

&

Construyendo APIs Descentralizados

Camila Ramos, Dev Rel @ Edge & Node



@camiinthisthang

&

Camila Ramos

EDUCADOR
DEVELOPER
INGENIERÍA @ PAYPAL

&

Construir Un API Decentralized

GRAPHQL IN WEB 3

@camiinthisthang

Blockchain 101

Un blockchain es una red
distribuida, peer-to-peer de
transacciones públicas

Blockchain 101

Un blockchain es una **red distribuida, peer-to-peer de transacciones publicas.**

Blockchain 101

Un blockchain es una red distribuida, **peer-to-peer** de transacciones públicas.

Blockchain 101

Un blockchain es una red distribuida, peer-to-peer de transacciones públicas.

Que Es un API?

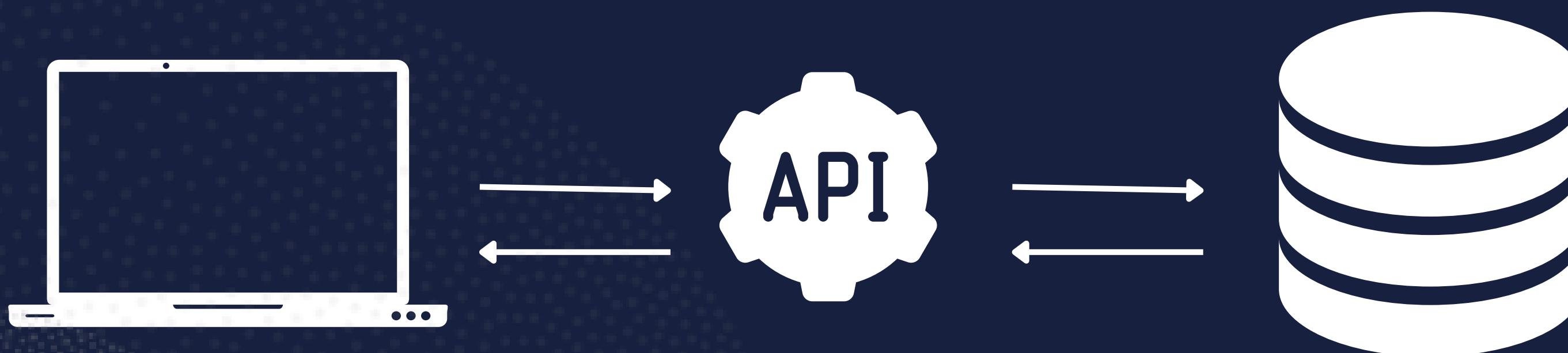
API - Interfaz de programación de aplicaciones

Web API - Un mecanismo para que los servicios web se comuniquen entre sí a través del protocolo HTTP.

Read about web APIs, the client-server architecture, and http request methods

&

GraphQL en Web2



GraphQL en Web2

- ▶ Fuente de datos agnóstico
- ▶ Leer de una base de datos/obtener datos de una
- ▶ API REST

Resolvers aplican la lógica para construir y devolver un objeto definido en su esquema

GraphQL en Web3

- ▶ Cadena agnóstica
- ▶ Leer de una blockchain
- ▶ Resolvers aplican la lógica para construir y devolver un objeto definido en su esquema

&

Lens Subgraph

&

?????????

@camiinthisthang

15,095,760

&

**Puedes incluir ~70
transacciones o pagos básicos
en un bloque.**

&

**Los datos que desea
consultar están aquí... en
alguna parte**

The Graph



The Graph es un protocolo de indexación y query que se utiliza para leer de manera eficiente los datos de una blockchain y los expone vía el GraphQL query API.

&

The Graph



The Graph es un protocolo de indexación y query que se utiliza para leer **de manera eficiente** los datos de una blockchain y los expone vía el GraphQL query API.

