



Heavy Duty Builders

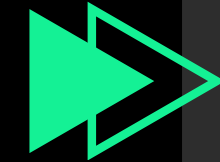
De Rust a Solana

por:
Dayana Rodrigues

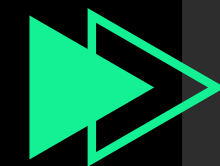
¿Qué es Solana?

- Plataforma blockchain de alta velocidad y eficiencia
- Capaz de manejar miles de transacciones por segundo
- Utiliza un mecanismo de consenso Proof of History

¿Qué es un contrato inteligente?

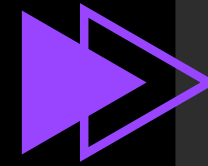


Programa que se ejecuta automáticamente en una blockchain

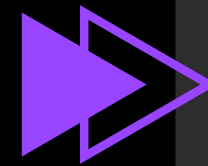


Permite realizar transacciones entre dos o más partes sin necesidad de intermediarios

¿Qué son las dApps?

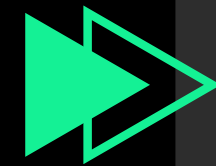


Son aplicaciones que funcionan en una blockchain

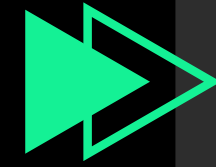


Combinan contratos inteligentes e interfaces de usuario tradicionales

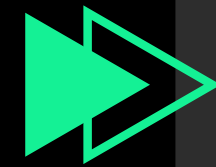
¿Por qué Solana?



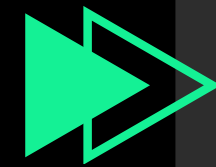
Escalabilidad y rendimiento excepcional



Bajos costos de transacción



Frameworks y herramientas



Comunidad y ecosistema en crecimiento

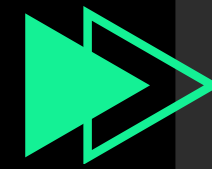
Conceptos básicos

▶▶ Modelo de cuentas de solana

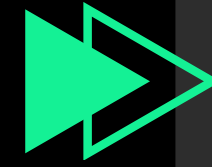
▶▶ Programas en solana

▶▶ Cómo desarrollar en Solana con Rust

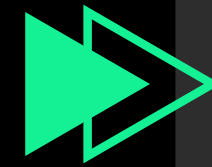
Modelo de Cuentas



Todo en Solana se gestiona a través de cuentas.

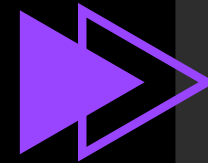


Cada cuenta tiene un propietario con la autoridad para modificar su contenido

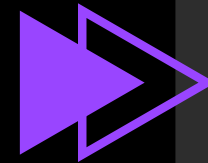


Las cuentas deben tener suficiente saldo para cubrir el coste de la renta

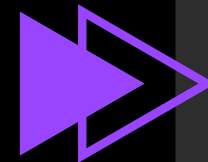
Programas en Solana



Son equivalentes a los contratos inteligentes en otras blockchains

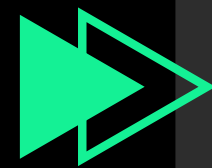


Invocados a través de transacciones que contienen instrucciones

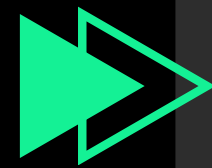


Se dividen en: Programas del sistema y Programas de usuarios.

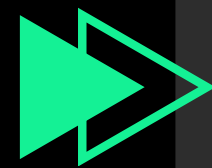
Cómo desarrollar en Solana con Rust



Solana CLI

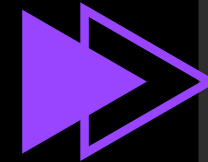


Solana Program Library
(SPL)

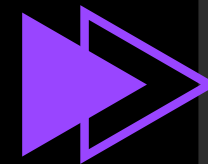


Rust

Solana CLI



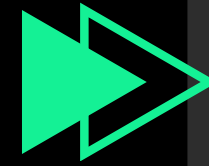
Proporciona un conjunto de comandos para interactuar con la blockchain de Solana



Permite enviar transacciones, consultar el estado de la red, gestionar cuentas, desplegar programas, etc.



