

Music Bot Quiz on Discord

Instituição:

Pontifícia Universidade Católica de São Paulo

Matéria:

Laboratório de Projetos de Software /LPS-2021

Professor responsável:

Italo Santiago Vega

Equipe:

Kahao Yu,
Leonardo de Jesus Diz Conde,
Marcelo Maués Botelhos,
Victor Yuji Saito.

Data:
22/09/2021

V.4.0

Sumário:

- [1. Objetivo \(Definição do projeto\).](#)
 - 1.1. Contexto do Problema
 - 1.2. Introdução
 - 1.3. Complicações
- [2. Análise do problema](#)
 - 2.1. Considerações prévias do grupo
 - 2.2. Análise Prévia do Objetivo
 - 2.3. Análise do Domínio
 - 2.3.1. Conceito
 - 2.3.2. Objetos
 - 2.4. Mapeamento conceitual
 - 2.4.1. Regras de Negócio
 - 2.4.2. Casos de usos
 - 2.5. Modelo de Domínio
 - 2.6. Modelo de Desenho
 - 2.7. Requisitos
- [3. Desenho](#)
- [4. Implementação](#)
- [5. Conclusões](#)

1. Objetivo (Definição do projeto):

1.1. Contexto do Problema:

O nosso objetivo é criar um *bot* destinado a rede social chamada *Discord* que terá as seguintes funcionalidades: **tocar** músicas, ter um **quiz** de músicas temáticas e comando para auxiliar no uso dos comandos disponíveis dele.

A funcionalidade de **tocar** música deverá permitir tocar músicas requisitadas pelo usuário, além de apresentar alguns comandos para o melhor controle das músicas.

O participante do **quiz** de músicas temáticas tem como objetivo acertar o maior número de "origens" da música (exemplo: se o tema é *anime*, o usuário deverá acertar o *anime* em que a música aparece). Contudo, para dar chance a todos os jogadores, o *bot* deve aumentar a chance de uma música conhecida aparecer, mas não ao ponto de todas serem conhecidas, para que também possa despertar a curiosidade de alguns usuários em buscar novas experiências.

Já o comando de **auxilio**, poderá ser feito de 2 maneiras: a menção ao *bot* no *chat* de texto (exemplo de mensagem: "@< Nome do bot >") ou pelo prefixo do *bot* junto da palavra "help".

1.2. Introdução:

Nossas inspirações vêm principalmente de 2 lugares: [Anime Music List \(AMQ\)](#), e dos *bots* que a rede social [Discord](#), sendo a [Loritta](#) um exemplo que o grupo considera como um projeto de sucesso e inspiração. Fora isso, os integrantes do grupo, em um geral, estavam interessados em criar uma aplicação real-time e que pudéssemos aplicar conceitos de outras matérias que estamos vendo nesse semestre.

1.3. Complicações:

RNF1: O projeto envolve o uso de interface que não é gerenciada pelos membros do projeto, portanto a primeira complicação é que o projeto deve seguir as regulamentações desse serviço (*Discord*) que está disponível no [site de políticas aos desenvolvedores](#). Um problema que os membros irão se deparar é com uma questão que surgiu após a idealização desse projeto é que o *bot* de música mais usado pela comunidade do *Discord* (o [Groovy](#)), onde ele foi forçado a ser imobilizado por questões de integrações com algumas API's de músicas com o *Youtube*, mais informações disponíveis na [matéria feita pelo portal Tecmundo](#) e o [pronunciamento do criador do Groovy](#). (última visualização dos links: 06/08/2021)

RNF2: Tendo um mês se passado da polêmica do *Groovy*, mais dois *bots* acabaram tendo suas ações paralizadas pelas mesmas questões, um deles foi o [Rythm](#) e o outro foi o [Hydra](#). (última visualização dos links: 21/09/2021)

2. Análise do problema:

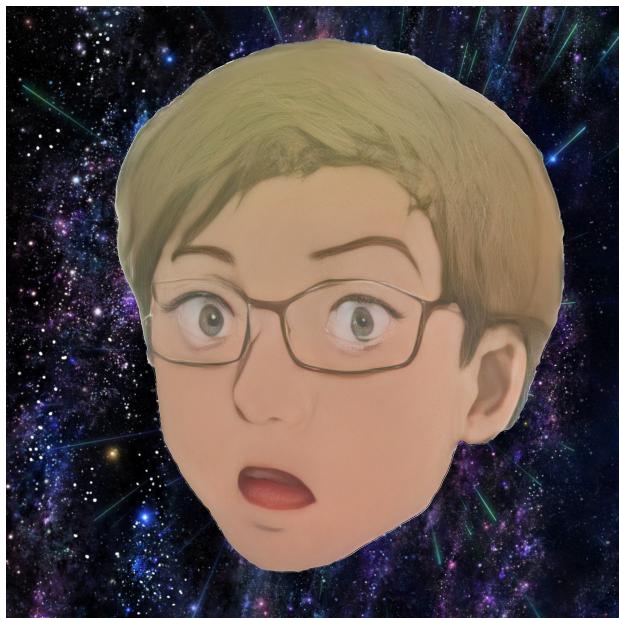
2.1. Decisões prévias:

