

# Ejercicio Restaurante Maestro – Detalles

Ing. Edgar F. Montaluisa P. MSc

ASP .NET MVC

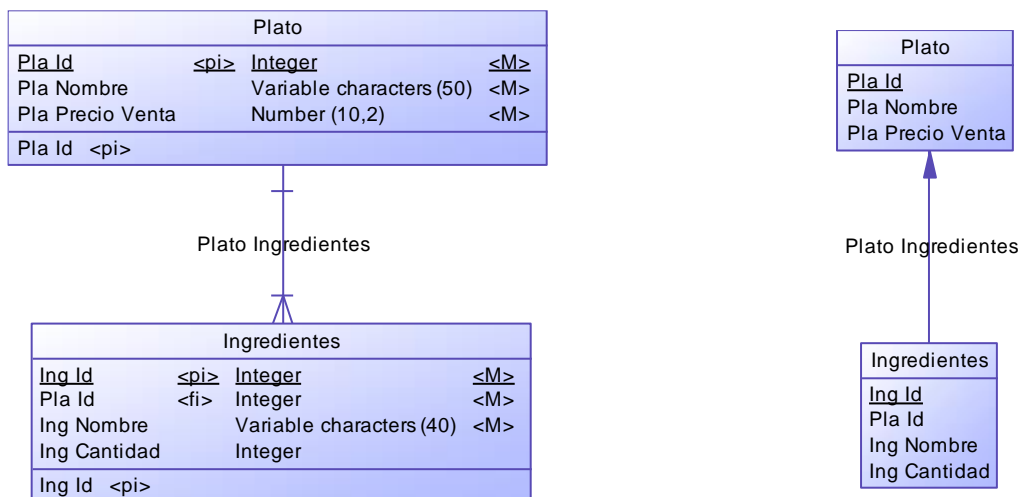
+593 983411530

<https://github.com/efmontaluisa/RestauranteMD>

## Descripción

El ejercicio describe la utilización de los objetos de modelo para capturar datos de la Vista en una relación Maestro Detalle, para lo cual se realizará el registro de un Plato y sus ingredientes que lo componen para poder publicitarlo en un Restaurante.

## Modelo de Datos



## Script de Base de Datos

```
/*=====*/  
/* DBMS name:   Microsoft SQL Server          */  
/*=====*/
```

```
if exists (select 1  
  from sys.sysreferences r join sys.sysobjects o on (o.id = r.constid and o.type = 'F')  
  where r.fkeyid = object_id('ingredientes') and o.name = 'fk_ingredie_plato_ing_plato')  
alter table ingredientes  
  drop constraint fk_ingredie_plato_ing_plato  
go  
  
if exists (select 1  
  from sysindexes  
  where id = object_id('ingredientes')  
  and name = 'plato_ingredientes_fk'  
  and indid > 0)
```

```
        and indid < 255)
drop index ingredientes.plato_ingredientes_fk
go
```

```
if exists (select 1
           from sysobjects
           where id = object_id('ingredientes')
           and type = 'U')
drop table ingredientes
go
```

```
if exists (select 1
           from sysobjects
           where id = object_id('plato')
           and type = 'U')
drop table plato
go
```

```
/*=====*/
/* Table: ingredientes */
/*=====*/
create table ingredientes (
    ing_id      int      identity,
    pla_id      int      not null,
    ing_nombre  varchar(40) not null,
    ing_cantidad int      null,
    constraint pk_ingredientes primary key nonclustered (ing_id)
)
go
```

```
/*=====*/
/* Index: plato_ingredientes_fk */
/*=====*/
create index plato_ingredientes_fk on ingredientes (
    pla_id asc
)
go
```

```
/*=====*/
/* Table: plato */
/*=====*/
create table plato (
    pla_id      int      identity,
    pla_nombre  varchar(50) not null,
    pla_precio_venta numeric(10,2) not null,
    constraint pk_plato primary key nonclustered (pla_id)
)
go
```

```
alter table ingredientes
```

```
add constraint fk_ingredie_plato_ing_plato foreign key (pla_id)
references plato (pla_id)
go
```

New Database

Select a page

- General
- Options
- Filegroups

Script Help

Database name: Restaurante

Owner: <default>

☒ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
Restaurante	ROWS...	PRIMARY	8	By 64 MB, Unlimited
Restaurante...	LOG	Not Applicable	8	By 64 MB, Unlimited

