

React - Parte 1




React es una biblioteca de JavaScript para construir interfaces de usuario interactivas y reutilizables. Permite crear componentes encapsulados que gestionan su propio estado y se componen para construir interfaces de usuario complejas. Con su modelo de programación basado en componentes y un enfoque en la eficiencia y el rendimiento, React ha ganado popularidad en el desarrollo web moderno.

Instalación React

Existen muchas formas de instalar **REACT** en esta primer parte, lo instalaremos de la forma más básica, y que ha día de hoy no se recomienda usar según la nueva documentación, mira: ['create-react-app ha muerto' de 'midudev'](#)

terminal


 Copiar código

```
npx create-react-app .
```

Instalación Bootstrap

En este proyecto usaremos **Bootstrap** puedes consultar su implementación y recursos [aquí](#)


terminal

 Copiar código

```
npm i bootstrap
```

Para que funcione de manera correcta, en el `src/index.js` deberás importar el siguiente archivo `.css`


src/index.js

 Copiar código

```
bootstrap/dist/css/bootstrap.min.css
```

Quedando configurado de la siguiente manera:

src/index.js

 Copiar código

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

// importamos bootstrap
import 'bootstrap/dist/css/bootstrap.min.css';
```

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <Auth0Provider
      domain="dev-yec4afk3otwxzw26.us.auth0.com"
      clientId="LAKpTticMSq0nZGhv8RLzWwLxXy1XcfR"
      authorizationParams={{
        redirect_uri: window.location.origin
      }}
    >
      <App />
    </Auth0Provider>,
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

NOTA

!Listo de esta manera ya podras usar bootstrap en tu proyecto!


Implementación de Cloudflare

Cloudflare es una herramienta esencial para los propietarios de sitios web que buscan mejorar la seguridad, el rendimiento y la fiabilidad de sus sitios web y aplicaciones en línea

Para su implementación simplemente copia y pega el siguiente código dentro de las etiquetas

<head></head> del archivo `public/index.html`

src/index.js

 Copiar código

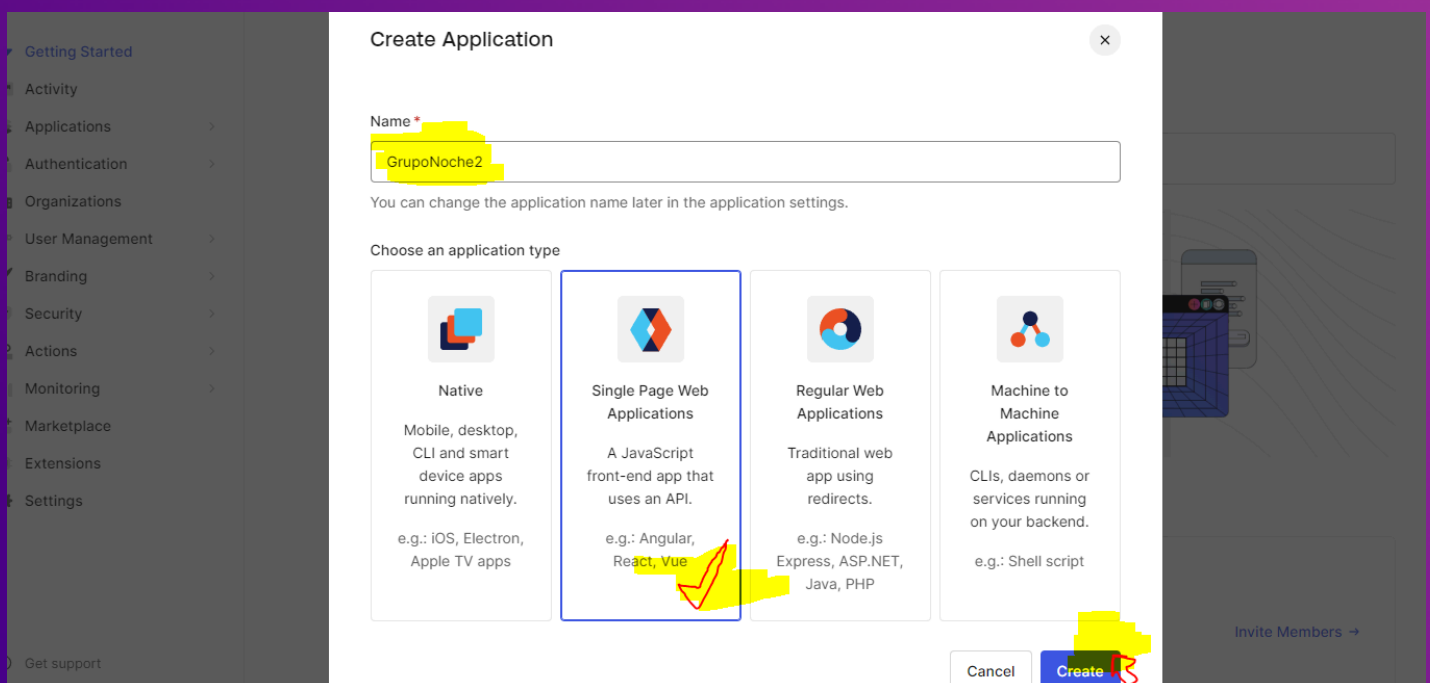
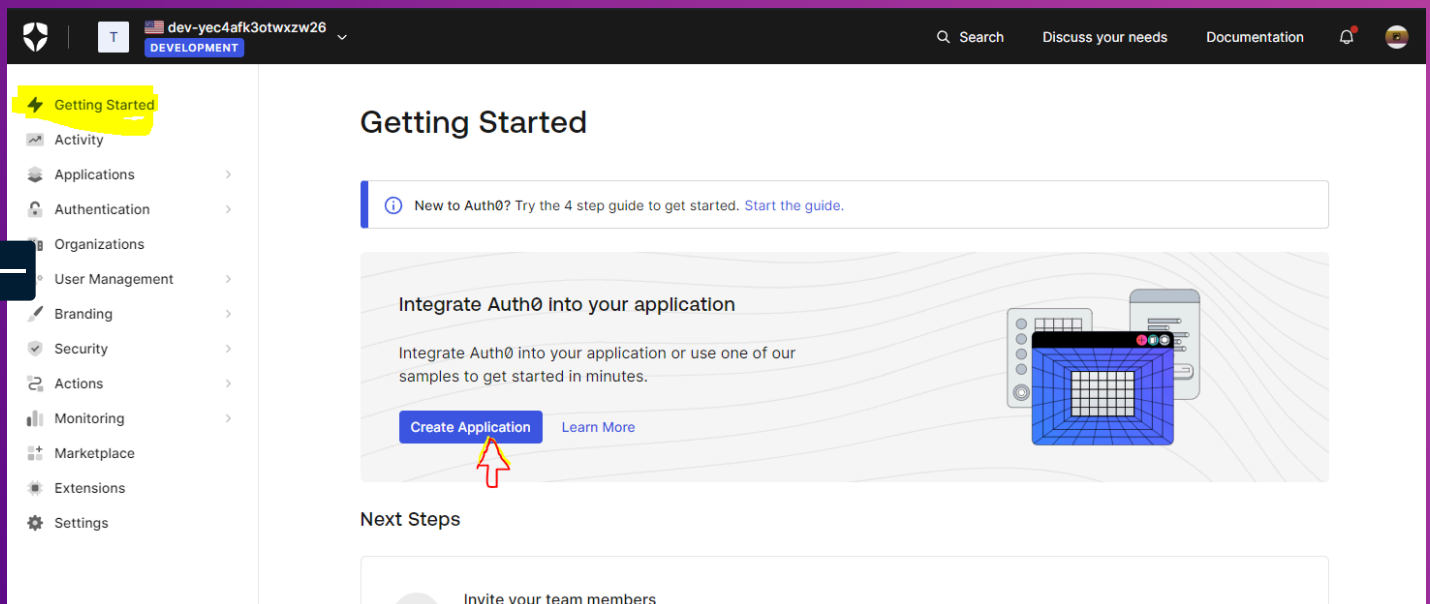
