**Web Development con React**

Ing. Mario N. Flores C.

Agosto 2019

Tabla de Contenido

[1. Ambiente de desarrollo 4](#_Toc536547675)

[1.1 Instalar Web Storm 4](#_Toc536547676)

[1.2 Configuración de Git 4](#_Toc536547677)

[1.2.1 Configuración global para Git 4](#_Toc536547678)

[1.3 GitLab 4](#_Toc536547679)

[2 NodeJS 5](#_Toc536547680)

[3 React CLI create-react-app 5](#_Toc536547681)

[4 Generar y servir un proyecto React usando create-react-app 5](#_Toc536547682)

[5 Enlazar al repositorio online 6](#_Toc536547683)

[6 Configurar el proyecto React para usar Reactstrap 7](#_Toc536547684)

[6.1 Añadir una barra de navegación de Bootstrap 8](#_Toc536547685)

[6.2 Adicionar un componente catalogo 8](#_Toc536547686)

[7 Componentes React 8](#_Toc536547687)

[7.1 Ejercicio 2: Paso de información entre componentes jerárquicos 11](#_Toc536547688)

[7.2 Eventos 16](#_Toc536547689)

[7.3 Ciclo de vida de React 18](#_Toc536547690)

[8 Laboratorio 1: Componentes de react 19](#_Toc536547691)

[8.1 Objetivos y Resultados 19](#_Toc536547692)

[8.2 Requisitos de asignación 19](#_Toc536547693)

[8.3 Tarea 1 19](#_Toc536547694)

[8.4 Tarea 2 20](#_Toc536547695)

[8.5 Tarea 3 20](#_Toc536547696)

[8.6 Rubrica Tarea 1 23](#_Toc536547697)

[9 Ejercicio 3 Componentes presentational y Container 23](#_Toc536547698)

[10 Ciclo de vida Update 25](#_Toc536547699)

[11 Componentes funcionales 25](#_Toc536547700)

[12 Font Awesome 26](#_Toc536547701)

[13 Header y Footer 27](#_Toc536547702)

[14 Ejercicio: React Router 29](#_Toc536547703)

[14.1 Elementos de Navegacion 31](#_Toc536547704)

[15 Single Page Applications Parte 1 32](#_Toc536547705)

[16 Single Page Applications parte 2 39](#_Toc536547706)

[17 Laboratorio 2: 41](#_Toc536547707)

[17.1 Tarea 1 42](#_Toc536547708)

[17.2 Tarea 2 43](#_Toc536547709)

[17.3 Tarea 3 43](#_Toc536547710)

[17.4 Rubrica Laboratorio 2 45](#_Toc536547711)

[18 Ejercicio: Controlled Forms 45](#_Toc536547712)

[19 Validaciones en formularios controlados 49](#_Toc536547713)

[20 Uncontrolled forms 51](#_Toc536547714)

[21 Redux 53](#_Toc536547715)

[22 React Redux Form 54](#_Toc536547716)

[23 React Redux Form Validation 56](#_Toc536547717)

[24 Laboratorio 3: React forms y Redux 59](#_Toc536547718)

[24.1 Tarea 1 59](#_Toc536547719)

[24.2 Tarea 2 59](#_Toc536547720)

[24.3 Tarea 3 59](#_Toc536547721)

[25 Combinar Reducers 60](#_Toc536547722)

[26 Acciones de Redux 62](#_Toc536547723)

[27 Redux Thunk 63](#_Toc536547724)

[28 React-Redux-Form 68](#_Toc536547725)

[29 Configurar Json-server 69](#_Toc536547726)

[30 Comunicación Cliente Servidor: Fetch 69](#_Toc536547727)

[31 Manejo de errores 71](#_Toc536547728)

[32 Post 72](#_Toc536547729)

[33 Despliegue de la aplicación con Webpack 72](#_Toc536547730)

[34 Docker 72](#_Toc536547731)

[35 Dockerizacion del cliente 73](#_Toc536547732)

[36 Recursos 74](#_Toc536547733)

[37 Definiciones 77](#_Toc536547734)

# Ambiente de desarrollo

## Instalar Web Storm

Ejecutar instalador y aceptar parámetros por defecto.

## Configuración de Git

Para instalar Git en su computadora, vaya a <https://git-scm.com/downloads> para descargar el instalador de Git para su plataforma informática específica. Luego, siga los pasos de instalación.

Puede encontrar más detalles sobre la instalación de Git en <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>.

Los usuarios Windows es recomendable incorporar git para manejo de consola.

### Configuración global para Git

Abra una ventana o terminal de cmd en su computadora.

Compruebe para asegurarse de que Git esté instalado y disponible en la línea de comandos, escribiendo lo siguiente en el símbolo del sistema:

git --version

Para configurar su nombre de usuario para ser utilizado por Git, escriba lo siguiente:

git config --global user.name "Tu nombre"

Para configurar su correo electrónico para ser utilizado por Git, escriba lo siguiente:

git config --global user.email edsonariel@gmail.com

Puede verificar su configuración global predeterminada de Git, puede escribir lo siguiente:

git config --list

Generar su llave publica (IMPORTANTE: poner enter cuando pregunte passphrase):

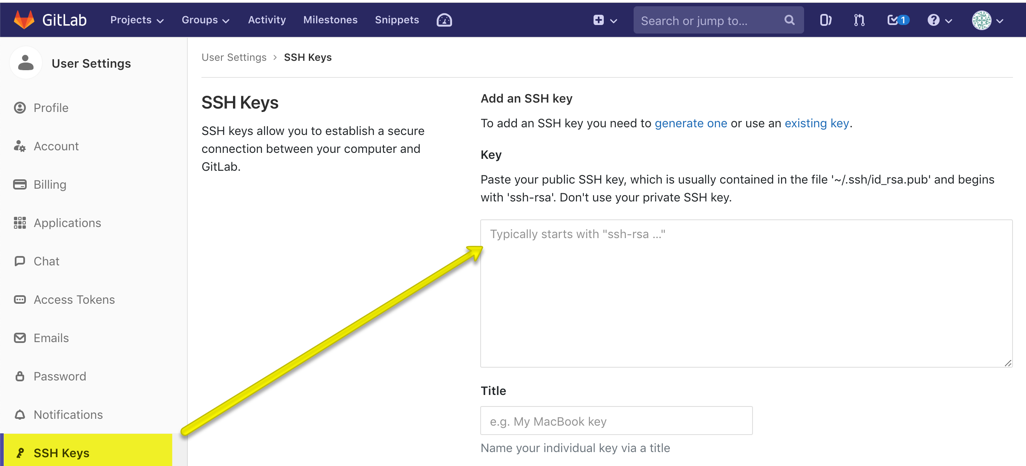
<https://git-scm.com/book/en/v2/Git-on-the-Server-Generating-Your-SSH-Public-Key>

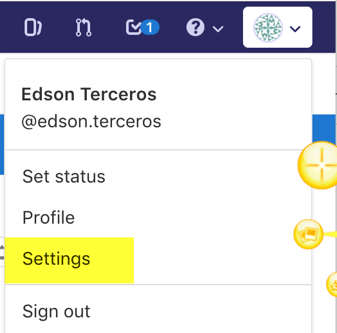
## GitLab

Crear una cuenta de GitLab en <http://gitlab.com/>

Añadir la llave publica a la cuenta de gitLab copiándola del comando:

cat ~/.ssh/id\_rsa.pub





# NodeJS

Puede descargar node desde https://nodejs.org

Se sugiere no utilizar gestores de instalación de paquetes como brew.

Utilice una cuenta de administrador para instalar Node y siga el asistente de instalación de NodeJs.

Verificar con el comando

node -v

npm -v

Instalar yarn siguiendo las instrucciones en la pagina:

https://yarnpkg.com/en/docs/install

# React CLI create-react-app

Instalar create-react-app de forma global con el siguiente comando:

yarn global add create-react-app@1.5.2

Si desea usar npm alternativamente equivale a:

npm install -g create-react-app@1.5.2

Para mas información del CLI escribe:

create-react-app -help

# Generar y servir un proyecto React usando create-react-app

Cree una carpeta de trabajo llamada src en el directorio de su usuario y muévase a esa carpeta:

cd ~

mkdir src

cd src

Luego escriba lo siguiente para crear una nueva aplicación React llamada ssi-react-app:

create-react-app ssi-react-app

Abrir WebStorm y seleccionar que desea abrir un proyecto:



Luego seleccione la carpeta del proyecto que creamos anteriormente con el CLI de react.

Podrá observar que hay un archivo gitignore el cual filtra el contenido que no deseamos sincronizar en el repositorio git. A este debemos adicionar la carpeta .idea para ignorarla.

También podemos ver la estructura básica de un proyecto React.

El archivo package.json maneja las dependencias del proyecto.

El CLI hace que el proyecto tenga automáticamente un repositorio local git, para verificar escribimos en la consola:

git status



# Enlazar al repositorio online

Crear un repositorio llamado ssi-react-app en GitLab.

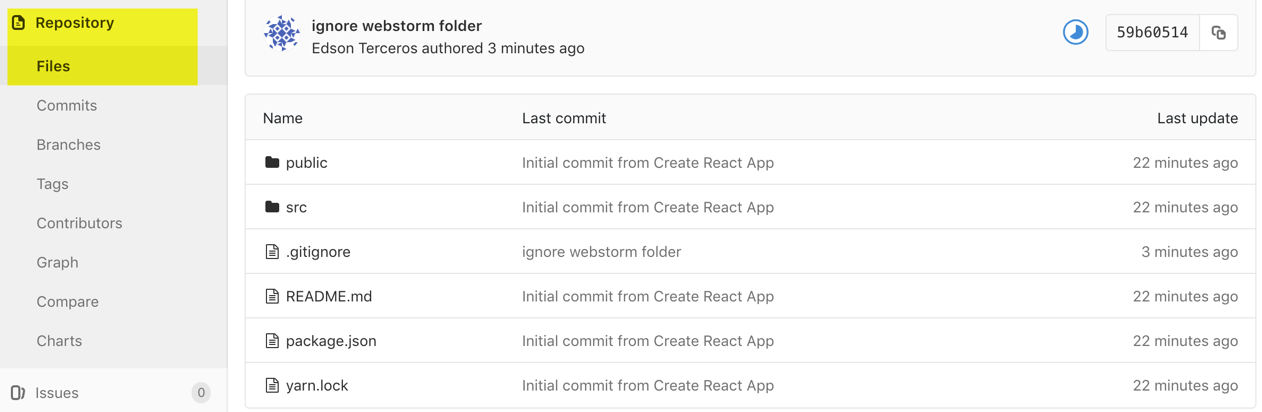
Luego enlázalo al proyecto de tu maquina siguiendo las instrucciones en la pagina luego de crear el proyecto, que seria similar a esto:

cd ~/src/ssi-react-app

git remote add origin git@gitlab.com:[nombre-de-cuenta]/ssi-react-app.git

git push -u origin master

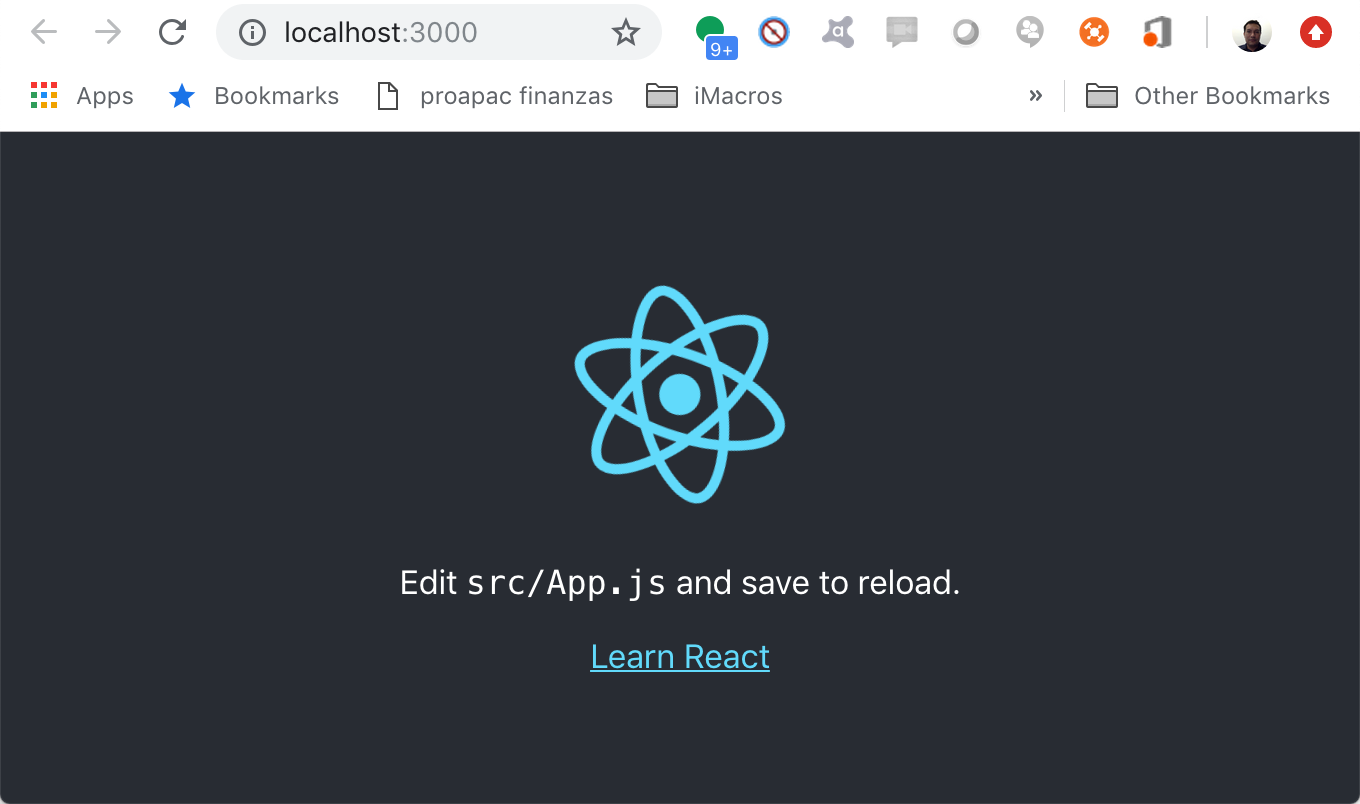
Observar en gitLab que ahora el contenido del proyecto esta en línea o el remoto.



Para desplegar el proyecto en la pestaña de comando de Webstorm escriba:

yarn start

Se abrirá la aplicación en su navegador en la dirección http://localhost:3000/



# Configurar el proyecto React para usar Reactstrap

Para usar reactstrap, escriba en consola para instalar reactstrap, y Bootstrap 4:

yarn add bootstrap@4.0.0

yarn add reactstrap@5.0.0

yarn add [react-popper@0.9.2](mailto:react-popper@0.9.2)

Si usas npm el equivalente es

npm install <package> --save

En el archivo index.js en la carpeta src y agregue la siguiente línea en las importaciones:

import 'bootstrap/dist/css/bootstrap.min.css';

## Añadir una barra de navegación de Bootstrap

Abra App.js en la carpeta src y actualícelo asi:

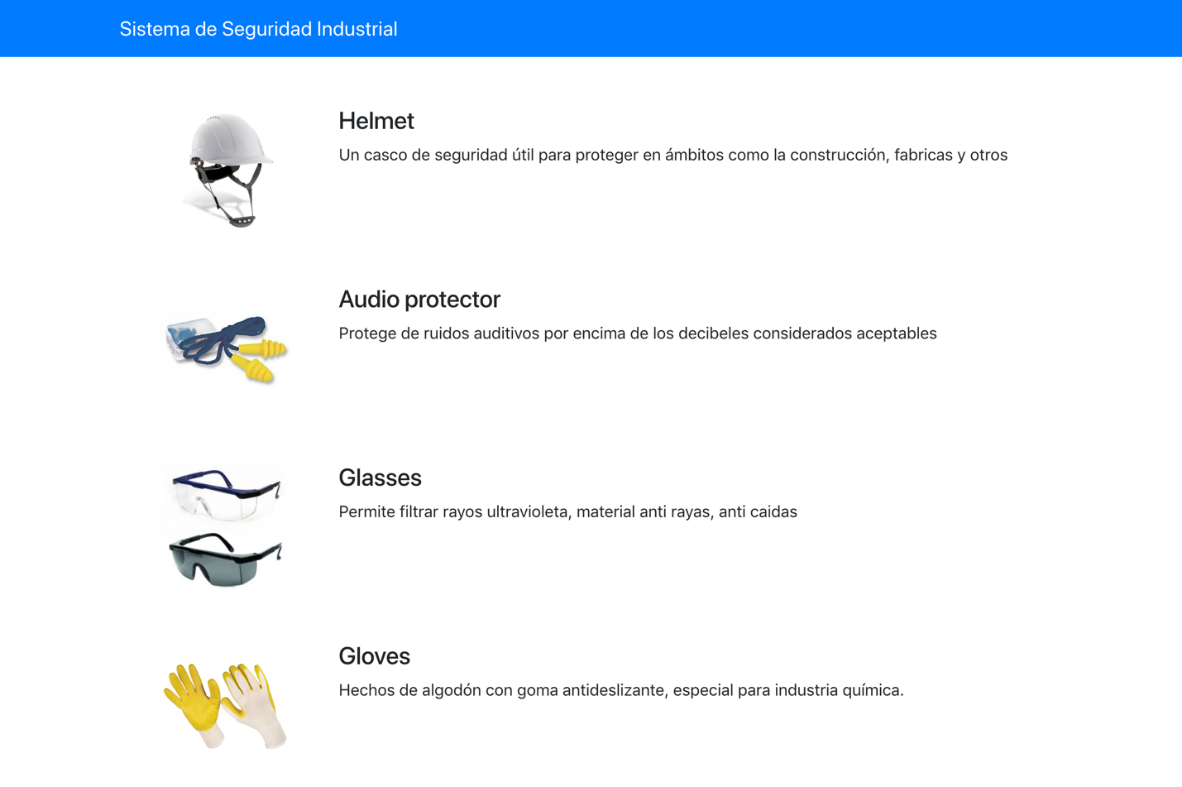
**class** App **extends** Component {  
 render() {  
 **return** (  
 <**div className="App"**>  
 <**Navbar dark color="primary"**>  
 <**div className="container"**>  
 <**NavbarBrand href="/"**>Sistema de Seguridad Industrial</**NavbarBrand**>  
 </**div**>  
 </**Navbar**>  
 </**div**>  
 );  
 }  
}

Hacer commit con el mensaje “Configurando React”.

