

AUT02_05 Musica



Jorge Hernández Toledo

2º DAW - A

Jorge Hernández Toledo
2ºDAW - A

Índice

Vistazo General a las diferentes vistas.....	1
Home.....	1
Album 1.01.....	2
Album 1.02.....	3
Artistas 1.01.....	4
Artistas 1.02.....	5
Comentarios.....	6
Al pulsar en Discos aparece la lista de los discos ordenados de forma descendente por número de id y sacando 15 registros.....	8
Solo lista con 15 registros ordenados de manera descendente por nombre (Si añades añade artistas que empiecen por z).....	9
Se tendrá que crear un nuevo modelo propio llamado Review para poder realizar comentarios sobre los artistas. Este nuevo modelo tendrá que tener al menos los campos necesarios para registrar un título, descripción y valoración.....	10
Borrado en cascada.....	12
Listado de Canciones.....	13
Mostrar el nombre del artista.....	14
Problemas encontrados.....	15
Algoritmo de estrellas.....	16
Fin del informe.....	17


Vistazo General a las diferentes vistas

Home

Home - Index

HernandezJorge_MusicaInicioDiscosArtistasLoginAdmin


La Discografía



Albums

Consulta la colección de albums de nuestro catálogo.

[Ver Albums](#)



Artistas

Conoce a los artistas más punteros del panorama musical

[Ver Artistas](#)

© 2023 - HernandezJorge_Musica - [Privacy](#)

Home - Login

HernandezJorge_MusicaInicioDiscosArtistasLoginAdmin

Página en Construcción

Home - Admin

HernandezJorge_MusicaInicioDiscosArtistasLoginAdmin

Error.

An error occurred while processing your request.

Request ID: 00-59e0bcd0f225e3cff937390ef5ea58e-aae8cad71e37797e-00

Development Mode

Swapping to **Development** environment will display more detailed information about the error that occurred.

The **Development** environment shouldn't be enabled for deployed applications. It can result in displaying sensitive information from exceptions to end users. For local debugging, enable the **Development** environment by setting the **ASPNETCORE_ENVIRONMENT** environment variable to **Development** and restarting the app.

Album 1.01

Album - Index

HernandezJorge_Musica Inicio Discos Artistas

Login Admin

Listado de Albums

[Crear nuevo album](#)

Titulo	Nombre	
Bzr Music Session n45	Bzrp	Editar Detalles Borrar
Koyaanisqatsi (Soundtrack from the Motion Picture)	Philip Glass Ensemble	Editar Detalles Borrar
Mozart: Chamber Music	Nash Ensemble	Editar Detalles Borrar
Monteverdi: L'Orfeo	C. Monteverdi, Nigel Rogers - Chiaroscuro; London Baroque; London Cornett & Sackbu	Editar Detalles Borrar
Schubert: The Late String Quartets & String Quintet (3 CD's)	Emerson String Quartet	Editar Detalles Borrar
Respighi:Pines of Rome	Eugene Ormandy	Editar Detalles Borrar
Locatelli: Concertos for Violin, Strings and Continuo, Vol. 3	Mela Tenenbaum, Pro Musica Prague & Richard Kapp	Editar Detalles Borrar
Great Recordings of the Century - Shubert: Schwanengesang, 4 Lieder	Gerald Moore	Editar Detalles Borrar
Liszt - 12 Études D'Execution Transcendante	Michele Campanella	Editar Detalles Borrar
Great Recordings of the Century: Paganini's 24 Caprices	Itzhak Perlman	Editar Detalles Borrar
Nielsen: The Six Symphonies	Göteborgs Symfoniker & Neeme Järvi	Editar Detalles Borrar
Szymanowski: Piano Works, Vol. 1	Martin Roscoe	Editar Detalles Borrar

Album - Crear

HernandezJorge_Musica Inicio Discos Artistas

Login Admin

Crear

Album

Titulo

ArtistaId

AC/DC

[Crear](#)

[Volver a la lista](#)

Album - Editar

HernandezJorge_Musica Inicio Discos Artistas

Login Admin

Editar

Album

Titulo

Bzr Music Session n45

ArtistaId

276

[Guardar Cambios](#)

[Volver a la lista](#)

