




Article

# Emotion-Driven Music and IoT Devices for Collaborative Exer-Games

Pedro Álvarez \*, Jorge García de Quirós  and Javier Fabra 

Computer Science and Systems Engineering Department, Engineering Research Institute of Aragon (I3A), University of Zaragoza, 50018 Zaragoza, Spain; jgarciagg@unizar.es (J.G.d.Q.); jfabra@unizar.es (J.F.)

\* Correspondence: alvaper@unizar.es; Tel.: +34-876-555541

**Abstract:** Exer-games are interactive experiences in which participants engage in physical exercises to achieve specific goals. Some of these games have a collaborative nature, wherein the actions and achievements of one participant produce immediate effects on the experiences of others. Music serves as a stimulus that can be integrated into these games to influence players' emotions and, consequently, their actions. In this paper, a framework of music services designed to enhance collaborative exer-games is presented. These services provide the necessary functionality to generate personalized musical stimuli that regulate players' affective states, induce changes in their physical performance, and improve the game experience. The solution requires to determine the emotions that each song may evoke in players. These emotions are considered when recommending the songs that are used as part of stimuli. Personalization seeds based on players' listening histories are also integrated in the recommendations in order to foster the effects of those stimuli. Emotions and seeds are computed from the information available in *Spotify* data services, one of the most popular commercial music providers. Two small-scale experiments present promising preliminary results on how the players' emotional responses match the affective information included in the musical elements of the solution. The added value of these affective services is that they are integrated into an ecosystem of Internet of Things (IoT) devices and cloud computing resources to support the development of a new generation of emotion-based exer-games.

**Keywords:** music and emotions; music recommendations; music information retrieval; wearables; collaborative games; physical exercises



**Citation:** Álvarez, P.; García de Quirós, J.; Fabra, J. Emotion-Driven Music and IoT Devices for Collaborative Exer-Games. *Appl. Sci.* **2024**, *14*, 10251. <https://doi.org/10.3390/app142210251>

Academic Editor: Pedro Couto

Received: 24 September 2024

Revised: 24 October 2024

Accepted: 5 November 2024

Published: 7 November 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Exercise games (or exer-games) combine gamification strategies and physical activity to motivate individuals to engage in physical exercise within a more playful context [1]. Recent technological advances in virtual environments and the Internet of Things (IoT) have enabled the development of new games that enhance the perceived value of the experience and improve the effectiveness of interventions during physical activity. These advantages have drawn the attention of researchers from fields beyond entertainment, particularly in health and well-being domains [2]. In these contexts, games not only impact the physical state of the player but also influence emotional well-being [3]. This additional effect necessitates the integration of technologies capable of assessing and understanding the emotions and affective responses of players during activities.

These technological ecosystems must align with players' expectations and preferences. Various studies have identified key requirements for the design of exer-games from the players' perspective. Findings indicate that the inclusion of music and social aspects is essential for enhancing both the gaming experience and its outcomes [4,5]. Music has been shown to produce positive effects on players, both physically and emotionally. These effects amplify the impact of the exercises on the players and, consequently, increase the long-term benefits of the proposed activities [6]. As a result, music has been consistently

integrated into the development of games, from early commercial exer-games based on dancing to its use in setting the ambiance and tracking players' progress and achievements. More recently, certain acoustic properties of music (primarily rhythm) have been employed to enhance physical activities and guide players toward their goals [7,8]. Despite this, the full potential of music in these games remains underexplored, particularly in terms of personalization, which considers players' musical preferences and the emotions evoked by the music they listen to. Several studies suggest that these emotions are also influenced by the social model employed in the game design [4,6,9]. For instance, multi-player models tend to increase users' motivation more effectively than equivalent single-player games and are generally perceived as more enjoyable.

Recent advances in wearable devices and artificial intelligence have improved methods and tools for recognizing users' emotions. Music emotion recognition systems have leveraged these advances to explore the relationship between music and listeners' emotions. These systems have subsequently been applied to develop music recommendation systems that incorporate the emotional dimension as a decision criterion [10]. Such systems have proven to be powerful tools for personalization, extending beyond the simple consideration of listeners' music preferences. Given the importance of emotions in physical activity, these systems could play a significant role in exer-game design, particularly in the context of music-driven customization of game elements (such as ambiance, progress monitoring, and performance tracking, as well as serving as a motivational tool). Nevertheless, the integration of affective computing solutions in exer-games remains an open challenge. Current proposals primarily focus on monitoring players' physiological parameters and using that data to adjust certain game conditions [11–13]. While these parameters can be correlated with the emotions players experience, only the model in [14] recognizes emotions based on physiological data and uses this information to configure the background music of the game accordingly.

This paper presents a technological solution aimed at integrating emotion-based music into modern exer-games. The game model involves multiple players competing against one another by performing various physical activities in a virtual world. Players participate in a sensorized environment that incorporates IoT devices to manage their actions, monitor the progress of activities, recognize players' emotions during the game, and adjust the playing environment accordingly. The objective is for music and emotions to serve as primary instruments in programming various aspects of the game. Players' emotions are recognized during gameplay and analyzed to introduce changes in their activities and generate stimuli that affect their physical and emotional responses, as well as those of their opponents. Additionally, these stimuli are personalized for each player to enhance the gaming experience and meet individual expectations and preferences. The solution combines IoT architectures and cloud computing technologies, facilitating integration into online games. To the best of our knowledge, no similar technological system exists in the field of gaming.

The paper focuses primarily on describing the emotion-based music system proposed for the aforementioned game model. The main contributions of this proposal are as follows:

- It is a complex engineering result that combines wearables, smart devices, intelligent systems, IoT, service-oriented architectures, and cloud computing.
- It recognizes players' emotions during gameplay.
- It generates musical stimuli capable of influencing players' emotional behavior.
- The music used in these stimuli is automatically selected using emotion-aware recommendation and personalization algorithms.
- The solution is implemented using commercial devices and is integrated into the *Spotify* service platform.

The remainder of this paper is structured as follows. Section 2 reviews the use of music and affective computing solutions in the development of exer-games. Section 3 presents the six-layer architecture based on the IoT paradigm designed for the implementation of exer-games that incorporate music and emotions. In Section 4, the emotion-based music system

is described, including the wearable devices used by players, the systems for emotion recognition, and the generation of music stimuli based on those emotions. Additionally, a framework of services specializing in music recommendation and personalization is introduced. These music services, detailed in Section 5, are implemented as Azure Functions and leverage core functionalities of the *Spotify* developer platform. Finally, Section 6 presents the main conclusions and outlines directions for future work.

## 2. Related Work

In this section, the music-based exer-games are introduced first. Following that, a review of research works that consider players' emotions during gaming activities is presented. Most of these studies rely on the analysis of physiological data as an alternative to the explicit measurement of emotions.

### 2.1. Music and Exer-Games

Music has always played a significant role in sports as a motivational tool [1,15]. This importance is directly related to the emotions that music can evoke in listeners during physical activity. In the context of exer-games, several studies have explored the positive effects of music on players [6,16], concluding that music increases participants' motivation, reduces the perceived exertion during exercise, and adds value to the activity. As a result of these benefits, music also improves player adherence and engagement in physical exercise. This long-term participation is particularly valuable in games designed for medical rehabilitation or the treatment of disorders [11,17,18].

Despite these advantages, the use of music is not always optimized to enhance the player's experience or the effectiveness of physical interventions. In various studies, players have expressed the need for personalized strategies that maximize the benefits of music during gameplay [5,9]. These strategies can be based on listeners' musical tastes [19] or on their achievements over the course of the game [20]. The latter could more effectively combine the logic of the game and the individual tastes of each player. Unfortunately, many exer-games are primarily designed for specific demographic groups (such as commercial games targeting younger audiences), and often use music that is less suitable for other populations, such as older adults [18]. This mismatch can lead to players losing interest in the game and discontinuing participation early [6,9].

Dancing games were among the first applications to integrate music with physical activity. The advent of virtual reality has enabled a new generation of these games, offering players a different dancing experience [21]. These experiences can be single- or multi-player. Single-player games are mainly designed for entertainment and fitness training, while multi-player games focus on social interaction and improving dancing skills [22]. In all these applications, music is a key gaming element, selected according to the goals of each activity. Some of the most popular commercial dancing games include Dance Dance Revolution, Just Dance, Eye Toy Groove, and Beat Saber [23]. The first two games score the player's ability to move in rhythm with the music, with different songs played according to the difficulty of each level. These games also support multi-player experiences, where players compete without directly interacting for the best score. In the latter two games, players attempt to hit oncoming objects in sync with the music. The level of physical effort required during gameplay is influenced by the song being played. Additionally, these games include a multi-player mode based on challenges, where one player selects a song and challenges others to achieve the best score.

Music is also used as an ambient element in the design of games. The selected music tracks or song excerpts are chosen based on the type of exercises to be completed, the difficulty level and pace of the exercises, or the remaining time for completion, among other criteria. The primary goal is to motivate and guide players toward achieving their milestones [24]. Music can also serve as background for the narrative that ties together the different exercises in a game [25]. The use of narrative to connect exercises is an engagement strategy aimed at increasing the effectiveness of exer-games. In these cases,

game developers typically select the same ambient music for all players. This selection is usually based on the acoustic characteristics of the music and aims to elicit specific emotions in participants (such as relaxation, stress, or happiness). Acoustic characteristics, particularly rhythm, tempo, and beats, have been shown to be highly effective in controlling exercise timing and enhancing player performance [26–28].

Musical rhythm has been explicitly used in the design of some exercise-based games, particularly in the health and well-being fields. In [7], a game is developed that consists of various rhythm-based activities for the rehabilitation of Parkinson's patients. Finger-tapping exercises are synchronized with music rhythms to improve participants' motor skills. Results indicate that these activities increase patient motivation and engagement, as well as the effectiveness of rehabilitation efforts. A similar gaming application is proposed in [8], targeting cognitive disorder treatment through rhythm-based tasks that improve motor coordination, memory, and spatial perception. In [29], the Dividat Senso system [30] is also used to treat cognitive disorders through music-based exercises. This system features a training program consisting of games that combine sound, music videos, and physical activities. Additionally, in the domain of music education, rhythm-based games have also been explored. In [31], players use virtual reality controllers to select rhythm patterns that are translated into game actions. Gaming strategies and music are combined to train players' musical perceptual skills.

## 2.2. Emotions in Exer-Games

One conclusion that can be drawn from the previously discussed games is that music is typically selected during the design phase and remains the same for all players. To increase the effectiveness of music, it would ideally adapt to the progression of the game, players' performance, or their interactions with game elements. These adaptations could be based on the physical and emotional responses of players during different game activities [32]. Players' physiological signals could be monitored over time and analyzed to inform decisions that would allow for more flexible use of music.