## Adicionar un componente catalogo

# Componentes React

OBJETIVO:



Descargar el compreso images.zip de la carpeta compartida en github, descomprimirlo y copiar el folder images al proyecto en la ruta “public/assets”.

Crear un folder components en src, donde estarán todos los components que crearemos. Crear un archivo llamado CatalogComponent.js en esta carpeta.

Esto es un live template de react que crea un componente de tipo clase.

En VSC presionar la teclas ctrl + espacio y seleccionar **rcc**.

En el método render adiciona un contenedor de bootstrap y una fila en ella con un objeto de tipo media que desplegara nuestro componente catalog que definiremos luego:

render() {  
 **return** (  
 <**div className="container"**>  
 <**div className="row"**>  
 <**Media list**>  
 {catalog}  
 </**Media**>  
 </**div**>  
 </**div**>  
 );  
}

Con la ayuda del IDE hacer que catalog sea una variable.

En el constructor asignar a la variable this.state un document vacio y en el crear la variable items como un arreglo, asi:

constructor(props, context) {  
 **super**(props, context);  
  
 **this**.**state**={  
 **items**:[]  
 }  
}

Agrege el siguiente contenido al arreglo de items:

{  
 name: 'Helmet',  
 image: '/assets/images/helmet.jpg',  
 category: 'safety',  
 label: 'ultra safe',  
 price: '4.99',  
 description: 'Un casco de seguridad útil para proteger en ámbitos como la construcción, fabricas y otros'  
},  
{  
 name: 'Audio protector',  
 image: '/assets/images/audio.jpg',  
 category: 'safety',  
 label: 'noise filter',  
 price: '12.99',  
 description: 'Protege de ruidos auditivos por encima de los decibeles considerados aceptables'  
},  
{  
 name: 'Glasses',  
 image: '/assets/images/glasses.jpg',  
 category: 'safety',  
 label: 'New technology',  
 price: '31.99',  
 description: 'Permite filtrar rayos ultravioleta, material anti rayas, anti caidas'  
},  
{  
 name: 'Gloves',  
 image: '/assets/images/gloves.jpg',  
 category: 'safety',  
 label: '',  
 price: '12.99',  
 description: 'Hechos de algodón con goma antideslizante, especial para industria química.'  
}

Ahora asignaremos a la variable items este array que creamos asi:

var catalog= this.state.items;

Ahora usaremos la function map para crear nuestro componente element del catalog asi:

**var** catalog= **this**.**state**.**items**.map(item => {  
 **return** (  
 <**div key=**{item.**id**} **className="col-12 mt-5"**>  
 <**Media tag="li"**>  
 <**Media left middle**>  
 <**Media object src=**{item.**image**} **alt=**{item.**name**} />  
 </**Media**>  
 <**Media body className="ml-5"**>  
 <**Media heading**>{item.**name**}</**Media**>  
 <**p**>{item.**description**}</**p**>  
 </**Media**>  
 </**Media**>  
 </**div**>  
 );  
});

Renombraremos nuestro componente a solo Catalog en la definición de la clase y el export.

Con esto tenemos listo nuestro componente catalog, para usarlo abrir App.js y luego del componente Navbar ponga una etiqueta Catalog así:

<Catalog/>

Puesto que no estamos usando los estilos de App.css, borraremos ese contenido.

Por tanto, como tampoco usamos esos estilos en App.js borraremos la clase App definida en el primer div.

Como al visualizar se ve que las imágenes son demasiado grandes arreglamos definiendo tamaños de estas en una constante llamada imgStyle aplicándola luego al media object que muestra las imágenes.

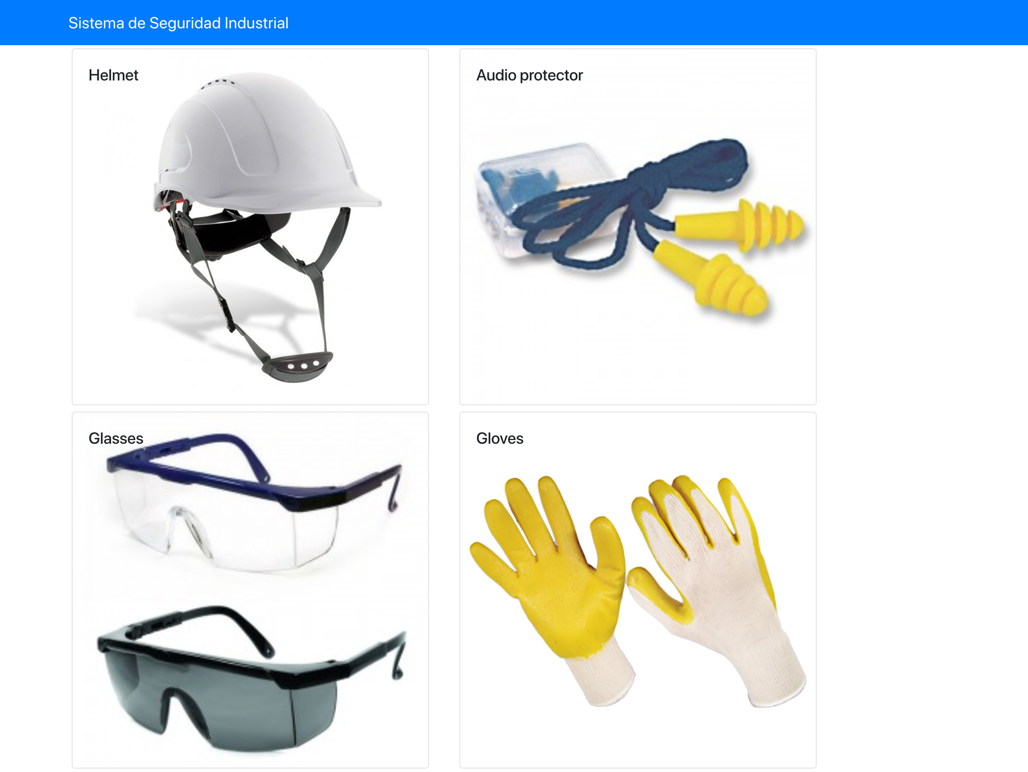
render() {  
 **const** imgStyle = {  
 **maxHeight**: 128,  
 **maxWidth**: 128  
 };  
 **var** catalog= **this**.**state**.**items**.map(item => {  
 **return** (  
 <**div key=**{item.**id**} **className="col-12 mt-5"**>  
 <**Media tag="li"**>  
 <**Media left middle**>  
 <**Media object src=**{item.**image**} **style=**{imgStyle} **alt=**{item.**name**} />

**ESTRUCTURA DEL PROYECTO**

|  |
| --- |
| /src/components/CatalogComponent.js  /src/App.js  /public/assets/images/audio.jpg  /public/assets/images/helmet.jpg  /public/assets/images/glasses.jpg  /public/assets/images/gloves.jpg |
| import React, {Component} from 'react';  import './App.css';  import {Navbar, NavbarBrand} from "reactstrap";  import Catalog from "./components/CatalogComponent";  class App extends Component {  render() {  return (  <div className="App">  <Navbar dark color="primary">  <div className="container">  <NavbarBrand href="/">Sistema de Seguridad Industrial</NavbarBrand>  </div>  </Navbar>  <Catalog/>  </div>  );  }  }  export default App; |
| import React, {Component} from 'react';  import {Media} from "reactstrap";  class Catalog extends Component {  constructor(props, context) {  super(props, context);  this.state = {  items: [  {  id: 0,  name: 'Helmet',  image: '/assets/images/helmet.jpg',  category: 'safety',  label: 'ultra safe',  price: '4.99',  description: 'Un casco de seguridad útil para proteger en ámbitos como la construcción, fabricas y otros'  },  {  id: 1,  name: 'Audio protector',  image: '/assets/images/audio.jpg',  category: 'safety',  label: 'noise filter',  price: '12.99',  description: 'Protege de ruidos auditivos por encima de los decibeles considerados aceptables'  },  {  id: 2,  name: 'Glasses',  image: '/assets/images/glasses.jpg',  category: 'safety',  label: 'New technology',  price: '31.99',  description: 'Permite filtrar rayos ultravioleta, material anti rayas, anti caidas'  },  {  id: 3,  name: 'Gloves',  image: '/assets/images/gloves.jpg',  category: 'safety',  label: '',  price: '12.99',  description: 'Hechos de algodón con goma antideslizante, especial para industria química.'  }  ]  }  }  render() {  let catalog = this.state.items.map(item => {  const imgStyle = {  maxHeight: 128,  maxWidth: 128  };  return (  <div key={item.id} className="col-12 mt-5">  <Media tag="li">  <Media left middle>  <Media object src={item.image} alt={item.name} style={imgStyle}/>  </Media>  <Media body className="ml-5">  <Media heading>{item.name}</Media>  <p>{item.description}</p>  </Media>  </Media>  </div>  );  });  return (  <div className="container">  <div className="row">  <Media list>  {catalog}  </Media>  </div>  </div>  );  }  }  export default Catalog; |

## Ejercicio 2: Paso de información entre componentes jerárquicos

OBJETIVOS



Realizaremos cambios para que en lugar de lista de ítems usemos Cards para visualizar el catalogo.

Crear una carpeta llamada shared en src

mkdir src/shared

Allí crear un archivo llamado items.js

touch src/shared/items.js

Poner el contenido de ítems en el archivo creado en una constante llamada ITEMS, el contenido del archivo debe quedar así:

export const ITEMS = [  
 {  
 name: 'Helmet',  
 image: '/assets/images/helmet.jpg',  
 category: 'safety',  
 label: 'ultra safe',  
 price: '4.99',  
 description: 'Un casco de seguridad útil para proteger en ámbitos como la construcción, fabricas y otros',  
 comments: [  
 {  
 rating: 5,  
 comment: "Imagine all safety PPE in SSI!",  
 author: "Mr T",  
 date: "2012-10-16T17:57:28.556094Z"  
 },  
 {  
 rating: 4,  
 comment: "Gives me a lot of confort, I wish I could get my mother-in-law to have it!",  
 author: "Tania Barrancos",  
 date: "2014-09-05T17:57:28.556094Z"  
 },  
 {  
 rating: 3,  
 comment: "Use it, just use it!",  
 author: "Michael Rodriguez",  
 date: "2015-02-13T17:57:28.556094Z"  
 },  
 {  
 rating: 4,  
 comment: "Ultimate, Reaching for the stars!",  
 author: "Diana Cardozo",  
 date: "2013-12-02T17:57:28.556094Z"  
 },  
 {  
 rating: 2,  
 comment: "It is simply very nice!",  
 author: "50 Cent",  
 date: "2011-12-02T17:57:28.556094Z"  
 }  
 ]  
 },  
 {  
 name: 'Audio protector',  
 image: '/assets/images/audio.jpg',  
 category: 'safety',  
 label: 'noise filter',  
 price: '12.99',  
 description: 'Protege de ruidos auditivos por encima de los decibeles considerados aceptables',  
 comments: [  
 {  
 rating: 5,  
 comment: "Imagine all safety PPE in SSI!",  
 author: "Mr T",  
 date: "2012-10-16T17:57:28.556094Z"  
 },  
 {  
 rating: 5,  
 comment: "Gives me a lot of confort, I wish I could get my mother-in-law to have it!",  
 author: "Tania Barrancos",  
 date: "2014-09-05T17:57:28.556094Z"  
 },  
 {  
 rating: 3,  
 comment: "Use it, just use it!",  
 author: "Michael Rodriguez",  
 date: "2015-02-13T17:57:28.556094Z"  
 },  
 {  
 rating: 4,  
 comment: "Ultimate, Reaching for the stars!",  
 author: "Diana Cardozo",  
 date: "2013-12-02T17:57:28.556094Z"  
 },  
 {  
 rating: 2,  
 comment: "It is simply very nice!",  
 author: "50 Cent",  
 date: "2011-12-02T17:57:28.556094Z"  
 }  
 ]  
 },  
 {  
 name: 'Glasses',  
 image: '/assets/images/glasses.jpg',  
 category: 'safety',  
 label: 'New technology',  
 price: '31.99',  
 description: 'Permite filtrar rayos ultravioleta, material anti rayas, anti caidas',  
 comments: [  
 {  
 rating: 5,  
 comment: "Imagine all safety PPE in SSI!",  
 author: "Mr T",  
 date: "2012-10-16T17:57:28.556094Z"  
 },  
 {  
 rating: 5,  
 comment: "Gives me a lot of confort, I wish I could get my mother-in-law to have it!",  
 author: "Tania Barrancos",  
 date: "2014-09-05T17:57:28.556094Z"  
 },  
 {  
 rating: 3,  
 comment: "Use it, just use it!",  
 author: "Michael Rodriguez",  
 date: "2015-02-13T17:57:28.556094Z"  
 },  
 {  
 rating: 4,  
 comment: "Ultimate, Reaching for the stars!",  
 author: "Diana Cardozo",  
 date: "2013-12-02T17:57:28.556094Z"  
 },  
 {  
 rating: 2,  
 comment: "It is simply very nice!",  
 author: "50 Cent",  
 date: "2011-12-02T17:57:28.556094Z"  
 }  
 ]  
 },  
 {  
 name: 'Gloves',  
 image: '/assets/images/gloves.jpg',  
 category: 'safety',  
 label: '',  
 price: '12.99',  
 description: 'Hechos de algodón con goma antideslizante, especial para industria química.',  
 comments: [  
 {  
 rating: 5,  
 comment: "Imagine all safety PPE in SSI!",  
 author: "Mr T",  
 date: "2012-10-16T17:57:28.556094Z"  
 },  
 {  
 rating: 5,  
 comment: "Gives me a lot of confort, I wish I could get my mother-in-law to have it!",  
 author: "Tania Barrancos",  
 date: "2014-09-05T17:57:28.556094Z"  
 },  
 {  
 rating: 3,  
 comment: "Use it, just use it!",  
 author: "Michael Rodriguez",  
 date: "2015-02-13T17:57:28.556094Z"  
 },  
 {  
 rating: 4,  
 comment: "Ultimate, Reaching for the stars!",  
 author: "Diana Cardozo",  
 date: "2013-12-02T17:57:28.556094Z"  
 },  
 {  
 rating: 2,  
 comment: "It is simply very nice!",  
 author: "50 Cent",  
 date: "2011-12-02T17:57:28.556094Z"  
 }  
 ]  
 }  
];

Para formatear el código puede seleccionar el área y luego click derecho, luego aplicar la opción “Format Selection” o “Formatear Selección”

En el constructor de la Clase App agrege en state la variable items asignada a la constante del paso anterior, así:

**class** App **extends** Component {  
  
 constructor(props, context) {  
 **super**(props, context);  
 **this**.**state**={  
 **items**: ITEMS  
 }  
 }

En el tag Catalog pase la información de los ítems como props, así:

<Catalog items={this.state.items}/>

En el CatalogComponent puesto que quitamos ítems de state, debemos referenciarla desde props, así:

**var** catalog = **this**.**props**.**items**.map(item => {

En el método render del componente de catalogo usaremos Card en lugar de Media como componente principal y quitamos el tag de list de este.

En el div que contiene el Card modificamos la clase a className="col-12 col-md-5 m-1"

Usaremos CardImg directamente para renderizar la imagen en lugar de Media, especificando witdh 100%. En lugar del tag Media body usaremos CardImgOverlay sin especificar ninguna clase. En lugar de Media heading usaremos CardTitle. Removeremos la descripción.

