Herramientas de Programación 2

Tema Nº9:REACT ROUTER EN UN PROYECTO I

Indicador de logro Nº9:Crea páginas web dinámicas usando los fundamentos de React aprendidos en clase.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº9:**

REACT ROUTER EN UN PROYECTO I

**Subtema 9.1:**

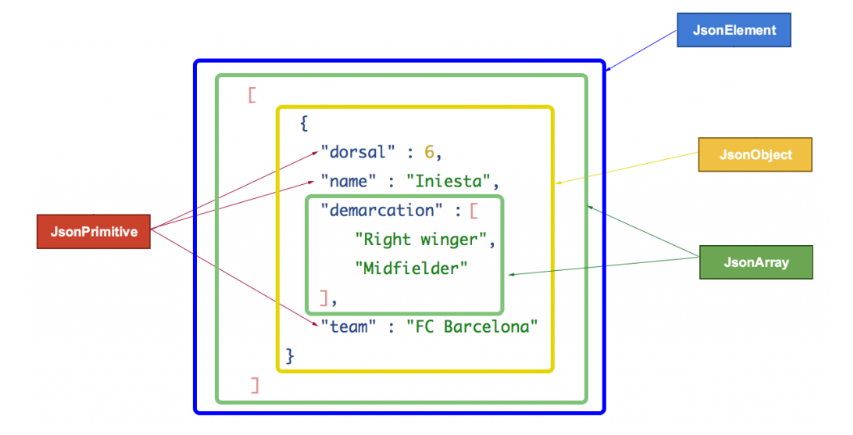
JSON

JSON es un formato ampliamente usado en distintos lenguajes del lado del servidor para mandar datos al Front End y poderlos procesar y mostrar al usuario.

Ventajas de Json sobre XML:

* Formato sumamente simple
* Velocidad de procesamiento alta
* Archivos de menor tamaño

Por otro lado, JSON cuenta con un formato más simple y fácil de procesar, por lo que es ideal para dispositivos de bajo nivel de procesamiento, como los teléfonos móviles, los cuales son beneficiados ya que se requieren de menos consumo de datos para descargarlos y un consumo menor de batería al procesarlos, desde luego que los teléfonos modernos ya son capaces de procesar información mucho más rápido pero no podemos confiarnos en todo mundo tendrá uno de esos.

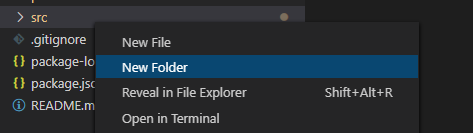


Paso 1. Crear datos en formato Json: lista de Categorías.

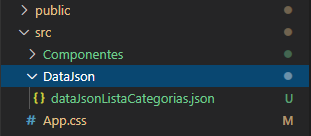
1. Abrimos en Visual Studio Code el proyecto React del taller de la semana anterior.
2. Ejecutamos la aplicación para verificar que no existan errores.



1. Creamos la carpeta *DataJson* dento de la carpeta *src*.



1. Dentro de la carpeta *DataJson* creamos el archivo *dataJsonListaCategorias.json*



1. Agregamos datos en formato Json dentro del archivo *dataJsonListaCategorias.json* de acuerdo a:

[

    {

        "id": 3,

        "nombre": "ZAPATILLAS",

        "activo": true

    },

    {

        "id": 5,

        "nombre": "POLOS",

        "activo": true

    },

    {

        "id": 7,

        "nombre": "CASACAS",

        "activo": true

    },

    {

        "id": 10,

        "nombre": "BUZOS",

        "activo": true

    },

    {

        "id": 12,

        "nombre": "ACCESORIOS",

        "activo": true

    }

]

Paso 2. Mostrar datos Json usando map.

