

PRACTICA DIRIGIDA DE MVC

Objetivo .- Aplicar @HTMl.Helpers, con generación de códigos ,insertando ,actualizando y eliminando registros de una tabla de datos.

1.- Se pide realizar la matricula de alumnos en los cursos de extensión, para ello muestra la siguiente pagina:

Cursos de Extension

Seleccione Especialidad Programacion ▼

Codigo	ASignatura	Costo	Vacantes	Matricula
C0002	VisuaL Basic	300	14	Alumnos
C0003	VisuaL Net	500	10	Alumnos
C0004	Power Builder	400	4	Alumnos

2.- Al hacer un clic en el hipervínculo alumnos va mostrar la pagina webMatricula.aspx (7 puntos)

Matricula de Alumnos

Nro de matricula

Nombres y Apellidos

Tipo deAlumno ☐ 1/2 Beta ☐ 1/4 Beta ☐ Regular

Pago

Lista de Alumnos del Curso **Visual Net** Costo **500**

Nro Matricula	Nombre y Apellido	Tipo	Pago	Anular
A00002	Herrera Hidalgo, Marlon	B	120	Eimina
A00005	Alvis Vidal, Sivestre	A	160	Eimina
A00002	Miranda Hernadez, Maria	B	170	Eimina

Condiciones:

- El código de alumno se genera a partir de 000001 (2 Puntos)
- Mostrar solo los alumnos del curso seleccionado (2 puntos)
- El pago se calcula automáticamente de acuerdo al tipo de alumno que se elige (1 Puntos)
- Emplear un store Procedure para grabar(las vacantes dismiuyen en 1),

actualizando el listado además (2 Puntos)

Paso 1.- Ejecutar el script de base de datos

USE master

DROP DATABASE IF EXISTS bdMatricula;

CREATE DATABASE bdMatricula;

USE bdMatricula;

```
CREATE TABLE ALUMNO (  
    NROMAT char (6) NOT NULL primary key ,  
    CODCUR char (4) NULL ,  
    APENOM VARCHAR (25) NULL,  
    TIPO CHAR(1) NULL ,  
    PAGO NUMERIC(8,0) NULL  
);
```

```
CREATE TABLE TABESP (  
    CODESP char(2) NOT NULL PRIMARY KEY ,  
    NOMESP char (20) NULL  
);
```

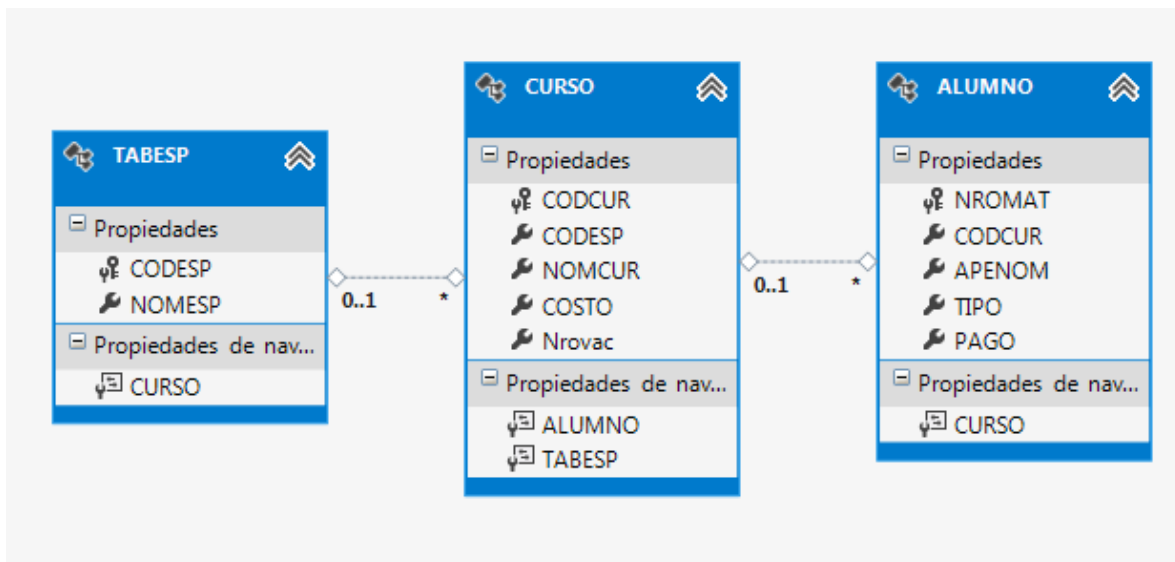
```
CREATE TABLE CURSO (  
    CODCUR char(4) NOT NULL PRIMARY KEY ,  
    CODESP char(2) NULL ,  
    NOMCUR char (15) NULL ,  
    COSTO numeric(8, 0) NULL,  
    Nrovac int  
);
```

