

Tarea Integradora 3 NeoTunes
Andrés Cabezas A00394772
Samuel Alvarez A00394750

PROBLEM SPECIFICATION TABLE

| | |
|-----------------------------|---|
| CLIENT | NeoTunes |
| USER | NeoTunes Administrator |
| REQUIREMENTS | <ul style="list-style-type: none">- R1: Register producer- R2: Register consumer- R3: Register audio- R4: Create a Playlist- R5: Edit a Playlist- R6: Share a playlist.- A7: Simulate playing a song or podcast (standard and premium).- R8: Buy a song.- R9: Generate reports with the recorded data |
| CONTEXT OF THE PROBLEM | Develop a Multimedia Content application with functions and characteristics similar to those that already exist on the market, but which also has new functions that allow it to stand out from the others. |
| NON-FUNCTIONAL REQUIREMENTS | <ul style="list-style-type: none">- RN1: It will be done in 2 deliveries, and the dates are stipulated by the client- RN2: The code will be done in the JAVA language.- RN3: The documentation must be in the forms stipulated by the client. |

Traceability

| Functional Requirement | Class | Name Method Name |
|----------------------------|------------|--------------------|
| R1: Register producer user | Controller | registerProducer() |
| | Controller | searchUser() |
| | Controller | addArtist() |
| | Artist | Artist() |
| | Controller | addCreator() |
| | Creator | Creator() |
| R2: Register consumer user | Controller | registerConsumer() |
| | Controller | searchUser () |
| | Controller | registerStandard() |
| | Standard | Standard() |
| | Controller | registerPremium() |
| | Premium | Premium() |
| R3: Register audio | Controller | registerAudio() |
| | Controller | searchAudio() |
| | Controller | registerSong() |
| | Song | Song() |
| | Controller | registerPodcast() |
| | Podcast | Podcast() |
| R4: Create a Playlist | Controller | createPlaylist() |
| | Controller | searchUser() |
| | Playlist | Playlist() |
| | Standard | addPlaylist() |
| | Premium | addPlaylist() |
| R5: Edit a Playlist | Controller | editPlaylist() |
| | Controller | searchUser() |

| | | |
|-----------------------------|------------|---------------------------|
| | Controller | showUserPlaylists() |
| | Controller | showNeoTunesAudio() |
| | Controller | addAudioToPlaylist() |
| | Playlist | addAudio() |
| | Controller | showPlaylistAudios() |
| | Controller | deleteAudioToPlaylist () |
| | Playlist | deleteAudio() |
| | Controller | renamePlaylist() |
| | Playlist | setName() |
| R6: Share Playlist | Controller | sharePlaylist() |
| | Controller | showUserPlaylist() |
| | Controller | searchUser() |
| | Consumer | searchPlaylist() |
| | Playlist | showMatrix() |
| | Playlist | createPlaylistCode() |
| R7: Simulate audio playback | Controller | simulateAudioPlayback() |
| | Controller | searchUser() |
| | Consumer | searchPlaylist() |
| | Playlist | searchAudio() |
| | Song | play() |
| | Podcast | play() |
| | Controller | addPlaybacks() |
| R8: Buy Song | Controller | buySong() |
| | Controller | showNeoTunesSongs() |
| | Controller | searchAudio() |
| | Consumer | addSong() |
| | Song | setSaleDate() |
| | Controller | addSales() |
| R10: Generate report with | Controller | generateInform() |

| | | |
|---------------|------------|--------------------------|
| recorded data | Controller | searchUser() |
| | Controller | searchAudio() |
| | Controller | showAudioTotalPlayback() |
| | Controller | mostListenGenre() |
| | Controller | mostListenCategory() |
| | Controller | showTop5ProducersInf() |
| | Controller | showTop10AudioInf() |
| | Controller | showGenreInfo() |
| | Controller | showInMostSale() |