Solana Program Library (SPL)

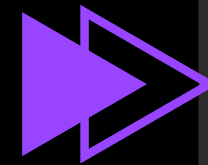


Conjunto de programas y bibliotecas reutilizables para la blockchain de Solana

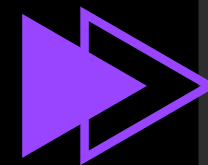


Proporciona herramientas y estándares que ayudan a desarrollar programas y dApps de manera más eficiente

Rust

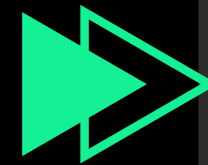


Es el lenguaje de programación más utilizado para escribir programas en Solana

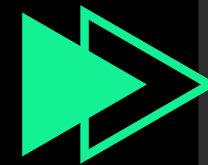


Solana tiene herramientas y bibliotecas diseñadas específicamente para este lenguaje.

Escribiendo Programas con Rust

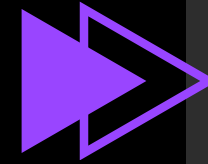


Rust es un lenguaje oficialmente compatible para escribir programas on-chain en Solana.

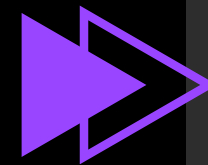


Existen varias diferencias clave en el uso del lenguaje al momento de desarrollar programas en Solana

Dependencias del proyecto

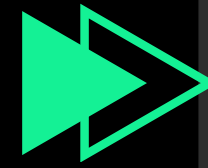


El crate
solana-program es
requerido para todo
programa escrito en
Rust

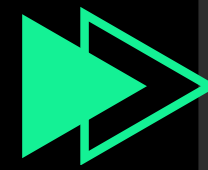


Esta es la biblioteca
base para todos los
programas on-chain
en Rust

Punto de entrada del programa



En lugar de una función principal main, los programas de Solana utilizan la macro `entrypoint!`



Este símbolo es exportado y luego llamado por el runtime de Solana cuando el programa se ejecuta

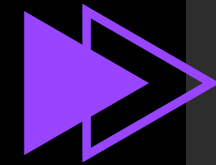
rust

```
entrypoint!(process_instruction);
```

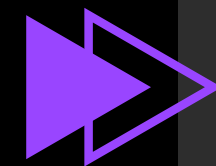
rust

```
pub fn process_instruction(  
    program_id: &Pubkey,  
    accounts: &[AccountInfo],  
    instruction_data: &[u8],  
) -> ProgramResult {  
  
    //program code goes here  
  
}
```

Construcción y pruebas



Para compilar el programa al bytecode necesario para ejecutarse en Solana se utilizan los comandos: `cargo build-bpf` o `cargo build-spf`

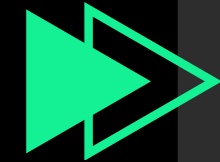


El `solana-test-validator` crea un clúster de pruebas local, donde desplegar programas y ejecutar pruebas.

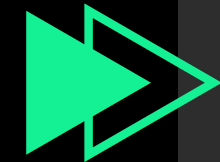
```
→ ~ solana config set --url localhost
Config File: /home/dayanarp/.config/solana/cli/config.
yaml
RPC URL: http://localhost:8899
WebSocket URL: ws://localhost:8900/ (computed)
Keypair Path: /home/dayanarp/.config/solana/id.json
Commitment: confirmed
→ ~ solana-test-
```

```
→ ~
```

Limitaciones de paquetes

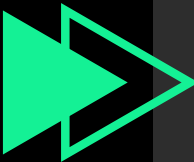


El runtime de Solana tiene restricciones de recursos y debe ejecutarse de manera determinista.

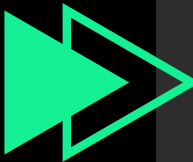


Esto limita el uso de algunos crates estándar de Rust

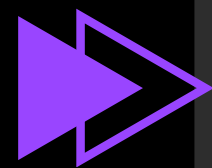
Paquetes no disponibles:

- 
- rand
 - std::fs
 - std::net
 - std::future
 - std::process
 - std::sync
 - std::task
 - std::thread
 - std::time

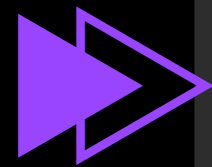
Paquetes con funcionalidad limitada:

- 
- std::hash
 - std::os

Cambios en macros

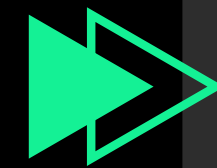


La macro `println!` es reemplazada por la macro `msg!` que se utiliza para registrar mensajes en los logs del programa