Heart rate is one of the most widely used physiological signals in sports, with the advantage that numerous low-cost devices are available for monitoring it. Ref. [33] presents a dancing game in which songs are selected according to the user's heart rate. Players' heart rates are measured to determine their physical exertion at any given moment, and the game integrates a feedback model to select the most appropriate song to regulate that exertion. These decisions are based on the relationship between the tempo of the songs (an acoustic characteristic) and the desired physical demands of each activity. However, this work does not account for players' physical training or experience, which is considered one of its main limitations [34]. Ref. [35] proposes a similar solution, combining heart rate and galvanic skin response to select songs from a limited discography (consisting solely of techno and orchestral music). Decisions are also based on the intensity of the music, and tests are conducted in different adventure games. Although the authors discuss the potential for integrating their solution into exer-games, no results are presented.

In [11–13], various physiological signals are monitored to modify exer-game conditions or to analyze the criteria used in their design. Unfortunately, music is not incorporated into these research proposals. Ref. [11] presents a cycling game, *EXplore Orion*, where heart rate is used to increase or decrease pedaling speed or to suggest breathing exercises that promote relaxation. Ref. [12] conducts experiments to identify the sources of stress experienced by players during gameplay, measuring stress levels through skin conductance and electrocardiograms. The study concludes that factors related to game interaction design and external stimuli are the primary sources of stress. Finally, Ref. [13] analyzes the cardiovascular exertion of players during rounds of an exer-game, seeking to identify correlations between game design criteria and the physical benefits of the activities. In this study, heart rate is used to estimate the exertion of participants.

Exer-games have the potential to elicit positive emotional responses in players, and these emotions, in turn, can influence the success and outcomes of the physical activi-

ties. This bidirectional relationship between exer-games and emotions is examined in [3], which concludes that exer-games improve emotional health, quality of life, and perceived well-being. These improvements are associated with reductions in anxiety, stress, and depressive symptoms, alongside increases in happiness, motivation, and vitality. Consequently, emotions should be inherently integrated into the control and progression of game activities. In [14], players' physiological data are translated into emotional states to dynamically configure background music. Artificial intelligence models and physiological signals are used to recognize these emotions, while an emotion-based game interaction system selects and synchronizes songs with the game's narrative.

### 3. An IoT-Based Architecture for Exer-Games

The main characteristics of our game model are as follows: it is multi-player, based on physical exercises, with each player individually performing exercises in a sensorized environment; players' emotions are recognized and incorporated into the game's control system; musical stimuli, based on those emotions, are generated to induce changes in the players' physical and emotional responses during exercises, and these stimuli are personalized and can be directed toward one or more players (often configured to evoke different reactions in each participant). This model serves as the conceptual foundation for programming various exer-games. Due to the complexity involved in programming these games, a robust software architecture and a set of highly reusable tools are necessary for game development. In this section, we present the architecture designed to implement exer-games based on the model described above.

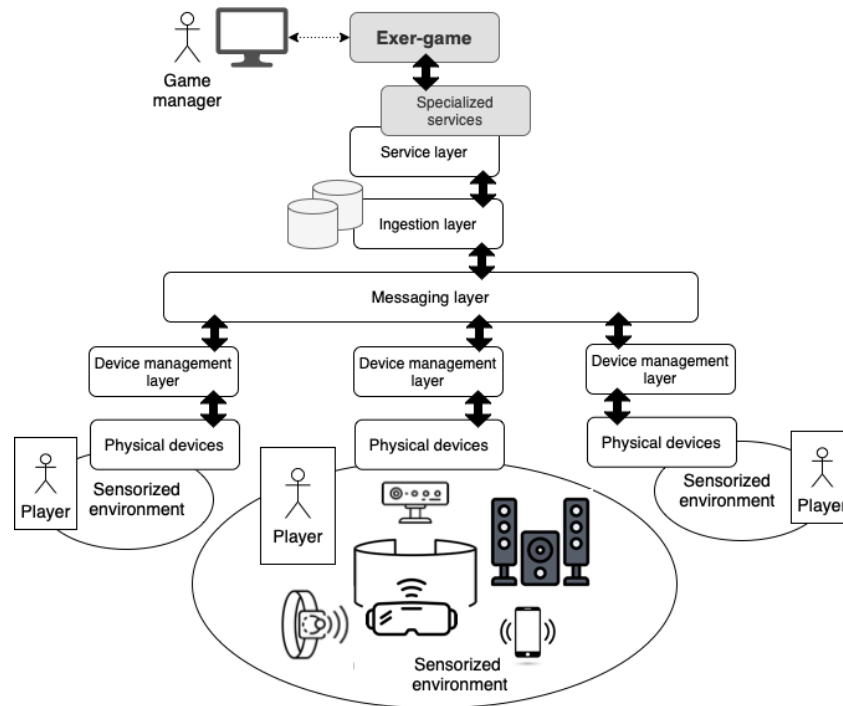
The proposed software architecture draws inspiration from architectural models commonly used in IoT systems. These models have evolved from solutions structured in three functional layers [36,37] to more modern architectures consisting of six layers [37,38]. The six-layer model has been recently adopted by major cloud service providers who offer an extensive catalog of services for programming IoT-based systems and by leading standardization initiatives in the IoT domain, such as the IoT Reference Architecture defined by ISO [39].

Figure 1 illustrates the six-layer architecture designed for the development of collaborative exer-games. Two types of actors are involved in these games: the players and the game manager. Players wear various devices (such as smart glasses, physiological monitors, or motion sensors) and must complete a series of physical tasks to reach the final goal of the game. Due to the collaborative nature of these games, the milestones achieved by one player may influence the progress of tasks for other players. The game manager oversees the overall progress of the game and can optionally introduce new challenges that modify the game's control and the tasks players must complete.

The bottom layer of the architecture is composed of the Physical devices integrated into the controlled environment. In this work, particular focus is placed on virtual reality glasses, devices for monitoring players' physiological signals, and equipment used to deliver music as a stimulus capable of inducing changes in the player during the game. The Device management layer maintains a registry of the specific devices available in the environment and is responsible for ensuring the security requirements related to those devices.

Devices generate various types of events related to players' actions, their physical and emotional responses during activities, the progress of the game, and more. These events must be transmitted to the functional components in the upper architectural layers, for example, to be interpreted for updating game conditions. Similarly, devices can receive response events from those components to modify game activities or adjust environmental conditions. The Messaging layer is responsible for managing this flow of events between devices and components. In this architecture, this layer serves as a transversal element, facilitating the integration of different distributed device environments.





**Figure 1.** High-level architecture of a collaborative game.

The functional core of the games is organized into three layers. The Ingestion layer filters and processes events received from devices to generate relevant knowledge for game control and player states. This knowledge is stored in data repositories, which are accessible to the other functional components in the solution. Some of these components provide general-purpose functionality, while others offer specific support for the development of exer-games; both are encapsulated as services. The Service layer integrates all these network-accessible components, which can be reused in the development of various games. Finally, the Application layer contains the applications that configure the games and manage the overall control of each game round. These actions are facilitated by integrating services from the lower layers and leveraging the knowledge stored in the repositories.

#### 4. Integrating Music into Exer-Games

As discussed above, the focus is on integrating music and emotions into the programming of exer-games. A technological framework based on the architecture presented in Figure 1 is developed to support this integration. The functionality of this framework can be reused to program various games designed in accordance with the described game model.

In this section, the requirements for using music and emotions to enhance exer-games are first detailed. Then, the technological framework developed to meet these requirements is presented.

##### 4.1. Problem Description and Requirements

When a participant is playing, the sensors in their environment acquire data that describe their actions and physiological responses to physical activity. These data are processed by the game application to modify the conditions of the current activity or to determine the next activities the player will complete. As part of these decisions, the application also selects the music to be played during activities to influence the player's emotions. Typically, this music is intended to increase motivation and thereby improve the outcomes of the physical tasks the player is performing. However, it could also be used for other purposes, such as creating distractions that lead to overexertion.

In collaborative games, the actions and achievements of one player may influence other participants. This principle also applies to music: the application can select or modify the music that a player hears based on the performance of other players. For example, it may increase the motivation of players who are lagging behind in their activities or create a distracting atmosphere for those who are ahead. These decisions are more complex than those affecting individual players, but both types of decisions share the same technological requirements.

The requirements related to emotion-based music in exer-games are as follows (these are listed to facilitate reference in subsequent sections):

- R1: Recognize players' emotions during physical activities.
- R2: Recognize the emotions that songs are likely to evoke in listeners in order to create a catalog of songs labeled from an emotional perspective.
- R3: Make music recommendations based on emotional criteria.
- R4: Personalize emotion-based music recommendations for each player, considering their tastes and musical preferences.
- R5: Integrate decision mechanisms that use music as a stimulus to influence game progress.

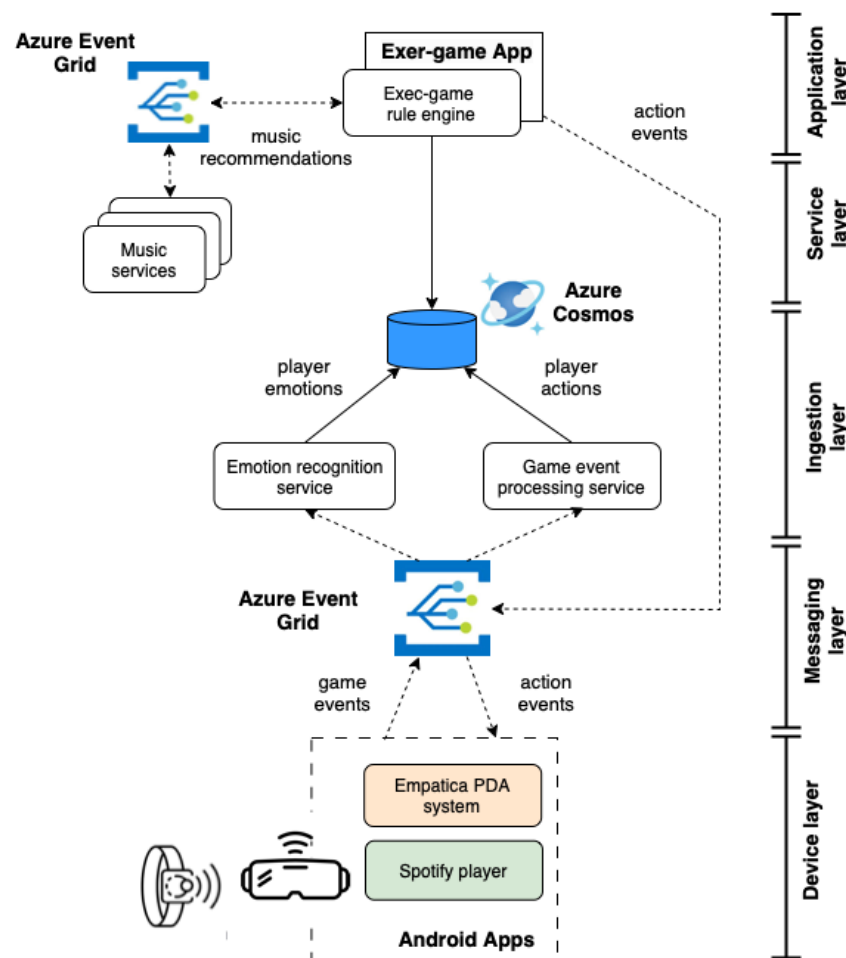
These requirements are translated into a set of software components that work in conjunction with sensors and devices available in players' environments to provide the necessary functionality. These components are organized and orchestrated based on the architectural model presented above. Additionally, cloud computing principles are adopted to address the technical and integration challenges of the proposed solution.

