

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Mariano Dorneles de Freitas

**CLASS SERVICES: DESENVOLVIMENTO DE UM BOT PARA
AMBIENTES DE APRENDIZAGEM NA FERRAMENTA DISCORD**

Santa Maria, RS
2021

Mariano Dorneles de Freitas

CLASS SERVICES: DESENVOLVIMENTO DE UM BOT PARA AMBIENTES DE APRENDIZAGEM NA FERRAMENTA DISCORD

Trabalho Final de Graduação apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**.


ORIENTADORA: Prof.^a Andrea Schwertner Charão

Mariano Dorneles de Freitas

CLASS SERVICES: DESENVOLVIMENTO DE UM BOT PARA AMBIENTES DE APRENDIZAGEM NA FERRAMENTA DISCORD

Trabalho Final de Graduação apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**.

Aprovado em 9 de fevereiro de 2021:



Andrea Schwertner Charão, Dra. (UFSM)
(Presidenta/Orientadora)



Roseclea Duarte Medina, Dra. (UFSM)



Joao Vicente Ferreira Lima, Dr. (UFSM)

RESUMO

CLASS SERVICES: DESENVOLVIMENTO DE UM BOT PARA AMBIENTES DE APRENDIZAGEM NA FERRAMENTA DISCORD

AUTOR: Mariano Dorneles de Freitas

ORIENTADORA: Andrea Schwertner Charão

Diante do contexto do ensino remoto emergencial provocado pela pandemia do novo coronavírus, os ambientes virtuais de aprendizagem (AVA) se tornaram uma alternativa à sala de aula. Entretanto, há relatos que demonstram que estes ambientes nem sempre são a melhor opção e que, em alguns casos, a escolha de um ambiente em que os alunos possuam familiaridade é vantajoso para o processo de aprendizado. Com isso em mente, explorou-se a oportunidade de utilizar Discord, uma plataforma de comunicação popular entre os jovens e gratuita, como ambiente de sala de aula. A partir de uma experiência inicial, notou-se a falta de algumas funcionalidades, presentes em AVAs, que auxiliam o trabalho do professor no ambiente nativo da plataforma. Este trabalho relata a criação de um bot para ambientes de aprendizagem, que auxilia os professores nesse contexto, juntamente com um dashboard para melhor visualização de informações úteis. Ao final da implementação, o bot foi posto a testes com usuários que faziam parte do público-alvo e foram coletados dados para fins de melhorias. Os testes apontaram que o bot cumpriu de forma eficaz sua proposta, entretanto, ainda há espaço para melhorias.

Palavras-chave: Bot. Ambientes de aprendizagem. Discord. Dashboard.

ABSTRACT

CLASS SERVICES: A BOT DEVELOPMENT FOR LEARNING ENVIRONMENTS IN THE DISCORD TOOL

AUTHOR: Mariano Dorneles de Freitas

ADVISOR: Andrea Schwertner Charão

Due to emergency remote education context caused by the new coronavirus pandemic, the virtual learning environments (VLE) became an alternative to classrooms. However, there are reports that demonstrates that these environments are not always the best option and, in some cases, the choice of an environment which students have familiarity is beneficial for the learning process. Considering this, it was explored the opportunity to use Discord, a free and popular communication platform among young people, as a classroom. From an initial experience, it was noticed a lack of some functionalities usually available in VLEs, which assists the teacher's job in the platform's native environment. This work reports the creation of a bot to learning environments, that assists the teachers in this context, along with a dashboard for a better visualisation of useful information. At the end of the implementation, the bot was put through tests with target audience user's and data was collected for improvement purposes. The tests indicates that the bot effectively accomplished it's purpose, however, there is space to improvements.

Keywords: Bot. Learning environments. Discord. Dashboard.

LISTA DE FIGURAS

Figura 2.1 – Captura de tela da página de um servidor do Slack.	11
Figura 2.2 – Captura de tela da página inicial do Discord: (1) Aba com os servidores, (2) Histórico de conversas e grupos, (3) Lista de amigos e (4) Atividade atual dos amigos.	11
Figura 2.3 – Captura de tela da página de um servidor do Discord: (1) Canais e categorias, (2) Histórico do bate-papo e (3) Lista de membros do servidor. ..	12
Figura 2.4 – Página de gerenciamento de cargos, no menu de servidores do Discord.	13
Figura 2.5 – Início da lista de permissões que podem ser atribuídas aos cargos.	13
Figura 4.1 – Quadro com os comandos disponíveis no bot.	18
Figura 4.2 – Exemplo de uso do comando 'cs-help'.	19
Figura 4.3 – Exemplo de uso incorreto do comando "cs-userla".	19
Figura 4.4 – Exemplo de uso do comando "cs-userla".	20
Figura 4.5 – Exemplo de uso do comando "cs-usercount".	20
Figura 4.6 – Exemplo de uso do comando "cs-setlang".	21
Figura 4.7 – Exemplo de uso do comando "cs-settz".	21
Figura 4.8 – Exemplo de uso do comandos "cs-setname"e "cs-getnames".	22
Figura 4.9 – Exemplo de uso do comandos "cs-setrole".	23
Figura 4.10 – Tela inicial do dashboard.	24
Figura 4.11 – Tela de autorização do Discord.	24
Figura 4.12 – Tabela com a relação nome de usuário/nome real.	25
Figura 4.13 – Ranking de usuários mais ativos.	25
Figura 4.14 – Gráfico com o número de usuários no servidor.	26
Figura 5.1 – Gráfico com as respostas da pergunta 'Qual(is) das ferramentas de interação a seguir você já ouviu falar?'.	29
Figura 5.2 – Gráfico com as respostas da pergunta 'Qual(is) das ferramentas de interação a seguir você já utilizou em aulas síncronas?'.	30
Figura 5.3 – Gráfico com as respostas da pergunta 'Qual(is) características você considera essencial(ais) em um software de interação síncrona?'.	30
Figura 5.4 – Gráfico com as respostas para a pergunta 'Você conseguiu concluir a tarefa?', arranjado pela ordem das perguntas.	31
Figura 5.5 – Gráfico com o grau de concordância para a afirmação 'No geral, achei os comandos do bot (Class Services) fáceis de se utilizar'.	31
Figura 5.6 – Gráfico com o grau de concordância para a afirmação 'No geral, achei as tarefas (do pdf de testes) fáceis de se cumprir'.	32

SUMÁRIO

1	INTRODUÇÃO	6
1.1	OBJETIVO GERAL	7
1.2	CONTEXTO E JUSTIFICATIVA	7
2	REVISÃO DE LITERATURA	9
2.1	BOTS	9
2.2	PLATAFORMAS COM SUPORTE A BOTS	10
2.3	DISCORD	11
2.3.1	Servidores	12
2.3.2	Cargos e Categorias	12
2.3.3	Discord API	14
2.4	BOTS DO DISCORD	14
3	METODOLOGIA	15
4	PROJETO E IMPLEMENTAÇÃO	17
4.1	BOT	17
4.1.1	Exibir os comandos disponíveis	17
4.1.2	Mostrar última atividade dos usuários	18
4.1.3	Exibir a quantidade de usuários no servidor	19
4.1.4	Definir idioma e fuso horário	20
4.1.5	Criar relação entre nome de usuário e nome real	22
4.1.6	Determinar cargo para utilizar os comandos	22
4.2	DASHBOARD	23
4.3	HOSPEDAGEM	26
5	AVALIAÇÃO	27
5.1	TESTES COM USUÁRIOS	27
5.2	RESULTADOS E DISCUSSÃO	29
6	CONSIDERAÇÕES FINAIS	33
	REFERÊNCIAS BIBLIOGRÁFICAS	34

1 INTRODUÇÃO

A pandemia do novo coronavírus trouxe problemas variados para todos os setores da sociedades e exigiu adequação para adaptarem-se ao novo normal. Na educação, vídeo-chamadas substituíram o contato em sala de aula e os Ambientes Virtuais de Aprendizagem (AVA) serviram de mural para os professores se comunicarem com seus alunos. A interação entre professor e aluno é uma parte essencial do aprendizado (TIJIBOY; PEREIRA; WOICIECHOSKI, 2009), porém, a escolha de ferramenta também possui influência no processo (THOMS; ERYILMAZ, 2014).

Diante desse cenário, AVAs se popularizaram ainda mais, tornando-se a principal ferramenta para manter o vínculo aluno-instituição durante a pandemia. Estas plataformas costumam incorporar diferentes ferramentas de interação, tanto síncrona como assíncrona, embora se perceba uma ênfase nesta última. Por exemplo, no Google Classroom, que tornou-se popular (VYNCK; BERGEN, 2020) e é a escolha de diversas instituições públicas e privadas, a maior parte dos recursos se presta à interação assíncrona e compartilhamento de materiais com a turma, embora exista alguma integração com a plataforma Google Meet para facilitar encontros síncronos. Num cenário de ensino remoto emergencial, entendemos que a interação síncrona seja um importante recurso para construção de vínculos entre professores e alunos, porém não são todos os ambientes que possuem essa integração com uma ferramenta de interação síncrona audiovisual (GROSSI; MORAES; BRESCIA, 2013). Por outro lado, para além do contexto educacional, existem plataformas com forte ênfase nesse tipo de recurso. Assim, visando explorar outras ferramentas e possibilidades neste contexto, surgiu a oportunidade de utilizar a plataforma Discord.

O Discord é uma plataforma gratuita, com milhões de usuários e muito popular entre os jovens, principalmente na comunidade *gamer* (DISCORD, 2020a). No Discord é possível comunicar-se com outros usuários através de chamadas privadas ou em servidores e, nestes servidores, é possível adicionar *bots*, ou seja, robôs que auxiliam em tarefas diversas de organização, entretenimento, etc. O acesso gratuito possui alguns limites que são contornáveis através de um plano pago, mas com a ajuda dos *bots* os usuários conseguem aproveitar a ferramenta em diferentes contextos de interação, mesmo na sua versão gratuita (CITRON, 2020). Essa flexibilidade levou muitos usuários a adotar a plataforma em contextos educacionais, porém foi observado em experimentos anteriores (LACHER; BIEHL,) que há uma deficiência quando se utiliza a plataforma como ambiente de aprendizagem. De fato, os usuários não possuem qualquer tipo de visualização de estatísticas no cliente nativo, tampouco possuem ferramentas para auxiliá-los nos processos típicos da interação em uma sala de aula.

1.1 OBJETIVO GERAL

Este trabalho tem como objetivo a implementação de um bot para a ferramenta Discord, que auxilie professores a utilizar a ferramenta como alternativa para a interação síncrona em aulas à distância, provendo estatísticas e informações úteis. Também objetiva-se a criação de um dashboard para tornar a utilização do bot mais amigável e gráfica. Os objetivos específicos estão descritos a seguir:

- Investigar a utilidade de um *bot* ao utilizar a plataforma como sala de aula;
- Permitir o uso do *bot* para usuários;
- Coletar dados dos alunos a fim de gerar informações úteis aos professores;
- Implementar um dashboard para visualização das informações processadas;
- Auxiliar na utilização do Discord por parte do professor, tornando a transição de plataforma mais amigável;
- Criar um cenário de testes, com ambientes letivos, incentivando o uso do bot desenvolvido e coletando feedbacks dos usuários, a fim de refinar as funções do bot e implementar pequenas modificações, se sugeridas.