Debe quedar así:

var catalog = this.props.items.map(item => {  
 return (  
 <div key={item.id} className="col-12 col-md-5 m-1">  
 <Card>  
 <CardImg width="100%" src={item.image} alt={item.name}/>  
 <CardImgOverlay>  
 <CardTitle>{item.name}</CardTitle>  
 </CardImgOverlay>  
 </Card>  
 </div>

En el return del componente quitaremos el tag Media usando directamente {catalog} asi:

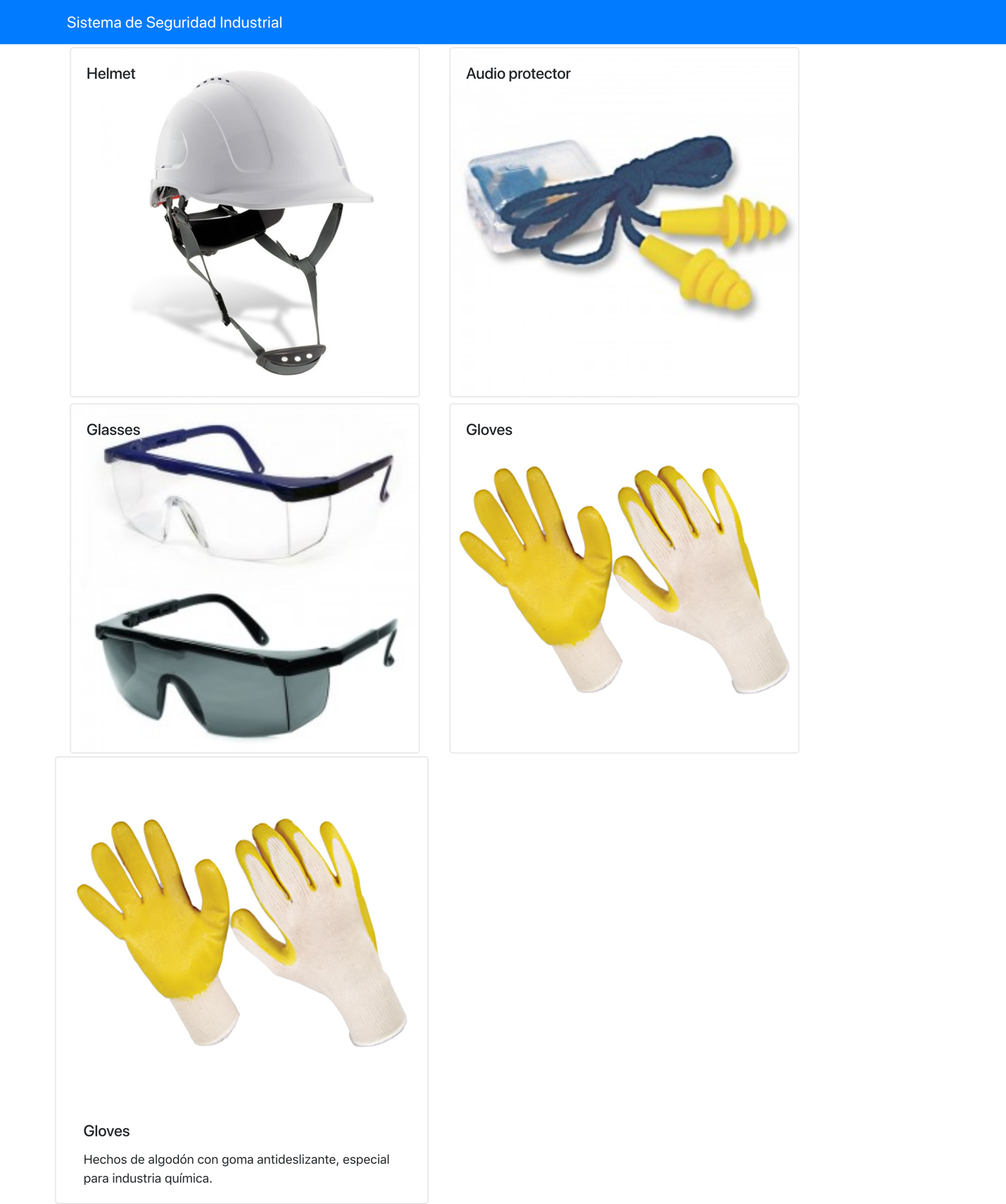
**return** (  
 <**div className="container"**>  
 <**div className="row"**>  
 {catalog}  
 </**div**>  
 </**div**>  
);

Hacer un commit con mensaje uso de props y Card.

|  |
| --- |
| /src/components/CatalogComponent.js  /src/shared/item.js  /src/App.js |
| import React, {Component} from 'react';  import {Card, CardImg, CardImgOverlay, CardTitle} from "reactstrap";  class Catalog extends Component {  constructor(props, context) {  super(props, context);  }  render() {  let catalog = this.props.items.map(item => {  return (  <div key={item.id} className="col-md-5 col-12 mt-5">  <Card>  <CardImg width="100%" src={item.image} alt={item.name}/>  <CardImgOverlay>  <CardTitle>{item.name}</CardTitle>  </CardImgOverlay>  </Card>  </div>  );  });  return (  <div className="container">  <div className="row">  {catalog}  </div>  </div>  );  }  }  export default Catalog; |
| export const ITEMS = [  {  id: 0,  name: 'Helmet',  image: '/assets/images/helmet.jpg',  category: 'safety',  label: 'ultra safe',  price: '4.99',  description: 'Un casco de seguridad útil para proteger en ámbitos como la construcción, fabricas y otros',  comments: [  {  rating: 5,  comment: "Imagine all safety PPE in SSI!",  author: "Mr T",  date: "2012-10-16T17:57:28.556094Z"  },  {  rating: 4,  comment: "Gives me a lot of confort, I wish I could get my mother-in-law to have it!",  author: "Tania Barrancos",  date: "2014-09-05T17:57:28.556094Z"  },  {  rating: 3,  comment: "Use it, just use it!",  author: "Michael Rodriguez",  date: "2015-02-13T17:57:28.556094Z"  },  {  rating: 4,  comment: "Ultimate, Reaching for the stars!",  author: "Diana Cardozo",  date: "2013-12-02T17:57:28.556094Z"  },  {  rating: 2,  comment: "It is simply very nice!",  author: "50 Cent",  date: "2011-12-02T17:57:28.556094Z"  }  ]  },  {  id: 1,  name: 'Audio protector',  image: '/assets/images/audio.jpg',  category: 'safety',  label: 'noise filter',  price: '12.99',  description: 'Protege de ruidos auditivos por encima de los decibeles considerados aceptables',  comments: [  {  rating: 5,  comment: "Imagine all safety PPE in SSI!",  author: "Mr T",  date: "2012-10-16T17:57:28.556094Z"  },  {  rating: 5,  comment: "Gives me a lot of confort, I wish I could get my mother-in-law to have it!",  author: "Tania Barrancos",  date: "2014-09-05T17:57:28.556094Z"  },  {  rating: 3,  comment: "Use it, just use it!",  author: "Michael Rodriguez",  date: "2015-02-13T17:57:28.556094Z"  },  {  rating: 4,  comment: "Ultimate, Reaching for the stars!",  author: "Diana Cardozo",  date: "2013-12-02T17:57:28.556094Z"  },  {  rating: 2,  comment: "It is simply very nice!",  author: "50 Cent",  date: "2011-12-02T17:57:28.556094Z"  }  ]  },  {  id: 2,  name: 'Glasses',  image: '/assets/images/glasses.jpg',  category: 'safety',  label: 'New technology',  price: '31.99',  description: 'Permite filtrar rayos ultravioleta, material anti rayas, anti caidas',  comments: [  {  rating: 5,  comment: "Imagine all safety PPE in SSI!",  author: "Mr T",  date: "2012-10-16T17:57:28.556094Z"  },  {  rating: 5,  comment: "Gives me a lot of confort, I wish I could get my mother-in-law to have it!",  author: "Tania Barrancos",  date: "2014-09-05T17:57:28.556094Z"  },  {  rating: 3,  comment: "Use it, just use it!",  author: "Michael Rodriguez",  date: "2015-02-13T17:57:28.556094Z"  },  {  rating: 4,  comment: "Ultimate, Reaching for the stars!",  author: "Diana Cardozo",  date: "2013-12-02T17:57:28.556094Z"  },  {  rating: 2,  comment: "It is simply very nice!",  author: "50 Cent",  date: "2011-12-02T17:57:28.556094Z"  }  ]  },  {  id: 3,  name: 'Gloves',  image: '/assets/images/gloves.jpg',  category: 'safety',  label: '',  price: '12.99',  description: 'Hechos de algodón con goma antideslizante, especial para industria química.',  comments: [  {  rating: 5,  comment: "Imagine all safety PPE in SSI!",  author: "Mr T",  date: "2012-10-16T17:57:28.556094Z"  },  {  rating: 5,  comment: "Gives me a lot of confort, I wish I could get my mother-in-law to have it!",  author: "Tania Barrancos",  date: "2014-09-05T17:57:28.556094Z"  },  {  rating: 3,  comment: "Use it, just use it!",  author: "Michael Rodriguez",  date: "2015-02-13T17:57:28.556094Z"  },  {  rating: 4,  comment: "Ultimate, Reaching for the stars!",  author: "Diana Cardozo",  date: "2013-12-02T17:57:28.556094Z"  },  {  rating: 2,  comment: "It is simply very nice!",  author: "50 Cent",  date: "2011-12-02T17:57:28.556094Z"  }  ]  }  ]; |
| import React, {Component} from 'react';  import './App.css';  import {Navbar, NavbarBrand} from "reactstrap";  import Catalog from "./components/CatalogComponent";  import {ITEMS} from "./shared/items";  class App extends Component {  constructor(props, context) {  super(props, context);  this.state = {  items: ITEMS  }  }  render() {  return (  <div className="App">  <Navbar dark color="primary">  <div className="container">  <NavbarBrand href="/">Sistema de Seguridad Industrial</NavbarBrand>  </div>  </Navbar>  <Catalog items={this.state.items}/>  </div>  );  }  }  export default App; |

## Eventos

**Objetivo**



Al hacer click en una imagen del catalogo mostraremos el detalle del item selccionado con el click.

Para esto en el state del comopnente de catalogo crearemos la variable selectedItem asignada a null inicialmente.

En el componente Card usaremos el evento onClick llamando a onItemSelect pasando como parámetro el item seleccionado, asi:

<**Card onClick=**{()=>{**this**.onItemSelect(item)}}>

En la función onItemSelect asignaremos el item pasado como parámetro a selecteItem del state asi:

onItemSelect(item) {  
 **this**.setState({  
 **selectedItem**: item  
 })  
}

Implementaremos un método para renderizar el item seleccionado asi:

renderItem(item) {  
 if (item != null) {  
 return (  
 <Card className="col-5”>  
 <CardImg width="100%"

src={item.image} alt={item.name}/>  
 <CardBody>  
 <CardTitle>{item.name}</CardTitle>  
 <CardText>{item.description}</CardText>  
 </CardBody>  
 </Card>  
 );  
  
 } else {  
 return (  
 <div/>  
 );  
 }  
  
}

Donde renderizamos el catalogo a continuación en una nueva fila mostraremos el item seleccionado asi:

**return** (  
 <**div className="container"**>  
 <**div className="row"**>  
 {catalog}  
 </**div**>  
 <**div className="row"**>  
 {**this**.renderItem(**this**.**state**.**selectedItem**)}  
 </**div**>  
 </**div**>  
);

/src/components/CatalogComponent.js

/src/shared/item.js

/src/App.js

|  |
| --- |
| import React, {Component} from 'react';  import {Card, CardBody, CardImg, CardImgOverlay, CardText, CardTitle} from "reactstrap";  class Catalog extends Component {  constructor(props, context) {  super(props, context);  this.state = {  selectedItem: null  }  }  renderItem(item) {  if (item != null) {  return (  <Card>  <CardImg width="100%" src={item.image} alt={item.name}/>  <CardBody>  <CardTitle>{item.name}</CardTitle>  <CardText>{item.description}</CardText>  </CardBody>  </Card>  )  } else {  return (  <div/>  )  }  }  render() {  let catalog = this.props.items.map(item => {  return (  <div key={item.id} className="col-md-5 col-12 mt-5">  <Card onClick={() => {  this.onItemSelect(item)  }}>  <CardImg width="100%" src={item.image} alt={item.name}/>  <CardImgOverlay>  <CardTitle>{item.name}</CardTitle>  </CardImgOverlay>  </Card>  </div>  );  });  return (  <div className="container">  <div className="row">  {catalog}  </div>  <div className="row">  {this.renderItem(this.state.selectedItem)}  </div>  </div>  );  }  onItemSelect(item) {  this.setState(  {  selectedItem: item  }  )  }  }  export default Catalog; |
| import React, {Component} from 'react';  import './App.css';  import {Navbar, NavbarBrand} from "reactstrap";  import Catalog from "./components/CatalogComponent";  import {ITEMS} from "./shared/items";  class App extends Component {  constructor(props, context) {  super(props, context);  this.state = {  items: ITEMS  }  }  render() {  return (  <div className="App">  <Navbar dark color="primary">  <div className="container">  <NavbarBrand href="/">Sistema de Seguridad Industrial</NavbarBrand>  </div>  </Navbar>  <Catalog items={this.state.items}/>  </div>  );  }  }  export default App; |

## Ciclo de vida de React

En el constructor de catalogo poner un log de consola:

*console*.log("Catalog constructor es invocado");

similar en el método render.

Sobreescribir el método componentDidMount y hacer similar log.

Adicionar el plugin react developer tools en Chrome [React Dev Tools](https://github.com/facebook/react-devtools)

Vemos en la consola un warning que no estamos enviando el id como key, adicionamos el id en la constante de ITEMS en cada elemento partiendo de 0.

# Laboratorio 1: Componentes de react

En este laboratorio, usted agregará un nuevo componente a la aplicación React para mostrar los detalles de un item seleccionado. Utilizará el componente Tarjeta y el componente de lista sin estilo de Bootstrap para preparar la vista para este nuevo componente.

En esta asignación, agregará un nuevo componente a la aplicación React para mostrar los detalles de un item seleccionado. Utilizará el componente Card y el componente de lista sin estilo Bootstrap para preparar la vista para este nuevo componente.

Instrucciones paso a paso de asignación

## Objetivos y Resultados

Continuará trabajando con la aplicación React que ha estado desarrollando en los ejercicios. Agregará un nuevo componente llamado ItemdetailComponent que mostrará los detalles de un plato seleccionado. A continuación, diseñará la vista para el componente utilizando el componente de tarjeta. Al final de esta tarea, debe haber completado las siguientes tareas:

* Haber creado un nuevo ItemdetailComponent y lo agregó a su aplicación React.
* Haber actualizado la vista de ItemdetailComponent para mostrar los detalles del item seleccionado usando un componente de tarjeta reactstrap.
* Haber actualizado la vista de ItemdetailComponent para mostrar la lista de comentarios sobre el item usando el componente Bootstrap Unstyled list.

## Requisitos de asignación

Completar las siguientes tareas. Las instrucciones detalladas para cada tarea se dan a continuación. La imagen de la página web completa que se incluye, indica la ubicación dentro de la página web que se actualizará mediante las tres tareas.

## Tarea 1

En esta tarea, creara un nuevo componente ItemdetailComponent para mostrar los detalles de un item específico en su propio componente (mover la implementación actual a su propio componente) e incluirá el componente en la vista del componente catalogo para que se muestren allí:

* Cree una nueva clase ItemDetail en un archivo llamado ItemdetailComponent.js en la carpeta de componentes.
* Reemplace la tarjeta que muestra el item seleccionado en la vista de CatalogComponent por el nuevo componente ItemdetailComponent, y asegúrese de pasar la información del item seleccionado como props al ItemdetailComponent.
* Exporte la clase ItemDetail de este archivo para que pueda importarse en CatalogComponent.js y usarse para construir la vista del item seleccionado.
* Devuelva un <div> en la función render(). Este <div> debe usar la clase row de Bootstrap para colocar el contenido dentro de <div> (Tip: el div de la clase catalog). Este div mostrará los detalles del item en una Card y en otro div la lista de comentarios en paralelo para pantallas medianas a extra grandes, pero los apilará para las pantallas xs y sm a continuación se da el detalle.
* El Card debe estar encerrado dentro de un <div> con las clases de columna apropiadas de Bootstrap para que ocupe las 12 columnas completas para los tamaños de pantalla xs y sm, y 5 columnas para las pantallas md y superiores. También aplique una clase de m-1 a este div.
* Los comentarios deben incluirse en un <div> al que se aplican las clases de columna adecuadas para que ocupen las 12 columnas completas para los tamaños de pantalla xs y sm, y 5 columnas para las pantallas md y superiores. También aplique una clase de m-1 a este div. Por el momento puede dejar el div vacio.
* Si el item es nulo, deberías devolver un <div> vacío.

## Tarea 2

En esta tarea, usted agregará un componente Card a la vista ItemdetailComponent para mostrar los detalles del item especificado arriba:

* Implemente una función llamada renderItem() que toma el item como un parámetro y devuelve el código JSX para presentar los detalles del item en un Card de reactstrap. Ya lo ha visto como parte de la clase CatalogComponent en el ejercicio anterior.
* Muestra el nombre del item como el título de la Tarjeta y la descripción como el texto de la Tarjeta.

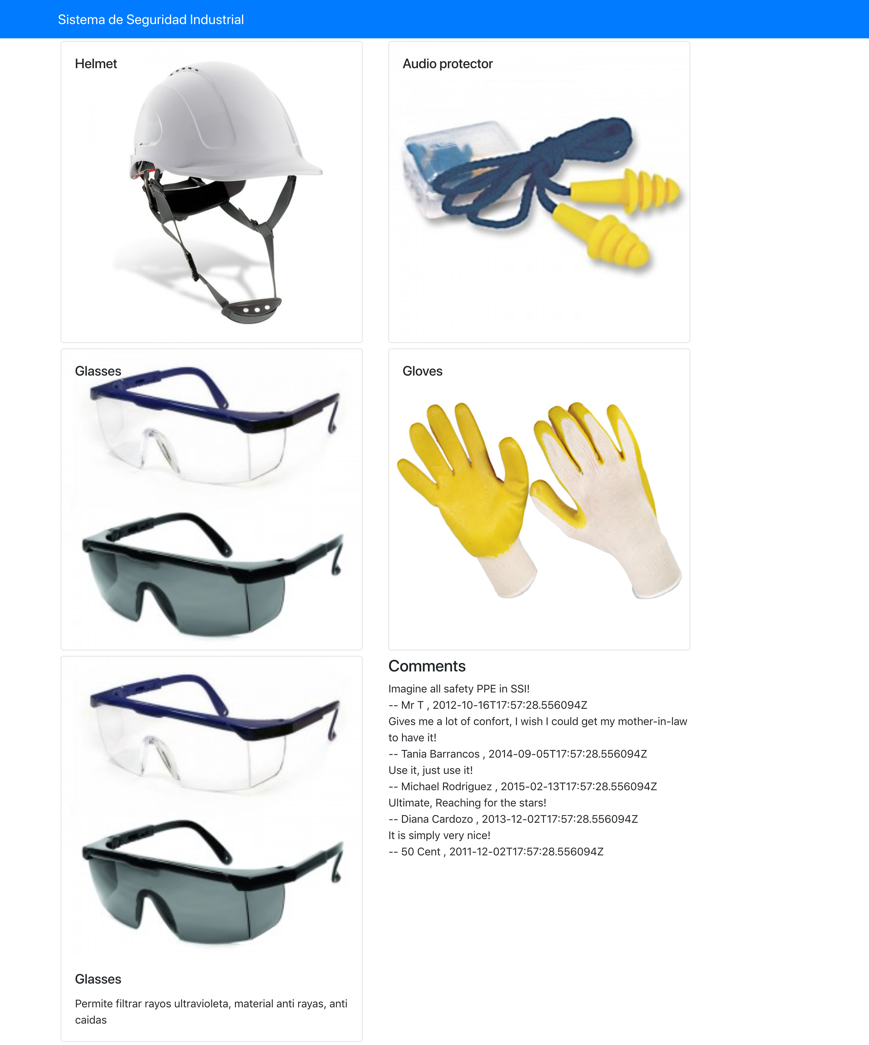
## Tarea 3

En esta tarea, usted utilizará los comentarios que se incluyen en el objeto de item anterior para mostrar una lista de los comentarios para el item. Utilice su conocimiento de JavaScript para recordar cómo accedería a una propiedad interna en un objeto de JavaScript que apunta a una matriz de objetos de JavaScript (comments). Esta tarea implica los siguientes pasos:

* Implemente una función llamada renderComments() que toma la matriz de comentarios como un parámetro (cuide de pasar nulo si no hay item seleccionado) y expone cada comentario como se muestra en la imagen a continuación. Debe utilizar la clase Bootstrap list-unstyled en la lista.
* Cada comentario debe mostrarse en dos líneas, la primera mostrando el comentario y la segunda línea mostrando el nombre del autor del comentario y la fecha.
* Los comentarios deben contener un encabezado <h4> con la palabra "Comentarios".
* Recuerde incluir el encabezado y los comentarios dentro de un <div> antes de devolver el código JSX. De lo contrario, React no hará el diseño correctamente.
* Si los comentarios son nulos, debe devolver un <div> vacío.