Connection

## Código MVC: Modelo PLATO

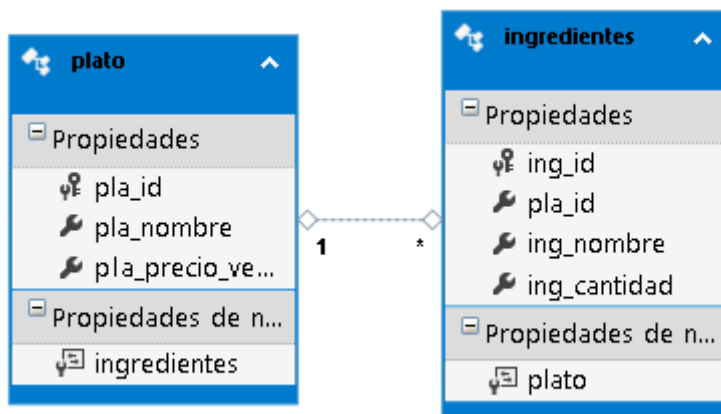
```
//-----  
// <auto-generated>  
//     Este código se generó a partir de una plantilla.  
//  
//     Los cambios manuales en este archivo pueden causar un comportamiento  
//     inesperado de la aplicación.  
//     Los cambios manuales en este archivo se sobrescribirán si se regenera el  
//     código.  
// </auto-generated>  
//-----  
  
namespace Restaurante  
{  
    using System;  
    using System.Collections.Generic;  
    using System.ComponentModel;  
    using System.ComponentModel.DataAnnotations;  
  
    public partial class plato  
    {  
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",  
"CA2214:DoNotCallOverridableMethodsInConstructors")]  
        public plato()  
        {  
            this.ingredientes = new HashSet<ingredientes>();  
        }  
  
        public int pla_id { get; set; }  
        [Required]  
        [DisplayName("Nombre")]  
        public string pla_nombre { get; set; }  
        [Required]  
        [DisplayName("Precio")]  
        public decimal pla_precio_venta { get; set; }  
  
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",  
"CA2227:CollectionPropertiesShouldBeReadOnly")]  
        [DisplayName("Ingredientes")]  
        public virtual ICollection<ingredientes> ingredientes { get; set; }  
    }  
}
```



## Código MVC: Modelo INGREDIENTES

```
//-----  
// <auto-generated>  
//     Este código se generó a partir de una plantilla.  
//  
//     Los cambios manuales en este archivo pueden causar un comportamiento  
//     inesperado de la aplicación.  
//     Los cambios manuales en este archivo se sobrescribirán si se regenera el  
//     código.  
// </auto-generated>  
//-----
```

```
namespace Restaurante  
{  
    using System;  
    using System.Collections.Generic;  
    using System.ComponentModel;  
    using System.ComponentModel.DataAnnotations;  
  
    public partial class ingredientes  
    {  
        [Required]  
        [DisplayName("Id")]  
        public int ing_id { get; set; }  
        [Required]  
        [DisplayName("Plato")]  
        public int pla_id { get; set; }  
        [Required]  
        [DisplayName("Ingrediente")]  
        public string ing_nombre { get; set; }  
        [DisplayName("Cantidad")]  
        public Nullable<int> ing_cantidad { get; set; }  
  
        public virtual plato plato { get; set; }  
    }  
}
```



## Código MVC: Controller PLATO

Se incluye únicamente el código de los ActionResult añadidos al controlador PLATO para crear la operatividad de Maestro Detalle, que como se puede ver es sumamente simple ya que solo invoca a SaveChanges y él es quien realiza todo el trabajo. El éxito de la funcionalidad esta enteramente en la programación de la Vista.

```
// GET: plato/CreatePlatoCompleto
public ActionResult CreatePlatoCompleto()
{
    return View();
}

// POST: plato/CreatePlatoCompleto
// To protect from overposting attacks, please enable the specific
properties you want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult CreatePlatoCompleto( plato plato)
{
    if (ModelState.IsValid)
    {
        db.plato.Add(plato);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(plato);
}
```

## Código MVC: Vista CreatePlatoCompleto

```
@model Restaurante.plato

@{
    ViewBag.Title = "Crear Plato Completo";
}

<h2>Crear Plato Completo</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.pla_nombre, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.pla_nombre, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.pla_nombre, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.pla_precio_venta, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.pla_precio_venta, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.pla_precio_venta, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
```

```



```

```

</div>

```

```

</div>

```

```

<!-- Tabla de ingredientes-->

```

```

<div class="form-group">

```

```

    <div class="col-md-offset-2 col-md-10">

```

```

        <table class="table table-striped" id="tablaIngredientes">

```

```

            <tr>

```

```

                <th>#</th>

```

```

                <th>Ingrediente</th>

```

```

                <th>Cantidad</th>

```

```

            </tr>

```

```

        </table>

```

```

    </div>

```

```

</div>

```

```

}

```

```

<script>

```

```

    var numFila = 0;

```

```

    function agregar() {

```

```

        var fila = '<tr>' +

```

```

            '<th>' + (numFila+1) + '</th>' +

```

```

            '<th><input type="text" name="ingredientes[' + numFila + '].ing_nombre" size="40" maxlength="40" /></th>' +

```

```

            '<th><input type="text" name="ingredientes[' + numFila + '].ing_cantidad" size="10" maxlength="10" /></th>' +

```

```

            '</tr>';

```

```

        var tabla = document.getElementById("tablaIngredientes");

```

```

        tabla.insertRow(-1).innerHTML = fila;

```

```

        numFila++;

```

```

    }

```

```

    function borrar() {

```

```

        numFila--;

```

```

        var tabla = document.getElementById("tablaIngredientes");

```

```

        var rowCount = tabla.rows.length;

```

```

        if (rowCount >1)

```



```

        tabla.deleteRow(rowCount - 1);
    }
</script>
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

## Crear Plato Completo

Nombre

Precio

[Crear Plato](#) | 
 [Regresar a Lista](#) | 
 [+ Agregar Ingrediente](#) | 
 [- Eliminar Ingrediente](#)

#	Ingrediente	Cantidad
1	<input type="text" value="Lomo"/>	<input type="text" value="1"/>
2	<input type="text" value="Chuleta"/>	<input type="text" value="1"/>
3	<input type="text" value="Pollo"/>	<input type="text" value="1"/>
4	<input type="text" value="Embutidos"/>	<input type="text" value="4"/>
5	<input type="text" value="Papas"/>	<input type="text"/>
6	<input type="text" value="Limonada"/> ×	<input type="text"/>