Herramientas de Programación 2

Tema Nº5:

ENTITY FRAMEWORK CORE (PARTE 1)

Indicador de logro Nº5:

Identifica y comprueba los componentes de SOA aplicando los métodos HTTP aprendidos

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº5:**

ENTITY FRAMEWORK CORE (PARTE 1)

**Subtema 5.1:**

Verbos HTTP

Para realizar solicitudes a un servidor por medio del protocolo HTTP, es obligatorio el uso de unos métodos que vienen pre-definidos en este protocolo los cuales le indican al servidor cual es la acción que se desea realizar sobre uno o varios recursos en concreto, a estos métodos también se les conoce por el nombre de “verbos”.

GET.

Este método se utiliza es cuando se necesita adquirir un archivo o recurso que se encuentran en un servidor web, este método devuelve tanto las cabecera que contienen los metadatos del recurso solicitado y el recurso en sí.

POST.

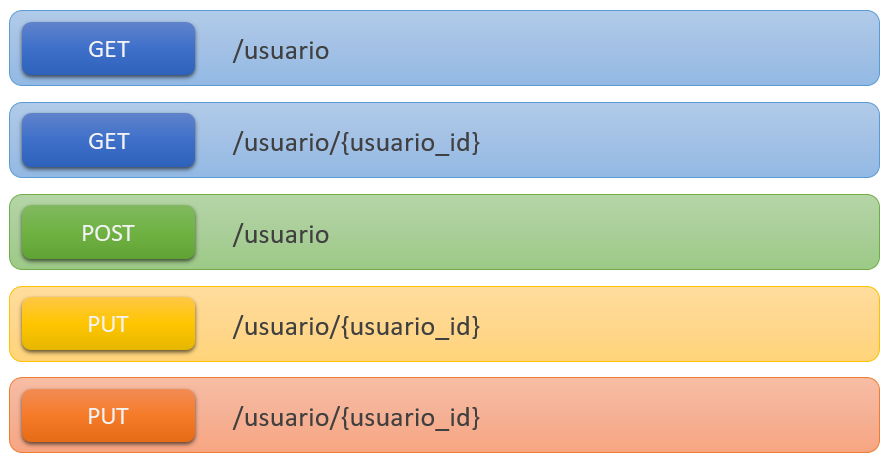
Este método se usa cuando se necesita enviar información o un elemento al servidor y que lo enviado sea almacenado como un “hijo” o subelemento de un elemento o recurso ya existentes en el servidor. Este método se usa para enviar información a un recurso web del servidor mas no para cargar/crear un elemento nuevo como tal. Este tipo de método se usa principalmente en el envío de formularios que se encuentran en las páginas web.

PUT.

Crea/Carga un nuevo recurso al servidor, o en caso de que el objeto ya exista en el servidor reemplaza el recurso existente con el recurso que se carga.

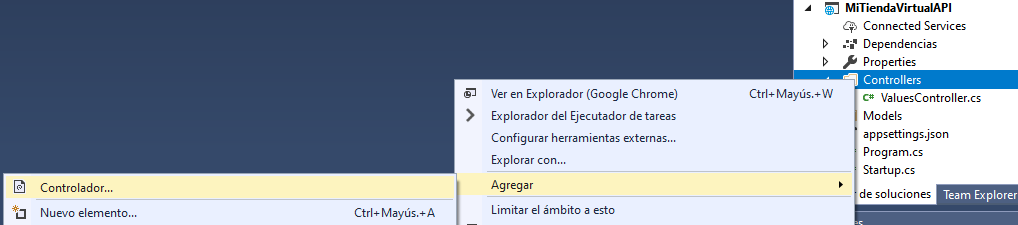
DELETE:

Este método le solicita al servidor web que se borre un recurso en específico.



Paso 1. Implementar servicios con GET. Parte 1.

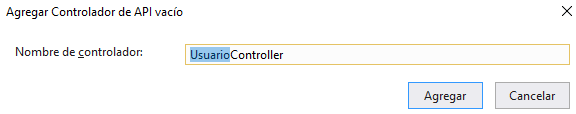
1. Abrimos en Visual Studio el proyecto del taller anterior.
2. Agregamos un controlador API en blanco mediante la opción: *Agregar > Controlador…*



