REPRODUCTOR MÚSICA

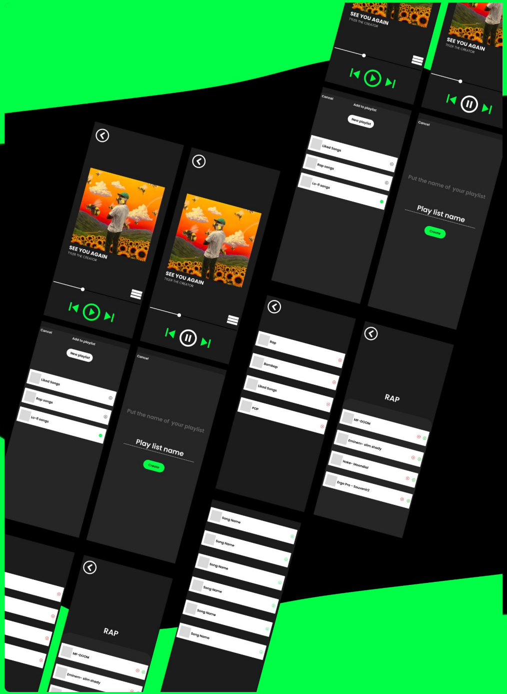
YOUSSEF FDIL
DENÍS POUTCHNINE
MARC GÜELL
JAN ESTRADA

INDEX

ANDROID

SEE YOU AGAIN
TYLER THE CREATOR
SEE YOU AGAIN
TYLER THE CREATOR
Rap songs
Lo-fi songs
Put the name of you
Play list name
Create
Add to playlist
New playlist
Cancel
Rap
Bombap

(UI) Interficie



Paleta de colors

🟩	🟥	⬜	⬛
00FF47	FF0000	F4F4F4	000000

FONT: POPPINS

THIN

EXTRA LIGHT

LIGHT

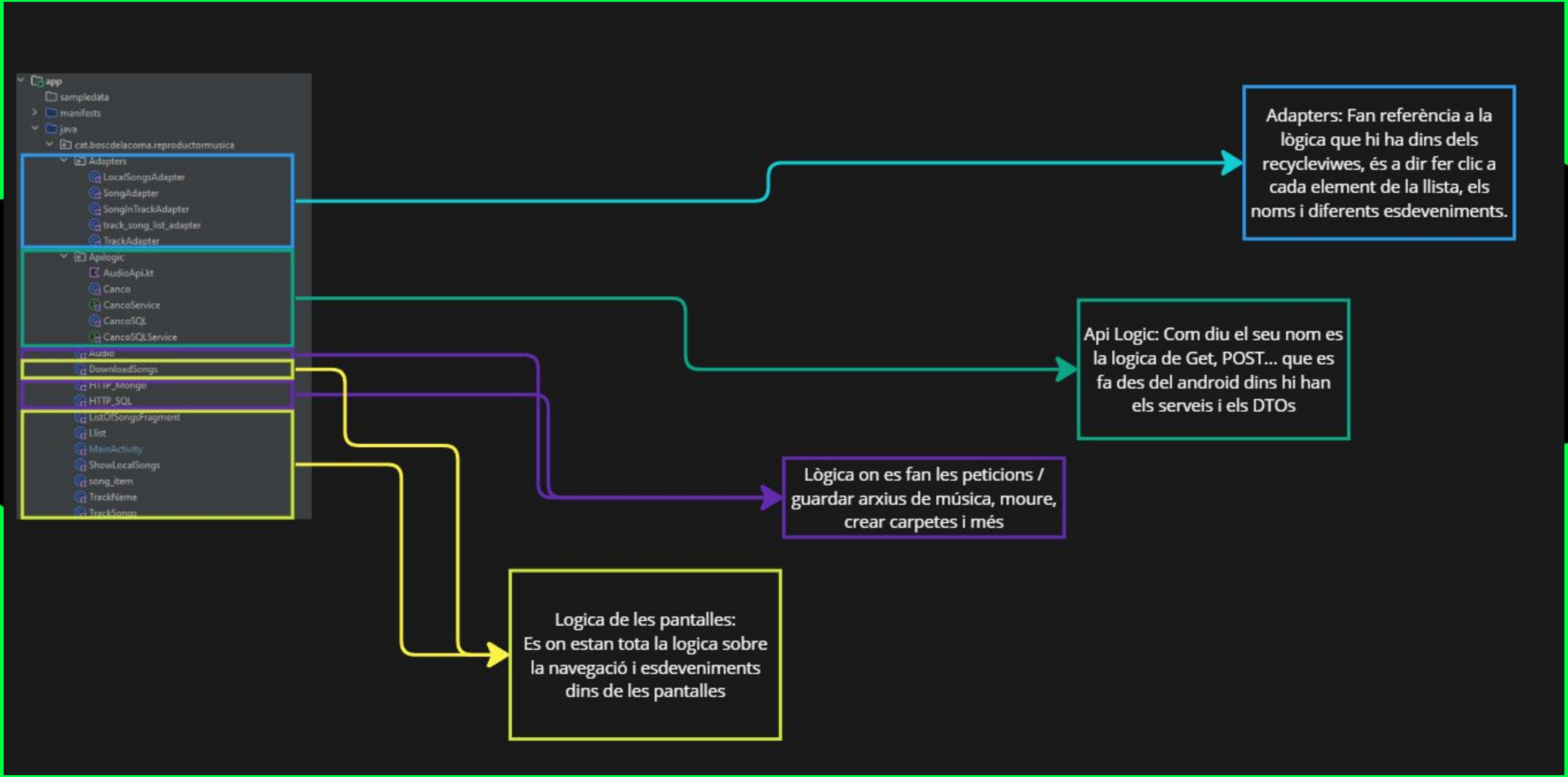
REGULAR

MEDIUM

SEMI-BOLD

BOLD

ESTRUCTURA DE CARPETAS



Adapters: Fan referència a la lògica que hi ha dins dels recycleviwes, és a dir fer clic a cada element de la llista, els noms i diferents esdeveniments.

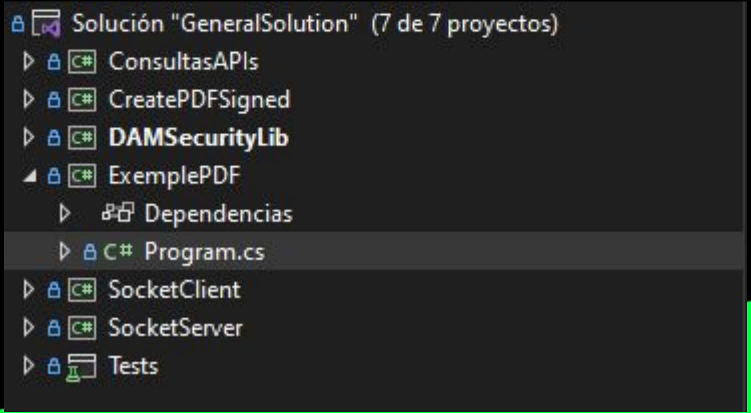
Api Logic: Com diu el seu nom es la logica de Get, POST... que es fa des del android dins hi han els serveis i els DTOs

Lògica on es fan les peticions / guardar arxius de música, moure, crear carpetes i més

Logica de les pantalles:
Es on estan tota la logica sobre la navegació i esdeveniments dins de les pantalles

