UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DEVELOPMENT OF A MUSIC RECOMMENDER SYSTEM TO PROMOTE EMOTIONAL WELL-BEING

ANTONIO ARDURA CARNICERO
JUNIO 2023

TRABAJO DE FIN DE GRADO

Título:	Desarrollo de un sistema de recomendación musical para fomentar el bienestar emocional
Título (inglés):	Development of a Music Recommender System to Promote Emotional Well-being
Autor:	Antonio Ardura Carnicero
Tutor:	Sergio Muñoz López
Departamento:	Departamento de Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:	——
Vocal:	——
Secretario:	——
Suplente:	——

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes

TRABAJO FIN DE GRADO

DEVELOPMENT OF A MUSIC RECOMMENDER SYSTEM TO PROMOTE EMOTIONAL WELL-BEING

Antonio Ardura Carnicero

Junio 2023

Resumen

En los últimos años se ha producido un gran avance en el estudio de las emociones humanas y su regulación. Las emociones pueden manifestarse de diferentes maneras y afectar a distintos ámbitos de la vida de las personas, no sólo a las relaciones intrapersonales, sino también a las interpersonales. Regular las emociones es esencial para el bienestar y el funcionamiento diario. Cuando alguien aprende a controlar sus emociones, es más capaz de gestionar el estrés, mantener relaciones sanas y tomar buenas decisiones. La regulación de las emociones también ayuda a prevenir comportamientos impulsivos y fomenta la estabilidad emocional y la resiliencia, que son cruciales para mantener una buena salud mental.

Tradicionalmente se han utilizado técnicas como el control de la respiración o la meditación para lograr una adecuada gestión emocional. Sin embargo, actualmente se están explorando nuevas técnicas basadas en la inteligencia artificial. Una de las soluciones más populares es el uso de la música para influir en nuestro estado emocional. Las soluciones existentes suelen carecer de personalización y precisión en el uso de la música para la regulación emocional. Por ello, se propone un proyecto para cubrir estas carencias mediante el desarrollo de un sistema recomendador de playlists. Este sistema aborda el problema de la subjetividad en la interpretación de la música para la regulación emocional mediante la incorporación de ciertos parametros que recogen las preferencias y necesidades individuales de cada usuario.

El proyecto comenzará con un análisis del estado del arte en relación con la regulación emocional, el análisis de emociones y las técnicas de aprendizaje automático. A continuación, se definirán los requisitos del sistema y se diseñará su arquitectura. Se utilizarán herramientas como Flutter y la API de Spotify para desarrollar la aplicación. Por último, se realizarán pruebas y evaluaciones para medir la eficacia del sistema en la regulación emocional y la recomendación de música adaptada a la emoción del usuario.

Palabras clave: Emociones, Música, Experiencia emocional, Aplicación, Recomendación, Spotify, Flutter, BLoC

Abstract

In recent years there has been a great advance in the study of human emotions and their regulation. Emotions can manifest themselves in different ways and affect different areas of people's lives, not only intrapersonal relationships but also interpersonal ones. Regulating emotions is essential for well-being and daily functioning. When someone learns to control their emotions, they are better able to manage stress, maintain healthy relationships, and make good decisions. Emotion regulation also helps prevent impulsive behaviors and fosters emotional stability and resilience, which are crucial to maintaining good mental health.

Traditionally, techniques such as breathing control or meditation have been used to achieve adequate emotional management. However, new techniques based on artificial intelligence are currently being explored. One of the most popular solutions is the use of music to influence our emotional state. Existing solutions often lack personalization and precision in the use of music for emotional regulation. Therefore, a project is proposed to fill these gaps by developing a playlist recommender system. This system addresses the problem of subjectivity in the interpretation of music for emotional regulation by incorporating certain parameters that reflect the individual preferences and needs of each user.

The project will start with an analysis of the state of the art about emotional regulation, emotion analysis, and machine learning techniques. Then, the system requirements will be defined and its architecture will be designed. Tools such as Flutter and the Spotify API will be used to develop the application. Finally, tests and evaluations will be carried out to measure the effectiveness of the system in emotional regulation and the recommendation of music adapted to the user's emotions.

Keywords: Emotions, Music, Emotional experience, App, Recommendation, Spotify, Flutter, BLoC

Agradecimientos

Es difícil expresar con palabras la profunda gratitud que siento hacia cada una de las personas que han estado conmigo durante este recorrido, por el apoyo brindado durante toda la carrera y en especial durante mi Trabajo de Fin de Grado. Ha sido un camino lleno de desafíos, pero vuestra presencia ha sido mi apoyo en momentos de duda y cansancio.

A mis queridos padres, gracias por su confianza en mis capacidades, por animarme a dar lo mejor de mí y por ser mi constante fuente de inspiración. Sus palabras de aliento y sus sacrificios han sido el motor que me ha llevado a este momento. No puedo agradecerles lo suficiente.

A mis hermanos, gracias por estar siempre dispuestos a escuchar mis inquietudes. En especial a mis hermanas Elena y Lola, por preguntarme constantemente y ayudarme siempre que lo he necesitado.

A mis amigos, quienes han sido como una familia para mí. Sus palabras de aliento, consejos y risas han iluminado mi camino durante esta etapa tan dura y exigente. Gracias por entender mis ausencias, por escucharme en mis momentos de bajón y por celebrar conmigo cada pequeño logro.

Por último,quiero expresar mi gratitud a todo el departamento del GSI. A Sergio, su orientación experta, paciencia y toda la dedicación puesta, han sido fundamentales en el desarrollo de este TFG. Gracias por brindarme esta oportunidad y su tiempo y conocimiento.

Contents

List of Figures

CHAPTER 1

Introduction

This chapter will present both the context in which the project is being developed and the specific objectives on which it is focused. In addition, an overview of the organization and structure of this document will be given, accompanied by a summary of the different sections described.

1.1 Context

There has been a great advance in the study of human emotions and how to regulate them in recent years. Emotions can manifest themselves in different ways and affect different areas of people's lives. Good management of emotions enables people to lead a life of healthy relationships and to make good decisions [3].

The emotional aspect of our well-being is essential for our overall health and enjoyment of life. Over time, different techniques have been explored and employed to enhance and foster our emotional state. Among these methods, music has emerged as a powerful tool with proven positive effects [4]. This relationship between music and feelings comes directly from the very definition of the word music, which is a set of successive sounds combined, which produce an aesthetic or expressive effect on individuals. This description emphasizes the expressive purpose of music, that is, its ability to vividly manifest feelings.

Throughout history, music has been a form that humans have used for expression and communication. However, beyond its artistic and cultural significance, a great deal of scientific research has been able to confirm the benefits that music has on emotional health [5]. Several studies have demonstrated the ability of music to shape and affect our mood and how to manage it [4] [5]. Certain elements and characteristics of music, such as rhythm and tone, can trigger specific emotions in the listener [6]. For example, slow, mellow melodies are often associated with creating feelings of calm and relaxation, while faster, more rhythmic songs can generate feelings of happiness and excitement.

In addition to its effect on mood, music can also be used as a method of therapy and a form of emotional self-care. An example of this is music therapy. The use of music and its characteristics potentiates or restores some functions of the individual to achieve better integration, both at intrapersonal and interpersonal levels. In this way, there is an improvement in the quality of life. Several meta-analyses have been carried out to verify the effect that this technique has on patients. It was concluded that music therapy can significantly improve specific aspects of patients with ASD [7] and can potentially improve emotional well-being, reducing symptoms of anxiety and depression [8].

Some studies have shown that music reduces stress and anxiety in daily life by lowering levels of cortisol, the hormone that is released in response to stress [4]. It was also found in the same study that music for relaxation purposes decreased the activity of alpha-amylase, an enzyme that may be a possible biomarker of stress in workers [9].

It should be noted that the impact of music on mood varies from person to person, as musical preferences and personal tastes play a decisive role. People may experience more intense positive emotional responses when they listen to music that is meaningful to them or that brings back memories of positive emotions [10].

Individual preferences and tastes in listening to music have been shown to directly affect the emotion experienced. The study by Thoma et al. (2019) [11] found that listening to self-selected music significantly reduced negative emotions and increased positive emotions in participants. This study also reflected that music positively impacts physiological arousal, as indicated by changes in heart rate variability. This indicates that music can change both subjective emotional experiences and physiological responses.

This project is developed around the theme of Mental Health and its relationship with music. The objective is to develop an application based on musical recommendations to promote emotional regulation, improving the mental and emotional well-being of the user. Although solutions similar to the proposed one already exist, they often lack personalization and precision in the use of music for emotional regulation. To address these shortcomings, this project proposes a music recommendation system that assesses the user's emotional state and how they want to feel, based on the area they are in. In addition, this system addresses the problem of subjectivity in the interpretation of music for emotional regulation by incorporating music recommendations according to the preferences and tastes of each user.

The proposed system is intended to help regulate emotions or improve the well-being of users. It aims to improve their mood, improve their quality of life and help to cope with uncontrolled moods such as episodes of anxiety, depression, or anger.

In addition to the technologies used for the development of this project, we have also explored the different musical genres most listened to, the relationships between musical characteristics, how they are associated with different emotions, and the use of music in emotional regulation and human well-being.

1.2 Project goals

This Final Project has different objectives, which have been defined from the beginning under a main objective. This main objective is the development of a music recommendation tool for the management and improvement of personal emotions. The application will incorporate technologies to make the recommendation of song playlists adapted to the user based on the data that this system extracts. In addition to the main objective, the project includes the following sub-objectives:

- Study of the characteristics of music and its relationship with emotions.
- Identification of the system requirements.
- Design and development of the application.
- Use of techniques to recommend the appropriate music for each specific emotion.
- Evaluation of the system in a practical case.

1.3 Structure of this document

This section provides a brief overview of the chapters included in this document. The structure is as follows:

Chapter 1. Introduction. Presents the context in which the project took place and the specific goals it focuses on.

Chapter 2. State of the Art. Different relevant aspects are addressed, offering an overview of emotions, their study, and their relationship with musical characteristics. The main technologies on which this project is based are also explained.

Chapter 3. Requirements.Describes the use cases of the proposed system, highlighting the different functionalities and scenarios in which it will be applied.

Chapter 4. Architecture. Details the architectural design and framework of the system, including its components, modules, and interactions.

Chapter 5. Case-study. Presents a detailed case study of the application, showcasing the practical implementation of the system and walking through all available functionalities.

Chapter 6. Conclusion. Summarizes the key findings, contributions, and insights gained throughout the project, as well as provides recommendations for future work and potential areas of improvement.

CHAPTER 2

State of the Art

In this chapter, the state of the art of the project is presented, addressing different relevant aspects. First, an initial study on emotions is carried out, exploring their nature and their influence on human beings. Next, the close relationship between emotions and music is analysed, examining how certain musical elements can evoke and transmit emotions in listeners. Finally, the technologies employed in the project are explored, highlighting the tools and methodologies used.

2.1 Emotion study

Emotions are psychophysiological reactions experienced in various situations in response to internal or external changes or stimuli. They are an intrinsic characteristic of human beings and play a fundamental role in the way we perceive, interpret and respond to the outside world [12].

The characteristics of an emotion are unique to each of them, yet they all share certain common aspects.

- Emotions affect multiple aspects of our experience. At the physiological level, they can cause changes in heart rate, blood pressure, respiration, and hormonal activity. At the cognitive level, emotions influence our thoughts, beliefs and perceptions of reality. In addition, emotions are also expressed through observable behaviors, such as facial expressions, gestures and body postures [13].
- Emotions are transitory emotional states that tend to have a limited duration [14].
- Each individual experiences emotions differently, being unique and personal. That is why emotional responses may vary due to individual factors, such as personality, cultural background and past experiences [13].
- Emotions are characterized by different levels of arousal and by their positive or negative valence. Activation is the degree of intensity of the emotion, ranging from low activation (such as serenity) to high activation emotions (such as anger or fear). Valence refers to the subjective evaluation of whether an emotion is positive or negative [14].

The functioning of emotions is still unknown in the field of psychology, but there are several theories and approaches that attempt to explain this process. One of the most prominent is the theory of cognitive appraisal, proposed by Richard Lazarus. This theory affirms that emotions are the result of a cognitive evaluation carried out before environmental stimuli. Assessment involves complex processes such as appraisal of relevance, assessment of coping skills, and interpretation of physiological signals.[15].

Another important theory is that of emotional response systems by Stanley Schachter and Jerome Singer. It holds that emotions are triggered by a combination of physiological arousal and cognitive interpretation of that arousal. According to this theory, a physiological change is experienced first, followed by evaluation to determine the emotion and the cause of the emotion [16].

These theories emphasize the importance of cognitive appraisal and interpretation of physiological cues in emotional experience. However, it is important to note that there are numerous additional theories that address how music affects emotional state. These theories explore various aspects, such as the emotional response to musical elements, the connection between music and emotional memory, and the therapeutic effects of music on emotional regulation.

Despite the fact that the origin and functioning of emotions remain largely unknown, progress has been made in their classification. Different approaches and models have been proposed to understand and categorize emotions. One of the most helpful approaches to this project is the Russell [17] model, which proposes a two-dimensional classification based on valence and emotional arousal. Activation and valence characteristics are characteristics included in music, which makes this model perfect for the relationship between emotions and music. It is considered that emotions can be classified according to their degree of liking or disliking (valence) and their level of activation (arousal). This two-dimensional classification provides a basis for understanding and comparing emotions.