Functional requirements analysis table

| | | | |
|--|--|-----------|--|
| Name or identifier | R1: Register User Producer | | |
| Summary | This registered a new User Producer (Artist or Creator) | | |
| Entries | Entry name | Data type | Selection or repetition |
| | condition nickName | String | |
| | id | String | The id doesn't exist |
| | name | String | |
| | url | String | |
| | artistOrCreator | int | (1-Artist : 2-Creator) |
| General activities necessary to obtain the results | <ol style="list-style-type: none"> 1. Choose if it is an Artist or Creator 2. Create the Artist or Creator 3. Add it to the list of users | | |
| Result or postcondition | A new registered user | | |
| Outputs | Input name | Data type | Selection or repetition condition |
| | Message | String | If all goes well the approval message appears. |

| | | | |
|--|---------|--------|--|
| | Message | String | If the process is not correct, an error message is output. |
|--|---------|--------|--|

| | | | |
|--|--|-----------|--|
| Name or identifier | R2: Register User Consumer | | |
| Summary | This registered a new User consumer(standard or Premium) | | |
| Entries | Entry name | Data type | Selection or repetition |
| | nickName | String | |
| | id | String | The id doesn't exist |
| | name | String | |
| | url | String | |
| | standardOrPremium | int | (1-Standard : 2-Premium) |
| General activities necessary to obtain the results | <ol style="list-style-type: none"> 1. Choose if it is Standard or Premium 2. Create Standard or Premium 3. Add it to the list of users 4. Show the process message | | |
| Result or postcondition | A new registered user | | |
| Outputs | Input name | Data type | Selection condition or repetition |
| | Message | String | If everything goes well, the approval message appears. |
| | Message | String | If the process is not correct, an error message is output. |

| | | | |
|--|--|-----------|--|
| Name or identifier | R3: Register audio | | |
| Summary | This registered a new audio to a Producer | | |
| Entries | Entry name | Data type | Selection or repetition |
| | idProducer | String | The idProducer already exist |
| | name | String | |
| | url | String | |
| | duration | int | Integer Number |
| | album | String | |
| | saleValue | double | Double Number |
| | typeGenre | int | Integer Number |
| | description | String | |
| | category | int | Integer Number |
| | songOrPodcast | int | (1-Song : 2-Podcast) |
| General activities necessary to obtain the results | <ol style="list-style-type: none"> 1. Check that the producer exists 2. Select the producer that will record the audio 3. Check that the audio does not already exist 4. Show the message of the process | | |
| Result or postcondition | A new audio registered | | |
| Outputs | Input name | Data type | Selection or repetition condition |
| | Message | String | If everything goes well, the approval message appears. |
| | Message | String | If the process is not correct, an error message is output. |

| | |
|--------------------|--|
| Name or identifier | R4: Create a Playlist |
| Summary | This registered a new Playlist for a User selected |

| | | | |
|--|--|-----------|--|
| Entries | Entry name | Data type | Selection or repetition |
| | idUser | String | The idUser already exist |
| | name | String | |
| General activities necessary to obtain the results | <ol style="list-style-type: none"> 1. Select the User who is going to create aPlaylist 2. Show the process message | | |
| Result or postcondition | A new Playlist registered to a selected user | | |
| Outputs | Input name | Data type | Selection or repetition condition |
| | Message | String | If everything goes well, the approval message appears. |
| | Message | String | If the process is not correct, an error message is output. |

| | | | |
|--|---|-----------|------------------------------|
| Name or identifier | R5: Edit a Playlist | | |
| Summary | This edit a Playlist selected, of a user selected | | |
| Entries | Entry name | Data type | Selection or repetition |
| | idUser | String | The idUser already exist |
| | numberPlaylist | int | Integer Number |
| | nameAudio | String | |
| | idProducer | String | The idProducer already exist |
| | newName | string | |
| General activities necessary to obtain the results | <ol style="list-style-type: none"> 1. Select the user who has the Playlist 2. Select the Playlist you want to edit 3. Select the menu option (add - delete - rename) 4. Select the Song to edit (add - delete) 5. Enter the new name of the Playlist | | |

| | | | |
|-------------------------|-----------------------------|-----------|--|
| | 6. Show the process message | | |
| Result or postcondition | An edited Playlist | | |
| Outputs | Input name | Data type | Selection or repetition condition |
| | Message | String | If everything goes well, the approval message appears. |
| | Message | String | If the process is not correct, an error message is output. |