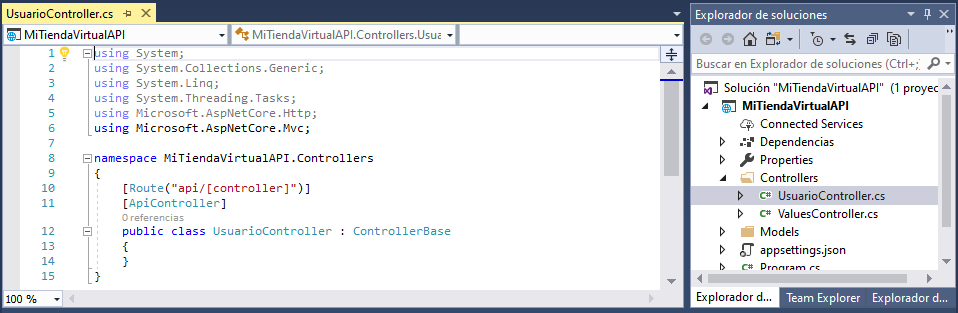
1. En el formulario siguiente seleccionamos la opción: *Controlador de API: en blanco.*



1. Le ponemos un nombre al controlador: *UsuarioController*, y presionamos en el botón *Agregar*.



1. Verificamos que se ha creado el controlador.



1. Agregamos el primer método GET en el controlador, modificamos el código del controlador de acuerdo a:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using MiTiendaVirtualAPI.Models;

namespace MiTiendaVirtualAPI.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class UsuarioController : ControllerBase

{

private readonly TiendaVirtualDBContext BD;

public UsuarioController(TiendaVirtualDBContext context)

{

BD = context;

}

//GET. /api/usuario

[HttpGet]

public IEnumerable<Usuario> ListaDeUsuarios()

{

return BD.Usuario.ToList();

}

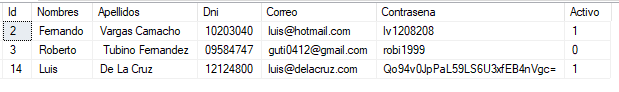
}

}

1. Ejecutamos la aplicación y resolvemos la petición GET: */api/usuario*



1. Validamos el resultado obtenido con la BD.



1. Cerramos el explorador.

Paso 2. Implementar servicios con GET. Parte 2.

1. Agregamos el segundo método GET en el controlador de acuerdo a:

//GET. /api/usuario/2

[HttpGet("{id}", Name = "UsuarioCreado")]

public IActionResult DevolverUsuario(int id)

{

var usuario = BD.Usuario.FirstOrDefault(u => u.Id == id);

if(usuario == null)

{

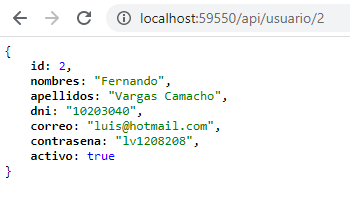
return NotFound();

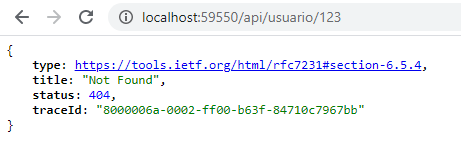
}

return Ok(usuario);

}

1. Ejecutamos la aplicación y resolvemos la petición GET: */api/usuario/2*





1. Cerramos el explorador.

Paso 3. Implementar servicios con POST.

1. Agregamos el siguiente método POST en el controlador *UsuarioController.cs*:

//POST. /api/usuario

[HttpPost()]

public IActionResult CrearUsuario([FromBody] Usuario pUsuario)

{

if (ModelState.IsValid)

{

//guardamos el usuario en la BD

BD.Usuario.Add(pUsuario);

BD.SaveChanges();

//devolvemos el usurio recientemente creado

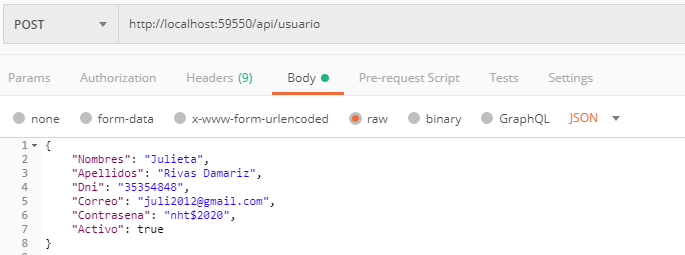
return new CreatedAtRouteResult("UsuarioCreado", new { id = pUsuario.Id}, pUsuario);