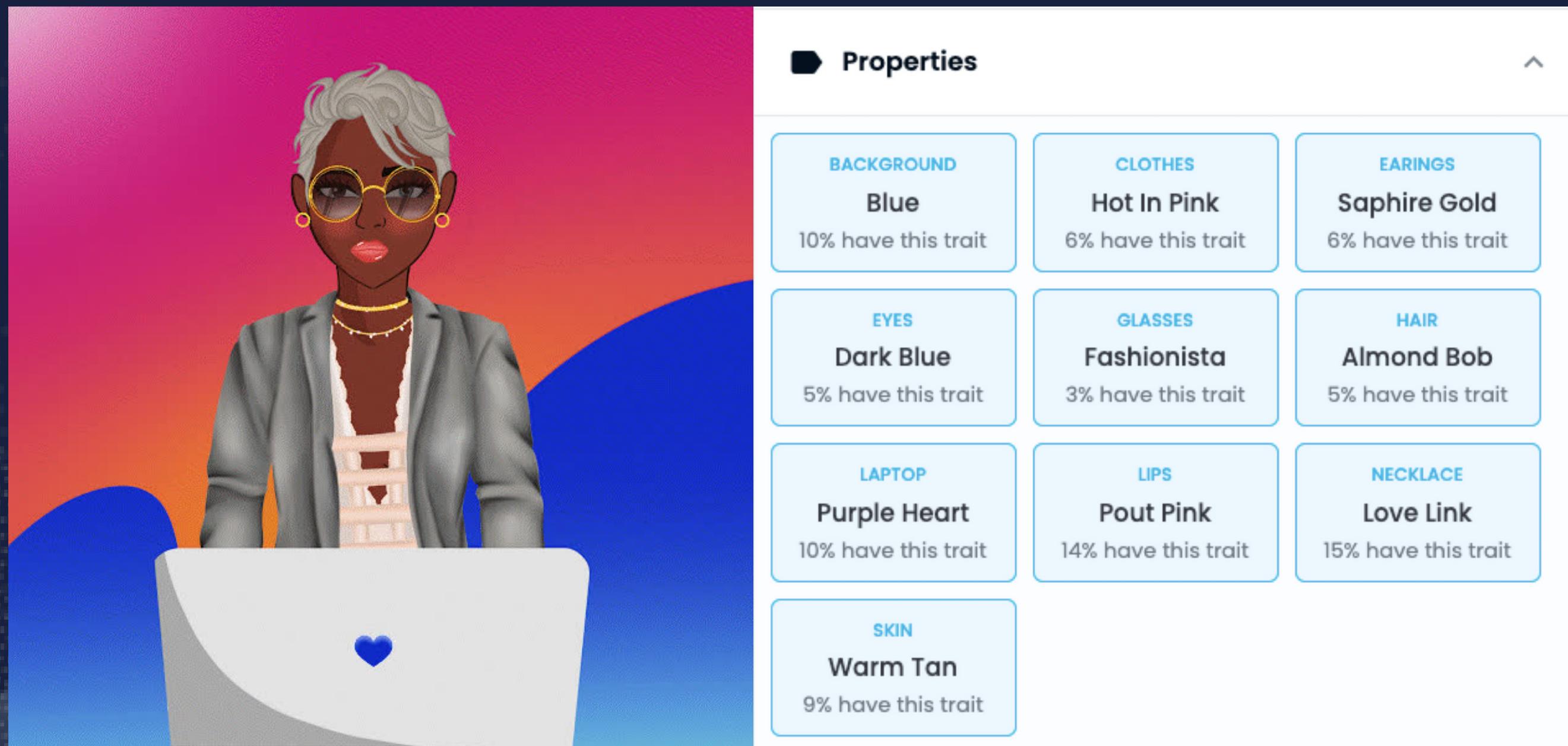
What is Indexing?

La indexación generalmente se refiere al proceso de convertir datos no ordenados en un orden que maximizará la eficiencia de una consulta realizada contra esos datos.

&

Construyendo un Web3 API Sin The Graph

Escenario: desea crear una API que permita a los usuarios buscar un NFT con rasgos específicos.