1. Abrimos el archivo *CompListaCategorias.js* y modificamos su codigo de acuerdo a:

import React, {Component} from 'react';

import dataJsonListaCategorias from '../DataJson/dataJsonListaCategorias.json';

class CompListaCategorias extends Component {

  render(){

    return  (

        <div className="col-3">

            <p className="h5 text-secondary">Categorías ({dataJsonListaCategorias.length + 1})</p>

            <div className="card">

                <ul className="list-group list-group-flush">

                  <li className="list-group-item"><a href="productos/destacados" className="text-secondary">DESTACADOS</a></li>

                  {

                    dataJsonListaCategorias.map((categoria, indice) => {

                      var url = "productos/porcategoria/" + categoria.id;

                    return <li className="list-group-item" key={categoria.id}><a href={url} className="text-secondary">{categoria.nombre}</a></li>

                    })

                  }

                </ul>

            </div>

        </div>

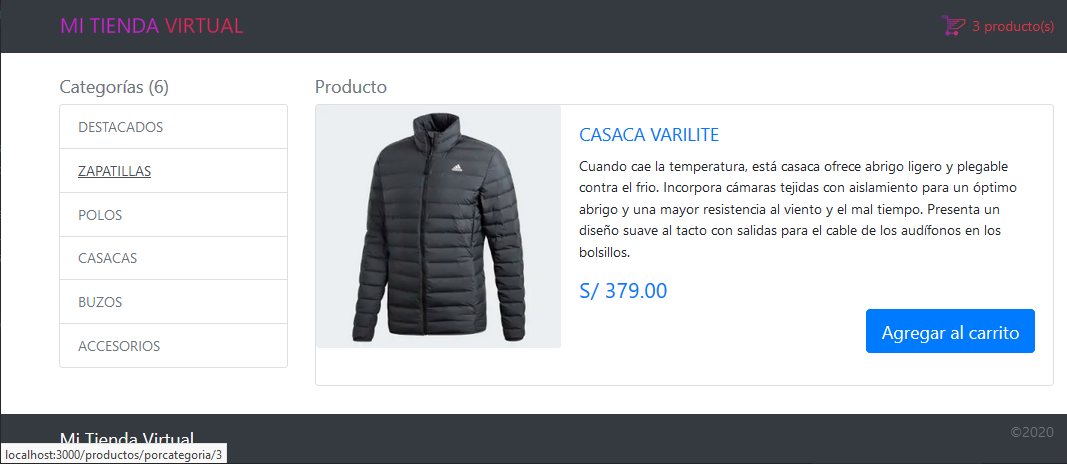
    )

  }

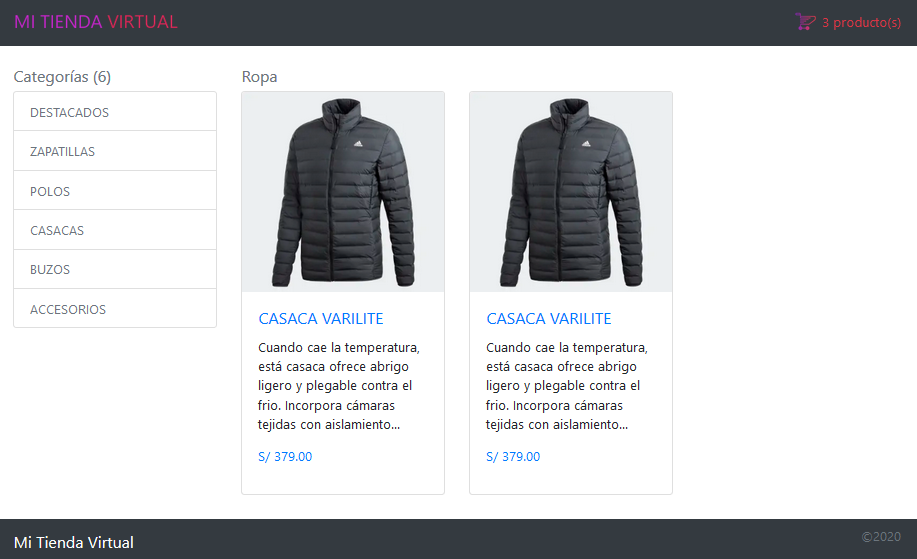
}

export default CompListaCategorias;

1. Observamos el explorador web para ver los cambios (F5 si fuera necesario)



Paso3. Crear los componentes para mostrar la página con la lista de los productos.