2.1.1. Funcionais:

Para facilitar na produção da aplicação, o grupo definiu que o jogo irá se basear inicialmente em 10 músicas por jogo, se possível tentaremos tornar esse aspecto modular.

2.1.2. Cosméticas:

Para o ícone do *bot* definimos que será a imagem a baixo:



Ela foi produzida com base em uma foto de um integrante do grupo, passamos por um filtro do Snapchat, usamos a imagem em um aplicativo chamado [Remini](#) e ao final fizemos uma edição de imagem com o Gimp para recortar a cabeça e colocar no fundo de [espaço](#). Quanto ao nome, definimos que seria "Baka!" por conta de piadas internas do grupo.

2.2. Análise Prévia do Objetivo:

- Quiz de músicas (analogia):

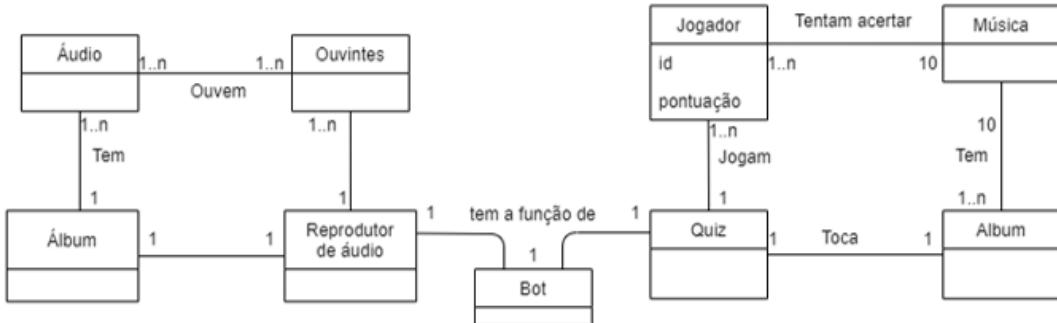
Normalmente nos jogos desse gênero, uma música é apresentada por um período curto e os usuários devem acertar o nome, gênero, época, ou banda/grupo/artista (depende do jogo para definir o objetivo) e ao final da partida é mostrada a resposta.

- Áudio Player (componente conceitual):

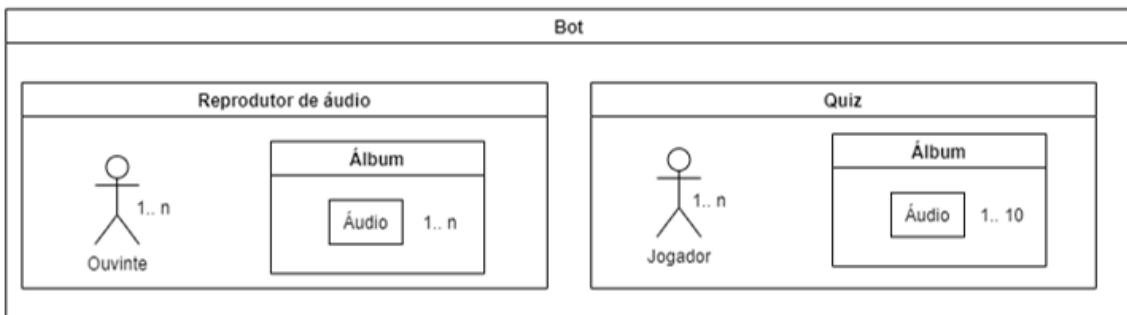
Os *bots* que envolvem áudio, quase que por sua totalidade, implementam a funcionalidade de áudio *player*, ou seja, ele irá reproduzir uma música em um canal de voz no momento que um usuário pede, além desse ter a possibilidade de pausar/tocar a música, adicionar mais músicas, pular e parar a reprodução.