#### 4.2. Solution Design

Figure 2 illustrates the devices and software components that constitute the emotion-based music system for exer-games. The right-hand side of the figure outlines the relationship between the six-layer architecture and the system elements. The connectivity between elements across different architectural layers is primarily implemented using an event-driven interaction model. This model enhances the decoupling of distributed components, improves system scalability, and increases fault tolerance.

The player wears two devices: an Empatica E4 physiological wearable [40] and Meta Quest 3 smart glasses [41]. The former allows real-time monitoring of the player's physiological signals (such as heart rate, electrodermal activity, blood volume, and temperature), which are then used to recognize the emotions the player is experiencing. The glasses display game elements and manage the player's interaction with these virtual components. Additionally, the glasses play a crucial role in game control for two reasons: (1) they generate events describing relevant conditions about game progress (e.g., player actions, challenges overcome, activities completed) as well as events concerning the player's physiological response during the game, and (2) they receive events specifying changes in game conditions, environment configurations, and stimuli aimed at influencing player behavior. Some of these input/output events are programmed directly using the glasses' core utilities, while others require specific applications executed on the glasses.

As part of this work, two applications are developed. The first, the Empatica Physiological Data Acquisition (PDA) system, is an Android application capable of communicating with the Empatica E4 wearable to remotely acquire and filter the player's physiological data and generate corresponding events. These events primarily contain measurements of physiological signals over time. This application is built with the Android 12.0 version, API level 31. The second application, the *Spotify Player*, is another Android application that plays songs from the *Spotify* discography. This application is responsible for executing musical stimuli through the audio of the glasses.



**Figure 2.** High-level design of the solution.

These devices and applications comprise the Device layer of the solution. The flow of events between these elements and the other components of the music system is managed by the Messaging layer, which essentially functions as an event bus. In this work, the Azure Event Grid service is chosen as the integration bus [42].

Data ingestion services subscribe to the event bus to receive events from the glasses. These services specialize in processing specific events and storing the results in data repositories, which are accessible to services in the upper layers. The Ingestion layer of the music system includes two services: the Emotion Recognition Service and the Game Event Processing Service. The Emotion Recognition Service processes events related to the player's physiological data, using machine learning models to determine the player's current emotional state based on their electrodermal activity (i.e., translating physiological events into emotions) [43]. Many physiological events are generated for each player during a match, allowing for the progressive computation of their emotional sequence, which helps define the player's mood and emotional changes over time. The Game Event Processing Service, on the other hand, analyzes the player's behavior patterns during the match, translating game progress events into behavior patterns to identify each player's achievements and challenges.

The emotional and behavioral data processed by these ingestion services are stored in a COSMOS database [44], which is accessible to the games that operate within the Application layer. Game applications use these data to execute rules that control gameplay. Internally, an Event–Condition–Action (ECA) rule engine is used as follows: certain behavior patterns (events) trigger changes in the player's activities and/or the generation of musical stimuli to influence those activities (actions), while these actions may also be conditioned by the



player's emotions (conditions). These actions are then converted into events, which are sent to one or more players' glasses through the messaging layer's event bus.

A musical stimulus consists of one or more songs selected to evoke a specific emotion in the listener. The chosen songs depend on the player's current emotional state and the emotion that the system aims to induce. The game engine interacts with a music recommendation system that takes both emotional perspectives into account when determining stimuli. Additionally, the recommendation system includes personalization features that enhance musical decisions based on the listener's preferences. In this proposal, the recommendation and personalization systems are developed to work with *Spotify*, the most popular music streaming provider. The *Spotify* discography (containing over 100 million songs) was previously classified from an emotional perspective using the RIADA system [45], a distributed infrastructure developed by the authors to recognize the emotions conveyed by *Spotify* songs. All these systems are encapsulated as services and integrated into a framework within the Service layer. The framework is based on Azure technology and publishes its functional interfaces through the event bus.

Finally, the requirements outlined in Section 4.1 are mapped to the software elements of the proposed solution: R1 is addressed by the Empatica Physiological Data Acquisition system (Device layer) and the Emotion Recognition Service (Ingestion layer); R2, R3, and R4 are handled by the music services framework (Service layer); and R5 is managed by the game engine (Application layer).

#### 4.3. Example of Emotion-Based Musical Stimulation

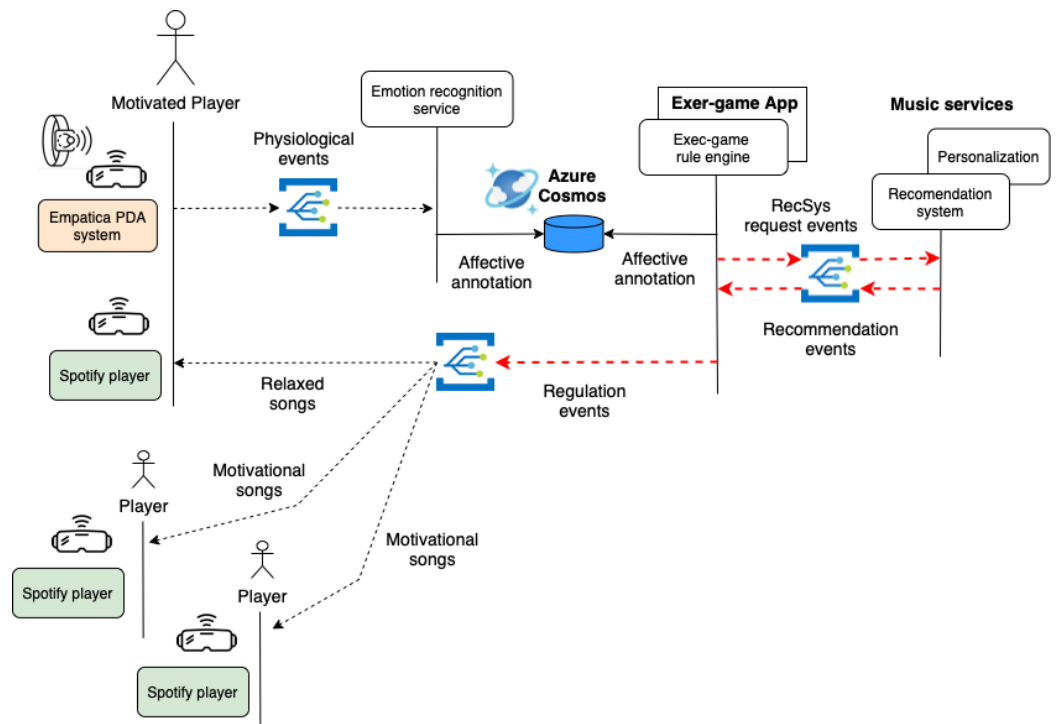
In this section, the interaction between the components shown in Figure 2 is illustrated through an specific example of stimulation scenario. This interaction consists of a flow of messages, some of which contain information about the players' emotions and the songs to be played as part of the affective regulation actions. So that, before explaining the flow, the affective model to represent emotions and the way of identifying songs are briefly introduced.

Various models to represent emotions have been proposed in the field of affective computing. The most popular model is Russell's circumplex model [46]. It represents affective states over a two-dimensional space defined by the valence (X-axis) and arousal (Y-axis) dimensions. The combination of these two dimensions (valence/arousal) determines four distinct quadrants: the aggressive (negative/positive), the happy (positive/positive), the sad (negative/negative), and the relaxed (positive/negative) quadrants. Each emotion is then mapped to a point within this two-dimensional space and is thus located in one of these quadrants. Considering the Russell's model, we decided to represent an emotion by means of two vectors of four values: the first vector determines the probability that the emotion is mapped to each of the four quadrants, and the second which of these probabilities are significant (a relevance threshold was experimentally calculated). Two examples of emotions are as follows:

- The annotation ([0.174, 0.865, 0.155, 0.006], [false, true, false, false]) represents that the affective state is happy with a 0.865 probability. The aggressive, sad, and relaxed probabilities (0.174, 0.155 and 0.006, respectively) are lower than the threshold and therefore the state is also defined as not sad, not aggressive and not relaxed. This annotation could correspond with a positive emotion with a high arousal value, for example, with the emotion "excited".
- The annotation ([0.087, 0.695, 0.213, 0.578], [false, true, false, true]) represents that the affective state is happy and relaxed (and not aggressive and not sad). It could correspond with a positive emotion closer to the X-axis than the one of the previous example, for example, with the emotion "delighted".

Regarding the identification of songs, the *Spotify IDs* of tracks are reused. This decision facilitates the use of the *Spotify* player, which is integrated into the smart glasses, and the interaction with the *Spotify* online services to provide the music-based functionality.

After presenting the representation models, the stimulation scenario is introduced. We suppose that a player is particularly motivated and is successfully completing the physical tasks (player’s predominant emotion would be “excited” at that moment). The other players are performing their activities more slowly and are lagging behind the motivated player (they would feel “relaxed”). The emotion-based music system wants to regulate the players’ affective states to balance their task outcomes. Figure 3 synthesizes the flow of messages that would happen in this scenario.



**Figure 3.** Interactions to stimulate multiple players.

The motivated player wears an Empatica E4 device which monitors their physiological signals. The Empatica PDA system, integrated into the player’s glasses, periodically accesses these raw signals and filters the information related to the electrodermal activity (EDA). Then, EDA data are packaged in an event message to be published in the bus. Each package contains the information extracted from a five-minutes signal fragment. The Emotion Recognition Service is subscribed to be notified when new physiological events are available. It applies a series of machine learning models capable of translating the received EDA data to an emotion. This emotion represents the predominant affective state that the player is most likely to have felt during those five minutes. In this scenario, the affective annotation would be like the one in the first example to determine that the player feels “excited” (an emotion that corresponds with high-motivation states). That annotation is stored in the COSMOS database as part of the mood of the player during the gameplay.