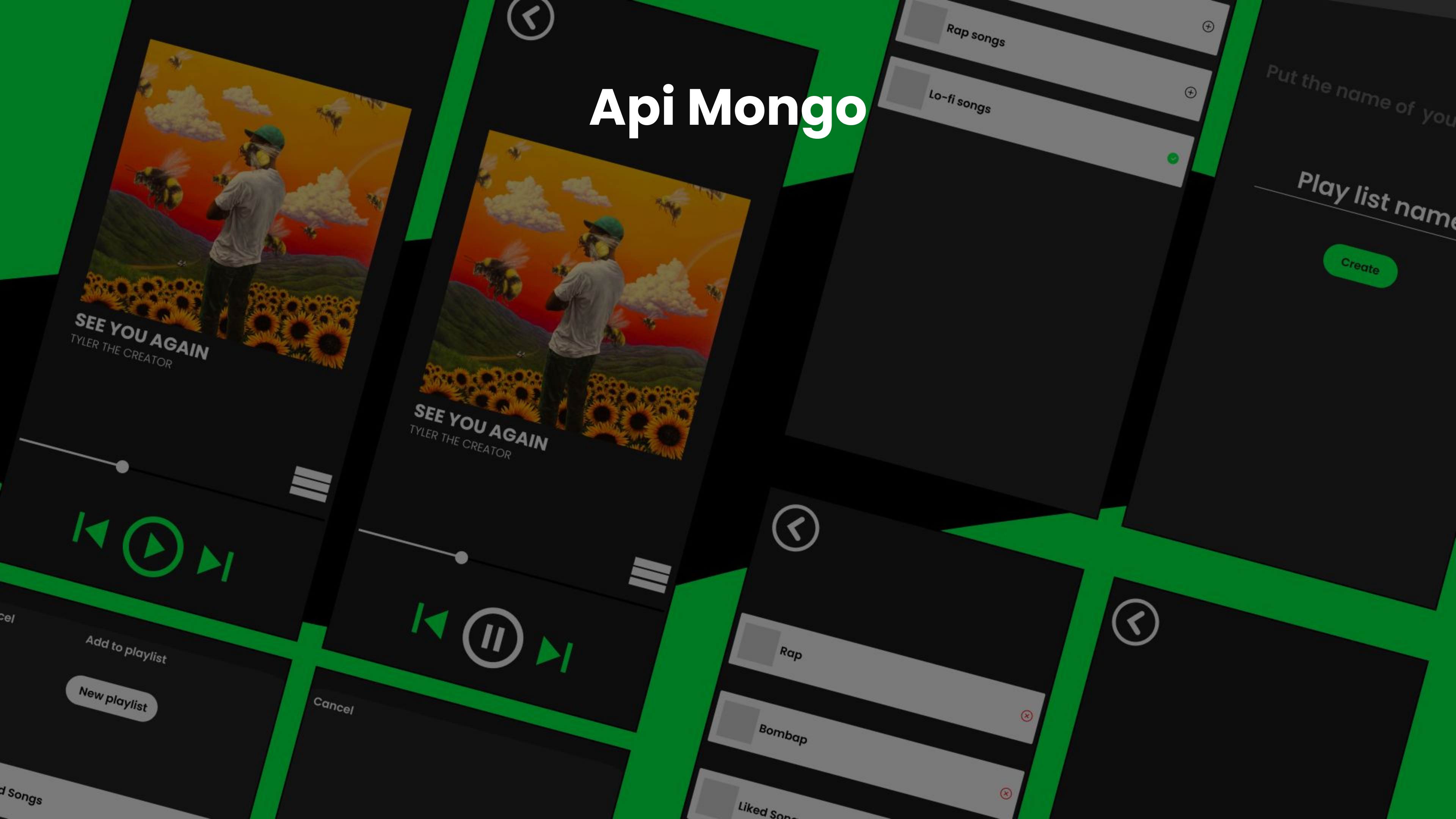
Criptografia

PLANTEJAMENT

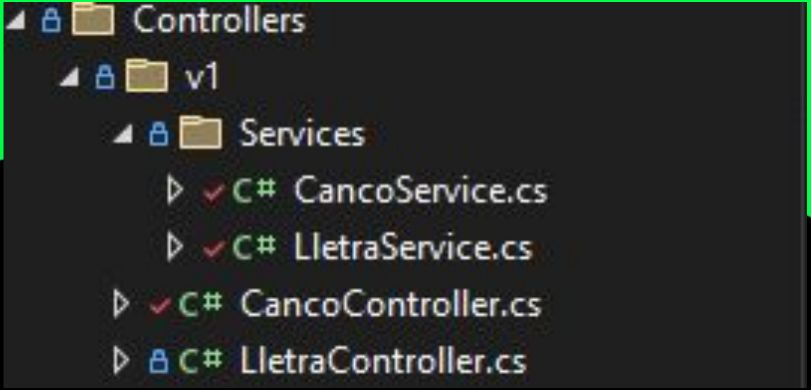


```
Solución "GeneralSolution"  (7 de 7 proyectos)
  ConsultasAPIs
  CreatePDFSigned
  DAMSecurityLib
  ExemplePDF
    Dependencias
    Program.cs
  SocketClient
  SocketServer
  Tests
```

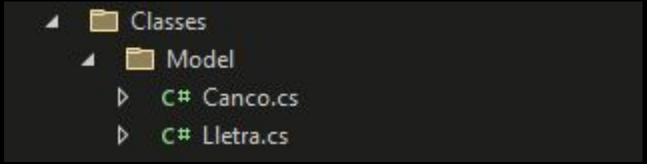
Api Mongo


Rap songs
Lo-fi songs
Put the name of you
Play list name
Create
SEE YOU AGAIN
TYLER THE CREATOR
SEE YOU AGAIN
TYLER THE CREATOR
cel
Add to playlist
New playlist
d Songs
Cancel
Rap
Bombap
Liked Son

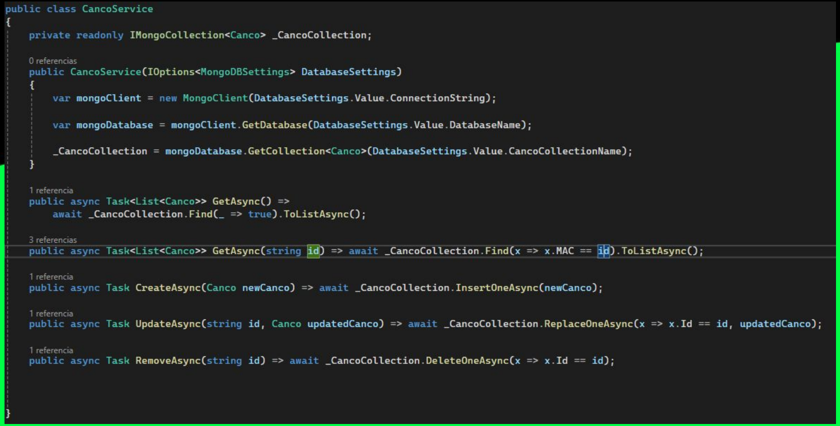
PLANTEJAMENT



```
Controllers
    v1
        Services
            C# CancoService.cs
            C# LletraService.cs
        C# CancoController.cs
        C# LletraController.cs
```

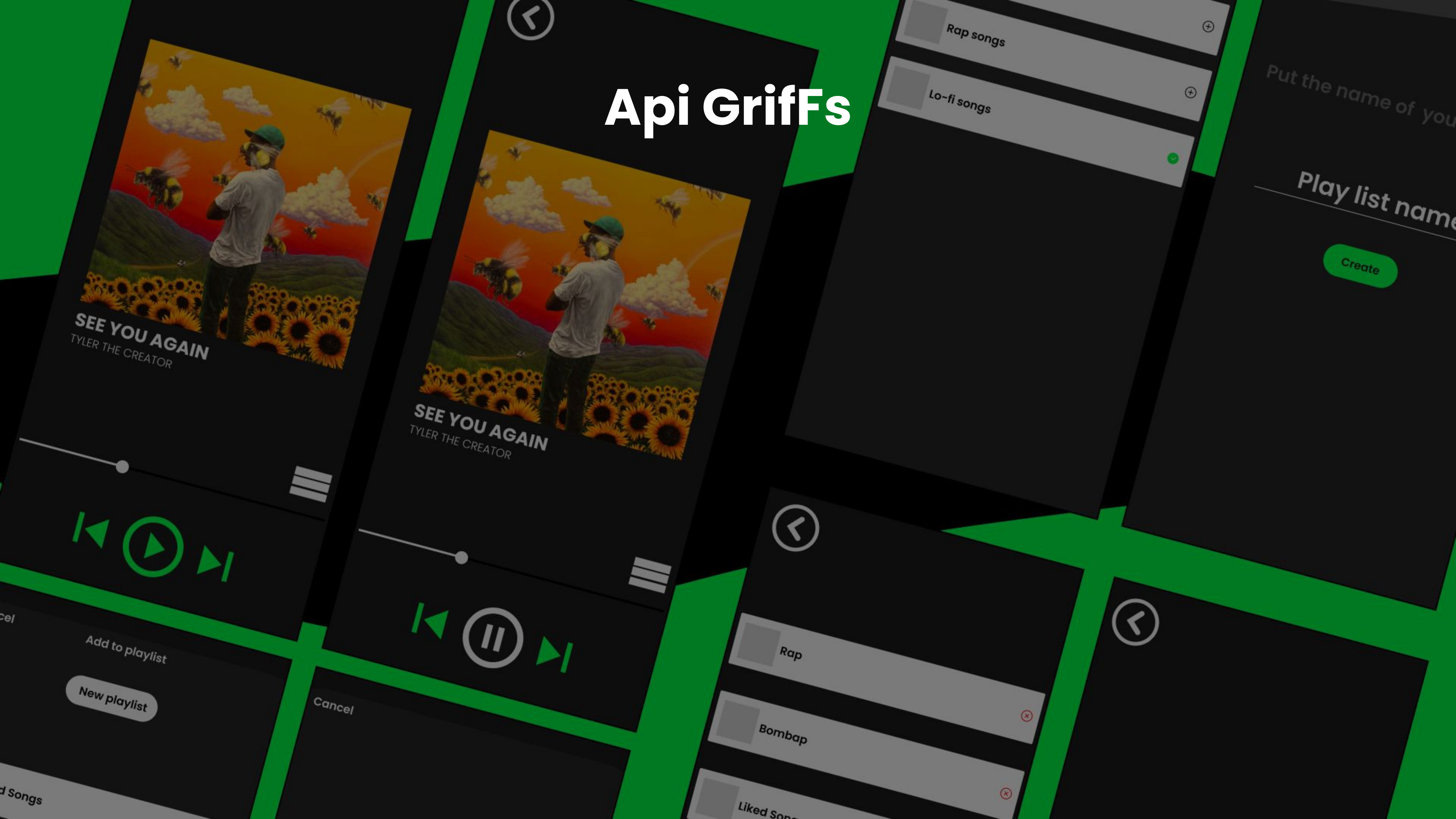


```
Classes
    Model
        C# Canco.cs
        C# Lletra.cs
```

