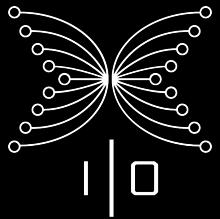


INPUT | OUTPUT

# Web2 -> Web3

Cómo transformar un proyecto NEXTjs de Web2 a Web3



01

WEB 1, 2, 3

## **Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS**

---

<b>WEB 1</b>	<b>WEB 2</b>	<b>WEB 3</b>

## **Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS**

---

<b>WEB 1</b>	<b>WEB 2</b>	<b>WEB 3</b>
Centrado alrededor de "leer"		

## **Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS**

---

<b>WEB 1</b>	<b>WEB 2</b>	<b>WEB 3</b>
Centrado alrededor de "leer"	Centrado alrededor de "leer", "interactuar", y "publicar"	

## **Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS**

---

<b>WEB 1</b>	<b>WEB 2</b>	<b>WEB 3</b>
Centrado alrededor de "leer"	Centrado alrededor de "leer", "interactuar", y "publicar"	Centrado alrededor de "leer", "interactuar", "publicar", y "poseer"

## **Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS**

---

<b>WEB 1</b>	<b>WEB 2</b>	<b>WEB 3</b>
Centrado alrededor de "leer"	Centrado alrededor de "leer", "interactuar", y "publicar"	Centrado alrededor de "leer", "interactuar", "publicar", y "poseer"
Páginas estáticas		

## **Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS**

---

<b>WEB 1</b>	<b>WEB 2</b>	<b>WEB 3</b>
Centrado alrededor de "leer"	Centrado alrededor de "leer", "interactuar", y "publicar"	Centrado alrededor de "leer", "interactuar", "publicar", y "poseer"
Páginas estáticas	Páginas y aplicaciones dinámicas	

## **Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS**

---

<b>WEB 1</b>	<b>WEB 2</b>	<b>WEB 3</b>
Centrado alrededor de "leer"	Centrado alrededor de "leer", "interactuar", y "publicar"	Centrado alrededor de "leer", "interactuar", "publicar", y "poseer"
Páginas estáticas	Páginas y aplicaciones dinámicas	Páginas y aplicaciones dinámicas y "descentralizadas"

## **Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS**

---

<b>WEB 1</b>	<b>WEB 2</b>	<b>WEB 3</b>
Centrado alrededor de "leer"	Centrado alrededor de "leer", "interactuar", y "publicar"	Centrado alrededor de "leer", "interactuar", "publicar", y "poseer"
Páginas estáticas	Páginas y aplicaciones dinámicas	Páginas y aplicaciones dinámicas y "descentralizadas"
Páginas renderizadas en el servidor (el cliente no hacía mucho)		

## **Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS**

---

<b>WEB 1</b>	<b>WEB 2</b>	<b>WEB 3</b>
Centrado alrededor de "leer"	Centrado alrededor de "leer", "interactuar", y "publicar"	Centrado alrededor de "leer", "interactuar", "publicar", y "poseer"
Páginas estáticas	Páginas y aplicaciones dinámicas	Páginas y aplicaciones dinámicas y "descentralizadas"
Páginas renderizadas en el servidor (el cliente no hacía mucho)	Tanto el servidor como el cliente trabajan	

## **Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS**

---

<b>WEB 1</b>	<b>WEB 2</b>	<b>WEB 3</b>
Centrado alrededor de "leer"	Centrado alrededor de "leer", "interactuar", y "publicar"	Centrado alrededor de "leer", "interactuar", "publicar", y "poseer"
Páginas estáticas	Páginas y aplicaciones dinámicas	Páginas y aplicaciones dinámicas y "descentralizadas"
Páginas renderizadas en el servidor (el cliente no hacía mucho)	Tanto el servidor como el cliente trabajan	Idem Web 2, pero con mayor descentralización

