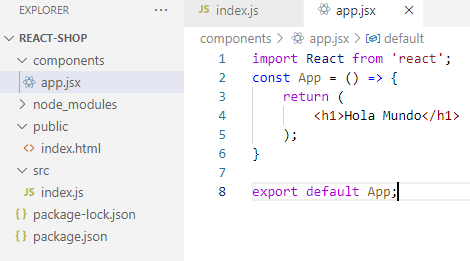
**Conceptos Básicos**

* jsx Plantillas (javascript con xml)
* virtual DOM (solo se refrescan ciertos elementos)
* ciclo de vida
* Estado
* Eventos (clicks, mover mouse)
* Hooks (engloba los componentes)

**INICIALIZAR PROYECTO**

1. npm init
   1. entry point src/index.js
   2. Licence MIT
2. npm install react react-dom
3. Crear estructura



1. **Instala Babbel, webpack Plugin y loaders**

* npm install @babel/core @babel/preset-env @babel/preset-react
* npm install webpack webpack-cli webpack-dev-server //commandos consola, desarrollo local
* npm install babel-loader html-loader html-webpack-plugin //

1. Crear Archivo .babelrc (se agrega la configuracion de babbel) para que reconozca los presets
2. Crear archivo webpack.config.js
3. Package.json agregar scripts (start and build)
4. Escribir public/index.html
5. Clase 6 (Agregando estilos sass).
6. npm install mini-css-extract-plugin css-loader style-loader sass -D, agregamos las reglas y los pluginds en nuestro archive de webpack config
   1. npm install sass-loader -D

Clase 7 y 8 Transformando HTML y CSS a react

1. Copiar los elementos html de body de index.html a Login.jsx
2. Crear el archivo Login.css con los estilos y agregar las variables a global.css y \_vars.css
3. Se crean los contenedores Layout y Login.jsx
4. Se agregan los contenedores a el componente de App.jsx

Clase 9 React Router DOM

1. Instalar npm install react-router-dom
2. Mover el archive App.jsx a la carpeta de Routes y se cambia la programación por la sintaxis de react router-dom
3. Creamos las carpetas para home.jsx y Nofound.jsx, en webpack se agrega devserver y publicPath: '/',

## Clase 11 Header (componentes)

## Clase 15 tipos de componentes

Este es un ejemplo para useState, podemos darle un valor inicial, el cual puede ser cambiado por un evento que se puede asignar a este mismo componente, o a otros componentes y hasta pasarlo por medio de un hijo para que éste cambie el inicial.

Para poder usar componentes stateful es necesario llamar useState desde React, la forma de importarlo y usarlo es la siguiente:

 const Button = () => {

    const [name, setName] = useState('Hola Mundo');

    return (

        <div>

            <h1>{name}</h1>

        </div>

    );

 }

Los componentes stateless servirán para pasar un estilo visual o props, pero no tendrá otra función más que esa.

import React from 'react';

const Button2 = () => {

        <div>

            <h1>name</h1>

        </div>

 };

## Clase 16 Imágenes

1. Se copian los archivos a la carpeta src/assets y se agregan al webpack.config

            {

                test: /\.(png|svg|jpg|gif)$/,

                type: 'asset'

            }

1. Agregar los alias de las rutas en el webpack.config
2. importar las variables en todos nuestros archivos, si se repiten los css hay que hacer únicas las clases

## clase 17 Use state.

1. importar el uso de state

import React, { useState } from 'react';

1. Ejemplo de boton

import '../styles/ProductItem.scss';

/\* CLASE 6 \*/

import add\_to\_cart from '@icons/bt\_add\_to\_cart.svg';

const ProductItem = () => {

const [cart, setCart] = useState('hey');   //Estado, funcion modificadora

const handleClick = () =>{  //por convencion

  setCart('Hola Mundo');

}

return (

  <div className="ProductItem">

    <img src="https://images.pexels.com/photos/276517/pexels-photo-276517.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=650&w=940" alt="" />

    <div className="product-info">

      <div>

        <p>$120,00</p>

        <p>Bike</p>

      </div>

      <figure onClick={ handleClick }>

        <img src={add\_to\_cart} alt="" />

      </figure>

      {cart}

    </div>

  </div>

);

}

export default ProductItem;

## Clase 19 useEffect y consume de APP

Se instalan las dependencias y se agregan en el babelrc

    "plugins": [

        "@babel/plugin-transform-runtime"

    ]

Ejemplo para consumir la API

import React, { useState } from 'react';

import '../styles/ProductItem.scss';

/\* CLASE 6 \*/

import add\_to\_cart from '@icons/bt\_add\_to\_cart.svg';

const ProductItem = () => {

const [cart, setCart] = useState([]);   //Estado, funcion modificadora

const handleClick = () =>{  //por convencion

  setCart([]);

}

return (

  <div className="ProductItem">

    <img src="https://images.pexels.com/photos/276517/pexels-photo-276517.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=650&w=940" alt="" />

    <div className="product-info">

      <div>

        <p>$120,00</p>

        <p>Bike</p>

      </div>

      <figure onClick={ handleClick }>

        <img src={add\_to\_cart} alt="" />

      </figure>

      {}

    </div>

  </div>

);

}

export default ProductItem;

# 21 Hooks useref Pasar la contraseña

Se utiliza el handlesubmit, ejemplo para el login

import React, { useRef} from 'react';

import '@styles/Login.scss';

const Login = () => {

    const form = useRef(null);

    /\* Manejador del submit  https://www.youtube.com/watch?v=raJjjm3rhhU \*/

    const handleSubmit = (event) => {

        /\*Para prevenir que se mande el evento por def<ult de html y se pasen los parametros por get\*/

        event.preventDefault();

        const formData = new FormData(form.current);

        /\*Objeto para ver la informacion name de los imputs\*/

        /\*Tenemos que cambiar id por name en la forma\*/

        const  data = {

            username: formData.get('email'),

            password: formData.get('password')

        }

        console.log(data);

    }

    return (

        <div className="login">

        <div className="login-container">

          <p className="title">Iniciar sesión</p>

          <form action="/" className="form" ref={form}>

            <label htmlFor="email" className="label">Usuario</label>

            <input type="text" name="email" id="email" placeholder="ejemplo@247.com" className="input input-email"/>

            <label htmlFor="password" className="label">Contraseña</label>

            <input type="password" name="password" placeholder="\*\*\*\*\*\*\*\*\*" className="input input-password"/>

            <button

                className="primary-button login-button"

                onClick={handleSubmit}

            >Ingresar

            </button>

          </form>

          <button className="secondary-button signup-button">Registrarse</button>

          <a className="forget" href="/">¿Olvidaste tu contraseña?</a>

        </div>

      </div>

    );

}

export default Login;