```
// TaskContext.js
import React, { createContext, useReducer, useContext } from 'react';

// Acciones disponibles para el reducer
const ACTIONS = {
  ADD_TASK: 'ADD_TASK',
  DELETE_TASK: 'DELETE_TASK',
  FILTER_TASKS: 'FILTER_TASKS',
};

// Función reductora para gestionar el estado de las tareas
const taskReducer = (state, action) => {
  switch (action.type) {
    case ACTIONS.ADD_TASK:
      return [...state, { id: Date.now(), text: action.payload }];

    case ACTIONS.DELETE_TASK:
      return state.filter((task) => task.id !== action.payload);

    case ACTIONS.FILTER_TASKS:
      return state.filter((task) =>
        task.text.toLowerCase().includes(action.payload.toLowerCase())
      );

    default:
      return state;
  }
};
```

```
// Crear el contexto y proveedor para el estado de las tareas
const TaskContext = createContext();

export const useTaskContext = () => useContext(TaskContext);

export const TaskProvider = ({ children }) => {
  const [tasks, dispatch] = useReducer(taskReducer, []);

  return (
    <TaskContext.Provider value={{ tasks, dispatch }}>
      {children}
    </TaskContext.Provider>
  );
};
```