The Exer-game engine has the following rule: (EVENT: “When a player is performing the physical tasks much better and faster than the opponents”; CONDITION: “That player is feeling a positive emotion with a high arousal value”; ACTION: “Send a relaxing stimulus to the player and motivating stimuli to the rest of players”). The rule is activated when that game event is generated by the Game Even Processing service. This high-level event is a composite event, formed from game events published by the players’ glasses. The engine queries in the database the player’s latter emotion (“excited”, a positive emotion) to evaluate the rule condition and, since it is fulfilled, it must generate the corresponding actions. These actions consists of personalized stimuli based on the emotions to be evoked.

The Game application knows the identity of the players which is needed to personalize the music stimuli. In Figure 3, the red connectors represent the sending and/or receiving

of a set of events to/from the bus. In accordance to this representation, the application publishes a recommendation request event in the bus per each of these participants. A recommendation event mainly contains the identity of player and the emotion to be evoked by the recommended songs: in this case, “relaxed” for the motivated player and “enthusiastic” for the rest. An Emotion-based music recommendation service receives these events and searches a set of candidate songs available in the *Spotify* discography capable of evoking the requested emotion. These songs are then ranked and filtered by a Personalization service to customize the response to the music preferences of each player. The identity of the player determines the personalization profile to be applied in each recommendation event. The algorithms of emotion-based recommendation and personalization are detailed in Section 5. The result is a response event for each recommendation request, containing a list of suggested songs, more specifically a list of *Spotify IDs* tracks.

Finally, the application publishes the events of recommended songs in the bus. The glasses of each player are notified and recovery the corresponding event, and the list of tracks is locally played through the *Spotify player*. From a moment on, the physiological data of each player should be used to evaluate whether the music-based regulation actions have effect: the motivated player reduces the performance intensity and the rest of players increases their motivation and the performance of physical tasks.

### 5. Music Recommendation and Personalization Based on Emotions

This section explores the integration of emotional insights into music recommendation systems, focusing on the methodologies used to recognize emotional content in music based on audio features and machine learning models, and how these insights are adapted to user emotions for personalized recommendations. The discussion begins with an examination of the architectural framework for implementing emotion-aware music functionalities. Following this, the methodologies and technologies used to accurately identify and interpret the emotional content of music are presented. The focus then shifts to how user preferences and emotional responses are utilized to build comprehensive profiles that enhance recommendation accuracy. Finally, the section concludes with an analysis of techniques for tailoring music suggestions to individual users’ emotional states and preferences.

#### 5.1. Function-Based Design of the Music Services

Figure 4 illustrates the Music services developed to support the generation of musical stimuli. These services offer the functionalities required to achieve requirements R2 (Section 5.2), R3 (Section 5.4), and R4 (Section 5.3) described in Section 4.1. These functionalities are accessible through an event bus, as shown on the right side of the figure. Essentially, two types of service requests can be published: registering a new user and obtaining music recommendations based on emotions.

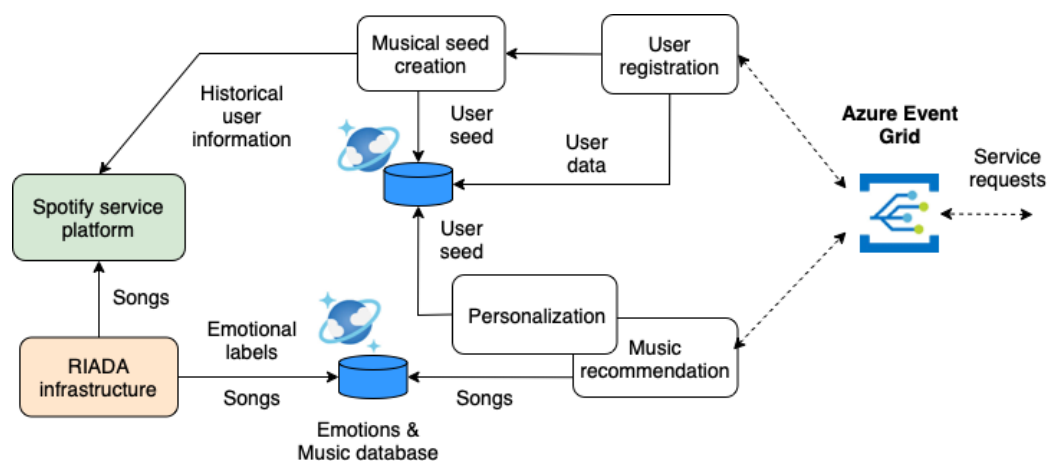


Figure 4. Services involved in the generation of music stimuli.

User registration is required to provide personalized recommendations. When a new user is registered, a musical seed that describes their tastes and preferences is generated. This seed is derived from the user's *Spotify* playlists and recently played songs on the streaming platform. Access to this information requires that the user holds a premium license and provides the necessary credentials. Once access is granted, the seed is generated automatically without user intervention. The user's registration data and musical seed are then stored in a repository. The two services responsible for these operations are the User Registration and Musical Seed Creation services. The latter interacts with the *Spotify* developer platform [47] to compute the user's seed.

Game applications request recommended songs to generate musical stimuli. A recommendation request must include at least the listener's identity (for personalization purposes) and the emotion to be evoked. The Recommendation service is responsible for identifying candidate songs that best match the requested emotion. As part of this process, a Personalization service uses the listener's seed to filter the candidate songs based on their preferences, ensuring a customized result. A repository of emotionally labeled songs supports these emotion-based recommendations. As described earlier and shown on the left side of Figure 4, the RIADA system employs a Random Forest machine learning model to classify songs based on their emotional content, using a set of audio features provided by *Spotify* (e.g., Valence, Energy, Tempo, Acousticness, Danceability). This system labels each song into one of four emotional quadrants—happy, sad, relaxed, and angry—which are then stored in the song and label repository. These labels serve as the basis for generating music recommendations aligned with the user's emotional state.

All music services (represented as white rounded rectangles) are implemented as Azure Durable Functions [48], a type of serverless solution that reduces the costs associated with programming and executing in cloud environments. Azure Durable Functions extend the capabilities of standard Azure Functions by enabling the creation of stateful workflows. They are designed to manage and coordinate complex, long-running business processes and stateful operations without requiring developers to manage the underlying infrastructure.

Durable functions are particularly useful in our scenario, as our problem workflows involve long-running processes as music tag processing, music recommendation, human interactions, or multiple steps that need to be executed in sequence. These functions provide a way to build reliable and scalable applications while abstracting away the complexity of state management.

In this work, three types of durable functions in Azure Functions are used:

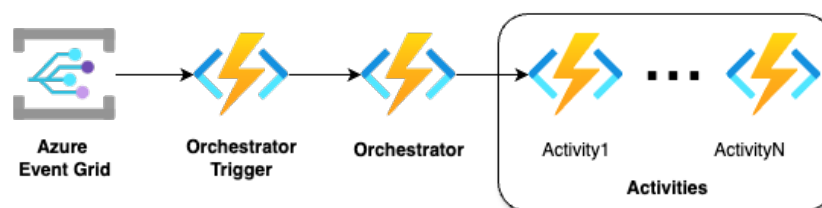
- Activity functions: These are the building blocks of durable workflows, responsible for performing tasks or operations. They are called by orchestrator functions and can be executed in parallel or sequentially.
- Orchestrator functions: These functions define the workflow of the durable application. They manage the coordination and state of activities and control the flow of execution. Orchestrator functions are durable and can handle long-running processes by maintaining their state across restarts.
- Client functions: These functions are responsible for starting and interacting with orchestrator functions. They provide an entry point for initiating durable workflows and can be used to pass inputs and retrieve results from orchestrator functions.

The architecture described is designed to integrate Azure Durable Function components to handle and manage complex workflows efficiently. The process is triggered through the Azure Event Grid, which provides a unified event routing mechanism that can handle events from multiple sources. The triggers for the process can be categorized into three main types: HTTP Trigger, Event Trigger, and Timer Trigger.

- HTTP trigger: This trigger type serves as an endpoint for clients to initiate the process. When a client sends an HTTP request to the designated endpoint, an HTTP-triggered function is activated. This function acts as a gateway to start the orchestration workflow.

- **Event trigger:** This trigger type is used to initiate the process based on events launched from various services. For example, an event generated by a service like Azure Service Bus or Azure Blob Storage can activate the function, which then starts the orchestration process.
- **Timer trigger:** This trigger is employed for scheduled executions of the process. A Timer-triggered function activates based on a specified schedule, such as daily or weekly intervals, thus enabling the orchestration process to run periodically.

Figure 5 depicts the generic flow described. Upon activation by one of the aforementioned triggers, the process is captured by a function designated as an orchestration trigger. This function is responsible for initiating the orchestration of the workflow. Specifically, it starts an orchestrator function, which manages the workflow and coordinates the execution of various tasks.



**Figure 5.** Generic flow of Azure Durable Function's execution in our proposal.

The orchestrator function, once initiated, executes a series of activity functions. These activity functions represent the individual tasks or operations that need to be performed as part of the workflow. The sequence of activity functions, ranging from Activity 1 to Activity N, is executed as defined by the orchestrator. Each activity function performs a specific operation that contributes to fulfilling the overall functionality required by the process.

The orchestrator function ensures that the activities are executed in the correct order and handles the state management necessary for long-running processes. It maintains the state of the workflow, allowing for the management of complex, multi-step processes and providing resilience against failures and restarts.

Overall, this architecture leverages Azure Durable Functions to create a robust, scalable solution for managing and executing stateful workflows, with triggers facilitating the initiation of processes and orchestrators coordinating the execution of activities.

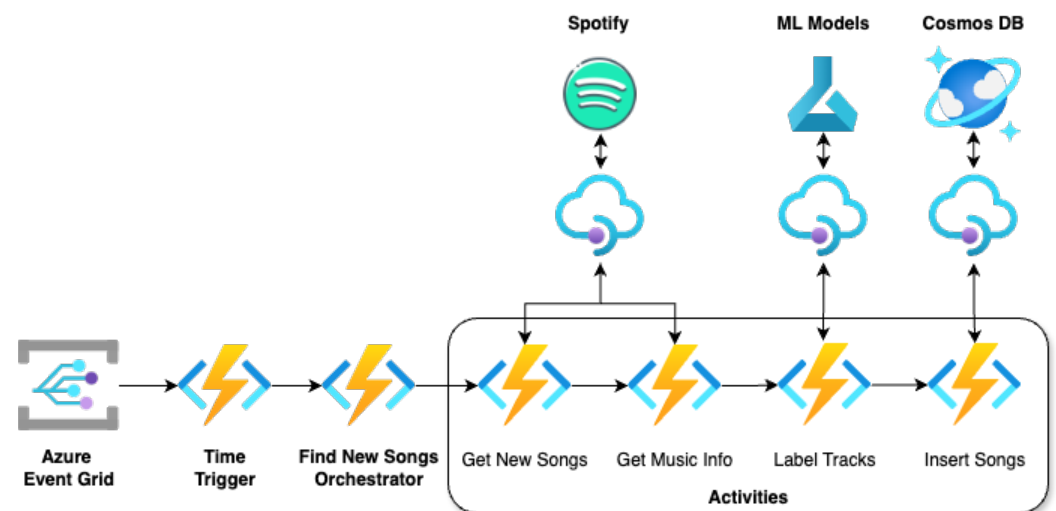
### 5.2. Music Emotion's Recognition

The emotion recognition process implemented in this study is based on the models introduced in the *RIADA* system [45], designed to tag songs from an emotional perspective. The input for these models consists of various audio features provided by *Spotify*, including Valence, Energy, Acousticness, Danceability, Instrumentalness, Loudness, Duration, Speechiness, Tempo, Key, Mode, and Liveness.