Crear el procedimiento para generar códigos e insertar

```
Create PROCEDURE spadiAlu  
(@codc char(4),@ape varchar(25),@tipo char(1),@pago numeric(8,0))  
as  
begin  
declare @coda char(6)  
declare @cuenta int  
select @cuenta=max(nromat)+1 from alumno  
set @coda=right('00000'+ltrim(@cuenta),6);  
insert into alumno values(@coda,@codc, @ape,@tipo,@pago);  
end
```

paso2' - Crear un proyecto MvcMatricula

crear Entity Framework con la base Matricula en la capa modelo luego compilar



En la capa Controlador , en la clase Home. Listar las especialidades y los cursos de una especialidad.

Paso 3 Creacion de los codigos

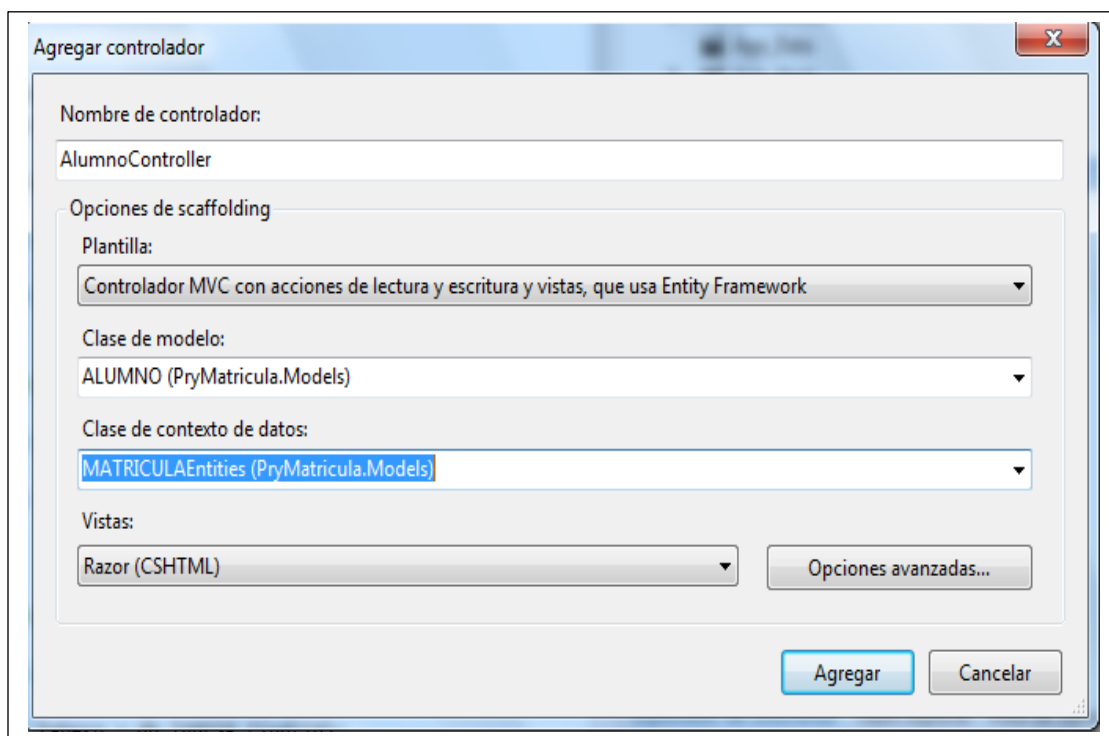
```

using PryMatricula.Models;
namespace PryMatricula.Controllers
{
    public class HomeController : Controller
    {
        // MATRICULAEntities obj = new MATRICULAEntities();

        public ActionResult Index()
        {
            //mostrar la lista de Especialidades
            using (MATRICULAEntities obj = new MATRICULAEntities())
            {
                return View(obj.TABESP.ToList());
            }
        }
        public ActionResult Cursos(string code, string nome)
        {
            using(MATRICULAEntities obj = new MATRICULAEntities()){
                var sql = from c in obj.CURSO
                           where c.CODESP == code
                           select c;
                ViewBag.nombre = nome;
                return View(sql.ToList());
            }
        }
    }
}
  
```

```
public ActionResult Alumnos(string codc, string nomc, string pcos)
{
    using (MATRICULAEntities obj = new MATRICULAEntities())
    {
        var sql = from c in obj.ALUMNO
                   where c.CODCUR == codc
                   select c;
        Session["nomc"] = nomc;
        Session["costo"] = pcos;
        Session["codc"] = codc;
        return View(sql.ToList());
    }
}
```