1.2 CONTEXTO E JUSTIFICATIVA

No início da pandemia do novo coronavírus, as instituições de ensino precisaram se adaptar ao ensino remoto de forma emergencial. Durante essa transição, notou-se que os ambientes tradicionais possuem alguns pontos negativos que impactam diretamente o processo de aprendizado dos alunos (THOMS; ERYILMAZ, 2014).

Perante o contexto apresentado, utilizar o Discord aparentava ser uma opção viável para interações síncronas. Uma vez que a ferramenta não foi desenvolvida com o propósito específico de ambiente de aprendizado, ela acaba por sofrer de algumas limitações, tais como a falta de uma forma de controle sobre quem está acessando, ou não, o servidor, e até mesmo visualizar de forma simples a quantidade de usuários no servidor. Entretanto, os *bots* conseguem suprir algumas dessas limitações. A documentação para desenvolvedores é bem completa (DISCORD, 2020b) e permite o desenvolvimento de uma ampla gama de comandos e fácil manipulação dos dados.

Durante o início do ensino remoto emergencial, o Discord foi utilizado dentro de uma disciplina complementar de graduação sob responsabilidade da orientadora deste trabalho e com a participação do autor orientado. Durante essa experiência, foi possível

observar na prática algumas dessas limitações. A plataforma disponibiliza, de forma nativa, a funcionalidade de cargos e um *log*, o que auxilia o professor no gerenciamento de sua sala de aula, porém, não há um histórico de uso, uma relação entre o nome de usuário e o nome real do aluno e mesmo estatísticas básicas disponíveis fora da API.

Com isso em mente, o presente trabalho propõe uma melhor visualização das estatísticas de uso, de modo a complementar o uso das ferramentas nativas por parte do professor, bem como auxiliar no contato inicial e intermediário do aluno com o professor, auxiliando este no período de familiarização na plataforma, de modo a tornar o ambiente de aprendizado mais descontraído e amigável e, então, mais propício ao estudo (TIJIBOY; PEREIRA; WOICIECHOSKI, 2009).

2 REVISÃO DE LITERATURA

O ensino à distância possui as suas vantagens e desvantagens (NONATO; PINTO, 2013), contudo, no período excepcional que a pandemia do novo coronavírus proporcionou, o ensino remoto emergencial tornou-se a única opção viável para a sequência dos anos letivos. Como já sabido, o encontro síncrono é essencial para o aprendizado no modelo de ensino à distância (MENDONÇA; GRUBER, 2019), uma vez que é nesse momento em que o professor pode interagir com os seus alunos (TIJIBOY; PEREIRA; WOICIECHOSKI, 2009). A partir deste cenário, entende-se que algumas plataformas não possuem todo o aparato desejado para abrigar um ambiente de aprendizagem, fazendo-se necessário utilizar ferramentas complementares. Neste capítulo, são apresentados trabalhos relacionados, plataformas consideradas para utilização no projeto e um aprofundamento dentro da plataforma e ferramentas escolhidas, considerando a contribuição dos mesmos para a concretização deste projeto.

2.1 BOTS

Bots são aplicações versáteis e que possuem diversas utilidades. Como apontam alguns autores (MURGIA et al., 2016), algumas destas aplicações são bem intencionadas e trazem avanços positivos aos usuários como, por exemplo, bots que monitoram bate-papos e protetores de spam, contudo, existem também os mal intencionados, como os invasores que tentam quebrar senhas de segurança e os que interagem como humanos em redes sociais.

Um de seus usos tem se tornado popular entre empresas de grande e médio porte: os chatbots (SHAWAR; ATWELL, 2007) – bots que conversam com o usuário – são utilizados para realizar o atendimento inicial ao cliente, de forma a poupar recursos humanos e agilizar a solução de problemas recorrentes (BRANDTZAEG; FØLSTAD, 2017), sem que o usuário necessite procurar em uma lista de *FAQ (Frequently Asked Questions)* – Perguntas Frequentes. Porém, algumas vezes estes bots não são tão eficazes e acabam provocando o efeito contrário (ADAM; WESSEL; BENLIAN, 2020). O autor destaca que a maneira com que o chatbot conduz a conversa influencia nos pedidos feitos aos usuários, de forma a reforçar que um usuário que se sente à vontade conversando com o bot é mais suscetível a acatar pedidos de feedbacks. Isso entra em sintonia com a tese levantada por alguns autores (ADAM; WESSEL; BENLIAN, 2020; MURGIA et al., 2016; ISHOWO-OLOKO et al., 2019; MOZAFARI; WEIGER; HAMMERSCHMIDT, 2020), que afirmam que bots que possuem um comportamento mais humano, ou se passam por um, obtêm mais sucesso em suas tentativas de interações.

No contexto educacional, os chatbots são um modelo popular (BARBALHO, 2017; ANDRADE, 2012; SMUTNY; SCHREIBEROVA, 2020), sendo que a maior parte é focada para o ensino superior de Ciência da Computação (KUYVEN et al., 2018) e funciona em modelo básico de “Perguntas e Respostas”, alguns atuando como uma variante de um buscador, outros como um quiz. Dentro da literatura buscada, não se encontrou registros acadêmicos de bots auxiliares ou de administração de turma.

Alguns autores também alertam sobre os possíveis malefícios dos bots quando utilizados de forma agressiva em redes sociais (FERRARA et al., 2016), podendo afetar negativamente discussões e influenciar pessoas através de *fake news* (SHAO et al., 2017; BESSI; FERRARA, 2016). Nota-se que os bots, quando mascarados de pessoas reais – *social bots* (FERRARA et al., 2016) – acabam trazendo malefícios ao ambiente em que se encontram e enganando usuários menos informados, de forma a moldar suas opiniões através destas estratégias. Por consequência, tais ações fazem com que os administradores das redes sociais tomem medidas para mitigar o dano causado por estes bots, limitando suas possíveis interações.

2.2 PLATAFORMAS COM SUPORTE A BOTS

Existem diversas plataformas que possuem suporte a bots como, por exemplo, o aplicativo Telegram, o Facebook Messenger e a plataforma Slack, além do próprio Discord que é foco deste trabalho. Dentre esses, pode-se dizer que o Slack (Figura 2.1) é o que mais se aproxima do Discord. Focado em comunicação empresarial, possui diversos recursos interessantes, como, por exemplo, a possibilidade de criar um canal coletivo com outras empresas/corporações que possuem utilizem o programa, além de diversas integrações com apps como Google Calendar (aplicativo de calendário) e Zoom (aplicativo de comunicação). Apesar de possuir uma versão gratuita, esta apresenta diversas limitações como permitir somente a inserção de 10 apps (equivalentes aos bots do Discord) no canal, assim como limitar a quantidade de mensagens que podem ser consultadas no canal. Outro ponto negativo no ponto de vista educacional é o fato do Slack não possuir canais de voz para comunicações de voz síncrona. Para executar esta tarefa é recomendada a utilização da integração com o Zoom (SLACK, 2020a), que, por sua vez, possui as suas limitações na versão gratuita. Embora exista um desconto na versão paga para propósitos educacionais, não há gratuidade (SLACK, 2020b). Entretanto, os apps possuem uma integração maior com o Slack quando comparados aos bots do Discord atualmente (NELLY, 2020).



Figura 2.1 – Captura de tela da página de um servidor do Slack.

2.3 DISCORD

Conforme já mencionado, a plataforma escolhida para o presente trabalho foi o Discord¹. O Discord é uma plataforma de comunicação de voz através de IP (VoIP), que possui uma versão gratuita e uma versão paga. O Discord oferece versões para computadores (Windows, MacOS e Linux), para dispositivos móveis (Android e iOS) e uma versão web. O Discord permite que um usuário adicione outro como amigo e também ingresse em servidores, em um máximo de 100 servidores. A versão paga do Discord retira algumas das limitações presentes na versão gratuita, porém, este trabalho não abordará nenhum recurso desta versão.

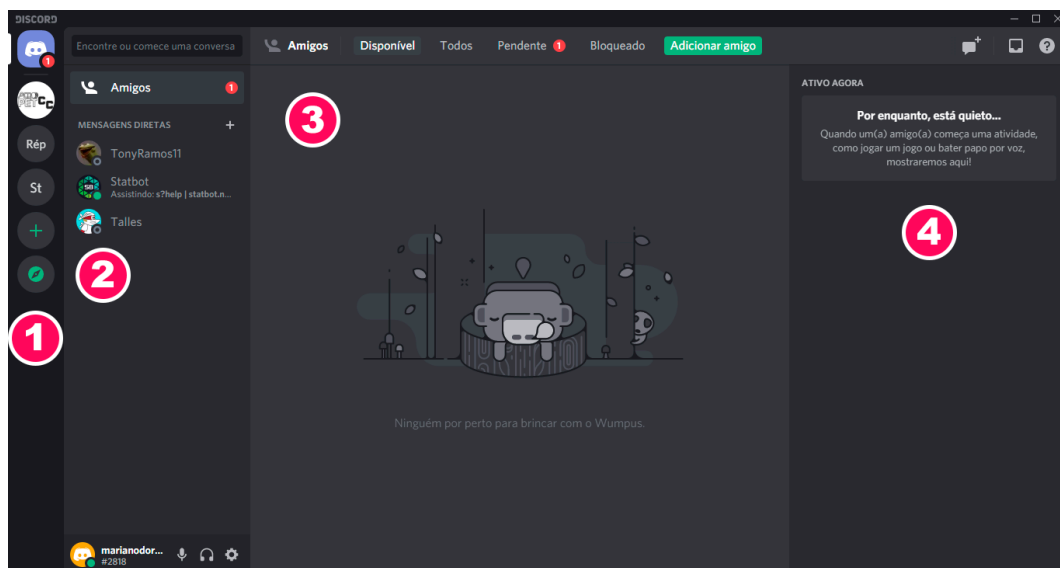


Figura 2.2 – Captura de tela da página inicial do Discord: (1) Aba com os servidores, (2) Histórico de conversas e grupos, (3) Lista de amigos e (4) Atividade atual dos amigos.

¹<https://discord.com/>

2.3.1 Servidores

A maior parte das interações feitas na plataforma ocorre através dos servidores. Dentro de um servidor, é possível comunicar-se através de voz ou texto, além de compartilhar tela, enviar arquivos – limitados a 8 MB na versão gratuita – e transmitir vídeo de câmera. Os servidores também possuem ferramentas para auxiliar no seu gerenciamento, os cargos e as categorias. A figura 2.3 representa uma captura de tela de um servidor. No item 1 da figura, observa-se a organização de canais e categorias, o item 2 representa o bate-papo de texto do canal “bem-vindo” e o item 3 mostra os usuários conectados ao servidor, separados por categorias. É possível observar a etiqueta de “Bot”, atribuída automaticamente pela plataforma, indicando se o usuário é um robô ou uma pessoa.