Jorge Hernández Toledo
2ºDAW - A

Album 1.02

Album - Detalles

HernandezJorge_Musica Inicio Discos Artistas

Login Admin

Detalles

Album

Título For Those About To Rock We Salute You
Artista AC/DC

Listado de canciones del album:

- For Those About To Rock (We Salute You)
- Put The Finger On You
- Let's Get It Up
- Inject The Venom
- Snowballed
- Evil Walks
- C.O.D.
- Breaking The Rules
- Night Of The Long Knives
- Spellbound

[Editar](#) | [volver a la lista](#)

Album - Borrar

HernandezJorge_Musica Inicio Discos Artistas

Login Admin

Borrar

¿ Está seguro de que desea borrar esto?

Album

Título For Those About To Rock We Salute You
Artista AC/DC

[Borrar](#) | [Volver a la lista](#)

Artistas 1.01

Artistas - Index

HernandezJorge_Musica Inicio Discos Artistas

Login Admin

Listado de artistas

[Crear nuevo artista](#)

Nombre	
zzzooolander	Editar Detalles Borrar Discos del artista
Zeca Pagodinho	Editar Detalles Borrar Discos del artista
Yo-Yo Ma	Editar Detalles Borrar Discos del artista
Yousseou N'Dour	Editar Detalles Borrar Discos del artista
Yehudi Menuhin	Editar Detalles Borrar Discos del artista
Xis	Editar Detalles Borrar Discos del artista
Wilhelm Kempff	Editar Detalles Borrar Discos del artista
Whitesnake	Editar Detalles Borrar Discos del artista
Vinicius, Toquinho & Quarteto Em Cy	Editar Detalles Borrar Discos del artista
Vinicius E Quarteto Em Cy	Editar Detalles Borrar Discos del artista
Vinicius E Odette Lara	Editar Detalles Borrar Discos del artista
Vinicius De Moraes & Baden Powell	Editar Detalles Borrar Discos del artista
Vinicius De Moraes	Editar Detalles Borrar Discos del artista
Velvet Revolver	Editar Detalles Borrar Discos del artista
Various Artists	Editar Detalles Borrar Discos del artista

Artistas - Crear

HernandezJorge_Musica Inicio Discos Artistas

Login Admin

Crear

Artista

Nombre

[Crear](#)

[Volver a la lista](#)

Artistas - Editar

HernandezJorge_Musica Inicio Discos Artistas

Login Admin

Editar

Artista

Nombre

zzzooolander

[Guardar](#)

[Back to List](#)

Jorge Hernández Toledo
2ºDAW - A

Artistas 1.02

Artistas - Detalles

HernandezJorge_Musica Inicio Discos Artistas

Login Admin

Detalles

Artista

Nombre zzzoolander

Comentarios [Crear](#)

Título ★★★★★

Descripción detallada de ejemplo

[Editar](#) | [Volver a la lista](#)

Artistas - Borrar

HernandezJorge_Musica Inicio Discos Artistas

Login Admin

Borrar

¿Está seguro de borrar este artista?

Artist

Nombre zzzoolander

[Borrar](#) | [Volver a la lista](#)

Artistas - Discos del Artista

HernandezJorge_Musica Inicio Discos Artistas

Login Admin

Discos del Artista

- For Those About To Rock We Salute You
- Let There Be Rock

[Volver a la lista](#)

