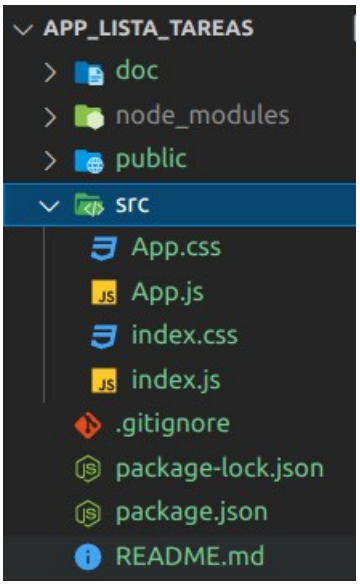
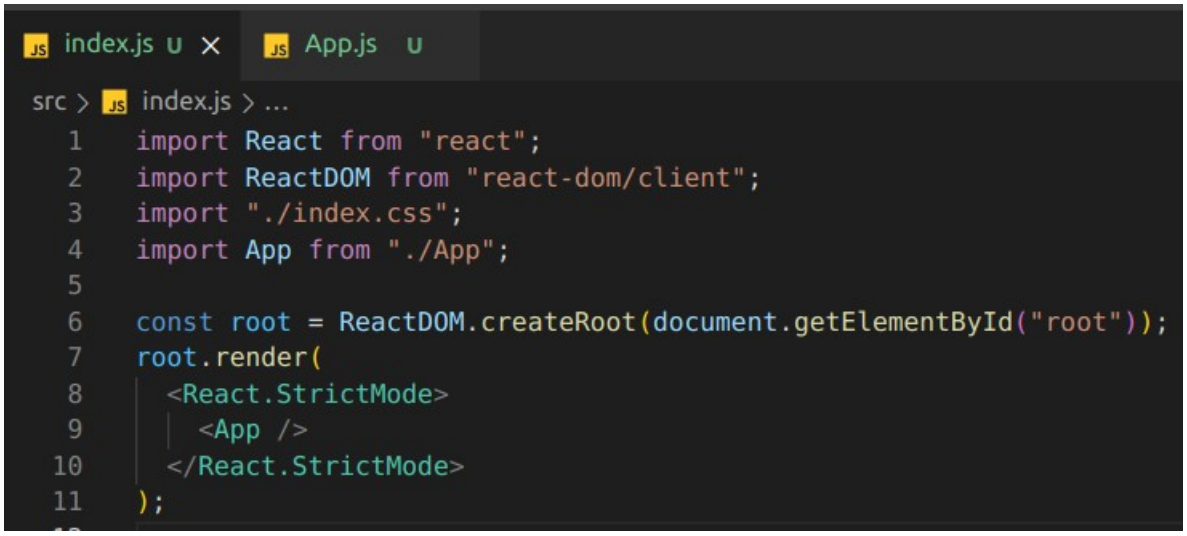
LISTA DE TAREAS

Creando aplicación
$> npx create-react-app app_lista_tareas

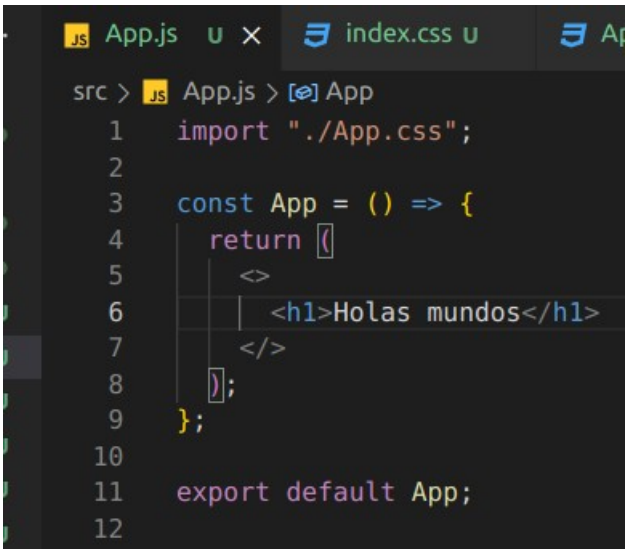
Eliminando archivos que no se necesitan
La carpeta “src” quedaría con los siguientes archivos:



Archivos
– archivo “index.js” quedaría:



– Archivo “App.js”:



Colocando estilos en los archivos “app.css” e “index.css”

Creando el encabezado de la App

Instalar font awesome: https://fontawesome.com/docs/web/use-with/react/
Creando componente “Header.js”:

```
import React from "react";
/* FontAwesomeIcon */
import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
import { faEyeSlash } from "@fortawesome/free-solid-svg-icons";

const Header = () => {
return (
<div>
<header className="header">
<h1 className="header__titulo">Lista De Tareas</h1>
<button className="header__boton">
No mostrar completadas{" "}
<FontAwesomeIcon icon={faEyeSlash} className="header__icono-boton" />
</button>
</header>
</div>
);
};

export default Header;
```

Creando el formulario para agregar tareas

Instalar el paquete “uuid” *npm install uuid*
En “App.js”:

App.js M × FormTask.js U

src > App.js > App

```
import React, { useState } from "react";

import "./App.css";
import FormTask from "./components/FormTask";
import Header from "./components/Header";

const App = () => {
  /* Estados */
  const [tareas, cambiarTareas] = useState([
    {
      id: 1,
      texto: "Lavar ropa",
      completada: false,
    },
    {
      id: 2,
      texto: "Caminar",
      completada: false,
    },
  ]);

  console.log("arreglo", tareas);

  return (
    <div className="contenedor">
      {/* Componente Header */}
      <Header />
      {/* Componente Formulario con propiedades:
      tareas => Enviando el arreglo
      cambiarTareas => Enviando la función para agregar nueva tarea
      */}
      <FormTask tareas={tareas} cambiarTareas={cambiarTareas} />
    </div>
  );
};

export default App;
```

En “FormTask.js”:

```
import React, { useState } from "react";
/* Fontawesome */
import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
import { faPlus } from "@fortawesome/free-solid-svg-icons";
/* uuid */
import { v4 as uuidv4 } from "uuid";

/* Componente con propiedades que vienen de App.js */
const FormTask = ({ tareas, cambiarTareas }) => {
/* Estado para el input */
const [inputTarea, cambiarInputTarea] = useState("");

const handleInput = (event) => {
cambiarInputTarea(event.target.value);
};

const handleSubmit = (event) => {
/* No recargar la página al hacer clic */
event.preventDefault();
/* Agregando la tarea */
cambiarTareas([
...tareas,
{
id: uuidv4(),
texto: inputTarea,
completada: false,
},
]);
};

return (
<form className="formulario-tareas" onSubmit={handleSubmit}>
<input
type="text"
className="formulario-tareas__input"
placeholder="Escribe la tarea"
value={inputTarea}
onChange={(event) => handleInput(event)}
/>
<button type="submit" className="formulario-tareas__btn">
<FontAwesomeIcon
icon={faPlus}
className="formulario-tareas__icono-btn"
/>
</button>
</form>
);
};

export default FormTask;
```

Creando la lista de tareas

Componente para lista de tareas (TasksList.js), enviando las tareas al componente de tarea (Task.js):

App.js M | TasksList.js U | Task.js U

src > components > TasksList.js > default

```
1   import React from "react";
2   import Task from "./Task";
3
4   const TasksList = ({ tareas }) => {
5     return (
6       <ul>
7         {tareas.length > 0 ? (
8           tareas.map((tarea) => <Task key={tarea.id} tarea={tarea} />)
9         ) : (
10          <div className="lista-tareas__mensaje">~ No existen tareas ~</div>
11        )}
12      </ul>
13    );
14  };
15
16  export default TasksList;
17
```

Llamando este componente desde App.js enviándole la lista de tareas:

App.js M | TasksList.js U | Task.js U

src > App.js > App

```
8   const App = () => {
9     /* Estado para las tareas */
10    const [tareas, cambiarTareas] = useState([
11      {
12        id: 1,
13        texto: "Lavar ropa",
14        completada: false,
15      },
16      {
17        id: 2,
18        texto: "Caminar",
19        completada: false,
20      },
21    ]);
22
23    return (
24      <div className="contenedor">
25        {/* Componente Header */}
26        <Header />
27        {/* Componente Formulario con propiedades:
28        tareas => Enviando el arreglo
29        cambiarTareas => Enviando la función para agregar nueva tarea
30         */}
31        <FormTask tareas={tareas} cambiarTareas={cambiarTareas} />
32        {/* Componente Lista de tareas */}
33        <TasksList tareas={tareas} />
34      </div>
35    );
36  };
37
```