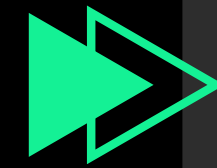


Las macros `panic!`, `assert!`, y cualquier pánico interno también se registran en los logs del programa por defecto

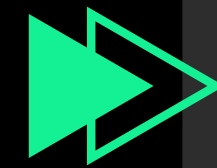
Frameworks y entornos de desarrollo



Solana Playground

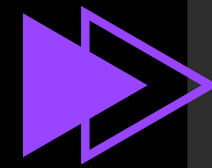


Framework Anchor

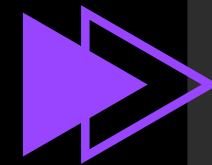


Framework Steel

Solana Playground



IDE basado en el navegador que permite crear, probar y desplegar programas de Solana.



Admite tanto Rust nativo como el desarrollo con Anchor

Solana Playground | Solana IDE

beta.solpg.io

BUILD & DEPLOY

Build

Program ID

Import/export program keypair or input a public key for the program.

NewImport

Program ID:

Your program's public key

Note that you need to have this program's authority to upgrade

Program binary

IDL

Resources

Cookbook

Detailed explanations and guides for building applications on Solana.

Learn more

Anchor

Everything related to developing on Solana with Anchor framework.

Learn more

Seahorse

Write Anchor-compatible Solana programs in Python.

Learn more

SolDev

Solana content aggregator with easy discoverability for all your development needs.

Learn more

Solana Docs

The core Solana documentation used to provide deep understanding of Solana concepts.

Learn more

Metaplex Docs

Documentation for understanding how to work with NFTs on Solana using the Metaplex Standards.

Learn more

Tutorials

Popular crates for Solana development are available to use.

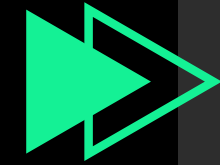
See the list of available crates and request new crates from <https://github.com/solana-playground/solana-playground>

Type `help` to see all commands.

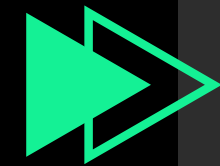
\$

Not connected

Framework Anchor

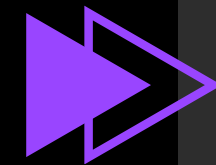


Te permite crear programas en Solana de forma rápida utilizando Rust.

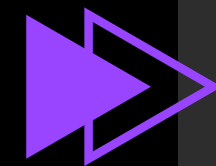


Realiza ciertas comprobaciones de seguridad de forma automática.

Framework Steel

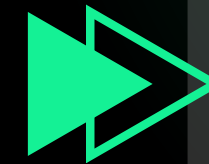


Facilita la organización de proyectos al proporcionar una estructura predefinida

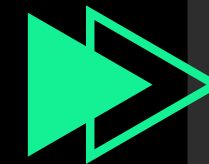


Ofrece una variedad de funciones reutilizables para analizar cuentas, validar estados y ejecutar CPIs

Recursos para desarrolladores



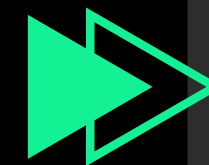
Documentación oficial de Solana



Libro de cocina de Solana



Cursos en línea gratuitos



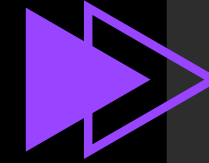
Repositorios públicos en GitHub

Algunos Recursos

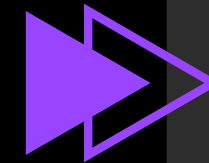


Heavy Duty Builders

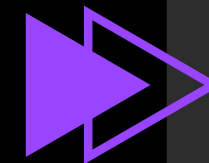
@HeavyDutyBuild



Comunidad global de
desarrolladores



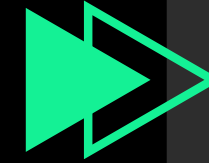
Apasionados de las
nuevas tecnologías



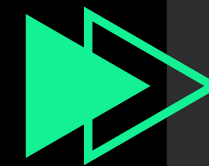
Formaciones gratuitas
durante todo el año

La Familia

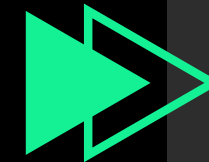
@LaFamilia_so



Comunidad española
de Solana



Busca impulsar
proyectos españoles
en el ecosistema



100% en español

iGracias!

Dayana Rodríguez
@dayanarp_



Heavy Duty Builders



<https://linktr.ee/heavydutybuilders>