

# Temario

---

- Tipos de Blockchain
- Smart Contract
- Substrate & !Ink

# Tipos de blockchain

---

- 1era Gen
- 2da Gen
- 3era Gen

# Tipos de blockchain: 1era Gen

---

## Bitcoin(2009)

- 1 bloque c/10 min
- 6 confirmaciones
- cada bloque peso max 2MB (2500 txns aprox.)
- transfers

whitepaper: <https://bitcoin.org/bitcoin.pdf>

# Tipos de blockchain: 2da Gen

---

## Ethereum(2015)

- 1 bloque c/15 segs aprox
- 64 confirmaciones
- cada bloque 200 txns aprox
- transfers y ejecución de código(smart contracts)

whitepaper: <https://ethereum.org/en/whitepaper/>

# Tipos de blockchain: 3ra Gen

---

## Polkadot(2020)

- Multichain (Relay Chain)
- Parachain
- Bridges

whitepaper: <https://polkadot.network/whitepaper/>

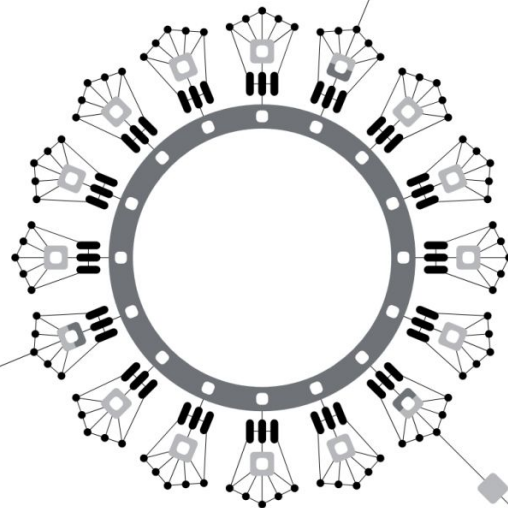
# Polkadot: arquitectura

Arquitectura de Polkadot

9 ●

## Conectando los puntos

Polkadot une una red de fragmentos heterogéneos de cadenas de bloques llamados parachains. Estas cadenas se conectan y son aseguradas por la Relay chain de Polkadot. También pueden conectarse con redes externas a través de puentes.



# Polkadot: arquitectura



## **Relay Chain**

El corazón de Polkadot, responsable de la seguridad de la red, el consenso y la interoperabilidad de la cadena cruzada.



## **Parachains**

Cadenas de bloques soberanos que pueden tener sus propios tokens y optimizar su funcionalidad para casos de uso específicos. Para conectarse a la Relay Chain, las Parachains pueden pagar a medida que avanzan o alquilan un espacio para una conectividad permanente.



## **Bridges**

Cadenas de bloques especiales que permiten a los fragmentos de Polkadot conectarse y comunicarse con redes externas como Ethereum y Bitcoin.

# Smart contract

---

Es código que se ejecuta de manera automática y autónoma en una blockchain.

Está escrito en un lenguaje de programación específico y se almacena en la blockchain como parte de un contrato digital. Una vez desplegado en la blockchain, el smart contract se ejecuta automáticamente cuando se cumplen ciertas condiciones predefinidas.

El propósito es automatizar y asegurar la ejecución de acuerdos sin necesidad de confiar en terceros. Al estar basados en tecnología blockchain, los smart contracts son transparentes, inmutables y verificables por todos los participantes de la red. Esto brinda confianza y reduce la necesidad de intermediarios, lo que puede agilizar y simplificar procesos comerciales y legales.



# Smart contract

---

Lenguajes de programación:

- Solidity - Ethereum
- Vyper - Ethereum
- Rust - Polkadot, Solana, Cosmos
- Plutus - Cardano

# Polkadot - Substrate & !Ink

---

Substrate es un framework para poder crear una parachain sobre polkadot :

<https://substrate.io/vision/substrate-and-polkadot/>

!Ink es un sdk que nos permite desarrollar smart contracts sobre blockchain construidas con substrate:

<https://use.ink/es/>

# !Ink: tools local

---

1- install: <https://crates.io/crates/cargo-contract>

2- en vez de rust-analyzer se puede usar <https://crates.io/crates/ink-analyzer-ir>

3- comandos:

- a- cargo contract new nombre\_del\_contrato

- b- cargo contract build

- c- cargo test

# !Ink: testnet

---

1- wallets: <https://wiki.polkadot.network/docs/wallets>

-firefox: <https://addons.mozilla.org/es/firefox/addon/polkadot-js-extension/>

-otros(chrome): <https://polkadot.js.org/extension/>

-firefox: <https://addons.mozilla.org/en-US/firefox/addon/polkgate/>

-otros(chrome, etc):<https://polkgate.xyz/>

2- faucet a zero: <https://faucet.test.azero.dev/>

3- explorer a zero: <https://polkadot.js.org/apps/?rpc=wss%3A%2F%2Fws.test.azero.dev#/accounts>

4- cliente deploy testnet a zero: <https://contracts-ui.substrate.io/?rpc=wss://ws.test.azero.dev>

5- ink! : <https://use.ink/es/>

extra: llamar de un contrato a otro contrato: <https://use.ink/basics/cross-contract-calling>

# !Ink

---

faucet a zero: <https://faucet.test.azero.dev/>

explorer a zero: <https://polkadot.js.org/apps/?rpc=wss%3A%2F%2Fws.test.azero.dev#/accounts>

cliente deploy testnet a zero: <https://contracts-ui.substrate.io/?rpc=wss://ws.test.azero.dev>

cargo contract new nombre\_del\_contrato

cargo contract build

llamar de un contrato a otro contrato: <https://use.ink/basics/cross-contract-calling>

# !Ink

---

faucet a zero: <https://faucet.test.azero.dev/>

explorer a zero: <https://polkadot.js.org/apps/?rpc=wss%3A%2F%2Fws.test.azero.dev#/accounts>

cliente deploy testnet a zero: <https://contracts-ui.substrate.io/?rpc=wss://ws.test.azero.dev>

cargo contract new nombre\_del\_contrato

cargo contract build

llamar de un contrato a otro contrato: <https://use.ink/basics/cross-contract-calling>