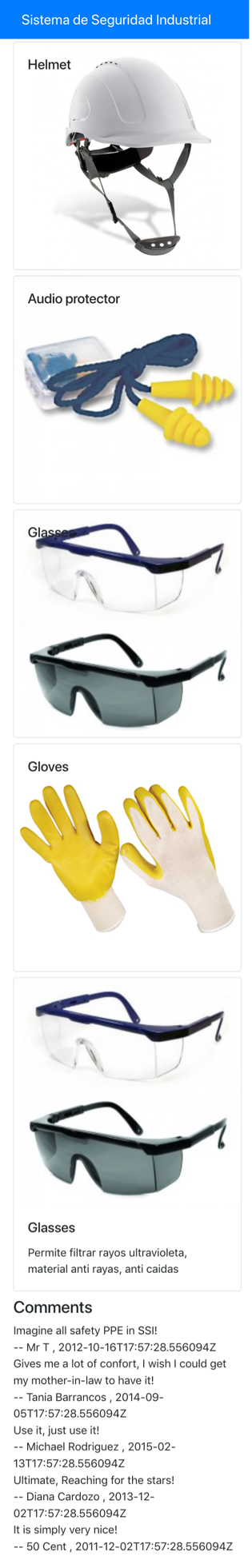
En pantallas grandes debe lucir asi:

**OBJETIVOS**



|  |
| --- |
| /src/components/CatalogComponent.js  /src/components/ItemDetailComponent.js  /shared/ítems.js  /src/App.js |
| import React, {Component} from 'react';  import {Card, CardImg, CardImgOverlay, CardTitle} from "reactstrap";  import ItemDetail from "./ItemdetailComponent";  class Catalog extends Component {  constructor(props, context) {  super(props, context);  this.state = {  selectedItem: null  };  }  onItemSelect(item) {  this.setState({  selectedItem: item  })  }  render() {  var catalog = this.props.items.map(item => {  return (  <div key={item.id} className="col-12 col-md-5 m-1">  <Card onClick={() => {  this.onItemSelect(item)  }}>  <CardImg width="100%" src={item.image} alt={item.name}/>  <CardImgOverlay>  <CardTitle>{item.name}</CardTitle>  </CardImgOverlay>  </Card>  </div>  );  });  return (  <div className="container">  <div className="row">  {catalog}  </div>  <ItemDetail selectedItem={this.state.selectedItem}/>  </div>  );  }  }  export default Catalog; |
| import React, {Component} from 'react';  import {Card, CardBody, CardImg, CardText, CardTitle} from "reactstrap";  class ItemDetail extends Component {  renderItem(item) {  if (item != null) {  return (  <Card>  <CardImg width="100%" src={item.image} alt={item.name}/>  <CardBody>  <CardTitle>{item.name}</CardTitle>  <CardText>{item.description}</CardText>  </CardBody>  </Card>  );  } else {  return (  <div/>  );  }  }  renderComments(comments) {  if (comments != null) {  const commentsItems = comments.map((comment) => {  return (  <li key={comment.id}>  <ul className="list-unstyled">  <li>{comment.comment}</li>  <li>-- {comment.author} , {comment.date}</li>  </ul>  </li>  );  });  return (  <div>  <h4>Comments</h4>  <ul className="list-unstyled">  {commentsItems}  </ul>  </div>  );  } else {  return (  <div/>  );  }  }  render() {  return (  <div className="row">  <div className="col-12 col-md-5 m-1">  {this.renderItem(this.props.selectedItem)}  </div>  <div className="col-12 col-md-5 m-1">  {this.renderComments(this.props.selectedItem ? this.props.selectedItem.comments : null)}  </div>  </div>  );  }  }  export default ItemDetail; |
| import React, {Component} from 'react';  import './App.css';  import {Navbar, NavbarBrand} from "reactstrap";  import Catalog from "./components/CatalogComponent";  import {ITEMS} from "./shared/items";  class App extends Component {  constructor(props, context) {  super(props, context);  this.state = {  items: ITEMS  }  }  render() {  return (  <div className="App">  <Navbar dark color="primary">  <div className="container">  <NavbarBrand href="/">Sistema de Seguridad Industrial</NavbarBrand>  </div>  </Navbar>  <Catalog items={this.state.items}/>  </div>  );  }  }  export default App; |

En pantallas pequeñas debe lucir asi



## Rubrica Tarea 1

Estudiante que fue evaluado: …………………………….…………………………….

Estudiante que realizo la evaluación: …………………………….…………………………….

|  |  |  |
| --- | --- | --- |
| Actividades | Puntaje | Cumple |
| T1.1: Se ha agregado un nuevo componente ItemdetailComponent a su aplicación React. | 15 |  |
| T1.2: Incluyó el ItemDetail en la vista de su CatalogComponent para mostrar el item seleccionado. | 10 |  |
| T1.3: Se esta pasando el item seleccionado como props al componente ItemDetail. | 5 |  |
| T1.4 Se usa las clases de Bootstrap apropiadas para la tarjeta, de modo que ocupe toda la fila para los tamaños de pantalla xs y sm, y 5 columnas para las pantallas md y superiores. | 10 |  |
| T1.5: Se usa las clases de Bootstrap apropiadas en el div que contiene la lista de comentarios, de modo que ocupe toda la fila para los tamaños de pantalla xs y sm, y 5 columnas para las pantallas md y superiores. | 10 |  |
| T2: Usó el componente Tarjeta para mostrar los detalles del item. | 25 |  |
| T3: Incluyó una lista de comentarios sobre el plato en la vista itemdetail. | 25 |  |
| PUNTUACION TOTAL |  |  |

Envie la nota de si compañero a [edsonariel@gmail.com](mailto:edsonariel@gmail.com)

# Ejercicio 3 Componentes presentational y Container

A continuación la idea es dejar el componente App.js limpio y crear un nuevo componente que se encargue de almacenar los datos que dejamos en App.js y de pintar los elementos html que están en App.js.

Haremos que el componente Catalog sea un componente puramente presentacional y lo mismo el componente ItemDetail y haremos fluir hacia arriba la información de estos en un componente contenedor que crearemos en este ejercicio.

Como el nuevo componente hara lo que hace App.js entonces:

Copie App.js y pegue en la carpeta components con el nombre MainComponent.js y actualice su contenido de la siguiente manera:

Modifique la ruta de los imports como sea necesario.

Renombre la clase a Main y cuide que el export sea consecuente con ese nombre.

ItemDetail se desplegara como un componente hermano a mismo nivel que el catalogo, por lo cual extraeremos la información necesaria (selectedIem) al componente padre MainComponent. Tip ponga en pantalla dividida y extraiga las propiedades de un lado al otro.

Borrar el constructor de Catalog por estar sin información.

Tambien mueva el método onItemSelect hacia MainComponent, por lo cual el componente Catalog ya no conoce este método y hay que enviárselo como props y enviarlo con el parámetro solo de ItemId asi:

<**Catalog items=**{**this**.**state**.**items**}  
 **onClick=**{(itemId) => {**this**.onItemSelect(itemId)}}/>

En el componente Catalog hacia uso de el evento onClick en un Card donde necesitamos reflejar que ahora viene desde props y que ademas ahora solo recibe de parámetro un id asi:

**var** catalog = **this**.**props**.**items**.map(item => {  
 **return** (  
 <**div key=**{item.**id**} **className="col-12 col-md-5 m-1"**>  
 <**Card onClick=**{() => {**this**.**props**.onClick(item.**id**)}}>

Modifique la function onItemSelect para reflejar que ahora recibe un id como parámetro.

En el tag ItemDetail se pasara el parametro item extrayendo del arreglo de items mediante la función filter haciendo match por el id y extrayendo el resultado en la posición 0 asi:

<ItemDetail item={this.state.items.filter((item)=>item.id===this.state.selectedItem)[0]}/>

No olvide cambiar de nombre de la variable selectedItem por item tal como especifica props.

Ahora el componente App ya no necesita nada mas que el componente Main, por lo cual limpiaremos todo dato irrelevante asi:

**import** React, {Component} **from 'react'**;  
**import** Main **from "./components/MainComponent"**;  
  
**class** App **extends** Component {  
  
 render() {  
 **return** (  
 <**div**>  
 <**Main**/>  
 </**div**>  
 );  
 }  
}  
  
**export default** App;

Como ahora el Item detail se muestra en Main component debemos quitarlo de Catalog component.

Para que la fecha mostrada en los comentarios aparesca adecuadamente usar este ejemplo:

{new Intl.*DateTimeFormat*('en-US',{year:'numeric', month: 'short',day:'2-digit'}).format(new *Date*(*Date*.parse(comment.date)))}

No olvide remover el className en App en el componente Main.

En la consola aparece error por que los comments no tienen id, solucione este problema.

Haga commit con el mensaje “Presentational, container components”

# Ciclo de vida Update

En el constructor de Main component adicione:

*console*.log("Main constructor es invocado");

en el metodo render adicione:

*console*.log("Main render es invocado");

y adicione este método al componente Main:

componentDidMount() {  
 *console*.log("Main componentDidMount es invocado");  
}

Para ItemDetail adicione:

componentDidMount() {  
 *console*.log("ItemDetail componentDidMount es invocado");  
}  
  
componentDidUpdate() {  
 *console*.log("ItemDetail componentDidUpdate es invocado");  
}

en el método render de ItemDetail adicione este log:

*console*.log("ItemDetail render es invocado");

Observe la secuencia mostrada en el navegador en la consola cuando abrimos la pagina y cuando actualizamos el item seleccionado por medio del clic.

Haga commit como “ciclo de vida update”

# Componentes funcionales

Puesto que Catalog y ItemDetail no mantienen datos en state y su función es solo el renderizado de la información basado en props los convertiremos en componentes funcionales.

Eliminar el constructor de Catalog component.

Eliminar el import y la definición de Clase de Catalog.

Eliminar la definición del método render.

Ahora tomaremos el Card y haremos su componente funcional, para esto defina una función llamada

RenderCatalogItem(props)

Otra forma de definir la función es expandiendo props asi:

RenderCatalogItem({item, onClick}){}

Use la función return para crear un elemento de react y devolver el Card. Note que ahora las propiedades item y onClick son parámetros directo ya no pasamos props debe quedar asi:

**function** *Render*Catalog*Item*({item, onClick}) {  
 **return** (  
 <**Card onClick=**{() => {  
 onClick(item.**id**)  
 }}>  
 <**CardImg width="100%" src=**{item.**image**} **alt=**{item.**name**}/>  
 <**CardImgOverlay**>  
 <**CardTitle**>{item.**name**}</**CardTitle**>  
 </**CardImgOverlay**>  
 </**Card**>  
  
 )  
}

Otra forma de definir un componente funcional es mediante una arrow function asignada a una constante, lo haremos para el componente que llamaremos Catalog como se ve en el ejemplo:

**const** Catalog= (props) => {  
  
};

Copie el contenido restante dentro de esta nueva función. Debe quedar asi:

**const** *Catalog* = (props) => {  
 **var** catalog = **this**.**props**.**items**.map(item => {  
 **return** (  
 <**div key=**{item.**id**} **className="col-12 col-md-5 m-1"**>  
 </**div**>  
 );  
 });  
 **return** (  
 <**div className="container"**>  
 <**div className="row"**>  
 {catalog}  
 </**div**>  
 </**div**>  
 );  
};

Adicione el ItemDetail al div del return de la variable catalog asi:

<div key={item.id} className="col-12 col-md-5 m-1">  
 <RenderCatalogItem item={item} onClick={props.onClick}/>  
</div>

Revise la variable props.items de catalog ahora viene directamente como parametros props

Para crear un componente presenacional sin estado o funcional presione

CTRL+J rsf

Haga la misma idea para ItemDetail, quitar la definición de clase y luego convertir renderItem a una función que defina un componente funcional y descomponga props para recibir item:

**function** *RenderItem*({item}) {

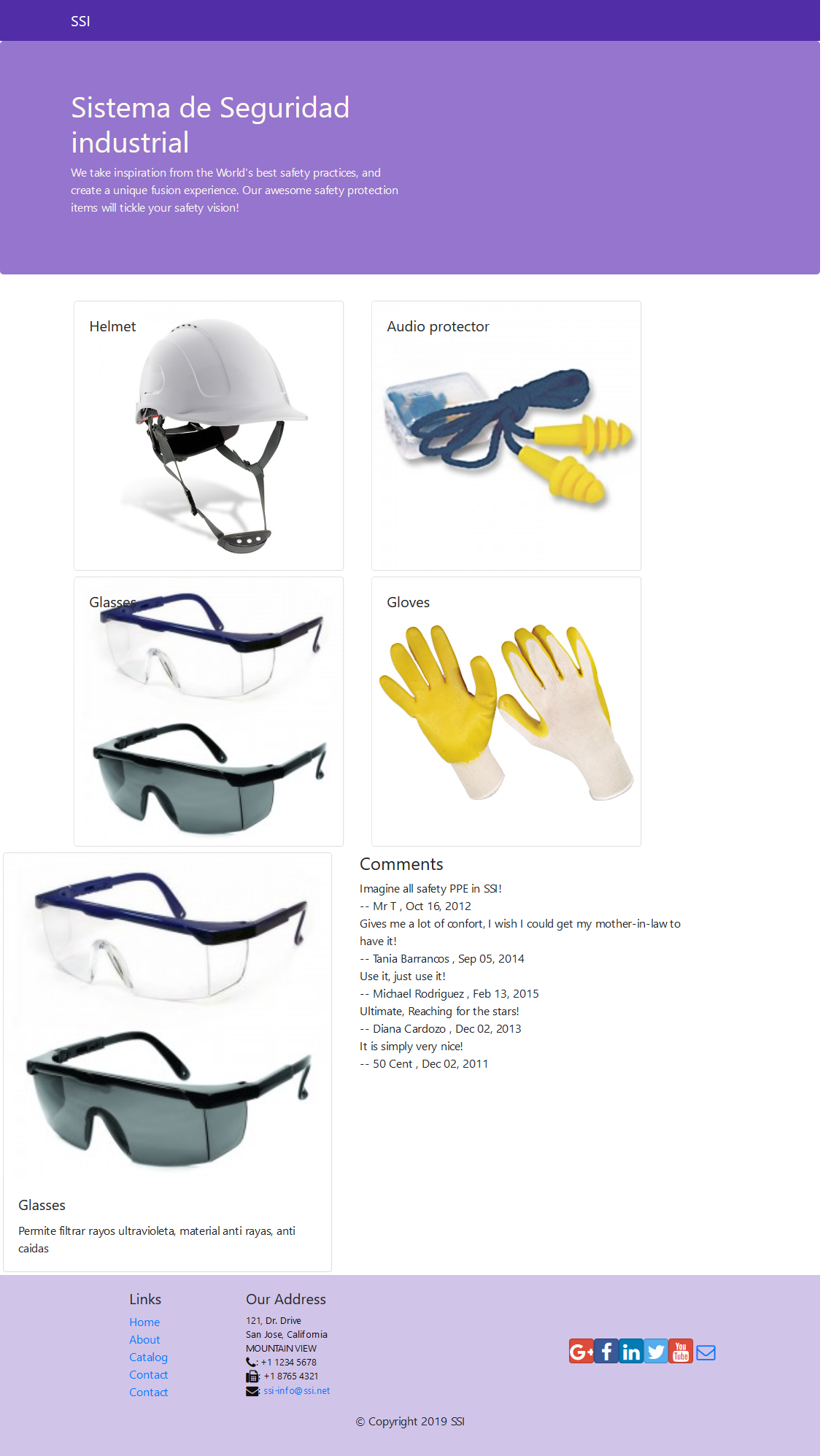
lo propio para la function renderComments

la function render debe definer ahora un componente functional llamado ItemDetail asi:

Dentro esta function reemplace las llamadas a las antiguas funciones por los Tag de los components funcionales que acabamos de crear, asi:

# Font Awesome

OBJETIVO



Para adicionar Font awesome ejecute:

Npm install font-awesome@4.7.0

Npm install bootstrap-social@5.1.1

En index.js despues del import de bootstrap importe Font awesome:

import 'font-awesome/css/font-awesome.css';  
import 'bootstrap-social/bootstrap-social.css';

# Header y Footer

Crear un archivo llamado HeaderComponent.js en el ponga el siguiente contenido:

**import** React, {Component} **from 'react'**;  
**import** {*Navbar*, *NavbarBrand*, *Jumbotron*} **from 'reactstrap'**;  
  
**class** Header **extends** Component {  
 render() {  
 **return** (

/\*agrupa varios elementos sin necesidad de un div su forma abreviada es <></>\*/  
 <**React.Fragment**>  
 <**Navbar dark**>  
 <**div className="container"**>  
 <**NavbarBrand href="/"**>SSI</**NavbarBrand**>  
 </**div**>  
 </**Navbar**>  
 <**Jumbotron**>  
 <**div className="container"**>  
 <**div className="row row-header"**>  
 <**div className="col-12 col-sm-6"**>  
 <**h1**>Sistema de Seguridad industrial</**h1**>  
 <**p**>We take inspiration from the World's best safety practices, and create a unique  
 fusion experience. Our awesome safety protection items will tickle your safety  
 vision!</**p**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**Jumbotron**>  
 </**React.Fragment**>  
 );  
 }  
}  
  
**export default** Header;

Podemos ver que ahora este componente es el que contiene el Navbar por lo cual lo removeremos del componente Main.

