

Desenvolvimento de Bots com Discord.py: Uma Abordagem Prática para Iniciação à Programação

Danilo Silva de Oliveira, Débora Tavares Gonçalves,
Rafael Henrique Dalegrave Zottesso

¹IFPR - Instituto Federal do Paraná - Campus Paranavaí (IFPR)
Av. José Felipe Tequinha, 1400 - Jardim das Nações, Paranavaí - PR

{20223018999, 20213011009}@estudantes.ifpr.edu.br

rafael.zottesso@ifpr.edu.br

Abstract. *This paper presents the use of the discord.py library, in Python, as a pedagogical tool for teaching programming in technical education. Bots were developed on Discord to explore concepts such as programming logic, object-oriented design, data handling, asynchronous functions, and database integration. The research combined qualitative and quantitative approaches, using a questionnaire applied to students from technical computing courses. Results show that Discord, already part of students' daily lives, can serve as a motivating learning environment, helping to overcome difficulties in logic and databases. Therefore, bot development proves to be an effective strategy to bridge theory and practice, fostering engagement and meaningful learning.*

Resumo. *Este trabalho apresenta o uso da biblioteca discord.py, em Python, como recurso pedagógico para o ensino de programação em cursos técnicos. Foram desenvolvidos bots no Discord que exploram conceitos como lógica de programação, orientação a objetos, manipulação de dados, funções assíncronas e integração com banco de dados. A pesquisa combinou abordagem qualitativa e quantitativa, utilizando questionário aplicado a estudantes de cursos técnicos em Informática. Os resultados indicam que o Discord, já presente na rotina dos alunos, pode atuar como ambiente motivador para aprendizagem, contribuindo para superar dificuldades em lógica e banco de dados. Assim, a criação de bots mostra-se uma estratégia eficaz para aproximar teoria e prática, promovendo engajamento e aprendizagem significativa.*

1. Introdução

O Discord é uma das plataformas mais utilizadas no mundo, sendo atualmente uma das ferramentas de comunicação mais populares, especialmente entre adolescentes e jovens adultos. Criado inicialmente para oferecer suporte a comunidades de jogos online, o aplicativo expandiu suas funcionalidades, tornando-se um espaço para interações sociais, reuniões virtuais, compartilhamento de conteúdos e integração de diversas comunidades. Entre os recursos oferecidos estão voz, vídeo, texto e a possibilidade de personalização por meio de bots. O Discord se consolidou, assim, como um ambiente digital dinâmico e familiar para grande parte do público jovem, além de ser totalmente gratuito.

A presença dessas tecnologias no cotidiano dos estudantes reforça seu potencial educacional. Segundo Kenski [Kenski 2012], a incorporação de ferramentas digitais já presentes na cultura juvenil amplia as possibilidades de ensino e favorece a integração entre aprendizagem e realidade social. Nesse contexto, o Discord pode ser explorado além do entretenimento e da socialização, assumindo também um papel educativo.

Uma das ferramentas que permite essa adaptação é o `discord.py`, uma biblioteca em Python voltada à criação de bots que interagem dentro da plataforma. Por meio dela, professores podem utilizar o próprio ambiente do Discord para introduzir conceitos de programação, tornando o aprendizado mais acessível, prático e motivador. Conforme aponta Prensky [Prensky 2001], metodologias que utilizam tecnologias familiares aos alunos têm maior potencial de engajamento e participação ativa.

Ensinar programação aos jovens dentro de um espaço que já faz parte da rotina deles torna o processo de aprendizagem mais acessível e significativo. A criação de bots com `discord.py` envolve práticas fundamentais de lógica, estruturas de dados, programação orientada a eventos e integração de sistemas. Papert [Papert 1980] defende que aprender produzindo artefatos digitais tangíveis favorece a construção do conhecimento de forma criativa e contextualizada, o que se aplica diretamente a esse tipo de atividade.

Portanto, unir o potencial de engajamento do Discord à aplicabilidade do `discord.py` representa uma oportunidade inovadora de ensino. Essa abordagem não apenas aproxima os conteúdos de programação da realidade dos estudantes, mas também promove uma aprendizagem interativa e significativa, contribuindo para o desenvolvimento de habilidades tecnológicas valorizadas no mundo atual.