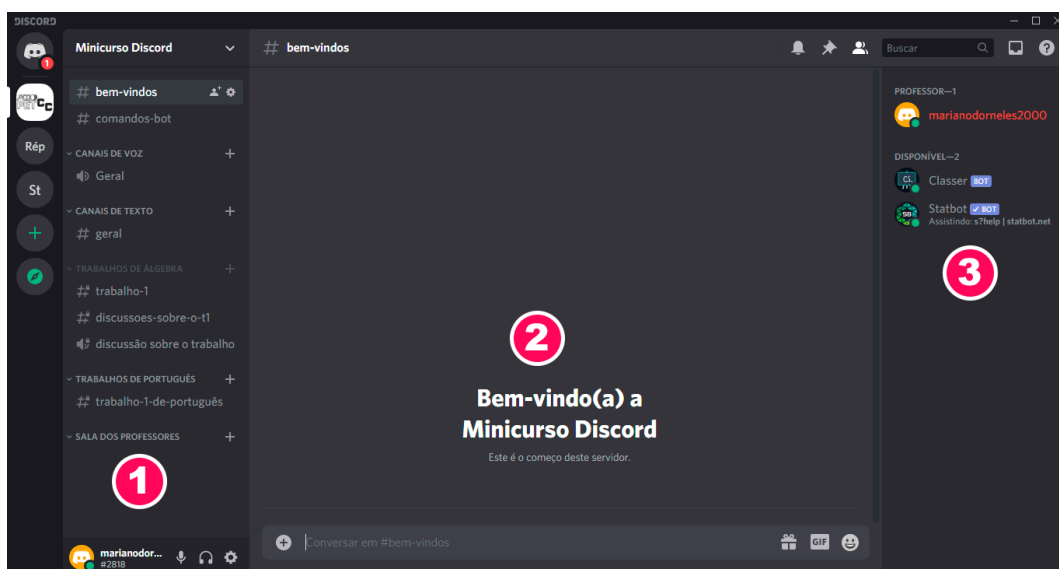


Figura 2.3 – Captura de tela da página de um servidor do Discord: (1) Canais e categorias, (2) Histórico do bate-papo e (3) Lista de membros do servidor.

2.3.2 Cargos e Categorias

Os cargos permitem que um administrador gerencie as ações que um usuário pode executar dentro do servidor. É possível nomeá-los e ajustar as permissões e, ao atribuir um cargo a um usuário, suas permissões são ajustadas de acordo com o cargo. É possível, também, criar canais de texto e voz restritos aos cargos. Desta forma, um administrador pode limitar o conteúdo disponível a cada usuário. As figuras 2.4 e 2.5 são capturas de tela da página de gerenciamento de cargos. A primeira mostra a página inicial do gerenciamento, que conta com uma lista de dos cargos já criados. É possível, também, criar uma hierarquia entre os cargos. Ao selecionar um cargo, é possível ver e gerenciar as permissões a ele atribuídas. A segunda figura demonstra alguns tipos de permissões disponíveis.

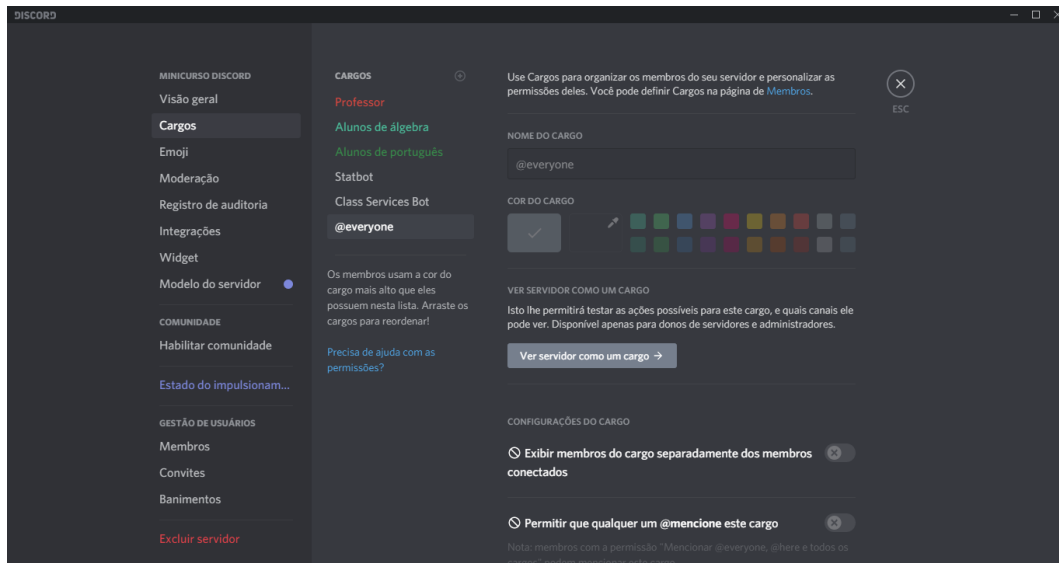


Figura 2.4 – Página de gerenciamento de cargos, no menu de servidores do Discord.

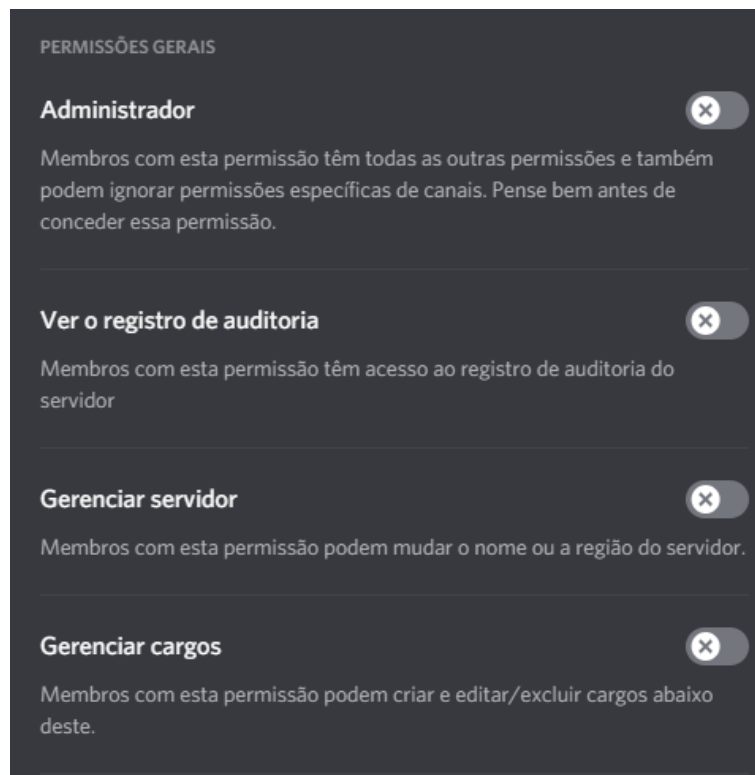


Figura 2.5 – Início da lista de permissões que podem ser atribuídas aos cargos.

Assim como os cargos, categorias servem para organizar os servidores. Elas agrupam canais de voz e texto em espécies de pastas e também podem receber as restrições dos cargos.

2.3.3 Discord API

O Discord disponibiliza uma API (DISCORD, 2020b), que conta com instruções variadas e que retornam informações dos servidores e dos usuários. Muitas destas informações não são disponibilizadas nativamente na plataforma e outras têm seu acesso facilitado através de comandos simples. A API do Discord já está implementada em pacotes de algumas linguagens, como é o caso do Python e o Discord.py². O pacote implementa todas as funções disponibilizadas pela API usando uma implementação assíncrona.

2.4 BOTS DO DISCORD

No início deste trabalho, realizou-se uma investigação em torno dos bots para o Discord, com a finalidade de entender quais eram as ferramentas que já estavam implementadas e estudar as possibilidades que a plataforma disponibilizava para esse tipo de aplicação. Foram investigados diversos bots, sendo que os principais estão listados abaixo:

- Carl – Focado em gerenciamento de servidor, permite um maior controle entre os cargos e os logs do servidor.
- Classroom Bot - Uma opção para uma sala de aula com o Discord, contudo, está disponível apenas em inglês e suas funções, apesar de interessantes, demonstram-se pouco úteis em um cenário real de sala de aula online, pois visa limitar a comunicação aluno-professor, ao invés de incentivá-la.
- Apolo – Apresenta uma integração de calendário, com funções de agendamento de eventos e notificações no privado.
- StatBot – Implementa uma série de estatísticas do servidor e disponibiliza também em um dashboard.

²<https://discordpy.readthedocs.io/en/latest/index.html>

3 METODOLOGIA

O presente trabalho seguiu uma abordagem de pesquisa exploratória, focada em estudo de caso (VENTURA, 2007). Nas experiências anteriormente relatadas, notou-se a possibilidade de melhorar a utilização da plataforma Discord para o uso de sala de aula através de *bots* auxiliares. No pressuposto da existência de bibliotecas já implementadas da API do Discord e que esta API poderia entrar com as ferramentas necessárias para se tornar o facilitador, optou-se por dar início ao projeto.

A linguagem escolhida para o desenvolvimento do *bot* foi Python, por preferência do autor, e a plataforma alvo foi o Discord. O início da programação foi pensado para auxiliar os professores na gestão de seus ambientes letivos, disponibilizando algumas estatísticas e relações que poderiam ser úteis ao ambiente desejado, ferramentas de controle de sala de aula, visando uma possível intervenção do professor baseado nas estatísticas disponibilizadas, com isto em mente, os primeiros comandos do bot foram concebidos. O comando 'cs-userla' permite que o professor veja a data da última interação de seus alunos no chat, pensado para que os professores tenham mais controle sobre quem está participando das aulas síncronas. Os outros comandos também seguiram a linha de facilitar o controle dos usuários dentro do Discord, aspirando trazer à plataforma informações comuns em outros ambientes de aprendizado tornando-a, assim, mais chamativa a quem deseja utilizá-la.

Alguns comandos, como, por exemplo, definir o fuso horário de resposta, demandaram que o bot guardasse informações para consultas futuras, dessa forma, pensando na baixa complexidade dos dados que seriam armazenados, optou-se por utilizar um sistema de arquivos XML, manipulando os dados com a biblioteca ElementTree XML. Após a implementação dos comandos básicos, o programa foi readaptado visando sua disponibilidade para a maior quantidade de usuários. Assim, foi feita uma refatoração para que suportasse diferentes línguas, de forma que futuramente a comunidade possa contribuir com as traduções da forma mais simples possível.

O bot também sofreu adaptações após buscas na área de interação com usuários. As respostas aos comandos são feitas de forma mais pessoal, utilizando, inclusive, primeira pessoa, uma vez que estudos constataam que bots que interagem de maneira mais humana, tendem a receber respostas mais positivas. Outra modificação foi em nome, de 'Class Services', passou-se a utilizar o pseudônimo 'Classer' - junção das palavras 'Class' e 'Services' - no intuito de facilitar sua pronúncia e aproximá-lo de um nome próprio. Com o fim da implementação, o bot foi hospedado em um servidor na nuvem e ficou disponível para o público.

Com o bot disponível, foram realizados testes de aceitação informal, durante cinco dias, com a base de usuários alvo: professores que realizam encontros síncronos durante o ensino remoto. Para isso, recorreu-se aos professores que já haviam realizado um mi-

nicurso sobre o Discord. Estes professores lecionam nos ensinos fundamental e médio de escolas públicas de Santa Maria e possuem média familiaridade com ferramentas computacionais. Dos trinta concluintes do minicurso, oito participaram dos testes. Os testes seguiram um roteiro com sete tarefas para serem realizadas, sem monitoramento ou limite de tempo. Ao final dos testes, foi solicitado aos participantes que respondessem um formulário para coleta das informações. O formulário possuía duas seções, uma para traçar o perfil dos usuários e entender suas preferências de plataforma e outra para perguntas específicas sobre o bot e sua utilização. A primeira pergunta da segunda seção era para relatar o desempenho obtido durante o teste, com resposta booleana, enquanto as outras perguntas relacionadas ao bot foram avaliadas na escala Likert. Havia, também, uma pergunta discursiva para sugestões e retornos sobre o bot.