The output of the emotion recognition model is a pair of values: a binary value and a continuous value that predict whether the emotions perceived by listeners fall within the corresponding quadrant. There is a separate model for each quadrant.

Figure 6 illustrates the flow of the Azure Durable Function responsible for finding and labeling new songs. The function is triggered by an orchestrator, which sequentially executes tasks to incorporate new data into the database. The first task, *Get New Songs*, retrieves a list of *Spotify* track identifiers by sending requests for New Releases, retrieving up to 100 songs from key international markets. The following task, *Get MusicInfo*, extracts general song information (title, artist, etc.) as well as audio features from *Spotify*, sending *Several Tracks and Audio Features* requests via the *Spotipy* library.





**Figure 6.** Flow of Azure Durable Function for finding and labeling new songs.

Once the data are gathered, the Label Tracks task applies the pre-trained Random Forest models, loaded using joblib, to assign the four different emotional labels to each song based on the extracted audio features. Finally, the Insert Songs task stores the songs' general information, features, and emotional labels in the COSMOS database.

This function is designed to periodically update the COSMOS database with newly released songs from *Spotify*, ensuring that the database remains current with the most recent emotionally labeled music. The process is triggered weekly using an Azure timer. Two of the activities interact with *Spotify* web services, represented by the *Spotify* logo, while another activity accesses Random Forest pre-trained machine learning models, symbolized by a blue test tube. The final activity interacts with the COSMOS database.

Let us now depict a more detailed technical level of the process. Before feature selection, normalization was applied to ensure that the values of all features fall within the range of zero to one. This was achieved using the MinMaxScaler from Scikit-learn version 1.5 [49], and the transformation process was saved to ensure it could be applied to new data during model deployment.

Feature selection was performed through a voting system based on three metrics: Chi-Square, ANOVA F-value, and Mutual Information (all available in Scikit-learn [50]). Features were ranked according to their relevance, where 1 indicated the least important and 12 the most important. The feature with the highest cumulative rank across the three voting systems was considered the most relevant, while the feature with the lowest score was the least. This ranking process was conducted independently for each of the four emotional quadrants. The following feature combinations, ordered from least to most important, yielded the best results for each emotion:

- Sad: danceability, key, speechiness, mode, instrumentalness, tempo, duration, liveness, loudness, valence, acousticness, energy;
- Happy: danceability, key, speechiness, mode, instrumentalness, tempo, duration, liveness, loudness, valence, acousticness, energy;
- Angry: key, valence, mode, duration, instrumentalness, tempo, danceability, liveness, speechiness, loudness, energy, acousticness;
- Relaxed: key, tempo, liveness, mode, duration, speechiness, danceability, valence, acousticness, loudness, energy, instrumentalness.

Then, a recognition model was built for each of Russell's affective quadrants. Different types of machine learning algorithms were considered for their building: Random Forest (RF), K-Nearest Neighbor (KNN), Support Vector Machines (SVM), Gradient Boosting (GB), and Multi-Layer Perceptron (MLP). The first three were built and analyzed during the development of the RIADA [45], while the last two were created as part of this work. Table 1

shows the results of the different combinations of algorithms and affective quadrants. All those models were trained using repeated five-fold cross-validation with a 70/30 data split and a randomized hyperparameter search. The best combinations are highlighted in green color. As conclusions, Random Forest models offer good accuracy results for the four quadrants and, for this reason, they were selected to be integrated in the process of music emotion recognition.

Different causes influence whether the Random Forest models offer those promising recognition results. In general, it has proven to be a model highly flexible and robust. The input dataset was slightly imbalanced (i.e., the number of songs of each of four affective quadrants is different) and it consisted of categorical and continuous features. Random Forest is able to handle more effectively the impact of these two characteristics in the results than the other alternative learning algorithms. With respect to the size of the dataset classes, the use of voting applied to decision trees trained from different data subsets reduces the effects of imbalance. In addition, the decision-making based on trees instead of distance measurement techniques facilitates the handling of the categorical features included in our dataset. On the other hand, the size of the input dataset could lead to overfitting of the resulting models. Although we applied cross-validation to reduce the risk of that overfitting, Random Forest is usually less sensitive to it than the other models.

**Table 1.** Results of the different models.

Algorithm	Tests	Happy	Angry	Sad	Relaxed
Random forest	accuracy	0.844	0.899	0.862	0.945
	f1	0.820	0.860	0.839	0.801
SVM	accuracy	0.767	0.872	0.8036	0.929
	f1	0.752	0.821	0.783	0.733
KNN	accuracy	0.843	0.876	0.842	0.935
	f1	0.822	0.824	0.816	0.784
GB	accuracy	0.844	0.892	0.862	0.933
	f1	0.823	0.850	0.839	0.773
MLP	accuracy	0.836	0.879	0.844	0.938
	f1	0.815	0.828	0.815	0.779

The following analysis of hyperparameters was carried out for the Random Forest models using RandomizedSearchCV from Scikit-learn (similar work was performed for the other models):

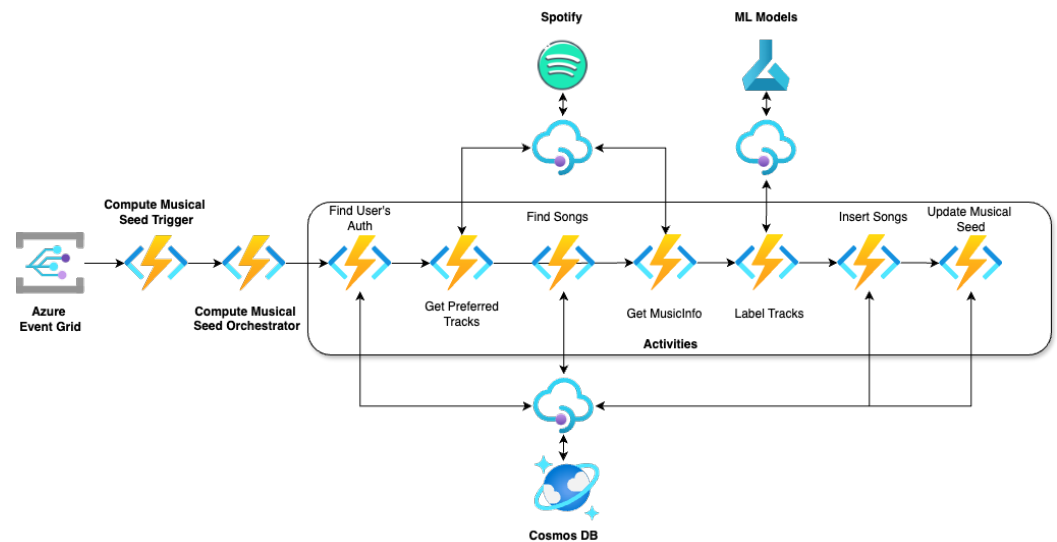
- Number of estimators: 10, 35, 60, 85, 110, 135, 160, 185, 210
- Criterion: gini or entropy
- Minimum samples per leaf: 2, 7, 12
- Minimum samples per split: 2, 5, 10
- Maximum depth of the tree: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
- Bootstrap: true or false

The selected hyperparameters for each of four Random Forest model were as follows:

- Sad: bootstrap = False, criterion = entropy, maximum tree depth = 30, minimum samples per leaf = 2, minimum samples per split = 5, number of estimators = 160;
- Happy: bootstrap = False, criterion = entropy, maximum tree depth = 30, minimum samples per leaf = 2, minimum samples per split = 5, number of estimators = 160;
- Angry: bootstrap = False, criterion = gini, maximum tree depth = 90, minimum samples per leaf = 2, minimum samples per split = 2, number of estimators = 185;
- Relaxed: bootstrap = False, criterion = gini, maximum tree depth = 90, minimum samples per leaf = 2, minimum samples per split = 2, number of estimators = 185.

### 5.3. Generation of a User Profile for Personalization

The personalization system is based on data services provided by *Spotify*. Functionally, the system uses the musical seed which represents the user's musical preferences and tastes. This seed, combined with the user's requested emotion, is then used to select the appropriate songs to return. Figure 7 illustrates the system's functionalities which are divided into distinct activities, each managed by an orchestrator and triggered via an HTTP Trigger accessible to the user.



**Figure 7.** Flow of Azure Durable Function for computing the personalization musical seed.

A *Spotify* premium account and an authorization token are required to access specific *Spotify* API requests and retrieve user data. Depending on the scope of the data, different permissions are necessary. In this case, the following permissions are utilized: *user-library-read*, *user-top-read*, *playlist-read-private*, *user-read-recently-played*, and *playlist-read-collaborative*. These scopes must be set prior to requesting the user's authorization, and this step is handled locally on the device using the Authorization Code with PKCE Flow, which is recommended for this environment. Once the user is registered in the application and grants the necessary permissions, the authentication details are stored in the COSMOS database.

The process begins when the Compute musical seed trigger is activated by an HTTP GET request. This trigger retrieves the user ID from the JSON body of the request and initiates the orchestrator with the recovered JSON as a parameter. It waits for the orchestrator to return the response to the client.

The Compute musical seed orchestrator is responsible for coordinating the execution of all subsequent activities. The first activity, Find user's auth activity, retrieves the user's authentication details from the COSMOS database. The user must have been previously registered, and their authentication token must have been requested.

Once the user's authentication is confirmed, the Get preferred tracks activity retrieves the identifiers of the songs that best match the user's preferences. This is accomplished through several *Spotify* API requests:

- Get Saved Tracks retrieves the user's saved songs.
- Get Recently Played Tracks returns the user's recently played songs.
- Get User's Top Items lists the top tracks identified by *Spotify* as most relevant to the user.
- Get User's Playlists, combined with Get Playlist Items, retrieves the user's playlists and the songs within each playlist.

The output of these API requests is a list of *Spotify* identifiers representing the user's favorite songs.