Tambien observamos que tiene un elemento nuevo de reactstrap llamado Jumbotron.

Observe la clase row-header en un div del Jumbotron esa será una clase nueva que escribiremos.

Adicione el componente Header en el componente MainComponent antes del catalogo.

Crear un archivo llamado FooterComponent.js en el ponga el siguiente contenido:

import React from 'react';  
  
function *Footer*(props) {  
 return (  
 <div className="footer">  
 <div className="container">  
 <div className="row justify-content-center">  
 <div className="col-4 offset-1 col-sm-2">  
 <h5>Links</h5>  
 <ul className="list-unstyled">  
 <li><a href="#">Home</a></li>  
 <li><a href="#">About</a></li>  
 <li><a href="#">Catalog</a></li>  
 <li><a href="#">Contact</a></li>  
 <li><a href="contactus.html">Contact</a></li>  
 </ul>  
 </div>  
 <div className="col-7 col-sm-5">  
 <h5>Our Address</h5>  
 <address>  
 121, Dr. Drive<br/>  
 San Jose, California<br/>  
 MOUNTAIN VIEW<br/>  
 <i className="fa fa-phone fa-lg"></i>: +1 1234 5678<br/>  
 <i className="fa fa-fax fa-lg"></i>: +1 8765 4321<br/>  
 <i className="fa fa-envelope fa-lg"></i>: <a href="mailto:ssi-info@ssi.net">  
 ssi-info@ssi.net</a>  
 </address>  
 </div>  
 <div className="col-12 col-sm-4 align-self-center">  
 <div className="text-center">  
 <a className="btn btn-social-icon btn-google" href="http://google.com/+"><i  
 className="fa fa-google-plus"></i></a>  
 <a className="btn btn-social-icon btn-facebook"  
 href="http://www.facebook.com/profile.php?id="><i className="fa fa-facebook"></i></a>  
 <a className="btn btn-social-icon btn-linkedin" href="http://www.linkedin.com/in/"><i  
 className="fa fa-linkedin"></i></a>  
 <a className="btn btn-social-icon btn-twitter" href="http://twitter.com/"><i  
 className="fa fa-twitter"></i></a>  
 <a className="btn btn-social-icon btn-google" href="http://youtube.com/"><i  
 className="fa fa-youtube"></i></a>  
 <a className="btn btn-social-icon" href="mailto:"><i className="fa fa-envelope-o"></i></a>  
 </div>  
 </div>  
 </div>  
 <div className="row justify-content-center">  
 <div className="col-auto">  
 <p>© Copyright 2019 SSI</p>  
 </div>  
 </div>  
 </div>  
 </div>  
 )  
}  
  
export default *Footer*;

Adicionar el footer después de ItemDetail en MainComponent.

Fianalmente copie el siguiente contenido de clases a App.css

No olvide verificar que App.js esta importando App.css asi:

import './App.css';

.row-header{

margin:0px auto;

padding:0px auto;

}

.row-content {

margin:0px auto;

padding: 50px 0px 50px 0px;

border-bottom: 1px ridge;

min-height:400px;

}

.footer{

background-color: #D1C4E9;

margin:0px auto;

padding: 20px 0px 20px 0px;

}

.jumbotron {

padding:70px 30px 70px 30px;

margin:0px auto;

background: #9575CD ;

color:floralwhite;

}

address{

font-size:80%;

margin:0px;

color:#0f0f0f;

}

.navbar-dark {

background-color: #512DA8;

}

Si usted observa que el fondo del jumbotron no esta reflejado en el browser puede asegurarse que se respete lo que definio en el css mediante: **!important** en el estilo correspondiente.

Haga un commit con el mensaje “Header y footer”.

|  |
| --- |
| /src/components/CatalogComponent.js  /src/components/FooterComponent.js  /src/components/HeaderComponent.js  /src/components/ItemdetailsComponent.js  /src/components/MainComponent.js  /shared/ítems.js |
| /src/App.js |
| import React from 'react';  import {Card, CardImg, CardImgOverlay, CardTitle} from "reactstrap";  function RenderCatalogItem({item, onClick}) {  return (  <Card onClick={() => {  onClick(item.id)  }}>  <CardImg width="100%" src={item.image} alt={item.name}/>  <CardImgOverlay>  <CardTitle>{item.name}</CardTitle>  </CardImgOverlay>  </Card>  );  }  const Catalog = (props) => {  var catalog = props.items.map(item => {  return (  <div key={item.id} className="col-12 col-md-5 m-1">  <RenderCatalogItem item={item} onClick={props.onClick}/>  </div>  );  });  return (  <div className="container">  <div className="row">  {catalog}  </div>  </div>  );  };  export default Catalog; |
| import React from 'react';  function Footer(props) {  return (  <div className="footer">  <div className="container">  <div className="row justify-content-center">  <div className="col-4 offset-1 col-sm-2">  <h5>Links</h5>  <ul className="list-unstyled">  <li><a href="#">Home</a></li>  <li><a href="#">About</a></li>  <li><a href="#">Catalog</a></li>  <li><a href="#">Contact</a></li>  <li><a href="contactus.html">Contact</a></li>  </ul>  </div>  <div className="col-7 col-sm-5">  <h5>Our Address</h5>  <address>  121, Dr. Drive<br/>  San Jose, California<br/>  MOUNTAIN VIEW<br/>  <i className="fa fa-phone fa-lg"></i>: +1 1234 5678<br/>  <i className="fa fa-fax fa-lg"></i>: +1 8765 4321<br/>  <i className="fa fa-envelope fa-lg"></i>: <a href="mailto:ssi-info@ssi.net">  ssi-info@ssi.net</a>  </address>  </div>  <div className="col-12 col-sm-4 align-self-center">  <div className="text-center">  <a className="btn btn-social-icon btn-google" href="http://google.com/+"><i  className="fa fa-google-plus"></i></a>  <a className="btn btn-social-icon btn-facebook"  href="http://www.facebook.com/profile.php?id="><i className="fa fa-facebook"></i></a>  <a className="btn btn-social-icon btn-linkedin" href="http://www.linkedin.com/in/"><i  className="fa fa-linkedin"></i></a>  <a className="btn btn-social-icon btn-twitter" href="http://twitter.com/"><i  className="fa fa-twitter"></i></a>  <a className="btn btn-social-icon btn-google" href="http://youtube.com/"><i  className="fa fa-youtube"></i></a>  <a className="btn btn-social-icon" href="mailto:"><i className="fa fa-envelope-o"></i></a>  </div>  </div>  </div>  <div className="row justify-content-center">  <div className="col-auto">  <p>© Copyright 2019 SSI</p>  </div>  </div>  </div>  </div>  )  }  export default Footer; |
| import React, {Component} from 'react';  import {Jumbotron, Navbar, NavbarBrand} from 'reactstrap';  class Header extends Component {  render() {  return (  /\*agrupa varios elementos sin necesidad de un div su forma abreviada es <></>\*/  <React.Fragment>  <Navbar dark>  <div className="container">  <NavbarBrand href="/">SSI</NavbarBrand>  </div>  </Navbar>  <Jumbotron>  <div className="container">  <div className="row row-header">  <div className="col-12 col-sm-6">  <h1>Sistema de Seguridad industrial</h1>  <p>We take inspiration from the World's best safety practices, and create a unique  fusion experience. Our awesome safety protection items will tickle your safety  vision!</p>  </div>  </div>  </div>  </Jumbotron>  </React.Fragment>  );  }  }  export default Header; |
| import React from 'react';  import {Card, CardBody, CardImg, CardText, CardTitle} from "reactstrap";  function RenderItem({item}) {  if (item != null) {  return (  <Card>  <CardImg width="100%" src={item.image} alt={item.name}/>  <CardBody>  <CardTitle>{item.name}</CardTitle>  <CardText>{item.description}</CardText>  </CardBody>  </Card>  );  } else {  return (  <div/>  );  }  }  function RenderComments({comments}) {  if (comments != null) {  const commentsItems = comments.map((comment) => {  return (  <li key={comment.id}>  <ul className="list-unstyled">  <li>{comment.comment}</li>  <li>-- {comment.author} , {new Intl.DateTimeFormat('en-US', {  year: 'numeric',  month: 'short',  day: '2-digit'  }).format(new Date(Date.parse(comment.date)))}</li>  </ul>  </li>  );  });  return (  <div>  <h4>Comments</h4>  <ul className="list-unstyled">  {commentsItems}  </ul>  </div>  );  } else {  return (  <div/>  );  }  }  const ItemDetail = (props) => {  console.log("ItemDetail render es invocado");  return (  <div className="row">  <div className="col-12 col-md-5 m-1">  <RenderItem item={props.item}/>  </div>  <div className="col-12 col-md-5 m-1">  <RenderComments comments={props.item ? props.item.comments : null}/>  </div>  </div>  );  };  export default ItemDetail; |
| import React, {Component} from 'react';  import {ITEMS} from "../shared/items";  import Catalog from "./CatalogComponent";  import ItemDetail from "./ItemdetailComponent";  import Header from "./HeaderComponent";  import Footer from "./FooterComponent";  class Main extends Component {  constructor(props, context) {  super(props, context);  this.state = {  items: ITEMS,  selectedItem: null  };  console.log("Main constructor es invocado");  }  componentDidMount() {  console.log("Main componentDidMount es invocado");  }  onItemSelect(itemId) {  this.setState({  selectedItem: itemId  })  }  render() {  console.log("Main render es invocado");  return (  <div>  <Header/>  <Catalog items={this.state.items}  onClick={(itemId) => {  this.onItemSelect(itemId)  }}/>  <ItemDetail item={this.state.items.filter((item) => item.id === this.state.selectedItem)[0]}/>  <Footer/>  </div>  );  }  }  export default Main; |
| import React, {Component} from 'react';  import './App.css';  import {ITEMS} from "./shared/items";  import Main from "./components/MainComponent";  class App extends Component {  constructor(props, context) {  super(props, context);  this.state = {  items: ITEMS  }  }  render() {  return (  <div className="App">  <Main/>  </div>  );  }  }  export default App; |

# Ejercicio: React Router

**OBJETIVO**

|  |  |
| --- | --- |
|  |  |

Aprendemos a usar el Router de React para configurar la navegación entre varias páginas en una aplicación React.

Instale react router ejecutando:

yarn add [react-router-dom@4.2.2](mailto:react-router-dom@4.2.2)

En App.js encierre todo el contenido del return de render en el tag BrowserRouter asi:

**import** {BrowserRouter} **from "react-router-dom"**;  
  
**class** App **extends** Component {  
  
 render() {  
 **return** (  
 <**BrowserRouter**>  
 <**div**>  
 <**Main**/>  
 </**div**>  
 </**BrowserRouter**>  
 );  
 }

Observe que el tag es importado de react-router-dom

Cree un nuevo archivo llamado HomeComponent.js en la carpeta de components con

Escriba rsc para crear el componente sin estado funcional con el nombre Home.

En el div de return ponga la clase container. En este div de momento adicione un h4 con el texto Home. Debe quedar asi:

**import** React **from 'react'**;  
  
**const** *Home* = () => {  
 **return** (  
 <**div className="container"**>  
 <**h4**>Home</**h4**>  
 </**div**>  
 );  
};  
  
**export default** *Home*;

En el componente Main quite la variable selectedItem de state. Tambien quite el método onItemSelect.

En la función render quite Catalog e ItemDetail quedando solo header y footer.

Despues de header inserte un tag Switch de react-router-dom

Dentro el Swith agrege tag Route y Redirect necesarios con las propiedades path y component asi:

<Switch>  
 <Route path='/home' component={*HomePage*} />  
 <Route exact path='/catalog' component={() => <Catalog items={this.state.items} />} />  
 <Redirect to="/home" />  
</Switch>

Defina un componente funcional para home dentro la función render asi:

render() {  
 **const** *HomePage* = () => {  
 **return**(  
 <**Home**/>  
 );  
 };

Podemos verificar en este momento que el browser esta mostrando el componente Home.

Podemos también probar cualquier ruta desconocida y verificar que redirect funciona.

Podemos también probar cambiando la ruta a catalog en el browser y verificar que nos permite navegar ese componente.

## Elementos de Navegacion

En header en el tag Navbar especifique la propiedad expand=”md” para que el menú se muestre expandido en pantallas mas grandes que médium.

<Navbar dark expand="md">

En el NavbarBrand especificar la clase de bootstrap para margen derecho automatico:

<NavbarBrand className="mr-auto" href="/">SSI</NavbarBrand>

Reemplace el texto SSI por una imagen asi:

<img src="assets/images/safetylogo.png" alt="SSI" width="40" height="40"/>

Despues de NavbarBrand agregue navegaciones asi:

<Nav navbar>  
 <NavItem>  
 <NavLink className="nav-link" to='/home'><span className="fa fa-home fa-lg"></span> Home</NavLink>  
 </NavItem>  
 <NavItem>  
 <NavLink className="nav-link" to='/aboutus'><span className="fa fa-info fa-lg"></span> About Us</NavLink>  
 </NavItem>  
 <NavItem>  
 <NavLink className="nav-link" to='/catalog'><span className="fa fa-list fa-lg"></span> Catalog</NavLink>  
 </NavItem>  
 <NavItem>  
 <NavLink className="nav-link" to='/contactus'><span className="fa fa-address-card fa-lg"></span> Contact Us</NavLink>  
 </NavItem>  
</Nav>

Todos los tag corresponden a reactstrap a excepción de NavLink que debe ser de react-router-dom.

Visualice como queda a momento el menú y verifique que la navegación funciona desde el menú.

Para lograr que en pantallas pequeñas el menú se colapse envuelva todos los Nav en un tag Collapse de reactstrap asi:

<Collapse navbar>  
 <Nav navbar>  
 <NavItem>

Para lograr que el colapse se active y desactive adicionaremos un botón justo encima de NavbarBrand:

<NavbarToggler onClick={this.toggleNav} />

Este toogler usara una función toogleNav que implementaremos luego.

El tag Collapse necesita una propiedad llamada isOpen que le dice cuando debe contraerse o expandir, nosotros haremos que su valor sea una variable llamada isNavOpen en state, adiconela asi:

<Collapse isOpen={this.state.isNavOpen} navbar>

Como necesitamos state en el componente Header, adicione un constructor con esa variable en state:

constructor(props, context) {  
 **super**(props, context);  
 **this**.**state**={  
 **isNavOpen**:**false** }  
}

Defina la función toogleNav que llamamos en NavbarToggler asi:

toggleNav() {  
 **this**.setState({  
 **isNavOpen**: !**this**.**state**.**isNavOpen** });  
}  
  
render() {

Puesto que en NavbarToggler llamamos a la función sin un arrow function necesitamos hacer binding en el constructor para esta función sea accesible al escribir this.toogleNav asi:

constructor(props, context) {  
 **super**(props, context);  
 **this**.**state**={  
 **isNavOpen**:**false** };  
 **this**.toggleNav = **this**.toggleNav.bind(**this**);  
}

Verifique el funcionamiento del toogle en pantallas pequeñas en el browser.