| | | | |
|--|--|-----------|--|
| Name or identifier | R6: Share Playlist | | |
| Summary | This method shares a playlist of a Consumer | | |
| Entries | Entry name | Data type | Selection or repetition |
| | idConsumer | String | The idConsumer already exist |
| | playlistPos | int | Integer number |
| General activities necessary to obtain the results | <ol style="list-style-type: none"> 1. Select the user who has the Playlist 2. Select the Playlist to share 3. Show the playlist matrix 4. Show the playlist code | | |
| Result or postcondition | The playlist matrix and his code | | |
| Outputs | Input name | Data type | Selection or repetition condition |
| | Message | String | If everything goes well, the approval message appears. |

| | | | |
|--|---------|--------|--|
| | Message | String | If the process is not correct, an error message is output. |
|--|---------|--------|--|

| | | | |
|--|--|-----------|--|
| Name or identifier | R7: Simulate audio playback | | |
| Summary | This method simulates the playback of a song or podcast | | |
| Inputs | Input name | Data type | Selection or repetition |
| | idConsumer | String | The idConsumer already exist |
| | playlistPos | int | Integer Number |
| | audioName | String | |
| General activities necessary to obtain the results | <ol style="list-style-type: none"> 1. Select the user who has the Playlist 2. Select the Playlist that has the audio to play 3. Select the name of the Audio to play 4. Add playback and time played to the NeoTunes and the user audio 5. Play audio, or advertising before it (If applicable) | | |
| Result or postcondition | Audio played or promotion before it (If applicable) | | |
| Outputs | Input name | Data type | Selection or repetition condition |
| | Message | String | If everything goes well, the approval message appears. |
| | Message | String | If the process is not correct, an error message is output. |

| | | | |
|--------------------|--|-----------|-------------------------|
| Name or identifier | R8: Buy song | | |
| Summary | This method allows to buy a song of NeoTunes | | |
| Inputs | Input name | Data type | Selection or repetition |

| | | | |
|--|--|-----------|--|
| | idConsumer | String | The idConsumer already exist |
| | idProducer | String | The idProducer already exist |
| | audioName | String | |
| General activities necessary to obtain the results | <ol style="list-style-type: none"> 1. Select the user who will buy the song 2. Select the song to be purchased 3. Add the song to the user's song arraylist (if there's space available) 4. Add a new sale to the song | | |
| Result or postcondition | Song purchased by the consumer | | |
| Outputs | Input name | Data type | Selection or repeat condition |
| | Message | String | If everything goes well the approval message appears. |
| | Message | String | If the process is not correct, an error message is output. |

| | |
|--------------------|--|
| Name or identifier | R9: Generate reports with the registered data |
| Summary | <p>With this requirement it is possible to generate reports through the registered data in the system. There will be a variable called option that will allow you to choose the required report.</p> <p>The available options are:</p> <ul style="list-style-type: none"> - For each type of audio, songs and podcast, inform the total accumulated reproductions in the entire platform. - Report the most listened song genre (name and number of plays) for a specific user and for the entire platform and its number of plays. - Report the most listened to podcast category (name and number of views) for a specific user and for the entire platform. - For each of the members of the Top 5 artists and Top 5 content creators on the platform, report the name and number of total reproductions. |

| | | | |
|--|--|-----------|--|
| | <ul style="list-style-type: none"> - For each of the members of the Top 10 songs and Top 10 podcasts, report the name, genre or category, and total number of reproductions. - For each genre, report the number of songs sold and the total value of sales (\$). - Of the best-selling song on the platform, report the total number of sales and the total sale value (\$). | | |
| Inputs | Input name | Data type | Selection or repetition condition |
| | option | int | Integer number |
| | | | |
| | | | |
| | | | |
| General activities necessary to obtain the results | <ol style="list-style-type: none"> 1. Read the option that the user will choose for the report 2. Generate reports 3. Show the generated report on the screen | | |
| Result or postcondition | The generated report is displayed. | | |
| Outputs | Input name | Data type | Selection or repetition condition |
| | Message | String | If everything goes well the approval message appears. |
| | Message | String | If the process is not correct, an error message is output. |