&

Construyendo un Web3 API Sin The Graph

Escenario 1:

- Tome cada ID de token único que se asigna a una Mujer
- Leer los metadatos
- Agregar
- Escribir un programa para filtrar

&

Construyendo un Web3 API Sin The Graph

Escenario 2:

- Manejar su propio servidor
- Procesar y almacenar transacciones realizadas en el blockchain
- Guardar en una base de datos
- Índice de base de datos para tiempos de consulta razonables
- Cree la API para exponer datos a través de un punto final

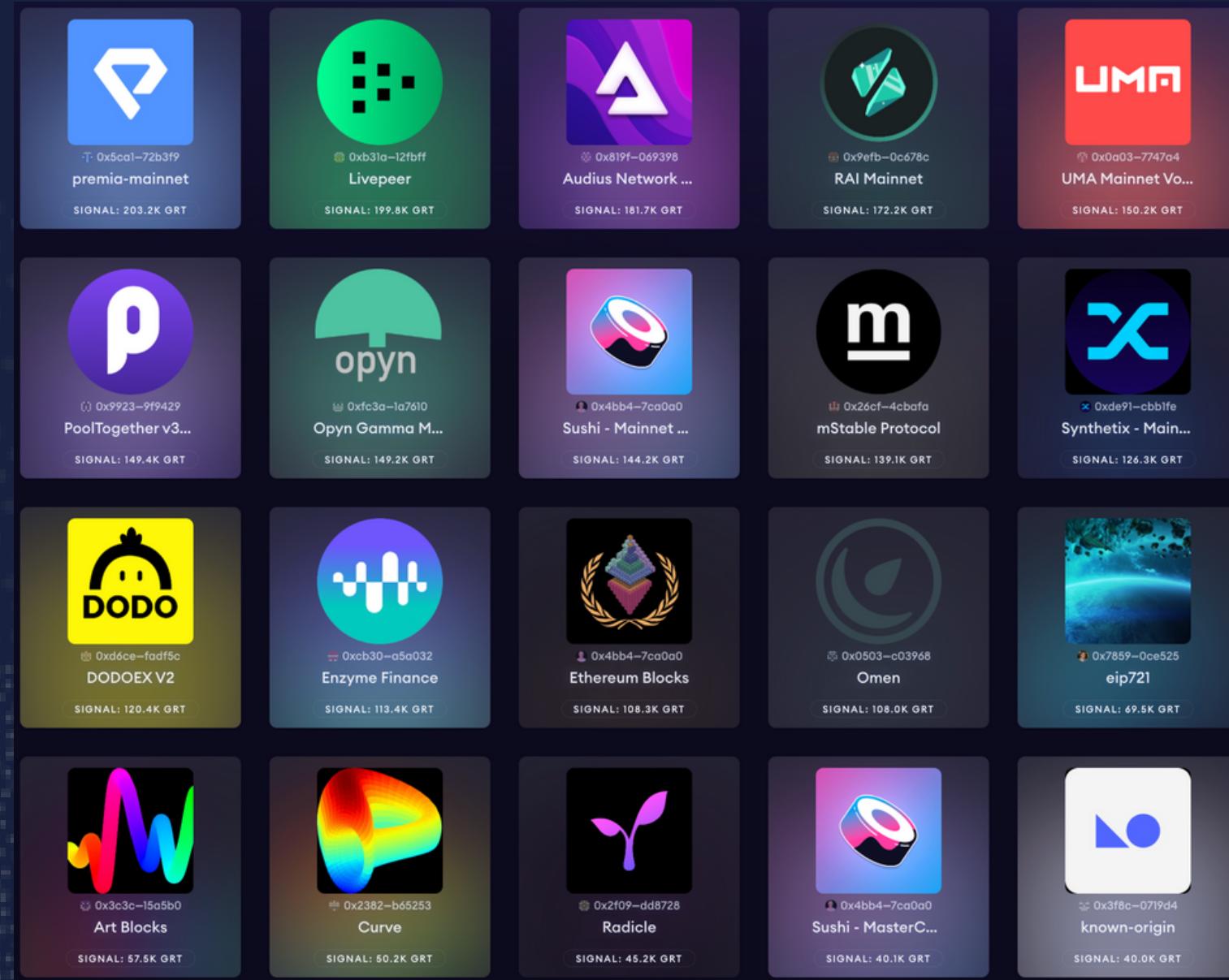
&

Construir APIs Con The Graph

FILTRADO, ASIGNACIÓN DE RELACIONES, BÚSQUEDA DE TEXTO
COMPLETO

&

Creando un Subgraph



Cualquiera puede desarrollar un subgraph: ¡hay cientos de subgraphs ya implementados y cualquiera puede acceder a su API después de que la haya implementado.

& Subgraph en el Stack



Crear Un Subgraph

Para construir un API descentralizada con The Graph, un desarrollador crea un subgraph que define:

- la fuente de datos
- Los datos que nos interesa
- Eventos que desea escuchar
- Resolvers que se ejecutan cuando se emiten eventos

Solidity Events

"Un evento es un miembro heredable del contrato, que guarda los argumentos pasados en los registros de transacciones cuando se emite. Generalmente, los eventos se utilizan para informar a la aplicación que llama sobre el estado actual del contrato, con la ayuda de la función de registro de EVM ." – Geeksforgeeks