Comentarios

Comentarios - Crear

HernandezJorge_Musica Inicio Discos Artistas

Login Admin

Crear Comentario

Artist

Título

Comentario

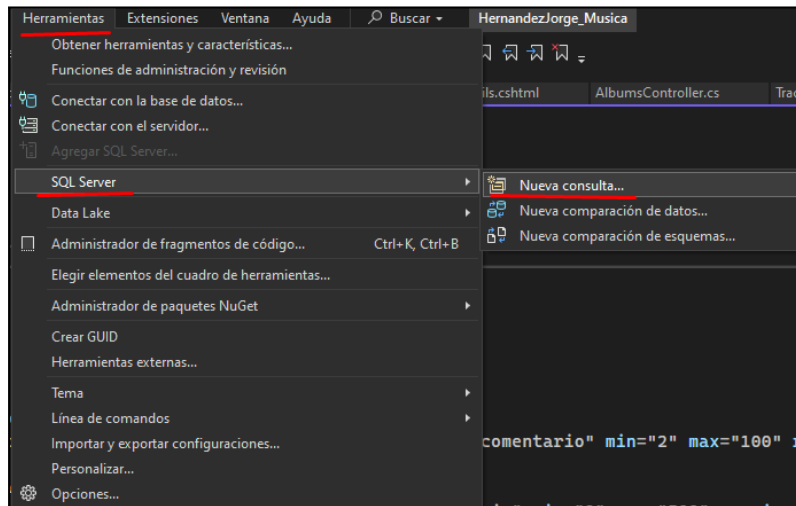
Puntuación

[Comentar](#)

[Volver a la lista](#)

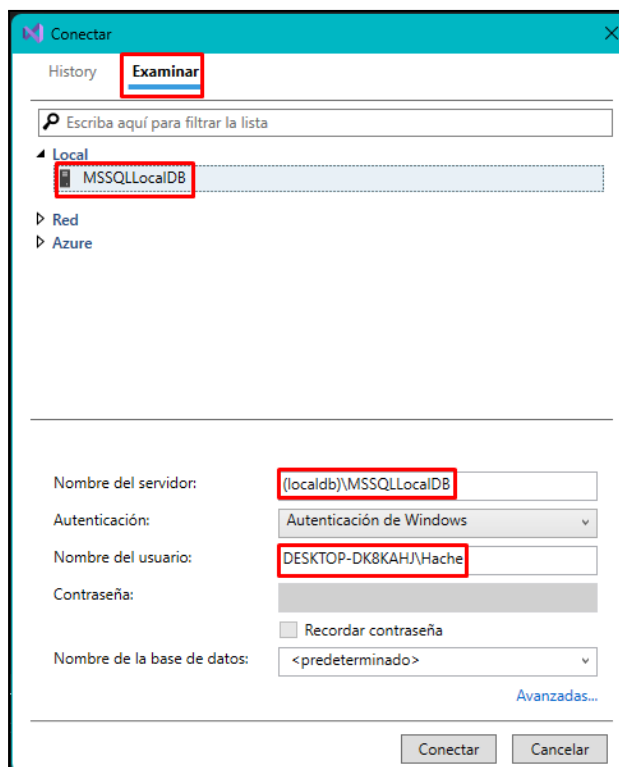
Jorge Hernández Toledo
2ºDAW - A

Insertar la base de datos para extraer datos para el proyecto



Tenemos que usar la base de datos local del sistema.

Esto supone un problema, y es que los cambios que hagamos, no los verá otra persona si no usa el mismo dispositivo. Por ejemplo, el profesor correspondiente no verá los cambios que hemos hecho, pero sí podrá ver los que el/ella haga en su ordenador.



Scaffolding a la base de datos para generar los modelos y controladores.

```
Scaffold-DbContext "Server=(localdb)\MSSQLLocalDB; Database=Chinook; Trusted_Connection=True" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models -Tables Album,Artist,Track -ContextDir Data
```

Agregar el contexto de la base de datos.

The image shows two screenshots from a Visual Studio IDE. The top screenshot displays the `appsettings.json` file with the `ConnectionStrings` section highlighted, showing the connection string for the `ChinookContext`. The bottom screenshot displays the `Program.cs` file with the `builder.Services.AddDbContext` call highlighted, showing the configuration of the `ChinookContext`.

```
appsettings.json: {
  "ConnectionStrings": {
    "ChinookContext": "Server=(localdb)\\MSSQLLocalDB; Database=Chinook; Trusted_Connection=True"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

```
Program.cs:
using HernandezJorge_Musica.Data;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddControllersWithViews();

builder.Services.AddDbContext<ChinookContext>(options =>
{
    options.UseSqlServer(builder.Configuration.GetConnectionString("DbContext"));
});
```

Al pulsar en Discos aparece la lista de los discos ordenados de forma descendente por número de id y sacando 15 registros.

