Share State

En esta sesión vamos a continuar con la app en la que insertamos un formulario para introducir perfiles, y en esta ocasión vamos a crear un listado con los perfiles que guardemos.

Para ello necesitamos generar un nuevo estado de componente en nuestro componente principal: `App`.

```
import React, {useState} from "react";
import { Form } from "./components";
import "./App.scss";

const App = () => {
  const [profiles, setProfiles] = useState([]);

  const addProfile = (profile) => {
    const newProfiles = [ ...profiles, profile ];
    setProfiles(newProfiles);
  };

  return (
    <div className="app">
      <h1>Listado de perfiles</h1>
      <Form addProfile={addProfile} />
    </div>
  );
```

```
};

export default App;
```

Ahora en nuestro componente, aparte de haber importado debidamente `useState` de React, tenemos un estado que almacenará la lista de perfiles que guardemos. Como podéis ver, también hemos añadido la función `addProfile` que volcará un perfil al estado al invocarla.

Además de esto con spread operator vamos a copiar los nuevos perfiles generados para poder ir sumando todos los que vayamos generando.

Una vez tengamos esto claro, vamos a crear un nuevo componente llamado `ProfileCard`, el cual pintará una carta con los perfiles generados:

```
import React from "react";

const ProfileCard = ({ profiles }) => {
  return (
    <>
      {profiles.map((profile) => {
        return (
          <div key={JSON.stringify(profile)}>
            <h3>
              {profile.name} {profile.address}
            </h3>
            <p>Ciudad: {profile.location}</p>
            <img src={profile.image} alt="" width="200" />
          </div>
        );
      })}
    </>
  );
};
export default ProfileCard;
```

Vamos a desglosar lo que acabamos de hacer:

- Nada más crear nuestro componente le estamos pasando por `prop` todos los perfiles con `({profiles})`(prop que pasaremos al final), pero al querer individualizar cada carta tenemos que hacer un `return` interno que nos haga un bucle de todos los perfiles. Para ello utilizaremos un `{profiles.map((profile)` que nos permita pintar uno a uno dichos perfiles.
- Este mapeo lo hemos hecho en unas etiquetas vacías llamadas `Fragment`, tienen el mismo uso que un `div` común con la diferencia de no tener reflejo alguno en el HTML. Esto nos permite estructurar de mejor forma lo que hacemos. Al igual que todas las etiquetas tienen que abrirse y cerrarse debidamente para su correcto funcionamiento `<>` `</>`.
- En el return interno hemos generado un div que contiene una `key` por cada perfil que se pinte. Una `key` es un atributo especial string que ayuda a React a identificar los ítems que se agregan, cambian o eliminan.
- Y para terminar hemos dividido en diferentes campos los elementos que queremos pintar del formulario, pudiendo utilizar gracias a los pasos anteriores un `profile.(el campo deseado)`.

Ahora tenemos un componente que renderizará los campos del state que hayamos creado a través del formulario, pero para que funcione tenemos que actualizar nuestro componente `Form` para que `SubmitForm` tenga en cuenta los cambios realizados y los añada al state:

```
const submitForm = (ev) => {
    ev.preventDefault();
    const { name, address, location, image } = state;

        if (!name || !address || !location || !image) {
            console.log('Debes rellenar todos los campos');
            return;
        }

        console.log(state);
        props.addProfile(state);
        setState(INITIAL_STATE);
    };
```

Con este condicional nos aseguramos de que los campos que incluyamos sean requeridos y completados antes de hacer el `submit`.

Ya solo nos queda invocar a nuestro nuevo componente `ProfileCard` desde `App` y pasarle los perfiles por prop como hemos mencionado al crear el componente:

```
return (
    <div className="app">
      <h1>Listado de perfiles</h1>
      <Form addProfile={addProfile} />
      <ProfileCard profiles={profiles} />
    </div>
  );
```

Ahora al ejecutar nuestra aplicación y hacer `submit` de nuestro formulario completo nos irá pintando uno a uno los perfiles generados:

Listado de perfiles

Nombre

Apellidos

Ciudad

Imagen

Guardar Perfil

Grabar