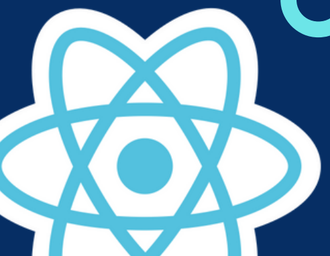
Daniel Paredes Sánchez
Pablo Concepción De Gouveia
Julio 2022

React
Para principiantes
Con esteroides



¿Quiénes somos?

Cloud Solutions Developer

Pablo

pconcepcion@encamina.com

Cloud Solutions Developer

Dani

dparedes@encamina.com

Índice de contenidos

¿Qué es React?

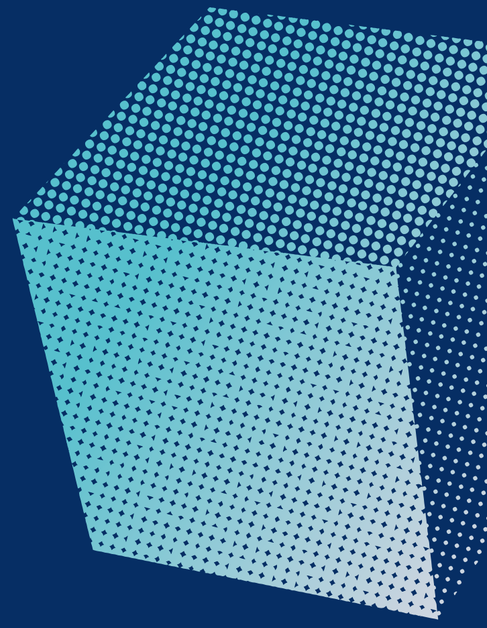
Es una librería *open source* de JavaScript para desarrollar interfaces de usuario. Fue lanzada en el año 2013 y desarrollada por Facebook.

¿Para qué se utiliza?

Permite desarrollar aplicaciones web de una manera más ordenada, óptima, sencilla, cómoda y con menos código que otros framework, o incluso Javascript puro o jQuery que se basan en la manipulación del DOM.

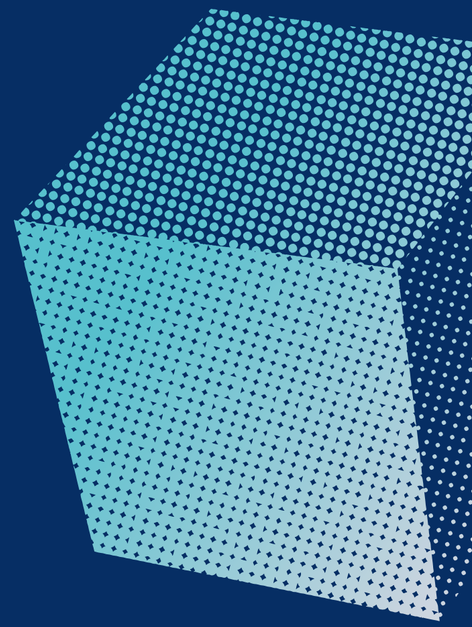
¿Por qué React?

- Virtual DOM
- Amplia comunidad
- Componentización
- Comodidad
- Ecosistema React
- Facilidad de aprendizaje



Virtual DOM

- Representación en memoria del DOM (ReactDOM)
- Intermediario entre el estado de la aplicación y el DOM de la interfaz gráfica.
- Renderiza los componentes en el DOM
  - Cuando se actualiza un componente en el DOM Virtual, React compara con el DOM Real y sólo se renderizan aquellos componentes que cambiaron.