```
// GET: Albums
3 referencias
public async Task<IActionResult> Index()
{
    // Consulta los 15 discos más recientes, ordenados de forma descendente por AlbumId.
    var albums = _context.Albums.Include(a => a.Artist).OrderByDescending(a => a.AlbumId).Take(15);

    return View(albums);
}
```

He modificado el método que muestra el Index de Albums.

Esta linea de código crea una nueva variable llamada albums, le paso la clase de contexto y le digo que todo el contenido que lleve dentro, lo ordene descendientemente por ID y que solo coja 15 de ellos.

Solo lista con 15 registros ordenados de manera descendente por nombre (Si añades añade artistas que empiecen por z).

```
// GET: Artists
// GET: Artists
3 referencias
public async Task<IActionResult> Index()
{
    var artists = await _context.Artists
        .OrderByDescending(a => a.Name)
        .Take(15)
        .ToListAsync();

    return View(artists);
}
```

La función es parecida a la anterior.

Esta crea la variable artistas que realiza una consulta a la base de datos. Ordena por nombre descendientemente (alfabéticamente) y recoge los 15 primeros datos.

Finalmente, la consulta se ejecuta de manera asincrónica y se convierte en una lista de objetos Artist.

Se tendrá que crear un nuevo modelo propio llamado Review para poder realizar comentarios sobre los artistas. Este nuevo modelo tendrá que tener al menos los campos necesarios para registrar un título, descripción y valoración.

```
<label for="Titulo">Título</label>
<input type="text" value="" id="titulo" placeholder="Escribe aquí el título del comentario" min="2" max="100" required />
<br />
<label for="Comentario">Comentario</label>
<input type="text" value="" id="Comentario" placeholder="Escribe aquí el comentario" min="0" max="500" required />
<br />
<label for="Comentario">Puntuación</label>
<select>
  <option value="0">0</option>
  <option value="0.5">0.5</option>
  <option value="1">1</option>
  <option value="1.5">1.5</option>
  <option value="2">2</option>
  <option value="2.5">2.5</option>
  <option value="3">3</option>
  <option value="3.5">3.5</option>
  <option value="4">4</option>
  <option value="4.5">4.5</option>
  <option value="5">5</option>
</select>
<br />
<a asp-action="Index" asp-route-id="@Model?.ArtistId">
  <button style="padding: 5px; color: white; background: dodgerblue; border: 0px solid; border-radius: 5px;">Comentar</button>
</a>
```

```
0 referencias
public async Task<IActionResult> CrearComentarioAsync(int id)
{
    if (id == null || _context.Artists == null)
    {
        return NotFound();
    }

    var artist = await _context.Artists
        .FirstOrDefaultAsync(m => m.ArtistId == id);
    if (artist == null)
    {
        return NotFound();
    }

    return View();
}
```

Algoritmo de generación de estrellas:

```
<section style="border: 1px solid gray; padding: 10px;">
  <table>
    <tr>
      <td>
        <h4>Titulo</h4>
      </td>
      <td>
        @* Algoritmo para mostrar estrellas: *@
        @{
          decimal Puntuacion = 2.5M; //El numero de estrellas a mostrar.
          int PuntuacionTruncada = Convert.ToInt32(Math.Truncate(Puntuacion));

          <div style="display: flex; flex-gap: 5px;">
            @if (Puntuacion == PuntuacionTruncada) //Si la puntuacion introducida es igual a la truncada...
            {
              for(int i = 0; i < Puntuacion; i++)
              {
                 //Solo imprime tantas estrellas enteras como input haya.
              }
            }
            else
            {
              for(int i = 1; i <= PuntuacionTruncada; i++) //En caso contrario...
              {
                 //Imprime todas hasta la truncada...
              }
               //E imprime una ultima con la mitad.
            }
          </div>
        }
      </td>
    </tr>
  </table>
</section>
```

He declarado una variable donde “recoge” la puntuación del formulario. Pero en este caso, al ser solamente front, es una variable estática pero que recomiendo ir cambiando para comprobar su funcionamiento.

Una vez se recoge la puntuacion, la trunca para saber hasta cuanto hay que contar.

Si la puntuacion recogida y la truncada son iguales, se generan tantas estrellas como puntuacion haya.

En caso contrario, se hace lo mismo que en el anterior paso, pero al final se le añade media estrella.

Este algoritmo se me ocurrió en clase y lo hice a mano antes de plasmarlo.

Desafortunadamente, no es algo que el profesorado vaya a valorar, ya que no cumple con el requisito principal, que es almacenar el comentario en la base de datos.

Borrado en cascada

Estos son los pasos que he seguido para poder realizar un borrado en cascada para los artistas.

Algoritmo para borrar en cascada - ArtistsController

