nombre"
4   nombre = gets.chomp
5   tamaño = nombre.length
6   if (tamaño >= 5 )
7     puts "Su nombre tiene más de 5 caracteres"
8   else
9     puts "Su nombre tiene menos de 5 caracteres"
10  end
11  puts "\nPara salir presione la letra S"
12  iterador = gets.chomp
13 end
14 puts "Ha salido del programa"
```

```
Contraseña:
root@DEBIAN:/home/alex/nuevo_proyecto# ruby bucles.rb
Ingrese un nombre
alex
Su nombre tiene menos de 5 caracteres

Para salir presione la letra S
juan
Ingrese un nombre
juan
Su nombre tiene menos de 5 caracteres

Para salir presione la letra S
s
Ha salido del programa
root@DEBIAN:/home/alex/nuevo_proyecto#
```