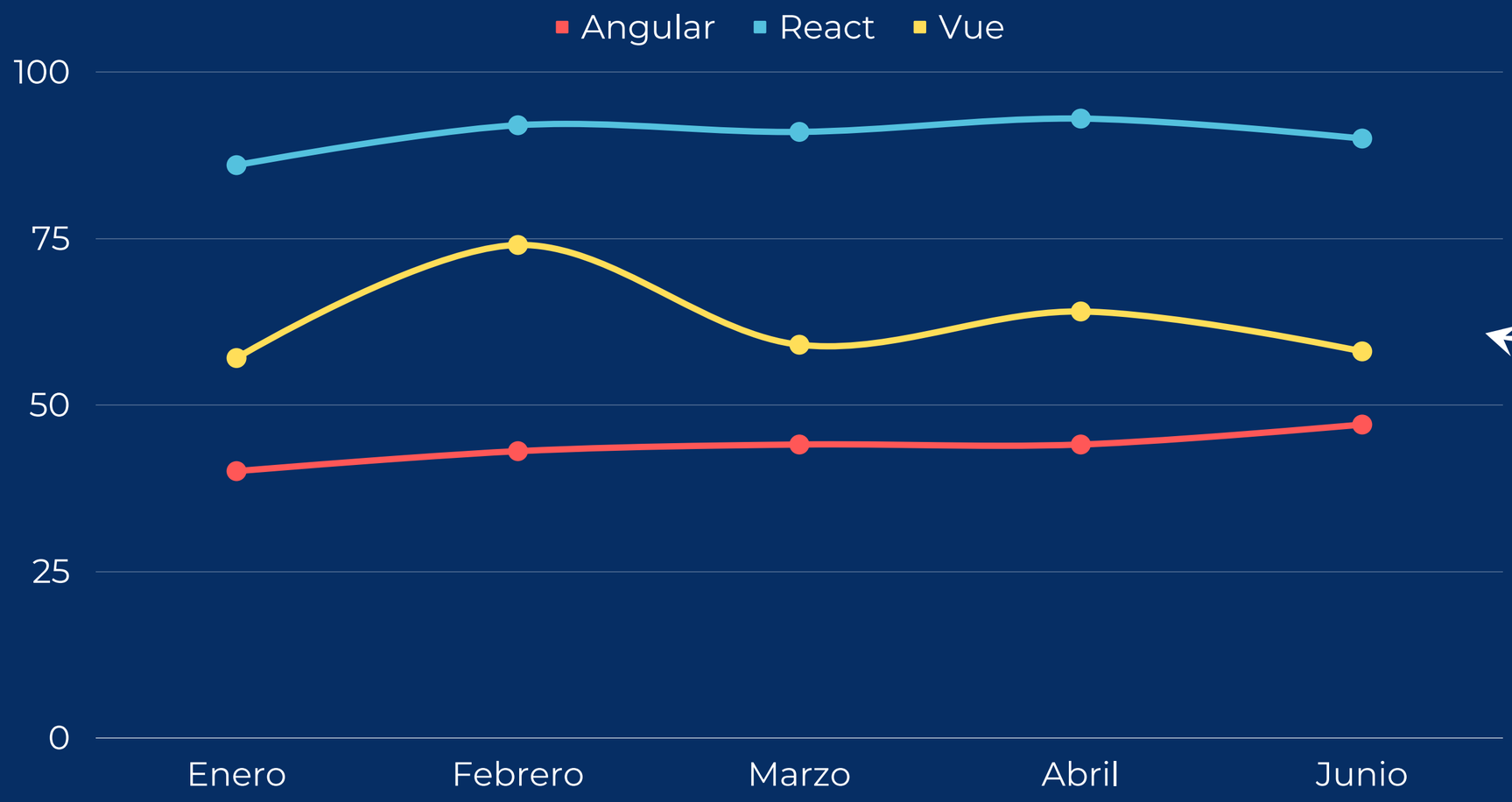


State & Props

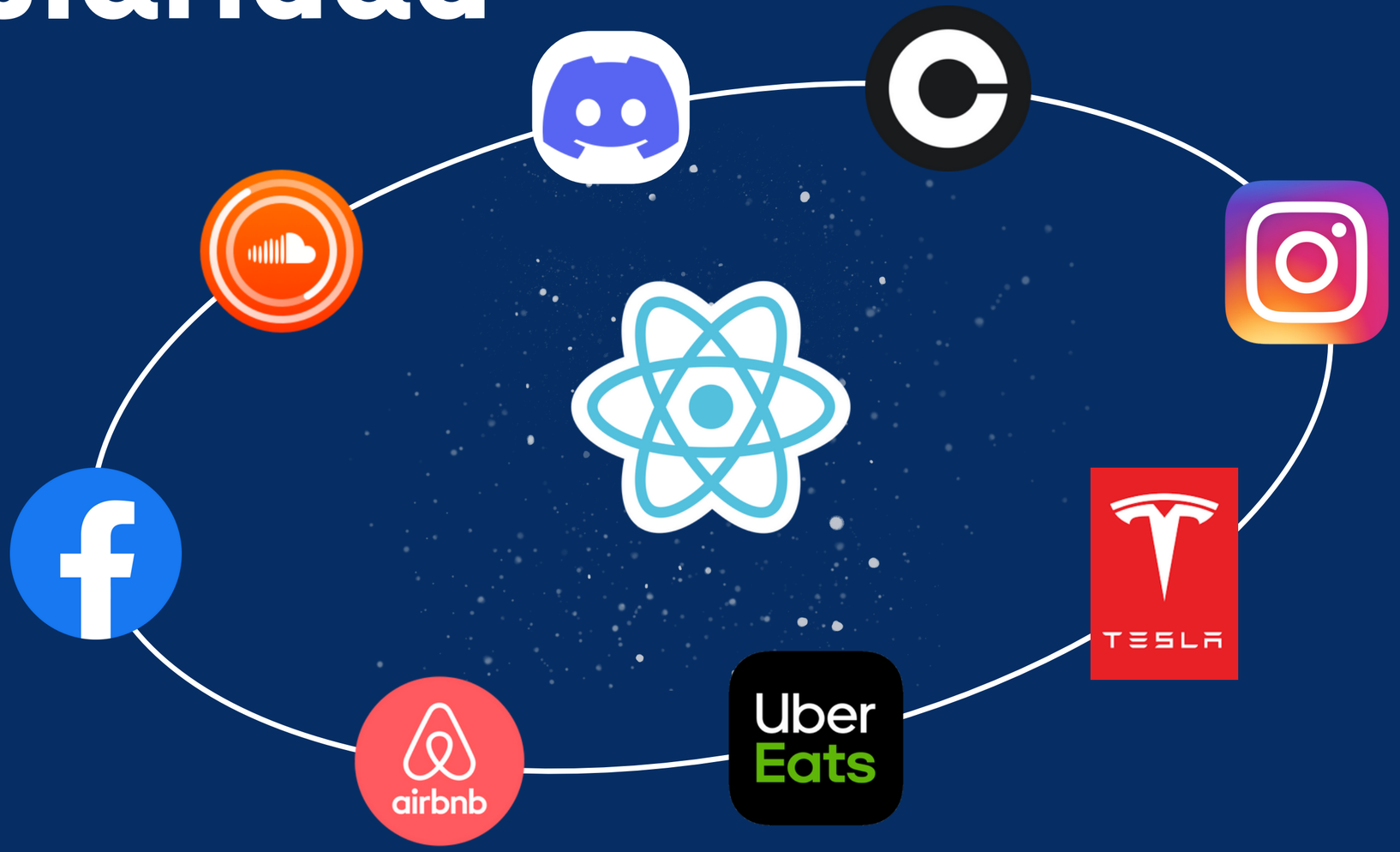
El **estado(state)** son sencillamente los datos que necesitamos para representar la interfaz, los cuales pueden cambiar (mutables).

Las **props** son los datos que vienen de un control padre, los cuales son inmutables.

