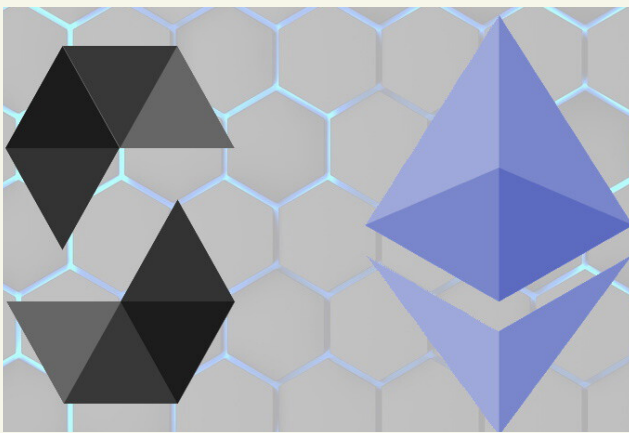
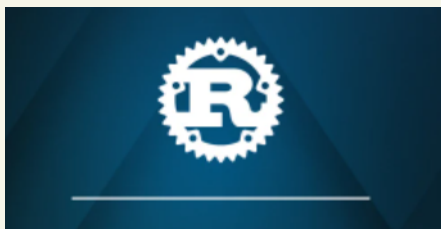


LENGUAJES DE PROGRAMACIÓN PARA EL DESARROLLO DE CONTRATOS INTELIGENTES



LENGUAJES PARA PROGRAMAR CONTRATOS INTELIGENTES

*Solidity
*Vyper
*Rholang
*Simplicity
*JavaScript
*C#
*Golang
*Java
*Python
*C++



CARACTERÍSTICAS DE LOS LENGUAJES RUST Y ASSEMBLY



Características de Rust:

- Ejecución dinámica de seguridad (errores y registros).
- Orientado a Objetos.
- Interfaz simple.
- Gestión automática de guardado.
- Inmutable.
- Compilación nativa y estática.
- Multiplataforma.
- Control de la memoria explícita.
- Permite cadenas UTF8.
- Multiparadigmático.
- Concurrente.

Característica de Assembly

- Puede usar código de operación mnemónico que numérico y también proporciona la información de cualquier error en el código.
- Este lenguaje ayuda a especificar el operando simbólico, lo que significa que no necesita especificar la dirección de la máquina de ese operando. Se puede representar en forma de símbolo.
- Los datos se pueden declarar mediante notación decimal.

VENTAJAS DE LOS LENGUAJES

- Seguridad de la memoria
- Inmutabilidad por defecto y mutabilidad explícita
- Distinción clara entre código seguro e "inseguro"
- Herramientas integradas y listas para usar
- Rigidez vía un estricto sistema de tipos
- Como trabaja directamente con el microprocesador al ejecutar un programa, pues como este lenguaje es el mas cercano a la máquina la computadora lo procesa mas rápido.
- Eficiencia de tamaño
- Flexibilidad porque todo lo que puede hacerse con una máquina, puede hacerse en el lenguaje ensamblador de esta máquina

APLICACIONES DE LOS LENGUAJES



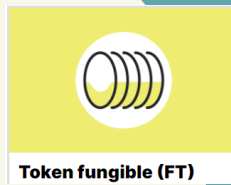
Grifo de prueba de trabajo



Ejemplo de integración de NEAR Wallet



Ejemplo de contador en AssemblyScript



Token fungible (FT)



Mensaje de estado



Token no fungible



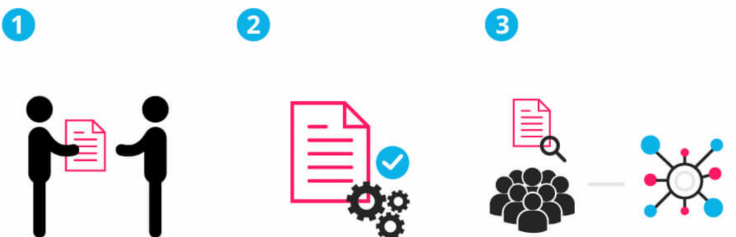
Libro de visitas



Contraejemplo en Rust

ARQUITECTURA DE UN CONTRATO INTELIGENTE CON EL PROTOCOLO NEAR

- Implementación de block-end
- Estado-almacenamiento
- Costos de operación
- Entorno Blockchain



Bibliografía.

- {1}<https://ethereum.org/es/developers/docs/smart-contracts/languages/>
[2]<https://examples.near.org/>