

Compilación y Ejecución en Rust

La ejecución de código en Rust puede realizarse mediante dos vías fundamentales:

- Compilador directo, rustc, para tareas sencillas.
- Cargo la herramienta estándar de gestión de proyectos, indispensable para el desarrollo moderno.

Usando rustc

Para comprender la esencia del proceso de compilación, es crucial familiarizarse con rustc, el compilador de Rust. Este método es ideal para archivos individuales o para entender cómo el código fuente se traduce en un ejecutable binario.

Fases del Proceso rustc

1. Paso 1: Creación del Módulo Fuente

Todo comienza con el código fuente, que tradicionalmente lleva la extensión `.rs`.

```
fn main() {  
    println!("Compilador directo rustc."); //Archivo: main.rs  
}
```



2. Paso 2: Compilación

Desde la terminal, se invoca a rustc, apuntando al archivo de entrada. El compilador lee el código y genera un archivo binario ejecutable en el mismo directorio.

```
rustc main.rs
```

En este proceso, rustc maneja internamente la verificación de tipos, el borrow checker y la generación del código máquina optimizado, utilizando LLVM.

3. Ejecución

Esto genera un ejecutable.

- Windows:

```
.\main.exe
```

- Linux/macOS:

```
./main
```

Resultado:

```
Compilador directo rustc.
```

Tu Primer Proyecto con Cargo

Crear un nuevo proyecto

```
# Crear un proyecto binario (aplicación)  
cargo new hola_mundo
```

```
# Entrar al directorio  
cd hola_mundo
```