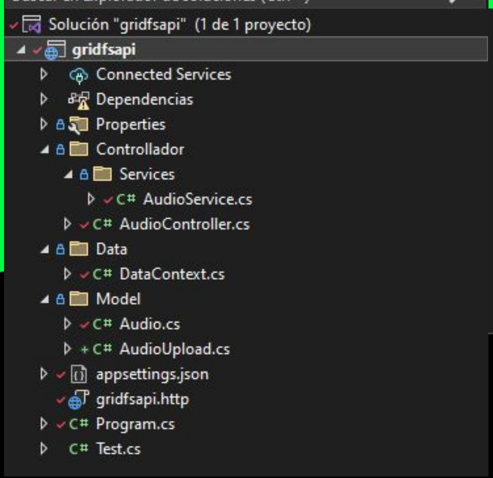


```
public class CancoService
{
    private readonly IMongoCollection<Canco> _CancoCollection;

    0 referencias
    public CancoService(IOptions<MongoDBSettings> DatabaseSettings)
    {
        var mongoClient = new MongoClient(DatabaseSettings.Value.ConnectionString);

        var mongoDatabase = mongoClient.GetDatabase(DatabaseSettings.Value.DatabaseName);

        _CancoCollection = mongoDatabase.GetCollection<Canco>(DatabaseSettings.Value.CancoCollectionName);
    }

    1 referencia
    public async Task<List<Canco>> GetAsync() =>
        await _CancoCollection.Find(_ => true).ToListAsync();

    3 referencias
    public async Task<List<Canco>> GetAsync(string id) => await _CancoCollection.Find(x => x.MAC == id).ToListAsync();

    1 referencia
    public async Task CreateAsync(Canco newCanco) => await _CancoCollection.InsertOneAsync(newCanco);

    1 referencia
    public async Task UpdateAsync(string id, Canco updatedCanco) => await _CancoCollection.ReplaceOneAsync(x => x.Id == id, updatedCanco);

    1 referencia
    public async Task RemoveAsync(string id) => await _CancoCollection.DeleteOneAsync(x => x.Id == id);


}
```

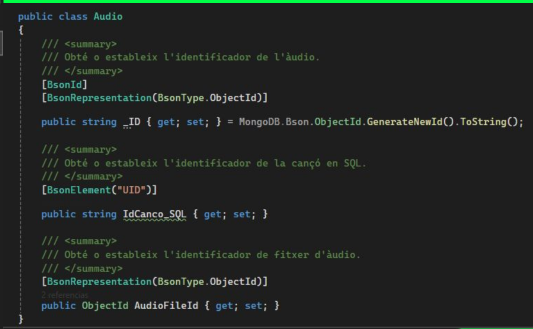
Api GrifFs


SEE YOU AGAIN
TYLER THE CREATOR
SEE YOU AGAIN
TYLER THE CREATOR
Rap songs
Lo-fi songs
Put the name of you
Play list name
Create
Add to playlist
New playlist
Cancel
Songs
Cancel
Rap
Bombap
Liked Son

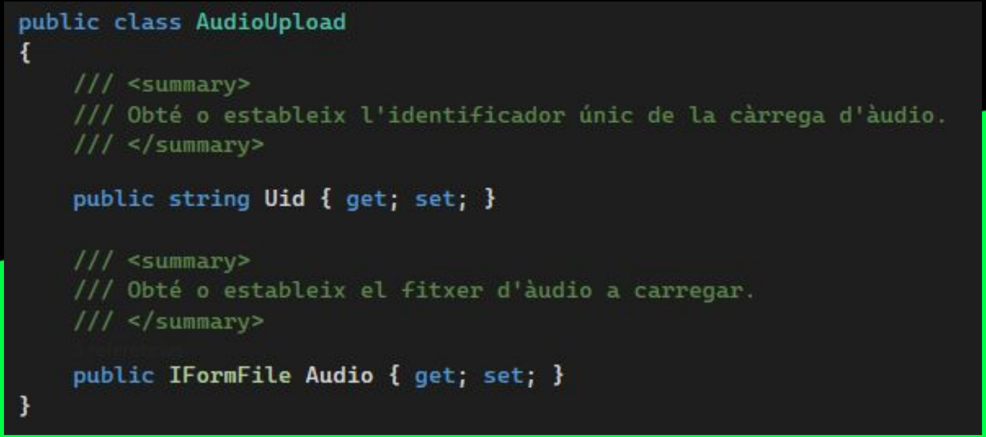
PLANTEJAMENT



```
Solución "gridfsapi" (1 de 1 proyecto)
  gridfsapi
    Connected Services
    Dependencias
    Properties
    Controllador
      Services
        AudioService.cs
      AudioController.cs
    Data
      DataContext.cs
    Model
      Audio.cs
      AudioUpload.cs
    appsettings.json
    gridfsapi.http
    Program.cs
    Test.cs
```



```csharp
public class Audio
{
    /// <summary>
    /// Obté o estableix l'identificador de l'àudio.
    /// </summary>
    [BsonId]
    [BsonRepresentation(BsonType.ObjectId)]

    public string _ID { get; set; } = MongoDB.Bson.ObjectId.GenerateNewId().ToString();

    /// <summary>
    /// Obté o estableix l'identificador de la cançó en SQL.
    /// </summary>
    [BsonElement("UID")]

    public string IdCanco_SQL { get; set; }

    /// <summary>
    /// Obté o estableix l'identificador de fitxer d'àudio.
    /// </summary>
    [BsonRepresentation(BsonType.ObjectId)]
    2 referencias
    public ObjectId AudioFileId { get; set; }
}
```

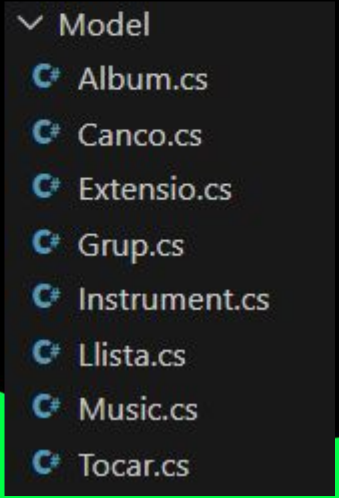


```csharp
public class AudioUpload
{
    /// <summary>
    /// Obté o estableix l'identificador únic de la càrrega d'àudio.
    /// </summary>

    public string Uid { get; set; }

    /// <summary>
    /// Obté o estableix el fitxer d'àudio a carregar.
    /// </summary>

    public IFormFile Audio { get; set; }
}
```

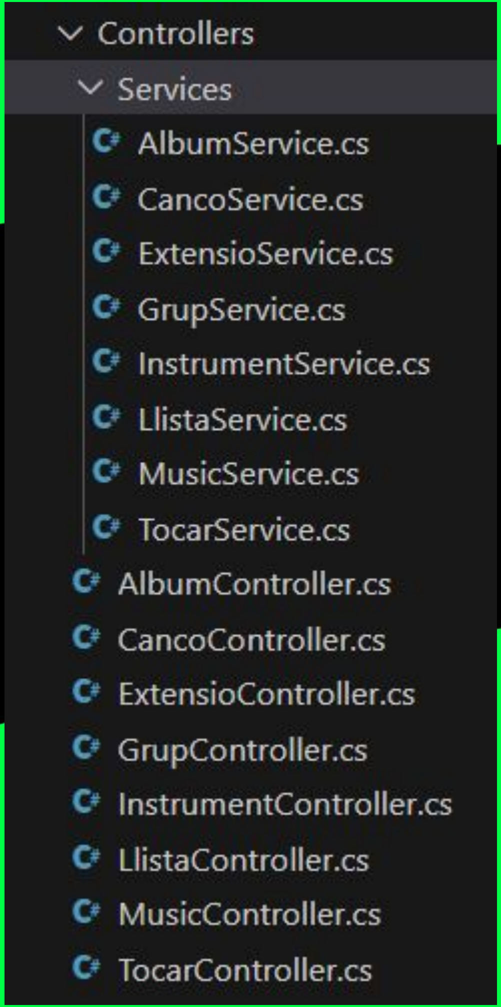
Api SQL

SEE YOU AGAIN
TYLER THE CREATOR
SEE YOU AGAIN
TYLER THE CREATOR
Rap songs
Lo-fi songs
Put the name of you
Play list name
Create
cel
Add to playlist
New playlist
d Songs
Cancel
Rap
Bombap
Liked Son