2.3. Análise do Domínio:

2.3.1. Conceito:



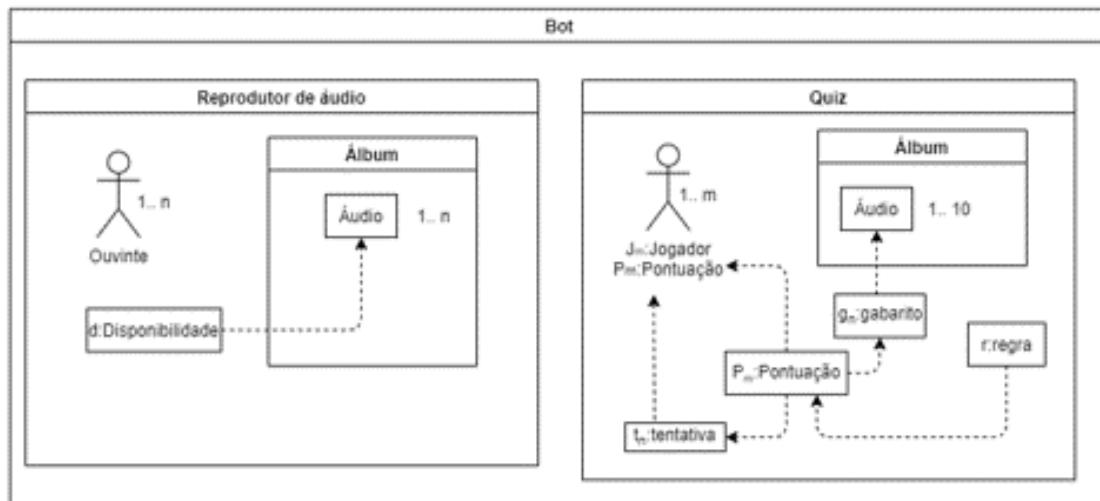
2.3.2. Objetos:



- Ouvinte
- Jogador
- Áudio
- Álbum
- Reprodutor de áudio
- Quiz

2.4. Mapeamento conceitual:

2.4.1. Regras de Negócio:



As regras de negócio que a aplicação deve seguir são:

- Gerais:

Um *bot*, assim como um usuário, só pode estar conectado a um canal de voz por servidor, por isso ele deverá ter o controle das ações que estão acontecendo para não deixar que as suas funcionalidades não entrem em conflito. Já com relação as mensagens de texto, as únicas limitações existentes são aquelas que infrigem as políticas do Discord (comentada no tópico 1.3. *Complicações*).

- Reprodutor de áudio:

Disponibilidade d= estar dentro do acervo de áudios disponibilizados pelo bot;

- Quiz:

A pontuação de cada jogador (P_m) é dada pela quantidade de músicas de músicas acertadas, representada pela fórmula abaixo:

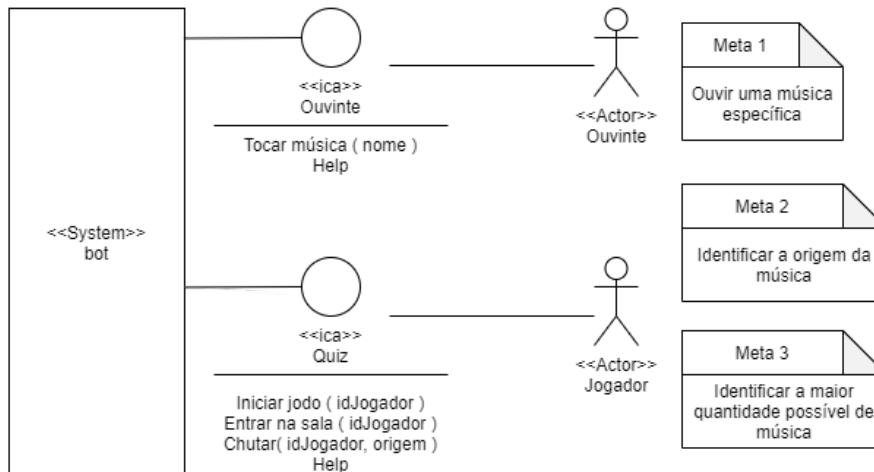
$$P_m = \sum_{n=1}^{10} (t_n = g_n) ? 1 : 0$$

Considerando t_n como a última tentativa do jogador P_m para cada música e g_n a resposta de cada música.

Já o ganhador do jogo, dada pela regra r, é obtida com base no jogador que concluir com maior pontuação, ou seja, o maior P_m .