The next activity, Find songs activity, checks the COSMOS database to find songs that are already stored. It returns the stored songs, including their audio features and emotional labels, in JSON format.

For songs not already stored in the database, the process continues with the following activities: Get Music Info, Label Tracks, and Insert Songs, which are described in Section 5.2.

Finally, the Update Musical Seed activity calculates the average of the musical features provided by *Spotify*, excluding key and mode due to their discrete nature as well as duration as it is less relevant to user preferences. If a song appears multiple times across the different *Spotify* requests, its features carry more weight in the seed calculation, emphasizing the importance of songs frequently listened to or saved by the user. The resulting musical seed is then updated in the COSMOS database.

#### 5.4. Music Recommendation

The music recommendation service is built upon personalization profiles. As depicted in Figure 8, when a user identifier and a desired emotion are provided, the request is triggered by the client.

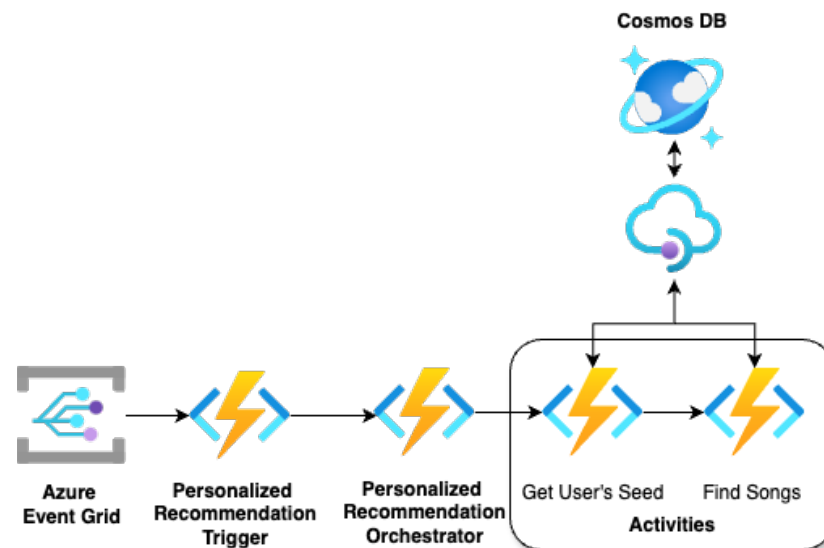


Figure 8. Flow of Azure Durable Function for recommending personalized songs.

The Get Personalized Recommendation Trigger is activated through an HTTP GET request. This trigger retrieves the user ID and the desired emotion label from the JSON body of the request and initiates the orchestrator with these data.

The Get Personalized Recommendation Orchestrator is responsible for coordinating the execution of all subsequent activities. The first activity, Get User's Seed Activity, retrieves the user's musical seed from the COSMOS database using the provided user ID. This seed is calculated as described previously, representing the user's preferences based on their listening habits.

Once the seed is retrieved, the Find Songs Activity searches for songs whose audio features fall within a 20% tolerance range of the user's musical seed, the selected *Spotify* audio features are normalized between 0 and 1, except for loudness, which ranges from  $-60$  to 0 and requires an adjustment. This range-based search was selected due to its efficiency through index creation in the database, with a worst case scenario in time complexity of  $O(\log(n))$ . Other recommendation systems [51] often employ similarity functions such as Euclidean Distance, Cosine Similarity, or Manhattan Distance, which are more precise and also more computationally expensive with a time complexity of  $O(n)$ . From our perspective,

a faster approach, despite being less precise, is preferred, as the result group of similar songs is randomized to introduce variability in the music recommendation. These songs are selected based on their similarity in audio features, ensuring that they align with the user's requested emotion. To achieve this, a range of values is defined for the emotional labels, maximizing the likelihood that the selected songs belong to the requested emotional quadrant while minimizing subjective user interpretations. The defined ranges for each emotion are as follows:

- Angry: Angry > 0.6, Sad < 0.2, Relaxed < 0.2, Happy < 0.4;
- Sad: Angry < 0.2, Sad > 0.8, Relaxed < 0.55, Happy < 0.2;
- Relaxed: Angry < 0.2, Sad < 0.6, Relaxed > 0.85, Happy < 0.2;
- Happy: Angry < 0.3, Sad < 0.2, Relaxed < 0.2, Happy > 0.6.

The system selects a default of five songs, provided enough songs meet the emotional criteria based on their labels. These labels are generated using the Random Forest classifier trained on *Spotify's* audio features, ensuring that the selected songs align with the emotional quadrant requested (happy, sad, relaxed, or angry). Songs are selected randomly from this pool to ensure diversity in the recommendations and avoid monotony.

Once the songs are selected, they are returned to the client as a personalized recommendation, providing a tailored experience based on the user's musical seed and emotional preferences.

### 5.5. IoT Principles and Computational Framework

In terms of the IoT principles underlying the proposed system, the architecture leverages a distributed model, where the computational workload is divided between edge devices (wearables) and cloud-based services. This approach ensures that low-latency operations, such as emotion recognition, are processed locally on the wearables, while more computationally intensive tasks, including music recommendation and personalized stimuli generation, are handled by cloud services like Azure Durable Functions. By distributing the workload, the system minimizes delays during real-time interactions, crucial for maintaining an uninterrupted gaming experience. Additionally, the system uses MQTT (Message Queuing Telemetry Transport) as the primary network protocol due to its lightweight nature and low bandwidth consumption, ideal for IoT ecosystems with constrained devices.

The neural network algorithm utilized for emotion recognition is a Random Forest model, trained using audio features provided by the *Spotify* API. This model was selected for its robustness and ability to handle non-linear relationships between input features and output labels, which is essential when working with complex emotional states. The network architecture used in this work was chosen after evaluating several alternatives, including Support Vector Machines and Gradient Boosting classifiers, as discussed in [45]. Random Forest provided the best balance between accuracy and computational efficiency, particularly in scenarios involving limited real-time processing power on edge devices. Furthermore, the computational complexity of the entire system is optimized through the use of serverless computing (Azure Durable Functions), where the pay-per-use model minimizes resource waste and scales automatically with demand, thus reducing the overall cost and complexity of managing a cloud-based infrastructure.

### 5.6. Experimentation

Two experiments with real users were conducted to corroborate the affective annotations of songs recommended by the programmed system and the musical seed used for the personalization of songs, respectively. We declare that these experiments are not a formal and thorough study of the emotion-based recommendation functionality provided by the system previously described. The scope of the paper focuses on presenting the technological issues of the proposal and therefore, in this section, we are simply interested in carrying out a preliminary analysis about the suitability of algorithms included in the solution.



With respect to the participants in the experiments, a total of 12 users with *Spotify* premium subscriptions participated in both experiments. The participants were aged between 20 and 30 years, 9 of them male and 3 female. Their musical preferences were varied, but their taste for the contemporary music predominated. Nevertheless, these preferences were not considered as part of these experiments.

#### 5.6.1. Perceived Emotions by the Listeners

The goal of the first experiment is to corroborate the affective annotations of songs with the emotions that users feel when listening to them. In the design of the experiment, we used a repository of *Spotify* songs that were annotated with the RIADA emotion recognition models as a part of a previous research project.

**Instruments:** On the one hand, a playlist was created from a repository of songs. It consists of 16 *Spotify* songs, 4 songs from each of Russell’s quadrants. The main criteria for selecting these songs was that their affective annotations were characterized by having one dominant emotion, i.e., songs that could evoke more than one emotion with high probability were excluded. The preferences or tastes of the participants in the experiment were not taken into account in the selection of the songs, and the songs were randomly ordered. On the other hand, a Google survey form was programmed to gather the participants’ responses. The survey asked each participant to introduce the emotion that they felt when listening to each song. In the design of the survey, the “Pick-A-Mood” (PAM) model [52] was used. PAM is a cartoon-based pictorial instrument for representing the user’s possible emotional states based on Russell’s affective model. More specifically, PAM expresses nine emotional states, two for each of the four quadrants and a neutral state. This visual representation reduces the time and efforts of the respondents, which makes the PAM model suitable for the design of these state-based emotions.

**Methodology:** The participants were in a relaxed environment and wearing headphones to listen to the playlist. The experiment consisted of playing each song and asking the participant what they felt while listening to it. Between songs, the listener had enough time to introduce the answer on the online survey. The duration of the experiment was approximately 50 min.

**Results:** Table 2 presents the main outcomes of this experiment. The rows represent the affective annotations of the songs included in the playlist and the columns show the listeners’ responses. For example, the component [Sad, Happy] represents the percentage of responses (6.25%) in which a listener declared to feel happy when listening to a sad song. Therefore, the diagonal of the table indicates the percentage of responses that matched the affective annotations of the songs. Overall, a high percentage of aggressive, happy, and relaxed songs were correctly recognized by the listeners (72.92%, 77.08%, and 89.58%, respectively). However, the percentage of sad songs recognized correctly was lower (64.58%), as many were identified as relaxed (both emotional categories share a similar valence).

**Table 2.** Affective annotations versus emotions perceived by the listeners.

Annotation/Responses	Aggressive	Happy	Sad	Relaxed
Aggressive	72.92	20.83	6.25	0
Happy	10.42	77.08	10.42	2.08
Sad	6.25	6.25	64.58	22.92
Relaxed	2.08	2.08	6.25	89.58

**Conclusions and improvements:** The number of participants in the experiment is small, and as such, the results should be interpreted with caution. Nevertheless, these preliminary results are promising and suggest that the accuracy of the affective annotations is good with respect to the emotions felt by the listeners. We acknowledge that the duration and conditions of the experiment limit its scalability. The participants must be relaxed and concentrated in the listening, avoiding interruptions or early termination of the activity. As a future

improvement, an alternative method for making a large-scale validation of annotations should be designed. Ideally, this validation should be automated, which requires a database of songs emotionally labelled by users. These labels should have similar semantics to those used in this work in order to ensure the validity of results.

#### 5.6.2. Recommendations Based on the Listener's Seed

The second experiment aimed to determine whether the musical seed computed for a user improves emotion-based music recommendations. As was described, this seed is based on the user's listening habits.

**Instruments:** A web application was programmed as part of this experiment. When a participant connects to the application, they must first introduce the information of their *Spotify* premium subscription. Then, the application uses this information to interact with the music recommendation service and to obtain a list of songs to be played. The list contains songs that match the participant's musical preferences and others that are randomly selected. The application plays the songs in a random order and requests the participant that rates how well each song matches their preferences. A rating of up to five stars could be given for each song, with one meaning the recommendation was not liked at all and five meaning the recommendation was spot on.

