

## Laboratorio 7

```
import * as types from '../types';

export const addToInventory = (
  id,
  name,
  price,
  qty,
  comentario,
) => ({
  type: types.INVENTORY_ADDED,
  payload: {
    id,
    name,
    price,
    qty,
    comentario,
  }
});

export const addToCart = id => ({
  type: types.PRODUCT_ADDED_TO_CART,
  payload: {
    id,
  }
});
```

El action fue un trabajo clave para el funcionamiento de todo. Esto es un action y el action son cargas de información que envían datos desde su aplicación a su tienda. los resultados obtenidos en el action fueron transportados de las siguientes maneras:

```

<Fragment>
  <input
    type="text"
    placeholder="@usuario"
    ref={ node => { this.nameInput = node; } }
  />
  <br></br>
  <input
    type="text"
    placeholder="Titulo"
    ref={ node => { this.priceInput = node; } }
  />
  <br></br>
  <input
    type="text"
    placeholder="Ingrese el contenido"
    ref={ node => { this.qtyInput = node; } }
  />
  <br></br>
  <button
    onClick={
      () => {
        onSubmit(
          this.nameInput.value,
          this.priceInput.value,
          this.qtyInput.value,
        );

        this.nameInput.value = "";
        this.priceInput.value = "";
        this.qtyInput.value = "";
        this.nameInput.focus();
      }
    }
  />

```

```

const InventoryItem = ({
  name,
  price,
  qty,
  comentario,
  onClick,
  onSubmit,
}) => (
  <div>
    <h2>
      {name}
    </h2>
    <h4>
      {price}
    </h4>
    <br></br>
    <Counter />
  </div>
)

```

Por otro lado también se trabajó un counter que realizará un contador de los likes tanto de las publicaciones como los comentarios. Es importante mencionar que se trabajó con bastante dispatch por medio de id. el uuid está comentado debido a que la compu windows 7 no la llegó aceptar. Se utilizó dispatch con el objetivo de trabajar bien los actions debido a que las acciones son cargas de información que envían datos desde su aplicación a su tienda. Ellos son la única fuente de información para la tienda. Los envía a la tienda usando el dispatch.

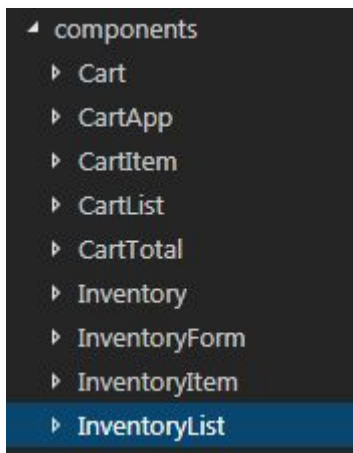
```

export default connect(
  (state, { id }) => ({
    ...state.inventory.byId[id],
  }),
  (dispatch, { id }) => ({
    onClick() {
      dispatch(actions.addToCart(id));
    },
    onSubmit(comentario) {
      //se comento uuid porque aunque si esta implementado no lo puedo ver porque ando con una compu vieja que no deja
      //dispatch(actions.addToInventory(uuid(), name, price, qty));
      dispatch(actions.addToInventory( comentario));
    }
  })
)

```

## Componentes:

Los componentes son pequeños bloques de código de nuestra aplicación en los cuales podemos encapsular elementos HTML, CSS y Javascript para un fin en específico, en este caso se usó JSX para facilitar la tarea, además, un componente puede contener uno o varios componentes más y estos pueden ser reutilizados cada vez que sea necesario. Entre los componentes utilizados son los que se encuentran en la carpeta de componentes



como principalmente:

1. Los input y botones con las funciones respectivas ya sea de los comentarios como el counter.
2. el ordenamiento de ingresar nuevos comentarios y poder comentarlos.

## Dummy

Según lo que se vio en clase los componentes Dummy también se denominan componentes 'presentacionales' porque su única responsabilidad es presentar algo al DOM. Una vez hecho esto, el componente está hecho con él. No hay que vigilarlo, no hay que registrarse de vez en cuando para ver cómo van las cosas. Los componentes en sí mismos sólo tienen un método render () (no necesitan ningún otro) y a menudo son solo funciones de Javascript. No tienen un estado interno para administrar. No sabrían cómo cambiar la información que presentan si se les pidiera. La ignorancia es felicidad. Se trabajó en base a lo que vimos en clase lo siguiente:

```
class InventoryFormDummy extends React.Component {
  render() {
    const { onSubmit } = this.props;

    return (
      <Fragment>
        <input
          type="text"
          placeholder="@usuario"
          ref={ node => { this.nameInput = node; } }
        />
        <br></br>
        <input
          type="text"
          placeholder="Titulo"
          ref={ node => { this.priceInput = node; } }
        />
        <br></br>
        <input
          type="text"
          placeholder="Ingrese el contenido"
          ref={ node => { this.qtyInput = node; } }
        />
        <br></br>
        <button
          onClick={
            () => {
              onSubmit(
                this.nameInput.value,
                this.priceInput.value,
                this.qtyInput.value,
              );
            }
          }
        />
      </Fragment>
    );
  }
}
```

Resultados:

## Reddit Blog:

### La Bella y la Bestia

Los componentes son pequeños bloques de código de nuestra aplicación en los cuales podemos encapsular elementos HTML, CSS y Javascript para un fin en especifico, en nuestro caso construiremos estos componentes con JSX para facilitar la tarea, además, un componente puede contener uno o varios componentes más y estos pueden ser reutilizados cada ves que sea necesario.

2

Like | Don't Like

Ingresar Comentario

Ingresar nuevo Comentario

### La casa del Lago

Cabe mencionar que el desarrollo en Rect esta basado en componentes (es el corazón de React) , pongamos como ejemplo un chat, nuestro componente principal sera todos los elementos de un chat y dentro de este podemos tener varios componentes, un componente para escribir el mensaje, otro componente para colocar los mensajes enviados, los botones de acción pueden ser otro componente y de esta forma reutilizarlo para tener mas de un botón.

0

Like | Don't Like

Los componentes son pequeños bloques de código de nuestra aplicación en los cuales podemos encapsular elementos HTML, CSS y Javascript para un fin en especifico, en este caso se uso JSX para facilitar la tarea, además, un componente puede contener uno o varios componentes más y estos pueden ser reutilizados cada vez que sea necesario. Entre los componentes utilizados son los que se encuentran en la carpeta de componentes

Los componentes son pequeños bloques de código de nuestra aplicación en los cuales podemos encapsular