

El Lenguaje de Programacion Rust

Jose Narvaez

Published
with GitBook



Tabla de contenido

Introducción	0
Primeros Pasos	1
Instalando Rust	1.1
¡Hola, mundo!	1.2
¡Hola, Cargo!	1.3
Aprende Rust	2
El Juego de las Adivinanzas	2.1
La Cena de los Filósofos	2.2
Rust dentro de otros Lenguajes	2.3
Rust Efectivo	3
La Pila y el Montículo	3.1
Pruebas	3.2
Compilación Condicional	3.3
Documentación	3.4
Iteradores	3.5
Concurrencia	3.6
Manejo de Errores	3.7
FFI	3.8
Borrow y AsRef	3.9
Canales de Distribución	3.10
Sintaxis y Semántica	4
Enlaces a Variables	4.1
Funciones	4.2
Tipos Primitivos	4.3
Comentarios	4.4
if	4.5
Ciclos	4.6
Pertenencia	4.7
Referencias y Préstamo	4.8
Tiempos de Vida	4.9

Mutabilidad	4.10
Estructuras	4.11
Enumeraciones	4.12
Match	4.13
Patrones	4.14
Sintaxis de Métodos	4.15
Vectores	4.16
Cadenas de Caracteres	4.17
Genéricos	4.18
Traits	4.19
Drop	4.20
if let	4.21
Objetos Trait	4.22
Closures	4.23
Sintaxis Universal de Llamada a Funciones	4.24
Crates y Módulos	4.25
`const` y `static`	4.26
Atributos	4.27
Alias `type`	4.28
Conversión entre Tipos	4.29
Tipos Asociados	4.30
Tipos sin Tamaño	4.31
Operadores y Sobrecarga	4.32
Coerciones Deref	4.33
Macros	4.34
Apuntadores Planos	4.35
`unsafe`	4.36
Rust Nocturno	5
Plugins del Compilador	5.1
Ensamblador en Línea	5.2
No stdlib	5.3
Intrínsecos	5.4
Ítems de Lenguaje	5.5
Enlace Avanzado	5.6

Pruebas de Rendimiento	5.7
Sintaxis Box y Patones	5.8
Patrones Slice	5.9
Constantes Asociadas	5.10
Glosario	6
Investigación Académica	7

% El Lenguaje de Programación Rust

¡Bienvenido! Este libro te enseñará acerca del [Lenguaje de Programación Rust](#). Rust es un lenguaje de programación de sistemas enfocado en tres objetivos: seguridad, velocidad y concurrencia. Rust logra estos objetivos sin tener un recolector de basura, haciendolo util para un numero de casos de uso para los cuales otros lenguajes no son tan buenos: embeber en otros lenguajes, programas con requerimientos específicos de tiempo y espacio, escritura de código de bajo nivel, como controladores de dispositivo y sistemas operativos. Rust mejora por sobre los lenguajes actuales en este nicho a través de un numero de chequeos en tiempo de compilación que no incurren ninguna penalidad en tiempo de ejecución, eliminando al mismo tiempo las condiciones de carrera. Rust también implementa 'abstracciones con cero costo', abstracciones que se sienten como las de un lenguaje de alto nivel. Aun así, Rust permite control preciso tal y como un lenguaje de bajo nivel lo haria.

“El Lenguaje de Programación Rust” esta dividido en siete secciones. Esta introducción es la primera. Después de esta:

- [Primeros Pasos](#) - Configura tu maquina para el desarrollo en Rust.
- [Aprende Rust](#) - Aprende programación en Rust a través de pequeños proyectos.
- [Rust Efectivo](#) - Conceptos de alto nivel para escribir excelente código Rust.
- [Sintaxis y Semántica](#) - Cada pieza de Rust, dividida en pequenos pedazos.
- [Rust Nocturno](#) - Características todavía no disponibles en builds estables.
- [Glosario](#) - Referencia de los términos usados en el libro.
- [Investigación Académica](#) - Literatura que influencio Rust.

Después de leer esta introducción, dependiendo de tu preferencia, querrás leer ‘Aprende Rust’ o ‘Sintaxis y Semántica’: ‘Aprende Rust’ si quieres comenzar con un proyecto o ‘Sintaxis y Semántica’, si prefieres comenzar por lo más pequeño aprendiendo un único concepto detalladamente antes de moverte al siguiente. Abundantes enlaces cruzados conectan dichas partes.

Contribuyendo

Los archivos fuente de los cuales este libro es generado pueden ser encontrados en Github: github.com/goyox86/elpr-sources

Una breve introducción a Rust

¿Es Rust un lenguaje en el cual estarías interesado? Examinemos unos pequeños ejemplos de código que demuestran algunas de sus fortalezas.

Esta es una muestra, haga clic en el enlace de descarga para obtener el tutorial completo