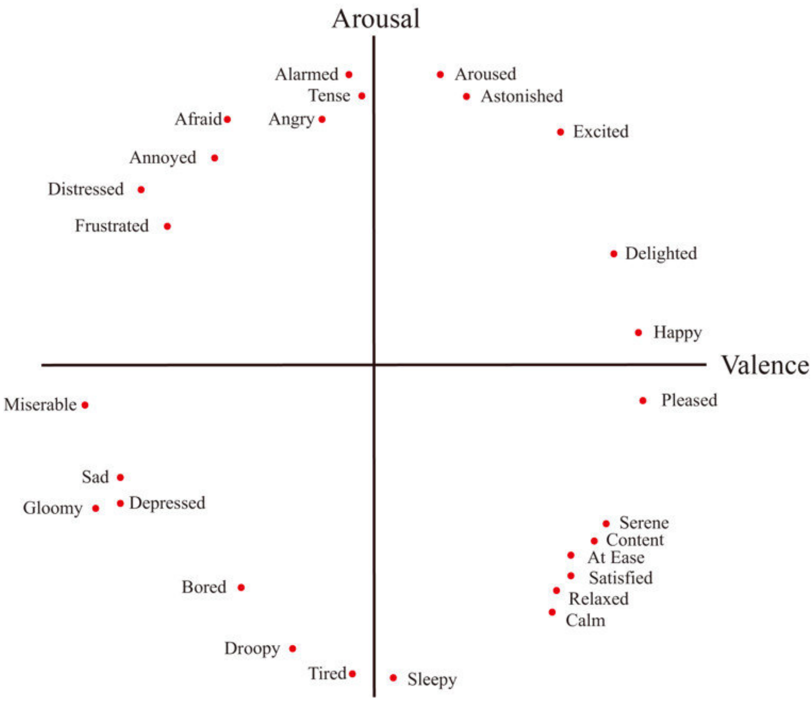


Figure 2.1: Russells Circumplex Model [1]

These two dimensions make it possible to classify and compare different emotions according to their position in the affective space.

2.2 Music features and their relationship to emotions

Musical characteristics and their relationship to human emotions have been the subject of several studies in the psychology of music. These investigations have shown that certain aspects of music can influence the emotional experience of listeners [5].

The model of musical emotion perception and production suggested by Juslin and Västfjäll (2008) is one of the most common approaches in the field [6]. According to this model, musical features trigger a series of cognitive and physiological processes that lead to emotional experience.

Some of these structural features that have been extensively researched in relation to emotions include tempo, tonality, modality, dynamics (loudness), timbre, valence and danceability [18].

- Tempo: It has been observed that pieces of music with a fast tempo tend to be associated with positive emotions and excitement, while pieces with a slow tempo are associated with more relaxed or melancholic emotions [19].
- Tonality: The key of a piece of music (major or minor) has also been studied. Pieces in major keys tend to evoke positive and happy emotions, while minor keys are more associated with sad or melancholic emotions [19].
- Loudness: Changes in the intensity or loudness of music can affect the emotional experience of listeners. For example, a sudden increase in dynamics can generate a sense of tension or intense emotion [19].
- Timbre: Timbre refers to the unique sonic characteristics of different musical instruments [19].
- Valence: refers to the dimension of emotion that ranges from positive to negative [19].
- Danceability: refers to the ability of a song to provoke the desire to move and dance. This characteristic has been associated with positive emotions and excitement [19].

In conclusion, music plays a significant role in people's emotional experiences. Through research in the psychology of music, it has been shown that characteristics such as tempo, key, dynamics, timbre, valence, and danceability can influence how we perceive and experience emotions when listening to music. The model of perception and production of musical emotions proposed by Juslin and Västfjäll has provided a solid theoretical framework for understanding how musical characteristics trigger cognitive and physiological processes that contribute to emotional experience. These findings are especially relevant in the context of this project, as they seek to explore how music can influence the emotional state of people in specific settings, such as music therapy or creating emotional environments.

2.3 Enabling Technologies

This section details the tools and technologies used during the development of the project. The programming languages used will be presented and the tools used for application programming and algorithm development will be discussed.

2.3.1 Dart

Dart is a programming language developed by Google, designed to be easy to use and learn. In addition, it is a language that provides high performance and efficiency. Dart is, like Python, a multi-paradigm language, that is, it allows several programming styles, such as object-oriented, imperative, and functional [20]. The main features of this language are:

- A readable syntax: A syntax that makes it easy to understand the code and decreases the possibility of making mistakes [20].
- Strong typing: Implies that variables once declared with a specific type cannot be used as if they were of another type [20].
- Efficient garbage collector: It is responsible for managing memory automatically and efficiently, freeing unused memory and avoiding memory leaks [20].
- High performance: Achieved through ahead-of-time (AOT) or just-in-time (JIT) compilation [20].
- Integration with Flutter: Flutter uses Dart as its core programming language, allowing the creation of native apps for both iOS and Android as well as web from the same code base [20].

2.3.2 Flutter

Flutter has become extremely popular among developers in recent years due to its outstanding features and advantages. According to statistics, its adoption has experienced significant growth, being chosen by approximately 30% developers in 2019 and increasing to 46% in 2022. This rapid growth has led Flutter to become the leading framework in terms of preference among developers [21].

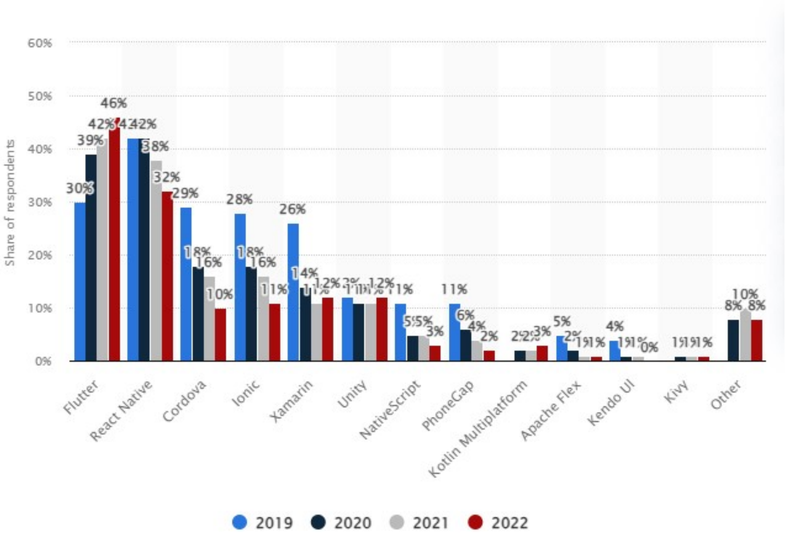


Figure 2.2: Cross-platform mobile frameworks most used

Flutter is an open-source application development framework created by Google that allows you to build high-quality user interfaces using the same code base for both mobile and web applications. Flutter has become popular among developers because of its main features and benefits [22].

Among the main features and advantages of Flutter we find:

- Performance and optimization: Flutter uses the Skia rendering engine, which provides fast and fluid performance for user interfaces. In addition, Flutter utilizes AOT to generate highly optimized native code, which translates into an agile and responsive user experience [22].

- Cross-platform development: With Flutter, you can build iOS and Android apps from a single code base. This means you don't need to develop and maintain two separate sets of code for each platform, saving time and effort on development and updates [22].

- Customizable UI: Flutter provides an extensive set of customizable widgets and visual styles, allowing you to create attractive and tailored user interfaces. In addition, Flutter facilitates the creation of smooth animations and visual effects to provide an engaging user experience [22].

- Hot Reload: Flutter offers a feature called “Hot Reload” that allows you to make changes to your code and see the results immediately, without restarting the application. This streamlines the development process, as developers can quickly experiment with different ideas and solve problems efficiently [22].
- Access to packages and libraries: Flutter provides access to a wide range of Flutter packages and libraries, making it easy to integrate the functionality and reuse existing code in web applications [22].

2.3.3 BLoC

The BLoC (Business Logic Component) [2][23] architecture pattern is a design pattern used in application development, especially in the context of Flutter, a cross-platform mobile application development framework.

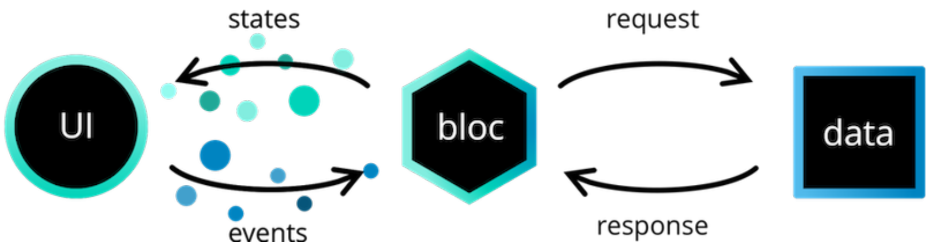


Figure 2.3: Bloc Architecture [2]

The BLoC pattern is based on the separation of business logic and user interface, which allows for better code organization and maintenance. It works through communication between three main components:

- **Business Logic Component (BLoC)**: this component is in charge of the business logic and contains the implementation of the algorithms, rules, and processes necessary for the application to function. The BLoC acts as a black box that takes events as input and produces states as output.
- **Events**: Events are the actions or occurrences that happen in the application, such as screen touches, user interactions, API responses, and so on. These events are sent to the BLoC so that it can process them and produce a response.
- **States**: States represent the different conditions or situations in which the application’s user interface may find itself in response to events. Each state has a specific visual representation. The BLoC issues updated states as changes in the business logic occur.

CHAPTER 3

Requirements

This chapter focuses on the detailed analysis of requirements in different scenarios. The application of use cases is explored, providing a complete understanding of their implementation in software development.

3.1 Use cases

Different scenarios are presented in which users can interact with the system. These use cases describe concrete situations in which users can use the different functionalities and features offered by the application.

1. Get recommendations: allows the user to select the emotion they are experiencing and how they want to feel so that the system can recommend a playlist.
2. Consult daily emotion log: allows the user to consult the emotions and characteristics recorded both in the current day and during the previous days.
3. Fill in satisfaction questionnaire: gives the user the option to rate the recommendation obtained and rate whether it has helped them with the emotion experienced.
4. Review saved playlist: allows to review the playlists the user has saved in his Spotify account.

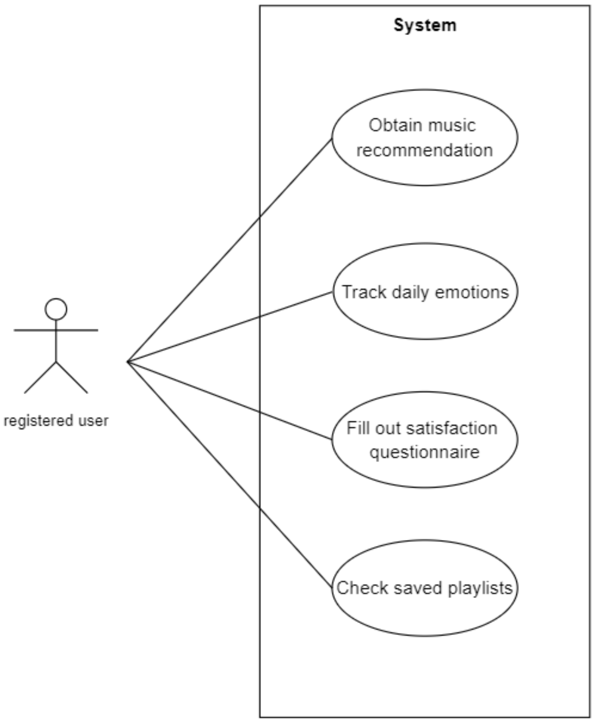

System
Obtain music recommendation
Track daily emotions
Fill out satisfaction questionnaire
Check saved playlists
registered user

Figure 3.1: Use cases

3.1.1 Use Case 1: Consult daily emotions record

Main actors: user

Initial event: Access the daily registration screen

Brief summary of the case: Track the emotions felt each day.

Sequence of actions in a typical use case:

i. The user requests to view his daily emotion log in the user interface.

ii. The system accesses the database to obtain the user's daily emotion log data.

iii. The system displays the daily emotion record data to the user.

Extensions: The user will be able to navigate to a more detailed screen for each emotion, with the data they have entered themselves.

3.1.2 Use Case 2: Obtain recommendations

Main actors: user

Secondary actors: Spotify API

Initial event: Go to the initial emotion selection screen

Brief summary of the case: The user selects the necessary characteristics for generating recommendations.

Sequence of actions in a typical use case:

i. The user selects the initial emotion.

ii. They navigate to the selection page of the area where they experienced that emotion.

iii. Finally, the user selects the objective emotion.

iv. The system uses an algorithm developed to determine the optimal path between emotions and collect the audio futures associated with each emotion.

v. The system makes a series of calls to the Spotify API to obtain a recommended song based on the characteristics obtained for the given emotions.

vi. The Spotify API returns the song recommendation.

vii. The system returns to the user a list with the data obtained from the API.

Extensions: The application also allows the user to provide a description of the problem that caused the initial emotion. Additionally, the user will be able to save the playlist to their Spotify account.

3.1.3 Use Case 3: Fill out a satisfaction questionnaire

Main actors: user

Initial event: Access the questionnaire screen.

Brief summary of the case: Conduct a satisfaction survey for the recommendation.

Sequence of actions in a typical use case:

i. The user selects the option to complete the satisfaction questionnaire.

ii. The questionnaire is displayed to the user.

iii. The user rates the playlist recommendation and provides additional comments.

iv. The system returns the user to the previous screen.

Extensions:The user will be able to provide additional detailed feedback by filling in a text, and commenting on their level of satisfaction.

3.1.4 Use Case 4: Review saved playlist

Main actors: user

Secondary actors: Spotify API

Initial event: Go to the saved playlist screen.

Brief summary of the case: Perform a check on the playlists saved in the user's Spotify account.

Sequence of actions in a typical use case:

i. The user requests to view his saved playlists.

ii. The system makes a few calls to the Spotify API to get the user's saved playlists.

iii. The Spotify API returns the playlists.

iv. The playlists are displayed as saved ones playlists to the user.

Extensions: The user will be able to access each playlist to check the songs within them.

CHAPTER 4

Architecture

In this chapter we cover the design phase of this project, as well as the implementation details of its architecture. First, we present an overview of the application. Then, we develop each module that makes up the project separately and in much more depth.