Popularidad



Popularidad



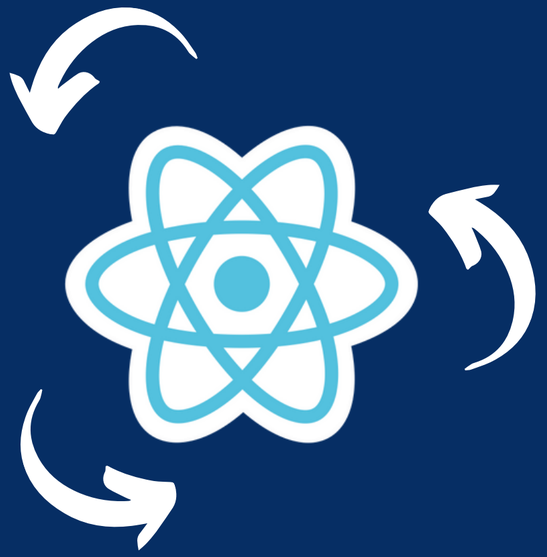
Componentes

Elementos de funcionalidad independientes (fragmentos de código), que se pueden reutilizar, que forman la estructura básica de todas las aplicaciones de React.

Clases

Funciones

Ciclo de vida



Montado

- constructor(props)
- componentWillMount()
- render()
- componentDidMount()

Actualización

Cuando hay cambios en las propiedades o en el estado de un componente.

Demontado

Antes de que un componente se elimine (desmonte) de la UI.

Clases & Funciones

Clases

```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

Funciones

```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

Funciones

- Hooks (useState, useEffect, etc).
- Menor complejidad en la estructura.
- Mejor curva de aprendizaje.

UseState & UseEffect

```
import { getAuth, onAuthStateChanged } from "firebase/auth";

const Home = () => {
  const auth = getAuth();
  const [authorized, setAuth] = useState(false);

  const AuthCheck = onAuthStateChanged(auth, (user) => {
    if (user) {
      setAuth(true);
    } else {
      setAuth(false);
    }
  });

  useEffect(() => {
    return () => AuthCheck();
  }, [auth]);
```

Reutilización

```tsx
import { createUseStyles } from "react-jss";

interface ITitleProps {
  text: string;
}

const Title = (props: ITitleProps) => {
  const classes = styles();

  return <h1 className={classes.title}>{props.text}</h1>;
};

export default Title;

const styles = createUseStyles({
  title: {
    fontFamily: "CuniaFont",
  },
});
```

```tsx
{!authorizated && (
  <>
    <Title text={pageText.LogInTitle} />
    <CenterText text={pageText.LogInDescription} />
  </>
)}
```

¿Por qué TypeScript?

- TypeScript presenta errores en el momento de la disposición mientras que JavaScript, en el tiempo de ejecución.
- TypeScript ayuda a organizar el código, gracias a que está fuertemente tipado.
- TypeScript mantiene las interfaces como modelo de datos.