Embora diversas ferramentas digitais já sejam amplamente utilizadas no ensino de programação — como Scratch, Roblox, MinecraftEdu e bots para Telegram —, muitas delas exigem que os alunos interajam com ambientes externos ao seu cotidiano ou com interfaces simplificadas, mais voltadas à introdução de conceitos básicos. O Discord, em contrapartida, apresenta um diferencial importante: trata-se de uma plataforma amplamente utilizada pelos próprios estudantes no dia a dia, com recursos nativos para comunicação, colaboração e integração de aplicações personalizadas.

Essa familiaridade prévia cria condições para que o ensino ocorra em um ambiente natural para os alunos, sem a necessidade de introduzir ferramentas completamente novas. Além disso, o desenvolvimento de bots com `discord.py` permite explorar conceitos avançados de programação em um contexto motivador, combinando aspectos técnicos com práticas sociais colaborativas.

Dessa forma, este trabalho busca responder à seguinte questão de pesquisa: **como o desenvolvimento de bots no Discord pode impactar a aprendizagem de lógica de programação em cursos técnicos?** A partir dessa problematização, pretende-se evidenciar o potencial pedagógico da ferramenta e seu diferencial em relação a outras abordagens educacionais já consolidadas.

2. Objetivo

O principal objetivo deste trabalho é demonstrar como a biblioteca `discord.py`, desenvolvida em *Python*, pode ser utilizada como um recurso pedagógico eficaz para a aprendizagem de programação. Essa ferramenta possibilita a aplicação prática de diversos conceitos

fundamentais abordados ao longo de um curso técnico de informática, tais como orientação a objetos, integração com bancos de dados, manipulação de variáveis, definição de funções, entre outros. Dessa forma, busca-se evidenciar como o desenvolvimento de *bots* para a plataforma *Discord* pode servir não apenas como um exercício de fixação de conteúdos, mas também como um meio motivador e interativo para estimular o interesse dos estudantes pela área de programação.

2.1. Objetivos Específicos

Para alcançar o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- Implementar um *bot* no *Discord* utilizando a biblioteca *discord.py*, aplicando conceitos fundamentais da linguagem *Python*.
- Explorar, por meio do desenvolvimento prático, conceitos de programação como orientação a objetos, manipulação de funções, estruturas de dados e variáveis.
- Integrar o *bot* a um banco de dados *SQLite* para demonstrar técnicas de persistência e recuperação de informações.
- Analisar como o uso de uma ferramenta interativa e de fácil acesso, como o *Discord*, pode tornar o processo de aprendizagem mais dinâmico e motivador.
- Relacionar o desenvolvimento de *bots* a situações reais que exigem lógica de programação, resolução de problemas e automação de tarefas.
- Avaliar o potencial do *discord.py* como recurso complementar no ensino de programação em cursos técnicos.

3. Fundamentação Teórica

A fundamentação teórica de um trabalho acadêmico tem como objetivo situar a pesquisa no campo científico, apresentando conceitos, estudos e abordagens que sustentam a proposta. No caso do ensino de programação em cursos técnicos, diversos autores apontam para as dificuldades enfrentadas pelos estudantes na compreensão de conceitos abstratos e na aplicação prática dos conteúdos [Valente 1999, Pereira and Bertagnolli 2020]. Além disso, teorias da aprendizagem, como a de Ausubel [Ausubel 1982], Piaget [Piaget 1975] e Vygotsky [Vygotsky 1998], fornecem aportes importantes para pensar estratégias que aproximem teoria e prática. A incorporação de tecnologias digitais em sala de aula, como destaca Kenski [Kenski 2012], surge como um caminho para tornar o aprendizado mais interativo e motivador. Assim, este capítulo busca discutir os desafios do ensino de programação em cursos técnicos e o potencial do uso de ferramentas digitais, como o *Discord*, para auxiliar nesse processo.