En el footer use el tag Link de react-router-dom en lugar del tag <a>, asi:

<ul className="list-unstyled">  
 <li><Link to='/home'>Home</Link></li>  
 <li><Link to='/aboutus'>About Us</Link></li>  
 <li><Link to='/catalog'>Catalog</Link></li>  
 <li><Link to='/contactus'>Contact Us</Link></li>  
</ul>

Haga commit con el mensaje “React router”

|  |
| --- |
| /src/components/CatalogComponent.js  /src/components/FooterComponent.js  /src/components/HeaderComponent.js  /src/components/HomeComponent.js  /src/components/ItemDetailComponent.js  /src/components/MainComponent.js  /shared/ítems.js  /src/App.js  /src/index.js |
| import React from 'react';  import {Card, CardImg, CardImgOverlay, CardTitle} from "reactstrap";  function RenderCatalogItem({item, onClick}) {  return (  <Card onClick={() => {  onClick(item.id)  }}>  <CardImg width="100%" src={item.image} alt={item.name}/>  <CardImgOverlay>  <CardTitle>{item.name}</CardTitle>  </CardImgOverlay>  </Card>  );  }  const Catalog = (props) => {  var catalog = props.items.map(item => {  return (  <div key={item.id} className="col-12 col-md-5 m-1">  <RenderCatalogItem item={item} onClick={props.onClick}/>  </div>  );  });  return (  <div className="container">  <div className="row">  {catalog}  </div>  </div>  );  };  export default Catalog; |
| import React from 'react';  function Footer(props) {  return (  <div className="footer">  <div className="container">  <div className="row justify-content-center">  <div className="col-4 offset-1 col-sm-2">  <h5>Links</h5>  <ul className="list-unstyled">  <li><a href="#">Home</a></li>  <li><a href="#">About</a></li>  <li><a href="#">Catalog</a></li>  <li><a href="#">Contact</a></li>  <li><a href="contactus.html">Contact</a></li>  </ul>  </div>  <div className="col-7 col-sm-5">  <h5>Our Address</h5>  <address>  121, Dr. Drive<br/>  San Jose, California<br/>  MOUNTAIN VIEW<br/>  <i className="fa fa-phone fa-lg"></i>: +1 1234 5678<br/>  <i className="fa fa-fax fa-lg"></i>: +1 8765 4321<br/>  <i className="fa fa-envelope fa-lg"></i>: <a href="mailto:ssi-info@ssi.net">  ssi-info@ssi.net</a>  </address>  </div>  <div className="col-12 col-sm-4 align-self-center">  <div className="text-center">  <a className="btn btn-social-icon btn-google" href="http://google.com/+"><i  className="fa fa-google-plus"></i></a>  <a className="btn btn-social-icon btn-facebook"  href="http://www.facebook.com/profile.php?id="><i className="fa fa-facebook"></i></a>  <a className="btn btn-social-icon btn-linkedin" href="http://www.linkedin.com/in/"><i  className="fa fa-linkedin"></i></a>  <a className="btn btn-social-icon btn-twitter" href="http://twitter.com/"><i  className="fa fa-twitter"></i></a>  <a className="btn btn-social-icon btn-google" href="http://youtube.com/"><i  className="fa fa-youtube"></i></a>  <a className="btn btn-social-icon" href="mailto:"><i className="fa fa-envelope-o"></i></a>  </div>  </div>  </div>  <div className="row justify-content-center">  <div className="col-auto">  <p>© Copyright 2019 SSI</p>  </div>  </div>  </div>  </div>  )  }  export default Footer; |
| import React, {Component} from 'react';  import {Collapse, Jumbotron, Nav, Navbar, NavbarBrand, NavbarToggler, NavItem} from 'reactstrap';  import {NavLink} from "react-router-dom";  class Header extends Component {  constructor(props, context) {  super(props, context);  this.state = {  isNavOpen: false  };  this.toggleNav = this.toggleNav.bind(this);  }  toggleNav() {  this.setState({  isNavOpen: !this.state.isNavOpen  })  }  render() {  return (  /\*agrupa varios elementos sin necesidad de un div su forma abreviada es <></>\*/  <React.Fragment>  <Navbar dark expand="md">  <div className="container">  <NavbarToggler onClick={this.toggleNav}/>  <NavbarBrand className="mr-auto" href="/">  <img src="assets/images/safetylogo.png" alt="SSI" width="40" height="40"/>  </NavbarBrand>  <Collapse isOpen={this.state.isNavOpen} navbar>  <Nav navbar>  <NavItem>  <NavLink className="nav-link" to='/home'><span  className="fa fa-home fa-lg"></span> Home</NavLink>  </NavItem>  <NavItem>  <NavLink className="nav-link" to='/aboutus'><span  className="fa fa-info fa-lg"></span> About Us</NavLink>  </NavItem>  <NavItem>  <NavLink className="nav-link" to='/catalog'><span  className="fa fa-list fa-lg"></span> Catalog</NavLink>  </NavItem>  <NavItem>  <NavLink className="nav-link" to='/contactus'><span  className="fa fa-address-card fa-lg"></span> Contact Us</NavLink>  </NavItem>  </Nav>  </Collapse>  </div>  </Navbar>  <Jumbotron>  <div className="container">  <div className="row row-header">  <div className="col-12 col-sm-6">  <h1>Sistema de Seguridad industrial</h1>  <p>We take inspiration from the World's best safety practices, and create a unique  fusion experience. Our awesome safety protection items will tickle your safety  vision!</p>  </div>  </div>  </div>  </Jumbotron>  </React.Fragment>  );  }  }  export default Header; |
| import React from 'react';  const Home = () => {  return (  <div className="container">  <h4>Home</h4>  </div>  );  };  export default Home; |
| import React from 'react';  import {Card, CardBody, CardImg, CardText, CardTitle} from "reactstrap";  function RenderItem({item}) {  if (item != null) {  return (  <Card>  <CardImg width="100%" src={item.image} alt={item.name}/>  <CardBody>  <CardTitle>{item.name}</CardTitle>  <CardText>{item.description}</CardText>  </CardBody>  </Card>  );  } else {  return (  <div/>  );  }  }  function RenderComments({comments}) {  if (comments != null) {  const commentsItems = comments.map((comment) => {  return (  <li key={comment.id}>  <ul className="list-unstyled">  <li>{comment.comment}</li>  <li>-- {comment.author} , {new Intl.DateTimeFormat('en-US', {  year: 'numeric',  month: 'short',  day: '2-digit'  }).format(new Date(Date.parse(comment.date)))}</li>  </ul>  </li>  );  });  return (  <div>  <h4>Comments</h4>  <ul className="list-unstyled">  {commentsItems}  </ul>  </div>  );  } else {  return (  <div/>  );  }  }  const ItemDetail = (props) => {  console.log("ItemDetail render es invocado");  return (  <div className="row">  <div className="col-12 col-md-5 m-1">  <RenderItem item={props.item}/>  </div>  <div className="col-12 col-md-5 m-1">  <RenderComments comments={props.item ? props.item.comments : null}/>  </div>  </div>  );  };  export default ItemDetail; |
| import React, {Component} from 'react';  import {ITEMS} from "../shared/items";  import Catalog from "./CatalogComponent";  import Header from "./HeaderComponent";  import Footer from "./FooterComponent";  import {Redirect, Route, Switch} from "react-router-dom";  import Home from "./HomeComponent";  class Main extends Component {  constructor(props, context) {  super(props, context);  this.state = {  items: ITEMS  };  console.log("Main constructor es invocado");  }  render() {  const HomePage = () => {  return (  <Home/>  );  };  console.log("Main render es invocado");  return (  <div>  <Header/>  <Switch>  <Route path='/home' component={HomePage}/>  <Route exact +'/catalog' component={() => <Catalog items={this.state.items}/>}/>  <Redirect to="/home"/>  </Switch>  <Footer/>  </div>  );  }  }  export default Main; |
| import React, {Component} from 'react';  import './App.css';  import {ITEMS} from "./shared/items";  import Main from "./components/MainComponent";  import {BrowserRouter} from "react-router-dom";  class App extends Component {  constructor(props, context) {  super(props, context);  this.state = {  items: ITEMS  }  }  render() {  return (  <BrowserRouter>  <div className="App">  <Main/>  </div>  </BrowserRouter>  );  }  }  export default App; |

# Single Page Applications Parte 1

OBJETIVOS

|  |  |
| --- | --- |
|  |  |
|  |  |

Agregue un nuevo componente llamado ContactComponent,js con el siguiente contenido:

**import** React **from 'react'**;  
  
**function** *Contact*(props) {  
 **return**(  
 <**div className="container"**>  
 <**div className="row row-content"**>  
 <**div className="col-12"**>  
 <**h3**>Location Information</**h3**>  
 </**div**>  
 <**div className="col-12 col-sm-4 offset-sm-1"**>  
 <**h5**>Our Address</**h5**>  
 <**address**>  
 121, Dr. Drive<**br** />  
 San Jose, California<**br** />  
 MOUNTAIN VIEW<**br** />  
 <**i className="fa fa-phone"**></**i**>: +1 1234 5678<**br** />  
 <**i className="fa fa-fax"**></**i**>: +1 8765 4321<**br** />  
 <**i className="fa fa-envelope"**></**i**>: <**a href="mailto:ssi@food.net"**>ssi@food.net</**a**>  
 </**address**>  
 </**div**>  
 <**div className="col-12 col-sm-6 offset-sm-1"**>  
 <**h5**>Map of our Location</**h5**>  
 </**div**>  
 <**div className="col-12 col-sm-11 offset-sm-1"**>  
 <**div className="btn-group" role="group"**>  
 <**a role="button" className="btn btn-primary" href="tel:+112345678"**><**i className="fa fa-phone"**></**i**> Call</**a**>  
 <**a role="button" className="btn btn-info"**><**i className="fa fa-skype"**></**i**> Skype</**a**>  
 <**a role="button" className="btn btn-success" href="mailto:ssi@security.net"**><**i className="fa fa-envelope-o"**></**i**> Email</**a**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 );  
}  
  
**export default** *Contact*;

En el componente MainComponent adicione la ruta para contact asi:

<Route exact path='/contactus' component={*Contact*} />} />

Verificar la navegación a este nuevo componente en el browser.

Remplace el contenido de ítems.js por el siguiente:

export const ITEMS = [  
 {  
 id: 0,  
 name: 'Helmet',  
 image: '/assets/images/helmet.jpg',  
 category: 'safety',  
 label: 'ultra safe',  
 price: '4.99',  
 featured: true,  
 description: 'Un casco de seguridad útil para proteger en ámbitos como la construcción, fabricas y otros',  
 },  
 {  
 id: 1,  
 name: 'Audio protector',  
 image: '/assets/images/audio.jpg',  
 category: 'safety',  
 label: 'noise filter',  
 price: '12.99',  
 featured: false,  
 description: 'Protege de ruidos auditivos por encima de los decibeles considerados aceptables',  
 },  
 {  
 id: 2,  
 name: 'Glasses',  
 image: '/assets/images/glasses.jpg',  
 category: 'safety',  
 label: 'New technology',  
 price: '31.99',  
 featured: false,  
 description: 'Permite filtrar rayos ultravioleta, material anti rayas, anti caidas',  
 },  
 {  
 id: 3,  
 name: 'Gloves',  
 image: '/assets/images/gloves.jpg',  
 category: 'safety',  
 label: '',  
 price: '12.99',  
 featured: false,  
 description: 'Hechos de algodón con goma antideslizante, especial para industria química.',  
 }  
];

Cree el archivo comments.js con el siguiente contenido:

**export const** COMMENTS =  
 [  
 {  
 **id**: 0,  
 **itemId**: 0,  
 **rating**: 5,  
 **comment**: **"Imagine all safety PPE in SSI!"**,  
 **author**: **"Mr T"**,  
 **date**: **"2012-10-16T17:57:28.556094Z"** },  
 {  
 **id**: 1,  
 **itemId**: 0,  
 **rating**: 4,  
 **comment**: **"Gives me a lot of confort, I wish I could get my mother-in-law to have it!"**,  
 **author**: **"Tania Barrancos"**,  
 **date**: **"2014-09-05T17:57:28.556094Z"** },  
 {  
 **id**: 2,  
 **itemId**: 0,  
 **rating**: 3,  
 **comment**: **"Use it, just use it!"**,  
 **author**: **"Michael Rodriguez"**,  
 **date**: **"2015-02-13T17:57:28.556094Z"** },  
 {  
 **id**: 3,  
 **itemId**: 0,  
 **rating**: 4,  
 **comment**: **"Ultimate, Reaching for the stars!"**,  
 **author**: **"Diana Cardozo"**,  
 **date**: **"2013-12-02T17:57:28.556094Z"** },  
 {  
 **id**: 4,  
 **itemId**: 0,  
 **rating**: 2,  
 **comment**: **"It's your birthday, we're gonna party!"**,  
 **author**: **"25 Cent"**,  
 **date**: **"2011-12-02T17:57:28.556094Z"** },  
 {  
 **id**: 5,  
 **itemId**: 1,  
 **rating**: 5,  
 **comment**: **"Imagine all safety PPE in SSI!"**,  
 **author**: **"Mr T"**,  
 **date**: **"2012-10-16T17:57:28.556094Z"** },  
 {  
 **id**: 6,  
 **itemId**: 1,  
 **rating**: 4,  
 **comment**: **"Gives me a lot of confort, I wish I could get my mother-in-law to have it!"**,  
 **author**: **"Tania Barrancos"**,  
 **date**: **"2014-09-05T17:57:28.556094Z"** },  
 {  
 **id**: 7,  
 **itemId**: 1,  
 **rating**: 3,  
 **comment**: **"Use it, just use it!"**,  
 **author**: **"Michael Rodriguez"**,  
 **date**: **"2015-02-13T17:57:28.556094Z"** },  
 {  
 **id**: 8,  
 **itemId**: 1,  
 **rating**: 4,  
 **comment**: **"Ultimate, Reaching for the stars!"**,  
 **author**: **"Diana Cardozo"**,  
 **date**: **"2013-12-02T17:57:28.556094Z"** },  
 {  
 **id**: 9,  
 **itemId**: 1,  
 **rating**: 2,  
 **comment**: **"It's your birthday, we're gonna party!"**,  
 **author**: **"25 Cent"**,  
 **date**: **"2011-12-02T17:57:28.556094Z"** },  
 {  
 **id**: 10,  
 **itemId**: 2,  
 **rating**: 5,  
 **comment**: **"Imagine all safety PPE in SSI!"**,  
 **author**: **"Mr T"**,  
 **date**: **"2012-10-16T17:57:28.556094Z"** },  
 {  
 **id**: 11,  
 **itemId**: 2,  
 **rating**: 4,  
 **comment**: **"Gives me a lot of confort, I wish I could get my mother-in-law to have it!"**,  
 **author**: **"Tania Barrancos"**,  
 **date**: **"2014-09-05T17:57:28.556094Z"** },  
 {  
 **id**: 12,  
 **itemId**: 2,  
 **rating**: 3,  
 **comment**: **"Use it, just use it!"**,  
 **author**: **"Michael Rodriguez"**,  
 **date**: **"2015-02-13T17:57:28.556094Z"** },  
 {  
 **id**: 13,  
 **itemId**: 2,  
 **rating**: 4,  
 **comment**: **"Ultimate, Reaching for the stars!"**,  
 **author**: **"Diana Cardozo"**,  
 **date**: **"2013-12-02T17:57:28.556094Z"** },  
 {  
 **id**: 14,  
 **itemId**: 2,  
 **rating**: 2,  
 **comment**: **"It's your birthday, we're gonna party!"**,  
 **author**: **"25 Cent"**,  
 **date**: **"2011-12-02T17:57:28.556094Z"** },  
 {  
 **id**: 15,  
 **itemId**: 3,  
 **rating**: 5,  
 **comment**: **"Imagine all safety PPE in SSI!"**,  
 **author**: **"Mr T"**,  
 **date**: **"2012-10-16T17:57:28.556094Z"** },  
 {  
 **id**: 16,  
 **itemId**: 3,  
 **rating**: 4,  
 **comment**: **"Gives me a lot of confort, I wish I could get my mother-in-law to have it!"**,  
 **author**: **"Tania Barrancos"**,  
 **date**: **"2014-09-05T17:57:28.556094Z"** },  
 {  
 **id**: 17,  
 **itemId**: 3,  
 **rating**: 3,  
 **comment**: **"Use it, just use it!"**,  
 **author**: **"Michael Rodriguez"**,  
 **date**: **"2015-02-13T17:57:28.556094Z"** },  
 {  
 **id**: 18,  
 **itemId**: 3,  
 **rating**: 4,  
 **comment**: **"Ultimate, Reaching for the stars!"**,  
 **author**: **"Diana Cardozo"**,  
 **date**: **"2013-12-02T17:57:28.556094Z"** },  
 {  
 **id**: 19,  
 **itemId**: 3,  
 **rating**: 2,  
 **comment**: **"It's your birthday, we're gonna party!"**,  
 **author**: **"25 Cent"**,  
 **date**: **"2011-12-02T17:57:28.556094Z"** }  
 ];

Cree el archivo employees.js con el siguiente contenido:

export const EMPLOYEES = [  
 {  
 id: 0,  
 name: 'Henry Cavill',  
 image: '/assets/images/henry.jpg',  
 jobPosition: 'Chief Enterprise Officer',  
 jobCode: 'CEO',  
 featured: true,  
 jobDescription: 'El CEO, esta encargado de dirigir la empresa para conseguir los objetivos alineados con las estrategias de la ' +  
 'empresa. Su rol fundamental es marcar las estrategias a seguir cuidando que los indicadores relacionados se cumplan y de ser ' +  
 'necesario proponer nuevas estrategias que permitan a la empresa adaptarse al mercado.'  
 },  
 {  
 id: 1,  
 name: 'Ben Affleck',  
 image: '/assets/images/ben.jpg',  
 jobPosition: 'Chief Technology Officer',  
 jobCode: 'CTO',  
 featured: false,  
 jobDescription: 'El CTO, tiene la function de garantizar la calidad de los productos de software cuidando que los recursos utiliados ' +  
 'sean adecuados en terminos de tiempo, costo y recurso, de manera que la satisfaccion del cliente este asegurada.'  
 },  
 {  
 id: 2,  
 name: 'Gal Gadot',  
 image: '/assets/images/gal.jpg',  
 jobPosition: 'Software Lead',  
 jobCode: 'CTO',  
 featured: false,  
 jobDescription: 'Tiene la function de dirigir los projectos de software garantizando que se utilice adecuadamente la major ' +  
 'tecnologia disponible, y tambien debe garantizar la calidad de todo el equipo del projecto'  
 },  
 {  
 id: 3,  
 name: 'Amy Adams',  
 image: '/assets/images/amy.jpg',  
 jobPosition: 'Executive User Experience Chef',  
 jobCode: 'EC',  
 featured: false,  
 jobDescription: 'Debe garantizar que la experiencia del usuario al estar frente al producto de software sea lo mas agradable ' +  
 'possible, cuidando que la usabilidad del producto sea maximizada, debe diseñar los flujos proponiendo nuevos procesos que agilicen ' +  
 'los actuales antes de la implementacion de las nuevas funcionalidades.'  
 }  
];

Ahora asigne estas constantes a variables en state de MainComponent asi:

**this**.**state** = {  
 **items**: ITEMS,  
 **comments**: COMMENTS,   
 **employees**: EMPLOYEES  
};

En MainComponent pasar como props a el tag Home el item featured o destacado. Haga lo mismo para employees asi:

<**Home  
 item=**{**this**.**state**.**items**.filter(item => item.**featured**)[0]}   
 **employee=**{**this**.**state**.**employees**.filter(employee => employee.**featured**)[0]}  
/>

En HomeComponent remueva el tag h4 y en su lugar escriba un div con la clase container y dentro de esta, ponga 1 div de clase row y align-items-start. Dentro del ultimo div agregue un div de clase col-12 col-md m-1, dentro del cual pondrá un tag RenderCard que implementaremos luego para pintar un item. Aplique lo mismo para employees al final debe verse asi:

const *Home* = (props) => {  
 return (  
 <div className="container">  
 <div className="row align-items-start">  
 <div className="col-12 col-md m-1">  
 <RenderCard item={props.item}/>  
 </div>  
 <div className="col-12 col-md m-1">  
 <RenderCard item={props.employee}/>  
 </div>  
 </div>  
 </div>  
 );  
};

Verifique que su componente funcional de Home reciba props en la definición.