}

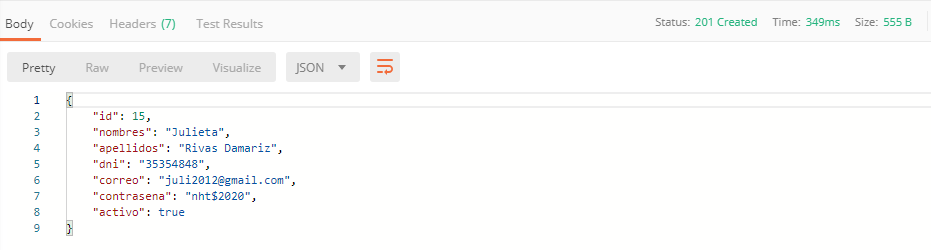
return BadRequest(ModelState);

}

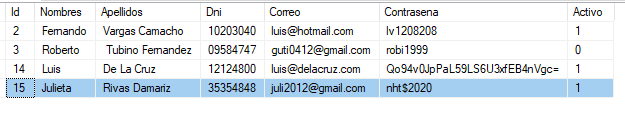
1. Ejecutamos la aplicación.
2. Para probar servicios POST que requieran que se les envié información (datos del nuevo usuario en nuestro caso) es necesario utilizar Postman.
3. Abrimos Postman y configuramos una nueva consulta de la siguiente forma:



1. Ejecutamos la consulta y obtenemos como respuesta:



1. Verificamos en la BD que el usuario ha sido creado con los datos enviados.



Paso 4. Implementar servicios con PUT.

1. Agregamos el siguiente método PUT en el controlador *UsuarioController.cs*:

//PUT. /api/usuario/5

[HttpPut("{id}")]

public IActionResult ModificarUsuario([FromBody] Usuario pUsuario, int id)

{

if (pUsuario.Id != id)

{

return BadRequest();

}

BD.Entry(pUsuario).State = EntityState.Modified;

BD.SaveChanges();

return Ok();

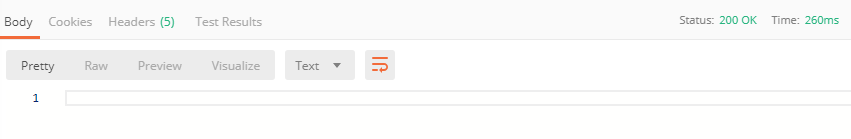
}

}

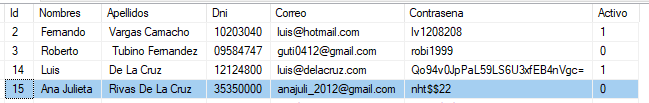
1. Ejecutamos la aplicación.
2. Abrimos Postman y configuramos una nueva consulta de la siguiente forma:



1. Ejecutamos la consulta y obtenemos como respuesta:



1. En este caso el método PUT no devuelve ninguna información.
2. Verificamos en la BD que se han actualizado los datos el usuario.



Paso 5. Implementar servicios con DELETE.

1. Agregamos el siguiente método DELETE en el controlador *UsuarioController.cs*:

//DELETE. /api/usuario/5

[HttpDelete("{id}")]

public IActionResult EliminarUsuario(int id)

{

var usuario = BD.Usuario.FirstOrDefault(u => u.Id == id);

if (usuario == null)

{

return NotFound();

}

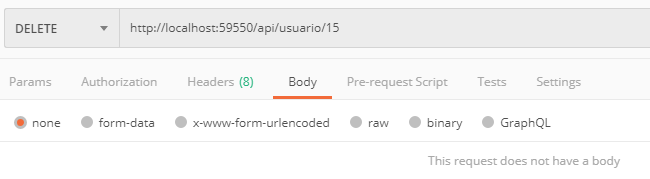
BD.Usuario.Remove(usuario);

BD.SaveChanges();

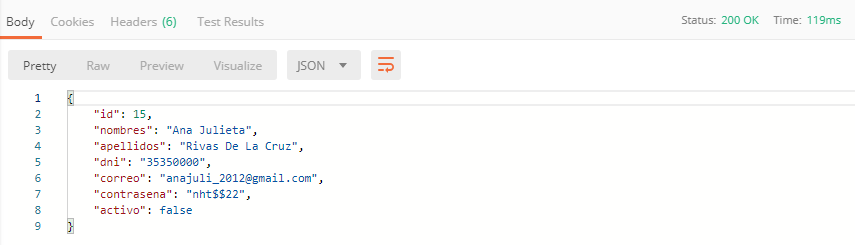
return Ok(usuario);

}

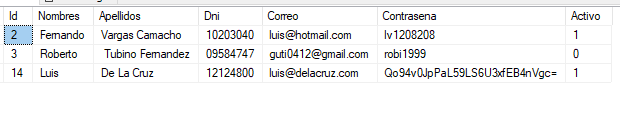
1. Ejecutamos la aplicación.
2. Abrimos Postman y configuramos una nueva consulta de la siguiente forma:



1. Ejecutamos la consulta y obtenemos como respuesta:



1. Verificamos en la BD que se ha eliminado el usuario de Id = 15.



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 el controlador *MarcaController.cs* e implementar los métodos que permitan hacer el CRUD de la tabla *Marca*.
     2. Crear el controlador *CategoriaController.cs* e implementar los métodos que permitan hacer el CRUD de la tabla *Categoria*.
     3. Crear el controlador *ProductoController.cs* e implementar los métodos que permitan hacer el CRUD de la tabla *Producto*. Debe agregar un método que devuelva la lista de productos destacados.
     4. Crear el controlador *PedidoController.cs* e implementar los métodos que permitan hacer el CRUD de la tabla *Pedido*.
     5. Investigar la forma de enviar archivos a un servicio API REST.
2. CONCLUSIONES DE LA EXPERIENCIA

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

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

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

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

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

**ACTIVIDAD** **VIRTUAL**:

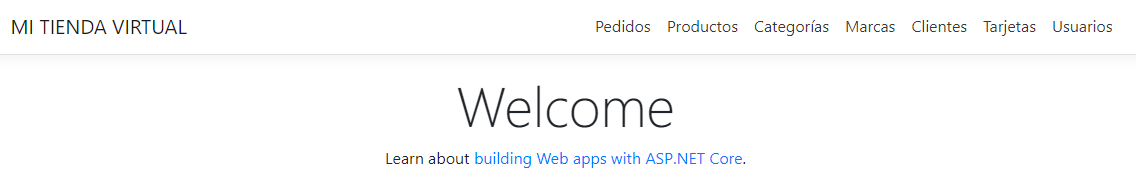
Ingresa a la plataforma virtual. Revisa y analiza el tema desarrollado en esta sesión, luego desarrolla las siguientes actividades propuestas:

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

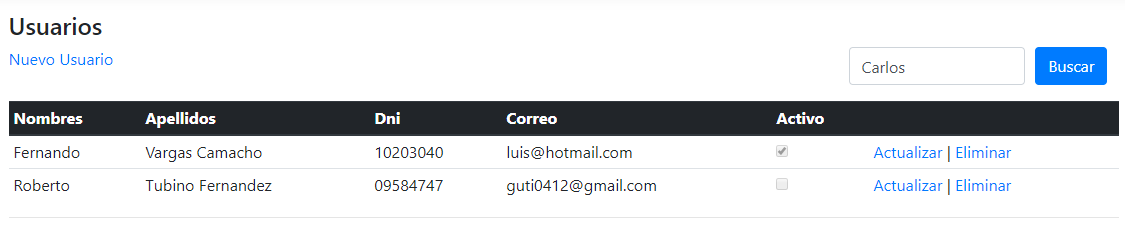
* Marca
* Categoría
* Producto
  + 1. Crear con Scaffold Controladores y vistas para el mantenimiento (CRUD) de las siguientes entidades (Tablas):
* Cliente
* Tarjeta

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:

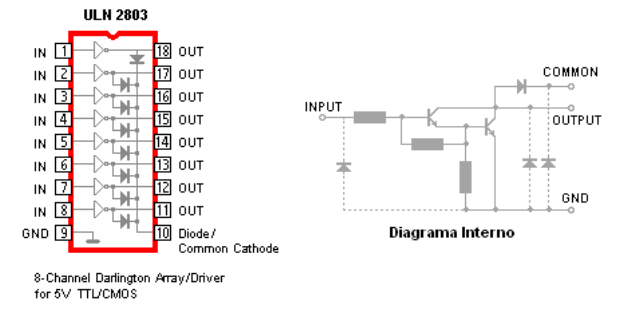


**Subtema 1.2:**

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.

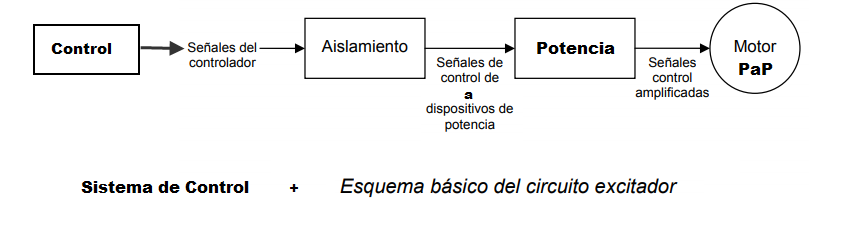
Por lo tanto si deseamos mayores corrientes debemos utilizar otros drivers o implementar uno concomponentes discretos , es decir con transistores de potencia.



**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.