Manifest

- bloque de inicio
- nombre de los entidades
- eventos y event handlers

```
! subgraph.yaml
1   specVersion: 0.0.2
2   schema:
3     file: ./schema.graphql
4   dataSources:
5     - kind: ethereum
6       name: Token
7       network: mainnet
8       source:
9         address: "0xf3E6DbBE461C6fa492CeA7Cb1f5C5eA660EB1B47"
10      abi: Token
11      startBlock: 12813031
12   mapping:
13     kind: ethereum/events
14     apiVersion: 0.0.5
15     language: wasm/assemblyscript
16   entities:
17     - Lady
18     - User
19   abis:
20     - name: Token
21       file: ./abis/Token.json
22   eventHandlers:
23     - event: Transfer(indexed address, indexed address, indexed uint256)
24       handler: handleTransfer
25   file: ./src/mapping.ts
26
```

&

Schema

Modelos de datos definidos por el desarrollador en los que ELLOS ESTÁN interesados para SU CASO DE USO ESPECÍFICO



&

Schema

```
schema.graphql
1 type Lady @entity {
2   id: ID!
3   tokenID: BigInt!
4   tokenURI: String!
5   owner: User
6   name: String
7   faceExpression: String
8   hairStyle: String
9   skinColor: String
10  eyeColor: String
11 }
12
13 type User @entity {
14   id: ID! # wallet address
15   Ladies: [Lady!] @derivedFrom(field: "owner")
16 }
```

&

Resolvers

Lógica para construir y devolver los objetos definidos en su esquema a su cliente.