Controladors i Serveis

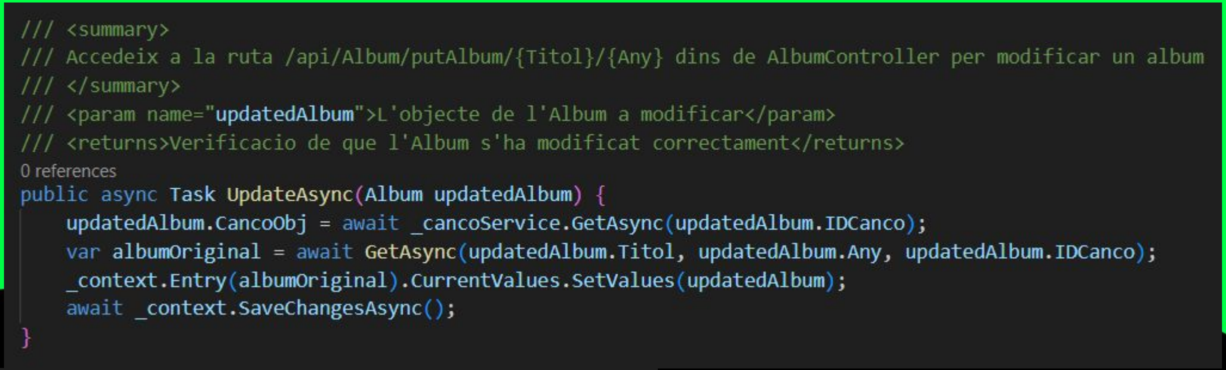


Model
- Album.cs
- Canco.cs
- Extensio.cs
- Grup.cs
- Instrument.cs
- Llista.cs
- Music.cs
- Tocar.cs

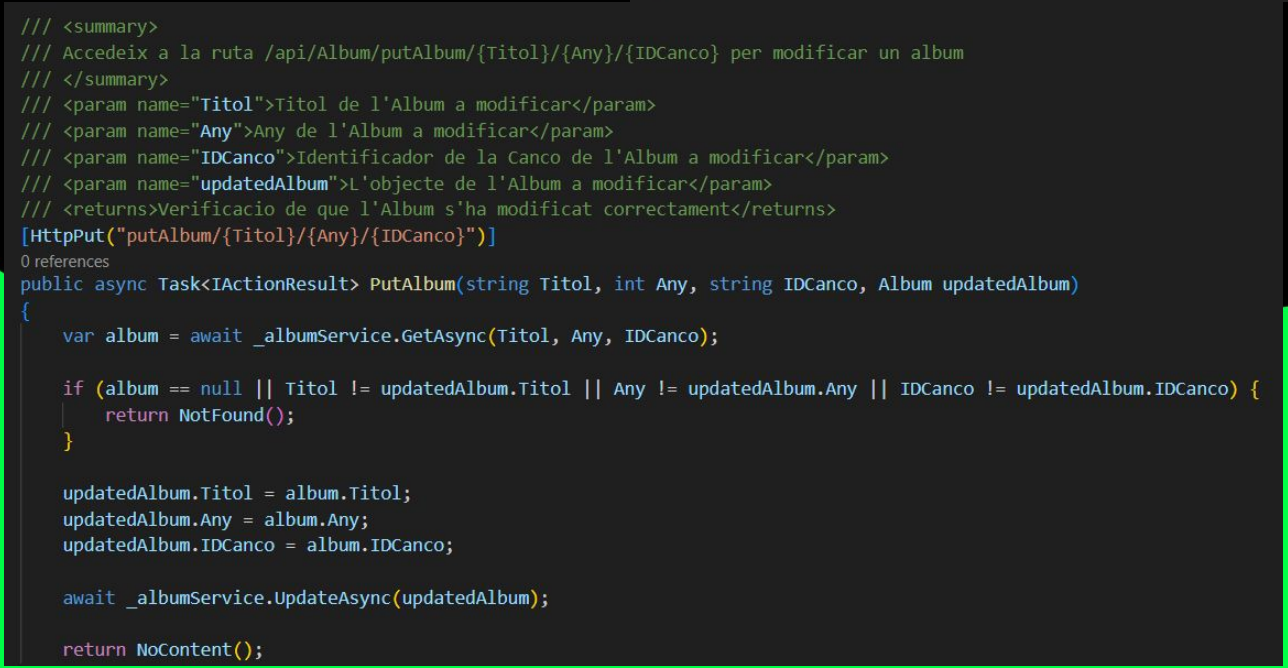


Controllers
- Services
  - AlbumService.cs
  - CancoService.cs
  - ExtensioService.cs
  - GrupService.cs
  - InstrumentService.cs
  - LlistaService.cs
  - MusicService.cs
  - TocarService.cs
- AlbumController.cs
- CancoController.cs
- ExtensioController.cs
- GrupController.cs
- InstrumentController.cs
- LlistaController.cs
- MusicController.cs
- TocarController.cs

Controladors i Serveis

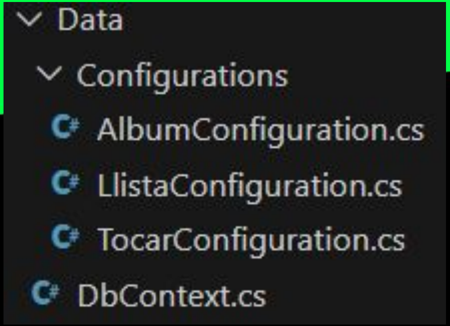


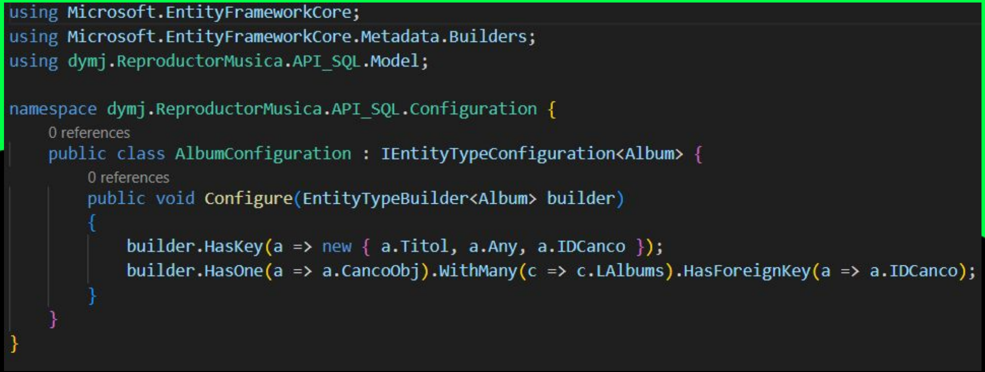
```
/// <summary>
/// Accedeix a la ruta /api/Album/putAlbum/{Titol}/{Any} dins de AlbumController per modificar un album
/// </summary>
/// <param name="updatedAlbum">L'objecte de l'Album a modificar</param>
/// <returns>Verificacio de que l'Album s'ha modificat correctament</returns>
0 references
public async Task UpdateAsync(Album updatedAlbum) {
    updatedAlbum.CancoObj = await _cancoService.GetAsync(updatedAlbum.IDCanco);
    var albumOriginal = await GetAsync(updatedAlbum.Titol, updatedAlbum.Any, updatedAlbum.IDCanco);
    _context.Entry(albumOriginal).CurrentValues.SetValues(updatedAlbum);
    await _context.SaveChangesAsync();
}
```