```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.mi
```

Implementación de Auth0

22.04.2024 S30-M2

Auth0 es un servicio que nos provee todo el mecanismo para administrar la autorización y la autenticación de nuestras plataformas. Auth0 pretende darle a los desarrolladores y a las compañías seguridad en sus aplicaciones sin ser unos expertos. La idea es que Auth0 es el experto y simplemente provee todo ese conocimiento.

Ingresa a siguiente [enlace de Auth0](#) y regístrate. Seguidamente dentro del dashboard realiza las siguientes configuraciones





GrupoNoche2

Single Page Application

Client ID qB4Sv4s4k8tgrIFqTg04kTTi0p8WnbIw

22.04.2024 5:35:02

Quickstart

Settings

Addons

Connections

Organizations

Single-Page App

What technology are you using for your web app?

Search by technology name



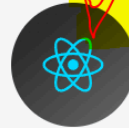
Angular



Flutter (Web)



JavaScript



React



Vue

Configure Logout URLs

A logout URL is a URL in your application that Auth0 can return to after the user has been logged out of the authorization server. This is specified in the `returnTo` query parameter. The logout URL for your app must be added to the **Allowed Logout URLs** field in your [Application Settings](#). If this field is not set, users will be unable to log out from the application and will get an error.

If you are following along with the sample project you downloaded from the top of this page, the logout URL you need to add to the **Allowed Logout URLs** field is `http://localhost:3000`.

Configure Allowed Web Origins

You need to add the URL for your app to the **Allowed Web Origins** field in your [Application Settings](#). If you don't register your application URL here, the application will be unable to silently refresh the authentication tokens and your users will be logged out the next time they visit the application, or refresh the page.

If you are following along with the sample project you downloaded from the top of this page, you should set the **Allowed Web Origins** to `http://localhost:3000`.

Install the Auth0 React SDK

Run the following command within your project directory to install the Auth0 React SDK:

```
npm install @auth0/auth0-react
```

Was this helpful? [Yes](#) / [No](#)

[← Back to Applications](#)



GrupoNoche2

Single Page Application

Client ID qB4Sv4s4k8tgrIFqTg04kTTi0p8WnbIw

Quickstart

Settings

Addons

Connections

Organizations

Pegar !

Allowed Callback URLs

http://localhost:3000

After the user authenticates we will only call back to any of these URLs. You can specify multiple valid URLs by comma-separating them (typically to handle different environments like QA or testing). Make sure to specify the protocol (`https://`) otherwise the callback may fail in some cases. With the exception of custom URI schemes for native clients, all callbacks should use protocol `https://` . You can use [Organization URL](#) parameters in these URLs.

Allowed Logout URLs

http://localhost:3000

Comma-separated list of allowed logout URLs for redirecting users post-logout. You can use wildcards at the subdomain level (`*.google.com`). Query strings and hash information are not taken into account when validating these URLs. [Learn more about logout](#)

Allowed Web Origins

http://localhost:3000

Comma-separated list of allowed origins for use with

Save changes CTRL + Enter

Cancel

Save Changes



GrupoNoche2

Single Page Application

Client ID qB4Sv4s4k8tgrIFqTg04kTT10p0WnbIw

Quickstart

Settings

Addons

Connections

Organizations

Install the Auth0 React SDK

Run the following command within your project directory to install the Auth0 React SDK:

```
1 npm install @auth0/auth0-react
```

Was this helpful? Yes / No

Configure the Auth0Provider component


22.04.2024 5:35 PM 2

Under the hood, the Auth0 React SDK uses [React Context](#) to manage the authentication state of your users. One way to integrate Auth0 with your React app is to wrap your root component with an `Auth0Provider` that you can import from the SDK.