O jogo terá seu fim quando todas as músicas terminarem de tocar. Com isso

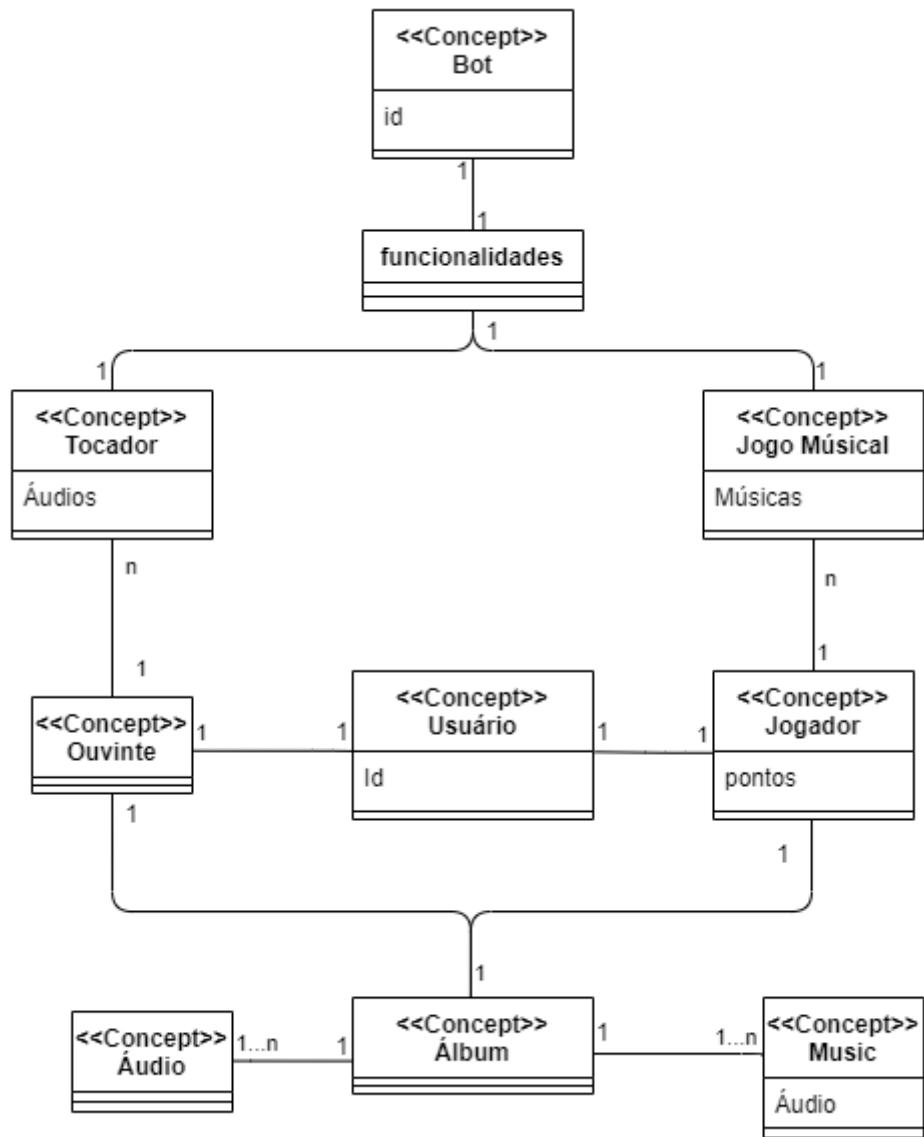
2.4.2. Casos de usos:



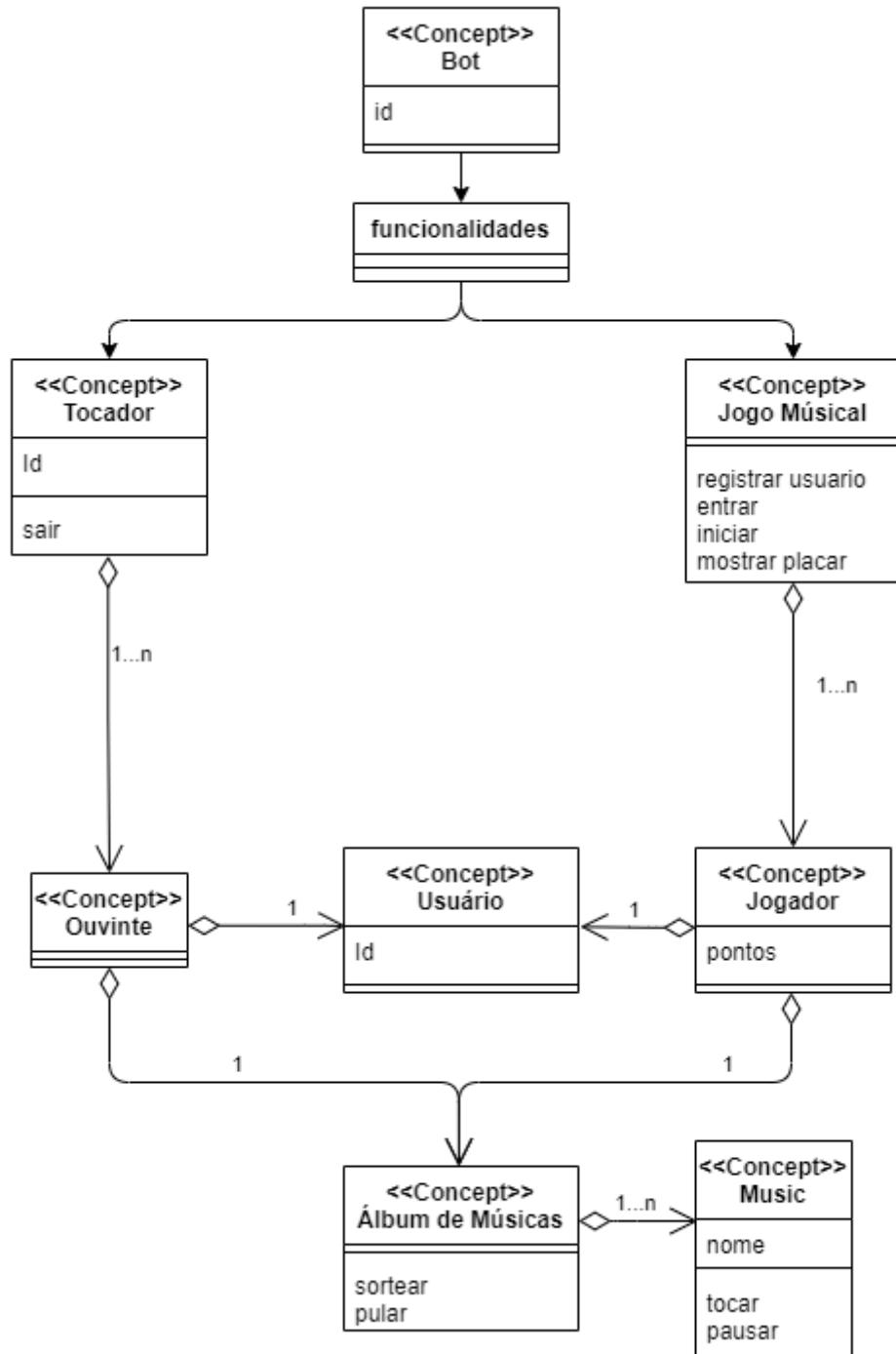
<<UseCase 1>>
1. Jogador pede o início do jogo 2. Bot cria uma sala para o quiz 3. Bot recebe outros jogadores durante um certo intervalo de tempo 4A. Se a sala tiver jogadores... 4A.1. O bot entra no canal de voz 4A.2. O bot toca um trecho da música 4A.3. Os jogadores tentam identificar a origem da música 4A.4. O bot contabiliza os chutes até terminar o trecho 4A.5. O bot sai do canal de voz 4A.6. O bot informa os pontos obtidos para cada jogador 4B. Se a sala não possui jogadores... 4B.1. O bot envia uma mensagem de operação cancelada 5. O bot termina as operações relacionadas ao quiz

<<UseCase 2>>
1. Jogador pede para tocar uma música 2. Bot cria uma sala para a música 3A. Se a sala tiver pelo menos um ouvinte... 3A.1. O bot entra no canal de voz 3A.2. O bot toca a música 3A.3. Se um dos ouvintes... 3A.3.1. Pedir outra música... 3A.3.1.1. O bot adiciona a música em uma fila 3A.3.1.2. Quando terminada a música, ele toca a próxima 3A.3.2. Pedir para pausar 3A.3.2.1. O bot pausa a música 3A.3.2.2. O ouvinte pede para despuçar 3A.3.2.3. O bot retoma a tocar a música 3A.3.3. Pedir para sair 3A.4. O bot sai do canal de voz 3B. Se a sala não possui ouvintes... 3B.1. O bot envia uma mensagem de operação cancelada 4. O bot termina as operações relacionadas a tocar música