1. Creamos el componente *CompProductoResumen.js*, agregamos codigo jsx de acuerdo a:

import React, {Component} from 'react';

class CompProductoResumen extends Component {

  render(){

    return  <div class="col mb-4">

                <div class="card h-100">

                    <a href="#">

                        <img src="imagenes/producto1.png" class="card-img-top" alt="..."/>

                    </a>

                    <div class="card-body">

                        <a class="text-primary" href="#">

                            <h5 class="card-title">CASACA VARILITE</h5>

                        </a>

                        <p class="card-text">Cuando cae la temperatura, está casaca ofrece abrigo ligero y plegable contra el frio. Incorpora cámaras tejidas con aislamiento...</p>

                        <p class="text-primary">S/ 379.00</p>

                    </div>

                </div>

            </div>

  }

}

export default CompProductoResumen;

1. Creamos el componente *CompListaProductos.js*, agregamos codigo jsx de acuerdo a:

import React, {Component} from 'react';

import CompProductoResumen from './CompProductoResumen';

class CompListaProductos extends Component {

  render(){

    return  <div class="col-9">

        <p class="h5 text-secondary">Ropa</p>

                    <div class="row row-cols-1 row-cols-md-3">

                        <CompProductoResumen />

                        <CompProductoResumen />

                    </div>

        </div>

  }

}

export default CompListaProductos;

1. Creamos el componente *PageListaProductos.js*, agregamos codigo jsx de acuerdo a:

import React, {Component} from 'react';

import CompCabecera from './CompCabecera';

import CompListaCategorias from './CompListaCategorias';

import CompPiePagina from './CompPiePagina';

import CompListaProductos from './CompListaProductos';

class PageListaProductos extends Component {

  render(){

    return  <div >

                <CompCabecera />

                <main role="main" className="container">

                    <div className="row">

                        <CompListaCategorias />

                        <CompListaProductos />

                    </div>

                </main>

                <CompPiePagina />

            </div>

  }

}

export default PageListaProductos;

1. Modificamos el codigo del archivo *App.js* de acuerdo a:

import React, {Component} from 'react';

import './App.css';

import PageListaProductos from './Componentes/PageListaProductos';

class App extends Component {

  render(){

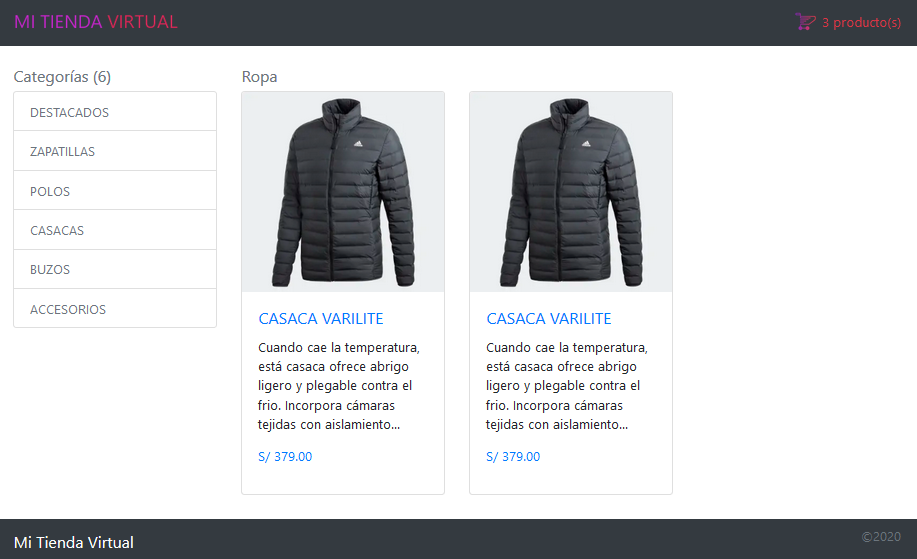
    return  <PageListaProductos />

  }

}

export default App;

1. Validamos los cambios en el explorador web.



Paso 4. Crear datos en formato Json: lista de Productos.

1. Dentro de la carpeta *DataJson* creamos el archivo *dataJsonListaProductos.json* con la siguiente información:

[

    {

        "id": 12,

        "nombre": "CASACA VARILITE",

        "descripcion": "Cuando cae la temperatura, está casaca ofrece abrigo ligero y plegable contra el frio. Incorpora cámaras tejidas con aislamiento...",

        "precio": 379.00,

        "url": "producto1.png",

        "desctacado": true,

        "activo": true

    },

    {

        "id": 5,

        "nombre": "POLO DOODLE BADGE",

        "descripcion": "Con este polo adidas, recordarás los días en que cubrías tus cuadernos con dibujos....",

