Funciones

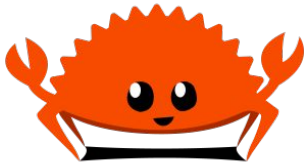
NACION CRYPTO By DECRY

Fn

```
fn nombre ( ) {
        Bloque_1
}
```

```
Fn foo (){

}
```

La palabra reservada **fn** indica que estamos ante la presencia de una función. Todo programa en Rust tiene al menos una función, “main”.

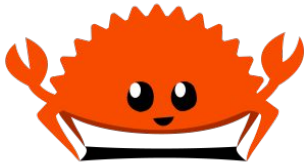


Funciones

```
fn nombre ( argumentos: tipo)  {
        Bloque_1
}
```

```
fn main() {

   imprimir_numero(5);
}

fn imprimir_numero(x: i32) {

   println!("x es: {}", x);

   }
```

Podemos definir los argumentos que recibe la función dentro del paramentesis

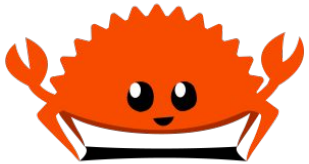


Funciones

```
fn nombre ( argumentos: tipo)  -> tipo_retorno {
        Bloque_1
}
```

```
fn main() {

    imprimir_suma(5, 6);

}

fn imprimir_suma(x: i32, y: i32) {

    println!("la suma es: {}", x + y);

}
```

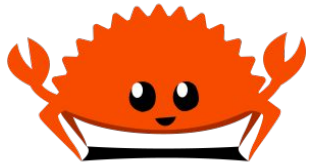
Se define el tipo de dato que retorna y los argumentos con su tipo y separados por ‘coma’



Return

```
fn retorno_anticipado(valor: i32) -> i32 {
        return valor;
        // No ejecutara la siguiente linea
        valor + 1
}
```

```
fn funcion_retorno(x: i32) -> i32 {
   return x + 1;
}
```

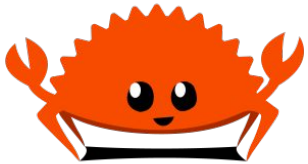


Closures

```
|argumentos| expresión;
```

```
let multiplicar_dos = |x: i32| x * 2;
assert_eq!(2, multiplicar_dos(1));
-----------------------------------
let restar_dos = |x| {
   let mut resultado: i32 = x;
   resultado -= 1;
   resultado -= 1;
   resultado
};

assert_eq!(4, restar_dos(6));
```

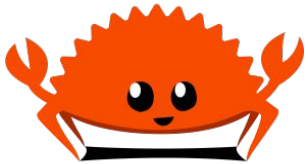
Es un mecanismo para envolver una función y las variables libres, permitiendo una reutilización del código. La expresión puede ser una sola línea ó multilínea en ese caso se utiliza { }



Closures

```
fn  suma   (x: i32) -> i32 { x + 1 }
let suma_uno = |x: i32| -> i32 { x + 1 };
let suma_un   = |x: i32|          x + 1  ;
```

En los closures no es necesario especificar el tipo de dato de retorno



BIBLIOGRAFIA:

- Programming Rust, Fast, safe systems development - 2nd edition, Jim blandy, Jason Orendorff and Leonora F. S. Tindall
- https://doc.rust-lang.org/book/
- https://doc.rust-lang.org/stable/std/
- https://www.programiz.com/rust/function
- https://polkadothub.io/rust/0-presentaci%C3%B3n/0-presentation
- https://www.youtube.com/watch?v=dad1NQdjd0I&t=619s