@camiinthisthang



AssemblyScript Mappings

```
if (metadata) {
    const value = json.fromBytes(metadata).toObject();
    if (value) {
        const name = value.get("name");
        if (name) {
            lady.name = name.toString();
        }
    }
    let attributes: JSONValue[];
    let ladyAttributes = value.get("attributes");
    if (ladyAttributes) {
        attributes = ladyAttributes.toArray();

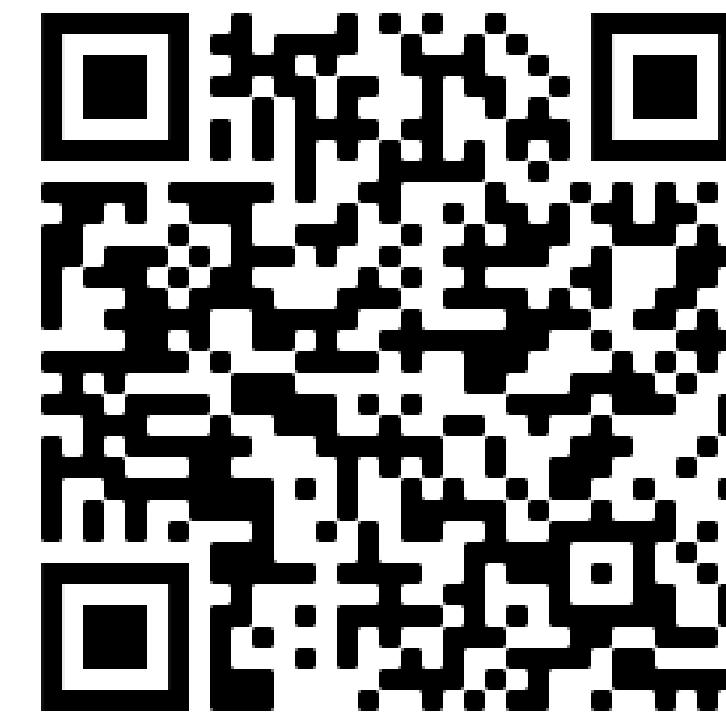
        for (let i = 0; i < attributes.length; i++) {
            let item = attributes[i].toObject();
            let trait: string;
            let traitName = item.get("trait_type");
            if (traitName) {
                trait = traitName.toString();
                let value: string;
                let traitValue = item.get("value");
                if (traitValue) {
                    value = traitValue.toString();
                    if (trait == "Hair") {
                        lady.hairStyle = value;
                    }
                    if (trait == "Skin") {
                        lady.skinColor = value;
                    }
                    if (trait == "Eyes") {
                        lady.eyeColor = value;
                    }
                    if (trait == "Face Expression") {
                        lady.faceExpression = value;
                    }
                }
            }
        }
    }
}
```

&

Construyamos

@camiinthisthang

&



github.com/camiinthisthang/thegraph-workshop

@camiinthisthang

Usando una Subgraph en un Dapp

- Query desde una aplicación React con bibliotecas GraphQL Client, es decir, Apollo Client o URQL

```
const LENS_QUERY = gql`  
query getUserInfo {  
  profiles(first: 20) {  
    imageURI  
    handle  
    totalFollowers  
    totalFollowings  
    totalPosts  
  }  
}
```

```
export default function Home() {  
  const { loading, error, data } = useQuery(LENS_QUERY);  
  console.log("Data", data);
```

&

Query varios subgraphs con Graph Client

- Query de varios subgraphs de diferentes chains
- Combinar consultas de varios subgraphs

