

Rust

Curso de Rust desde 0.

En este curso vamos a partir desde lo más básico y luego especializarnos en ciertas áreas más específicas para poder ver cómo podemos aprovechar las ventajas que posee Rust en ciertos entornos.

Lo recomendable, es iniciar con el primer bloque, y luego elegir el bloque que le interese más.

Es recomendable también ya saber programar en algún lenguaje de programación antes de tomar el curso, pues en el curso se asume que se sabe algún lenguaje de antemano. Lo ideal sería que tenga experiencia programando en C++ y Python (lenguajes que se harán referencias para ejemplificar a lo largo del curso), aunque otros lenguajes también están bien. No se recomienda para aprender a programar como primer lenguaje.

Descripción General del Curso

Este curso completo de Rust está diseñado para desarrolladores que desean dominar este lenguaje de programación de sistemas, reconocido por su seguridad, rendimiento y concurrencia. Estructurado en cinco bloques, abarca desde los fundamentos de Rust hasta aplicaciones avanzadas en sistemas embebidos, desarrollo web, Android y creación de librerías para Python. Basado en recursos oficiales como *The Rust Programming Language*, *The Embedded Rust Book*, *Rust and WebAssembly*, *Comprehensive Rust* y *PyO3 User Guide*. Los participantes aprenderán a escribir código seguro y eficiente, aplicar Rust en diversos dominios y desarrollar una aplicación de línea de comandos. Ideal para programadores con experiencia básica que buscan habilidades en programación de sistemas moderna.

Bloque 1: Fundamentos de Rust

(Rust Vanilla)

Basado en: [The Rust Programming Language](#)

1.01 - Introducción

1. Rust
2. Instalación
3. Compilador: `rustc` y `cargo`
4. IDEs
5. Hello World!

1.02 - Conceptos básicos

1. Variables y mutabilidad
2. Tipos de datos
3. Operaciones
4. *Input/Output*
5. Importaciones

6. `const`

1.03 - Control de flujo

1. Condicionales
2. Bucles

1.04 - Funciones y bloques

1. Funciones
2. Parámetros y retorno
3. Bloques y alcance

1.05 - *Ownership*

1. Reglas de propiedad
2. Cadenas de texto
3. Mover
4. Clonar y copiar
5. Funciones
6. Referencias y *borrowing*
7. El compilador

1.06 - Datos compuestos

1. Cadenas de Texto
2. Arreglos
3. Tuplas
4. Vectores
5. Hash Maps
6. `slice`

1.07 - Estructuras

1. Estructuras
2. Métodos

1.08 - *Enums*

1. *Enums*
2. Match
3. `if let`

1.09 - Paquetes y manejo de errores

1. Paquetes, crates y módulos
2. Manejo de errores

1.10 - Datos genéricos, *traits* y *lifetime*

1. Tipos de datos genéricos
2. *Traits*
3. *Lifetime*

1.11 - Pruebas

1. Pruebas (*tests*)

1.12 - Paradigmas de programación

1. Programación Funcional
2. Programación Orientada a Objetos

1.13 - Concurrencia y asincronía

1. *Fearless Concurrency*
2. Hilos
3. Transferencia de información
4. Programación asincrónica

1.14 - Conceptos avanzados

1. Punteros inteligentes
2. *Unsafe*
3. Macros

1.15 - Proyecto

(Aplicación de línea de comandos en Rust)

Basado en: [Command Line Applications in Rust](#)

Bloque 2: Sistemas Embebidos y Críticos

(Enfocado en bajo nivel, control de hardware, entornos sin sistema operativo, y aplicaciones de tiempo real.)

Basado en: [The Embedded Rust Book](#)

Bloque 3: WebAssembly, Servidores y Backend

(Rust para aplicaciones web y servicios escalables)

Basado en: [Rust and WebAssembly](#)

Bloque 4: Android

(Rust para software del sistema en Android)

Basado en: [Comprehensive Rust](#)

Bloque 5: Librerías para Python

(Desarrollo de librerías para Python desde Rust con PyO3)

Basado en: [PyO3 user guide](#)