

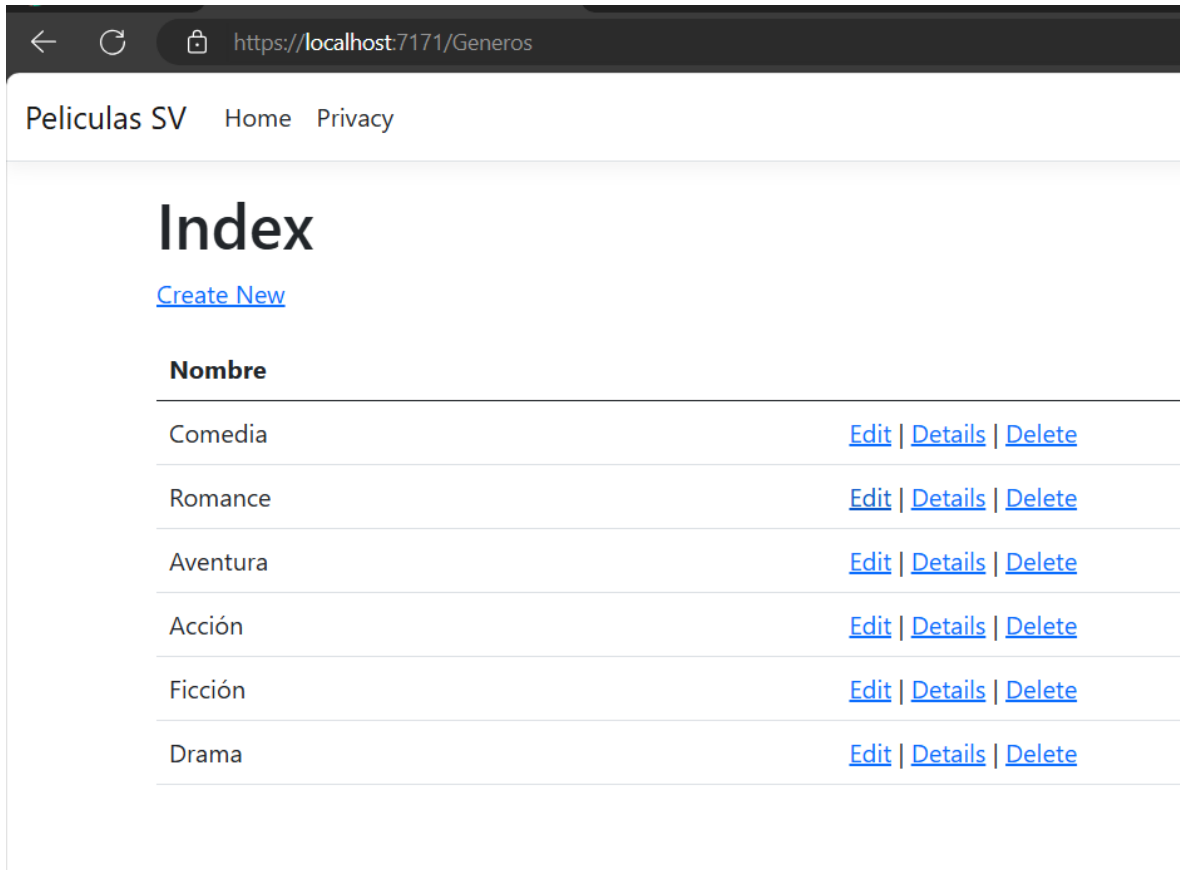
Eduardo Josué Deras Mancia

Desarrollo de Software Empresarial DES104 G02L

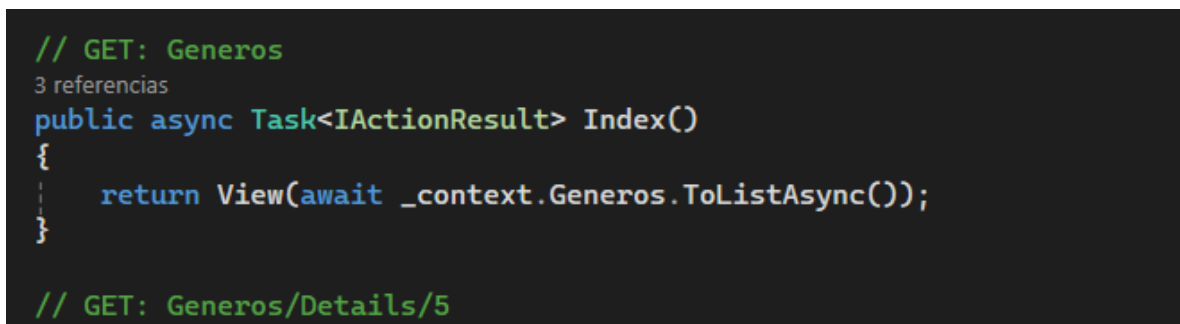
GUIA 2 Desarrollo de habilidades

Ejercicio 1

Complete la aplicación generando el código necesario para tener el CRUD de los géneros de películas.



Genero.controller



Genero.view

```
<h1>Index</h1>

<p>
  <a asp-action="Create">Create New</a>
</p>
<table class="table">
  <thead>
    <tr>
      <th>
        @Html.DisplayNameFor(model => model.Nombre)
      </th>
    </tr>
  </thead>
  <tbody>
    @foreach (var item in Model) {
      <tr>
        <td>
          @Html.DisplayFor(modelItem => item.Nombre)
        </td>
        <td>
          <a asp-action="Edit" asp-route-id="@item.Id">Edit</a> |
          <a asp-action="Details" asp-route-id="@item.Id">Details</a> |
          <a asp-action="Delete" asp-route-id="@item.Id">Delete</a>
        </td>
      </tr>
    }
```

Genero.model

```
namespace MVCPelicula.Models
{
    27 referencias
    public class Genero
    {
        19 referencias
        public int Id { get; set; }

        [StringLength(100)]
        [Required]
        19 referencias
        public string Nombre { get; set; }

        0 referencias
        public ICollection<Pelicula> Peliculas { get; set; }
    }
}
```

- Realice las modificaciones necesarias en la vistas de Películas, para que al momento de agregar y/o modificar un registro, aparezca el nombre del género en vez de su Id

Películas.controller

```
public async Task<IActionResult> Index(String textoABuscar)
{
    if(_context.Peliculas == null)
    {
        return Problem("No se ha inicializado el contexto");
    }
    var peliculas = from p in _context.Peliculas
                    .Include(p => p.Genero)
                    select p;
    if(!String.IsNullOrEmpty(textoABuscar))
    {
        peliculas = peliculas.Where(p => p.Titulo.Contains(textoABuscar));
    }

    return View(await peliculas.ToListAsync());
}
```

```
public async Task<ActionResult> Create([Bind("ID,Titulo,FechaLanzamiento,GeneroId,Precio,Director")] Pelicula pelicula)
{
    ModelState.Remove("Genero");
    if (ModelState.IsValid)
    {
        _context.Add(pelicula);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    ViewData["GeneroId"] = new SelectList(_context.Generos, "Id", "Nombre", pelicula.GeneroId);
    return View(pelicula);
}
```

```
// GET: Peliculas/Edit/5
0 referencias
public async Task<IActionResult> Edit(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var pelicula = await _context.Peliculas.FindAsync(id);
    if (pelicula == null)
    {
        return NotFound();
    }

    ViewData["GeneroId"] = new SelectList(_context.Generos, "Id", "Nombre", pelicula.GeneroId);
    return View(pelicula);
}
```