3.1. Ensino de programação em cursos técnicos

A fundamentação teórica deste trabalho tem como objetivo estabelecer a relação entre abordagens pedagógicas consolidadas e o uso de tecnologias digitais no ensino de programação. De acordo com Ausubel [Ausubel 1982], a aprendizagem significativa ocorre quando novos conteúdos são conectados ao conhecimento prévio do estudante, permitindo que a assimilação seja mais duradoura. Piaget [Piaget 1975] enfatiza a construção ativa do conhecimento por meio da interação prática com o ambiente, enquanto Vygotsky [Vygotsky 1998] destaca a importância da mediação social e cultural para potencializar o desenvolvimento cognitivo.

O uso de plataformas digitais como o Discord está diretamente alinhado a esses pressupostos teóricos. Por ser um ambiente de comunicação colaborativa e interativa, a plataforma permite que os estudantes aprendam de forma ativa, social e contextualizada. Além disso, ao desenvolver bots utilizando *discord.py*, os alunos lidam com situações reais de resolução de problemas, o que reforça a ligação entre teoria e prática — elemento central nas teorias construtivistas e sociointeracionistas.

Estudos recentes também apontam que o uso de ambientes digitais gamificados pode aumentar a motivação e o engajamento dos alunos em disciplinas técnicas [Silva and Pereira 2023, Moraes 2022]. Além disso, a popularização de tecnologias baseadas em inteligência artificial e automação tem ampliado as possibilidades de personalização do aprendizado, tornando as experiências mais dinâmicas e próximas da realidade dos estudantes [Almeida 2021]. Como ressalta Kenski [Kenski 2012], a incorporação de tecnologias digitais já presentes no cotidiano juvenil amplia as possibilidades de ensino e favorece a integração entre aprendizagem e realidade social.

Nesse contexto, Moderna Compartilha [Compartilha 2025] reforça que metodologias de ensino eficazes devem partir dos interesses e potencialidades dos alunos, conectando conteúdos curriculares à sua realidade. O uso do Discord como ferramenta pedagógica atende diretamente a essa perspectiva, pois aproveita um ambiente já conhecido pelos estudantes para introduzir conceitos de lógica, programação orientada a objetos e integração de sistemas de forma prática e significativa.

Nesse cenário, o uso de tecnologias digitais surge como estratégia pedagógica relevante. Kenski [Kenski 2012] aponta que a incorporação das Tecnologias da Informação e Comunicação (TICs) na educação amplia as possibilidades de mediação, permitindo experiências mais dinâmicas e próximas da realidade dos estudantes. Plataformas digitais, jogos e ambientes virtuais podem contribuir para tornar a aprendizagem de programação mais envolvente, aproximando os conteúdos das práticas cotidianas dos jovens [Prensky 2001].

O *Discord*, por exemplo, consolidou-se como uma das plataformas de comunicação mais utilizadas no mundo, especialmente entre adolescentes e jovens adultos [Statista 2023]. Originalmente voltado para comunidades de jogos, o aplicativo expandiu suas funcionalidades, oferecendo recursos de texto, voz, vídeo e compartilhamento de arquivos, além da possibilidade de integração com *bots* programáveis. Segundo Moderna Compartilha [Compartilha 2025], metodologias de ensino devem se organizar a partir dos interesses e potencialidades dos estudantes. Nesse sentido, utilizar o *Discord* como recurso pedagógico torna-se viável justamente por ser uma ferramenta já presente no cotidiano dos alunos, capaz de unir motivação, familiaridade e prática aplicada no processo de aprendizagem de programação.

4. A Biblioteca *discord.py*

A *discord.py* é uma biblioteca *open source* criada por Rapptz em 2015, projetada para que desenvolvedores possam construir *bots* para o *Discord* usando a linguagem *Python*. Ela funciona como um *wrapper* da API do *Discord*, fornecendo uma interface programática que abstrai muitos dos detalhes de conexão e protocolo.

Um dos diferenciais da biblioteca está no suporte ao paradigma assíncrono via *async/await*, recurso introduzido no *Python* 3.5 e que permite a execução concorrente

de múltiplas tarefas sem bloqueio, característica essencial em aplicações que demandam respostas rápidas e contínuas [Van Rossum and Drake 2014, Rapptz 2015]. Esse recurso é especialmente útil em contextos de *bots*, que precisam reagir a eventos de mensagens, comandos ou interações de usuários em tempo real.

