# React redux Duck

## Video 1 y 2

Para hacer la configuración inicial de redux tenemos que crear el store, la cual es nuestra tienda para que los estados estén disponibles en nuestra aplicación. El store une todas los reducers o los archivos ducks para que podamos consumir los state en la aplicación.

También se debe contar con un reducer.

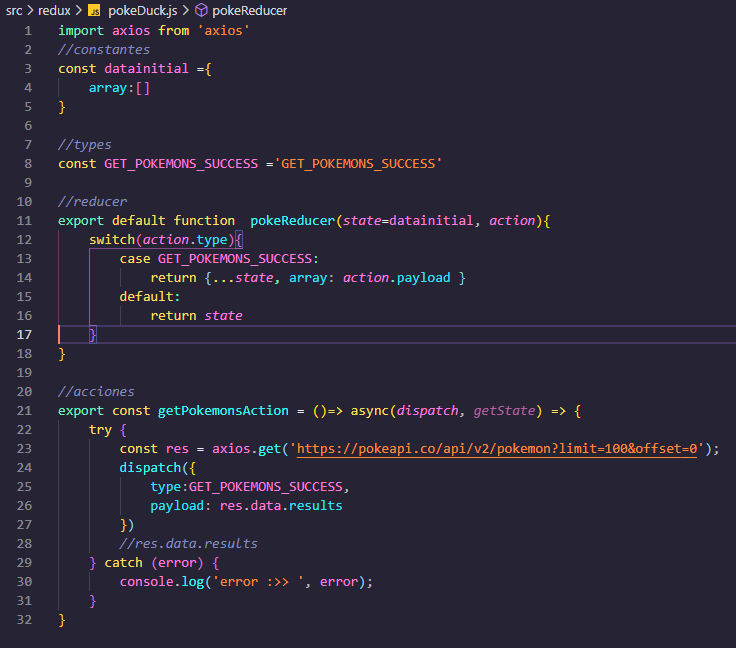
Cuando trabajamos con redux, existen 2 formas de trabajar.

1. Separar todas las acciones, los types, las constantes, los reducer en archivos individuales.
2. Ocupar la técnica del patito, unir todo eso y hacer un solo archivo, el cual es la metodología **duck.**

Con la metodología duck podemos manejar el state todo en un mismo archivo. **No confundamos** el hecho de que podemos tener un archivo duck por cada uno de los procedimientos, por ejemplo, un prodimientosDuck, loginDuck, carroComprasDuck, etc.

No importa con que metodología trabajemos redux, siempre va a requerir 3 puntos:

1. Constantes (las constantes las mandamos a llamar y podemos ocupar la data en todos los componentes)
2. Reducer (con esto aceptamos la lista de la data que mandamos a traer y lo enviamos a la constante)
3. Actions (por ejemplo, va a consumir el api y mandar a llamar la data)



## Video 3

La configuración para la tienda siempre va a ser igual por lo tanto es una configuración general para la mayoría o todos los store.js

Siempre en el store vamos a llamar a todos los reducers que tengamos en el proyecto, así como también debemos llamar a todos los archivos de ducks que tengamos, debemos importarlo en la store.

Una vez que tenemos todos los reducers vamos a combinar en el store todos los reducer

Para ello usamos combineReducers importado de redux así:

const rootReducer = combineReducers ({

    pokemons:pokeReducer

})

Adentro del objeto solamente vamos a ir anexando demás reduxer que creemos, en el lado izquierdo tenemos como se va a ver en la store (desde el browser) y del lado derecho a que reducer representa.

También se agrega la configuración de la extensión de para visualizar el store en el navegador



Así como también la generación de la tienda y asignación de los middlewares

export default function generateStore () {

    const store = createStore (rootReducer, composeEnhancers(applyMiddleware(thunk)));

    return store;

}

## Video 4

### UseDispatch y useSelector

Para que todos los componentes que agreguemos puedan leer la tienda y los archivos ducks etc, tenemos que envolver lo que está en la app en un provider.

El provider recibe una prop llamada store allí tenemos que mandar el return de la función que genera la tienda y combina todos los reducers en una función.

El **useDispatch** nos sirve para enviar una acción la cual realizará lo que tenemos estipulado en el archivo duck.

Para usarlo importamos useDispatch luego lo asignamos y enviamos el nombre de la acción que tenemos en el archivo duck, previamente importada también.

Para usar el useSelector solamente inicializamos, eso nos retorna el store y luego accedemos a la propiedad que queremos acceder, por ejemplo:

const pokemones= useSelector(*store*=>*store*.pokemons.array);