Durante a semana de testes, deu-se sequência ao desenvolvimento do dashboard. A linguagem escolhida também foi Python e, por conta de limitações na API do Discord, ele funciona de forma a integrar-se com o bot.

4 PROJETO E IMPLEMENTAÇÃO

Neste capítulo, é apresentado as decisões de projeto e ideias por trás dos comandos desenvolvidos para o bot, a descrição detalhada do desenvolvimento e um exemplo de uso de cada um. Também é relatado o processo de desenvolvimento do dashboard e apresentada a forma com que as métricas são obtidas.

4.1 BOT

Uma vez escolhida a linguagem de desenvolvimento para o bot, foi feito um estudo da API para que se pudesse entender as possibilidades e limitações que a plataforma dispunha para desenvolvedores. O estudo de bots similares auxiliou a entender alguns pontos e, durante o desenvolvimento, pode-se entender as decisões de projeto dos outros bots. Após o estudo de quais ferramentas poderiam ser úteis para o ambiente, fez-se um planejamento para verificar a melhor forma de implementação de cada um dos comandos. Todos os comandos utilizam um padrão assíncrono de implementação, o usuário invoca o comando através de palavras-chave e o bot, que é chamado através de seu prefixo, executa o comando definido e retorna uma resposta, também assíncrona, ao servidor. A definição do código para cada comando do bot é dada através de decorators. A Figura 4.1 retrata a ideia por trás de cada um dos comandos. O código-fonte do bot está disponível no GitHub¹ e as motivações por trás de cada um dos comandos, bem como a descrição de como foi implementado e um exemplo de uso estão detalhados nas subseções a seguir:

4.1.1 Exibir os comandos disponíveis

Uma das práticas comuns dos bots desenvolvidos para o Discord é possuir uma lista com todos os comandos disponíveis para uso e uma breve descrição, que mostre como usar cada um destes comandos e detalhando sua funcionalidade. Para que o bot responda a uma solicitação, é necessário utilizar um prefixo predeterminado juntamente com o comando desejado, no caso do Class Services, o prefixo definido foi 'cs-'. A descrição dos comandos é resgatada de um arquivo de strings em XML e então é exibido na tela uma mensagem com todos os comandos disponíveis. A figura 4.2 apresenta o uso do comando.

¹<https://github.com/marianod323/ClassServicesBot>

Comandos	Proposta	Forma de implementação
cs-help	Mostrar ao usuário uma lista de comandos disponíveis	Resgata-se as informações do arquivo XML e então exibe-se na tela
cs-userla	Mostra ao usuário a última atividade registrada dos outros participantes do servidor, em uma lista	A API não dispõe de forma nativa uma maneira de obter o fuso horário do servidor, por padrão, a última atividade do usuário é mostrada baseado no UTC. Utiliza-se o histórico de mensagens dos canais de textos
cs-usercount	Mostra o usuário a quantidade de membros presentes no servidor	É feita a varredura entre todos os membros e é feita a verificação de usuário é ou não um bot, salvando-se em um contador
cs-setlang	Permite que o usuário defina o idioma de resposta do bot	Para a implementação destes comandos, foi necessário manter uma persistência dos dados.
cs-settz	Permite que o usuário defina o fuso horário de resposta do bot	Optou-se por um sistema de arquivos devido à baixa complexidade dos dados e falta de relacionamentos no modelo criado. Desta forma, as informações são armazenadas em um XML e resgatadas quando necessário
cs-setname	Permite que um aluno forneça seu nome para o bot	Utilizando o banco de dados já criado, o elemento 'user' possui um atributo com o username e o valor com o nome real fornecido.
cs-setrole	Permite que um administrador do servidor defina o cargo mínimo para utilização de alguns dos comandos do bot	É salvo no arquivo do servidor o cargo mínimo que o administrador do servidor definiu para utilizar esses comandos. Foi adicionada uma verificação ao início de cada comando restrito e a comparação é feita através do nível do cargo do usuário que chamou o comando. É exibida uma mensagem de erro caso o usuário não possua permissão para executar o comando desejado.

Figura 4.1 – Quadro com os comandos disponíveis no bot.

4.1.2 Mostrar última atividade dos usuários

Uma vez que as interações em aulas síncronas à distância são parte fundamental do processo de ensino-aprendizagem (MENDONÇA; GRUBER, 2019), uma preocupação dos docentes é justamente saber quais alunos estão participando e quais não estão. Desta forma, o comando 'cs-userla' foi desenvolvido para informar ao professor a última atividade registrada dos membros do servidor no bate-papo de texto. Como a API não possui um registro de última atividade dos usuários, para implementar esse comando, realizou-se uma varredura nas últimas 150 mensagens de cada um dos canais de texto presentes no servidor em busca da última mensagem de cada um dos membros, utilizando uma busca linear simples. Por ser utilizada uma busca linear, a escalabilidade poderia se tornar um problema, contudo, como o ambiente de sala de aula normalmente não possui uma grande quantidade de alunos, foram feitos apenas alguns ajustes para otimização do código, como, por exemplo, a separação da exibição dos alunos em páginas de 10 usuários, o que permite

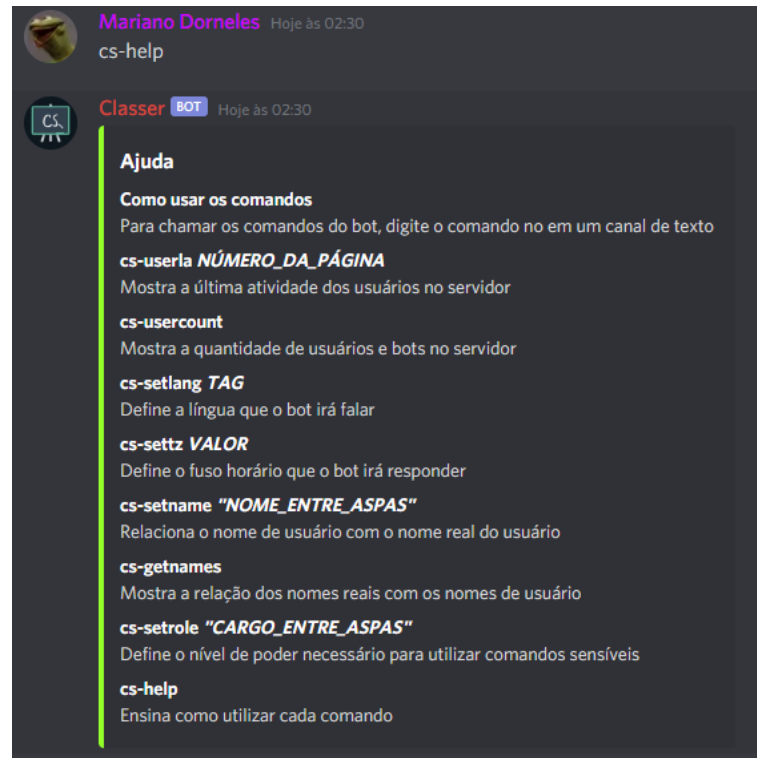


Figura 4.2 – Exemplo de uso do comando 'cs-help'.

que o processamento seja feito de forma gradual e a exibição de um resultado para o usuário não demande tanto tempo. O bot calcula o fuso horário da última atividade com base na informação que o usuário define através do comando 'cs-settz'. As figuras 4.3 e 4.4 demonstram o uso do comando.

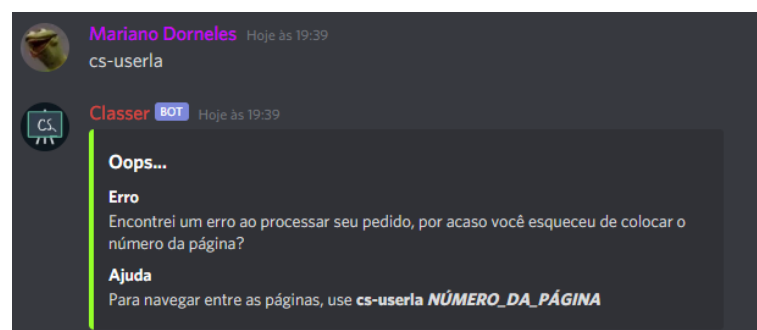


Figura 4.3 – Exemplo de uso incorreto do comando "cs-userla".

4.1.3 Exibir a quantidade de usuários no servidor

Na intenção de permitir um maior controle de quantos alunos já estão conectados ao servidor, foi criado um comando que exhibe a quantidade de usuários e bots, separadamente, presentes no servidor. Nativamente, o Discord disponibiliza uma contagem de

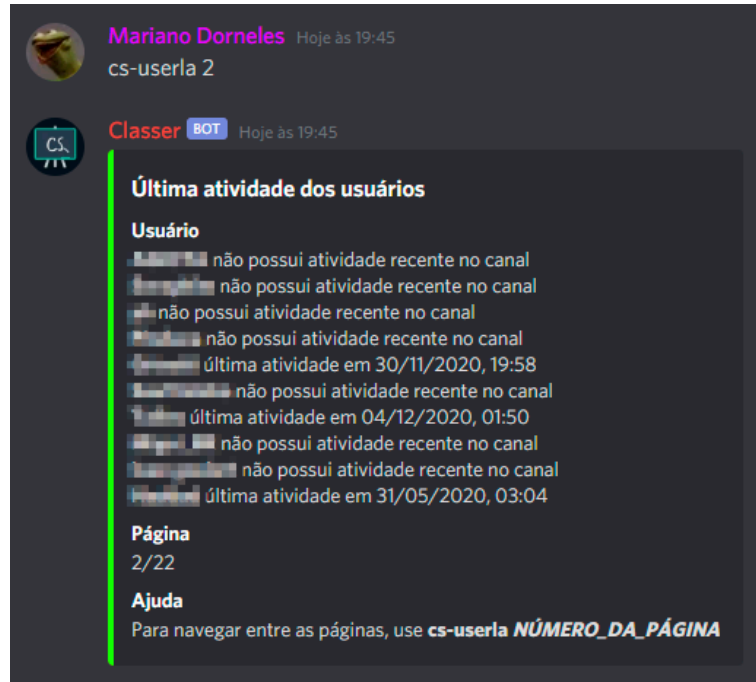


Figura 4.4 – Exemplo de uso do comando "cs-userla".

membros, porém, nesta contagem, não é feita a distinção entre um usuário humano e um robô. Este comando implementa uma varredura em todos membros do servidor e verifica se usuário é ou não um bot. Ao fazer a verificação, é feito o incremento em um contador do respectivo tipo e, logo após, exibe-se como demonstrado na Figura 4.5.

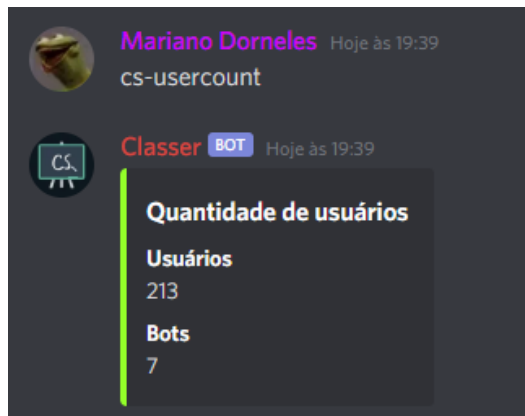


Figura 4.5 – Exemplo de uso do comando "cs-usercount".