Creando un nuevo Controlador.=> AlumnoControlador



```
using PryMatricula.Models;

namespace PryMatricula.Controllers
{
    public class AlumnoController : Controller
    {
```

```
private MATRICULAEntities db = new MATRICULAEntities();

//
// GET: /Alumno/

public ActionResult Index(string codc, string nomc, string pcos)
{
    var sql = from c in db.ALUMNO
              where c.CODCUR == codc
              select c;
    Session["nomc"] = nomc;
    Session["costo"] = pcos;
    Session["codc"] = codc;
    return View(sql.ToList());
}

public ActionResult Details(string id = null)
{
    ALUMNO alumno = db.ALUMNO.Find(id);
    if (alumno == null)
    {
        return HttpNotFound();
    }
    return View(alumno);
}

//
// GET: /Alumno/Create

public ActionResult Create()
{
    ViewBag.CODCUR = new SelectList(db.CURSO, "CODCUR",
"NOMCUR");
    return View();
}

//
// POST: /Alumno/Create

[HttpPost]
public ActionResult Create(ALUMNO a)
{
    if (ModelState.IsValid)
    {
        a.CODCUR = Session["codc"].ToString();
        db.spadiAlu(a.CODCUR, a.APENOM, a.TIPO, a.PAGO);
        //db.ALUMNO.Add(alumno);
        // db.SaveChanges();
    }
}
```

```
        return RedirectToAction("Index", new {
codc=Session["codc"], nomc=Session["nomc"], pcos=Session["costo"] });
    }

    // ViewBag.CODCUR = new SelectList(db.CURSO, "CODCUR",
"CODESP", alumno.CODCUR);
    a.CODCUR = Session["codc"].ToString();
    return View(a);
}

//
// GET: /Alumno/Edit/

public ActionResult Edit(string id = null)
{
    ALUMNO alumno = db.ALUMNO.Find(id);
    if (alumno == null)
    {
        return HttpNotFound();
    }
    ViewBag.CODCUR = new SelectList(db.CURSO, "CODCUR", "NOMCUR",
alumno.CODCUR);
    return View(alumno);
}

//
// POST: /Alumno/Edit/5

[HttpPost]
public ActionResult Edit(ALUMNO alumno)
{
    if (ModelState.IsValid)
    {
        db.Entry(alumno).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.CODCUR = new SelectList(db.CURSO, "CODCUR", "NOMCUR",
alumno.CODCUR);
    return View(alumno);
}

//
// GET: /Alumno/Delete/5

public ActionResult Delete(string id = null)
{
    ALUMNO alumno = db.ALUMNO.Find(id);
    if (alumno == null)
    {
```

```

        return HttpNotFound();
    }
    return View(alumno);
}

//
// POST: /Alumno/Delete/5

[HttpPost, ActionName("Delete")]
public ActionResult DeleteConfirmed(string id)
{
    ALUMNO alumno = db.ALUMNO.Find(id);
    db.ALUMNO.Remove(alumno);
    db.SaveChanges();
    return RedirectToAction("Index", new { codc =
Session["codc"], nomc = Session["nomc"], pcas = Session["costo"] });
}

protected override void Dispose(bool disposing)
{
    db.Dispose();
    base.Dispose(disposing);
}
}
}

```

Creación de las vistas

Al seleccionar una especialidad mostrara la lista de cursos.

Lista de Especialidades		
CODESP	NOMESP	
01	Lenguajes Visuales	Cursos
02	Programacion Web	Cursos
03	Redes	Cursos
04	Base de Dato	Cursos
05	Sistema Operativo	Cursos
06	Diseño Web	Cursos

Al seleccionar un curso va mostrar la lista de alumno.

Cursos de la Especialidad Programacion Web					
CODCUR	CODESP	NOMCUR	COSTO	Nro vac	
C007	02	Interdev	450.00	9	Alumnos
C008	02	Dream Weber	650.00	5	Alumnos
C009	02	HTML	450.00	11	Alumnos
C010	02	ASP	350.00	14	Alumnos

Alumnos del Curso Interdev Costo :450					
Registrar nuevo Alumno					
NROMAT	CODESP	APENOM	TIPO	PAGO	
000013	02	Gonza Patricia	2	220.00	Delete
000017	02	Mendoza Marisol	3	330.00	Delete
000027	02	Cotrina Cordova	3	330.00	Delete
000033	02	Bobadilla Patricia	2	220.00	Delete
000037	02	Cotrina Cordova	3	330.00	Delete
000052	02	Jesus Moreno	1	315.00	Delete
Pagina Inicial					