4.1 General Architecture

The goal of this project is to promote mental and emotional health by managing emotions through musical recommendations. To achieve this goal, the system interface must be intuitive and accessible to users, so that they can operate it without difficulty and understand it without complications. In addition, it is essential that the system interface be visually appealing to encourage people to use it.

It was decided to use BLoC (Business Logic Component) as the state management system, providing support for state management and data access from a central location in the project so that the system has a clean and scalable architecture. The developed architecture can be seen in Figure 4.1, where the interactions between the components are shown in a simplified diagram.

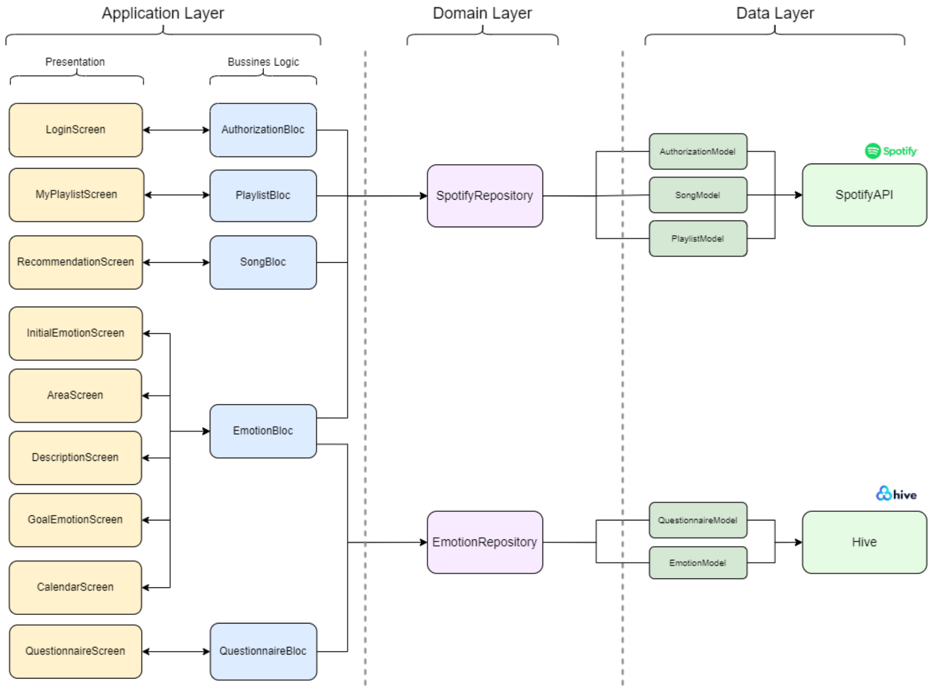


Figure 4.1: Architecture diagram

In the following sections, we will delve into the application, domain, and data layers of this architecture. In the application layer, we will explain the implementation of the user interface and the management of user interactions with the application. Then, in the domain layer, we will focus on the repositories in charge of connecting the application layer with the data layer. Finally, in the data layer, we will explore the integration with the Spotify API and the management of the Hive database for storing emotional data and questionnaire responses. Throughout these sections, we will examine each layer in detail, describing its functionality and how it integrates with the other layers.

4.2 Application Layer

The application layer plays a key role in managing the user interface logic and visual representation in the Flutter application. The BLoC design pattern allows this layer to be separated into two separate sub-layers: presentation and business logic.

The communication between the user interface and the repositories of the domain layer is done through BLoCs. These components allow the reception and sending of events that facilitate the flow of data and the interaction between the different layers of the architecture.

4.2.1 Presentation

The presentation sublayer is in charge of the visual representation of the user interface, including the different pages or screens of the application. It is where widgets are defined and used to display information on the screen and capture user interaction. These widgets can include elements such as buttons, input fields, lists, images, and other visual components.

Among the main screens, we find:

- Login Screen: This is the initial view of the app and displays a button that the user will tap to log in with their Spotify account. This page communicates with the “AuthorizationBloc”, which handles the button event for the repository to get the authorization code and token through the Spotify API connection.

- Initial Emotions Screen: On this page, the user can select an initial emotion, which represents the emotional state they are feeling. The selection of the emotion is done through a carousel and the pressing of a selection button. This event will be handled by the “EmotionBloc”.

- Area Screen: On this page, the user can select the area they are in when they have felt the initial emotion. The selection of the area is done through a card

- Description screen: The user can enter a summary or description of why it makes sense in a text box or include a few more selections. This text box does not have to be filled. Pressing the “Next” button establishes the management of the event and goes to the next screen.

- Goal Emotion Screen: This screen works very similar to the "AreaScreen". The user can select how he wants to feel by clicking on the card corresponding to said feeling

- MyPlaylists Screen: This page displays a list of playlists that the user has saved to their Spotify profile. Each playlist can be clicked to see the songs it contains

- Recommendation Screen: This is the view that shows a list of 10 songs obtained as system recommendations via the Spotify API. In addition, the screen contains the functionalities, through different buttons, to reload the recommendation and to save it in Spotify respectively.

- Calendar Screen: A small interactive calendar is presented that allows the user to select a specific day from today. By selecting a day on the calendar, communication is established with the "EmotionBloc" to display a list of emotions stored for that day.

- Questionnaire Screen: This page contains the satisfaction questionnaire, where the user has the opportunity to answer questions related to their emotional state and the musical recommendation made by the system. When the user submits their responses, they do so by clicking a "Submit" button and these responses are recorded for the recommendation. Each question in the questionnaire contains a horizontal slider, which allows scoring in a range between 1 and 5. There is also a text box where the user can add more specific comments.

As mentioned above, screens in Flutter are built using widgets. Widgets are organized in a hierarchical structure, where each widget can contain other widgets as child elements. This allows their flexibility and reuse, thus facilitating development. Within the BLoC pattern, widgets are very important when interacting with BLoCs. Widgets send events and in turn, receive updated state. It is a two-way communication that allows synchronization with the business logic and updates its appearance based on state changes.

4.2.2 Business Logic

The BLoC components, explained in this subsection, handle user requests and perform the necessary functions through the repositories. They function as an intermediate point between the user interface widgets and the application domain layer. Their main purpose is to receive and process the events generated by the widgets to perform the corresponding actions. The BLoCs are responsible for processing the events and updating the state of the application according to the defined rules and processes.

The “EmotionBloc” is a Flutter BLoC designed to manage emotions in the application. This BLoC has different responsibilities.

The BLoC is responsible for allowing the user to select an initial emotion, specify the area or context in which that emotion was experienced, facilitate the entry of a more detailed description of the sensation experienced and why, and the target emotion that the user wishes to achieve. Each selected feature is an event that the “EmotionBloc” processes and outputs an updated state.

The “EmotionBloc” collects all the information about the initial emotion, the area, the description, and the target emotion and sends it to “EmotionRepository”, which is in charge of storing this data in a Hive box. In addition, “EmotionBloc” uses this data to interact with “SpotifyRepository”, which is responsible for generating music recommendations based on the recorded emotions.

In addition to the above responsibilities “EmotionBloc” is in charge of displaying emotions on the Calendar page. To display the emotions recorded on each day on the Calendar page, the BLoC retrieves the data stored in Hive and uses it to paint the corresponding user interface.

Also, this logic contains more BLoCs (“AuthorizationBloc”, “PlaylistBloc”, “SongBloc” and “QuestionnaireBloc”). As in the first one, all these BLoCs follow the same state transition [Figure 4.2]. It starts upon receiving state and while executing the user’s request, the system displays a loading page, in case of successful data reception it will send the user to the next screen. In case of an error, the BLoC emits the state “error”, which can be used to display an error message to the user.

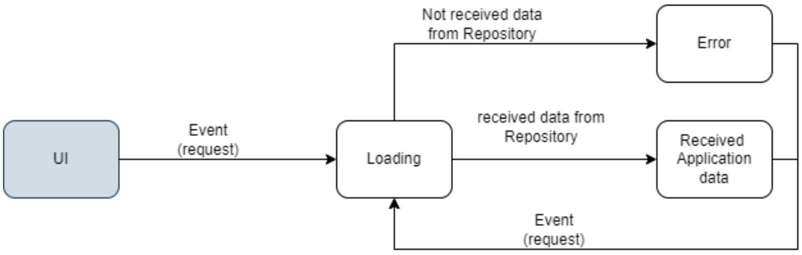


Figure 4.2: BLoCs state transition

In the “AuthorizationBloc” class, an instance of the “SpotifyRepository” class is created that provides methods to interact with the authorization system, such as obtaining the authorization code and the authorization token. The class also contains two objects of type “PublishSubject”: “‘authorizationTokenFetcher” and “authorizationCodeFetcher”. “PublishSubject” is a combination of an observer (Observable) and a data stream (Stream), i.e. it acts by emitting events and transmitting data.

“PlaylistBloc” is responsible for connecting the screen where users can view playlists in their Spotify profile with the “SpotifyRepository”. Its main purpose is to manage the logic related to playlists and facilitate their display and manipulation in the application. In the “PlaylistBloc” implementation, an instance of the “SpotifyRepository” is used to interact with the Spotify API and get the relevant information from the user’s playlists.

The “SongBloc” is the BLoC associated with the recommendations page in the application. Its main function is to get song recommendations and provide the necessary logic to display them in the user interface. For its implementation, the “SpotifyRepository” is used to interact with the Spotify API and get the recommended songs. In addition, methods are used to perform actions related to the recommended songs, such as reloading the list of recommendations and saving them as a playlist in the Spotify account.

When a user completes the questionnaire, an event is sent to “QuestionnaireBloc”. In response to this event, the BLoC updates its internal state to Loading, indicating that a request is being processed.

4.3 Domain Layer

In the layered architecture, the domain layer plays a key role in encapsulating the business logic and concepts of an application. This layer uses repositories as intermediaries to access the data and provide domain-specific methods.

Repositories are responsible for interacting with databases and providing query methods that are tailored to the particular needs of the domain. They act as an intermediate layer between the business logic and the data layer, allowing the business logic layer to work with the domain models without worrying about the implementation details of the databases.

To establish communication between Business Logic and the repositories, interfaces are established. These interfaces promote that the business logic layer does not depend directly on a specific implementation of the repositories, which facilitates substitution or change without affecting the business logic. This section will be divided into the two repositories used for the application.

4.3.1 SpotifyRepository

This repository plays a central role in the operation of the application. Its main purpose is to facilitate the interaction between the application and the Spotify API by providing specific methods to get, add and delete playlists and songs.

As seen in Figure 4.1 the system has a connection to the Spotify API. The audio streaming platform exposes an application programming interface (API) that allows interaction with the music and media content service where tracks are organized according to specific characteristics [Table 4.1].

Attribute	**Description**
Acousticness	Measure of a song's acoustics (0.0 to 1.0)
Danceability	Ability of a song to dance (0.0 to 1.0)
Duration_ms	Duration of the song in milliseconds
Energy	Measure of the energy of a song (0.0 to 1.0)
Instrumentalness	Estimate of the probability of a song being instrumental (0.0 to 1.0)
Key	Key of the song (0 to 11, represents the musical notes and their keys)
Liveness	Probability that a song has been performed live (0.0 to 1.0)
Loudness	Average volume in decibels (dB)
Mode	Indicates if the song is in major mode (1) or minor mode (0)
Speechness	Presence of spoken words in the song (0.0 to 1.0)
Tempo	Estimated tempo of the song in BPM (beats per minute)
Time_signature	Indicator of the number of beats per measure
Valence	Positivity of a song (0.0 to 1.0)

Table 4.1: Audio features

Song recommendation

In the context of this project, we seek to find a sequence of musical songs, represented by $m_1, m_2, ..., m_10$, that guide the user from the initial emotional state (E_s) to the objective emotional state (E_f). Therefore, once the user input data (initial emotion, area, and target emotion) has been collected, the optimal path from the initial emotion to the target emotion is used, following Russell's circumflex model explained in chapter 2.

The transition from one emotion to another does not usually occur while listening to a single piece of music, so these routes were chosen based on an algorithm implemented in Python external to the application.

The path from one emotion to another can be, around the circumflex, in both directions. That is why Markov decision processes have been used to find the best path, considering that going from one emotion to another is better than going through more positive emotions. For this, a series of rewards were defined in which the valence of the emotion was prioritized. For example, following the model, to go from sad to happy one could go through the emotions with the greatest activation (angry-excited) or through a step with less activation but with more positive emotions (relaxed-pleasant), after performing a series of episodes the algorithm concludes that the best path is the second one.

To find all the possible paths among all the emotions allowed in our system, this algorithm has been executed in a Jupyter Notebook and once compiled, the paths have been exported and saved in the application. The obtained paths can be seen in Appendix C. The path represents a sequence of states or emotions that are transited in the context of the analysis or system in question. For example, the states included in the path from Awful to Happy are "Awful", "Angry", "Excited", and "Happy". The system has it saved as a List<String>.

When a user enters an initial and a final emotion, the system obtains the path saved and with it calculates a progression of 10 values for each audio feature (using for this project Valence, Energy, Loudness, Tempo, and Danceability), achieving a smooth and coherent transition between said emotions.

This progression carried out through the "progressiveChangesOnPath()" function, is based on a first assignment of specific audio parameter ranges to each emotion, which has been carried out based on studies and analyses carried out during the development of the project. As for the optimal paths, the Appendix C contains such a prior assignment of specific ranges for each emotion.

The function receives a list of "emotionalPath" emotions and returns a map with the emotional characteristics and their respective sequences of progressive changes.

The logic of the function differs depending on the length of the “emotionalPath”. If it contains only one emotion, the emotion configuration, “emotionValues”, is obtained directly, and sequences of random changes are generated within the defined value ranges.

On the other hand, if “emotionalPath” contains multiple emotions, the case of several consecutive emotions is processed. The list of emotions is traversed and obtained for each pair of consecutive emotions. Random values are generated within the defined ranges, they can be seen in Appendix C, and a sequence of progressive changes is created for each emotional characteristic. The result is an array made up of the 5 characteristics of the songs (Valence, Energy, Loudness, Tempo, and Danceability) and 10 values for each one.