        "precio": 89.00,

        "url": "producto2.png",

        "desctacado": true,

        "activo": true

    },

    {

        "id": 31,

        "nombre": "POLERA BELLISTA",

        "descripcion": "Deja que tu ropa hable por ti. Esta polera actualiza el legado deportivo de adidas...",

        "precio": 249.00,

        "url": "producto3.png",

        "desctacado": true,

        "activo": true

    },

    {

        "id": 13,

        "nombre": "POLO NIÑA ORIGINALS",

        "descripcion": "Prepárate para el verano. Este polo luce un gran logo del Trifolio que resalta sobre detalles florales coloridos...",

        "precio": 79.00,

        "url": "producto4.png",

        "desctacado": true,

        "activo": true

    }

]

Paso 5. Enviar información a través de componentes.

1. Abrimos el archivo *PageListaProductos.js* y modificamos su codigo de acuerdo a:

import React, {Component} from 'react';

import CompCabecera from './CompCabecera';

import CompListaCategorias from './CompListaCategorias';

import CompPiePagina from './CompPiePagina';

import CompListaProductos from './CompListaProductos';

import dataJsonListaProductos from '../DataJson/dataJsonListaProductos.json';

class PageListaProductos extends Component {

  render(){

    return  <div >

                <CompCabecera />

                <main role="main" className="container">

                    <div className="row">

                        <CompListaCategorias />

                        <CompListaProductos pListaDeProductos={dataJsonListaProductos}/>

                    </div>

                </main>

                <CompPiePagina />

            </div>

  }

}

export default PageListaProductos;

1. Abrimos el archivo *CompListaproductos.js* y modificamos su codigo de acuerdo a:

import React, {Component} from 'react';

import CompProductoResumen from './CompProductoResumen';

class CompListaProductos extends Component {

  render(){

    return  <div class="col-9">

        <p class="h5 text-secondary">Ropa</p>

                    <div class="row row-cols-1 row-cols-md-3">

                        {this.props.pListaDeProductos.map( lp => <CompProductoResumen pDatosDelProducto={lp} key={lp.id}/>)}

                    </div>

        </div>

  }

}

export default CompListaProductos;

1. Abrimos el archivo *CompProductoResumen.js* y modificamos su codigo de acuerdo a:

import React, {Component} from 'react';

class CompProductoResumen extends Component {

  render(){

    var urlImagen = "imagenes/" + this.props.pDatosDelProducto.url;

    var urlProducto = "producto/" + this.props.pDatosDelProducto.id;

    return  <div class="col mb-4">

                <div class="card h-100">

                    <a href="producto/1">

                        <img src={urlImagen} class="card-img-top" alt="..."/>

                    </a>

                    <div class="card-body">

                        <a class="text-primary" href={urlProducto}>

                            <h5 class="card-title">{this.props.pDatosDelProducto.nombre}</h5>

                        </a>

                        <p class="card-text">{this.props.pDatosDelProducto.descripcion}</p>

                        <p class="text-primary">S/ {this.props.pDatosDelProducto.precio}</p>

                    </div>

                </div>

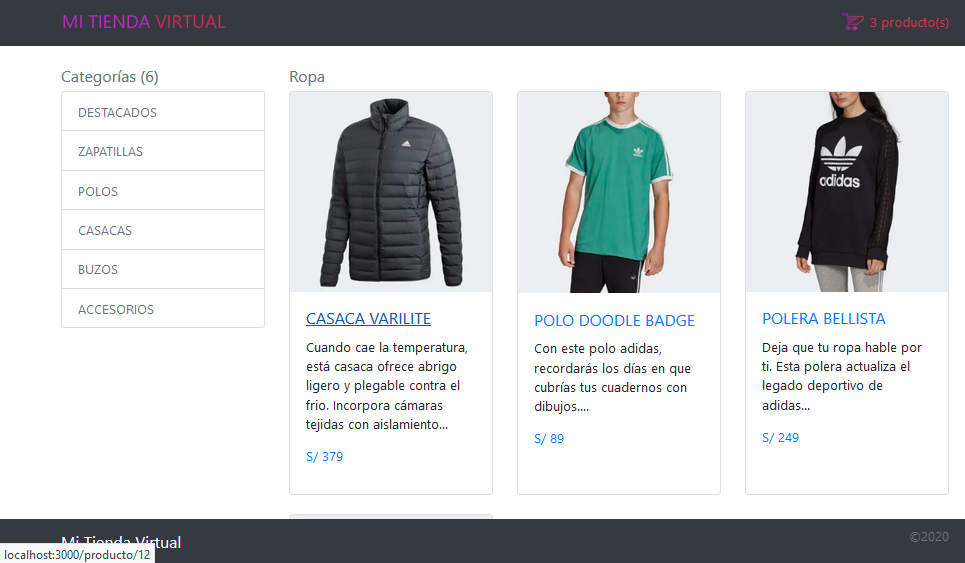
            </div>

  }

}

export default CompProductoResumen;

