- Identificación del problema y análisis de requerimientos

Alexis Jaramillo

Customer	**Neotunes**
User	Neotunes users/customers: Artists, content creators, standar users, premium users…
functional requirements	**RF1:** Register producer users (artists and content creators). **RF2:** Register consumer users (standard and premium). **RF3:** Register songs and podcasts. **RF4:** Create a playlist. **RF5:** Edit a playlist. **RF6:** Share a playlist. **RF7:** Simulate playback of a song or podcast (standard and premium user). **RF8:** Buy a song. **RF9:** **a.** For each type of audio, songs and podcast, report the total accumulated plays across the platform. **b.** Report the most listened song genre (name and number of plays) for a specific user and for the entire platform. **c.** Report the most listened podcast category (name and number of plays) for a specific user and for the entire platform. **d.** Report the name and total number of reproductions of each of the members of the Top 5 artists and Top 5 content creators on the platform. **e.** Inform the name, genre or category and total number of reproductions of each of the members of the Top 10 songs and Top 10 podcasts. **f.** For each genre, report the number of songs sold and the total sales value ($). **g.** Report the total number of sales and the total sales value ($) of the top selling song on the platform.
Contexto del problema	A Danish company, "Neotunes" is interested in competing in the streaming music and audio content market. Its focus is on user

	subscriptions and selling songs through the platform and requires a software solution that allows its users to truly own their music catalog.
Requerimientos no funcionales	RNF1: Scalable software. RNF2: the prototype design must contemplate the future creation of other types of audio. RNF3: the prototype design must contemplate the future creation of other types of users, both consumers and producers.

Name or identifier	RF1: Register producer users		
Abstract	To register producer users, we must consider that there are two types of producer users with similar characteristics, except for the type of production of each one, while an artist type producer user produces songs, the producer user, content creator, produces podcasts. With this in mind, we simply ask for the data and the type of producer user to be registered.		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**
	nickname	String	
	id	String	
	attachmentDate	LocalDate	
	name	String	
	representativeUrl	String	
General activities necessary to obtain the results	Perform two different methods, one that registers artist type users and another that does the same for content creators or perform only one method, but make use of "instanceOf" depending on the type of user required		
Result or postcondition	User registered and added to the "allUsers" arraylist.		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	msg	String	will indicate whether the user was able to register or not

Name or identifier	RF2: Register consumer users		
Abstract	To register consumer users, we must consider that there are two types of consumer users, (standard users and premium users) both share attributes, however, they differ in the limitations that each one has. With this in mind, we simply ask for the data and the type of consumer user you wish to register.		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**
	nickname	String	
	id	String	
	attachmentDate	LocalDate	
General activities necessary to obtain the results	Perform two different methods, one that registers standard type users and one that registers premium type users or perform only one method, but make use of "instanceOf" depending on the type of user required		
Result or postcondition	Consumer user registered and added to the arrayList of "allUsers".		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	msg	String	will indicate whether the user was able to register or not

Name or identifier	RF3: Record audio (songs and podcasts)		
Abstract	To record audio, we must consider that there are two types of audios, songs and podcasts, and, although both share attributes, they have their own attributes, therefore, the parameters received to record audio will vary depending on the type of audio being recorded.		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**
	name	String	
	distinctiveUrl	String	
	duration	double	

	album	String	Only if a song is registered
	genre	int	Only if a song is registered
	salesValue	double	Only if a song is registered
	description	String	Only if the type of audio to be recorded is a podcast.
	category	int	Only if the type of audio to be recorded is a podcast.
General activities necessary to obtain the results	Create two enumerations, one for the possible song genres and one for the existing podcast categories and then assign them their corresponding value. Perform two different methods, one that records audios of song type and one that records audios of podcast type.		
Result or Postcondition	Song or podcast successfully recorded		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	msg	String	indicates whether the type of audio could be recorded

Name or identifier	RF4: Create playlist		
Abstract	Consumer users have the possibility to create playlists, depending on the type of user, you can create as many playlists as possible, or you will have a limit on the number of playlists you can create.		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**
	playlistName	String	
	nickname	String	verify if you are a consumer user
General activities necessary to obtain the results	Request the name of the playlist to be created Display the existing consuming users Search for the selected user Add the corresponding playlist		

Result or Postcondition	Playlist created and added to the corresponding user		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	msg	String	will indicate whether the playlist could be created or not.

Name or identifier	RF5: Edit playlist		
Abstract	In the previous requirement it was about creating a playlist, and in this one, you will simply modify that playlist already created, either by adding songs, deleting them, changing the name of the playlist...		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**
	playlistName	String	validate that the playlist exists
	nickname	String	verify if you are a consumer user
General activities necessary to obtain the results	Show existing playlists Select playlist Add or remove audio types, as the case may be		
Result or Postcondition	Playlist modified with the new changes		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	msg	String	Playlist modified with the new changes

Name or identifier	RF6: Share playlist		
Abstract	As a user, I can share my created and modified playlists with other users by means of an auto-generated code.		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**

	nickname	String	
	playlistName	String	validate that the playlist name exists
General activities necessary to obtain the results	Show existing playlists Select playlist Auto-generate code to share playlist by means of a 6x6 matrix with random numbers from 0 to 9		
Result or Postcondition	The code that will allow others to access the playlist		
Outputs	**Nombre salida**	**Tipo de dato**	**Condición de selección o repetición**
	codeMessage	String	if the user selected a valid option, the code for that playlist will be autogenerated.