2.5. Modelo de Domínio:



2.6. Modelo de Desenho:

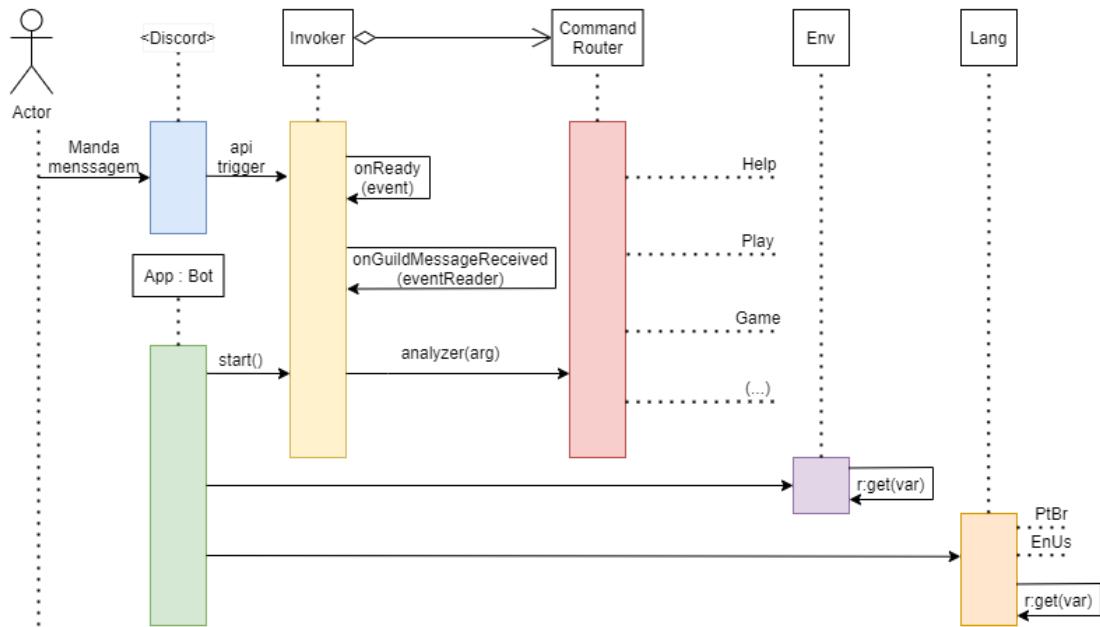


2.7. Requisitos:

1. O bot deve apresentar a possibilidade de aumentar a chance de uma música conhecida aparecer no quiz;
2. Quando uma música está sendo tocada (no modo do reproduutor de áudio), o usuário poderá usar comandos para pausar/despausar a música;
3. Ao final de cada jogo, o bot deverá apresentar o placar da pontuação obtida pelos jogadores;
4. O bot deverá ter a capacidade de criar um canal para suas funcionalidades de quiz e de reproduutor de áudios;
5. O bot deve apresentar um comando de “ajuda”;

3. Desenho:

3.1. Arquitetura:



[incompleto]

3.2. Implementação:

O código está disponível no link: <https://github.com/LeonardoConde/BakaBot>

[incompleto]

BakaBot:

```
| .env  
| .env.example  
| .gitattributes  
| .gitignore  
| gradlew  
| gradlew.bat  
| Out.txt  
| README.md  
| settings.gradle.kts  
|  
+---.gradle  
| +---7.2  
| | | gc.properties  
| | |  
| | +---dependencies-accessors  
| | | dependencies-accessors.lock  
| | | gc.properties  
| | |  
| | +---executionHistory  
| | | executionHistory.bin  
| | | executionHistory.lock
```

```
| | |
| | +---fileChanges
| | |     last-build.bin
| | |
| | +---fileHashes
| | |     fileHashes.bin
| | |     fileHashes.lock
| | |
| | ---vcsMetadata-1
| +---buildOutputCleanup
| |     buildOutputCleanup.lock
| |     cache.properties
| |     outputFiles.bin
| |
| +---checksums
| |     checksums.lock
| |     sha1-checksums.bin
| |
| ---vcs-1
|     gc.properties
|
+---.idea
|     .gitignore
|     compiler.xml
|     gradle.xml
|     jarRepositories.xml
|     misc.xml
|     runConfigurations.xml
|     vcs.xml
|     workspace.xml
|
+---app
| |     build.gradle.kts
| |
| +---build
| |     +---classes
| | |     ---kotlin
| | | |     ---main
| | | | |     Env.class
| | | | |
| | | |     +---bakaBotSrc
| | | | |     AppKt.class
| | | | |
| | | |     +---app
| | | | |     Bot.class
| | | | |     CommandRouter.class
| | | | |     Involker.class
| | | | |
| | | |     +---integration
| | | | |     +---list
| | | | | |     Anilist.class
| | | | | |     Mal.class
```

```
| | | | | |
| | | | |   ---music
| | | | |     Spotify.class
| | | | |
| | | |   +---locale
| | | |     EnUs.class
| | | |     LangId.class|||LangWhenMappings.class
| | | |     Lang.class
| | | |     PtBr.class
| | | |
| | |   +---model
| | |     AudioManager.class
| | |     DaoWrapper.class
| | |     Identifier.class
| | |     MessageManager.class
| | |
| |   +---module
| |     SignUpDao.class
| |
| |   ---service
| |     +---command
| |       Help.class
| |       PlayAudio.class
| |       SignUp.class
| |       SongGame.class
| |
| |   ---process
| |     Log.class
| |
| | ---META-INF
| |   app.kotlin_module
| |
| +---kotlin
|   ---compileKotlin
|     build-history.bin
|     last-build.bin
|   |
|   ---caches-jvm
|     +---inputs
|       source-to-output.tab
|       source-to-output.tab.keystream
|       source-to-output.tab.keystream.len
|       source-to-output.tab.len
|       source-to-output.tab.values.at
|       source-to-output.tab_i
|       source-to-output.tab_i.len
|   |
|   +---jvm
|     ---kotlin
|       class-attributes.tab
|       class-attributes.tab.keystream
|       class-attributes.tab.keystream.len
```

