

# React.JS

## Introducción

Módulo 03

# Componentes Route

# Route

El componente Route es quizás el componente más importante en React Router para comprender y aprender a usar bien.

Su responsabilidad más básica es representar algún UI cuando una ubicación coincide con la ruta de la ruta.

Hay 3 formas de representar algo con un **<Route>**:

```
<Route component>  
<Route render>  
<Route children>
```



Cada uno es útil en diferentes circunstancias.  
Debería usar solo uno de estos props en un **<Route>** dado.

# Route - Props

Los tres métodos de renderización se aprobarán los mismos tres apoyos de ruta: match, location, history .

```
<Route path="/user/:username" component={User}/>

const User = ({ match }) => {
  return <h1>Hello {match.params.username}!</h1>
}
```

Cuando usa el *component* (en lugar de *render* o *children*), el enrutador usa ***React.createElement*** para crear un nuevo elemento React a partir del componente dado.

Esto significa que si proporciona una función en línea al prop del componente, crearía un nuevo componente en cada procesamiento. Esto da como resultado el desmontaje del componente existente y el montaje del nuevo componente, en lugar de simplemente actualizar el componente existente.

Al usar una función en línea para renderizado en línea, use el *render* o el *prop children*.



```
<Route path="/home" render={() => <div>Home</div>}/>
```

Esto permite un conveniente encapsulamiento y propiedades en línea sin el remontaje no deseado explicado anteriormente.

En lugar de tener un nuevo elemento React creado para usted utilizando el componente *prop*, puede pasar una función para llamar cuando la ubicación coincida. El objeto de render recibe todos los mismos props de ruta que el componente render prop.

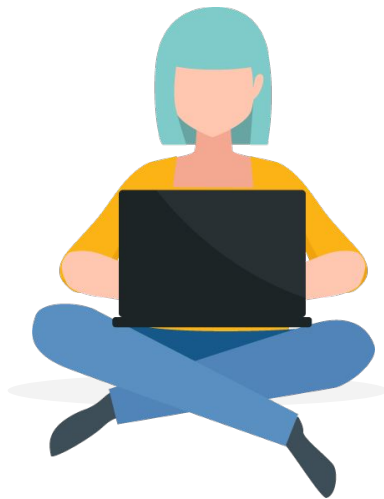
```
const FadingRoute = ({ component: Component, ...rest }) => (  
  <Route {...rest} render={props => (  
    <FadeIn>  
      <Component {...props}/>  
    </FadeIn>  
  )}/>  
)  
  
<FadingRoute path="/cool" component={Something}/>
```



En ocasiones, necesitas representar si la ruta coincide con la ubicación o no. En estos casos, puede usar la función ***children prop***. Funciona exactamente igual que *render*, excepto que se llama si hay coincidencia o no.

Los props de *render children* reciben los mismos elementos de la ruta que el componente y los métodos de representación, excepto cuando una ruta no coincide con la URL y la coincidencia es nula.

Esto le permite ajustar dinámicamente su UI en función de si la ruta coincide o no. Aquí estamos agregando una clase activa si la ruta coincide.



```
export default function Nav(props){  
  const ListItemLink = ({ to, ...rest }) => (  
    <Route path={to} children= {({ match }) => (  
      <li className={match ? 'active' : ''}>  
        <Link to={to} {...rest}></Link>  
      </li>  
    )}  
  )  
  />  
)  
  return (  
    <ul>  
      <ListItemLink to="/rutaA" />  
      <ListItemLink to="/rutaB" />  
    </ul>  
  )  
}
```



# ¡Muchas gracias!

¡Sigamos trabajando!