**Methodology:** The application was configured to play 48 songs, with 12 songs from each of the four affective quadrants (i.e., 12 songs annotated as happy, 12 as relaxed, and so on). Of each of these 12 songs, 9 were selected according to the listener's seed and 3 randomly. The participants could listen to each song in its entirety or stop the playing and rate its matching their preferences. The duration of the experiment was more than 2 h. For this reason, we offered the participants the option of conducting the experiment at home and in several stages.

**Results:** The personalized songs received an average score of 3.83, indicating that listeners generally agreed with or liked the recommendations. The randomly selected songs received an average score of 3.12, which, although lower than the personalized songs, was higher than expected. This may be attributed to participants' openness to listening to new or alternative songs. From the affective perspective, the highest score for personalized songs was for those songs annotated as happy (the average rating was 4.12), while the lower for those as sad (3.34). The relaxed and aggressive songs scored very similar results, 3.88 and 3.95, respectively.

**Conclusions and improvements:** As in the first experiment, the number of participants is small and, therefore, the results should be interpreted with caution. Nevertheless, the score of personalized songs suggests that the listeners' seed is suitable to improve the music recommendations. Again, the duration of the experiment is an obstacle for its scalability. As a future alternative, a new player for *Spotify* subscribers could be programmed. It could play only recommended songs (personalized and random songs) and gather information about the users' listening behavior. The player could subsequently be published in forums for developers and users of *Spotify*-based solutions or for researchers in advances in recommendation systems, for instance. Additionally, when exer-games based on musical stimuli are programmed, we are interested in monitoring the players' responses to those stimuli and in analyzing how those responses are correlated to the personalized recommendations.

## 6. Conclusions and Future Work

Finally, the main conclusions and the future challenges are presented.

### 6.1. Conclusions

In this paper, we presented a technological solution to enhance exer-games through the integration of emotion-based music. The system is designed for IoT-based games and consists of a service-oriented infrastructure that integrates smart devices, artificial intelligence models, cloud technologies, and online *Spotify* resources. During gameplay, players' emotions are monitored and used to influence the progression of the matches

through music. Musical stimuli are generated to regulate players' affective states and induce changes in their physical performance. These stimuli are personalized to maximize the impact of music-based interventions on participants, with the emotional responses of players providing further insights into the effectiveness of these stimuli.

The paper focused on presenting those Spotify-based systems involved in the recommendation of personalized music from an affective perspective. These systems leverage the data and resources provided by that music provider to offer the functionality needed to achieve the requirements involved in the generation of stimuli: the recognition of the emotions that *Spotify* songs are likely to evoke in players (requirement R2), the recommendation based on emotions (R3), and the characterization of players' musical preferences based on their listening habits in order to improve the recommendations (R4). The integration with *Spotify* allows to have available a large-scale catalog of songs and to automate the capture of participants' musical preferences. This contributes a novel and appealing approach to the system. Note that these music-based requirements need ideally to be combined with the recognition of players' emotions during the physical activity (Requirement R1) and the configuration of stimuli (R5). The solutions proposed by the authors in [53] were adapted to the Empatica device in order to accomplish Requirement R1, while R5 has to be addressed during the programming of concrete games.

One of the key advantages of the proposed solution lies in the use of Azure Durable Functions for the orchestration of music services. This serverless, event-driven architecture reduces operational complexity and provides scalability without the need for extensive infrastructure management. By enabling stateful workflows, Azure Durable Functions allow for the execution of long-running processes such as emotion recognition and music recommendation while optimizing resource consumption. The pay-per-execution model of this serverless approach also reduces costs, making it more accessible for large-scale deployments and adaptable to varying workloads in real-time gaming environments. Additionally, the inherent fault tolerance and auto-scaling capabilities of this architecture ensure system robustness and reliability during game operation.

This proposal represents a significant advancement in the role of music and emotions in the design of exergames. To the best of our knowledge, the combination of these two elements to guide gameplay progression and regulate player performance during physical tasks has not been explored previously. Another key contribution is the real-time monitoring of players' emotions and their integration into game decision-making processes. This affective dimension enhances the effectiveness of the stimuli, enables the assessment of their real impact, and facilitates the application of personalization strategies that improve the overall gaming experience.

## 6.2. Future Work

Looking ahead, several avenues of research and development remain open. First, we are working on expanding the system's capabilities by incorporating additional devices such as cameras and movement recognition technologies, which will allow for more precise monitoring of players' physical activities and emotional responses. The integration of these devices is expected to improve the accuracy of emotion-based interventions and provide new opportunities for assessing player performance.

We are currently developing two exergames to validate the concepts presented in this paper. These games evaluate the effectiveness of emotion-based musical interventions across different user profiles. User-centered design is being applied to ensure the usability and acceptance of emotion-based systems. The seamless integration of music and the personalization of stimuli are crucial factors in enhancing user engagement. An iterative design process is being followed, focusing on user motivation through personalization and social interaction. In the future, user testing will be expanded across different demographic groups, and challenges related to the synchronization of music preferences in multiplayer environments will be addressed. The evaluation will involve collaboration with experts in psychology, physical activity, and music to refine the emotion recogni-

tion system, optimize device integration, and address challenges in personalization and real-time music synchronization.

We also aim to explore real-world application scenarios by deploying the system in various environments outside of gaming. Potential applications include rehabilitation centers, where personalized music-based interventions could support physical therapy; gyms, where emotion-based music stimuli could enhance workout routines; and wellness programs, where the system could be used to promote physical and emotional well-being. Testing the system in these settings will provide critical insights into its scalability, usability, and overall impact in practical, non-gaming contexts.

Furthermore, in terms of sustainability, future work will focus on analyzing the environmental impact of the system, particularly the carbon footprint generated by the execution of cloud services like Azure Durable Functions. As cloud computing plays an increasingly important role in the deployment of scalable systems, understanding its environmental impact is critical. This analysis will examine the energy consumption and carbon emissions associated with maintaining a continuously operating cloud-based infrastructure. Additionally, we plan to explore more sustainable alternatives, comparing the environmental footprint of Azure Durable Functions with other serverless platforms such as AWS Lambda or Google Cloud Functions. These comparisons will help identify opportunities to reduce the environmental impact of the system. Strategies such as optimizing resource usage, minimizing idle time, and selecting data centers powered by renewable energy will also be considered as part of a broader commitment to sustainability in the development of IoT-based gaming systems.

In terms of cost, the current implementation of the system leverages cloud-based services such as Azure Durable Functions, which operate on a pay-per-execution model. This serverless architecture offers scalability and flexibility, but it also incurs costs related to cloud storage, data processing, and computational resources. These costs are dependent on the volume of users and the frequency of interactions, and while they are manageable for small- to medium-scale deployments, larger implementations could require a more detailed budget analysis.

To address cost concerns and reduce the overall budget, future iterations of the system could explore the feasibility of incorporating open-source frameworks. For example, open-source serverless platforms such as OpenFaaS could be evaluated as alternatives to commercial cloud services, reducing infrastructure costs. Additionally, the use of open-source machine learning libraries such as TensorFlow or PyTorch, combined with locally hosted databases, could further minimize costs. While these open-source options may offer financial advantages, their performance and scalability need to be assessed to ensure that they meet the system's requirements for real-time processing and personalization in exergames.

**Author Contributions:** Conceptualization, P.Á., J.G.d.Q. and J.F.; Methodology, P.Á. and J.F.; Software, J.G.d.Q.; Validation, P.Á. and J.G.d.Q.; Investigation, P.Á., J.G.d.Q. and J.F.; Resources, P.Á., J.G.d.Q. and J.F.; Data curation, J.G.d.Q.; Writing—original draft preparation, P.Á., J.G.d.Q. and J.F.; Writing—review and editing, P.Á., J.G.d.Q. and J.F.; Supervision, P.Á.; Project administration, P.Á.; Funding acquisition, P.Á. and J.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been partially funded by project TED2021-130374B-C22 granted by the Spanish Ministry of Science, Innovation, and Universities, and project DisCo-T21\_23R, granted by the Aragonese Government, Spain.

**Institutional Review Board Statement:** Ethical review and approval were waived for this study due to only survey results are collected and cannot be linked to specific participants. Nevertheless, the design of the experiments adhered rigorously to the Ethics Appraisal Procedure established by the European Union Horizon 2020 Program. This adherence includes an informed consent protocol in which it is explained to the participants the conditions of experiments and the subsequent management of the obtained results.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Conflicts of Interest:** The authors declare no conflicts of interest. All the authors have participated in the different stages of the research and the writing of the paper.