4.1.4 Definir idioma e fuso horário

Para fins de internacionalização do bot e abranger o maior público possível dentro do público-alvo, foram criados comandos para alterar o idioma de resposta do bot (Figura 4.6) e o fuso horário em que as atividades são mostradas (Figura 4.7). Para que estes

comandos funcionassem, seria necessário uma forma de guardar as preferências de cada servidor, pensando nisso, foi criado um sistema de arquivos XML, onde cada servidor é identificado por um número em um arquivo desta base de dados. Foi desenvolvida também a classe 'PrefHandler' com os *getters* e *setters* de cada uma das preferências. O *getter* do idioma, por exemplo, retorna o acesso ao arquivo do idioma definido e é utilizado nos outros comandos para retornar as strings que cada um deles utiliza, através de uma expressão XPath.

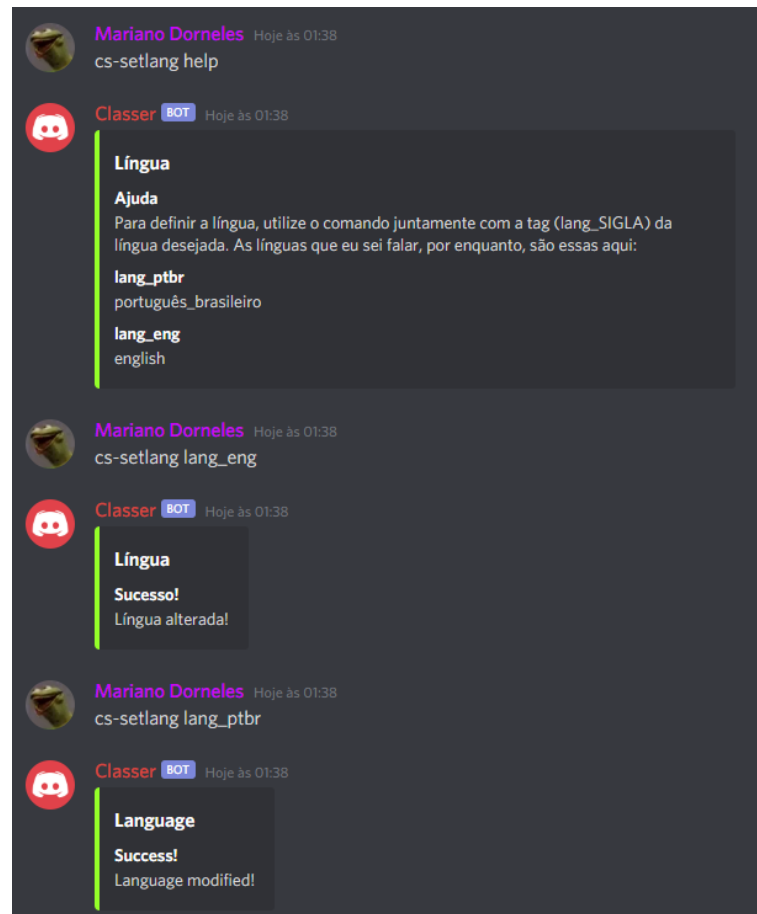


Figura 4.6 – Exemplo de uso do comando "cs-setlang".

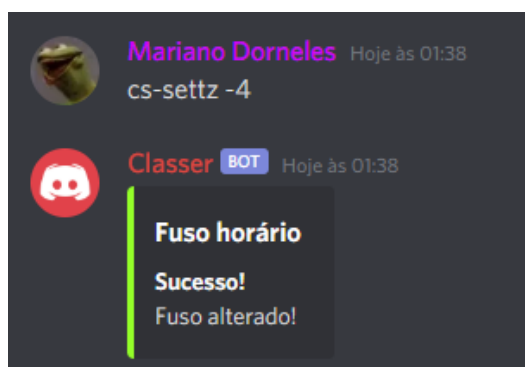


Figura 4.7 – Exemplo de uso do comando "cs-settz".

4.1.5 Criar relação entre nome de usuário e nome real

Dentro do Discord e em plataformas online é comum que os usuários possuam nomes de usuários diferentes de seus nomes reais, alguns, até mesmo não possuem ligação alguma com o nome real do usuário. Apesar de o Discord possuir uma ferramenta nativa para utilizar um apelido dentro de um servidor, a ideia do comando é justamente permitir que o usuário mantenha seu nome de usuário e apenas criar uma relação através do bot para controle do professor. Assim, foi desenvolvido os comandos 'cs-setname' e 'cs-getnames' (Figura 4.8). O primeiro relaciona o usuário que lançou o comando ao nome passado por parâmetro e registra essa atribuição no arquivo do servidor, dentro de um elemento "usuário" com um atributo contendo o nome de usuário e o valor sendo o nome real fornecido. O segundo, que pode ter seu uso limitado, retorna uma lista com a relação nome de usuário/nome real com todos os usuários que já registraram o seu nome real no bot.

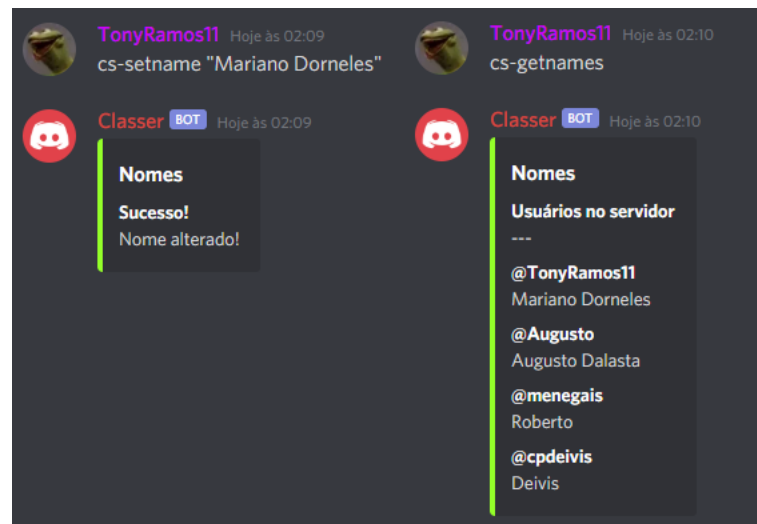


Figura 4.8 – Exemplo de uso do comandos "cs-setname" e "cs-getnames".

4.1.6 Determinar cargo para utilizar os comandos

Visto que alguns comandos possuem informações que podem ser sensíveis, isto é, que violariam a privacidade do aluno, pensou-se em uma forma de restringir o uso dos comandos utilizando a hierarquia de cargos dos Discord. Dessa forma, é possível restringir a utilização dos comandos (Figura 4.9) 'cs-userla' e 'cs-getnames' para o administrador do servidor e até um cargo, definido pelo administrador. Quando definido o cargo, é feita uma consulta prévia antes de executar cada um desses comandos e, caso o usuário não possua permissões, é exibida uma mensagem de erro no chat. O cargo é salvo no arquivo do servidor e é recuperado através do 'PrefHandler'.

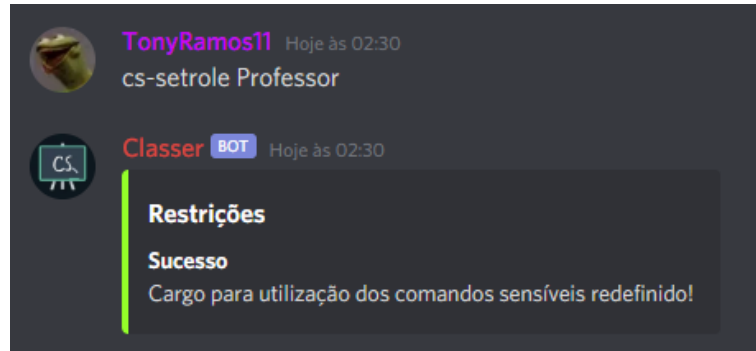


Figura 4.9 – Exemplo de uso do comandos "cs-setrole".

4.2 DASHBOARD

Com o objetivo de permitir uma visualização facilitada e gráfica por parte do professor, pensou-se no desenvolvimento de um dashboard. Nele, seria possível visualizar estatísticas de uso do servidor juntamente com tabelas mostrando os conteúdos de cada um dos comandos do bot, bem como administrar as preferências do bot através de uma interface gráfica. Entretanto, para aplicações que rodam fora do ambiente do Discord, a API limita-se bastante e bloqueia grande parte dos *endpoints* que disponibiliza para os bots. Por exemplo, não é possível acessar as mensagens dentro de um servidor através da API, se acessada fora do escopo de um bot.

Diante dessa situação, foi necessária uma adaptação na ideia inicial do dashboard. Para contornar essa limitação, precisou-se rodar o dashboard no mesmo processo que o bot, utilizando a biblioteca *threading* do python, e criou-se o comando 'cs-updatedashboard', que cria ou atualiza elementos que contém as informações do dashboard no arquivo XML do servidor que o invoca. Ao entrar no dashboard, essas informações são processadas e colocadas em um *dataframe* da biblioteca Pandas e, utilizando o framework 'Dash'², as informações são postas na tela. O framework possui integração com a biblioteca 'Plotly', que permite o uso de diversos estilos de gráficos e tabelas diferentes. Uma vez que essa limitação não era sabida ao início do projeto, o bot não foi projetado para coletar informações dos servidores periodicamente e manter um registro destas informações, por conta disso, a implementação final do dashboard ficou um pouco limitada.

Os comandos existentes escolhidos para visualização no dashboard foram o de contagem de usuários e relação de nome real com nome de usuário. Novamente, por decisões anteriores de projeto, optou-se por limitar os comandos a serem visualizados no dashboard, posto que as informações são guardadas em um sistema de arquivos simples e não em um banco de dados. Contudo, também foi desenvolvido uma estatística exclusiva para o dashboard, que é o ranking de usuários. O ranking ordena e mostra os dez usuários mais ativos nos canais de texto no intervalo das últimas 50 (cinquenta) mensagens de todos

²<https://dash.plotly.com/>

os canais.

Ao entrar no dashboard, o usuário depara com uma simples tela para realizar o login (Figura 4.10). O login é feito através do Discord (Figura 4.11), a implementação da autenticação foi feita utilizando o micro framework Flask, que facilita a criação de aplicações web.



Figura 4.10 – Tela inicial do dashboard.

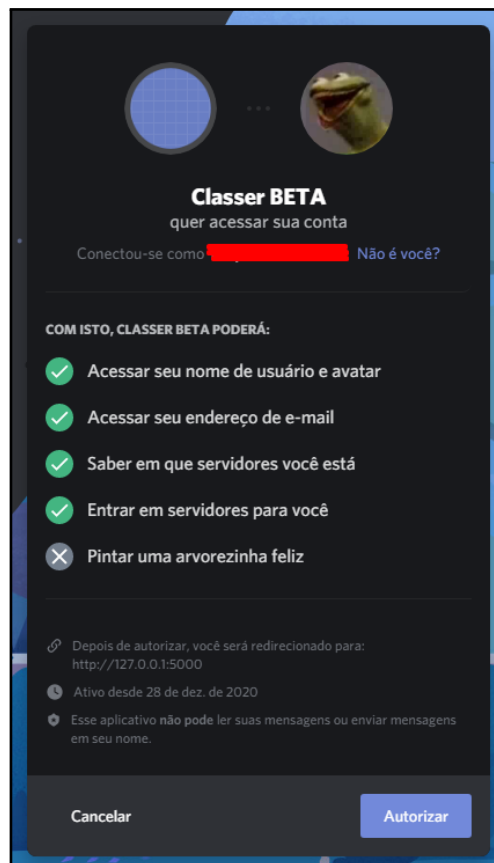


Figura 4.11 – Tela de autorização do Discord.