Componente para la tarea (Task.js) donde recibe una tarea:

App.js M | TasksList.js U | Task.js U

src > components > Task.js > Task

```
1  import React from "react";
2  /* Fontawesome */
3  import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
4  import { faCheck, faEdit, faTimes } from "@fortawesome/free-solid-svg-icons";
5
6  const Task = ({ tarea }) => {
7    return (
8      <li className="lista-tareas__tarea">
9        <FontAwesomeIcon
10         icon={faCheck}
11         className="lista-tareas__icono lista-tareas__icono-check"
12       />
13       <div className="lista-tareas__texto">{tarea.texto}</div>
14       <div className="lista-tareas__contenedor-botones">
15         <FontAwesomeIcon icon={faEdit} className="lista-tareas__icono-accion" />
16         <FontAwesomeIcon
17           icon={faTimes}
18           className="lista-tareas__icono-accion"
19         />
20       </div>
21     </li>
22   );
23 };
24
25 export default Task;
```

Formulario para editar las tareas

En el componente “Task.js”:

```
import React, { useState } from "react";
/* Fontawesome */
import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
import { faCheck, faEdit, faTimes } from "@fortawesome/free-solid-svg-icons";

const Task = ({ tarea }) => {
/* Estado para el saber si se va a editar */
const [editTask, editTaskHandler] = useState(false);
/* Estado para el input */
const [newTask, newTaskHandler] = useState(tarea.texto);
/* Manejar el submit */
const submitHandler = (e) => {
e.preventDefault(); // No actualizar la página
editTaskHandler(false);
};
return (
<li className="lista-tareas__tarea">
<FontAwesomeIcon
icon={faCheck}
className="lista-tareas__icono lista-tareas__icono-check"
/>
<div className="lista-tareas__texto">
{editTask ? (
<form className="formulario-editar-tarea" onSubmit={submitHandler}>
<input
type="text"
className="formulario-editar-tarea__input"
value={newTask}
onChange={(e) => newTaskHandler(e.target.value)}
```

```
/>
<button type="submit" className="formulario-editar-tarea__btn">
Actualizar
</button>
</form>
) : (
tarea.texto
)}
</div>
<div className="lista-tareas__contenedor-botones">
<FontAwesomeIcon
icon={faEdit}
className="lista-tareas__icono-accion"
onClick={() => editTaskHandler(!editTask)}
/>
<FontAwesomeIcon
icon={faTimes}
className="lista-tareas__icono-accion"
/>
</div>
</li>
);
};

export default Task;
```

Función para completar las tareas

Enviar la función para cambiar tarea al componente “TasksList.js”:

App.js M | TasksList.js U | FormTask.js | Task.js U

src > App.js > App

```
19          completada: true,
20        },
21      ]);
22
23      return (
24        <div className="contenedor">
25          {/* Componente Header */}
26          <Header />
27          {/* Componente Formulario con propiedades:
28          tareas => Enviando el arreglo
29          cambiarTareas => Enviando la función para agregar nueva tarea
30           */}
31          <FormTask tareas={tareas} cambiarTareas={cambiarTareas} />
32          {/* Componente Lista de tareas */}
33          <TasksList tareas={tareas} cambiarTareas={cambiarTareas} />
34        </div>
35      );
36    };
37
38    export default App;
39
```

El componente padre “TasksList.js” enviará una función (procesada por este) al hijo “Task.js” para que este envíe el argumento “id” para que sea procesada por el componente padre.

```
import React from "react";
import Task from "./Task";

/* Componete que recibe el arreglo de tareas y la función para actualizar tarea */
const TasksList = ({ tareas, cambiarTareas }) => {
/* Colocar las funciones en el componente padre */
const completedToggle = (id) => {
cambiarTareas(
tareas.map((tarea) => {
if (tarea.id === id) {
return {
...tarea,
completada: !tarea.completada,
};
}
return tarea;
})
);
};

return (
<ul>
{tareas.length > 0 ? (
tareas.map((tarea) => (
<Task
key={tarea.id}
tarea={tarea}
completedToggle={completedToggle}
/>
))
) : (
<div className="lista-tareas__mensaje">~ No existen tareas ~</div>
)}
</ul>
);
};

export default TasksList;
```