With these values (10 for each feature), the method described below “getSpotifyRecommendations” is called, which makes calls iteratively, one at a time, to get Spotify song recommendations. During each iteration, the same position on the list is taken for each feature, and that specific value is used to prompt Spotify for recommendations.

When the user selects a particular area, the system chooses a random music genre from the 2 assigned to that area and adds it to the API calls to make the recommendations. This assignment is based on research conducted in certain studies on the relationship between musical genres and different areas or contexts [24][25][26]. The genres assigned to the areas are as follows:

- Car:

  - Pop music: Many people choose pop songs to listen to while driving because of their energetic and catchy beat.

  - Rock music: Rock music is a popular genre to listen to while driving.

- Home:

  - Classical music: Chosen to create a calm and relaxing atmosphere in the home.

  - House music: Some people enjoy electronic genres while doing household chores or just to set the mood at home.

- Work:

  - Classical music: It is believed that classical music can help increase concentration and productivity in work environments.

  - Jazz music: A popular choice for having a pleasant musical background while working.

- Leisure:

  - Pop: They are usually pleasant to listen to and suits different activities such as hanging out with friends, watching movies, etc.

  - Reggaeton music: Common choice for dancing and enjoyment during leisure time.

- Sports:

  - EDM (Electronic Dance Music) or Hip-Hop music: These are usually chosen to accompany sports activities and provide an extra boost of energy.

Project Request Methods

For requests to the Spotify API, we use "http.Client", a class used to make HTTP requests that provide an interface to send requests to web servers and receive responses, and "SharedPreferences", another class used to retrieve and store the access token and other authentication-related data. Among the methods used for the development of the system, we find:

- *logout()* This method is responsible for logging out of the application. It removes the access token stored in the shared preferences and makes an HTTP request to the Spotify API to revoke the access token.

- *getSavedPlaylists()*: This method is used to get the playlists saved by the user. It uses the authorization token obtained through "AuthorizationTokenApiProvider" to authenticate the request. The result is returned as a list of Playlist objects.

- *getPlaylistSongs(Playlist playlist)*: This method is used to get the songs of a specific playlist. It uses the authorization token obtained through "AuthorizationTokenApiProvider" to authenticate the request. The result is returned as a list of Song objects, containing information such as the song name, image URL, and track URI.

- *getSpotifyRecommendations(map <String, List <double > parameters)*: This method is used to get song recommendations from Spotify. You receive a map of parameters obtained through an algorithm that is used to adjust recommendation criteria, such as energy and tempo. Thanks to the use of Spotify, the method will first obtain a list of the 5 songs most listened to by the user in the last month. Using these songs and the values obtained in the algorithm, calls are made in search of 1 song. The authorization token obtained through the "AuthorizationTokenApiProvider" is used to authenticate the request. The result is returned as a list of Song objects.

- *createPlaylist(List <Song> songs, String playlistName)*: This method is used to create a new playlist in Spotify and add songs to it. It uses the authorization token obtained through “AuthorizationTokenApiProvider” to authenticate the request. The playlist is created with the provided name and the songs are added using their track URIs.

4.3.2 EmotionRepository

The “EmotionRepository” is responsible for managing and storing the data entered by the user in relation to their emotions. This data includes the initial emotion, the area, the target emotion, the description, and the evaluations made in the questionnaire.

The repository acts as an intermediary layer between the application layer and the data storage layer. Its main function is to provide methods to save and retrieve the emotional data entered by the user. For the development of this part, we find the following methods:

- *saveEmotion*: This function is used to save a new emotion in the database. Receives parameters such as emotion name, date, area, description, and text associated with the emotion. In this function, a “FeelBox” object is created with the supplied data and saved to the database.

- *getEmotionsForCurrentWeek*: This function returns a list of all emotions recorded during the current week. It uses the current date to determine the start and end of the week. Then, it filters the “FeelBox” objects stored in the database and selects those whose dates fall within the range of the current week. Finally, it returns the list of corresponding emotions.

- *getLastEmotionForToday*: This function returns the last recorded emotion for the current day. First, use the “getEmotionsForCurrentWeek” function to get all the emotions recorded during the current week. It then iterates over the list of emotions and compares the dates to find the “FeelBox” object for the last emotion recorded for the current day. Finally, said object is returned.

- *saveRatings*: This function is used to store the ratings associated with an emotion in the “QuestionnaireBox”. You receive various metrics, including the “FeelBox” of the emotion being rated, as well as related ratings (such as emotion rating, appropriateness, playlist rating, and effective rating). In this function, a “QuestionnaireBox” object is created that stores the ratings and is related to the corresponding “FeelBox” object, then stored in the database.

4.4 Data layer

In this section, the data model used in the emotion-based music recommender system is described. The data model defines the entities and the relationships between them, providing a structure for storing and managing the relevant information.

The following diagram shows the entities and how they relate to each other. In the case of our system, the corresponding entity of collecting and storing the token is independent. “FeelBox” and “Playlist” have an indirect relationship, where for an experienced emotion 1 playlist of 10 songs is recommended.

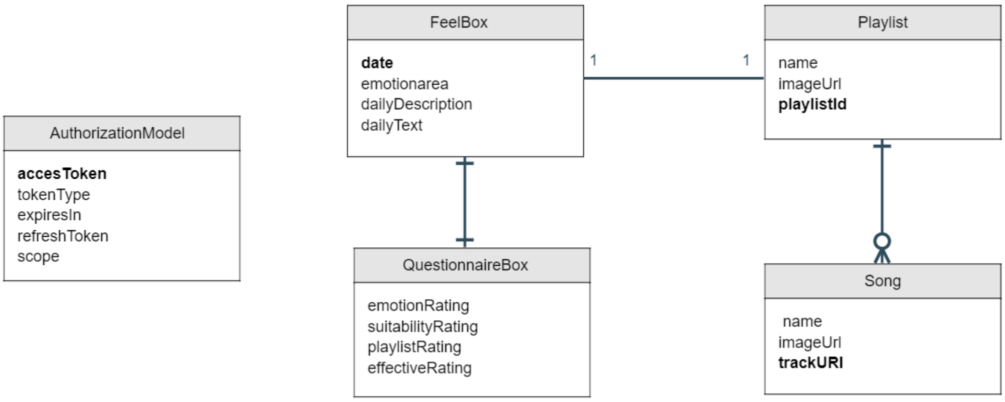


Figure 4.3: Data model diagram

Authorization entity

The AuthorizationModel class represents the authorization model in the system. It has the following attributes:

- accessToken: A string representing the access token used for authenticating requests to the API.
- tokenType: A string indicating the type of token, such as “Bearer”.
- expiresIn: An integer specifying the expiration time of the access token in seconds.
- refreshToken: A string representing the refresh token used to obtain a new access token when the current one expires.
- scope: A string indicating the scope or permissions granted to the access token.

The AuthorizationModel class has a constructor that requires values for all the attributes mentioned above.

This authorization model is used to manage authentication and authorization in your application, ensuring secure access to the Spotify API and handling the renewal of access tokens when needed.

Playlist entity

The “Playlist” entity represents a music playlist. Each playlist has the following attributes:

- name: the descriptive name of the playlist.
- imageUrl: the URL of the image associated with the playlist.
- playlistId: a unique identifier representing the playlist in the system.

Song entity

The “Song” entity represents an individual song. Each song has the following attributes:

- name: the name of the song.
- imageUrl: the URL of the image associated with the song.
- trackUri: the unique identifier representing the song in Spotify.

FeelBox entity

The “FeelBox” entity is used to store information related to emotions, and other user interactions. This entity has the following attributes:

- date: the date on which the entry is recorded in the database.
- emotion: the emotion associated with the entry.
- area: the subject area or context in which the emotion is experienced.
- dailyDescription: a brief description of the entry.
- dailyText: the full text of the entry.

QuestionnaireBox entity

The entity “QuestionnaireBox” is used to store information related to the evaluations that the user has made about a recommendation for an emotion. This entity has the following attributes:

- emotionRating: a numerical rating of the intensity of the emotion.
- suitabilityRating: a numerical rating of the suitability of the recommendation.
- playlistRating: a numerical rating of the quality of the recommended playlist.
- effectiveRating: a numerical rating of the overall effectiveness of the recommendation.

CHAPTER 5

Case study

This chapter presents the case study of the application, where the theoretical knowledge and methodologies described above will be put into practice. This study will provide a concrete vision of the operation and usefulness of the application, as well as an evaluation of its performance in a real environment.

The main objective of this case study is to demonstrate the effectiveness and usefulness of their recommender system while deepening the understanding of how it works in practice.

It will describe in detail how the user interacts with the user interface of the system. The application has been named 'MoodBalance', as it is a system whose objective is the regulation of an emotional state through a series of songs.

For the description of the actual operation, the user flow will be explored, detailing the steps and actions that the user performs throughout the process. The screens through which the user goes through will be shown, thus offering a complete view of the experience in the application.

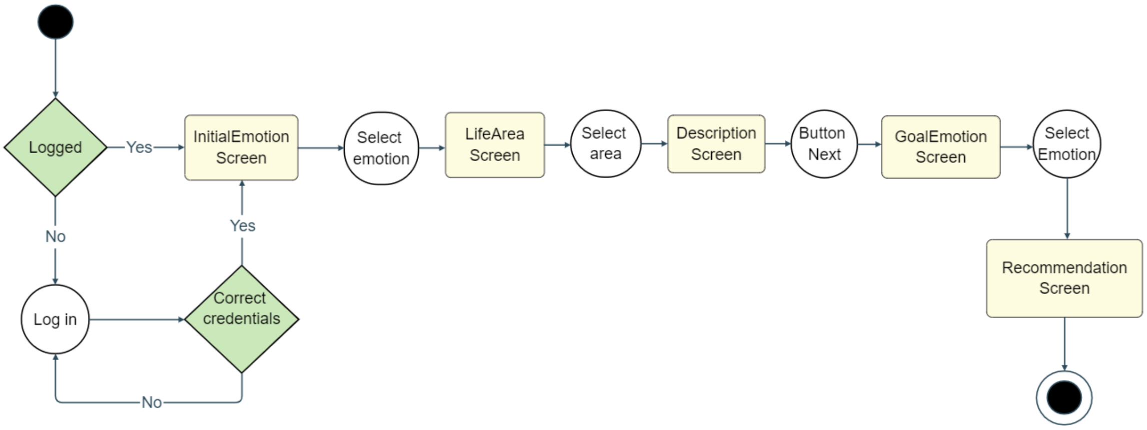


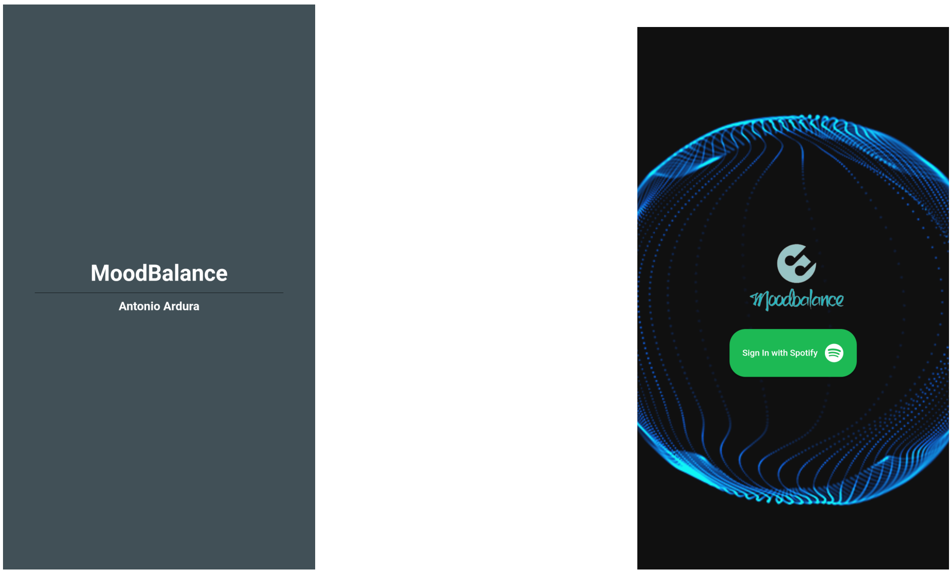
Figure 5.1: User flow of MoodBalance

5.1 Start of the app

As we can see in Figure 5.1 the journey starts in a SplashScreen [Figure 5.2a] where the app is presented while loading the main content. The system checks if the user is already logged in and if not, it sends the user to the registration page [Figure 5.2b]. The SplashScreen provides a visually pleasing transition from the moment the application is launched until the main user interface is displayed.

For registration, the user clicks on the Spotify login button and the system will redirect the user to the Spotify account service [Figure 5.3a]. If it is the user's first time accessing the application, they will have to accept that the application may access the necessary data it requests [Figure 5.3b)]. This step is necessary for the use of the application.

In case the user is already logged in, the system will already have the necessary user token and from the SplashScreen the user will be forwarded directly to the emotion selection page.

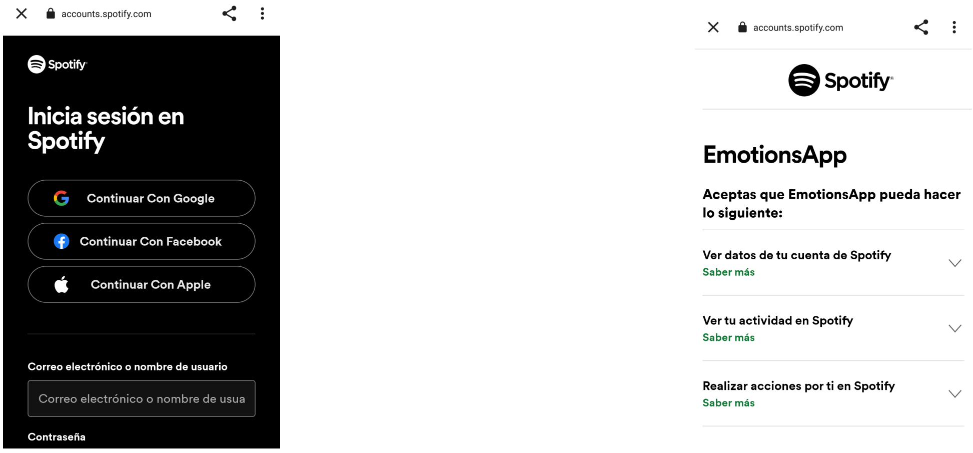


(a) Splash Screen

(b) Login Screen

Figure 5.2: Splash and Login Screens

The retrieval of the required token follows the steps explained in the chapter 4. Once the permissions have been accepted, the user will be redirected back to the app.