```
1 import React from 'react';
2 import { createRoot } from 'react-dom/client';
3 import { Auth0Provider } from '@auth0/auth0-react';
4 import App from './App';
5
6 const root = createRoot(document.getElementById('root'));
7
8 root.render(
9   <Auth0Provider
10     domain="dev-yec4afk3otwxzw26.us.auth0.com"
11     clientId="qB4Sv4s4kBgTgrIFqTgO4kTTiOp0WnbIw"
12     authorizationParams={{
13       redirect_uri: window.location.origin
14     }}
15   >
16     <App />
17   </Auth0Provider>,
18 );
```

Ahora entra al proyecto y en la terminal ejecuta:


src/index.js

 Copiar código

```
npm install @auth0/auth0-react
```

Seguidamente importo **Auth0** en el archivo `src/index.js`, y agrego el `Auth0Provider` de la siguiente manera:

src/index.js

 Copiar código

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
// importamos bootstrap
import 'bootstrap/dist/css/bootstrap.min.css';
// Importamos auth0
import { Auth0Provider } from '@auth0/auth0-react';
const root = ReactDOM.createRoot(document.getElementById('root'));
```

```
root.render(  
  <React.StrictMode>
```

22.04.2024 S30-M2

```
<Auth0Provider  
  domain="dev-yec4afk3otwxzw26.us.auth0.com"  
  clientId="LAKpTticMSq0nZGhv8RLzWwLxXy1Xcfr"  
  authorizationParams={{  
    redirect_uri: window.location.origin  
  }}  
>  
  <App />  
</Auth0Provider>,
```

!

```
</React.StrictMode>  
);
```

```
// If you want to start measuring performance in your app, pass a function  
// to log results (for example: reportWebVitals(console.log))  
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals  
reportWebVitals();
```

NOTA

Si por alguna razón la autenticación con **Auth0** no funciona, debes comprobar que **clientId** sea el correcto. Para verificar esto ve a el dashboard y verifica:

Getting Started

Activity

Applications

APIs

SSO Integrations

Authentication

Organizations

User Management

Branding

Security

Actions

Monitoring

Marketplace

Extensions

Settings

Get support

GrupoNoche

Single Page Application

Client ID LAKpTticMSq0nZGhv8RLzWwLxXy1Xcfr

Quickstart Settings Addons Connections Organizations

Basic Information

Name *

GrupoNoche

Domain

dev-yec4afk3otwxzw26.us.auth0.com

Client ID

LAKpTticMSq0nZGhv8RLzWwLxXy1Xcfr

Client Secret

.....

The Client Secret is not base64 encoded.

Save changes CTRL + Enter

Cancel Save Changes

Creación de componentes Login/Logout

Debemos crear tres archivos o componentes en la siguiente ruta `src/componentes`

Estructura de carpetas

[Copiar código](#)

```
↓ src
  ↓ componentes// New
    LoginButton.js// New
    LogoutButton.js// New
    Profile.js// New
```

Cada uno con la siguiente configuración:

`src/componentes/LoginButton.js`

[Copiar código](#)

```
import React from "react";
import { useAuth0 } from "@auth0/auth0-react";

const LoginButton = () => {
  const { loginWithRedirect } = useAuth0();

  return <button className="btn btn-success" onClick={() => loginWithRedirect()}>Log In</button>;
};

export default LoginButton;
```

`src/componentes/LogoutButton.js`

[Copiar código](#)

```
import React from "react";
import { useAuth0 } from "@auth0/auth0-react";


const LogoutButton = () => {
  const { logout } = useAuth0();

  return (
    <button className="btn btn-danger" onClick={() => logout({ logoutParams: { returnTo: window.}
      Log Out
    </button>
  );
};
```

```
export default LogoutButton;
```

22.04.2024 S30-M2

src/componentes/Profile.js

 Copiar código

```
import React from "react";
import { useAuth0 } from "@auth0/auth0-react";

const Profile = () => {
  const { user, isAuthenticated, isLoading } = useAuth0();


  if (isLoading) {
    return <div>Loading ...</div>;
  }

  return (
    isAuthenticated && (
      <div>
        <img src={user.picture} alt={user.name} />
        <h2>{user.name}</h2>
        <p>{user.email}</p>
      </div>
    )
  );
};

export default Profile;
```


Antes de mostrar esos componentes apliquemos algunos estilos `css` adicionales:

src/app.css

 Copiar código

```
body{
  background-color: #dbd9e4de;
}
.btn{
  margin: 1em;
}
.ver{
  border: dashed 2px greenyellow;
  background-color: #d6c8c8;
  margin: 1em auto;
  padding: 1em;
  border-radius: 4px;
  width: 30%;
}
.nav-login{
  text-align: right;
```