Após autenticar, o usuário é redirecionado para a página com as estatísticas de um dos servidores em que ele possui permissão de administrador e o bot está presente. Para exibir as informações, é necessário que o usuário execute pelo menos uma vez o comando 'cs-updatedashboard'. Após executar o comando, as estatísticas disponíveis são mostradas na tela. As Figuras 4.12, 4.13 e 4.14 representam capturas de tela do dashboard.

ClassServices - Dashboard

Nomes de usuário	Nomes reais
TonyRamos11	Mariano Dorneles
Augusto	Augusto Dalasta
menegais	Roberto
cpdeivis	Deivis

Figura 4.12 – Tabela com a relação nome de usuário/nome real.

Nomes de usuário	Número de mensagens
TonyRamos11	75
Augusto	25
René	25
schdck	18
cpdeivis	16
Lersch	14
«Sr.Anthony»	9
Fuhr	7
Dreyan	4
Talles	4

Ranking - Gráfico de pizza

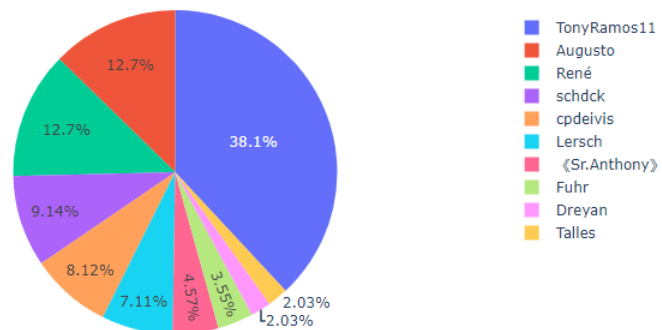


Figura 4.13 – Ranking de usuários mais ativos.

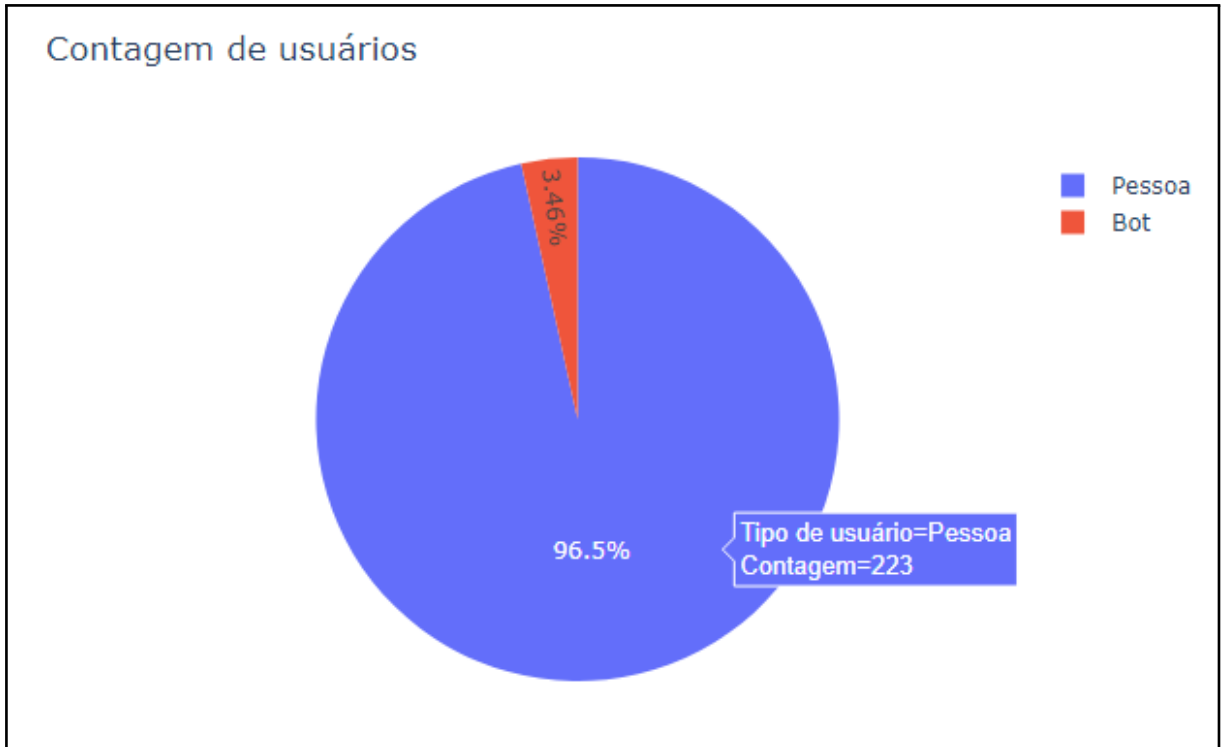


Figura 4.14 – Gráfico com o número de usuários no servidor.

4.3 HOSPEDAGEM

Após desenvolvido, as primeiras versões do bot foram inicialmente hospedadas em máquina local. Contudo, visando uma maior disponibilidade, buscou-se alternativas que permitissem uma hospedagem gratuita e eficaz. Desta forma, optou-se pela hospedagem da Heroku³. O servidor gratuito disponibilizado pela plataforma não possuía um *uptime*⁴ de 24 horas por dia, 7 dias por semana, assim, a alternativa buscada foi utilizar créditos que a plataforma provê aos estudantes do ensino superior. As atualizações foram liberadas conforme grandes mudanças ocorriam, visando um melhor gerenciamento de versões e estabilidade para o programa desenvolvido.

³<https://www.heroku.com/>

⁴Tempo de disponibilidade do serviço

5 AVALIAÇÃO

Ao final da implementação do bot, foram realizados testes para verificar a aceitação do público-alvo e então, através de um formulário, registrar o desempenho dentro do roteiro proposto. Neste capítulo, é detalhada a forma com que os testes aconteceram e é feita uma análise e discussão dos resultados.

5.1 TESTES COM USUÁRIOS

Os testes aplicados foram nos moldes de um teste de aceitação informal, focados na usabilidade da aplicação (FERREIRA; CORSO; SILVA, 2002) durante os dias 25 à 29 de janeiro de 2021. Como citado na metodologia, a base de usuários convidada para os testes foram professores que realizaram o Minicurso de Discord¹, oferecido pelo PET-CC (Programa de Educação Tutorial - Ciência da Computação) da UFSM, durante o mês de janeiro. Uma vez que, aprovados no minicurso, estes professores possuíam os conhecimentos necessários sobre a plataforma e o conhecimento básico sobre o que é e como funciona um bot. O minicurso teve cento e quatorze inscritos e trinta aprovados.

Foi criado um roteiro para os testes, que apresentava uma breve introdução da aplicação, um pequeno tutorial explicando o passo a passo de como adicionar um bot a um servidor, juntamente com o link de convite² do "Class Services" e uma sequência de tarefas a serem realizadas pelo voluntário.

As tarefas dos testes referiam-se a cada um dos comandos possíveis, com uma breve contextualização, sem informar explicitamente qual comando deveria ser utilizado, com exceção do comando de ajuda. As tarefas foram as seguintes:

- Execute o comando de ajuda (cs-help). (Se você possuir eventuais dúvidas, recorra a este comando);
- Você precisa saber quais alunos não estão comparecendo às aulas síncronas, para poder abordá-los. Confira se o fuso horário de resposta do bot está correto, se não estiver, defina para o fuso horário correto e solicite a última atividade dos usuários no chat;
- Os estagiários que auxiliam na sua turma querem ajudar você com o bot, porém, você não deseja lhes dar total poder (permissões de administrador). Defina um cargo que também pode utilizar os comandos sensíveis*;

¹<https://www.ufsm.br/pet/ciencia-da-computacao/2020/11/06/pet-cc-oferece-curso-de-discord/>

²<https://bit.ly/Classer>

- Alguns alunos de intercâmbio estão estudando em sua turma, altere a língua do bot para inglês e, depois, retorne para português;
- Alguns de seus alunos possuem um nome de usuário exótico e que não remete ao seu verdadeiro nome, você precisa que eles relacionem seus nomes com os nomes de usuário, mas antes, relacione o seu nome com o seu nome de usuário;
- Para poder fazer a chamada e preencher o seu diário de classe, você precisa saber o nome de seus alunos, para facilitar neste trabalho, solicite a lista de relação usuário/nome;
- Confira se todos os seus alunos já estão conectados ao servidor, para isso, solicite a quantidade de usuários no servidor.

Uma vez que o teste fora finalizado, solicitava-se uma resposta a um formulário de coleta. O formulário contava com as seguintes questões sobre o perfil dos candidatos:

- Qual(is) das ferramentas de interação a seguir você já ouviu falar?
- Qual(is) das ferramentas de interação a seguir você já utilizou em aulas síncronas?
- Qual o seu grau de familiaridade com o computador?
- Qual(is) características você considera essencial(ais) em um software de interação síncrona?
- Você utiliza/já utilizou o Discord em sala de aula?
- Você já utilizou bots no Discord?

Logo após estas questões, havia uma questão com resposta booleana, perguntando, para cada tarefa, se o voluntário conseguiu ou não cumprí-la.

Por fim, haviam mais três afirmações referentes à usabilidade do bot e era perguntado o quanto o voluntário concordava com cada uma das afirmações, baseado na escala Likert. Foi perguntado, também, se o voluntário possuiu alguma dificuldade durante o teste ou tinha alguma sugestão para o bot.

- No geral, achei os comandos do bot (Class Services) fáceis de se utilizar.
- No geral, achei as tarefas (do pdf de testes) fáceis de se cumprir.
- Gostei da forma com que o bot (Class Services) respondia aos meus comandos.

5.2 RESULTADOS E DISCUSSÃO

Oito participantes realizaram as tarefas propostas pelo roteiro de testes durante o período estipulado. A Figura 5.1 apresenta um gráfico de barras com as respostas à pergunta 'Qual(is) das ferramentas de interação a seguir você já ouviu falar?'. Os participantes, em sua totalidade, conheciam o Google Meet, Discord e Skype. Sete conheciam o Zoom, seis o Slack, três o BigBlueButton e apenas um conhecia o Jitsi. Já a Figura 5.2 mostra o gráfico com as respostas da pergunta 'Qual(is) das ferramentas de interação a seguir você já utilizou em aulas síncronas?'. Nota-se que todos os participantes já utilizaram o Google Meet e, apesar de todos conhecerem o Discord, apenas cinco chegaram a utilizar em alguma aula síncrona. Em relação ao grau de familiaridade com o computador, seis responderam com o valor máximo na escala de 1 (um) à 5 (cinco), os valores três e quatro obtiveram uma marcação cada.