| | | | class-attributes.tab.len
| | | | class-attributes.tab.values.at
| | | | class-attributes.tab_i
| | | | class-attributes.tab_i.len
| | | | class-fq-name-to-source.tab
| | | | class-fq-name-to-source.tab.keystream
| | | | class-fq-name-to-source.tab.keystream.len
| | | | class-fq-name-to-source.tab.len
| | | | class-fq-name-to-source.tab.values.at
| | | | class-fq-name-to-source.tab_i
| | | | class-fq-name-to-source.tab_i.len
| | | | internal-name-to-source.tab
| | | | internal-name-to-source.tab.keystream
| | | | internal-name-to-source.tab.keystream.len
| | | | internal-name-to-source.tab.len
| | | | internal-name-to-source.tab.values.at
| | | | internal-name-to-source.tab_i
| | | | internal-name-to-source.tab_i.len
| | | | java-sources-proto-map.tab
| | | | java-sources-proto-map.tab.keystream
| | | | java-sources-proto-map.tab.keystream.len
| | | | java-sources-proto-map.tab.len
| | | | java-sources-proto-map.tab.values.at
| | | | java-sources-proto-map.tab_i
| | | | java-sources-proto-map.tab_i.len
| | | | package-parts.tab
| | | | package-parts.tab.keystream
| | | | package-parts.tab.keystream.len
| | | | package-parts.tab.len
| | | | package-parts.tab.values.at
| | | | package-parts.tab_i
| | | | package-parts.tab_i.len
| | | | proto.tab
| | | | proto.tab.keystream
| | | | proto.tab.keystream.len
| | | | proto.tab.len
| | | | proto.tab.values.at
| | | | proto.tab_i
| | | | proto.tab_i.len
| | | | source-to-classes.tab
| | | | source-to-classes.tab.keystream
| | | | source-to-classes.tab.keystream.len
| | | | source-to-classes.tab.len
| | | | source-to-classes.tab.values.at
| | | | source-to-classes.tab_i
| | | | source-to-classes.tab_i.len
| | | | subtypes.tab
| | | | subtypes.tab.keystream
| | | | subtypes.tab.keystream.len
| | | | subtypes.tab.len
| | | | subtypes.tab.values.at
| | | | subtypes.tab_i

```
| | | | |     subtypes.tab_i.len  
| | | | |     supertypes.tab  
| | | | |     supertypes.tab.keystream  
| | | | |     supertypes.tab.keystream.len  
| | | | |     supertypes.tab.len  
| | | | |     supertypes.tab.values.at  
| | | | |     supertypes.tab_i  
| | | | |     supertypes.tab_i.len  
| | | |  
| | | | ---lookups  
| | | | |     counters.tab  
| | | | |     file-to-id.tab  
| | | | |     file-to-id.tab.keystream  
| | | | |     file-to-id.tab.keystream.len  
| | | | |     file-to-id.tab.len  
| | | | |     file-to-id.tab.values.at  
| | | | |     file-to-id.tab_i  
| | | | |     file-to-id.tab_i.len  
| | | | |     id-to-file.tab  
| | | | |     id-to-file.tab.keystream  
| | | | |     id-to-file.tab.keystream.len  
| | | | |     id-to-file.tab.len  
| | | | |     id-to-file.tab.values.at  
| | | | |     id-to-file.tab_i  
| | | | |     id-to-file.tab_i.len  
| | | | |     lookups.tab  
| | | | |     lookups.tab.keystream  
| | | | |     lookups.tab.keystream.len  
| | | | |     lookups.tab.len  
| | | | |     lookups.tab.values.at  
| | | | |     lookups.tab_i  
| | | | |     lookups.tab_i.len  
| | | |  
| | | | ---resources  
| | | |   ---main  
| | | | |     logback.xml  
| | | |  
| | | | ---src  
| | | |   +---main  
| | | |   |   +---kotlin  
| | | |   |   |   |     Env.kt  
| | | |   |   |  
| | | |   |   ---bakaBotSrc  
| | | |   |   |   |     App.kt  
| | | |   |   |  
| | | |   +---app  
| | | |   |   |     Bot.kt  
| | | |   |   |     CommandRouter.kt  
| | | |   |   |     Involker.kt  
| | | |   |  
| | | |   +---docs  
| | | |   |   ---UniClasses
```

```
|   |   |   |   ProjDiscordBot.md
|   |   |   |
|   |   |   ---resource
|   |   |       An lise de dominio.drawio.png
|   |   |       BotDiscord-Modelo de desenho.drawio.png
|   |   |       BotDiscord-Modelo de dominio.drawio (1).png
|   |   |       BotDiscord-Modelo de dominio.drawio.png
|   |   |       BotDiscord-Modulo4-An lise de dominio.Conceito.png
|   |   |       BotDiscord-Modulo4-An lise de dominio.drawio.png
|   |   |       BotDiscord-Modulo4-An lise de dominio.Objetos.png
|   |   |       LeiDeDemeter.png
|   |   |       MQB_Icon.jpg
|   |   |       regrasDeNeg cio.png
|   |   |
|   |   +---integration
|   |   |   +---list
|   |   |       Anilist.kt
|   |   |       Mal.kt
|   |   |
|   |   |   ---music
|   |   |       Spotify.kt
|   |   |
|   |   +---locale
|   |   |       EnUs.kt
|   |   |       Lang.kt
|   |   |       PtBr.kt
|   |   |
|   |   +---model
|   |   |       AudioManager.kt
|   |   |       DaoWrapper.kt
|   |   |       Identifier.kt
|   |   |       MessageManager.kt
|   |   |
|   |   +---module
|   |   |       SignUpDao.kt
|   |   |
|   |   ---service
|   |   |   +---command
|   |   |       Help.kt
|   |   |       PlayAudio.kt
|   |   |       SignUp.kt
|   |   |       SongGame.kt
|   |   |
|   |   |   ---process
|   |   |       Log.kt
|   |   |
|   |   ---resources
|   |       logback.xml
|   |
|   ---test
|   ---kotlin
|       ---bakaBotSrc
```

```
|          AppTest.kt  
|  
+---build  
|   ---kotlin  
|       ---sessions  
---gradle  
    ---wrapper  
        gradle-wrapper.jar  
        gradle-wrapper.properties
```