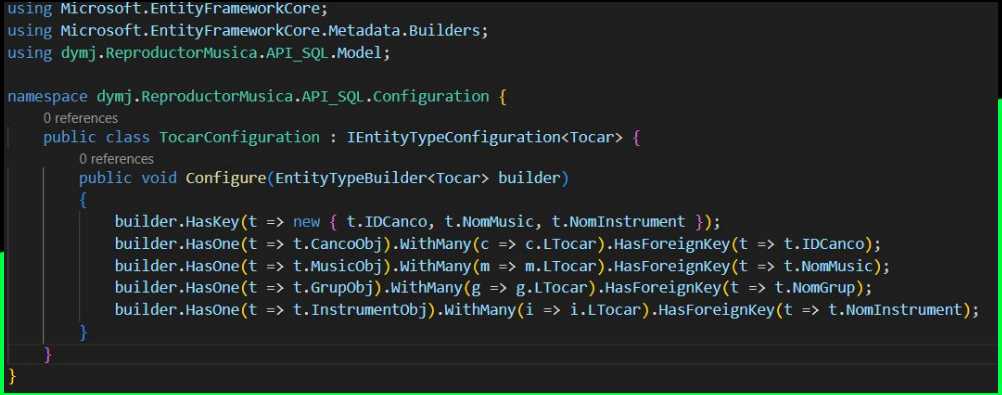
```
/// <summary>
/// Accedeix a la ruta /api/Album/putAlbum/{Titol}/{Any}/{IDCanco} per modificar un album
/// </summary>
/// <param name="Titol">Titol de l'Album a modificar</param>
/// <param name="Any">Any de l'Album a modificar</param>
/// <param name="IDCanco">Identificador de la Canco de l'Album a modificar</param>
/// <param name="updatedAlbum">L'objecte de l'Album a modificar</param>
/// <returns>Verificacio de que l'Album s'ha modificat correctament</returns>
[HttpPut("putAlbum/{Titol}/{Any}/{IDCanco}")]
0 references
public async Task<IActionResult> PutAlbum(string Titol, int Any, string IDCanco, Album updatedAlbum)
{
    var album = await _albumService.GetAsync(Titol, Any, IDCanco);

    if (album == null || Titol != updatedAlbum.Titol || Any != updatedAlbum.Any || IDCanco != updatedAlbum.IDCanco) {
        return NotFound();
    }

    updatedAlbum.Titol = album.Titol;
    updatedAlbum.Any = album.Any;
    updatedAlbum.IDCanco = album.IDCanco;

    await _albumService.UpdateAsync(updatedAlbum);

    return NoContent();
```

Configuracions

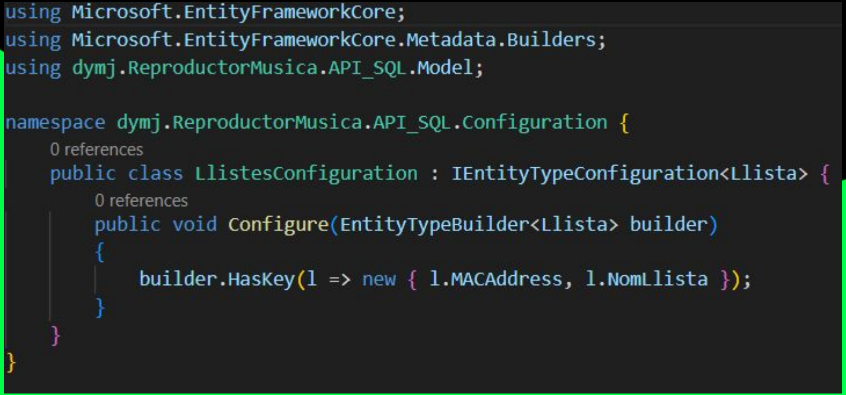




```
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata.Builders;
using dymj.ReproductorMusica.API_SQL.Model;

namespace dymj.ReproductorMusica.API_SQL.Configuration {
    0 references
    public class AlbumConfiguration : IEntityTypeConfiguration<Album> {
        0 references
        public void Configure(EntityTypeBuilder<Album> builder)
        {
            builder.HasKey(a => new { a.Titol, a.Any, a.IDCanco });
            builder.HasOne(a => a.CancoObj).WithMany(c => c.LAlbums).HasForeignKey(a => a.IDCanco);
        }
    }
}
```

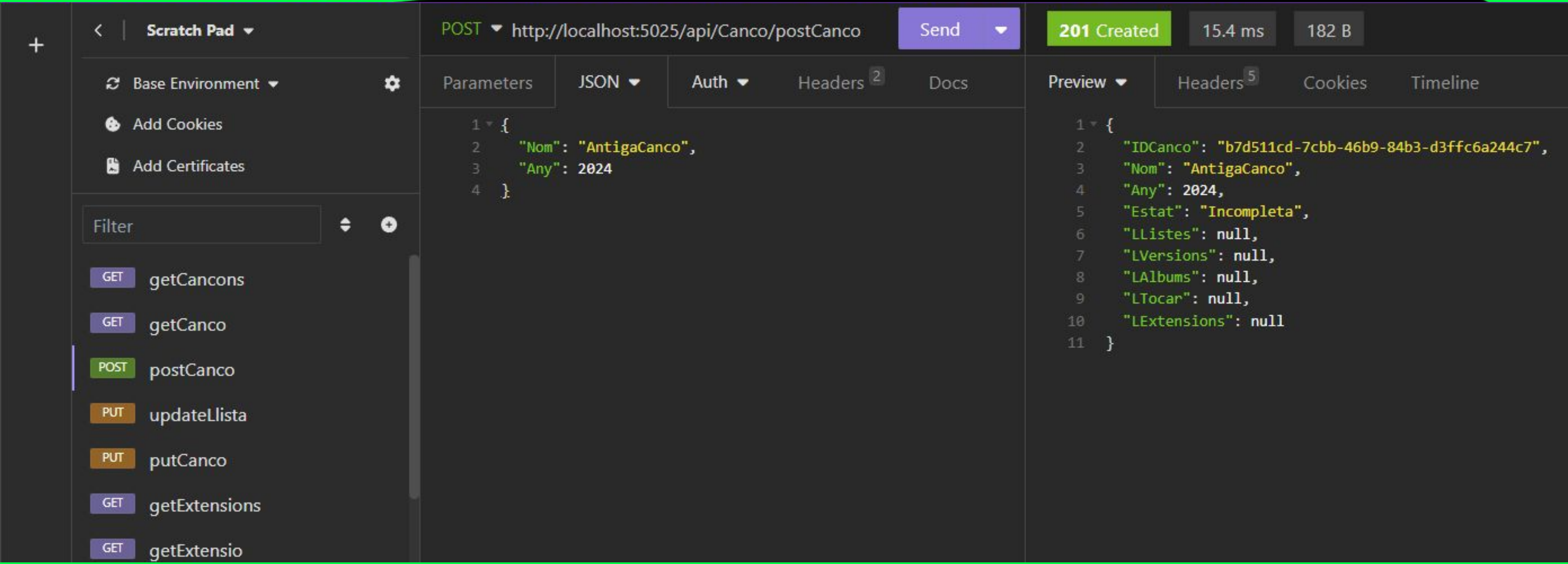


```
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata.Builders;
using dymj.ReproductorMusica.API_SQL.Model;

namespace dymj.ReproductorMusica.API_SQL.Configuration {
    0 references
    public class TocarConfiguration : IEntityTypeConfiguration<Tocar> {
        0 references
        public void Configure(EntityTypeBuilder<Tocar> builder)
        {
            builder.HasKey(t => new { t.IDCanco, t.NomMusic, t.NomInstrument });
            builder.HasOne(t => t.CancoObj).WithMany(c => c.LTocar).HasForeignKey(t => t.IDCanco);
            builder.HasOne(t => t.MusicObj).WithMany(m => m.LTocar).HasForeignKey(t => t.NomMusic);
            builder.HasOne(t => t.GrupObj).WithMany(g => g.LTocar).HasForeignKey(t => t.NomGrup);
            builder.HasOne(t => t.InstrumentObj).WithMany(i => i.LTocar).HasForeignKey(t => t.NomInstrument);
        }
    }
}
```



```
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata.Builders;
using dymj.ReproductorMusica.API_SQL.Model;

namespace dymj.ReproductorMusica.API_SQL.Configuration {
    0 references
    public class LlistesConfiguration : IEntityTypeConfiguration<Llista> {
        0 references
        public void Configure(EntityTypeBuilder<Llista> builder)
        {
            builder.HasKey(l => new { l.MACAddress, l.NomLlista });
        }
    }
}
```