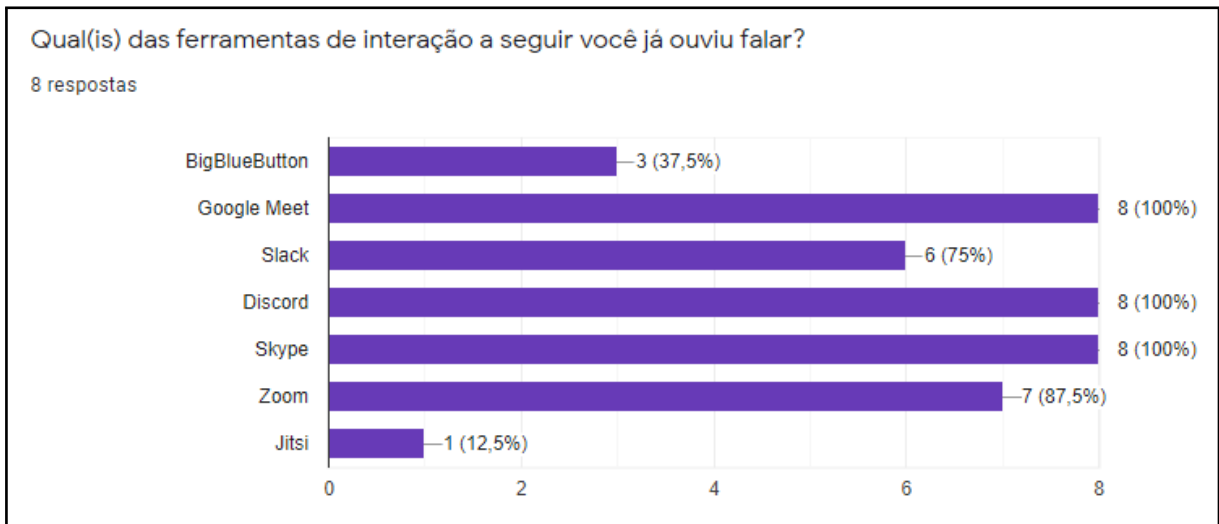


Figura 5.1 – Gráfico com as respostas da pergunta 'Qual(is) das ferramentas de interação a seguir você já ouviu falar?'.

Das características que os participantes julgaram ideais em um software para interação síncrona, a gratuidade é a maior delas, presente na resposta de todos os participantes da pesquisa. A disponibilidade em plataformas e a qualidade do áudio e vídeo também se destacam, como pode-se observar na Figura 5.3.

Quando perguntados a respeito da utilização de bots no Discord, seis participantes responderam que já utilizaram e dois responderam que não utilizaram. Contudo, como observado na Figura 5.4, quase todas as tarefas foram cumpridas por todos os participantes, com exceção da tarefa 3 - definir um cargo para o uso dos comandos, em que um participante não conseguiu concluir.

Mesmo conseguindo concluir todas ou grande parte das tarefas, alguns participantes não acharam os comandos fáceis de se utilizar e as tarefas fáceis de cumprir, como visto nas Figuras 5.5 e 5.6.

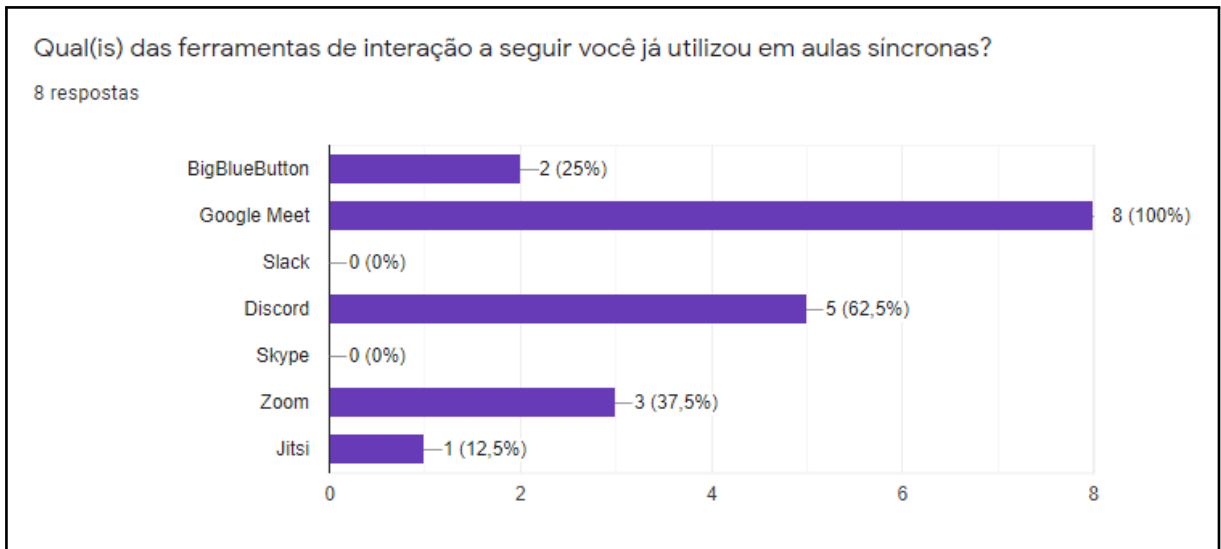


Figura 5.2 – Gráfico com as respostas da pergunta 'Qual(is) das ferramentas de interação a seguir você já utilizou em aulas síncronas?'.

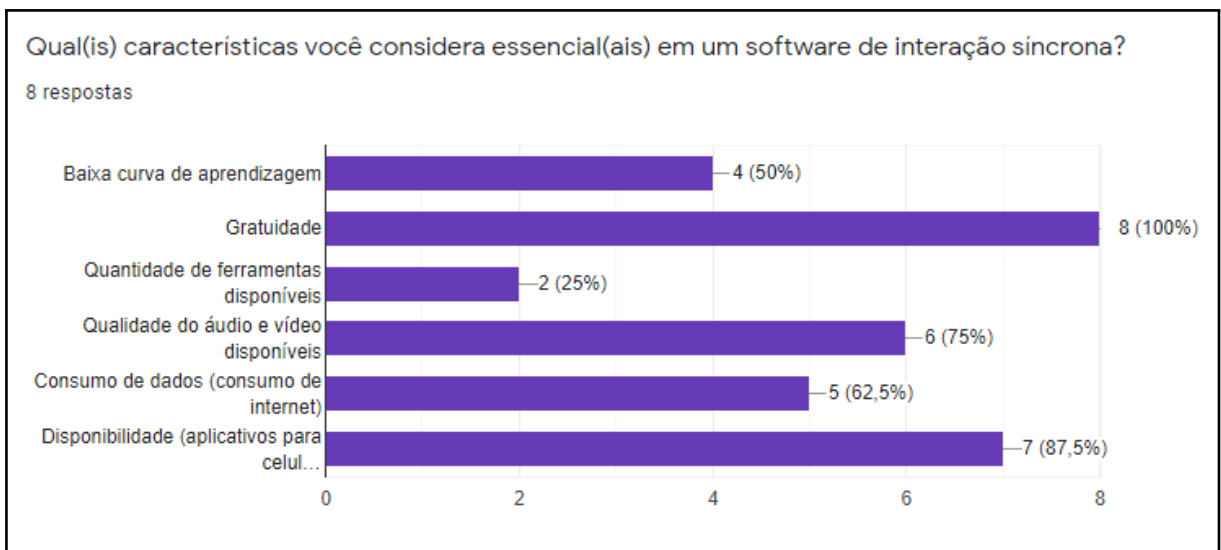


Figura 5.3 – Gráfico com as respostas da pergunta 'Qual(is) características você considera essencial(ais) em um software de interação síncrona?'.

A partir desses dados, pode-se observar que a disponibilidade e gratuidade do serviço é de extrema importância, uma vez que muitas escolas possuem recursos limitados para investir em sistemas e pacotes para esses fins ou simplesmente pela opção de não destinar recursos para uma solução que não será utilizada quando as aulas presenciais retornarem.

Mesmo que com alguma dificuldade, todos os participantes conseguiram concluir grande parte das tarefas propostas, o que solidifica uma das propostas do bot que é justamente auxiliar o professor a se familiarizar com Discord e complementar as lacunas que foram observadas.

Apesar do dashboard não ter sido disponibilizado durante os testes, acredita-se que

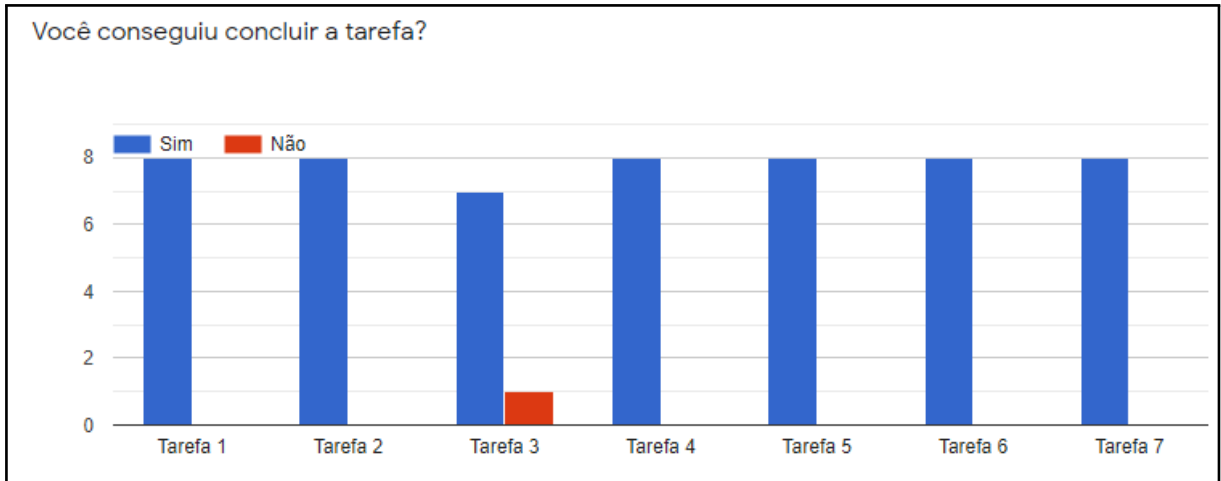


Figura 5.4 – Gráfico com as respostas para a pergunta 'Você conseguiu concluir a tarefa?', arranjado pela ordem das perguntas.

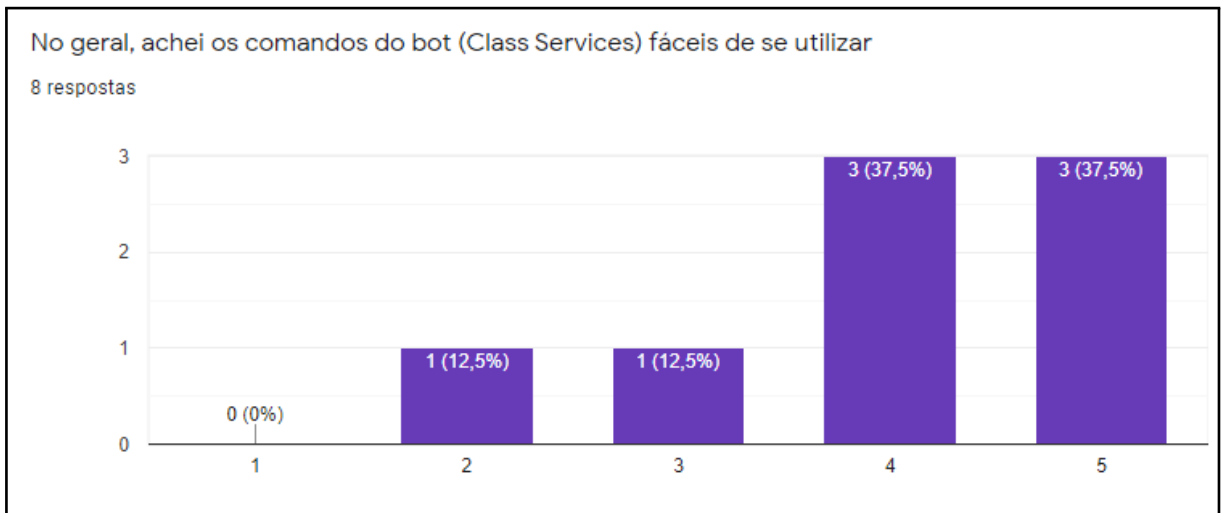


Figura 5.5 – Gráfico com o grau de concordância para a afirmação 'No geral, achei os comandos do bot (Class Services) fáceis de se utilizar'.