# **Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS**

---

<b>WEB 1</b>	<b>WEB 2</b>	<b>WEB 3</b>
Centrado alrededor de "leer"	Centrado alrededor de "leer", "interactuar", y "publicar"	Centrado alrededor de "leer", "interactuar", "publicar", y "poseer"
Páginas estáticas	Páginas y aplicaciones dinámicas	Páginas y aplicaciones dinámicas y "descentralizadas"
Páginas renderizadas en el servidor (el cliente no hacía mucho)	Tanto el servidor como el cliente trabajan	Idem Web 2, pero con mayor descentralización
HTML es 		

# Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS

---

WEB 1	WEB 2	WEB 3
Centrado alrededor de "leer"	Centrado alrededor de "leer", "interactuar", y "publicar"	Centrado alrededor de "leer", "interactuar", "publicar", y "poseer"
Páginas estáticas	Páginas y aplicaciones dinámicas	Páginas y aplicaciones dinámicas y "descentralizadas"
Páginas renderizadas en el servidor (el cliente no hacía mucho)	Tanto el servidor como el cliente trabajan	Idem Web 2, pero con mayor descentralización
HTML es 	HTML + JS + JSON comparten la 	

# Web 1, 2, y 3 : Diferencias entre Web 1, 2, y 3 - A GRANDES RASGOS

---

WEB 1	WEB 2	WEB 3
Centrado alrededor de "leer"	Centrado alrededor de "leer", "interactuar", y "publicar"	Centrado alrededor de "leer", "interactuar", "publicar", y "poseer"
Páginas estáticas	Páginas y aplicaciones dinámicas	Páginas y aplicaciones dinámicas y "descentralizadas"
Páginas renderizadas en el servidor (el cliente no hacía mucho)	Tanto el servidor como el cliente trabajan	Idem Web 2, pero con mayor descentralización
HTML es 	HTML + JS + JSON comparten la 	Idem Web 2

# **MARKETING!**

La evolución de la web es constante y continua (no discreta):

<https://thehistoryoftheweb.com/timeline/>

**Y no hay una razón técnica real para trazar la línea en ningún lado.**

## **Web 1, 2, y 3 : Qué le digo al jefe/inversor/gerente no técnico?**

---

## **Web 1, 2, y 3 : Qué le digo al jefe/inversor/gerente no técnico?**

---

- **Web1:** Página web con estática.

## **Web 1, 2, y 3 : Qué le digo al jefe/inversor/gerente no técnico?**

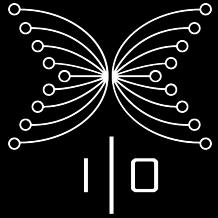
---

- **Web1:** Página web con estática.
- **Web2:** Página o aplicación web moderna e interactiva.

## **Web 1, 2, y 3 : Qué le digo al jefe/inversor/gerente no técnico?**

---

- **Web1**: Página web con estática.
- **Web2**: Página o aplicación web moderna e interactiva.
- **Web3**: Web2 + interactúa con alguna **blockchain**.



02

CREAR PROYECTO Web3

# Crear proyecto Web3 : Proyecto Web 2 + librerías que interactúan con 🏭⛓

---

## Crear proyecto Web3 : Proyecto Web 2 + librerías que interactúan con 🏭⛓

---

**Crear proyecto Web2:**

# Crear proyecto Web3 : Proyecto Web 2 + librerías que interactúan con 🏭⛓

---

## Crear proyecto Web2:

1. HTML+JS+CSS

## Crear proyecto Web2:

1. HTML+JS+CSS
2. Tu Framework favorito: NextJS , Gin, Echo, Fiber, Actix, Axum, Rocket, Tide, Vue, Elm, Flutter, Django, IHP, Play, ~~Laravel, Rails~~, etc.

### Crear proyecto Web2:

1. HTML+JS+CSS
2. Tu Framework favorito: NextJS , Gin, Echo, Fiber, Actix, Axum, Rocket, Tide, Vue, Elm, Flutter, Django, IHP, Play, ~~Laravel, Rails~~, etc.

### Instalar librerías para interactuar con la blockchain (Cardano):

## Crear proyecto Web2:

1. HTML+JS+CSS
2. Tu Framework favorito: NextJS , Gin, Echo, Fiber, Actix, Axum, Rocket, Tide, Vue, Elm, Flutter, Django, IHP, Play, ~~Laravel, Rails~~, etc.

