



Centro Universitario de Ciencias Exactas e Ingenierías



participación Rust 01

Pérez Tovar Santiago

**Programación de sistemas avanzados
Sección D01**

Fecha de Entrega: 08/Marzo /2025

Preguntas dejadas en clase:

Santiago Perez Toran 08/03/2025

13

¿Que es la inmutabilidad en Rust?

Es la característica que impide cambiar el valor de una variable una vez se le haya asignado, esto se hace después de declarar el `let` y con `mut`.

¿Por que Rust tiene variables inmutables por defecto?

Por lo que en Rust es por seguridad o garantizar en un futuro del código, que no se modifique una variable o un dato y provoque errores o fallas en el sistema.

¿Cual es la diferencia entre `mut` y shadowing?

Si bien `mut` permite cambiar el valor de la variable, tiene que ser el mismo tipo de dato, pero el shadowing permite cambiar hasta el tipo de dato al que es original mente, permitiendo pasar de un numerico a texto o demas.

San tiago Perez Tovar

08/03/2023

¿Cuándo usar `{ }` y `{:P}` en `println`?

El `{ }` se utiliza cuando se quiere imprimir la variable pero mas bien la que contiene, claro si es un arreglo se especifica la posición.

En cambio `{:P}` nos ayuda como a depurar la información en la memoria pues este nos imprime todo el arreglo y incluso en objetos que de ellos a imprimir el objeto con sus atributos.

¿Cuándo usar `const` en lugar de `let`?

Cuando queramos que una variable siempre sea inmutable y esta no cambie su información. Esta no puede volverse mutable como por ejemplo `let`.

Ejecución de código en clase:

Código:

```
1  fn main() {
2
3      let x = 5;
4
5      println!("Hello, world!");
6
7      //x=10; <---esto no es valido
8
9      //let mut y = 10;
10     //y = 20;
11     let x = 5;
12     {
13         let x = x + 1; // nueva variable
14         println!("E valor de x es: {}",x);
15     }
16     println!("E valor de x es: {}",x);
17     // VARIABLES //
18
19     let entero: i32 = 42;
20     let flotante: f64 = 3.1416;
21     let booleano: bool = true;
22     let caracter: char = 'a';
23     //tupla
24     let firulais: (i32, f64, char) = (43, 4.1416, 'b');
25     //array
26     let arreglo: [i32;3] = [1, 2, 3];
27
28     println!("Tupla(firulais) forma 1: {:?}",firulais);
29
30     println!("Tupla(firulais) forma 2: ({}, {}, {})",firulais.0,firulais.1,firulais.2);
31
32 }
```

Compilación y ejecución:

```
PS C:\Users\Santiago\Documents\Semestre 8\Programacion de sistemas avanzados\Actividad 4. Codigo> rustc main.rs
warning: unused variable: `x`
--> main.rs:3:9
3 | |
  | |
  | ^ help: if this is intentional, prefix it with an underscore: `_x`
  = note: `#[warn(unused_variables)]` on by default

warning: unused variable: `entero`
--> main.rs:19:9
19 | |
   | |
   | ^^^^^^ help: if this is intentional, prefix it with an underscore: `_entero`

warning: unused variable: `flotante`
--> main.rs:20:9
20 | |
   | |
   | ^^^^^^^ help: if this is intentional, prefix it with an underscore: `_flotante`

warning: unused variable: `booleano`
--> main.rs:21:9
21 | |
   | |
   | ^^^^^^^^ help: if this is intentional, prefix it with an underscore: `_booleano`

--> main.rs:22:9
22 | |
   | |
   | ^^^^^^^^ help: if this is intentional, prefix it with an underscore: `_caracter`

warning: unused variable: `arreglo`
--> main.rs:26:9
26 | |
   | |
   | ^^^^^^^ help: if this is intentional, prefix it with an underscore: `_arreglo`

PS C:\Users\Santiago\Documents\Semestre 8\Programacion de sistemas avanzados\Actividad 4. Codigo> .\main.exe
Hello, world!
E valor de x es: 6
E valor de x es: 5
o Tupla(firulais) forma 1: (43, 4.1416, 'b')
Tupla(firulais) forma 2: (43, 4.1416, b)
PS C:\Users\Santiago\Documents\Semestre 8\Programacion de sistemas avanzados\Actividad 4. Codigo> 
```

Link Github:

<https://github.com/SantiagoPerezT/SistemasAvanzadoCurso>

Para la actividad es la siguiente:

<https://github.com/SantiagoPerezT/SistemasAvanzadoCurso/tree/main/Actividad%204.%20Codigo>