Inicia sesión en Spotify

Correo electrónico o nombre de usuario

Correo electrónico o nombre de usua

Contraseña

EmotionsApp

Aceptas que EmotionsApp pueda hacer lo siguiente:

Ver datos de tu cuenta de Spotify

Ver tu actividad en Spotify

Realizar acciones por ti en Spotify

(a) Spotify LogIn Service

(b) Application Permissions

Figure 5.3: Spotify Accounts Service

5.2 Recommendation generation

The generation of a recommendation begins on the emotion selection screen. On this screen, the user is presented with the option of 8 emotions in a carousel format. Each emotion is associated with specific tempo, valence, energy, danceability, and loudness values. Once the emotion is selected, the user will proceed to the next screen.

For this specific case, the user selected the “Awful” emotion.

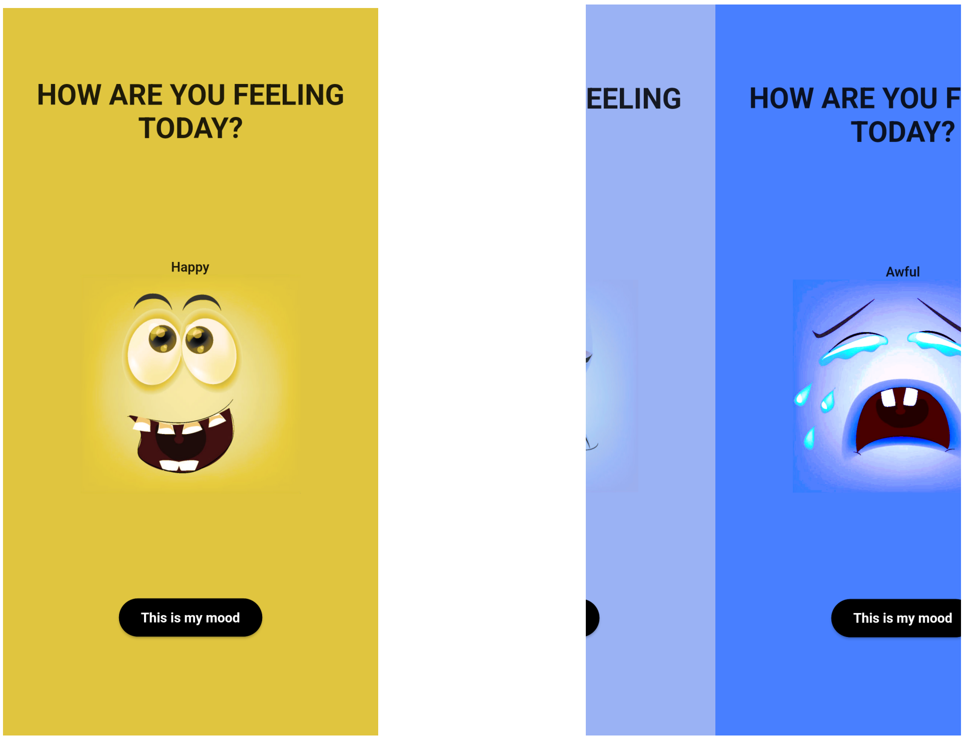
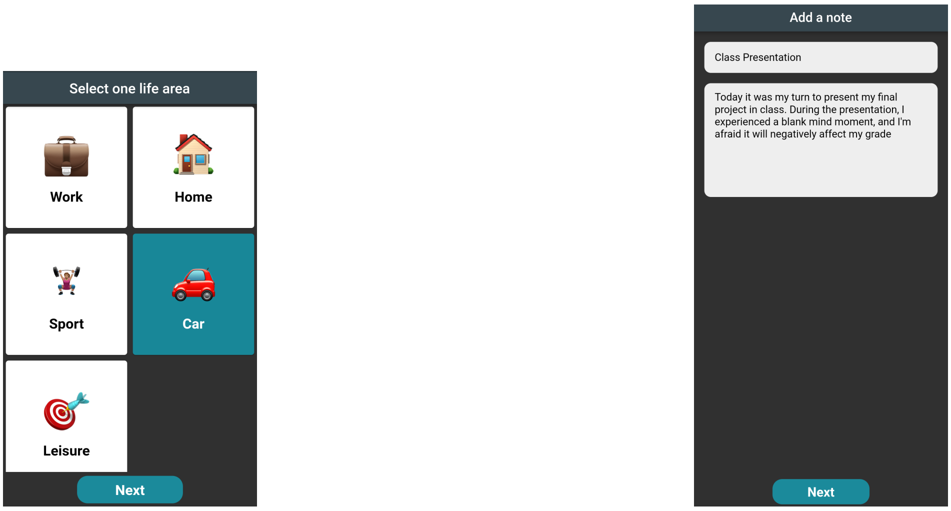


Figure 5.4: Initial Emotion Screen

At this point, the user will select the area they are currently in and where they are experiencing the selected emotion. Based on research studies, different genres of music are associated with different areas[24][25][26]. Therefore, as seen in the subsection 4.3.1, each area will have a couple of music genres associated with it, which will make the experience more personalized.

Once the area is selected, the user moves to the Description screen. On this screen, the user can enter a text describing why they have felt that way. The user doesn’t need to fill in these fields; they can proceed to the next screen by clicking a button.

In this case, “Car” is selected as the current area and a small text has been written explaining that the user had to give a presentation in class and experienced a blank mind moment.



Class Presentation

Today it was my turn to present my final project in class. During the presentation, I experienced a blank mind moment, and I'm afraid it will negatively affect my grade

Figure 5.5: Area and Description Screen

The last screen before generating the recommendation is the selection of the target emotion. In this screen, the same 8 options as in the initial screen are provided. Once the target emotion is selected, the user will have to press the 'generate recommendation' button, and the system will begin the process.

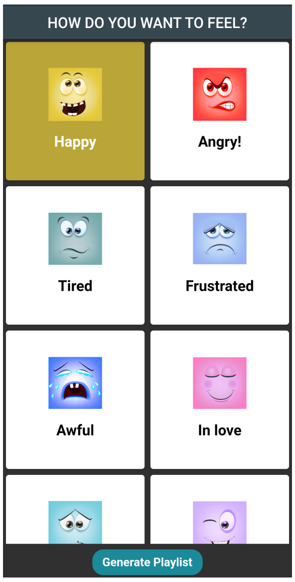
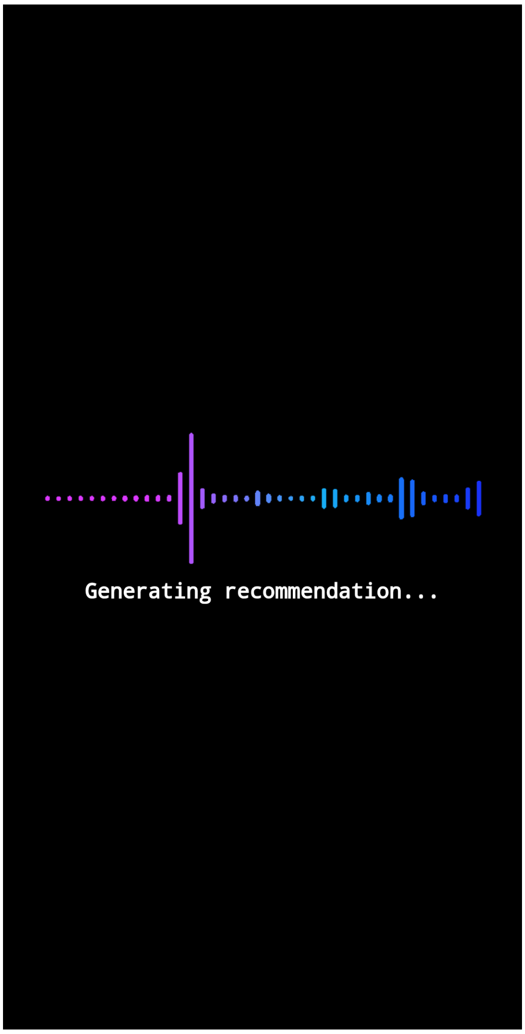


Figure 5.6: Goal Emotion Screen

Once the "Generate Playlist" button is pressed, the system uses the token to make calls to the Spotify API. The first call is made to retrieve the top 5 most listened songs in the last 30 days. This information is used to recommend songs based on the user's musical preferences.

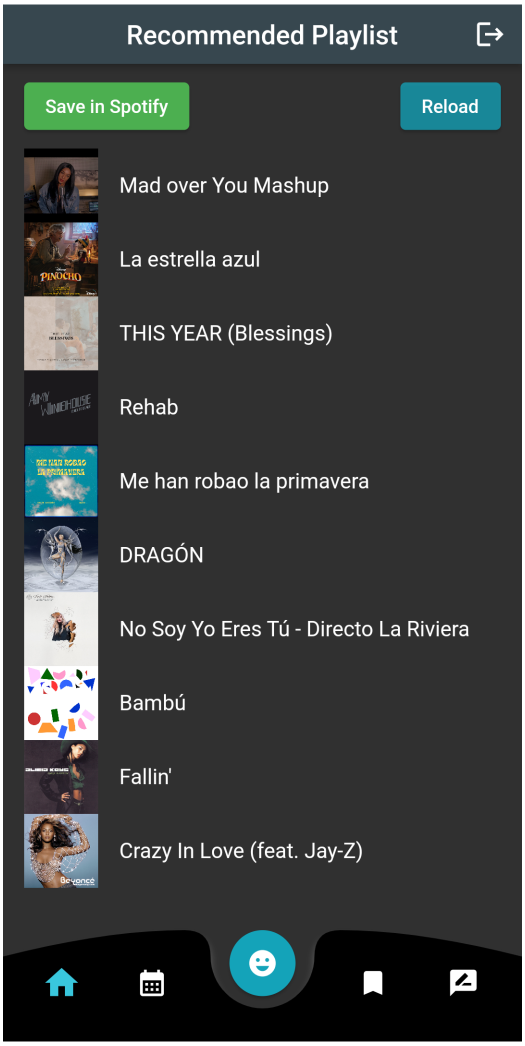
When the most listened songs have been obtained, the system makes 10 API calls, each one generating a recommendation based on these songs, the genre selected according to the chosen area, and a value for each of the features associated with emotions. As explained in Chapter 4 in the data flow section, a progression of 10 values is made for each feature, starting from the initial emotion and ending at the target emotion, following the optimal path resulting from the algorithm developed with Python.

While the system is making these requests, the user will be redirected to a loading screen [Figure 5.7] until the process is completed. At the end of the calls and having a set of 10 recommendations, they will be presented on the recommendation page [Figure 5.8] to the user as a playlist that can be saved.



Generating recommendation...

Figure 5.7: Loading Screen



Recommended Playlist

Mad over You Mashup

La estrella azul

THIS YEAR (Blessings)

Rehab

Me han robao la primavera

DRAGÓN

No Soy Yo Eres Tú - Directo La Riviera

Bambú

Fallin'

Crazy In Love (feat. Jay-Z)

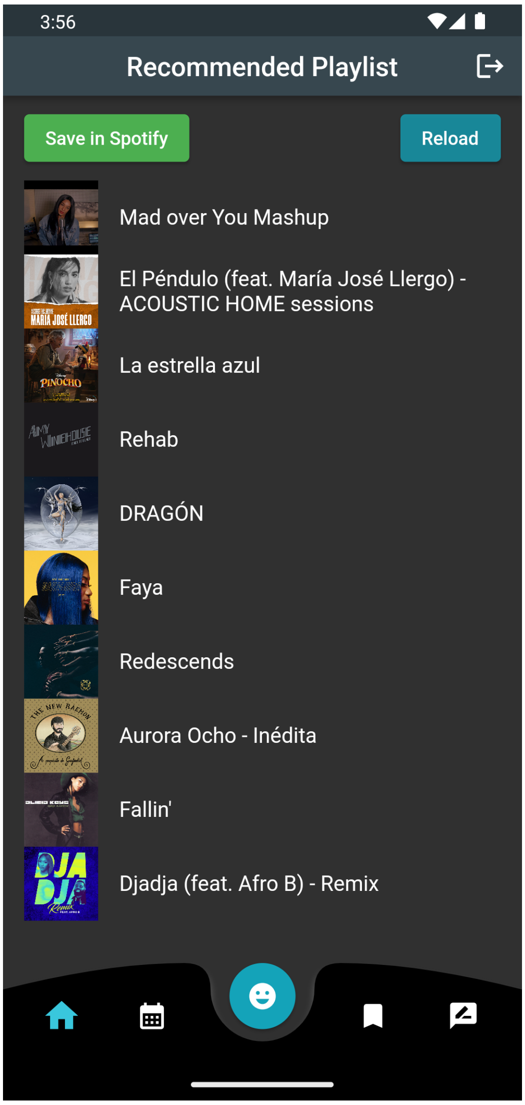
Figure 5.8: Recommendation Screen

5.3 Other functionalities

In this section, we find other possible functionalities that the user may find while using the app.

Reload Playlist

This functionality is activated by simply pressing a button located on the recommendation results screen. This button triggers the same API calls with the same features as the last recommendation.



