



# Rutas parametrizadas



**React Router** nos permite guardar valores en las rutas parametrizadas. Esto nos permite renderizar valores dentro de los componentes dependiendo de la información que nos llega por la ruta.



# Índice

1. [Definición](#)
2. [useParams\(\)](#)
3. [Implementación](#)

# 1 | Definición

# Definición

Cuando queremos comenzar a trabajar con rutas parametrizadas, lo primero que tenemos que hacer es definirlas.

Esto lo hacemos desde las rutas que definimos en el componente `<Route/>`.

Las rutas parametrizadas llevan dos puntos (:) y el nombre del dato. Es importante elegir el nombre del dato coherentemente porque es el nombre que luego usaremos para trabajar sobre él.

Nuestra ruta se debería ver algo así:

```
{}  
<Route path="/usuarios/:id" element={<Usuarios />} />
```

## 2 | useParams()

# useParams()

React Router nos provee de un hook (una función) que contiene información sobre la ruta que establecimos en el componente `<Route/>`.

Para usar esta información, podemos almacenarla en una variable sobre la que llamaremos al hook `useParams()`.

Dentro de esta variable, encontraremos un Objeto Literal que contendrá una propiedad (el nombre del parámetro que queremos leer) y cuyo valor es el dato enviado.

Esta propiedad contiene la información que viaja a través de la ruta parametrizada y por eso se torna importante en nuestro componente.

A través de ella podemos buscar los datos del producto que queremos renderizar.

# useParams()

Si queremos acceder al valor que viajó por ruta, deberíamos hacerlo de la siguiente manera dentro de nuestro componente.

Tengamos en cuenta que este fue el nombre que le asignamos a nuestra ruta.

```
{} <Route path="/usuarios/:id" element={<Usuarios />} />
```

```
{} function Componente (){  
    const id = useParams().id  
    return(  
        <div></div>  
    )  
}  
export default Componente
```



# 3 | Implementación

# Implementación

{}

```
function Componente (){  
  const usuarios = [...]  
  const id = useParams().id  
  const usuario = usuarios.find(user => user.id == id)  
  return(  
    <div>  
      <h3>{usuario.nombre}</h3>  
    </div>  
  )  
}  
export default Componente
```

Creamos dos variables:  
una que almacene todos  
los usuarios, y otra que  
almacene el dato que  
llega por la ruta.

# Implementación

{}

```
function Componente (){  
  const usuarios = [...]  
  const id = useParams().id  
  const usuario = usuarios.find(user => user.id == id)  
  return(  
    <div>  
      <h3>{usuario.nombre}</h3>  
    </div>  
  )  
}  
export default Componente
```

Dentro de otra variable almacenamos el resultado del método **.find()**. Este resultado deberá ser el usuario que coincida con el id que llega por la ruta.

# Implementación

{}

```
function Componente (){  
  const usuarios = [...]  
  const id = useParams().id  
  const usuario = usuarios.find(user => user.id == id)  
  return(  
    <div>  
      <h3>{usuario.nombre}</h3>  
    </div>  
  )  
}  
export default Componente
```

Ya podemos utilizar la variable para renderizar los datos únicamente del usuario al cual se accede por la ruta.

# Documentación



Para saber más podemos acceder a la documentación oficial de React Router DOM haciendo clic en el siguiente [link](#).

DigitalHouse>  
Coding School