

# Aplicación Redux-Hospital

1. Instalar una aplicación de react
2. Limpiar archivos
3. Instalar dependencias

react-redux: npm i react-redux

react-uuid: npm i react-uuid

redux: npm i redux

Bootstrap: npm i bootstrap

1. Importar Bootstrap

import 'bootstrap/dist/css/bootstrap.css'

import 'bootstrap/dist/js/bootstrap.bundle.min'

1. Crear carpeta de trabajo

Imagen que contiene Texto

Descripción generada automáticamente

1. Crear los componentes AgregarCita y ListarCita

Crear un formulario en Agregar citas

<https://github.com/Garcia091/Info_Redux/blob/main/AgregarCitas>

Crear lista en ListarCitas

<https://github.com/Garcia091/Info_Redux/blob/main/ListarCita>

Crear la interfaz de app

<https://github.com/Garcia091/Info_Redux/blob/main/App>

Adicionar los estilos en el contenedor

<https://github.com/Garcia091/Info_Redux/blob/main/style>

1. Crear en la carpeta container o pages un componente AgendaApp y renderizar los componentes AgregarCita y ListarCita.



## **Redux**

1. Crear los types (acciones) que se van a realizar en la aplicación

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Crear los reducer de citas CitasReducer.js

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

Reducer es una función exportable, no debemos usar nada de react

Inicializar el estado

Crear una función que se pueda exportar por defecto

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Importar los type desde el reducerCita y usarlos:

1. Importar el archivo types
2. Utilizar las acciones al interior de switch

Como queremo mantener la informacion del estado vamos a usar el spred para guardar los nuevos cambio en el state

Agregar la siguiente estructura para la correcta funcionalidad del reducer

Texto

Descripción generada automáticamente

Existen dos formas de combinar el reducer:

1. Por medio de un archivo externo
2. Al interior del store

**Por medio de un archivo externo:**

Crear un archivo index en reducer que nos permite combinarlos.

1. Importar combineReducers de redux

import {combineReducers} from 'redux';

