Practico 5 –

BetaGamer: Aficionados a los Juegos.

# Introducción

En este práctico vamos a agregar funcionalidad de comentar Juegos y luego visualizar los comentarios en la vista de detalle de Juego.

# modificar el modelo

Edite las propiedades que representan relaciones agregando el modificador Virtual de la siguiente forma:

En la clase Juego:

public virtual ICollection<Comentario> Comentarios { get; set; }

En la clase Comentario:

public virtual Juego Juego { get; set; }

# MODIFICAR JUEGOCONTROLLER

Vamos a crear dos nuevos métodos en JuegoController de nombre AgregarComentario.

El primer método recibirá un id del Juego al que se le quiere agregar comentarios. Con dicho id vamos a buscar el Juego y lo pasaremos a la vista utilizando ViewBag.

public ActionResult AgregarComentario(int id)

{

using (var db = new BetaGamerContext())

{

ViewBag.Juego= db.Juegos.Find(id);

return View();

}

}//Para esta acción vamos a crear una vista de detalle

El segundo método recibirá un nuevo comentario a agregar al Juego seleccionado. Adicionalmente se consultará un campo oculto de la vista en el que se almacenó temporalmente el id del Juego comentado.

[HttpPost]

public ActionResult AgregarComentario(Comentario newComentario) {

try{

//obtenemos el id del Juego comentado

var JuegoId = int.Parse(Request["JuegoID"]);

using (var db = new BetaGamerContext()){

//buscamos el Juego y lo asignamos al comentario

newComentario.Juego = db.Juegos.Find(JuegoId);

db.Comentarios.Add(newComentario);

db.SaveChanges();

}

return RedirectToAction("Index");

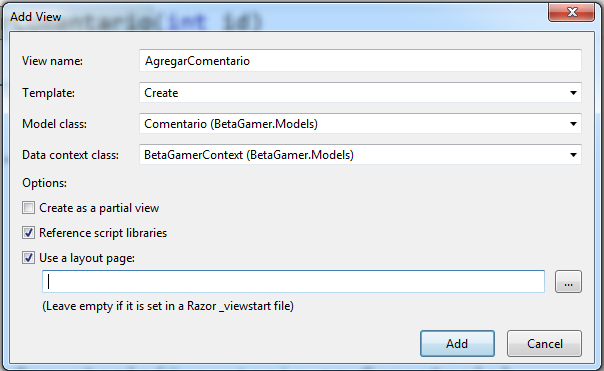
}catch{

return View();

}  
 }

# vista para las nuevas acciones del controller

Sobre el nuevo método hacer click derecho🡪Agregar Vista



Navegue a la carpeta Views. Note que se creó en la carpeta Juego una vista para esta acción.

Podrá ver que se generó los campos para crear un comentario, pero no tenemos los datos del Juego a comentar así que vamos a agregarlos después de la instrucción @Html.ValidationSummary() y antes de los datos de comentario.

<div>

<legend>Datos del Juego</legend>

<div class="panel-title">Id</div>

<div class="text-left">

<!--Este campo será utilizado para poder recibir y pasar en el post el id del objeto al controlador-->

<input type="text" name="JuegoID" id="JuegoID" value="@ViewBag.Juego.Id" />

</div>

<table class="table">

<tr>

<th>Titulo</th>

<th>FechaPublicacion</th>

<th>Descripción</th>

</tr>

<tr>

<td>ViewBag.Juego.Titulo</td>

<td>@{ String.Format("{0:g}", ViewBag.Juego.FechaPublicacion);}</td>

<td>@ViewBag.Juego.Descripción</td>

</tr>

</table>

</div>

Modifique el Action link al pie para que redirija al Details de Juego:

@Html.ActionLink("Ver comentarios", "Details", new { id=ViewBag.Juego.Id}) )

# MODIFIQUE el METODO Details DEL CONTROLADOR de Juego

// GET: /Juego/Details/5

public ActionResult Details(int id = 0)

{

//var db = new BetaGamerContext();

Juego juego = db.Juegos.Find(id);

//cargamos explicitamente sus comentarios

db.Entry(juego).Collection(l=>l.Comentarios);

if (juego == null)

{

return HttpNotFound();

}

return View(juego);

}

# modifique la vista de details del Juego

Vamos a modificar la vista de detalles del Juego agregando html para mostrar sus comentarios.

<h4>Comentarios</h4>

<div>

<table class="table">

<tr>

<th>@Html.Label("Id")</th>

<th>@Html.Label("Comentario")</th>

<th>@Html.Label("Fecha")</th>

</tr>

@foreach (var item in Model.Comentarios) {

<tr>

<td>@Html.DisplayFor(modelItem => item.Id)</td>

<td>@Html.DisplayFor(modelItem => item.Fecha)</td>

<td>@Html.DisplayFor(modelItem => item.Texto)</td>

</tr>

}

</table>

</div>

# Modifique elSEARCH index de Juego para poder llamar la accion de comentar

@Html.ActionLink("Ver Comentarios", "AgregarComentario",new {id=item.Id})

# Probar la aplicación

Cargue varios Juegos. Pruebe la nueva funcionalidad de comentar y visualice los detalles de los Juegos comentados.