

Desarrollo ágil de Software

Actividad: Ruby

Parte 1 - tiempo estimado 30 min	1
Objetivo	1
Video	1
Ejercicios	1
Parte 2 - tiempo estimado 30 min	3
Objetivo	3
Video	3
Ejercicios	3

Parte 1 - tiempo estimado 30 min

Objetivo

- Usar la herramienta **irb** para ejercitar expresiones en **Ruby**
- Aprender sintaxis básica de expresiones y los tipos de datos Integer, Float, nil, boolean, string

Video

Ver el [video](#) (7 min).

Ejercicios

Prerrequisitos:

- Ver el video,
- Tener el ambiente ruby 2.6 disponible: docker ejecutando la imagen obtenida del repositorio <https://github.com/kleer-la/docker-csd-ruby2.6.5>.
 - Plan B: es preferible que tengas ruby en tu equipo, si no lograste instalarlo, puedes practicar en línea: <https://joshnuss.github.io/mruby-web-irb/>

Referencia: https://www.digitalocean.com/community/tutorial_series/how-to-code-in-ruby

Realiza los siguiente ejercicios dentro de **irb**

1. String. Prueba estas expresiones. En caso que tengas un resultado inesperado, investiga la razón.
 - a. Strings: Prueba estas expresiones.
"Con Ruby nos podemos divertir".include? "Ruby"
"Con Ruby nos podemos divertir".start_with? "Ruby"

```

"Con Ruby nos podemos divertir".end_with? "Ruby"
s= "Scrum"
s.upcase
s.downcase
s.swapcase
s
s.downcase!
s

```

```

"Voy a lograrlo".sub( "Voy", "Vamos" )
"Voy a lograrlo".sub( "a", "@" )

```

- b. Interpolación de strings. Prueba estas expresiones.

```

nombre = "Indira"
edad = 26
puts "Su nombre es #{ nombre } y tiene #{ edad } años."

```

2. Busca las expresiones para resolver estas preguntas:
 - a. Si tengo "- - - -", ¿cómo puedo saber cuantos "-" tienen?
 - b. Si tengo algo como "- b - -", donde los "-" pueden ser reemplazados por letras ¿cómo puedo saber cuantos "-" tienen? ¿cuál es el primer "-"?
3. Números. Prueba estas expresiones. En caso que tengas un resultado inesperado, investiga la razón.

- a. Números y conversiones.

```

"23".to_i
"23".to_f
"".to_i
23.to_i
3 / 2
3 / 2.to_f
3 / 2.0
nil.to_i
nil.to_i +1
nil.to_i + 1
nil + 1

```

4. Boolean. Prueba estas expresiones. En caso que tengas un resultado inesperado, investiga la razón.

- a. true / false / nil

```

1==1
false==nil
!false==!nil

```

- b. Operadores and / or

```

false or true

```

- ```
1==1 and 1==2
```
- c. if en una línea
- ```
a= 3 if 1==1
```
- d. if multilínea
- ```
if 1==1
 a= 3
end
```

---

## Parte 2 - tiempo estimado 30 min

### Objetivo

- Aprender tipos de datos array, hash, symbol y operaciones básicas con ellos.
- Aprender sintaxis de bloques de código

### Video

Ver el [video](#) (11:30 min).

### Ejercicios

Prerrequisitos:

- Ver el video,
- Realizar la Actividad Ruby parte 1
  - Plan B: es preferible que tengas ruby en tu equipo, si no lograste instalarlo, puedes practicar en línea: <https://joshnuss.github.io/mruby-web-irb/>

Referencia: [https://www.digitalocean.com/community/tutorial\\_series/how-to-code-in-ruby](https://www.digitalocean.com/community/tutorial_series/how-to-code-in-ruby)

Realiza los siguiente ejercicios dentro de **irb**

1. Strings y arrays: Prueba estas expresiones. En caso que tengas un resultado inesperado, investiga la razón.  

```
"Pablitux clavo un clavitux".split(" ")
```

```
"Pablitux clavo un clavitux".split(" ").join("-")
```

```
"Pablitux clavo un clavitux".split("")
```
2. Arrays: Prueba estas expresiones. En caso que tengas un resultado inesperado, investiga la razón.  

```
%w{uno dos tres cuatro}.find {|e| e.include? "o"}
```

```
%w{uno dos tres cuatro}.select {|e| e.include? "o"}
```
3. Arrays: Prueba estas expresiones. En caso que tengas un resultado inesperado, investiga la razón.  

```
%w{uno dos tres cuatro}.find {|e| e.include? "o"}
```

```
%w{uno dos tres cuatro}.select {|e| e.include? "o"}
```

4. for y range: Prueba estas expresiones. En caso que tengas un resultado inesperado, investiga la razón.

```
(1..3).each {|i| puts i}
```

```
(1...3).each {|i| puts i}
```

```
for n in %w{uno dos tres cuatro}
```

```
 puts n
```

```
end
```

```
numeros= %w{uno dos tres cuatro}
```

```
for i in (1...numeros.length)
```

```
 puts i,numeros[i]
```

```
end
```

```
%w{uno dos tres cuatro}.each_with_index {|i, elem| puts
i,elem}
```

5. Busca las expresiones para resolver estas preguntas:
- Si tengo %w{- - -}, ¿cómo puedo saber cuantos "-" tienen?
  - Si tengo algo como %w{- b - -}, donde los "-" pueden ser reemplazados por letras ¿cómo puedo saber cuantos "-" tienen? ¿cuál es el primer "-"?