

Panic!

El uso de `panic` es el mecanismo más sencillo de control de errores de Rust.

Puede usar la macro `panic!` para emitir una alerta de pánico para el subproceso actual. Imprime un mensaje de error, libera recursos y, a continuación, sale del programa.

En este sencillo ejemplo se muestra cómo llamar a la macro `panic!`:

```
Rust Copiar  
  
fn main() {  
    panic!("Farewell!");  
}
```

Este programa se cerrará con el código de estado `101` e imprimirá el mensaje siguiente:


```
Resultados Copiar  
  
thread 'main' panicked at 'Farewell!', src/main.rs:2:5
```

La última parte del mensaje de pánico anterior muestra la ubicación correspondiente. Se produjo en el archivo `src/main.rs`, en el quinto carácter de la segunda línea.

En términos generales, debe usar `panic!` cuando un programa alcanza un estado irrecuperable, lo que significa que no hay ninguna manera de recuperarse del error.

Rust entra en pánico en algunas operaciones, como una división por cero o un intento de acceder a un índice que no se ha enviado previamente a una matriz, un vector o un mapa hash, como se muestra en el código siguiente:

Rust

 Copiar

```
let v = vec![0, 1, 2, 3];  
println!("{}", v[6]); // this will cause a panic!
```