1. Validamos los cambios en el explorador web.



ACTIVIDAD VIRTUAL

Ingresa a la plataforma virtual, luego desarrolla la siguiente actividad propuesta:

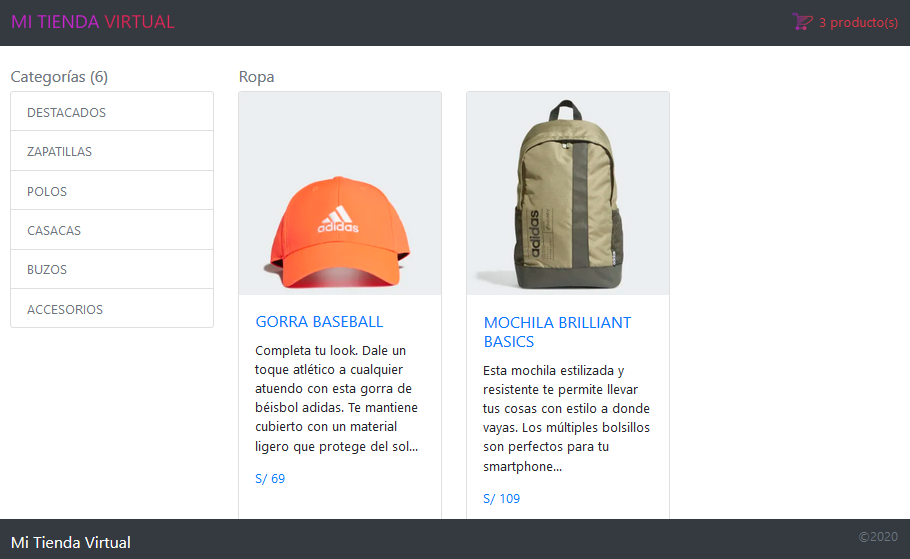
1. CUESTIONARIO TÉCNICO
   * 1. Agregar un control en el componente adecuado que permita seleccionar la cantidad del producto que debe agregarse al pedido.



* + 1. Agregar en el archivo *dataJsonListaProductos.json* los datos de dos productos que tengan el atributo *“activo”: false*.

Agregar también las imágenes de los dos productos en la carpeta *imagenes*.

* + 1. Modificar el código del componente adecuado de tal forma que se muestren solo los productos que tengan el atributo *“activo”: false*



* + 1. Modificar el código del componente adecuado de tal forma que se muestren solo los productos que tengan el atributo *“destacado”: true*
    2. Investigar la forma de obtener información desde un servicio API RESTful.

1. CONCLUSIONES DE LA EXPERIENCIA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ACTIVIDAD** **VIRTUAL**:

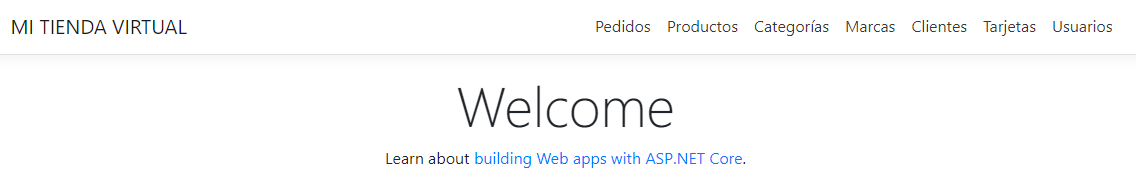
Ingresa a la plataforma virtual. Revisa y analiza el tema desarrollado en esta sesión, luego desarrolla las siguientes actividades propuestas:

1. CUESTIONARIO TÉCNICO
   * 1. Crear con Scaffold Controladores y vistas para el mantenimiento (CRUD) de las siguientes entidades (Tablas):