Testos API_SQL



POST http://localhost:5025/api/Canco/postCanco

201 Created 15.4 ms 182 B

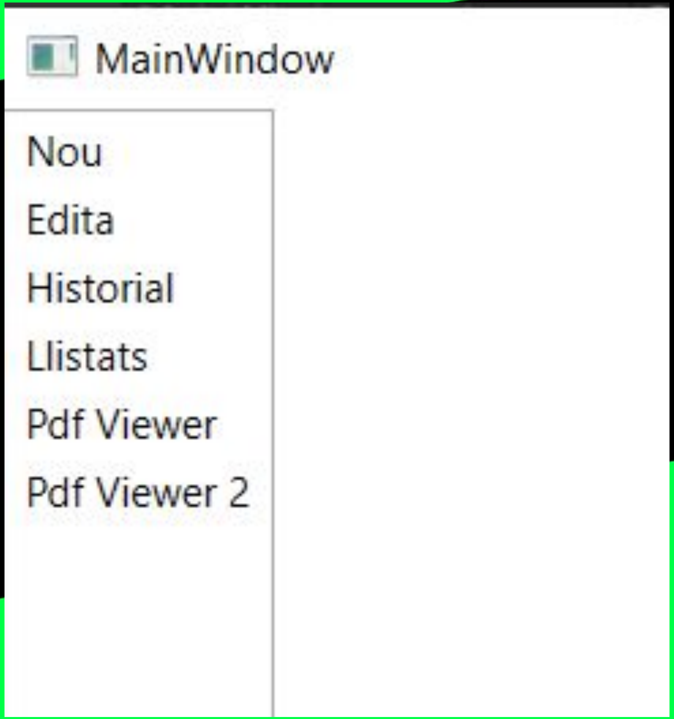
```
{
  "Nom": "AntigaCanco",
  "Any": 2024
}
```

```
{
  "IDCanco": "b7d511cd-7cbb-46b9-84b3-d3ffc6a244c7",
  "Nom": "AntigaCanco",
  "Any": 2024,
  "Estat": "Incompleta",
  "LListes": null,
  "LVersions": null,
  "LAlbums": null,
  "LTocar": null,
  "LExtensions": null
}
```

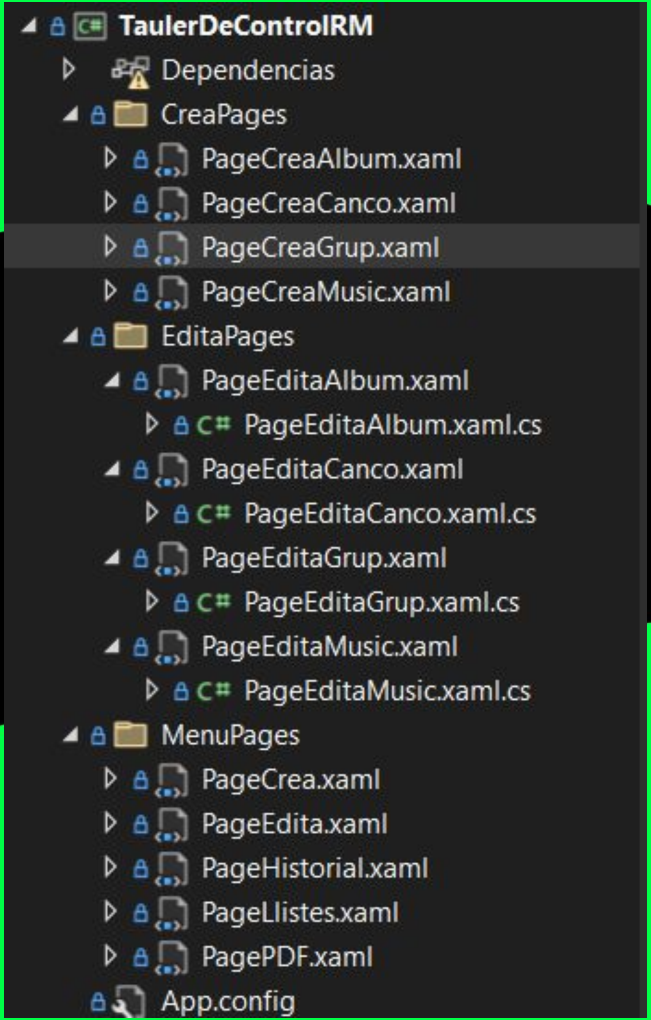
WPF

SEE YOU AGAIN
TYLER THE CREATOR
SEE YOU AGAIN
TYLER THE CREATOR
Rap songs
Lo-fi songs
Put the name of you
Play list name
Create
Add to playlist
New playlist
Cancel
Rap
Bombap
Liked Son

MENÚ

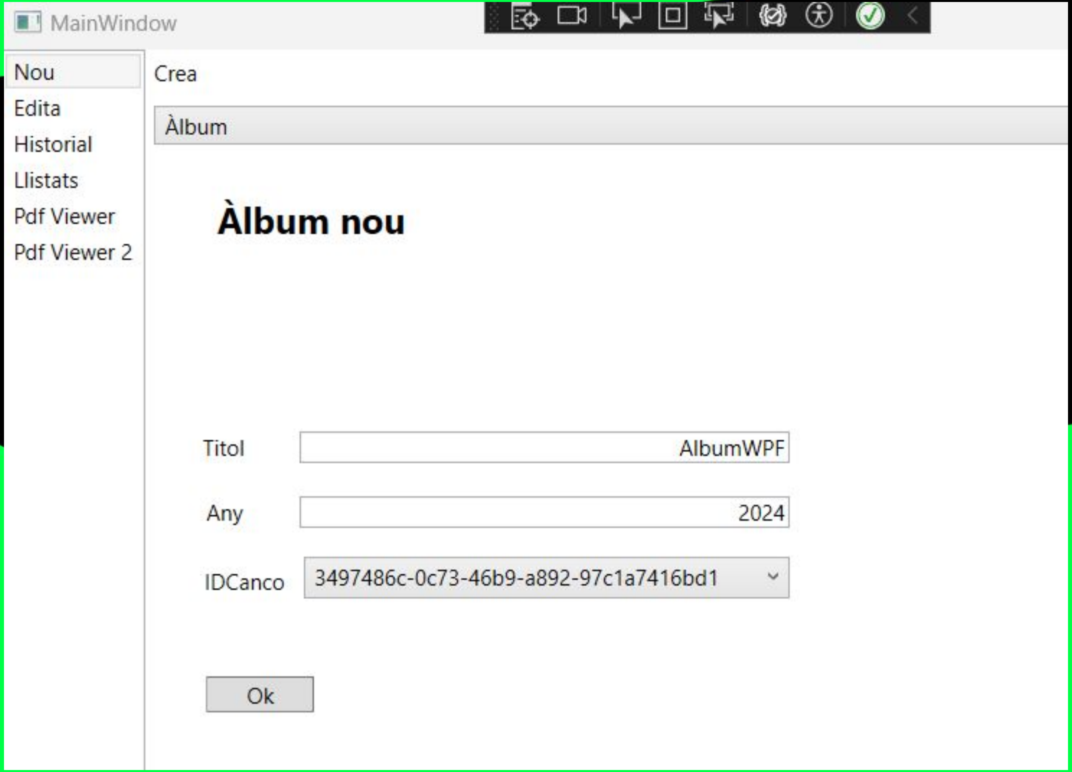

MainWindow
Nou
Edita
Historial
Llistats
Pdf Viewer
Pdf Viewer 2

CREAR I EDITAR



```
TaulerDeControlRM
    Dependencias
    CreaPages
        PageCreaAlbum.xaml
        PageCreaCanco.xaml
        PageCreaGrup.xaml
        PageCreaMusic.xaml
    EditaPages
        PageEditaAlbum.xaml
            PageEditaAlbum.xaml.cs
        PageEditaCanco.xaml
            PageEditaCanco.xaml.cs
        PageEditaGrup.xaml
            PageEditaGrup.xaml.cs
        PageEditaMusic.xaml
            PageEditaMusic.xaml.cs
    MenuPages
        PageCrea.xaml
        PageEdita.xaml
        PageHistorial.xaml
        PageLlistes.xaml
        PagePDF.xaml
    App.config
```