## References

1. Yim, J.; Graham, T.C.N. Using games to increase exercise motivation. In Proceedings of the 2007 Conference on Future Play (Future Play'07), Toronto, ON, Canada, 14–17 November 2007; Association for Computing Machinery: New York, NY, USA, 2007; pp. 166–173. [\[CrossRef\]](#)
2. Luckykumar Dwarkadas, A.; Talasila, V.; Challa, R.K.; Srinivasa, K.G. A review of the application of virtual and augmented reality in physical and occupational therapy. *Softw. Pract. Exp.* **2024**, *54*, 1378–1407. [\[CrossRef\]](#)
3. Marques, L.; Uchida, P.; Pinto Barbosa, S. The impact of Exergames on emotional experience: A systematic review. *Front. Public Health* **2023**, *11*, 1. [\[CrossRef\]](#)
4. Faric, N.; Potts, H.W.W.; Hon, A.; Smith, L.; Newby, K.; Steptoe, A.; Fisher, A. What Players of Virtual Reality Exercise Games Want: Thematic Analysis of Web-Based Reviews. *J. Med. Internet Res.* **2019**, *21*, e13833. [\[CrossRef\]](#)
5. Hensley, Z. The Effect of Virtual Reality Immersion Level on Mood, Enjoyment Level, and Intentions of Future Engagement in Exergames. Master's Thesis, The California State University, Long Beach, CA, USA, 2019.
6. Yansun, E.; Kim, D.; Wünsche, B.C. CoXercise—Perceptions of a Social Exercise Game and its Effect on Intrinsic Motivation. In Proceedings of the 2022 Australasian Computer Science Week (ACSW'22), Brisbane, Australia, 14–18 February 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 176–185. [\[CrossRef\]](#)
7. Dauvergne, C.; Bégel, V.; Gény, C.; Puyjarinet, F.; Laffont, I.; Dalla Bella, S. Home-based training of rhythmic skills with a serious game in Parkinson's disease: Usability and acceptability. *Ann. Phys. Rehabil. Med.* **2018**, *61*, 380–385. [\[CrossRef\]](#)
8. Vargas, A.; Díaz, P.; Zarronandia, T. Using Virtual Reality and Music in Cognitive Disability Therapy. In Proceedings of the 2020 International Conference on Advanced Visual Interfaces (AVI'20), Salerno Italy, 28 September–2 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; [\[CrossRef\]](#)
9. Eun, S.J.; Kim, E.J.; Kim, J.Y. Development and Evaluation of an Artificial Intelligence-Based Cognitive Exercise Game: A Pilot Study. *J. Environ. Public Health* **2022**, *4403976*, 15. [\[CrossRef\]](#)
10. Assuncao, W.G.; Piccolo, L.S.G.; Zaina, L.A.M. Considering emotions and contextual factors in music recommendation: A systematic literature review. *Multimed. Tools Appl.* **2022**, *81*, 8367–8407. [\[CrossRef\]](#)
11. Bayrak, A.; Wünsche, B.; Reading, S. "Let's do it together"—Designing an aerobic exercise game for long-term use. In Proceedings of the Hawaii International Conference on System Sciences (HICSS 2024), Waikiki, HI, USA, 3–6 January 2024; pp. 3778–3787.
12. Ohmoto, Y.; Takeda, S.; Nishida, T. Distinction of Intrinsic and Extrinsic Stress in an Exercise Game by Combining Multiple Physiological Indices. In Proceedings of the International Conference on Games and Virtual Worlds for Serious Applications (VS-Games), Skovde, Sweden, 16–18 September 2015; pp. 1–4. [\[CrossRef\]](#)
13. Burt, C. Having Fun, Working Out: Adaptive and Engaging Video Games for Exercise. Master's Thesis, Carleton University, Ottawa, ON, Canada, 2014.
14. Frachi, Y.; Takahashi, T.; Wang, F.; Barthet, M. Design of Emotion-Driven Game Interaction Using Biosignals. In *HCI in Games, Proceedings of the 4th International Conference, Virtual*, 26 June–1 July 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 160–179. [\[CrossRef\]](#)
15. Nedel, L.; Moni, R.; Nunes, M. Running Wheel: How an exergame can motivate people to perform repetitive, tedious exercises. *XRDS Crossroads ACM Mag. Stud.* **2019**, *25*, 26–29. [\[CrossRef\]](#)
16. Yin, B.; Bailey, S.; Hu, E.; Jayarekera, M.; Shaw, A.; Wünsche, B.C. Tour de Tune 2—Auditory-Game-Motor Synchronisation with Music Tempo in an Immersive Virtual Reality Exergame. In Proceedings of the 2021 Australasian Computer Science Week Multiconference (ACSW'21), Dunedin, New Zealand, 1–5 February 2021; Association for Computing Machinery: New York, NY, USA, 2021. [\[CrossRef\]](#)
17. Yoong, S.Q.; Wu, V.X.; Jiang, Y. Experiences of older adults participating in dance exergames: A systematic review and meta-synthesis. *Int. J. Nurs. Stud.* **2024**, *152*, 104696. [\[CrossRef\]](#)
18. Skjaeret, N.; Nawaz, A.; Morat, T.; Schoene, D.; Helbostad, J.L.; Vereijken, B. Exercise and rehabilitation delivered through exergames in older adults: An integrative review of technologies, safety and efficacy. *Int. J. Med. Inform.* **2016**, *85*, 1–16. [\[CrossRef\]](#)
19. Thingstad, J. The Impact of Spotify's AI-Driven Music Recommender on User Listener Habits. Master's Thesis, University of Agder, Kristiansand, Norway, 2014.
20. Nabizadeh, A.H.; Leal, J.P.; Rafsanjani, H.N.; Shah, R.R. Learning path personalization and recommendation methods: A survey of the state-of-the-art. *Expert Syst. Appl.* **2020**, *159*, 113596. [\[CrossRef\]](#)
21. Sarupuri, B.; Kulpa, R.; Aristidou, A.; Multon, F. Dancing in virtual reality as an inclusive platform for social and physical fitness activities: A survey. *Vis. Comput.* **2024**, *68*, 4055–4070. [\[CrossRef\]](#)



22. Chan, G.; Arya, A.; Orji, R.; Zhao, Z. Motivational strategies and approaches for single and multi-player exergames: A social perspective. *PeerJ. Comput. Sci.* **2019**, *5*, e230. [CrossRef]
23. Games, B. Beat Saber. 2024. Available online: <https://beatsaber.com> (accessed on 15 September 2024).
24. Stanley, K.G.; Livingston, I.; Bandurka, A.; Kapiszka, R.; Mandryk, R.L. PiNiZoRo: A GPS-based exercise game for families. In Proceedings of the International Academic Conference on the Future of Game Design and Technology (Futureplay'10), Vancouver, BC, Canada, 6–7 May 2010; Association for Computing Machinery: New York, NY, USA, 2010; pp. 243–246. [CrossRef]
25. Keskinen, T.; Hakulinen, J.; Turunen, M.; Heimonen, T.; Sand, A.; Paavilainen, J.; Parviainen, J.; Yrjänäinen, S.; Mäyrä, F.; Okkonen, J.; et al. Schoolchildren's user experiences on a physical exercise game utilizing lighting and audio. *Entertain. Comput.* **2014**, *5*, 475–484. [CrossRef]
26. McCrary, J.M.; Gould, M. Rhythm in sport: Adapted rhythmic training to optimize timing and enhance performance in athletes. *J. Sci. Med. Sport* **2023**, *26*, 636–638. [CrossRef]
27. Jones, L.; Karageorghis, C.; Ker, T.; Rushton, C.; Stephenson, S.; Wheeldon, I. The exercise intensity–music-tempo preference relationship: A decennial revisit. *Psychol. Sport Exerc.* **2024**, *74*, 102644. [CrossRef]
28. Dyck, E. Musical Intensity Applied in the Sports and Exercise Domain: An Effective Strategy to Boost Performance. *Front. Psychol.* **2019**, *10*, 1145. [CrossRef]
29. Swinnen, N.; Vandenbulcke, M.; de Bruin, E.; Akkerman, R.; Stubbs, B.; Firth, J.; Vancampfort, D. The efficacy of exergaming in people with major neurocognitive disorder residing in long-term care facilities: A pilot randomized controlled trial. *Alzheimer's Res. Ther.* **2021**, *13*, 13. [CrossRef]
30. Dividat. Cognitive-Motor Solution. Dividat Senso. 2024 Available online: <https://dividat.com/en/senso> (accessed on 15 September 2024).
31. Pesek, M.; Hirci, N.; Znidersic, K.; Marolt, M. Enhancing music rhythmic perception and performance with a VR game. *Virtual Real.* **2024**, *28*, 118. [CrossRef]
32. Pallavicini, F.; Pepe, A. Virtual Reality Games and the Role of Body Involvement in Enhancing Positive Emotions and Decreasing Anxiety: Within-Subjects Pilot Study. *JMIR Serious Games* **2020**, *8*, e15635. [CrossRef]
33. Baradoy, G. A Physiological Feedback Controlled Exercise Video Game. Master's Thesis, University of Calgary, Calgary, AB, Canada, 2012. [CrossRef]
34. Sell, K.; Lillie, T.; Taylor, J. Energy Expenditure During Physically Interactive Video Game Playing in Male College Students with Different Playing Experience. *J. Am. Coll. Health J ACH* **2008**, *56*, 505–511. [CrossRef]
35. Thies, M.J. Controlling Game Music in Real Time with Biosignals. Master's Thesis, University of Texas at Austin, Austin, TX, USA, 2012.
36. Yang, Z.; Yue, Y.; Yang, Y.; Peng, Y.; Wang, X.; Liu, W. Study and application on the architecture and key technologies for IOT. In Proceedings of the 2011 International Conference on Multimedia Technology, Hangzhou, China, 26–28 July 2011; pp. 747–751. [CrossRef]
37. Sethi, P.; Sarangi, S.R. Internet of Things: Architectures, Protocols, and Applications. *J. Electr. Comput. Eng.* **2017**, *2017*, 9324035. [CrossRef]
38. Thingom, I.B. Internet of Things: Design of a New Layered Architecture and Study of Some Existing Issues. *IOSR J. Comput. Eng. (IOSR-JCE)* **2015**, 26–30.
39. ISO/IEC 30141; Internet of Things (IoT)—Reference Architecture. Technical Report. IEC and ISO Organizations: Geneva, Switzerland, 2018.
40. Empatica. Empatica E4 Description. 2021. Available online: <https://www.empatica.com/en-eu/research/e4/> (accessed on 20 February 2024).
41. Meta. Meta Quest 3. 2024. Available online: <https://www.meta.com/es/quest/quest-3/> (accessed on 21 September 2024).
42. Microsoft. Azure Event Grid Documentation. 2024. Available online: <https://learn.microsoft.com/en-us/azure/event-grid/> (accessed on 5 September 2024).
43. Álvarez, P.; Zarazaga-Soria, F.; Baldassarri, S. Mobile music recommendations for runners based on location and emotions: The DJ-Running system. *Pervasive Mob. Comput.* **2020**, *67*, 101242. [CrossRef]
44. Microsoft. Azure Cosmos-DB Documentation. 2024. Available online: <https://learn.microsoft.com/en-us/azure/cosmos-db/> (accessed on 10 October 2024).
45. Álvarez, P.; de Quirós, J.G.; Baldassarri, S. RIADA: A Machine-Learning Based Infrastructure for Recognising the Emotions of Spotify Songs. *Int. J. Interact. Multimed. Artif. Intell.* **2023**, *8*, 168–181. [CrossRef]
46. Russell, J. A circumplex model of affect. *J. Personal. Soc. Psychol.* **1980**, *39*, 1161–1178. [CrossRef]
47. Spotify. Spotify for Developers. 2024. Available online: <https://developer.spotify.com> (accessed on 5 September 2024).
48. Microsoft. Azure Functions Documentation. 2024. Available online: <https://learn.microsoft.com/en-us/azure/azure-functions/> (accessed on 5 September 2024).
49. Scikit-Learn. Scikit-Learn Web Page. 2024. Available online: <https://scikit-learn.org/stable/> (accessed on 10 October 2024).
50. Scikit-Learn. Scikit-Learn Feature Selection Algorithms. 2024. Available online: [https://scikit-learn.org/1.5/api/sklearn.feature\\_selection.html](https://scikit-learn.org/1.5/api/sklearn.feature_selection.html) (accessed on 10 October 2024).
51. Ontañón, S. An Overview of Distance and Similarity Functions for Structured Data. *arXiv* **2020**, arXiv:2002.07420. [CrossRef]

52. Desmet, P.; Vastenburg, M.; Bel, D.V.; Romero, N. Pick-A-Mood: Development and application of a pictorial mood-reporting instrument. In Proceedings of the 8th International Conference on Design and Emotion: Out of Control Central Saint Martins college of Art Design, London, UK, 11–14 September 2012; pp. 1–12. [[CrossRef](#)]
53. Baldassarri, S.; García de Quirós, J.; Beltrán, J.R.; Álvarez, P. Wearables and Machine Learning for Improving Runners' Motivation from an Affective Perspective. *Sensors* **2023**, *23*, 1608. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.