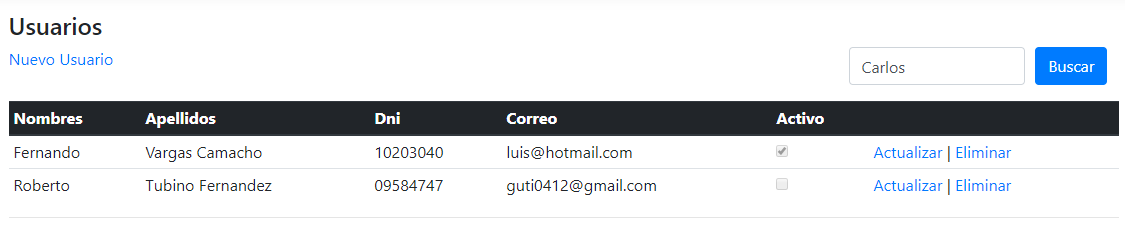
* Marca
* Categoría
* Producto
  + 1. Crear con Scaffold Controladores y vistas para el mantenimiento (CRUD) de las siguientes entidades (Tablas):
* Cliente
* Tarjeta

Estas dos entidades deben ser de solo lectura, esto es, no debe permitir crear, actualizar ni eliminar registros.

* + 1. Para los nuevos mantenimientos, desarrollar las personalizaciones de forma análoga al CRUD de Usuarios.
    2. Agregar en el menú superior links que permitan acceder a los nuevos mantenimientos. Además, alinear el menú a la derecha.



* + 1. Investigar la forma de agregar la funcionalidad de búsqueda en la lista de Usuarios, debe buscar por Nombres y Apellidos.



1. CONCLUSIONES DE LA EXPERIENCIA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Un motor paso a paso siempre debemos considerar su corriente de operación y los grados, como indica el ejemplo de la siguiente placa característica de la imagen mostrada:

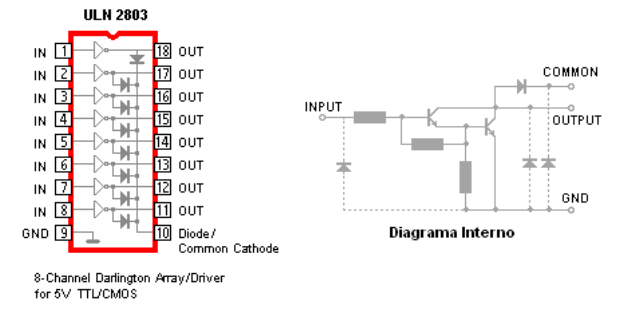


**Subtema 1.2:**

Transistores de potencia

Para operaciones con motores de bajo consumo tenemos circuitos integrados como el circuito ULN2803, el cual es un driver con transistores darlintong , con soporte de hasta 500 mA.

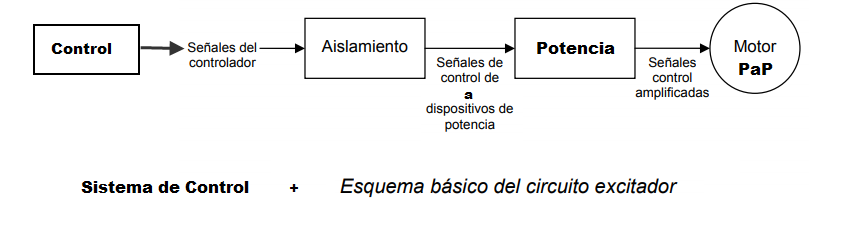
Por lo tanto si deseamos mayores corrientes debemos utilizar otros drivers o implementar uno concomponentes discretos , es decir con transistores de potencia.



**Subtema 1.3:**

Control del Motor

Las etapas para controlar un motor Paso a Paso



**Actividad:**

Los estudiantes desarrollarán ejercicios de Teoría de exponentes y radicales, propuestos por el docente. según lo aprendido en clase. a) CUESTIONARIO TÉCNICO

• ¿Qué diferencia hay entre motores paso a paso unipolares y bipolares?

• ¿Por qué es importante usar driver para motores paso a paso?

• ¿Qué características posee un motor paso a paso?

• ¿Para qué me sirve el puente H?

• Investigue sobre cómo controlar los motores paso a paso con PWM y LOGO! 8.