```
// POST: Artists/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
0 referencias
public async Task<IActionResult> DeleteConfirmed(int id)
{
    if (_context.Artists == null)
    {
        return Problem("Entity set 'ChinookContext.Artists' is null.");
    }
    var artist = await _context.Artists.FindAsync(id);
    if (artist != null)
    {
        var albumDel = _context.Albums.Where(a => a.ArtistId == id); //Guardo los albums con el id del artista
        _context.Albums.RemoveRange(albumDel); //Ahora borro todos los albums con ese ID de la lista.
        _context.Artists.Remove(artist); // Y ahora borro el artista, que ya no tiene ningun album asociado y no da error.
    }

    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}
```

Vista para borrar un artista - Artist - Delete.cshtml

```
<h1>Borrar</h1>
<h3>¿Está seguro de borrar este artista?</h3>
<h4 style="color: red;">Tambien se borrarán los discos del artista.</h4>
<div>
    <h4>Artista</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Name)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Name)
        </dd>
    </dl>
    <form asp-action="Delete">
        <input type="hidden" asp-for="ArtistId" />
        <input type="submit" value="Borrar" class="btn btn-danger" /> |
        <a asp-action="Index">Volver a la lista</a>
    </form>
</div>
```

He añadido un mensaje de que se borrarán los albums que tenga el artista. En rojo, para indicarle al usuario que puede estar haciendo algo que no quiere.

Vista real en la web

HernandezJorge_Musica Inicio Discos Artistas [Login](#) [Admin](#)

Borrar

¿Está seguro de borrar este artista?

Tambien se borrarán los discos del artista.

Artista

Nombre Bzrp

[Borrar](#) | [Volver a la lista](#)

Jorge Hernández Toledo
2ºDAW - A

Mostrar el listado de cada Album

Vistazo general

Detalles

Album

Título For Those About To Rock We Salute You

Artista AC/DC

Listado de canciones del album:

- For Those About To Rock (We Salute You)
- Put The Finger On You
- Let's Get It Up
- Inject The Venom
- Snowballed
- Evil Walks
- C.O.D.
- Breaking The Rules
- Night Of The Long Knives
- Spellbound

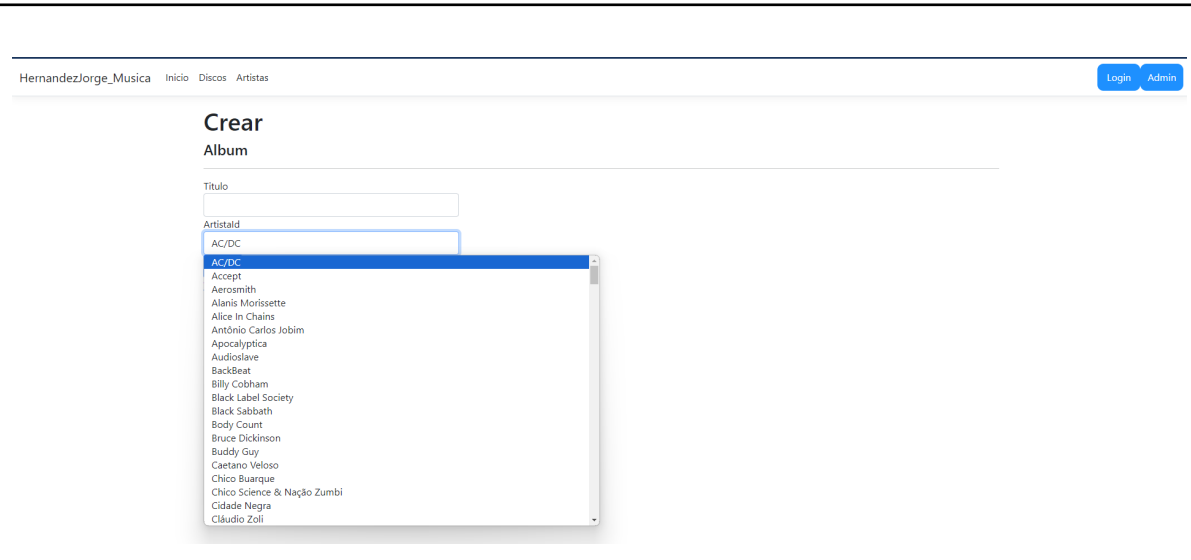
[Editar](#) | [volver a la lista](#)

Código - Albums - Details.cshtml