Name or identifier	RF7: Simulate audio playback		
Abstract	Depending on the type of user, the audio playback simulation will be different, in the case of the standard user, every two simulations, an advertisement must be played, while for the premium user there are no such restrictions.		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**
	nickname	String	
	audioName	String	
General activities necessary to obtain the results	Determine if the song is being listened to by a premium or standard user. In case of standard, display the default ads every two audios played. Search for the name of the audio entered by the user and start playing it.		
Result or Postcondition	simulation of the audio to be played		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**

	msg	String	if the audio exists

Name or identifier	RF8: Buy song		
Abstract	songs can be purchased by consumer users for a specified price		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**
	nickname	String	only if it is a consumer user
	audioName	String	only if it's a song
General activities necessary to obtain the results	Request the nickname of the user who wants to buy the song Search for the song typed by the user Display the song and its price Redirect the user so that he/she can pay for his/her song		
Result or Postcondition	Song purchased by the user		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	true	boolean	if the song could be purchased
	false	boolean	if the song could not be purchased

SUBDIVISIONS OF REQUIREMENT NINE

Name or identifier	RF9A: Generate a report of the total number of plays of songs and podcasts on the entire platform.		
Abstract	A report of the total accumulated plays of the songs and podcasts within the platform is required.		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**
	audioType	String	
General activities necessary to obtain the results	as it is a report of the total accumulated reporductions, podcast or songs, we only need to know which of the two reports the user wants to consult and show the accumulated reproductions.		

Result or Postcondition	the report generated successfully		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	Report	String	

Name or identifier	RF9B: Report the most listened song genre for a specific user and for the entire platform.		
Abstract	Report the name and number of plays of the most listened song genre for a specific user and for the entire platform.		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**
	nickname	String	if the report is for a specific user
	genre	int	
General activities necessary to obtain the results	in case the report of the most listened generated, is for a user, we must ask for the user, validate that he exists, and access his list of audios and from each song get the genre and start the counter for each genre, comparing the four genres to each other to see which one has a higher number. for the case of the whole platform, the same process is done, but with all the users registered in the platform.		
Result or Postcondition	the report generated successfully		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	report	String	if the report is for a specific user
	totalReport	String	if the report is for whole platform

Name or identifier	RF9C: Report the most listened podcast category for a specific user and for the entire platform.		
Abstract	Report the name and playback number of the most listened podcast category for a specific user and for the entire platform.		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**
	nickname	String	if the report is for a specific user
	category	Int (enum: categoryType)	
General activities necessary to obtain the results	in case the report of the most listened generated, is for a user, we must ask for the user, validate that he exists, and access his list of audios and from each podcast get the category and start the counter for each category, comparing the categories to each other to see which one has a higher number. for the case of the whole platform, the same process is done, but with all the users registered in the platform.		
Result or Postcondition	the report generated successfully		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	report	String	if the report is for a specific user
	totalReport	String	if the report is for whole platform

Name or identifier	RF9D: Generate information on top five artists and creators		
Abstract	the top 5 artists and the top 5 content creators on the platform should be generated showing the name, name and total number of reproductions of each one.		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**

General activities necessary to obtain the results	For both types of users, methods will be created that allow us to calculate the total number of reproductions they have had on the platform and will be ordered from highest to lowest, showing the name of the user and their reproduction number (in this case we will only consider the first five).		
Result or Postcondition	the report generated successfully		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	report	String	

Name or identifier	RF9E: generate top 10 songs and podcasts reports		
Abstract	the podcasts and songs have an accumulated number of reproductions, and depending on this, those with the highest number of reproductions will be placed in the top 10, showing their respective name and genre or category as appropriate.		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**
General activities necessary to obtain the results	Separate lists,one method for the most played songs and their genre and another for the most played podcasts and their category. Print the results of the previous methods		
Result or Postcondition	the report generated successfully		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	report	String	

Name or identifier	RF9F: Generate report of the number of sales of each genre
Abstract	of the four possible genre types, generate a report of the number of songs sold and the total sales value of those songs.

Inputs	**Input name**	**Data type**	**Selection or repetition condition**
General activities necessary to obtain the results	each song has a genre and a sales value, taking this into account we will perform methods to differentiate the songs by genre, and once this is done, we will verify those songs that have been sold, storing and counting them, once this data is obtained, we will obtain the attribute "sales value" of each song and we will add these values.		
Result or Postcondition	the report generated successfully		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	report	String	

Name or identifier	RF9G: Generate report of the best-selling song on the platform		
Abstract	For this requirement you must find the bestselling song on the platform and report the total number of sales and the total sales value ($).		
Inputs	**Input name**	**Data type**	**Selection or repetition condition**
General activities necessary to obtain the results	To have the above requirement get the number one song of the top count the number of times it was sold multiply the number of times it was sold with its sales value		
Result or Postcondition	the report generated successfully		
Outputs	**Output name**	**Data type**	**Selection or repetition condition**
	report	String	