Escriba una función de nombre RenderCard que servirá para renderizar ítem y employee. Como parámetro de-estructure el props enviado extrayendo item, en la función retorne un Card elemto react asi:

**function** *RenderCard*({item}) {  
 **return**(  
 <**Card**>  
 <**CardImg src=**{item.**image**} **alt=**{item.**name**} />  
 <**CardBody**>  
 <**CardTitle**>{item.**name**}</**CardTitle**>  
 {item.**jobPosition** ? <**CardSubtitle**>{item.**jobPosition**}</**CardSubtitle**> : **null** }  
 <**CardText**>{item.**description**}</**CardText**>  
 </**CardBody**>  
 </**Card**>  
 );  
}

Verifique que ahora Home despliega el empleado e item destacados asi como todo sigue funcional.

Haga commit con el mensaje “Single Page Applications Part 1”.

# Single Page Applications parte 2

Con el uso de las rutas mostraremos el item seleccionado del catalogo, para esto modificaremos CatalogComponent y elimine el evento onClick del Card del componente funcional RenderCatalogItem.

Lo propio donde usamos el tag RenderCatalogItem ya no pasaremos la función onClick a props.

En la función RenderCatalogItem adicionaremos un tag <Link> de react-router-dom para envolver todo el contenido dentro del tag <Card>.

Adicionaremos la ruta de navegación cuando se haga click en el card mediante la propiedad to del tag <Link asi:

Observe que usamos back qupte para que la expresión JS se evalue.

<Card>  
 <Link to={`/catalog/${item.id}`}>

En el return del componente Catalog antes del row que pinta el catalogo adicione este código adicone un breadcrumb de reactstrap asi:

return (  
 <div className="container">  
 <div className="row">  
 <Breadcrumb>  
 <BreadcrumbItem><Link to="/home">Home</Link></BreadcrumbItem>  
 <BreadcrumbItem active>Catalog</BreadcrumbItem>  
 </Breadcrumb>  
 <div className="col-12">  
 <h3>Catalog</h3>  
 <hr />  
 </div>  
 </div>  
 <div className="row">  
 {catalog}

En MainComponent dentro del método render implemente una función llamada ItemWithId asi:

**const** *ItemWithId* = ({match}) => {  
 **return**(  
 <**ItemDetail item=**{**this**.**state**.**items**.filter((item) => item.**id** === *parseInt*(match.**params**.**itemId**,10))[0]}  
 **comments=**{**this**.**state**.**comments**.filter((comment) => comment.**itemId** === *parseInt*(match.**params**.**itemId**,10))} />  
 );  
};

En el Switch adicione una nueva ruta justo debajo de la correspondiente a catalog para un item de catalogo especifico asi:

<Route path='/catalog/:itemId' component={*ItemWithId*}/>

En ItemDetailComponent donde usamos el tag RenderComments pasaremos directamente por props los comments (ya no dentro de items), asi:

<RenderComments item={props.comments}/>

Antes del row que muestra ítems y comments adicionar el breadcrumb asi:

**const** *ItemDetail* = (props) => {  
 **return** (  
 <**div className="container"**>  
 <**div className="row"**>  
 <**Breadcrumb**>  
 <**BreadcrumbItem**><**Link to="/catalog"**>Catalog</**Link**></**BreadcrumbItem**>  
 <**BreadcrumbItem active**>{props.**item**.**name**}</**BreadcrumbItem**>  
 </**Breadcrumb**>  
 <**div className="col-12"**>  
 <**h3**>{props.**item**.**name**}</**h3**>  
 <**hr**/>  
 </**div**>  
 </**div**>  
 <**div className="row"**>  
 <**div className="col-12 col-md-5 m-1"**>  
 <**RenderItem item=**{props.**item**}/>

En ContactComponent también adicione un breadcrumb asi dentro el div container:

function *Contact*(props) {  
 return (  
 <div className="container">  
 <div className="row">  
 <Breadcrumb>  
 <BreadcrumbItem><Link to="/home">Home</Link></BreadcrumbItem>  
 <BreadcrumbItem active>Contact Us</BreadcrumbItem>  
 </Breadcrumb>  
 <div className="col-12">  
 <h3>Contact Us</h3>  
 <hr />  
 </div>  
 </div>  
 <div className="row row-content">

Haga commit con el mensaje” Single Page Applications Parte 2”

# Laboratorio 2:

En este laboratorio, continuará trabajando con la aplicación React agregando un nuevo componente llamado AboutComponent para proporcionar los detalles de los empleados, y luego integrará AboutComponent en la aplicación de una sola página.

El contenido de AboutComponent.js parcialmente debe ser este:

**import** React **from 'react'**;  
**import** { *Breadcrumb*, *BreadcrumbItem*, *Card*, *CardBody*, *CardHeader*, *Media* } **from 'reactstrap'**;  
**import** { Link } **from 'react-router-dom'**;  
  
**function** *About*(props) {  
  
 **const** employees = props.**employees**.map((employee) => {  
 **return** (  
 <**p**>Employee {employee.**name**}</**p**>  
 );  
 });  
  
 **return**(  
 <**div className="container"**>  
 <**div className="row"**>  
 <**Breadcrumb**>  
 <**BreadcrumbItem**><**Link to="/home"**>Home</**Link**></**BreadcrumbItem**>  
 <**BreadcrumbItem active**>About Us</**BreadcrumbItem**>  
 </**Breadcrumb**>  
 <**div className="col-12"**>  
 <**h3**>About Us</**h3**>  
 <**hr** />  
 </**div**>  
 </**div**>  
 <**div className="row row-content"**>  
 <**div className="col-12 col-md-6"**>  
 <**h2**>Our History</**h2**>  
 <**p**>Started in 2013, SSI quickly established itself as a safety icon par excellence in USA. With its  
 unique brand of world fusion safety that can be found nowhere else, it enjoys patronage from the  
 A-list clientele in USA. Featuring four of the best three-star Award CEO in the world, you never  
 know what will arrive on your mind the next time you visit us.</**p**>  
 <**p**>The enterprise traces its humble beginnings to <**em**>The BUILDING SEC CBA</**em**>, a successful chain  
 started by our CEO, Msc. Valentin Laime, that featured for the first time the world's best  
 enterprises in building safety.</**p**>  
 </**div**>  
 <**div className="col-12 col-md-5"**>  
 <**Card**>  
 <**CardHeader className="bg-primary text-white"**>Facts At a Glance</**CardHeader**>  
 <**CardBody**>  
 <**dl className="row p-1"**>  
 <**dt className="col-6"**>Started</**dt**>  
 <**dd className="col-6"**>1 Jan. 2013</**dd**>  
 <**dt className="col-6"**>Major Stake Holder</**dt**>  
 <**dd className="col-6"**>Jatun SRL.</**dd**>  
 <**dt className="col-6"**>Last Year's Turnover</**dt**>  
 <**dd className="col-6"**>$2,478,220</**dd**>  
 <**dt className="col-6"**>Employees</**dt**>  
 <**dd className="col-6"**>40</**dd**>  
 </**dl**>  
 </**CardBody**>  
 </**Card**>  
 </**div**>  
 <**div className="col-12"**>  
 <**Card**>  
 <**CardBody className="bg-faded"**>  
 <**blockquote className="blockquote"**>  
 <**p className="mb-0"**>You better cut the pizza in four pieces because  
 I'm not hungry enough to eat six.</**p**>  
 <**footer className="blockquote-footer"**>Yogi Berra,  
 <**cite title="Source Title"**>The Wit and Wisdom of Yogi Berra,  
 P. Pepe, Diversion Books, 2014</**cite**>  
 </**footer**>  
 </**blockquote**>  
 </**CardBody**>  
 </**Card**>  
 </**div**>  
 </**div**>  
 <**div className="row row-content"**>  
 <**div className="col-12"**>  
 <**h2**>Corporate employeeship</**h2**>  
 </**div**>  
 <**div className="col-12"**>  
 <**Media list**>  
 {employees}  
 </**Media**>  
 </**div**>  
 </**div**>  
 </**div**>  
 );  
}  
  
