

WORKSHOP 1

OBJECT-ORIENTED PROGRAMMING

MEMBERS:

- MIGUEL ANGEL TAUTA GARCIA
20242005126
- LUISA DANIELA CORREA TORRES
20242020048

PROFESSOR

- ENG. CARLOS ANDRES SIERRA

PROJECT DEFINITION

Project Name: Apple Music.

Description: This project will be a small version of a music playback application, applying the principles of object-oriented programming. It will have various functions that allow users to search, play, and manage their favorite music easily.

Main Features:

- Song playback.
- Playlist management.
- Song and artist search.
- User registration.

These will be the main features of our project, which will be implemented and expanded as the project progresses.

PROJECT OBJECTIVES

- **Apply object-oriented programming principles in a practical way:** Implement basic object-oriented programming concepts such as classes, inheritance, polymorphism, and encapsulation in system development.
- **Create a functional music playback system:** Allow the management and playback of songs through a simple interface.
- **Implement playlist management:** In this application, users will be able to create, modify, and delete playlists.
- **Develop a scalable and modular structure:** Facilitate the expansion of the system with new functionalities in the future.
- **Optimize user experience:** Design an intuitive and easy-to-use interface.

REQUIREMENTS

Functional Requirements

1. **User Registration:** The system must allow users to register by providing their name, email address, and password.
2. **Login:** Users must be able to log in using their email and password.
3. **Music Browsing:** The system must allow users to search for and explore songs, albums, and artists.
4. **Custom Playlist:** The system must allow users to create, edit, and delete their own playlist.
5. **Save Favorite Songs:** Users must be able to mark songs as favorites and easily access them.
6. **Account Management:** Users must be able to modify their personal information and change their password.

Non-Functional Requirements

1. **Performance:** The application must load music results in under 5 seconds.
2. **Scalability:** The system must be able to handle at least 1,000 concurrent users.
3. **Usability:** The interface must be intuitive and adapt correctly to both mobile and desktop devices.
4. **Security:** Passwords must be securely stored.
5. **Compatibility:** The application must be compatible with modern browsers and mobile operating systems (IOS and Android).

USER STORIES

1. "As a user, I want the application to preload the next song before the current one ends, to avoid playback interruptions."
2. "As a user, I want to search within my playback history by date or song name, to easily find songs I recently listened to."
3. "As a user, I want the shuffle mode to have advanced options, such as avoiding repeated songs or prioritizing my favorites, so that the experience is more varied and adapted to my preferences."
4. "As a user, I want the application to have a sleep timer, so that the music stops automatically after a set period."
5. "As a user, I want to filter my searches by genre, artist, or song duration, to quickly find the type of music I want to listen to."

CRC CARDS

Class: User	
<ul style="list-style-type: none">- Register.- Log in.- Create and manage playlist.- Save favorite songs.	<ul style="list-style-type: none">- Playlist.- Song.

Class: Song	
<ul style="list-style-type: none">- Store information (name, artist, length, etc.).- Play songs.	<ul style="list-style-type: none">- User.- Playlist.

Class: Playlist.	
<ul style="list-style-type: none">- Add/remove songs.- Play songs in order.	<ul style="list-style-type: none">- User.- Song.

Class: Recommendation System	
<ul style="list-style-type: none">- Analyze user history.- Suggest songs or playlists.	<ul style="list-style-type: none">- User.- Song.

MOCKUPS

- *Muckups apple music for mobile devices*



- *Muckups apple music for website*