CREAR I EDITAR



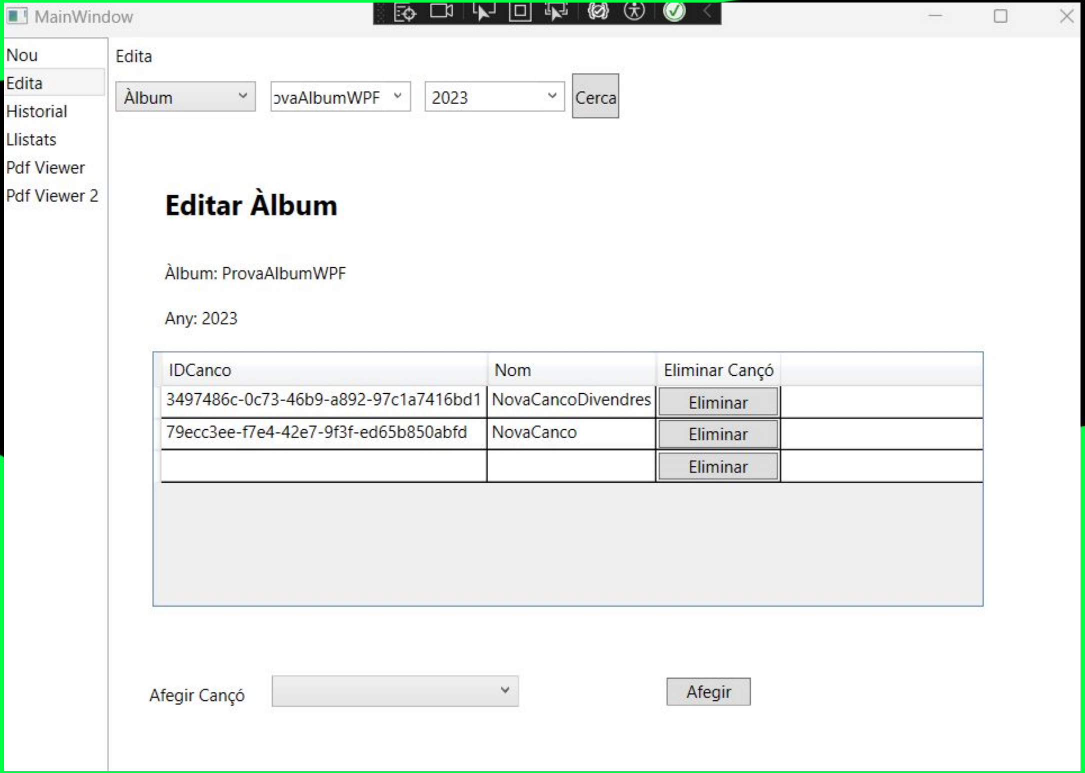
Àlbum nou

Titol AlbumWPF

Any 2024

IDCanco 3497486c-0c73-46b9-a892-97c1a7416bd1

CREAR I EDITAR



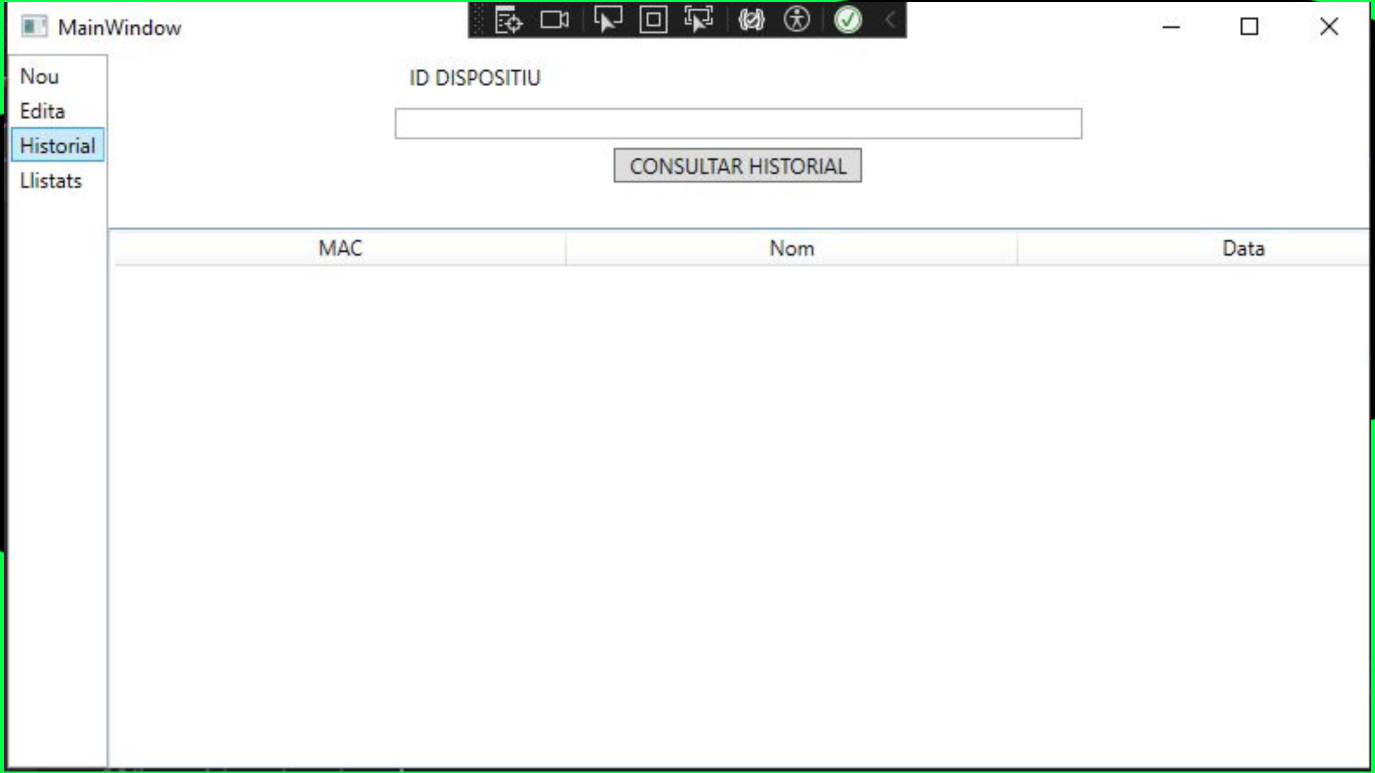
Editar Àlbum

Àlbum: ProvaAlbumWPF

Any: 2023

IDCanco	Nom	Eliminar Cançó
3497486c-0c73-46b9-a892-97c1a7416bd1	NovaCancoDivendres	Eliminar
79ecc3ee-f7e4-42e7-9f3f-ed65b850abfd	NovaCanco	Eliminar
		Eliminar