Comunicación backend

Fetch

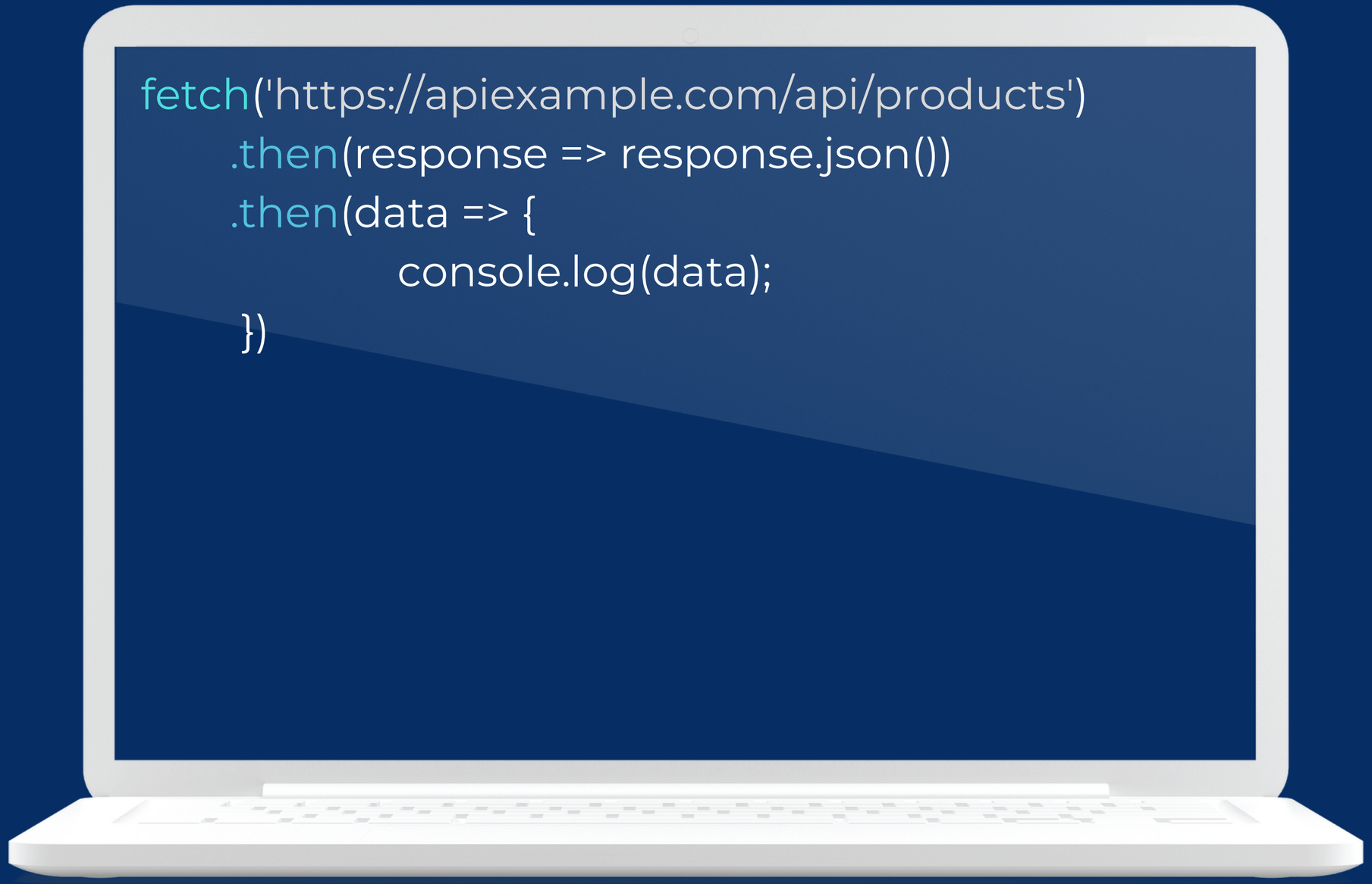
- API JavaScript nativa

Axios

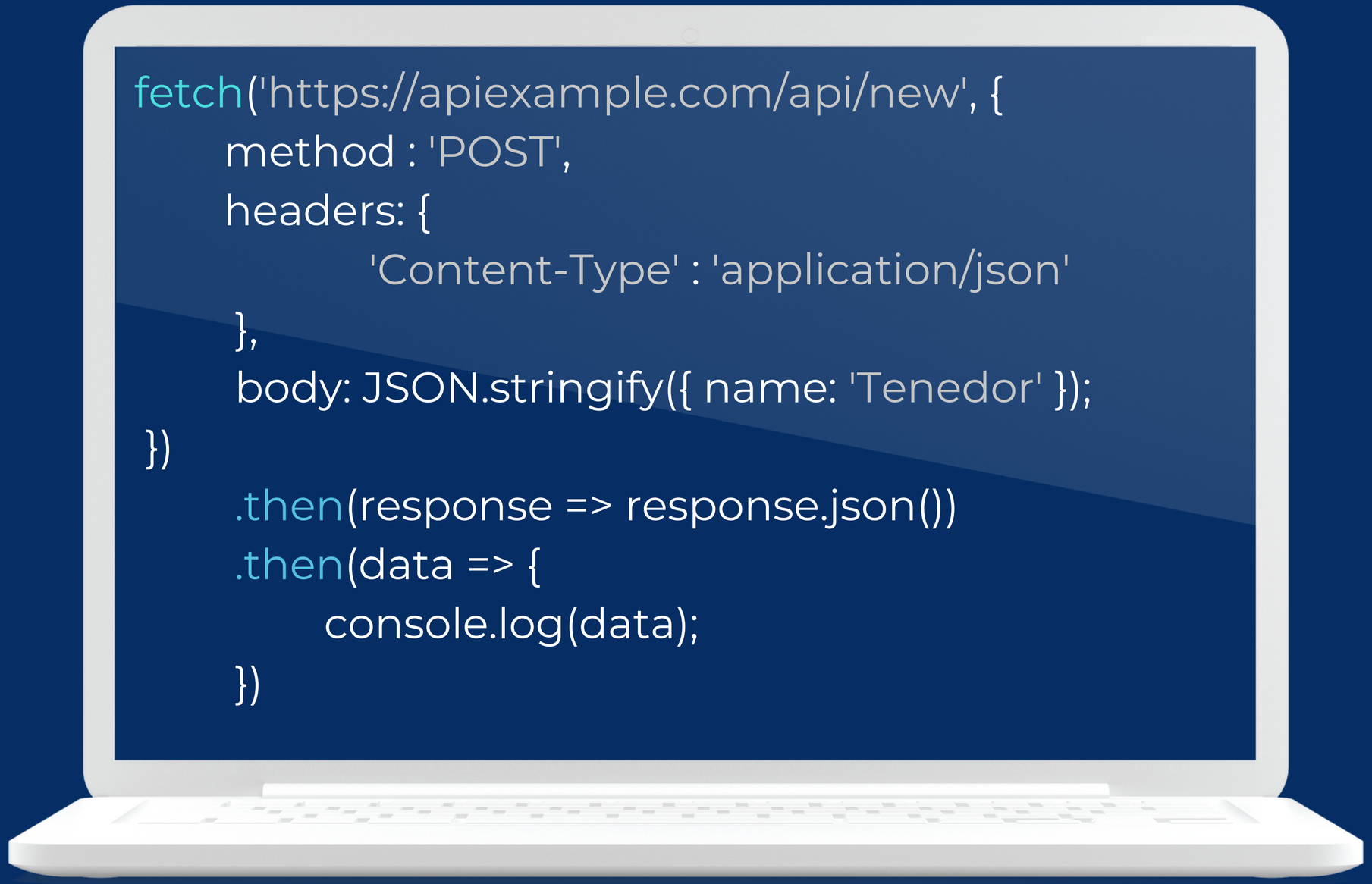
- Librería Javascript

Fetch

Get


fetch('https://apiexample.com/api/products')
.then(response => response.json())
.then(data => {
console.log(data);
})

Fetch
Post


fetch('https://apiexample.com/api/new', {
method : 'POST',
headers: {
'Content-Type' : 'application/json'
},
body: JSON.stringify({ name: 'Tenedor' });
})
.then(response => response.json())
.then(data => {
console.log(data);
})

Axios

Get


axios.get('https://apiexample.com/api/products')
.then(response => console.log(response.data));

Axios
Post


const configuracion = {
method: 'POST',
url: 'https://apiexample.com/api/new',
data: {
title: 'New post a la api',
body: 'Nuevo producto',
userId: 10520
}
}
axios(configuracion)
.then(res => console.log(res.data);

Esteroides

- Front
  - Styles
  - React UI Librarys

- Back

Firebase

Cloudinary

Styles

- Sass
- JSS
- Styled-Components

```tsx
const Title = (props: ITitleProps) => {
  const classes = styles();

  return <h1 className={classes.title}>{props.text}</h1>;
};

export default Title;

const styles = createUseStyles({
  title: {
    fontFamily: "CuniaFont",
  },
});
```

Styles

- Sass
- JSS
- Styled-Components

```
const Title = styled.h1`
  font-size: 1.5em;
  text-align: center;
  color: palevioletred;
`;

const Wrapper = styled.section`
  padding: 4em;
  background: papayawhip;
`;

render(
  <Wrapper>
    <Title>
      Hello World!
    </Title>
  </Wrapper>
);
```

React UI Librarys

- Fluent UI
- Next UI
- Material UI
- Semantic UI
- Blueprint UI

Gracias

Pablo Concepción De Gouveia
Daniel Paredes Sánchez

Julio 2022