4.Implementação:

4.1. Objetos de Implementação:

[Não sabemos]

4.2. Testes:

Os testes foram feitos na mão tanto pelo desenvolvedor da tarefa, quanto os outros integrantes, por uma falta de tempo e medo pelo desconhecimento dos limites da API do Discord nós acabamos não implementando testes únicos que a estrutura do gradle nos possibilita fazer.

5. Conclusões:

- **Kahao Yu:**

Apesar de não conseguirmos concretizar o projeto em toda a sua essência, por diversas dificuldades, o presente resultado foi de certa forma satisfatória, levando em conta aspectos como aprendizado.

- **Leonardo de Jesus Diz Conde:**

O projeto foi legal de desenvolver esse projeto, mesmo ficando longe do esperado, foi escalarecedor passar por todos esses processos, estava bem mais claro os passos quando comparado o desenvolvimento do app de flashcard no semestre passado.

- **Marcelo Maués Botelho de Souza:**

Fiquei mais na parte teórica, mas me encomodou a quantidade de pastas que arquivo gradle gera logo de inicio.

- **Victor Yuji Saito:**

A impressão que me ficou mediante ao trabalho é que para implementar algo próximo dos requisitos com o tempo que cada um de nós tinha disponível, teria sido muito melhor ter implementado em python que é mais simples e tem muito mais código de apoio na internet.