Mad over You Mashup

El Péndulo (feat. María José Llergo) - ACOUSTIC HOME sessions

La estrella azul

Rehab

DRAGÓN

Faya

Redescends

Aurora Ocho - Inédita

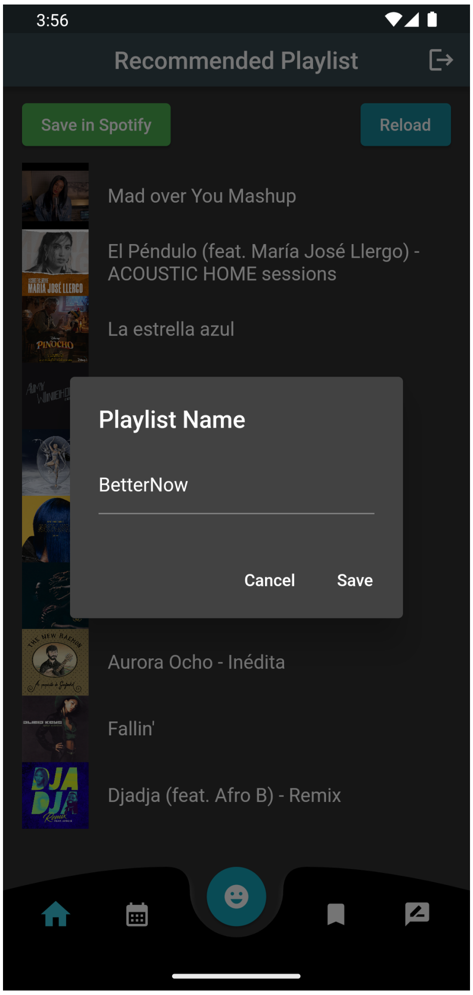
Fallin'

Djadja (feat. Afro B) - Remix

Figure 5.9: Result when reload

Save Playlist

The user can save the songs as a playlist on their Spotify account by clicking the button located on the recommendation results screen, just like the reload button. This button will prompt a dialog box where the user can enter the desired name for the playlist to be saved. Once saved, the system will display a survey dialog for the user to provide feedback on the recommendation and the system itself.



Mad over You Mashup

El Péndulo (feat. María José Llergo) - ACOUSTIC HOME sessions

La estrella azul

Playlist Name

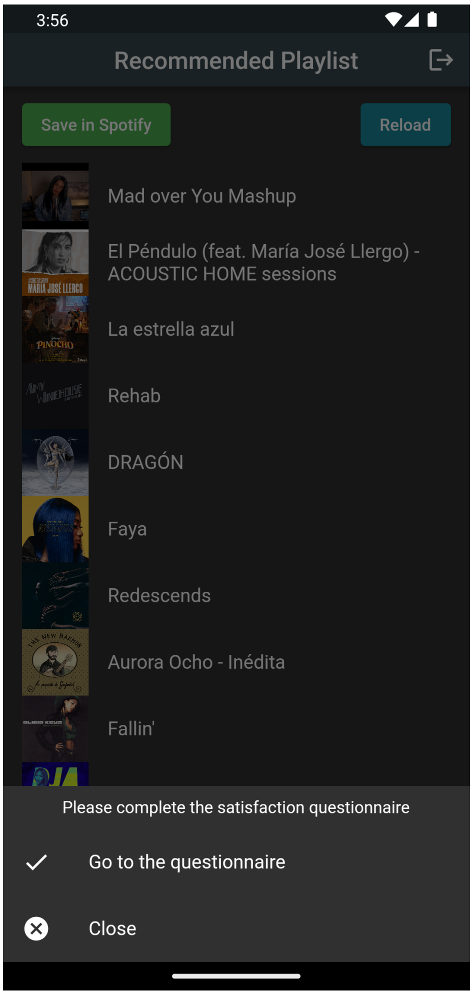
BetterNow

Aurora Ocho - Inédita

Fallin'

Djadja (feat. Afro B) - Remix

Figure 5.10: Playlist Name dialog



Mad over You Mashup

El Péndulo (feat. María José Llergo) - ACOUSTIC HOME sessions

La estrella azul

Rehab

DRAGÓN

Faya

Redescends

Aurora Ocho - Inédita

Fallin'

Please complete the satisfaction questionnaire

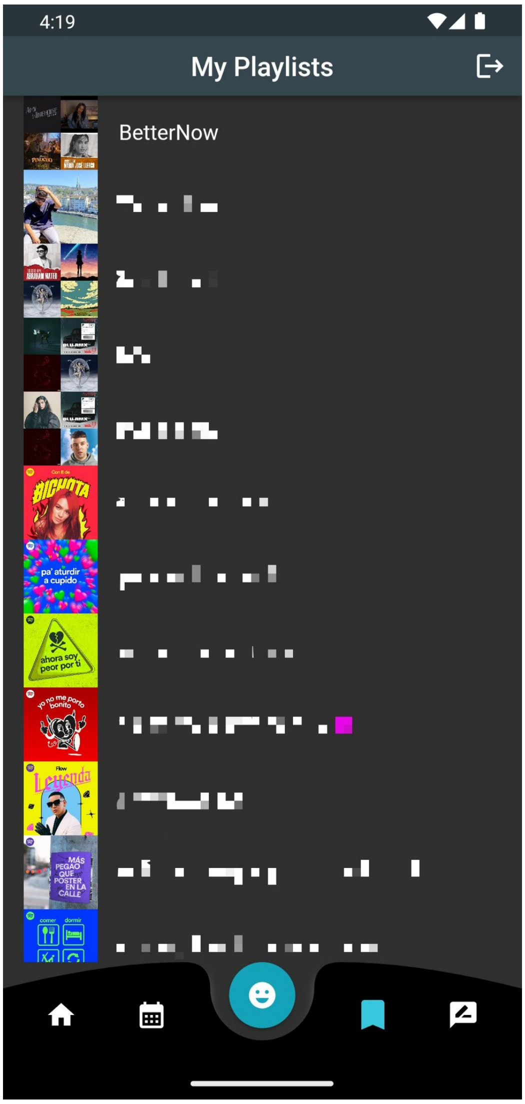
✓ Go to the questionnaire

Ⓧ Close

Figure 5.11: fill questionnaire dialog

My Playlists Screen

The user can access this page through the navigation bar. On this page, the user will have a list of the playlists they have saved as a Spotify user. Additionally, if the user clicks on any of the playlists, they will navigate to a page displaying the songs in that playlist. This way, the user can verify if the recommendations they have chosen have been saved correctly to their user account.



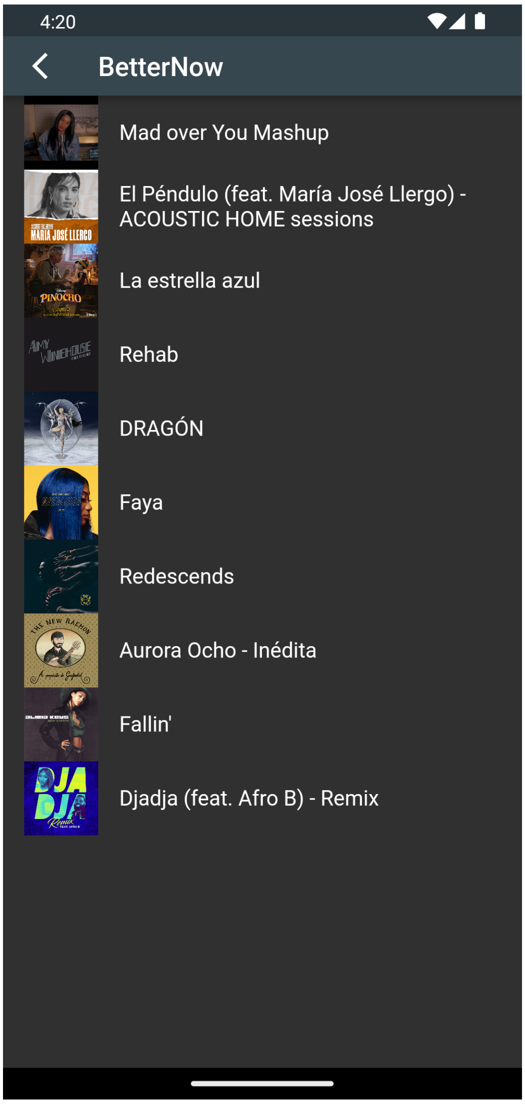
4:19

My Playlists

BetterNow

[redacted]

Figure 5.12: List of Playlists



4:20

BetterNow

Mad over You Mashup

El Péndulo (feat. María José Llergo) - ACOUSTIC HOME sessions

La estrella azul

Rehab

DRAGÓN

Faya

Redescends

Aurora Ocho - Inédita

Fallin'

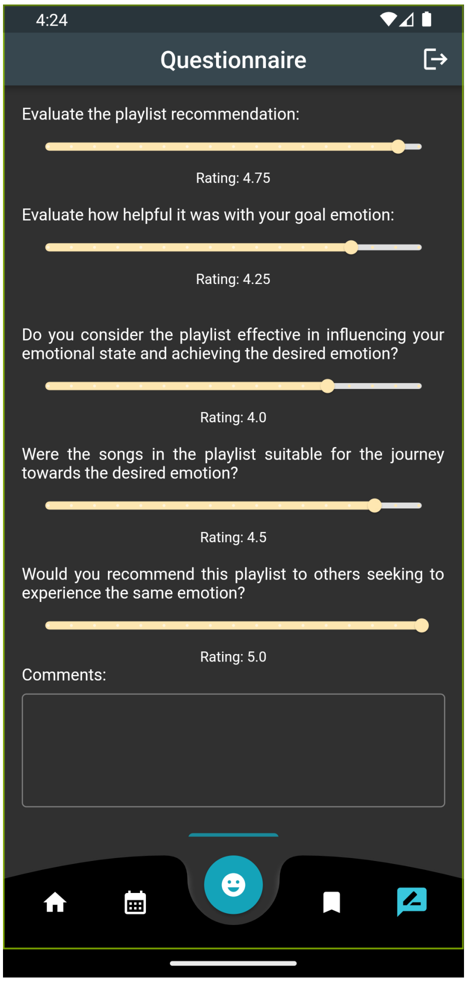
Djadja (feat. Afro B) - Remix

Figure 5.13: Playslist Details

Questionnaire Screen

This page can be accessed in two ways: through the navigation bar or through the dialog that appears after saving the recommendations to a Spotify playlist. In the questionnaire, several questions are asked regarding the playlist recommendation and how it has influenced the user's mood. The ratings will range from 1 to 5.

The user will also have the option to provide additional detailed comments in a comment box.



Questionnaire

Evaluate the playlist recommendation:

Rating: 4.75

Evaluate how helpful it was with your goal emotion:

Rating: 4.25

Do you consider the playlist effective in influencing your emotional state and achieving the desired emotion?

Rating: 4.0

Were the songs in the playlist suitable for the journey towards the desired emotion?

Rating: 4.5

Would you recommend this playlist to others seeking to experience the same emotion?

Rating: 5.0

Comments:

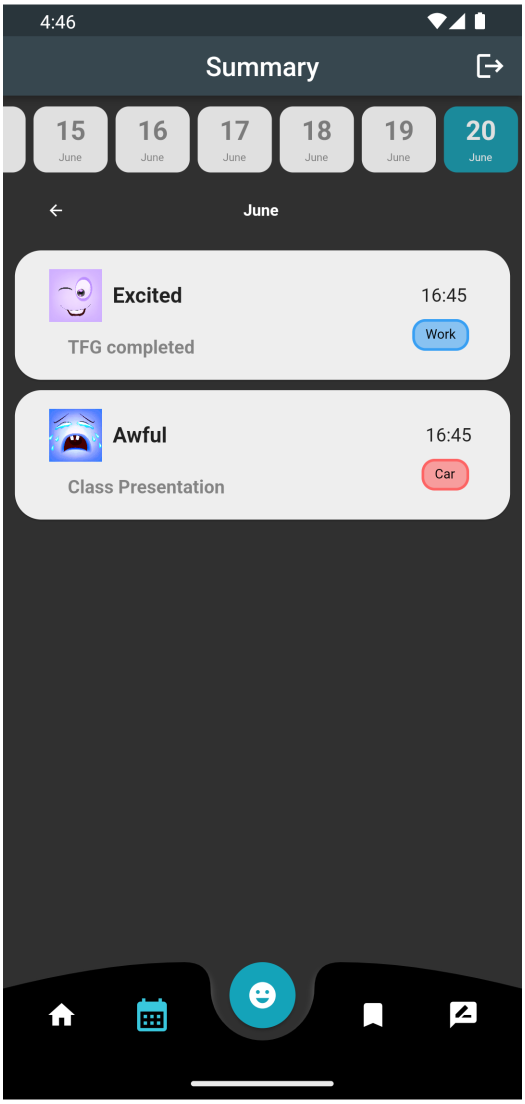
Figure 5.14: Questionnaire Screen

Calendar Screen

This screen will be accessible from the navigation bar, where in addition to accessing the previously mentioned screens, the user can initiate a new recommendation by starting the process from the emotion selection screen.

From this page, the user can observe the emotions recorded over the days. They can select both the day and the month, and the emotions registered on that date will be displayed. All this data will be obtained from the Hive database, where all the data has been stored each time a new recommendation was made.

If the user clicks on an emotion, they will be taken to a details page where they will find both the description they have provided and the star ratings they have given for the recommendation related to that emotion.



