**Actividades**

**Ejercicios 1**

* Agregar un nuevo proyecto llamado **Entities**.
  + Dentro del proyecto **Entities** crear dos clases llamadas **Genre** y **Movie**.
* Agregar un nuevo proyecto llamado **Data.**
  + Dentro del proyecto **Data** mover la clase llamada **MoviesContext y**agregar una colección para retornar los generos (Genre objects)
* Modificar la acción **Index** de cada controlador de tal forma que la vista reciba la lista de view models:
  + var viewModel = new GenresViewModel();  
    viewModel.Genres = …
  + return View(viewModel);
* Agregar en la vista **Index** (tipada con el viewModel correspondiente) de cada entidad una lista que muestre las distintos registros recuperados de la base:
* Retornar las películas ordenadas alfabeticamente

**Ejercicios 2**: Permitir agregar una nueva película:

* Crear una nueva acción **Create** en el controlador de películas:

public ActionResult Create()

{

return this.View(new MoviesCreateModel() { ViewAction = ViewAction.Create, Movie = new Movie()});

}

* Agregar una nueva vista **Create** con un formulario que permita cargar cada campo de una película. El formulario debe ir por **POST** a una nueva acción **Create** del controlado:

[HttpPost]

public ActionResult Create(Movie movie)

{

if (this.ModelState.IsValid)

{

this.movieService.AddMovie(movie);

}

return this.View(new MoviesCreateModel() { ViewAction = ViewAction.Create, Movie = movie});

}

**Ejercicio 3**: Agregar validaciones al alta de películas:

* Campos requeridos: **Name**, **ReleaseDate**, **Plot**, **CoverLink** y **Runtime**.
* **ReleaseDate** tiene que ser una fecha valida.
* El rango valido de valores para **Runtime** es de 30 a 300.
* El campo **Name** solo permite 100 caracteres.

**Ejercicio 4**: Si la película se guardó exitosamente retornar a la grilla de películas y mostrar mensaje de éxito.

* El mensaje debe contener el nombre de la película que se creó:

this.TempData["successmessage"] = "Se ha agregado exitosamente la pelicula: " + movie.Name;

**Ejercicio5:** Agregar la funcionalidad de permitir editar una película. El usuario debería ser capaz de poder editar la película mediante haciendo click en un link botón “Editar” en la grilla de películas:

* Agregar una nueva acción en el controlador de películas (reutilizar la vista de creación):

public ActionResult Edit(Guid id)

{

var movie = this.movieService.GetMovieById(id);

return this.View("Create", new MoviesCreateModel() { ViewAction = ViewAction.Edit, Movie = movie });

}

* El formulario en este caso debe ir por **POST** a la acción **Edit** del controlado:

[HttpPost]

public ActionResult Edit(Movie movie)

{

if (this.ModelState.IsValid)

{

var movieDb = this.movieService.GetMovieById(movie.Id);

movieDb.Name = movie.Name;

movieDb.ReleaseDate = movie.ReleaseDate;

movieDb.Runtime = movie.Runtime;

movieDb.CoverLink = movie.CoverLink;

movieDb.Plot = movie.Plot;

this.movieService.Update(movie);

this.TempData["successmessage"] = "Se ha actualizado exitosamente la pelicula: " + movie.Name;

return this.RedirectToAction("Index");

}

else

{

return this.View("Create", new MoviesCreateModel() { ViewAction = ViewAction.Edit, Movie = movie });

}

}

**Ejercicio 6**: En la creación y edición de películas, incorporar la posibilidad de ingresar cuales son los géneros de la película. En la grilla de películas, por cada película mostrar cuáles son sus géneros.

* Agregar un **ListBox** en la pantalla de creación de películas:

@Html.ListBox("selectedGenres", Model.Genres.Select(s => new SelectListItem() { Text = s.Name, Value = s.Id.ToString(), Selected = Model.Movie.Genres.Any(a => a.Id == s.Id) }), new { @class = "form-control", size = "8" })

* Agregar un parámetro en la acción Crear/Editar que reciba la lista géneros seleccionados:

IEnumerable<Guid> selectedGenres

* Agregar una nueva propiedad al viewModel que sea una lista de generos:

return this.View("Create", new MoviesCreateModel() { ViewAction = ViewAction.Edit, Movie = movie, Genres = this.genreService.GetGenres() });

* Modificar las acciones para que se guarden las relaciones entre peliculas y generos:

if (selectedGenres != null)

{

foreach (var selectedGenreId in selectedGenres)

{

var genre = this.genreService.GetGenreById(selectedGenreId);

if (genre != null)

{

movie.Genres.Add(genre);

}

}

}

**Ejercicio 7**: En la grilla de películas, agregar un link “**Delete**” para borrar la película:

$.ajax({

url: '@Url.Action("Delete")/' + movieId

}).done(function () {

alert("The movie has been deleted");

row.remove();

}).fail(function () { alert("error"); });