

## 16. Ordenar los registros por sus columnas

Para agregar el ordenamiento de registros según sus columnas a la página Index de Categoría, se agregará el siguiente Código:

Agregue la funcionalidad de ordenamiento al método Index del controlador CategoríasController:

```
public async Task<IActionResult> Index(string sortOrder)
{
    ViewData["NombreSortParm"] = String.IsNullOrEmpty(sortOrder) ? "nombre_desc"
: "";
    ViewData["DescripcionSortParm"] = sortOrder == "descripcion_asc" ?
"descripcion_desc" : "descripcion_asc";
    var categorias = from s in _context.Categorias
                    select s;
    switch (sortOrder)
    {
        case "nombre_desc":
            categorias = categorias.OrderByDescending(s => s.Nombre);
            break;
        case "descripcion_asc":
            categorias = categorias.OrderBy(s => s.Descripcion);
            break;
        case "descripcion_desc":
            categorias = categorias.OrderByDescending(s => s.Descripcion);
            break;
        default:
            categorias = categorias.OrderBy(s => s.Nombre);
            break;
    }
    return View(await categorias.AsNoTracking().ToListAsync());
    //return View(await _context.Categorias.ToListAsync());
}
```

Este código recibe un parámetro `sortOrder` de la cadena de consulta en la URL. El valor de cadena de consulta es proporcionado por ASP.NET Core MVC como un parámetro al método de acción. El parámetro será una cadena que sea "nombre " o "descripcion" opcionalmente seguido de un subrayado y la cadena "desc" o "asc" para especificar el orden descendente o ascendente. El orden predeterminado es la columna nombre\_asc nombre ascendente.

La primera vez que se solicita la página de `Index`, no hay ninguna cadena de consulta. Las categorías se muestran en orden ascendente por Nombre, que es el valor predeterminado establecido por el caso por defecto en la instrucción `switch`. Cuando el

usuario hace clic en un hipervínculo de encabezado de columna, se proporciona el valor apropiado de `sortOrder` en la cadena de consulta.

La vista utiliza los dos elementos ViewData (NombreSortParm y DescripcionSortParm) para configurar los hipervínculos de encabezado de columna con los valores de cadena de consulta apropiados.

El método utiliza LINQ para especificar la columna por la que ordenar. El código crea una variable `IQueryable` antes de la instrucción `switch`, la modifica en la instrucción `switch` y llama al método `ToListAsync` después de la instrucción `switch`. Al crear y modificar variables `IQueryable`, no se envía ninguna consulta a la base de datos. La consulta no se ejecuta hasta convertir el objeto `IQueryable` en una colección llamando a un método como `ToListAsync`. Por lo tanto, este código da como resultado una consulta única que no se ejecuta hasta la sentencia `return View`.

### Agregando hipervínculos de encabezado de columna a la vista Index de las categorías

Reemplace el código en *Views/Categorias/Index.cshtml*, con el siguiente código para agregar hipervínculos de encabezado de columna. Las líneas cambiadas se resaltan.

```
<thead>
    <tr>
        <th>
            <a asp-action="Index" asp-route-
sortOrder="@ViewData["NombreSortParm"]">@Html.DisplayNameFor(model => model.Nombre)</a>
        </th>
        <th>
            <a asp-action="Index" asp-route-
sortOrder="@ViewData["DescripcionSortParm"]">@Html.DisplayNameFor(model =>
model.Descripcion)</a>
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Estado)
        </th>
    </tr>
</thead>
```