4:46

Summary

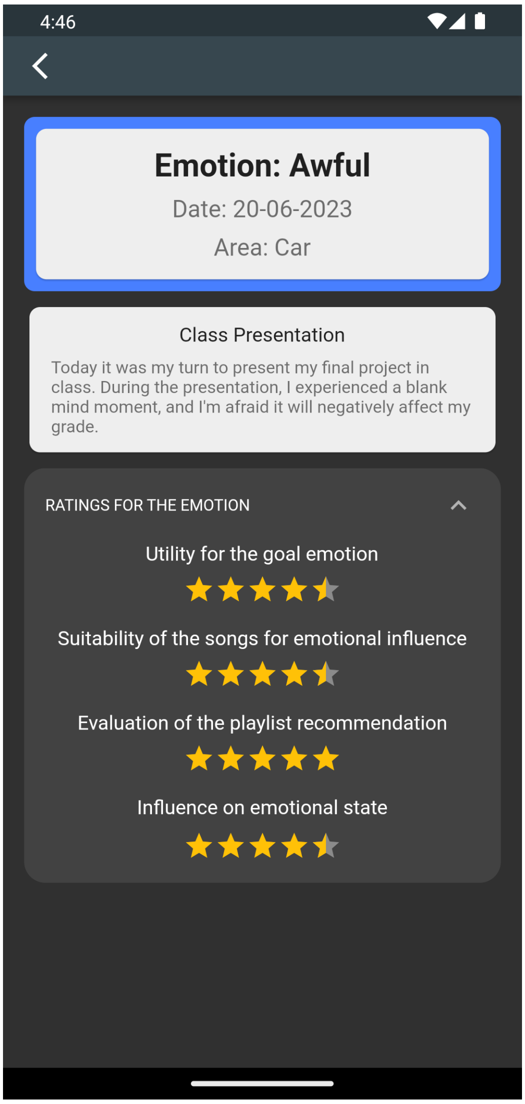
15 June	16 June	17 June	18 June	19 June	20 June

June

Excited 16:45
TFG completed
Work

Awful 16:45
Class Presentation
Car

Figure 5.15: Emotion’s Calendar



4:46

Emotion: Awful
Date: 20-06-2023
Area: Car

Class Presentation
Today it was my turn to present my final project in class. During the presentation, I experienced a blank mind moment, and I'm afraid it will negatively affect my grade.

RATINGS FOR THE EMOTION

Utility for the goal emotion
★★★★⯪

Suitability of the songs for emotional influence
★★★★⯪

Evaluation of the playlist recommendation
★★★★★

Influence on emotional state
★★★★⯪

Figure 5.16: Emotion Details

5.3.1 User Flow for other functionalities

The following figure shows the flowchart that the user follows once a playlist has been recommended. The user will be able to perform the functionalities that have been described and shown in this section.

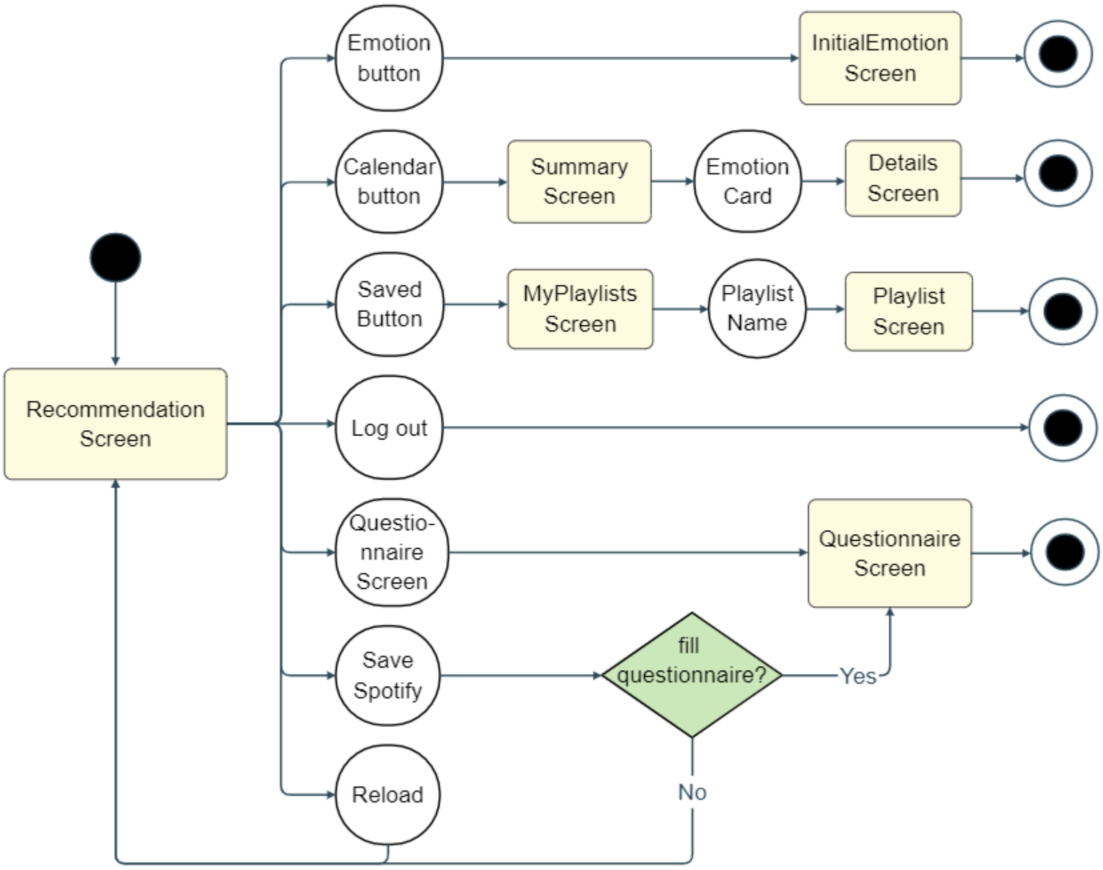


Figure 5.17: User flow other functionalities

CHAPTER 6

Conclusions and future work

In this chapter, we will describe the conclusions extracted from this project and the thoughts about future work.

6.1 Conclusions

The emotional dimension of our well-being plays a crucial role in our overall health and in our ability to derive satisfaction from life experiences. That is why a system based on musical recommendations was proposed to promote emotional regulation, improving the user's mental and emotional well-being.

In this project, first, a study of the state of the art in the field of emotions and their relationship with music was carried out. Different emotional theories were explored, especially Russell's Circumplex model, which provides a basis for understanding the interaction between music and human emotions. This study allowed establishing the theoretical foundations necessary for the development of this project, an emotion-based music recommendation application.

It has been observed that the induction of a target emotional state is generally not achieved by listening to a single song, but it is recommended to listen to different songs that generate intermediate emotions. This suggests the importance of considering the entire emotional trajectory to achieve an effective emotional transition. Therefore, the system uses an algorithm that finds the path between two emotions. The application searches for the optimal resulting path between the initial and the final emotion following Russell's Circumplex model mentioned above.

In the app, users can select an initial emotion and the area in which they have felt it. To get better help in emotion regulation the user selects a target emotion and has the possibility to provide a summary. With the data entered, the system uses the algorithm and finds the optimal path between emotions.

The app uses audio feature values, such as energy, volume, tempo, danceability, and valence, to calculate a progression of 10 values for each feature from the initial emotion to the final emotion through the values of the emotions in the optimal path. With these value sets, the application uses the Spotify API to obtain recommendations of 10 songs, one for each feature set, and displays them to the user on a recommendation screen. The use of Spotify is a key element in the system, which also facilitates the personalization of the recommendations, allowing the collection of the most listened-to songs in the last 30 days to know the user's tastes.

Additionally, to the music recommendation functionality, the application allows the user to access their saved playlists in their Spotify account and complete a questionnaire. The data of the initial emotion, final emotion, area of interest, and answers to the questionnaire are stored locally. This data is also collected to display the daily log to the user, where the saved data is displayed according to the day.

In summary, this work has succeeded in developing an emotion-based music recommendation application. The application offers a personalized and emotionally relevant experience for users, allowing them to explore and discover music that aligns with their emotional states and preferences.

6.2 Achieved goals

The main objective of the music recommendation application is to manage and enhance personal emotions by providing users with a personalized and emotionally satisfying music experience. Throughout this project, several key objectives have been achieved that contribute to the attainment of this goal. The main achievements are described below:

Study of the characteristics of music and its relationship to emotions: Extensive research was conducted to understand how music can influence human emotions. Various musical elements, such as rhythm, melody, harmony, and instrumentation, were analyzed to identify how they relate to different emotional states. This study provided a solid foundation for the development of the emotion-based music recommendation system.

Application Development: A functional music recommendation application has been designed and implemented that allows users to select an initial emotion, an emotional area, and a target emotion to generate an optimal path between the selected emotions. In addition, the application allows users to enter a summary and offers additional features, such as the visualization of playlists and the possibility to complete a questionnaire. This development has laid the groundwork for providing users with a personalized and emotionally relevant music experience.

Integration with the Spotify API: Successful integration with the Spotify API has been achieved, allowing the application to access Spotify's vast music catalog and obtain song recommendations based on user's preferences and emotions. This enriches the music experience by providing users with a wide selection of songs that match their emotional state and preferences.

Implementation of the Optimal Path Finding Algorithm: An algorithm based on Russell's Circumplex theory has been developed to find the optimal path between the initial emotion and the target emotion. This algorithm takes into account the emotional transitions and the rewards associated with each transition to guide the generation of the optimal path. This ensures that the music recommendations consistently match the user's desired emotional progression.

6.3 Future work

The work done has opened up new opportunities for future research and improvements in emotion-based music recommendation. Some possible lines of work are presented below:

- Validation of the rewards used: In this work, the rewards used in the optimal path finding algorithm have been assigned in a pre-defined way. As future work, an improvement of the algorithm could be carried out using reinforcement learning techniques that include the user's opinions in each recommendation.
- Improving the accuracy of recommendations: Different recommendation techniques and algorithms can be explored to improve the accuracy of music recommendations. This could include applying machine learning approaches and utilizing more audio features and song metadata.
- Integration of Natural Language Processing (NLP) techniques: The application of NLP techniques can be investigated to improve the understanding of the summaries provided by users. This could allow a better interpretation of their preferences and emotions, which in turn could lead to more accurate and personalized recommendations.
- Emotional impact assessment: It would be interesting to conduct studies to assess the emotional impact of music recommendations on users. This could involve collecting subjective feedback, physiological measurements, or neuroscience studies to understand how the recommended music influences users' emotions.

In conclusion, the development of the emotion-based music recommendation application has laid the groundwork for future research and improvements in the field of music recommender systems. The exploration of emotions, the application of NLP techniques, and the improvement of recommendation accuracy are key areas that can be addressed in future work. By continuing to research and improve this application, a more personalized and emotionally impactful music experience for users can be achieved.

APPENDIX A

Impact of this project

This appendix reflects, quantitatively or qualitatively, on the possible impact of this project.

A.1 Social impact

The music recommendation application developed in this project has an important social impact by providing users with a personalized and emotionally satisfying music experience. Music is a universal form of expression and communication, and this application helps people discover songs that fit their emotional state and musical preferences. This can have a positive impact on users' emotional well-being, as music plays an important role in managing emotions and improving mood.

In addition, the app promotes musical diversity by offering recommendations based on different emotions and emotional areas. This allows users to explore and discover music from various genres, styles, and cultures, which contributes to the appreciation of cultural diversity and fosters inclusion and respect for different artistic expressions.

A.2 Economic impact

Successful integration with the Spotify API in this application can have a positive economic impact. By providing users with music recommendations based on their preferences and emotions, the use and subscription to streaming music services such as Spotify is promoted. This can increase the demand for music platform subscriptions and generate revenue for the music industry as a whole.

Moreover, the application can generate economic opportunities for artists and record labels by promoting the playback and discovery of their music. By recommending songs based on specific emotions and preferences, the application can help emerging artists reach new audiences and increase their visibility in the music industry.

A.3 Enviromental impact

In terms of environmental impact, this application has minimal direct impact. The digital nature of the application and its focus on music recommendation do not generate large consumption of natural resources or polluting emissions. However, it is important to consider the energy consumption associated with the use of electronic devices and data transmission over the Internet.

A.4 Ethical impact

The music recommendation application has been developed with an ethical approach in mind. Users' privacy is respected and relevant data protection regulations and policies are followed. The data collected, such as selected emotions and questionnaire responses, is used exclusively to improve the user experience and provide more accurate music recommendations. It is ensured that the data is not used for malicious purposes or shared with third parties without the user's consent.

Likewise, we have sought to avoid bias and discrimination in music recommendations. Unbiased techniques and algorithms have been used to provide recommendations based on emotions and personal preferences, without discrimination based on gender, race, religion or other identity aspects. An inclusive and diverse experience has been fostered, where all musical expressions are valued and respected.

APPENDIX B

Economic budget

This appendix details the financial budget for the music recommendation application development project. The budget is divided into several key categories:

B.1 Physical resources

In the development of this project, various physical resources and software licenses have been used, which have implied certain costs. The resources used and their estimated prices are detailed below:

- MSI laptop with Windows 11: The project has been developed on an MSI laptop, which meets the hardware and software requirements necessary to run the programming tools and development environments used. The estimated cost of the laptop is around 1500€. Considering a depreciation period of 5 years, the annual depreciation cost would be approximately 300€.

- Mobile devices: An Android device BQ Aquaris X Pro and an iPhone 11 have been used to test and verify the operation of the application in a real mobile environment with different operating systems. The estimated cost of the BQ Aquaris X Pro is around 250€, and the estimated cost of the iPhone 11 is around 600€. Assuming a depreciation period of 2 years for the android device, the annual depreciation cost would be approximately 125€ for the BQ Aquaris X Pro. For the iOS device, a 5-year depreciation period is assumed, so the depreciation cost would be 120€.

the personal computer has been used for only 6 months and the mobile phones were used only in the last month of testing. Therefore, the cost of the computer would be 150€ while the cost of the phones would be 10,50€ for the BQ and 10€ for the iPhone.

B.2 Human resources

During this project we estimate a time investment of about 390 hours, covering activities such as research, conceptualization, design, implementation, testing, documentation and presentation of the project.

Considering my experience and the level of complexity of the project, I have established an hourly rate of 13€ to estimate the costs. This rate takes into account my knowledge and skills, as well as the effort invested in the project.

By multiplying the estimated number of hours (390) by the hourly rate (14€), the total human resource cost for this project is estimated at approximately 5,460€.