```
<h2>Listado de canciones del album: </h2> @*Algoritmo que muestra los nombres de las canciones de un album mediante un foreach.*@
<ul>
    @foreach (var item in Model.Tracks)
    {
        <li>@item.Name</li>
    }
</ul>
```

Mostrar el nombre del artista al crear un album.

Vistazo general



HernandezJorge_Musica Inicio Discos Artistas Login Admin

Crear Album

Título

ArtistaId

- AC/DC
- Accept
- Aerosmith
- Alanis Morissette
- Alice In Chains
- António Carlos Jobim
- Apocalyptica
- Audioslave
- BackBeat
- Billy Cobham
- Black Label Society
- Black Sabbath
- Body Count
- Bruce Dickinson
- Buddy Guy
- Caetano Veloso
- Chico Buarque
- Chico Science & Nação Zumbi
- Cidade Negra
- Cláudio Zoli

Código - Albums - Create.cshtml

```
<h1>Crear</h1>
<h4>Album</h4>
<hr />
<div class="row">
  <div class="col-md-4">
    <form asp-action="Create">
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>
      <div class="form-group">
        <label asp-for="Title" class="control-label"></label>
        <input asp-for="Title" class="form-control" min="2" max="100" required/>
        <span asp-validation-for="Title" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="ArtistId" class="control-label"></label>
        <select asp-for="ArtistId" class="form-control" asp-items="ViewBag.ArtistId"></select>
      </div>
      <div class="form-group">
        <input type="submit" value="Crear" class="btn btn-primary" />
      </div>
    </form>
  </div>
</div>
```

Problemas detectados

El primer problema y el cual me hizo borrar varias veces la práctica, fue crear las vistas antes que el controlador.

Puede parecer raro e incluso imposible... Pero de verdad, he aprendido a leerme las guías paso a paso antes de lanzarme al código.

El segundo problema fue el contexto de la BD. Tuve un error de escritura a la hora de insertarlo en el código y no reconocía la base de datos.

El tercer problema que encontré, fue no haber metido las tablas que necesitaba al crear el programa. Hice mal en usar solo Album y Artist.

Es necesario hacer un análisis general a la base de datos y al enunciado antes de lanzarse al código. Ver las conexiones, etc.

Y por último, las claves foráneas en Artistas dentro de la base de datos.

Resulta que es imposible borrar únicamente al artista si ya tiene un disco asociado.

Para ello, he tenido que desarrollar un borrado en cascada, que primero borra los álbums que tenga el id del artista seleccionado. Y una vez borrados, se puede borrar el artista porque no entra en conflicto. Es una solución no muy compleja, pero útil. Obviamente, he de avisar al usuario de que se borrarán todos los álbums de ese artista.

Algoritmo para generar las estrellas

Aunque el profesorado no llega a valorar la generación automática de estrellas para los comentarios publicados, he querido diseñar un pequeño algoritmo que lo haga.

Gracias al conocimiento que de programación y de front, sumado a una gran creatividad...

Hice este algoritmo en "pseudo-código" a mano en clase para luego implementarlo en el código.

Algoritmo de puntuaciones

Crear

Título: String → display <h3>
Comentario: String → display <p>
value: ~~double~~ double → display <select>

↓

En primer lugar truncamos la puntuación
↳ Math.Truncate(Puntuación) = PuntuaciónEntera

Ahora comprobamos:

```
if (PuntuaciónEntera == Puntuación) {  
    for for (int i = 0; i < PuntuaciónEntera; i++) {  
        <img src = "Estrella">  
    }  
else {  
    una estrella hasta el elemento truncado y al final  
    del bucle, imprimir media estrella.
```

Fin del informe

Enlace a github en caso de corrupción de archivos en el zip o evagd

https://github.com/JorgeHdezT/HernandezJorge_Musica.git

Realizado por Jorge Hernández Toledo

2ºDAW - A

Jorge Hernández Toledo

2ºDAW - A