**1.- Reducer:** Creamos el reducer con 2 objetos que van a contener:

* El estado inicial de cada variable general (InitialState)
* Nombre de las acciones (actionTypes)

A continuación se crea cada uno de los cases con switch. En cada case en lugar de poner comillas y el nombre del case, se pone actionTypes. Seguido del nombre que tiene cada parámetro en el objeto actionTypes.

**2.- Stateprovider:** Se crea un contexto llamado StateContext.

Creamos el componente StateProvider que va a tener 3 parámetros desestructurados (reducer, initialState y children), reducer e initialState los recibirá de forma exportada en index.js mientras que children se refiere a lo que contenga dentro (en este caso es App.js en index.js).

Por otro lado se asigna un parámetro “value” que va a contener el hook (la función) useReducer. Esto permite usar las herramientas de useReducer en otros componentes sin necesidad de importarlo.

Se crea UseStateValue que contiene una función con el hook useContext, esto permite que los componentes puedan usar este hook sin necesidad de importarlo ni declararlo, solo importando useStateValue (ver App.js línea 16)

El contexto del componente StateProvider será StateContext (el contexto que hemos creado previamente)

**3.- index.js:** Se envuelve App con StateProvider y a este se le asigna dos parámetros (initialState y reducer) los valores de estos parámetros se obtienen importando del componente reducer. Esto permite usar useReducer completamente ya que se obtienen los parámetros necesarios para usarlo en cualquier componente.

**4.- App.js y resto de componentes:** Se obtienen los parámetros de initialState (reducer.js) los cuales son isopen, user y darkmode (de forma desestructurada) y con la función dispatch podemos cambiar el valor de estos según queramos.