

StateShape

El documento en gith está dividido en estos diferentes caracteres según su nombre de carpeta.

Action :

El action cuenta con:

1. addToCart el cual significa que es el boton de poder añadir un elemento más a la bolsa.
2. removeCart el cual determina quitar un elemento de la bolsa.

estas actions se ven afectadas por el contador, el cual determina que tanto a gastado la persona. el Add fue añadido dentro del menú y el remove es añadido dentro de la bolsa.

código:

```
export const ADD = 'Agregar';  
export const REMOVE = 'Remover';
```

```
export function addToCart(item) {  
  return {  
    type : ADD,  
    item  
  }  
}
```

```
export function removeFromCart(item) {  
  return {  
    type : REMOVE,  
    item  
  }  
}
```

Reducers:

Es importante mencionar que dentro del reducer se trabajaron los actions antes mencionados. Se definió el inventario inicial, en un arreglo , en donde se definieron todas las comidas implementadas y las que llegarán a ser añadidas al ingresar el input. Se definio el total el cual tiene un valor 0 que sin embargo es influenciado por el add y remove antes mencionado en donde se le suma o se le resta uno respectivamente. Sin embargo debido

al hecho que el programa debe manejar cantidades, se le tienen que sumar más veces a pesar de que ya a sido presionado una vez y esto también afecta en el caso del reducir. Por último, se implementa un update para finalizar la suma y por el hecho de que este se debiera mostrar dentro del componente de la bolsa.

```
import {ADD, REMOVE} from "../action/index";
```

```
let initialState = {  
  inventory: [  
    {fooditem: "Sushi Juanito", price: 72.00},  
    {fooditem: "Mongolian Beef", price: 67.00},  
    {fooditem: "Ramen", price: 46.00},  
    {fooditem: "Pie de Queso", price: 20.00},  
  
  ],  
  cart: [],  
  total: [{totalvalue:0}]  
};
```

```
const todos = (state = initialState, action) => {  
  switch (action.type) {  
    case ADD :  
      let alreadyInCart = false;  
      if(state.cart.length>0) {  
        state.cart = state.cart.map((cartitem) => {  
          console.log(cartitem.fooditem+"\n"+action.item.fooditem);  
          if (cartitem.fooditem===action.item.fooditem) {  
            cartitem.qty+=1;  
            alreadyInCart=true;  
          }  
          return cartitem;  
        });  
      }  
    }  
  }  
};
```

```
    console.log(state.cart);  
  }  
}
```

```
    if(!alreadyInCart)  
    {  
      state.cart.push({  
        fooditem: action.item.fooditem,
```

```

        price: action.item.price,
        qty: 1
    });
}
updateTotal(state);

return {
    ...state,
    [action.cart] : state.cart
};

```

State Shape

El estado se desarrolla principalmente en el ManagerView el cual es el que ingresa el input y trabaja con el input text fuera de esto se estuvo trabajando estado pero más en props en componentes.

Componentes: (smart, dummy, props)

Entre los componentes se dividió el trabajo en componentes según la bolsa , el menú y el manager view. Se trabajó principalmente con props en el cual se trabajó el estado en arreglo y el dispatch, es decir se trabajó el remove qué fue antes mencionado en el cual se realizó un dispatch y esto también funciona de la manera inversa.