El componente hijo “Task.js” recibe la función del padre y le envía el “id” para que este lo procese.

```
import React, { useState } from "react";
/* Fontawesome */
import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
import {
faCheckSquare,
faEdit,
faTimes,
faSquare,
} from "@fortawesome/free-solid-svg-icons";

/* Componente que recibe la tarea y la función para cambiar de estado de tarea a completada */
const Task = ({ tarea, completedToggle }) => {
/* Estado para el saber si se va a editar */
const [editTask, editTaskHandler] = useState(false);
/* Estado para el input */
const [newTask, newTaskHandler] = useState(tarea.texto);
/* Manejar el submit */
```

```
const submitHandler = (e) => {
e.preventDefault(); // No actualizar la página
editTaskHandler(false);
};
return (
<li className="lista-tareas__tarea">
<FontAwesomeIcon
icon={tarea.completada ? faCheckSquare : faSquare}
className="lista-tareas__icono lista-tareas__icono-check"
onClick={() => completedToggle(tarea.id)}
/>
<div className="lista-tareas__texto">
{editTask ? (
<form className="formulario-editar-tarea" onSubmit={submitHandler}>
<input
type="text"
className="formulario-editar-tarea__input"
value={newTask}
onChange={(e) => newTaskHandler(e.target.value)}
/>
<button type="submit" className="formulario-editar-tarea__btn">
Actualizar
</button>
</form>
) : (
tarea.texto
)}
</div>
<div className="lista-tareas__contenedor-botones">
<FontAwesomeIcon
icon={faEdit}
className="lista-tareas__icono-accion"
onClick={() => editTaskHandler(!editTask)}
/>
<FontAwesomeIcon
icon={faTimes}
className="lista-tareas__icono-accion"
/>
</div>
</li>
);
};

export default Task;
```

Funciones para editar y borrar tareas

De la misma forma anterior, crear la función que procesará la eliminación en el componente padre y enviar los datos (id a eliminar) desde el componente hijo.

```
import React from "react";
import Task from "./Task";

/* Componete que recibe el arreglo de tareas y la función para actualizar tarea */
const TasksList = ({ tareas, cambiarTareas }) => {
/* Colocar las funciones en el componente padre */
const completedToggle = (id) => {
cambiarTareas(
tareas.map((tarea) => {
if (tarea.id === id) {
return {
...tarea,
completada: !tarea.completada,
};
}
```

```
return tarea;
})
);
};

/* Función para actualizar el texto de la tarea */
const updateTask = (id, texto) => {
cambiarTareas(
tareas.map((tarea) => {
if (tarea.id === id) {
return {
...tarea,
texto: texto,
};
}
return tarea;
})
);
};

/* Función para eliminar el texto de la tarea */
const deleteTaskHandler = (id) => {
cambiarTareas(
tareas.filter((tarea) => {
if (tarea.id !== id) {
return tarea;
}
return;
})
);
};

return (
<ul>
{tareas.length > 0 ? (
tareas.map((tarea) => (
<Task
key={tarea.id}
tarea={tarea}
completedToggle={completedToggle}
updateTask={updateTask}
deleteTaskHandler={deleteTaskHandler}
/>
))
) : (
<div className="lista-tareas__mensaje">~ No existen tareas ~</div>
)}
</ul>
);
};

export default TasksList;
```