Paso 4 Codificación de las vistas

Home=> Index.csthml

```
el IEnumerable<PryMatricula.Models.TABESP>

@{
    ViewBag.Title = "Index";
    Layout = "~/Views/_Layout.cshtml";
}
<center>
```



```

<h2 class="marco">Lista de Especialidades</h2>

<table class="mGrid">
  <tr>
    <th>
      @Html.DisplayNameFor(model => model.CODESP)
    </th>
    <th>
      @Html.DisplayNameFor(model => model.NOMESP)
    </th>
    <th></th>
  </tr>

  @foreach (var item in Model) {
    <tr>
      <td>
        @Html.DisplayFor(modelItem => item.CODESP)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.NOMESP)
      </td>
      <td>
        @Html.ActionLink("Cursos", "Cursos", new {
          code=item.CODESP,nome=item.NOMESP }) |
      </td>
    </tr>
  }
</table>
</center>

```

Home=> Cursos.csthml

```

@model IEnumerable<PryMatricula.Models.CURSO>

@{
    ViewBag.Title = "Cursos";
    Layout = "~/Views/_Layout.cshtml";
}
<center>
<h2>Cursosde la Especialidad @ViewBag.nombre</h2>

<table class="mGrid">
  <tr>
    <th>
      @Html.DisplayNameFor(model => model.CODCUR)
    </th>

```

```
<th>
    @Html.DisplayNameFor(model => model.CODESP)
</th>
<th>
    @Html.DisplayNameFor(model => model.NOMCUR)
</th>
<th>
    @Html.DisplayNameFor(model => model.COSTO)
</th>
<th>
    @Html.DisplayNameFor(model => model.Nrovac)
</th>
<th></th>
</tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.CODCUR)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.CODESP)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.NOMCUR)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.COSTO)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Nrovac)
        </td>
        <td>
            @Html.ActionLink("Alumnos", "../Alumno/Index", new {
codc=item.CODCUR,nomc=item.NOMCUR,pcos=item.COSTO }) |

        </td>
    </tr>
}

</table>
</center>
```

Creación de las vistas del Alumno

```
@model PryMatricula.Models.ALUMNO
```

```
@{
    ViewBag.Title = "Create";
```

```
Layout = "~/Views/_Layout.cshtml";

}

<h2 class="marco">Registrar alumnos del Curso:@Session["nomc"]
  <br />Codigo curso :@Session["codc"]
  <br />Costo del Curso :@Session["costo"]
</h2>

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

    <fieldset class="marco">
        <legend>ALUMNO</legend>

        @Html.Hidden("wcosto",@Session["costo"].ToString())

        <div class="editor-label">
            @Html.LabelFor(model => model.APENOM)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.APENOM)
            @Html.ValidationMessageFor(model => model.APENOM)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.TIPO)
        </div>
        <div class="editor-field">
            @Html.RadioButtonFor(model =>
model.TIPO, "1", new { onchange="ver(1)" }) 1/4 beca
            @Html.RadioButtonFor(model =>
model.TIPO, "2", new { onchange="ver(2)" }) 1/2 beca
            @Html.RadioButtonFor(model =>
model.TIPO, "3", new { onchange="ver(3)" }) Regular

            @Html.ValidationMessageFor(model => model.TIPO)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.PAGO)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.PAGO)
            @Html.ValidationMessageFor(model => model.PAGO)
        </div>
    </fieldset>
}
```

```
<p>
    <input type="submit" value="Create" />
</p>
</fieldset>
}
<br />
<div>
    @Html.ActionLink("Back to List", "Index", new { codc =
Session["codc"], nomc = Session["nomc"], pcost = Session["costo"] })
</div>
@*
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
*@

<script type="text/javascript">

    function ver(op) {
        alert(op)
        costo = parseFloat(wcosto.value)
        alert(costo)
        if (op == 1) cos = costo * 0.7
        if (op == 2) cos = costo * 0.5
        if (op == 3) cos = costo
        PAG0.value = cos;
    }

</script>
```

Nota.-La edición, anulación y detalle se han generado automáticamente

Registrar alumnos del Curso:Interdev
Codigo curso :C007
Costo del Curso :450

ALUMNO
APENOM
Juan Lee

TIPO
☐ 1/4 beca ☒ 1/2 beca ☐ Regular

PAGO
225

Create

[Back to List](#)

TRABAJO AUTONMO

Realizar el proyecto de viajes efectuando con Linq pasarlo a MVC