

# Teoría React I

## ¿Cómo navego en una SPA?

En el desarrollo de aplicaciones web con React, una de las funcionalidades más importantes es la capacidad de navegar entre diferentes vistas o páginas sin necesidad de recargar toda la página (SPA - Single Page Application).

React Router DOM es una librería que permite gestionar la navegación y el enrutamiento en aplicaciones de React de una manera declarativa y sencilla.

La navegación se controla mediante la manipulación del estado de la aplicación y la actualización dinámica de los componentes en función de la ruta (URL).

## ¿Qué es un enrutador de RRD?

El enrutamiento del lado del cliente permite que tu aplicación actualice la URL a partir de un clic en un enlace sin necesidad de hacer otra solicitud de otro documento desde el servidor. En cambio, tu aplicación puede renderizar inmediatamente una nueva interfaz de usuario y hacer solicitudes de datos con fetch para actualizar la página con nueva información.

Esto permite experiencias de usuario más rápidas porque el navegador no necesita solicitar un documento completamente nuevo ni reevaluar los activos de CSS y JavaScript para la próxima página. También permite experiencias de usuario más dinámicas con cosas como animaciones.

En resumen, el enrutador de RRD se encargará de mostrar por cada ruta definida: un componente de interfaz de React.

## ¿Cómo creo un enrutador de RRD?

Utiliza el método `createBrowserRouter` de React Router DOM para crear una instancia del enrutador de RRD y define las rutas de tu aplicación como un array de objetos.

Unset

```tsx

```
const browserRouter = createBrowserRouter([ {}, {}, {} ]);  
...
```

Para definir una ruta, cada objeto del array debe tener como mínimo las propiedades `path` (para definir **qué ruta**) y `element` (para definir **que debe renderizar esa ruta**):

```
Unset  
```tsx  
  
const browserRouter = createBrowserRouter([  
  { path: "/", element: <HomePage /> },  
  { path: "/contact", element: <ContactView /> },  
  { path: "/about-us", element: <AboutInterface /> }  
]);  
...
```

## ¿Cómo navego entre los componentes de vistas?

Utiliza el componente `<Link>` con el atributo `to` para permitir al usuario navegar a otro componente de página haciendo clic o tocándolo. En RRD, `<Link>` renderiza un elemento `<a>` accesible con un verdadero `href` que apunta al recurso al que está enlazando.

```
Unset  
```tsx  
  
<Link to="/">Home</Link>  
  
<Link to="/contact">Contact</Link>  
  
<Link to="/about-us">About</Link>  
...
```

## ¿Cómo definir Rutas Dinámicas?

En muchas aplicaciones, es común que las rutas no sean estáticas, sino que contengan información variable, como identificadores únicos (id), nombres de usuarios o cualquier otro dato relevante para la vista que se desea mostrar. En React Router, podemos crear rutas dinámicas utilizando parámetros en las URL. Los parámetros pueden tener cualquier nombre y se definen con `:`.

```
Unset
```tsx

const browserRouter = createBrowserRouter([

  ...

  { path: "/product/:productId", element: <ProductDetail /> },

  { path: "/profile/:role", element: <Profile /> }

]);

...

```

En el primer ejemplo, la ruta `/product/:productId` tiene el parámetro dinámico `productId` y `/profile/:role` tiene el parámetro `role`. Desde los componentes correspondientes podemos acceder a estos parámetros mediante la función/hook `useParams()` que devuelve un objeto con todos los parámetros de la ruta:

```
Unset
```tsx

const { productId } = useParams()

...

```

Ahora con este parámetro dinámico estamos listos para desarrollar la lógica necesaria para generar vistas dinámicas.

💡 Como sugerencia, resulta beneficioso recurrir a fuentes oficiales de información al enfrentar dudas o al buscar una comprensión más profunda sobre algún tema. A continuación, te proporcionamos enlaces de utilidad para que puedas investigar a tu propio ritmo en el momento que lo consideres

oportuno:  [React](#)  
 [React Router DOM](#)