é uma ferramenta que complementa o uso do bot, de forma a abstrair parte da dificuldade do bot, que é executar os comandos.

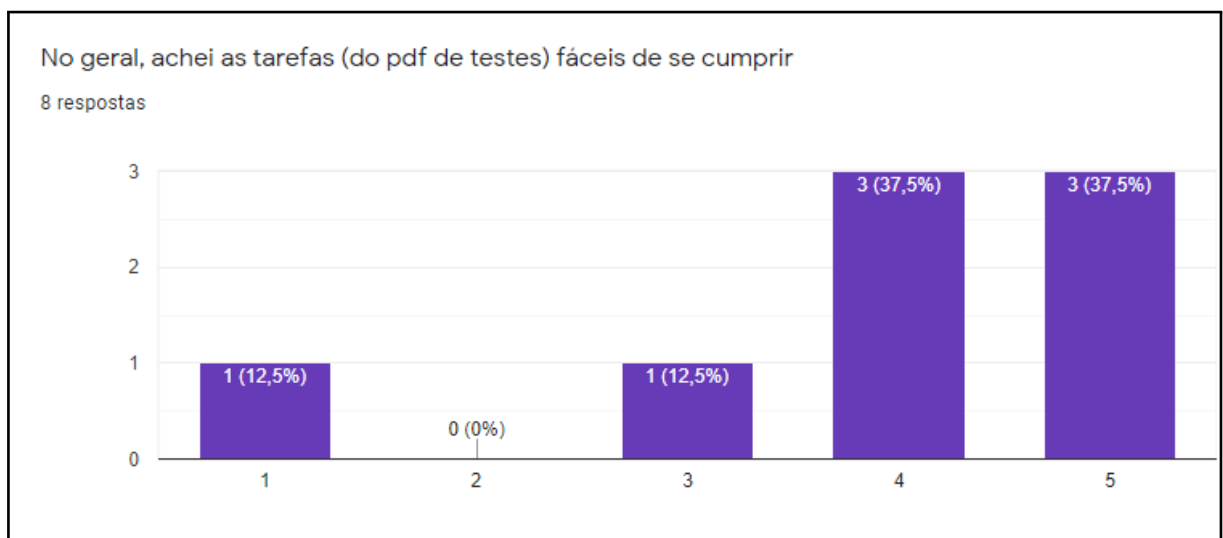


Figura 5.6 – Gráfico com o grau de concordância para a afirmação 'No geral, achei as tarefas (do pdf de testes) fáceis de se cumprir'.

6 CONSIDERAÇÕES FINAIS

A adaptação das instituições de ensino provocada pela pandemia desencadeou um crescimento dos ambientes que pudessem cumprir a função de substituir minimamente um contexto de sala de aula. Com o passar do tempo, tanto os professores, quanto os alunos, foram definindo as ferramentas que mais se adequavam às necessidades de suas realidades.

O crescimento da base de usuários do Discord reforça que soluções gratuitas e multiplataformas acabam por reunir mais público e diversificando o seu uso para além do originalmente desenvolvido. Observando esse crescimento e entendendo o público-alvo do aplicativo, o bot foi pensado e desenvolvido para o Discord.

A discussão quanto à escolha da plataforma é um assunto amplo, pois cada uma possui prós e contras. A escolha pela mais popular se deu justamente para auxiliar na transição, por parte do aluno, do ensino presencial ao ensino remoto dado que a familiaridade dos alunos com o ambiente auxilia no processo de ensino-aprendizagem. Uma vez que os professores poderiam ter dificuldades com o Discord e até mesmo sentir falta de algumas funcionalidades que estão presentes em Ambientes Virtuais de Aprendizagem, as ferramentas desenvolvidas propõem-se à auxiliar nessas dificuldades e prover algumas das funcionalidades indisponíveis.

Dado que a utilização de bots no Discord é muito popular, realizar a intervenção no ambiente através de um bot e conceder ao professor acesso a informações sobre seus alunos, transforma a ferramenta que por vezes é utilizada para entretenimento, em algo útil para auxiliar o uso do Discord dentro de um ambiente letivo. Os testes confirmam que o bot consegue cumprir aquilo que propõe de forma eficaz e, mesmo se utilizado em maiores escalas, necessitará de poucas adaptações. O dashboard foi implementado parcialmente de forma a complementar o bot e facilitar o que foi relatado como dificuldade pelos usuários durante os testes, nesse caso, a utilização dos comandos.

Entende-se que a ferramenta pode trazer mais estatísticas, principalmente para o dashboard, e que modificações na forma em que o bot armazena e coleta os dados podem ser implementadas em trabalhos futuros a fim de facilitar a exibição destes dados. O bot poderia ainda, com o auxílio de inteligência artificial, analisar os dados recolhidos e realizar uma interferência, por meio de mensagens privadas, notificando os alunos que não possuem participação contínua no servidor, por exemplo.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAM, M.; WESSEL, M.; BENLIAN, A. Ai-based chatbots in customer service and their effects on user compliance. **Electronic Markets**, Springer, p. 1–19, 2020.

ANDRADE, R. M. Mobile bot: um chatterbot educacional para dispositivos móveis. **Revista Brasileira de Computação Aplicada**, v. 4, n. 2, p. 83–91, 2012.

BARBALHO, C. M. e Alexandra Gomes e Jéssica Leite e Kléber Silva e T. Donuts: um bot como instrumento facilitador do processo de ensino-aprendizagem na disciplina construção de algoritmos. **Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação**, v. 1, n. 7, 2017. ISSN 2446-7634. Disponível em: <<https://revistas.setrem.com.br/index.php/reabtic/article/view/244>>.

BESSI, A.; FERRARA, E. Social bots distort the 2016 us presidential election online discussion. **First Monday**, v. 21, n. 11-7, 2016.

BRANDTZAEG, P. B.; FØLSTAD, A. Why people use chatbots. In: SPRINGER. **International Conference on Internet Science**. [S.l.], 2017. p. 377–392.

CITRON, J. **Discord: Your Place to Talk**. 2020. Disponível em: <https://blog.discord.com/your-place-to-talk-a7ffa19b901b>.

DISCORD. **About Discord – Our Mission and Values**. 2020. Disponível em: <https://discord.com/new/company>.

_____. **API Reference. Discord Developer Portal**. 2020. Disponível em: <https://discord.com/developers/docs/reference>.

FERRARA, E. et al. The rise of social bots. **Communications of the ACM**, ACM New York, NY, USA, v. 59, n. 7, p. 96–104, 2016.

FERREIRA, K. G.; CORSO, M. d. F. de; SILVA, C. I. P. da. Teste de usabilidade. **Trabalho de conclusão de curso. Universidade Federal de Minas Gerais. Belo Horizonte, MG**, 2002.

GROSSI, M. G. R.; MORAES, A. L.; BRESCIA, A. T. Interatividade em ambientes virtuais de aprendizagem no processo de ensino e aprendizagem na educação a distância. **Revista@ rquivo Brasileiro de Educação, Belo Horizonte**, v. 1, n. 1, p. 75–92, 2013.

ISHOWO-OLOKO, F. et al. Behavioural evidence for a transparency–efficiency tradeoff in human–machine cooperation. **Nature Machine Intelligence**, Nature Publishing Group, v. 1, n. 11, p. 517–521, 2019.

KUYVEN, N. L. et al. Chatbots na educação: uma revisão sistemática da literatura. **RE-NOTE**, v. 16, n. 1, 2018.

LACHER, L. L.; BIEHL, C. Investigating team effectiveness using discord: A case study using a gaming collaboration tool for the cs classroom.

MENDONÇA, I.; GRUBER, C. Interação síncrona na educação a distância a partir do olhar dos estudantes. v. 22, p. 159–174, 11 2019.

MOZAFARI, N.; WEIGER, W.; HAMMERSCHMIDT, M. Resolving the chatbot disclosure dilemma: Leveraging selective self-presentation to mitigate the negative effect of chatbot disclosure. In: . [S.l.: s.n.], 2020.

MURGIA, A. et al. Among the machines: Human-bot interaction on social q&a websites. In: **Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems**. [S.l.: s.n.], 2016. p. 1272–1279.

NELLY. **The Future of Bots on Discord**. 2020. <https://blog.discord.com/the-future-of-bots-on-discord-4e6e050ab52e>.

NONATO, H. P.; PINTO, E. N. Educação a distância—vantagens e desvantagens. **Instituto de Informática—Universidade Federal de Goiás (UFG). Goiânia/Go. Brasil**, 2013.

SHAO, C. et al. The spread of fake news by social bots. **arXiv preprint arXiv:1707.07592**, ArXiv e-prints, v. 96, p. 104, 2017.

SHAWAR, B.; ATWELL, E. Chatbots: Are they really useful? **LDV Forum**, v. 22, p. 29–49, 01 2007.

SLACK. **Guia do Slack para o ensino superior**. 2020a. Disponível em: <https://slack.com/intl/pt-br/resources/using-slack/your-guide-to-slack-for-higher-education>.

_____. **Inscreva-se para um desconto no programa Slack para educação**. 2020b. Disponível em: <https://slack.com/intl/pt-br/help/articles/206646877-Inscreva-se-para-um-desconto-no-programa-Slack-para-educa>

SMUTNY, P.; SCHREIBEROVA, P. Chatbots for learning: A review of educational chatbots for the facebook messenger. **Computers & Education**, Elsevier, v. 151, p. 103862, 2020.

THOMS, B.; ERYILMAZ, E. How media choice affects learner interactions in distance learning classes. **Computers & Education**, v. 75, p. 112 – 126, 2014. ISSN 0360-1315. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360131514000360>>.

TIJIBOY, A. V.; PEREIRA, E. A.; WOICIECHOSKI, L. R. Interação com afeto: aprendizagem em ambientes virtuais de aprendizagem. v. 7, 2009. ISSN 1679-1916. Disponível em: <<https://www.seer.ufrgs.br/renote/article/view/13986>>.

VENTURA, M. M. O estudo de caso como modalidade de pesquisa. **Revista SoCERJ**, v. 20, n. 5, p. 383–386, 2007.

VYNCK, G. D.; BERGEN, M. **Google Classroom Users Doubled as Quarantines Spread**. 2020. Disponível em: <https://www.bloomberg.com/news/articles/2020-04-09/google-widens-lead-in-education-market-as-students-rush-online>.