## Instalar librerías para interactuar con la blockchain (Cardano):

1. JS: MeshJS, Plu-ts, Lucid Evolution, Blaze, etc.

## Crear proyecto Web2:

1. HTML+JS+CSS
2. Tu Framework favorito: NextJS , Gin, Echo, Fiber, Actix, Axum, Rocket, Tide, Vue, Elm, Flutter, Django, IHP, Play, ~~Laravel, Rails~~, etc.

## Instalar librerías para interactuar con la blockchain (Cardano):

1. JS: MeshJS, Plu-ts, Lucid Evolution, Blaze, etc.
2. Python: PyCardano

## Crear proyecto Web2:

1. HTML+JS+CSS
2. Tu Framework favorito: NextJS , Gin, Echo, Fiber, Actix, Axum, Rocket, Tide, Vue, Elm, Flutter, Django, IHP, Play, ~~Laravel, Rails~~, etc.

## Instalar librerías para interactuar con la blockchain (Cardano):

1. JS: MeshJS, Plu-ts, Lucid Evolution, Blaze, etc.
2. Python: PyCardano
3. Haskell: Atlas

## Crear proyecto Web2:

1. HTML+JS+CSS
2. Tu Framework favorito: NextJS , Gin, Echo, Fiber, Actix, Axum, Rocket, Tide, Vue, Elm, Flutter, Django, IHP, Play, ~~Laravel, Rails~~, etc.

## Instalar librerías para interactuar con la blockchain (Cardano):

1. JS: MeshJS, Plu-ts, Lucid Evolution, Blaze, etc.
2. Python: PyCardano
3. Haskell: Atlas
4. Scala: Scalus

## Crear proyecto Web2:

1. HTML+JS+CSS
2. Tu Framework favorito: **NextJS**, Gin, Echo, Fiber, Actix, Axum, Rocket, Tide, Vue, Elm, Flutter, Django, IHP, Play, ~~Laravel~~, ~~Rails~~, etc.

## Instalar librerías para interactuar con la blockchain (Cardano):

1. JS: **MeshJS**, Plu-ts, Lucid Evolution, Blaze, etc.
2. Python: PyCardano
3. Haskell: Atlas
4. Scala: Scalus

## Crear proyecto Web3 : Proyecto Web 2 + librerías que interactúan con 🏭⛓

---

- Crear proyecto NextJS (sin app router):

```
npx create-next-app@latest --typescript
```

- Instalar MeshJS

```
npm install @meshsdk/core @meshsdk/react
```

## Crear proyecto Web3 : Agregar `MeshProvider` `context`

---

## Crear proyecto Web3 : Agregar `MeshProvider` `context`

---

- `<MeshProvider>` provee contexto relacionado con Mesh.

## Crear proyecto Web3 : Agregar `MeshProvider` `context`

---

- `<MeshProvider>` provee contexto relacionado con Mesh.
- Esto permite mantener el estado de nuestra Dapp sincronizado entre páginas.

## Crear proyecto Web3 : Agregar MeshProvider context

---

- <MeshProvider> provee contexto relacionado con Mesh.
- Esto permite mantener el estado de nuestra Dapp sincronizado entre páginas.
- Para utilizarlo, envolvemos el root de nuestro proyecto con el provider en app.tsx:

```
export default function MyApp({ Component, pageProps }: AppProps) {  
  return (  
    <MeshProvider>  
      <Component {...pageProps} />  
    </MeshProvider>  
  );  
}
```

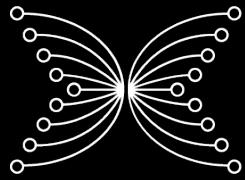
## Crear proyecto Web3 : LISTO!

---

# LISTO!

Ahora podemos usar funciones y componentes siguiendo los docs en:

<https://meshjs.dev/>



INPUT | OUTPUT

# Preguntas?