&

Empieza A Construir

NEAR - \$800 million

Harmony One - \$300 million

Solana - Large treasury, up to \$5 million for past projects

The Graph - Over \$100 million in past 12 months

Celo - Over \$100 million

Sandbox - \$1 billion

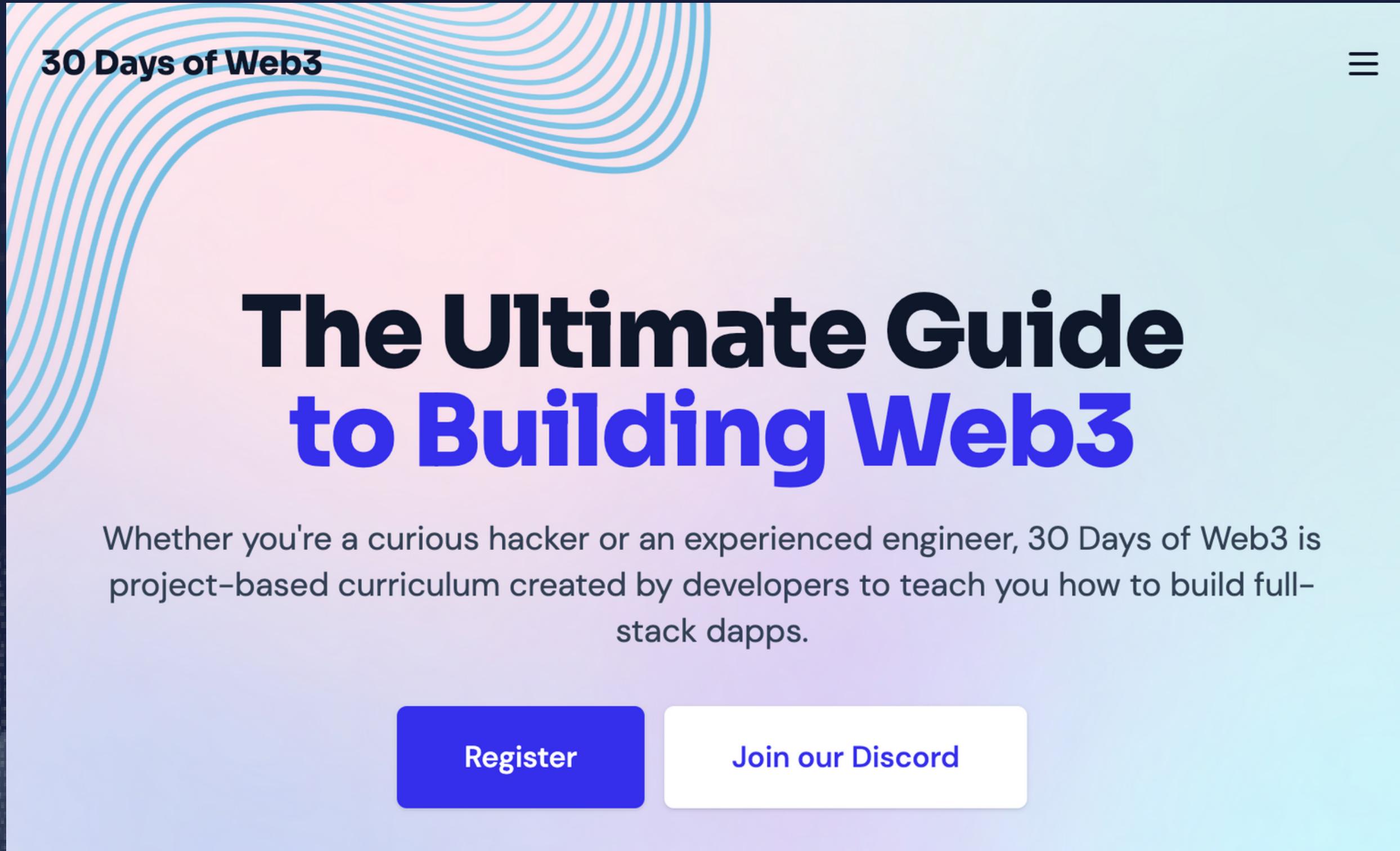
Polygon

Ethereum Foundation

Chainlink

&

30 Days of Web3 - Building Fullstack Dapps



The screenshot shows the landing page for '30 Days of Web3'. The top left features a blue wavy graphic with the text '30 Days of Web3' in white. The top right has a three-line menu icon. The main title 'The Ultimate Guide to Building Web3' is displayed in large, bold, dark blue and purple letters. Below the title is a descriptive paragraph: 'Whether you're a curious hacker or an experienced engineer, 30 Days of Web3 is a project-based curriculum created by developers to teach you how to build full-stack dapps.' At the bottom are two buttons: a blue 'Register' button and a white 'Join our Discord' button.

30 Days of Web3

The Ultimate Guide
to Building Web3

Whether you're a curious hacker or an experienced engineer, 30 Days of Web3 is a project-based curriculum created by developers to teach you how to build full-stack dapps.

Register

Join our Discord

@camiinthisthang

&

Resources

The Developer's Guide To Getting
Started With The Graph

Calling a Subgraph from A Client
Application

&

Gracias!



@camiinthisthang



@camiinthisthang



@camiinthisthang