B.3 Taxes

Regarding fees, it is important to take into account the estimated total price of the resources and services used in this project, including hardware, software licenses and services rendered. In accordance with current legislation, a value added tax (VAT) of 21% will be applied to the total price.

B.4 Summary table

Besides the costs indicated in the previous sections, we have also taken into account indirect costs (general expenses) and industrial profit

LABOR COST (direct cost)	**Hours**	**Hourly Rate**	**Total**
	390	13€	**5,070 €**

MATERIAL RESOURCES COST (direct cost)	**Purchase Price**	**Months of Use**	**Amortization (in years)**	**Total**
Personal computer (Including Software)	€1,500.00	6	5	€150.00
Android mobile phone (BQ Aquaris X Pro)	€250.00	1	2	€10.50
iOS mobile phone (iPhone 11)	€600.00	1	5	€10.00

TOTAL COST OF MATERIAL RESOURCES	**€170.50**

GENERAL EXPENSES (indirect costs)	15%	over DC	**€786.10**
INDUSTRIAL PROFIT	6%	over DC + IC	**€361.60**

SUBTOTAL BUDGET		**€6,388.20**
APPLICABLE VAT	21%	**€1,341.52**

TOTAL BUDGET	**€7,729.72**

Table B.1: Economic Budget

APPENDIX C

Paths and values for emotions

In the following code, you can see all the calculated paths between all the emotions. the path name is composed of InitialEmotion_GoalEmotion:

```
 Map<String, List<String>> caminosOptimos = {
’Happy_Excited’: [’Happy’, ’Excited’],
’Angry!_Excited’: [’Angry!’, ’Excited’],
’Tired_Exited’: [’Tired’, ’Relaxed’, ’Pleasent’, ’Happy’,
   ’Excited’],
’Frustrated_Excited’: [’Frustrated’, ’Excited’],
’Awful_Excited’: [’Awful’, ’Angry!’, ’Excited’],
’Pleasent_Excited’: [’Pleasent’, ’Happy’, ’Excited’],
’Relaxed_Excited’: [’Relaxed’, ’Pleasent’, ’Happy’,
   ’Excited’],
’Excited_Excited’: [’Excited’],
’Happy_Happy’: [’Happy’],
’Angry!_Happy’: [’Angry!’, ’Excited’, ’Happy’],
’Tired_Happy’: [’Tired’, ’Relaxed’, ’Pleasent’, ’Happy’],
’Frustrated_Happy’: [’Frustrated’, ’Tired’, ’Relaxed’,
   ’Pleasent’, ’Happy’],
’Awful_Happy’: [’Awful’, ’Angry!’, ’Excited’, ’Happy’],
’Pleasent_Happy’: [’Pleasent’, ’Happy’],
’Relaxed_Happy’: [’Relaxed’, ’Pleasent’, ’Happy’],
’Excited_Happy’: [’Excited’, ’Happy’],
’Pleasent_Pleasent’: [’Pleasent’],
’Angry!_Pleasent’: [’Angry!’, ’Excited’, ’Happy’,
   ’Pleasent’],
’Tired_Pleasent’: [’Tired’, ’Relaxed’, ’Pleasent’],
’Frustrated_Pleasent’: [’Frustrated’, ’Tired’, ’Relaxed’,
   ’Pleasent’],
’Awful_Pleasent’: [’Awful’, ’Excited’, ’Happy’, ’Pleasent’],
’Happy_Pleasent’: [’Happy’, ’Pleasent’],
’Relaxed_Pleasent’: [’Relaxed’, ’Pleasent’],
’Excited_Pleasent’: [’Excited’, ’Happy’, ’Pleasent’],
’Relaxed_Relaxed’: [’Relaxed’],
’Angry!_Relaxed’: [’Angry!’, ’Excited’, ’Happy’, ’Relaxed’],
’Tired_Relaxed’: [’Tired’, ’Relaxed’],
’Frustrated_Relaxed’: [’Frustrated’, ’Tired’, ’Relaxed’],
’Awful_Relaxed’: [’Awful’, ’Excited’, ’Happy’, ’Relaxed’],
’Pleasent_Relaxed’: [’Pleasent’, ’Relaxed’],
’Happy_Relaxed’: [’Happy’, ’Pleasent’, ’Relaxed’],
’Excited_Relaxed’: [’Excited’, ’Happy’, ’Relaxed’],
’Happy_Tired’: [’Happy’, ’Relaxed’, ’Tired’],
’Angry!_Tired’: [’Angry!’, ’Frustrated’, ’Awful’, ’Tired’],
’Frustrated_Tired’: [’Frustrated’, ’Tired’],
’Awful_Tired’: [’Awful’, ’Tired’],
’Tired_Tired’: [’Tired’],
’Pleasent_Tired’: [’Pleasent’, ’Relaxed’, ’Tired’],
’Relaxed_Tired’: [’Relaxed’, ’Tired’],
’Excited_Tired’: [’Excited’, ’Happy’, ’Relaxed’, ’Tired’],
’Happy_Awful’: [’Happy’, ’Relaxed’, ’Tired’, ’Awful’],
’Angry!_Awful’: [’Angry!’, ’Frustrated’, ’Awful’],
’Tired_Awful’: [’Tired’, ’Awful’],
’Frustrated_Awful’: [’Frustrated’, ’Awful’],
’Awful_Awful’: [’Awful’],
’Pleasent_Awful’: [’Pleasent’, ’Relaxed’, ’Tired’, ’Awful’],
  'Relaxed_Awful': ['Relaxed', 'Tired', 'Awful'],
  'Excited_Awful': ['Excited', 'Happy', 'Relaxed', 'Tired',
     'Awful'],
  'Happy_Frustrated': ['Happy', 'Relaxed', 'Tired',
     'Frustrated'],
  'Angry!_Frustrated': ['Angry!', 'Frustrated'],
  'Tired_Frustrated': ['Tired', 'Frustrated'],
  'Frustrated_Frustrated': ['Frustrated'],
  'Awful_Frustrated': ['Awful', 'Frustrated'],
  'Pleasent_Frustrated': ['Pleasent', 'Relaxed', 'Tired',
     'Frustrated'],
  'Relaxed_Frustrated': ['Relaxed', 'Tired', 'Frustrated'],
  'Excited_Frustrated': ['Excited', 'Frustrated'],
  'Happy_Angry!': ['Happy', 'Excited', 'Angry!'],
  'Angry!_Angry!': ['Angry!'],
  'Tired_Angry!': ['Tired', 'Frustrated', 'Angry!'],
  'Frustrated_Angry!': ['Frustrated', 'Angry!'],
  'Awful_Angry!': ['Awful', 'Frustrated', 'Angry!'],
  'Pleasent_Angry!': ['Pleasent', 'Happy', 'Excited',
     'Angry!'],
  'Relaxed_Angry!': ['Relaxed', 'Happy', 'Excited', 'Angry!'],
  'Excited_Angry!': ['Excited', 'Angry!'],
};
```

The following code shows the ranges assigned to each audio feature for each emotion in the system

```
    Map<String, Map<String, List<double>>> emotionValues = {
   'Happy': {
     'target_tempo': [120.0, 140.0],
     'target_energy': [0.7, 0.9],
     'target_valence': [0.6, 0.8],
     'target_danceability': [0.7, 0.9],
     'target_loudness': [-8.0, -6.0]
   },
   'Frustrated': {
```

```
   ’target_tempo’: [90.0, 110.0],
   ’target_energy’: [0.5, 0.7],
   ’target_valence’: [0.3, 0.5],
   ’target_danceability’: [0.4, 0.6],
   ’target_loudness’: [-10.0, -8.0]
  },
  ’Angry!’: {
   ’target_tempo’: [100.0, 120.0],
   ’target_energy’: [0.8, 1.0],
   ’target_valence’: [0.4, 0.6],
   ’target_danceability’: [0.7, 0.9],
   ’target_loudness’: [-6.0, -4.0]
  },
  ’Pleasent’: {
   ’target_tempo’: [80.0, 100.0],
   ’target_energy’: [0.6, 0.8],
   ’target_valence’: [0.7, 0.9],
   ’target_danceability’: [0.6, 0.8],
   ’target_loudness’: [-8.0, -6.0]
  },
  ’Awful’: {
   ’target_tempo’: [40.0, 60.0],
   ’target_energy’: [0.1, 0.3],
   ’target_valence’: [0.1, 0.3],
   ’target_danceability’: [0.1, 0.3],
   ’target_loudness’: [-16.0, -14.0]
  },
  ’Tired’: {
   ’target_tempo’: [60.0, 80.0],
   ’target_energy’: [0.3, 0.5],
   ’target_valence’: [0.2, 0.4],
   ’target_danceability’: [0.2, 0.4],
   ’target_loudness’: [-12.0, -10.0]
  },
  ’Relaxed’: {
   ’target_tempo’: [80.0, 100.0],
   ’target_energy’: [0.4, 0.6],
    ’target_valence’: [0.5, 0.7],
    ’target_danceability’: [0.4, 0.6],
    ’target_loudness’: [-10.0, -8.0]
  },
  ’Excited’: {
    ’target_tempo’: [140.0, 160.0],
    ’target_energy’: [0.9, 1.0],
    ’target_valence’: [0.8, 1.0],
    ’target_danceability’: [0.8, 1.0],
    ’target_loudness’: [-4.0, -2.0]
  },
};
```

Bibliography

[1] James A Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[2] Felix Angelov. Bloc. `https://pub.dev/packages/bloc`. Accessed 23-06-2023.

[3] James J. Gross. Emotion regulation: Conceptual and empirical foundations. *Handbook of Emotion Regulation*, pages 3–24, 2007.

[4] Alexandra Linnemann, Beate Ditzen, Jana Strahler, Johanna M Doerr, and Urs M Nater. Music listening as a means of stress reduction in daily life. *Psychoneuroendocrinology*, 60:82–90, 2015.

[5] María García Rodríguez. *La música como transmisora de emociones y su relación con factores emocionales que influyen en el entorno académico.* PhD thesis, Universidad Complutense de Madrid, Madrid, 2022.

[6] Patrik N. Juslin and Daniel Västfjäll. Emotional responses to music: The need to consider underlying mechanisms. *Behavioral and Brain Sciences*, 31(5), 2008.

[7] Maria Calleja-Bautista, Pilar Sanz-Cervera, and Raúl Tárraga-Mínguez. Efectividad de la musicoterapia en el trastorno de espectro autista: estudio de revisión. *Papeles del psicólogo*, 37(2):152–160, 2016.

[8] Christian Gold, Martin Voracek, and Tony Wigram. Effects of music therapy for children and adolescents with psychopathology: A meta-analysis. *Journal of Child Psychology and Psychiatry*, 54(3):329–343, 2013.

[9] Sergio Valverde Marques dos Santos, Luiz Almeida da Silva, Fábio de Souza Terra, Adriele Vieira de Souza, Foued Salmen Espindola, and otros. Asociación de la alfa amilasa salival con la ansiedad y el estrés en profesionales de enfermería. *Revista Latino-Americana de Enfermagem*, 29, 2021.

[10] Ana María Botella Nicolás and José Vicente Gimeno Romero. Psicología de la música y audición musical. distintas aproximaciones. *El Artista*, número 12:74–98, diciembre 2015.

[11] Myriam V. Thoma, Roberto La Marca, Rebecca Brönnimann, Linda Finkel, Ulrike Ehlert, and Urs M. Nater. The effect of music on the human stress response. *Medical Science Monitor*, 19:337–347, 2013.

[12] Psicológicamente (PLM). Emociones ¿qué son y para qué sirven? `https://www.psicologicamenteonline.com/emociones-que-son-y-para-que-sirven`. Accessed 26-04-2023.

[13] Enrique G. Fernández-Abascal and María Pilar Jiménez Sánchez. *Psicología de la Emoción*. Editorial Universitaria Ramón Areces, 2010.

[14] Mariano Chóliz Montañés. *Psicología de la emoción: el proceso emocional*. Universidad de Valencia, 2005.

[15] Sarah Mae Sincero. El estrés y la evaluación cognitiva. `https://explorable.com/es/el-estres-y-la-evaluacion-cognitiva`. Accessed 26-04-2023.

[16] Sarah Mae Sincero. Schachter-singer theory of emotion. `https://explorable.com/schachter-singer-theory-of-emotion`. Accessed 26-04-2023.

[17] Jonathan Posner, James A. Russell, and Bradley S. Peterson. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology*, 17(3):715–734, 2005.

[18] K. R. Scherer and M. R. Zentner. Efectos emocionales de la música: reglas de producción. *Música y emoción: teoría e investigación*, pages 361–387, 2001.

[19] Alf Gabrielsson. The relationship between musical structure and perceived expression. In Susan Hallam, Ian Cross, and Michael Thaut, editors, *The Oxford Handbook of Music Psychology*. Oxford University Press, 2 edition, 2016.

[20] The Dart Team. Dart. `https://dart.dev/`. Accessed 22-03-2023.

[21] Statista. Worldwide software developer working hours. `https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/`. Accessed 15-05-2023.

[22] esflutter. `https://esflutter.dev/`. Accessed 22-03-2023.

[23] XurxoDev. Introducción al patrón bloc. `https://xurxodev.com/introduccion-al-patron-bloc/`. Accessed 23-06-2023.

[24] Anneli B. Haake and Patrik N. Juslin. Effects of music listening on employee well-being. *Journal of Positive Psychology*, 9(6), 2014.

[25] Diogo A. Morgado and et al. The effect of different musical genres on the mood, anxiety, and athletic performance of university students. *Journal of Applied Sport Psychology*, 31(1), 2019.

[26] Costas I. Karageorghis and David-Lee Priest. Music and workout: The effects of background music on exercise behaviors. *Sports Medicine*, 48(5), 2018.

[27] Tiobe index for may 2023. `https://www.tiobe.com/tiobe-index/`. Accessed 27-06-2023.

[28] Brian Granger and Fernando Perez. Jupyter: Thinking and storytelling with code and data. *Nature*, 619(7843):263–266, July 2021.