Ahora para mostrar esos componentes solo debes llamarlos en `src/app.js` así:

`src/app.js` Copiar código


```
import './App.css';
import LoginButton from './Componentes/LoginButton';
import LogoutButton from './Componentes/LogoutButton';
import Profile from './Componentes/Profile';

function App() {
  return (
    <div className="App">
      <nav className='nav-login'>
        <LoginButton/>
        <LogoutButton/>
        <Profile />
      </nav>
    </div>
  );
}

export default App;
```

Inicializando React y comprobando Auth0

Inicializa React

`src/app.js` Copiar código

```
npm run start
```

Comprobado la funcionalidad de Login de **Auth0**

Log In

Log Out



—



Welcome

Log in to dev-yec4afk3otwxzw26 to continue to GrupoNoche.

Email address*

|

Password*



[Forgot password?](#)

Continue

Don't have an account? [Sign up](#)

OR



Continue with Google




 Iniciar sesión con Google

Selecciona una cuenta

para ir a [auth0.com](#)


 **BRUNO**
[redacted]@gmail.com

 [redacted]
[redacted]@gmail.com

 [redacted]
[redacted]@gmail.com

 [redacted]
[redacted]@gmail.com

 [redacted]
[redacted]@gmail.com

 Usar otra cuenta

Para continuar, Google compartirá tu nombre, tu dirección de correo electrónico, tu preferencia de idioma y tu foto de perfil con [auth0.com](#). Antes de usar esta aplicación, puedes leer la [política de privacidad](#) y los [términos del servicio](#) de [auth0.com](#).

Log In

Log Out



[redacted]

[redacted]@gmail.com

**Log In****Log Out**

Mira el repositorio en Github

Componentes

En React, los componentes son bloques de construcción reutilizables para crear interfaces de usuario. Pueden ser clases o funciones que retornan elementos de React. Los componentes encapsulan la lógica y la presentación, siguiendo el principio de responsabilidad única, facilitando así la escalabilidad y mantenimiento de las aplicaciones.


props

En React, los props (abreviatura de "properties") son un mecanismo para pasar datos de un componente padre a un componente hijo. Los props son objetos que contienen datos

inmutables y se utilizan para personalizar o configurar componentes. Permiten que los componentes sean configurables y reutilizables al pasarles información externa.

Ejemplos 1 - Usando **props** en el mismo archivo principal. En `src/app.js` tenemos lo siguiente

src/App.js

 Copiar código

```
import './App.css';

function App() {

  const nombre = "Ingeniero";

  return (
    <div className="App">
      <h2>{nombre}</h2>
      <Welcome name = "Sara" />
      <Welcome name = "Cahal" />
      <Welcome name = "Edite" />
      <h2>{nombre}</h2>
    </div>
  );
}

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

export default App;
```


Como lo ideal es dejar `src/app.js` solo para llamar mis componentes, realizaremos otro ejercicio donde creamos nuestros componentes fuera de nuestro archivo principal

Ejercicio 2 -

Para nuestros componentes hemos creado una carpeta llamada `Componentes` dentro de ella crearemos un nuevo componente llamado `CompoSaludo.js`. De esa manera separaremos nuestro componente dejándolo fuera de nuestro archivo principal que sería `src/app.js`


```
→ src
  → componentes// New
    CompoSaludo.js// New
  App.js
  ...
```

src/Componentes/CompoSaludo.js


 Copiar código

```
import React from "react";

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

export default Welcome;
```

src/App.js

 Copiar código

```
import './App.css';
import Welcome from './Componentes/CompSaludo';

function App() {

  const nombre = "Ingeniero";

  return (
    <div className="App">
      <h2>{nombre}</h2>
      <Welcome name = "Sara" />
      <Welcome name = "Cahal" />
      <Welcome name = "Edite" />
      <h2>{nombre}</h2>
    </div>
  );
}

export default App;
```

Repasa todo sobre `props` [aquí](#)

Bryan Hernández | Telento Tech DWFSV2-42 | 2024