**export default** *About*;

|  |
| --- |
| /src/components/CatalogComponent.js  /src/components/ContactComponent.js  /src/components/FooterComponent.js  /src/components/GoogleMapsComponent.js  /src/components/HeaderComponent.js  /src/components/HomeComponent.js  /src/components/ItemDetailsComponent.js  /src/components/Main.js  /shared/comments.js  /shared/employees.js  /shared/ítems.js  /src/app.js |
| LIBRERIAS A INSTALAR  "dependencies": {  "bootstrap": "^4.3.1",  "bootstrap-social": "^5.1.1",  "font-awesome": "^4.7.0",  "react": "^16.9.0",  "react-dom": "^16.9.0",  "react-google-maps": "^9.4.5",  "react-router-dom": "^5.0.1",  "react-scripts": "3.1.1",  "reactstrap": "^8.0.1"  }, |
| import React, { Component } from 'react';  import {Card, CardImg, CardImgOverlay, CardTitle, Row} from 'reactstrap';  import ItemDetailsComponent from './ItemDetailsComponent';  import {Breadcrumb, BreadcrumbItem} from 'reactstrap'  import {Link} from 'react-router-dom';  class CatalogComponent extends Component {  constructor(props,context) {  super(props,context)  }  renderItem(item) {    if(item != null){  console.log(item.comments);  return(  <div className="col-12 col-md-5 mt-5">  <Card>  <CardImg width="100%" src={item.image} alt={item.name}/>  <CardImgOverlay>  <CardTitle>{item.name}</CardTitle>  </CardImgOverlay>  </Card>  </div>  );  }else{  return(<div></div>);  }  }  render() {  const catalog = this.props.items.map((item,index)=>{  return (  <div key={index} className="col-12 col-md-5 mt-5" >  <Card>  <Link to={`/catalog/${item.id}`}>  <CardImg width="100%" src={item.image} alt={item.name}/>  <CardImgOverlay>  <CardTitle>{item.name}</CardTitle>  </CardImgOverlay>  </Link>  </Card>  </div>  );  });  return (  <div className="container">  <Breadcrumb>  <BreadcrumbItem><Link to="/home">Home</Link></BreadcrumbItem>  <BreadcrumbItem active>Catalog</BreadcrumbItem>  </Breadcrumb>  <div className="row">  {catalog}  </div>  </div>  );  }  }  export default CatalogComponent; |
| import React, { Component } from 'react';  import {Breadcrumb, BreadcrumbItem} from 'reactstrap';  import {Link} from 'react-router-dom';  import GoogleMapsComponent from './GoogleMapsComponent';  function ContactComponent(props) {  return(  <div className="container">  <div className="row">  <Breadcrumb>  <BreadcrumbItem><Link to="/home">Home</Link></BreadcrumbItem>  <BreadcrumbItem active>Contact Us</BreadcrumbItem>  </Breadcrumb>  <div className="col-12">  <h3>Contact Us</h3>  <hr />  </div>  </div>  <div className="row row-content">  <div className="col-12">  <h3>Location Information</h3>    </div>  <div className="col-12 col-sm-4 offset-sm-1">  <h5>Our Address</h5>  <address>  121, Dr. Drive<br />  San Jose, California<br />  MOUNTAIN VIEW<br />  <i className="fa fa-phone"></i>: +1 1234 5678<br />  <i className="fa fa-fax"></i>: +1 8765 4321<br />  <i className="fa fa-envelope"></i>: <a href="mailto:ssi@food.net">ssi@food.net</a>  </address>  </div>  <div className="col-12 col-sm-6 offset-sm-1">  <h5>Map of our Location</h5>  <GoogleMapsComponent  isMarkerShown  defaultCenter={{lat:-17.3870028,lng:-66.1930615}}  googleMapURL="https://maps.googleapis.com/maps/api/js?v=3.exp&libraries=geometry,drawing,places"  loadingElement={<div style={{ height: '300px' }} />}  containerElement={<div style={{ height: '400px' }} />}  mapElement={<div style={{ height: '300px' }} />}  />  </div>  <div className="col-12 col-sm-11 offset-sm-1">  <div className="btn-group" role="group">  <a role="button" className="btn btn-primary" href="tel:+112345678"><i className="fa fa-phone"></i> Call</a>  <a role="button" className="btn btn-info"><i className="fa fa-skype"></i> Skype</a>  <a role="button" className="btn btn-success" href="mailto:ssi@security.net"><i className="fa fa-envelope-o"></i> Email</a>  </div>  </div>  </div>  <div>      </div>  </div>  );  }  export default ContactComponent; |
| import React from 'react';  import {Link} from 'react-router-dom'  function FooterComponent(props) {  return (  <div className="footer mt-5">  <div className="container">  <div className="row justify-content-center">  <div className="col-4 offset-1 col-sm-2">  <h5>Links</h5>  <ul className="list-unstyled">  <li><Link to='/home'>Home</Link></li>  <li><Link to='/aboutus'>About Us</Link></li>  <li><Link to='/catalog'>Catalog</Link></li>  <li><Link to='/contact'>Contact Us</Link></li>  </ul>  </div>  <div className="col-7 col-sm-5">  <h5>Our Address</h5>  <address>  121, Dr. Drive<br/>  San Jose, California<br/>  MOUNTAIN VIEW<br/>  <i className="fa fa-phone fa-lg"></i>: +1 1234 5678<br/>  <i className="fa fa-fax fa-lg"></i>: +1 8765 4321<br/>  <i className="fa fa-envelope fa-lg"></i>: <a href="mailto:ssi-info@ssi.net">  ssi-info@ssi.net</a>  </address>  </div>  <div className="col-12 col-sm-4 align-self-center">  <div className="text-center">  <a className="btn btn-social-icon btn-google" href="http://google.com/+"><i  className="fa fa-google-plus"></i></a>  <a className="btn btn-social-icon btn-facebook"  href="http://www.facebook.com/profile.php?id="><i className="fa fa-facebook"></i></a>  <a className="btn btn-social-icon btn-linkedin" href="http://www.linkedin.com/in/"><i  className="fa fa-linkedin"></i></a>  <a className="btn btn-social-icon btn-twitter" href="http://twitter.com/"><i  className="fa fa-twitter"></i></a>  <a className="btn btn-social-icon btn-google" href="http://youtube.com/"><i  className="fa fa-youtube"></i></a>  <a className="btn btn-social-icon" href="mailto:"><i className="fa fa-envelope-o"></i></a>  </div>  </div>  </div>  <div className="row justify-content-center">  <div className="col-auto">  <p>© Copyright 2019 SSI</p>  </div>  </div>  </div>  </div>  )  }  export default FooterComponent; |
| import React from 'react';  import { withScriptjs, withGoogleMap, GoogleMap, Marker } from "react-google-maps";  const GoogleMapsComponent = withScriptjs(withGoogleMap((props) =>{  const dc = (props.defaultCenter?props.defaultCenter:{ lat: -34.397, lng: 150.644 })  return(  <GoogleMap  defaultZoom={8}  defaultCenter={dc}  >  {props.isMarkerShown && <Marker position={dc} />}  </GoogleMap>  );  }));    export default GoogleMapsComponent; |
| import React, {Component} from 'react';  import {Navbar, NavbarToggler, Collapse, NavbarBrand, Jumbotron,Nav,NavItem} from 'reactstrap';  import {NavLink} from 'react-router-dom'  class HeaderComponent extends Component {  constructor(context,props){  super(context, props)  this.state={  isNavOpen:false  }  this.toggleNav = this.toggleNav.bind(this);  }  toggleNav() {  this.setState({  isNavOpen: !this.state.isNavOpen  });  }    render() {  return (  /\*agrupa varios elementos sin necesidad de un div su forma abreviada es <></>\*/  <React.Fragment>  <Navbar dark expand="md">  <NavbarToggler onClick={this.toggleNav}/>  <NavbarBrand href="/">  <img src="assets/images/safetylogo.png" alt="SSI" width="40" height="40"/>  SSI  </NavbarBrand>  <Collapse isOpen={this.state.isNavOpen} navbar>  <Nav navbar>  <NavItem>  <NavLink className="nav-link" to='/home' active><span className="fa fa-home fa-lg"></span> Home</NavLink>  </NavItem>  <NavItem>  <NavLink className="nav-link" to='/aboutus'><span className="fa fa-info fa-lg"></span> About Us</NavLink>  </NavItem>  <NavItem>  <NavLink className="nav-link" to='/catalog'><span className="fa fa-list fa-lg"></span> Catalog</NavLink>  </NavItem>  <NavItem>  <NavLink className="nav-link" to='/contact'><span className="fa fa-address-card fa-lg"></span> Contact Us</NavLink>  </NavItem>  </Nav>  </Collapse>  </Navbar>  <Jumbotron>  <div className="container ">  <div className="row row-header">  <div className="col-12 col-sm-6">  <h1>Sistema de Seguridad industrial</h1>  <p>We take inspiration from the World's best safety practices, and create a unique  fusion experience. Our awesome safety protection items will tickle your safety  vision!</p>  </div>  <img src="assets/images/safetylogo.png" alt="SSI" style={{marginLeft:'100px'}} width="300" height="300"/>  </div>  </div>  </Jumbotron>  </React.Fragment>  );  }  }  export default HeaderComponent; |
| import React, { Component } from 'react';  import {Card, CardText, CardImg, CardTitle, CardBody, CardSubtitle} from 'reactstrap';  class HomeComponent extends Component {  constructor(context, props){  super(context, props);  }  renderCard(item) {  if(item != null){  console.log(item);  return(  <Card>  <CardImg src={item.image} alt={item.name} />  <CardBody>  <CardTitle>{item.name}</CardTitle>  {item.jobPosition ? <CardSubtitle>{item.jobPosition}</CardSubtitle> : null }  <CardText>{item.description}</CardText>  </CardBody>  </Card>  );  }else{  return (<div></div>);  }    }  render() {  return (  <div className="container">  <div className="row align-items-start">  <div className="col-12 col-md m-1">  {this.renderCard(this.props.employee?this.props.employee:null)}  </div>  <div className="col-12 col-md m-1">  {this.renderCard(this.props.item?this.props.item:null)}  </div>  </div>  </div>  );  }  }  export default HomeComponent; |
| import React, { Component } from 'react';  import {Card, CardImg, CardTitle, CardImgOverlay, Breadcrumb, BreadcrumbItem} from 'reactstrap';  import {Link} from 'react-router-dom';  class ItemDetailsComponent extends Component {  constructor(props, context){  super(props, context)  }  renderItem(item) {    if(item != null){  console.log(item.comments);  return(  <Card>  <CardImg width="100%" src={item.image} alt={item.name}/>  <CardImgOverlay>  <CardTitle>{item.name}</CardTitle>  </CardImgOverlay>  </Card>  );  }else{  return(<div></div>);  }  }  renderComments(comments) {    if(comments != null){  const resume = comments.map((c,index)=>{  return(  <li key={index}>  {c.comment}-{c.author}  </li>  );  });  return(  <ul>  {resume}  </ul>  );  }else{  return(<div></div>);  }  }  render() {  return (  <div className="container">  <Breadcrumb>  <BreadcrumbItem><Link to="/catalog">Catalog</Link></BreadcrumbItem>  <BreadcrumbItem active>{this.props.item.name}</BreadcrumbItem>  </Breadcrumb>  <div className="col-12 col-md-5">  {this.renderItem(this.props.item?this.props.item:null)}  </div>  <div className="col-12 col-md-5 ">  {this.renderComments(this.props.comments?this.props.comments:null)}  </div>  </div>  );  }  }  export default ItemDetailsComponent; |
| import React, { Component } from 'react';  import HeaderComponent from './HeaderComponent'  import FooterComponent from './FooterComponent'  import CatalogComponent from './CatalogComponent'  import {ITEMS} from '../shared/items';  import {COMMENTS} from '../shared/comments';  import {EMPLOYEES} from '../shared/employees'  import HomeComponent from './HomeComponent';  import ContactComponent from './ContactComponent';  import {Switch,Route,Redirect} from 'react-router-dom';  import ItemDetailsComponent from './ItemDetailsComponent';  class Main extends Component {  constructor(props,context) {  super(props,context)  this.state={  items: ITEMS,  comments: COMMENTS,  employees: EMPLOYEES  }  }  render() {  const HomeComponentRoute = () =>{    const employee = this.state.employees.filter(item=> item.featured)[0];  const item = this.state.items.filter(element=> element.featured)[0];  return (  <HomeComponent employee={employee} item={item}></HomeComponent>  );  }  const CatalogComponentRoute = () => {  return (  <CatalogComponent items={this.state.items}></CatalogComponent>  );  };  const ItemDetailComponentRoute = ({match}) =>{  const itemSelected = this.state.items.filter(element => element.id === parseInt(match.params.id))[0];  const commentsSelected = this.state.comments.filter(element => element.itemId === parseInt(match.params.id));  return (<ItemDetailsComponent item={itemSelected} comments={commentsSelected}></ItemDetailsComponent>);  }  return (  <div>  <HeaderComponent></HeaderComponent>  <Switch>  <Route path="/home" component={HomeComponentRoute}></Route>  <Route exact path="/catalog" component={CatalogComponentRoute}></Route>  <Route exact path="/catalog/:id" component={ItemDetailComponentRoute}></Route>  <Route path="/contact" component={ContactComponent}></Route>  <Redirect to="/home"></Redirect>  </Switch>  <FooterComponent></FooterComponent>  </div>  );  }  }  export default Main; |
| import React from 'react';  import './App.css';  import { ITEMS } from './shared/items';  import Main from './components/Main';  import {BrowserRouter} from 'react-router-dom'  class App extends React.Component {  constructor() {  super();  this.state = {  items: ITEMS  }  }  render(){  return (  <div className="App">  <BrowserRouter>  <Main></Main>  </BrowserRouter>  </div>  );  }  }  export default App; |
| export const COMMENTS =  [  {  id: 0,  itemId: 0,  rating: 5,  comment: "Imagine all safety PPE in SSI!",  author: "Mr T",  date: "2012-10-16T17:57:28.556094Z"  },  {  id: 1,  itemId: 0,  rating: 4,  comment: "Gives me a lot of confort, I wish I could get my mother-in-law to have it!",  author: "Tania Barrancos",  date: "2014-09-05T17:57:28.556094Z"  },  {  id: 2,  itemId: 0,  rating: 3,  comment: "Use it, just use it!",  author: "Michael Rodriguez",  date: "2015-02-13T17:57:28.556094Z"  },  {  id: 3,  itemId: 0,  rating: 4,  comment: "Ultimate, Reaching for the stars!",  author: "Diana Cardozo",  date: "2013-12-02T17:57:28.556094Z"  },  {  id: 4,  itemId: 0,  rating: 2,  comment: "It's your birthday, we're gonna party!",  author: "25 Cent",  date: "2011-12-02T17:57:28.556094Z"  },  {  id: 5,  itemId: 1,  rating: 5,  comment: "Imagine all safety PPE in SSI!",  author: "Mr T",  date: "2012-10-16T17:57:28.556094Z"  },  {  id: 6,  itemId: 1,  rating: 4,  comment: "Gives me a lot of confort, I wish I could get my mother-in-law to have it!",  author: "Tania Barrancos",  date: "2014-09-05T17:57:28.556094Z"  },  {  id: 7,  itemId: 1,  rating: 3,  comment: "Use it, just use it!",  author: "Michael Rodriguez",  date: "2015-02-13T17:57:28.556094Z"  },  {  id: 8,  itemId: 1,  rating: 4,  comment: "Ultimate, Reaching for the stars!",  author: "Diana Cardozo",  date: "2013-12-02T17:57:28.556094Z"  },  {  id: 9,  itemId: 1,  rating: 2,  comment: "It's your birthday, we're gonna party!",  author: "25 Cent",  date: "2011-12-02T17:57:28.556094Z"  },  {  id: 10,  itemId: 2,  rating: 5,  comment: "Imagine all safety PPE in SSI!",  author: "Mr T",  date: "2012-10-16T17:57:28.556094Z"  },  {  id: 11,  itemId: 2,  rating: 4,  comment: "Gives me a lot of confort, I wish I could get my mother-in-law to have it!",  author: "Tania Barrancos",  date: "2014-09-05T17:57:28.556094Z"  },  {  id: 12,  itemId: 2,  rating: 3,  comment: "Use it, just use it!",  author: "Michael Rodriguez",  date: "2015-02-13T17:57:28.556094Z"  },  {  id: 13,  itemId: 2,  rating: 4,  comment: "Ultimate, Reaching for the stars!",  author: "Diana Cardozo",  date: "2013-12-02T17:57:28.556094Z"  },  {  id: 14,  itemId: 2,  rating: 2,  comment: "It's your birthday, we're gonna party!",  author: "25 Cent",  date: "2011-12-02T17:57:28.556094Z"  },  {  id: 15,  itemId: 3,  rating: 5,  comment: "Imagine all safety PPE in SSI!",  author: "Mr T",  date: "2012-10-16T17:57:28.556094Z"  },  {  id: 16,  itemId: 3,  rating: 4,  comment: "Gives me a lot of confort, I wish I could get my mother-in-law to have it!",  author: "Tania Barrancos",  date: "2014-09-05T17:57:28.556094Z"  },  {  id: 17,  itemId: 3,  rating: 3,  comment: "Use it, just use it!",  author: "Michael Rodriguez",  date: "2015-02-13T17:57:28.556094Z"  },  {  id: 18,  itemId: 3,  rating: 4,  comment: "Ultimate, Reaching for the stars!",  author: "Diana Cardozo",  date: "2013-12-02T17:57:28.556094Z"  },  {  id: 19,  itemId: 3,  rating: 2,  comment: "It's your birthday, we're gonna party!",  author: "25 Cent",  date: "2011-12-02T17:57:28.556094Z"  }  ]; |
| export const EMPLOYEES = [  {  id: 0,  name: 'Henry Cavill',  image: '/assets/images/henry.jpg',  jobPosition: 'Chief Enterprise Officer',  jobCode: 'CEO',  featured: false,  jobDescription: 'El CEO, esta encargado de dirigir la empresa para conseguir los objetivos alineados con las estrategias de la ' +  'empresa. Su rol fundamental es marcar las estrategias a seguir cuidando que los indicadores relacionados se cumplan y de ser ' +  'necesario proponer nuevas estrategias que permitan a la empresa adaptarse al mercado.'  },  {  id: 1,  name: 'Ben Affleck',  image: '/assets/images/ben.jpg',  jobPosition: 'Chief Technology Officer',  jobCode: 'CTO',  featured: false,  jobDescription: 'El CTO, tiene la function de garantizar la calidad de los productos de software cuidando que los recursos utiliados ' +  'sean adecuados en terminos de tiempo, costo y recurso, de manera que la satisfaccion del cliente este asegurada.'  },  {  id: 2,  name: 'Gal Gadot',  image: '/assets/images/gal.jpg',  jobPosition: 'Software Lead',  jobCode: 'CTO',  featured: true,  jobDescription: 'Tiene la function de dirigir los projectos de software garantizando que se utilice adecuadamente la major ' +  'tecnologia disponible, y tambien debe garantizar la calidad de todo el equipo del projecto'  },  {  id: 3,  name: 'Amy Adams',  image: '/assets/images/amy.jpg',  jobPosition: 'Executive User Experience Chef',  jobCode: 'EC',  featured: false,  jobDescription: 'Debe garantizar que la experiencia del usuario al estar frente al producto de software sea lo mas agradable ' +  'possible, cuidando que la usabilidad del producto sea maximizada, debe diseñar los flujos proponiendo nuevos procesos que agilicen ' +  'los actuales antes de la implementacion de las nuevas funcionalidades.'  }  ]; |
| export const ITEMS = [  {  id: 0,  name: 'Helmet',  image: '/assets/images/helmet.jpg',  category: 'safety',  label: 'ultra safe',  price: '4.99',  featured: true,  description: 'Un casco de seguridad útil para proteger en ámbitos como la construcción, fabricas y otros',  },  {  id: 1,  name: 'Audio protector',  image: '/assets/images/audio.jpg',  category: 'safety',  label: 'noise filter',  price: '12.99',  featured: false,  description: 'Protege de ruidos auditivos por encima de los decibeles considerados aceptables',  },  {  id: 2,  name: 'Glasses',  image: '/assets/images/glasses.jpg',  category: 'safety',  label: 'New technology',  price: '31.99',  featured: false,  description: 'Permite filtrar rayos ultravioleta, material anti rayas, anti caidas',  },  {  id: 3,  name: 'Gloves',  image: '/assets/images/gloves.jpg',  category: 'safety',  label: '',  price: '12.99',  featured: false,  description: 'Hechos de algodón con goma antideslizante, especial para industria química.',  }  ]; |

## Tarea 1

En esta tarea, integrará AboutComponent en la aplicación de una sola página:

* Primero cree el archivo AboutComponent.js en la carpeta componentes.
* Actualice el archivo MainComponent.js para integrar AboutComponent en la aplicación de una sola página. Debería poder navegar a AboutComponent haciendo clic en los enlaces en la barra de navegación (del header) y en el pie de página.

## Tarea 2

En esta tarea, implementará un nuevo componente funcional llamado <RenderEmployee> a través de la función RenderEmployee():

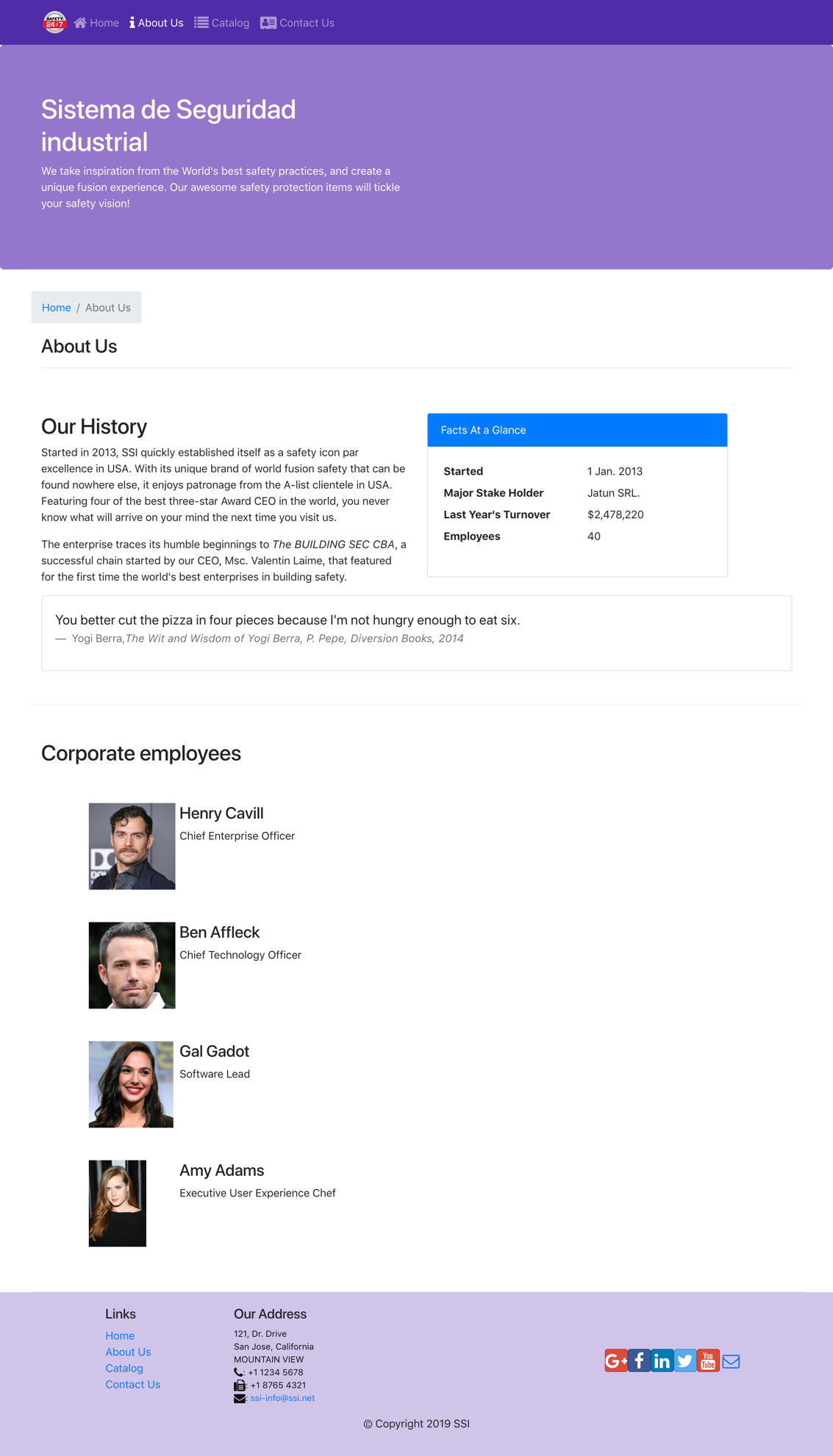
* La función RenderEmployee() recibe los detalles de un empleado como parámetro,
* Implemente la función RenderEmployee() para usar la información del Empleado recibido como parámetro y renderizar la información del empleado como se muestra en la captura de pantalla siguiente utilizando el componente reactstrap <Media>. Recuerde cómo usamos el componente <Media> las primeras clases para representar los detalles de un item en el CatalogComponent.
* Esto hará que el componente <RenderEmployee> esté disponible para su uso dentro de AboutComponent.

## Tarea 3

En esta tarea, utilizará el componente <RenderEmployee> creado en la tarea anterior para generar una lista de empleados dentro de su componente:

* Actualice la variable de empleados dentro de la función AboutComponent() para hacer uso del componente <RenderEmployee> para representar la lista de empleados.

El resultado de las tres tareas debe verse asi:



## Rubrica Laboratorio 2

Estudiante que fue evaluado: …………………………….…………………………….

Estudiante que realizo la evaluación: …………………………….…………………………….

|  |  |  |
| --- | --- | --- |
| Actividades | Puntaje | Cumple |
| T1.1: El archivo AboutComponent.js se ha descargado e integrado en nuestra aplicación React. | 20 |  |
| T1.2: La aplicación React se ha actualizado correctamente para permitir la navegación a la página About us de nuestra aplicación. | 20 |  |
| T2.1: Implementó un nuevo componente funcional <Renderemployee> en su aplicación. | 20 |  |
| T2.2 Usó el componente reactstrap <Media> para representar los detalles de un empleado. | 20 |  |
| T3: Se actualizó la variable de empleados dentro de la función AboutComponent() para hacer uso del componente <RenderEmployee> para representar la lista de empleados. | 20 |  |
| PUNTUACION TOTAL |  |  |

Envie la nota de si compañero a edsonariel@gmail.com