Open Opened 3 weeks ago by **Rubén Montero**

ChatRoom Offline: Lista de mensajes

Resumen

- Crearemos `ChatRoom.js` que de momento mostrará una lista de mensajes *hardcodeados*
- ¡Se verán en http://localhost:3000/chat_room!
- En el proceso, aprenderemos a usar `map`

Descripción

Un breve repaso:

En JSX sólo podemos escribir instrucciones declarativas. ¡No imperativas! Es decir, *no* podemos emplear bucles (`for` , `while` ...)

```
const EjemploComponente = (props) => {

  // ¡ESTO NO SE PUEDE HACER!
  return <ol>Un bucle de tres números: { for (let i = 0; i<3; i++) { <li>{i}</li> } }</ol>
}
```

Por ello, anteriormente, para trabajar con *listas* en nuestros componentes, hemos *sacado el bucle* "fuera" del JSX:

```
const EjemploComponente = (props) => {

  const lista = []
  for (let i = 0; i<3; i++) {
      lista.push(<li>{i}</li>);
  }
  // ESTO SÍ ES VÁLIDO
  return <ol>{lista}</ol>
}
```

Ha llegado la hora de la verdad: Esta solución no es muy común... Ni muy buena 😧

`map`

Usar `map` sí es algo típico y más cercano a la filosofía de React.

Este método *se invoca sobre una lista* y devuelve *la lista transformada*.

Siguiendo con el ejemplo anterior:

```
const EjemploComponente = (props) => {

  const lista = [1, 2, 3]

  // ¡BUENA SOLUCIÓN, USANDO MAP!
  return <ol>{lista.map(elemento => { return <li>elemento</li> })}</ol>
}
```

Como ves, a `map` sólo hay que *pasarle* (como parámetro) una función: Dicha función será invocada *para cada elemento* de la lista, produciendo así, una lista transformada (*mapeada*).

En el ejemplo, queremos convertir:

- `1` a `<li>1</li>`
- `2` a `<li>2</li>`
- `3` a `<li>3</li>`

Por ello, la *función* que pasamos a `map` es:

```
elemento => { return <li>elemento</li> }
```

¡Adelante! Continuemos con nuestro trabajo para crear una *chatroom offline*.

La tarea

Añade un nuevo componente `src/components/chat_room/ChatRoom.js` .

Dentro de `ChatRoom` , **declara** un estado `messagesList` **así**:

```
const [messagesList, setMessagesList] = useState([
    'Primer mensaje',
    'Vamos a ser los mejores'
]);
```

Luego, **escribe** el siguiente `return` para que dicha lista de *strings* aparezca visualmente como varios `<li>` dentro de un `<ol>` :

```
return <>
    <ul data-cy='chatroom_msg_list'>
        { messagesList.map(string => <li>{string}</li>) }
    </ul>
</>
```

Para terminar, **añade** el componente `<ChatRoom />` a la `ChatRoomPage.js` .

Verifica que funciona.

¿Ves los dos mensajes en http://localhost:3000/chat_room?

Por último

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

Rubén Montero @ruben.montero changed milestone to %Sprint 2 3 weeks ago