1. Importar CitasReducer
2. Crear una función export combineReducers

Texto

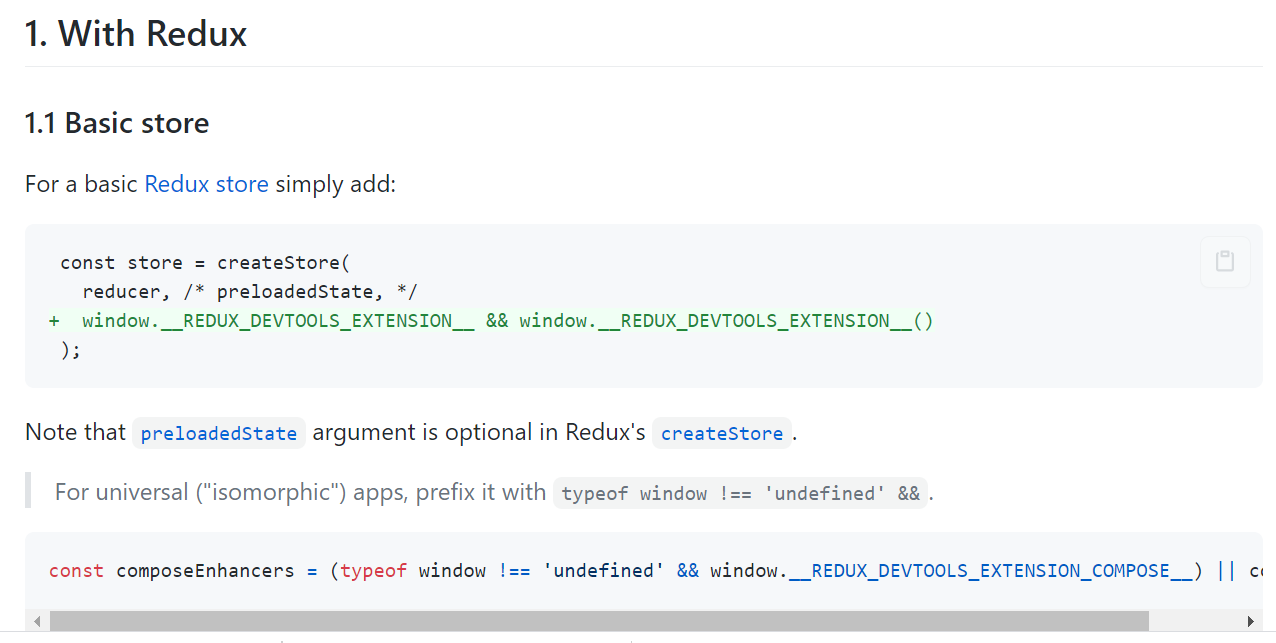
Descripción generada automáticamente

**Configurar el archivo store:**

1. Crear archivo store.js en la carpeta store
2. Ira las extensiones de redux devtools

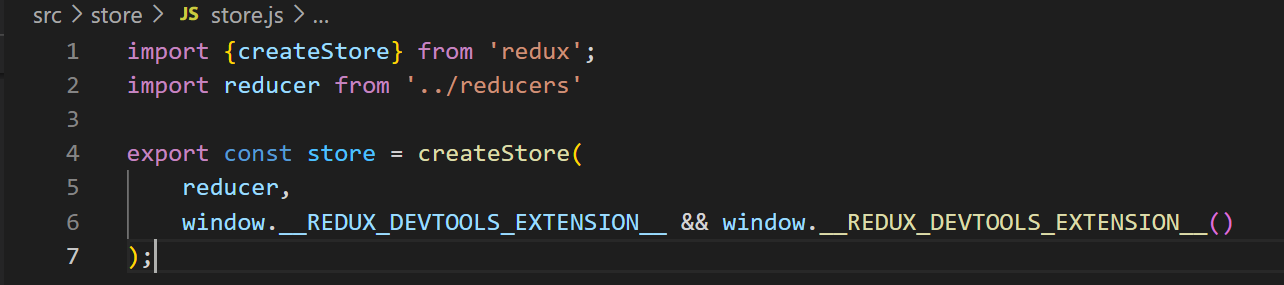
<https://github.com/zalmoxisus/redux-devtools-extension>

1. Importar de redux el componente createStore
2. De las extensiones de redux traer el siguiente código



1. necesitamos la información de lo reducers combinados

archivo externo



Al interior del archivo store

Texto

Descripción generada automáticamente

En la carpeta action crear un archivo ActionCitas.js

Agregar las acciones de la aplicación Crear, Borrar utilizando los types

Texto

Descripción generada automáticamente

En AgendaApp de la carpeta container agregar el componente Provider de redux

Importar el archivo Store, el provider necesita un parámetro llamado getState, por tal motivo le enviamos la configuración de store.

Texto

Descripción generada automáticamente

En las devtools del navegador ya entregamos la información del store

Captura de pantalla de un celular

Descripción generada automáticamente

Agregar la funcionalidad de los formularios para tener citas agregadas.

Crear en la carpeta hook el useForm

<https://github.com/Garcia091/Info_Redux/blob/main/useForm>

Importar useForm

Enviar los campos a useForm

Desestructurar los campos del formulario

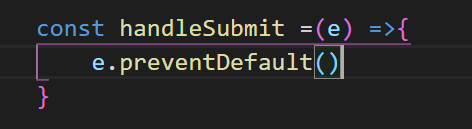
Pantalla de computadora con letras

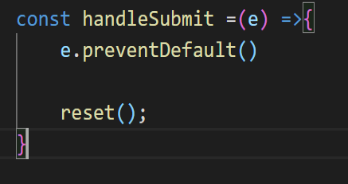
Descripción generada automáticamente con confianza media

Crear el evento handleSubmit

 <form onSubmit={handleSubmit}>

Prevenir la actualización del formulario



Resetear el formulario

En los inputs agregar el atributo value y el evento onChange

Texto

Descripción generada automáticamente

- Hasta el momento no se ha visto mayor cosa de los elementos de clase

- De redux hemos creado Action, store, reducer, type

- Vamos a hacer que esto funcione

En el archivo de agregar citas vamos a invocar los elementos de Redux, importar uuid para agregar el id de las citas.

Texto

Descripción generada automáticamente

Usar el Dispatch para actualizar él state

Llamar al archivo actionCitas.js e indicarle la función a usar

Crear una función para guardar al dispatch en el componente de AgregarCitas

En handleSubmit crear la información que se desea enviar a el archivo actionCitas por medio del dispatch

Texto

Descripción generada automáticamente

El código hasta el momento se ve así

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Actualizar los reducer

citas: [...state.citas, action.payload]

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Hasta el momento funciona correctamente.

Vamos a imprimir la información en el componente de ListarCitas.js

1. necesitamos importar a useSelector de la librería de react-redux

import {useSelector} from 'react-redux'

1. crear una constante para guardar la información que está en el store usando el useSelector
2. Texto