HISTORIAL



MainWindow

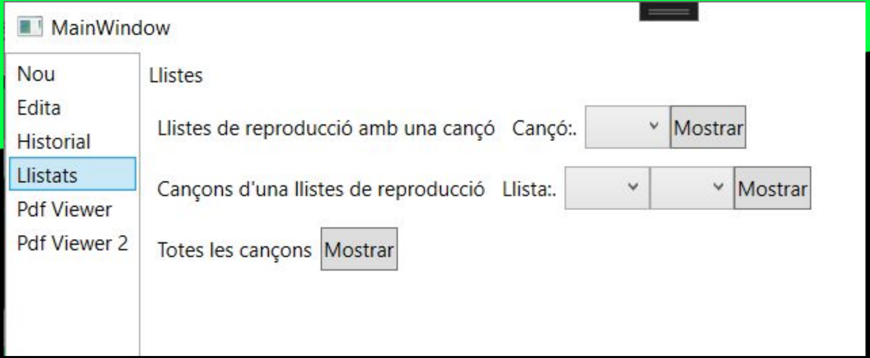
- Nou
- Edita
- Historial
- Llistats

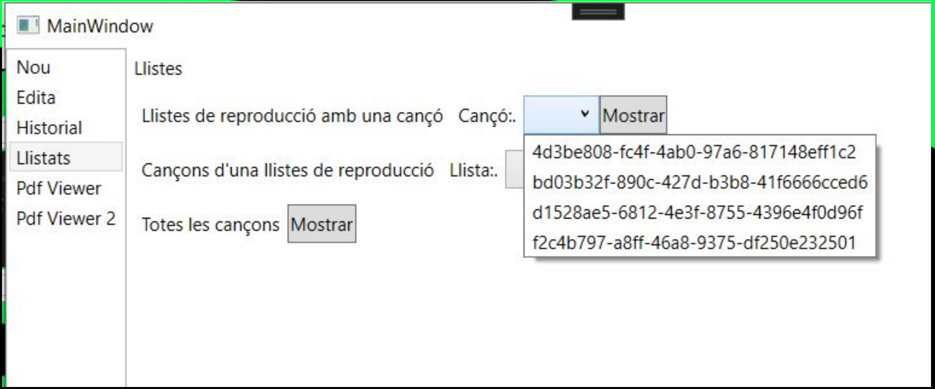
ID DISPOSITIU

CONSULTAR HISTORIAL

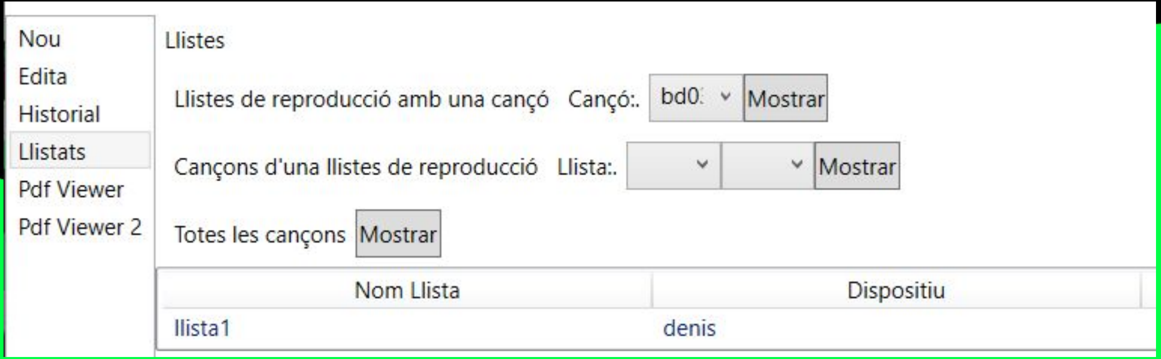
MAC	Nom	Data

LLISTATS



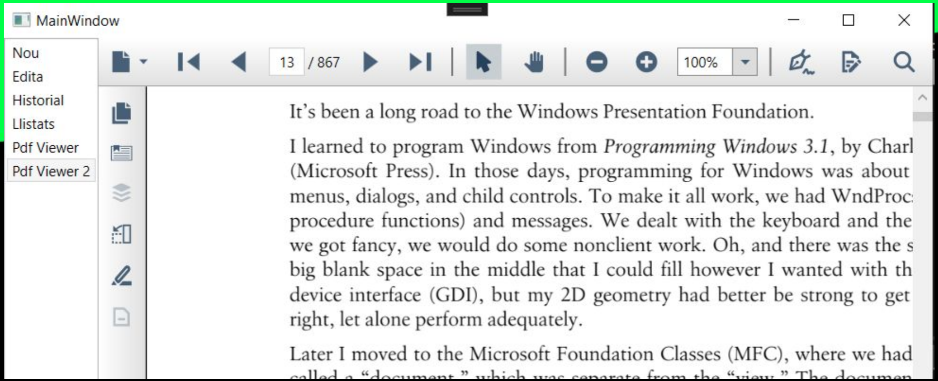


```
4d3be808-fc4f-4ab0-97a6-817148eff1c2
bd03b32f-890c-427d-b3b8-41f6666cced6
d1528ae5-6812-4e3f-8755-4396e4f0d96f
f2c4b797-a8ff-46a8-9375-df250e232501
```



Nom Llista	Dispositiu
llista1	denis

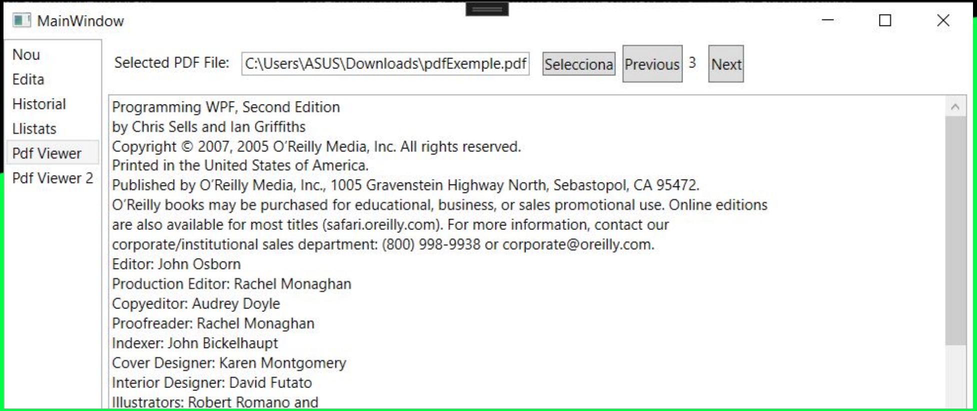
PDF



It's been a long road to the Windows Presentation Foundation.

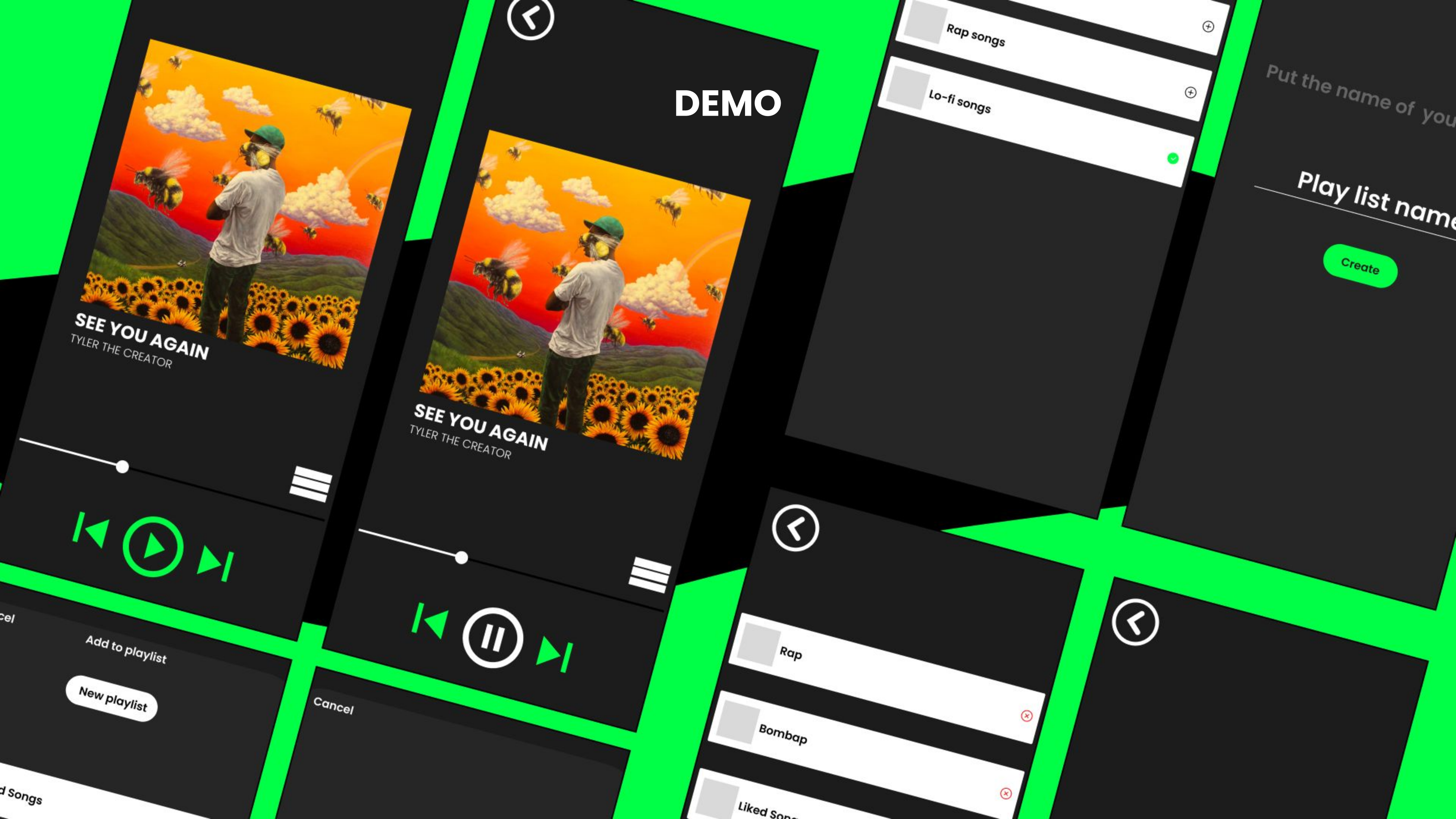
I learned to program Windows from *Programming Windows 3.1*, by Charl
(Microsoft Press). In those days, programming for Windows was about
menus, dialogs, and child controls. To make it all work, we had WndProc
procedure functions) and messages. We dealt with the keyboard and the
we got fancy, we would do some nonclient work. Oh, and there was the s
big blank space in the middle that I could fill however I wanted with th
device interface (GDI), but my 2D geometry had better be strong to get
right, let alone perform adequately.

Later I moved to the Microsoft Foundation Classes (MFC), where we had



Programming WPF, Second Edition
by Chris Sells and Ian Griffiths
Copyright © 2007, 2005 O'Reilly Media, Inc. All rights reserved.
Printed in the United States of America.
Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions
are also available for most titles (safari.oreilly.com). For more information, contact our
corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.
Editor: John Osborn
Production Editor: Rachel Monaghan
Copyeditor: Audrey Doyle
Proofreader: Rachel Monaghan
Indexer: John Bickelhaupt
Cover Designer: Karen Montgomery
Interior Designer: David Futato
Illustrators: Robert Romano and

DEMO


SEE YOU AGAIN
TYLER THE CREATOR
SEE YOU AGAIN
TYLER THE CREATOR
Rap songs
Lo-fi songs
Put the name of you
Play list name
Create
cel
Add to playlist
New playlist
d Songs
Cancel
Rap
Bombap
Liked Son