

# Comprendiendo el enrutado de lado cliente

## Resumen

- Usaremos `<a href>` directamente y `<Link>` para añadir enlaces de `Main.js` a las otras páginas
- Compararemos el comportamiento y extraeremos conclusiones sobre cómo funciona `react-router-dom`
- Añadiremos `<Link>` a la página `About.js` y a `Examples.js`

## Descripción

Normalmente empleamos `<a href>` para enlazar páginas web, ¿verdad?

Cuando un usuario pulsa en un `<a href>`, el navegador *pide* (petición `HTTP`) la nueva página web, la descarga, y la muestra.

Alternativamente, si usamos `<Link>` de `react-router-dom`... ¡la navegación es una *ilusión*! No se envía ninguna petición `HTTP`. Todo está en el navegador (cliente) desde el principio.

¡Veámolo!

## La tarea

Añade `src/screens/main/main.css`:

```
footer {  
  padding: 13px;  
  background-color: rgb(205, 205, 205);  
  color: rgb(90, 90, 220);  
}
```

Luego, en `Main.js`, importa dicho CSS y, a mayores, la clase `Link`:

```
import { Link } from "react-router-dom";  
import "./Main.css"
```

Y añade un `<footer>` dentro del `<div>` principal (después del `<h2>`), así:

```
<footer>  
  <div className='footer-section'>  
    <h3>Enlaces Link</h3>  
    <ul>  
      <li><Link to='/about'>Acerca de</Link></li>  
      <li><Link to='/examples'>Página ejemplos</Link></li>  
    </ul>  
  </div>  
</footer>
```

¿Puedes navegar desde <http://localhost:3000> a tus otras dos páginas, empleado esos enlaces `<Link>`?

¿No te parece una navegación sospechosamente... *rápida*?

## Comparando `<Link to>` con `<a href>`

Añade un nuevo `<div className='footer-section'>` a continuación del anterior, dentro del `<footer>`. Esta vez el título será *Enlaces href* en lugar de *Enlaces Link*:

```
<footer>  
  <div className='footer-section'>  
    <h3>Enlaces Link</h3>  
    <ul>  
      <li><Link to='/about'>Acerca de</Link></li>  
      <li><Link to='/examples'>Página ejemplos</Link></li>  
    </ul>  
  </div>  
</footer>
```

```
+ <div className='footer-section'>
+   <h3>Enlaces href</h3>
+   <ul>
+     <li><a href='/about'>Acerca de</a></li>
+     <li><a href='/examples'>Página ejemplos</a></li>
+   </ul>
+ </div>
</footer>
```

También, para que las dos secciones del `<footer>` aparezcan una *al lado* de la otra, en vez de *arriba y abajo*, **añade** lo siguiente a `Main.css`:

```
footer {

  /* otras propiedades */
  /* ... */

  overflow: auto; /* Esto hace que el contenedor footer no se colapse debido a que sus hijos tienen el atributo float */
}

.footer-section {
  padding-left: 50px;
  float: left;
}
```

**Compara** la navegación implementada en <http://localhost:3000>. ¿Percibes la diferencia?

Para terminar, **añade** en `About.js` y en `Examples.js` dos enlaces **Link debajo** del encabezado `<h2>`. Así:

```
<h4 data-cy='issue7link'>
  <Link to='/'>Regresar a principal</Link>
</h4>
```

¡Enhorabuena! Comienzas a construir un conocimiento significativo sobre el enrutado en React y cómo funciona la programación web lado cliente.

## 🏆 Por último

Recuerda que, en tus webs programadas con React, debes usar `<Link to>`, y no `<a href>`.

Sube tus cambios al repositorio en un nuevo *commit*.



Rubén Montero @ruben.montero changed milestone to %Sprint 3 3 hours ago