   Descripción generada automáticamente

Vamos a mandar la información a lista citas para que se refleje en la aplicación

Importa useSelector de react-redux

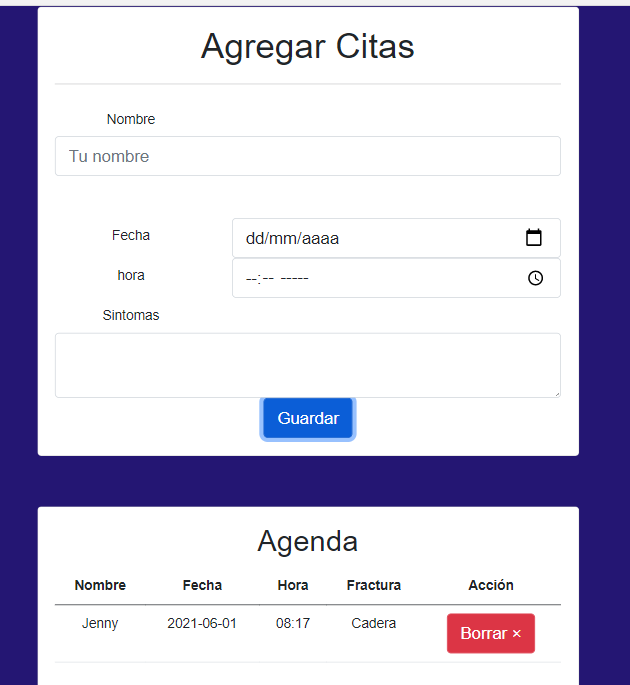


crear citas.map para recorrer el vector

Texto

Descripción generada automáticamente

La función ejecuta de esta manera.



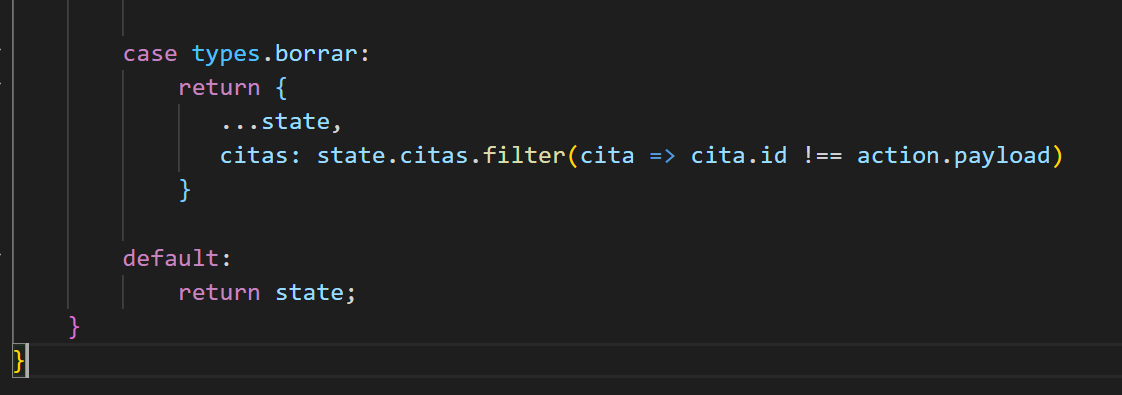
Como podemos borrar esa Cita:

1. Crear la acción en el archivo types

2. llamarlo desde las acciones y mandar por parámetros el id

3. ir a los reducer y crear el estado.

Como vamos a realizar un filtro de las citas para borrar, debemos incluir un filter y que nos bórrela la validación diferente a la que buscamos.



Para crear la función borrar debemos importar useDispatch desde el componente

Texto

Descripción generada automáticamente

En el botón eliminar mandamos el evento payload para borrar el elemento.

Primero debemos importar las acciones del componente de action

Con la información llamar al botón borrar

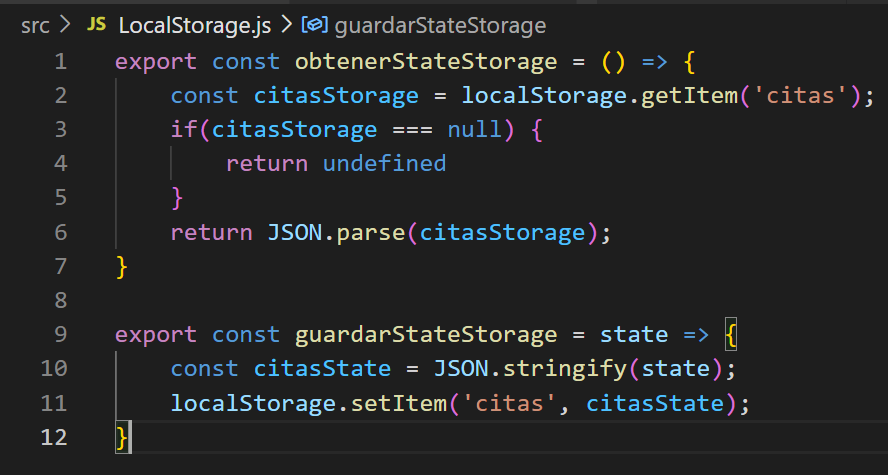
Texto

Descripción generada automáticamente con confianza media

**Leer en Local Store:**

Crear un archivo para la localstorage.js en la raíz de src

Crear las funciones básicas



En store importar las funciones de LocalStorege

import {obtenerStateStorage, guardarStateStorage} from '../LocalStorage'

en el archivo storage