En el componente hijo (Task.js):

```
import React, { useState } from "react";
/* Fontawesome */
import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
import {
faCheckSquare,
faEdit,
faTimes,
faSquare,
} from "@fortawesome/free-solid-svg-icons";

/* Componente que recibe la tarea y la función para cambiar de estado de tarea a
completada */
const Task = ({ tarea, completedToggle, updateTask, deleteTaskHandler }) => {
/* Estado para el saber si se va a editar */
const [editTask, editTaskHandler] = useState(false);
/* Estado para el input */
const [newTask, newTaskHandler] = useState(tarea.texto);
/* Manejar el submit */
const submitHandler = (e) => {
e.preventDefault(); // No actualizar la página
updateTask(tarea.id, newTask); // Actualiar tarea (proceso en componente "Tasks.js")
editTaskHandler(false);
};
return (
<li className="lista-tareas__tarea">
<FontAwesomeIcon
icon={tarea.completada ? faCheckSquare : faSquare}
className="lista-tareas__icono lista-tareas__icono-check"
onClick={() => completedToggle(tarea.id)}
/>
<div className="lista-tareas__texto">
{editTask ? (
<form className="formulario-editar-tarea" onSubmit={submitHandler}>
<input
type="text"
className="formulario-editar-tarea__input"
value={newTask}
onChange={(e) => newTaskHandler(e.target.value)}
/>
<button type="submit" className="formulario-editar-tarea__btn">
Actualizar
</button>
</form>
) : (
tarea.texto
)}
</div>
<div className="lista-tareas__contenedor-botones">
<FontAwesomeIcon
icon={faEdit}
className="lista-tareas__icono-accion"
onClick={() => editTaskHandler(!editTask)}
/>
<FontAwesomeIcon
icon={faTimes}
className="lista-tareas__icono-accion"
onClick={() => deleteTaskHandler(tarea.id)}
/>
</div>
</li>
);
};
```

```
export default Task;
```

Lógica para mostrar y ocultar las tareas completadas

Crear el estado en App.js para mostrar u ocultar las tareas completadas. Además, enviar el dato al componente TasksList.js para mostrar u ocultar las tareas.

En App.js:

```
import React, { useState } from "react";

import "./App.css";
import FormTask from "./components/FormTask";
import Header from "./components/Header";
import TasksList from "./components/TasksList";

const App = () => {
/* Estado para las tareas */
const [tareas, cambiarTareas] = useState([
{
id: 1,
texto: "Lavar ropa",
completada: false,
},
{
id: 2,
texto: "Caminar",
completada: true,
},
]);
/* Estado para mostrar tareas completadas */
const [showTaskCompleted, changeTaskCompleted] = useState(false);

return (
<div className="contenedor">
{/* Componente Header */}
<Header
showTaskCompleted={showTaskCompleted}
changeTaskCompleted={changeTaskCompleted}
/>
{/* Componente Formulario con propiedades:
tareas => Enviando el arreglo
cambiarTareas => Enviando la función para agregar nueva tarea
*/}
<FormTask tareas={tareas} cambiarTareas={cambiarTareas} />
{/* Componente Lista de tareas */}
<TasksList
tareas={tareas}
cambiarTareas={cambiarTareas}
showTaskCompleted={showTaskCompleted}
/>
</div>
);
};

export default App;
```

En TasksList.js:

```
import React from "react";
import Task from "./Task";

/* Componete que recibe el arreglo de tareas y la función para actualizar tarea */
const TasksList = ({ tareas, cambiarTareas, showTaskCompleted }) => {
/* Colocar las funciones en el componente padre */
const completedToggle = (id) => {
cambiarTareas(
tareas.map((tarea) => {
if (tarea.id === id) {
return {
...tarea,
completada: !tarea.completada,
};
}
return tarea;
})
);
};

/* Función para actualizar el texto de la tarea */
const updateTask = (id, texto) => {
cambiarTareas(
tareas.map((tarea) => {
if (tarea.id === id) {
return {
...tarea,
texto: texto,
};
}
return tarea;
})
);
};

/* Función para eliminar el texto de la tarea */
const deleteTaskHandler = (id) => {
cambiarTareas(
tareas.filter((tarea) => {
if (tarea.id !== id) {
return tarea;
}
return;
})
);
};

return (
<ul>
{tareas.length > 0 ? (
tareas.map((tarea) => {
if (showTaskCompleted) {
return (
<Task
key={tarea.id}
tarea={tarea}
completedToggle={completedToggle}
updateTask={updateTask}
deleteTaskHandler={deleteTaskHandler}
/>
);
} else if (!tarea.completada) {
```

```
return (
<Task
key={tarea.id}
tarea={tarea}
completedToggle={completedToggle}
updateTask={updateTask}
deleteTaskHandler={deleteTaskHandler}
/>
);
}
return;
})
) : (
<div className="lista-tareas__mensaje">~ No existen tareas ~</div>
)}
</ul>
);
};

export default TasksList;
```

Agregando localstorage a nuetra app