Outro aspecto relevante é o design orientado a objetos implementado pela biblioteca. Usuários, canais, mensagens e o próprio *bot* são tratados como objetos, cada um com atributos e métodos específicos. Essa abordagem aproxima o aprendizado prático de conceitos como classes, herança e encapsulamento, tradicionalmente ensinados de forma abstrata em cursos técnicos e superiores [Rapptz 2015, Deitel and Deitel 2013].

A biblioteca também se destaca por sua documentação bem estruturada e por uma comunidade ativa, que mantém exemplos oficiais e fóruns de suporte. Esse ecossistema facilita a entrada de iniciantes, oferecendo materiais de aprendizagem acessíveis e práticas de colaboração típicas de projetos de código aberto [discord.py 2025, Raymond 1999].

Apesar das vantagens, algumas limitações devem ser consideradas. É necessário conhecimento prévio de *Python*, especialmente no uso de funções assíncronas, além de acesso constante à internet para manter o *bot* ativo. Além disso, a biblioteca já passou por mudanças significativas entre versões principais (*breaking changes*), exigindo que os desenvolvedores acompanhem constantemente as atualizações [Rapptz 2015, discord.py 2025].

No campo educacional, a utilização de *discord.py* oferece grande potencial pedagógico. Ao permitir que os estudantes transformem teoria em prática em um ambiente familiar, ela contribui para o desenvolvimento de competências em lógica, modularização, persistência de dados e integração com APIs. Essa visibilidade imediata dos resultados favorece o aprendizado ativo, a motivação e o engajamento, características alinhadas às metodologias de ensino que valorizam a construção do conhecimento a partir de práticas significativas [Papert 1980, Kenski 2012, Pereira and Bertagnolli 2020].

5. Metodologia

Este trabalho caracteriza-se como uma pesquisa exploratória com abordagem quantitativa descritiva, realizada por meio da aplicação de um questionário estruturado com estudantes de cursos técnicos em Informática. O objetivo foi identificar percepções sobre dificuldades no aprendizado de programação e hábitos de uso da plataforma *Discord*.

O instrumento de coleta de dados foi elaborado no Google Forms e contou com 10 questões fechadas e abertas, distribuídas em dois blocos: (1) perfil e experiência prévia dos alunos com programação; (2) frequência e familiaridade com o *Discord*.

A amostra foi composta por 22 estudantes, todos matriculados no curso técnico em Informática do IFPR, com idades entre 16 e 20 anos. A participação foi voluntária e sem identificação nominal.

Os dados coletados foram organizados em planilhas e analisados por meio de estatística descritiva simples (frequências e porcentagens). As respostas abertas foram categorizadas qualitativamente, permitindo identificar percepções e padrões recorrentes.

Embora o desenvolvimento de *bots* com a biblioteca *discord.py* tenha servido de base prática para contextualizar a pesquisa, os alunos não participaram diretamente da im-

plementação. Essa etapa foi conduzida pelos autores do trabalho para ilustrar o potencial pedagógico da ferramenta.

5.1. Desenvolvimento do *Bot*

O bot apresentado nesta seção foi desenvolvido pelo autor com finalidade demonstrativa, para exemplificar o potencial pedagógico da integração entre lógica de programação e a plataforma Discord. Os estudantes não participaram diretamente do desenvolvimento, tendo apenas respondido ao questionário aplicado na pesquisa.

Foram criados dois *bots* distintos com a biblioteca *discord.py*. O primeiro, de caráter simplificado, tinha como objetivo explorar comandos básicos de administração, como *mute*, *kick* e *ban*, permitindo exercitar lógica condicional e eventos assíncronos. O segundo *bot*, inspirado em um sistema de *RPG*, foi desenvolvido para demonstrar a integração de um banco de dados *SQLite*, possibilitando registrar e manipular informações de usuários. Essa escolha buscou evidenciar como conceitos de persistência, manipulação de dados e modularização podem ser aplicados de forma prática.

```

1 # bot.py
2 import discord
3 from discord.ext import commands
4 from database import init_db
5 from commands import appearance
6
7 intents = discord.Intents.default()
8 intents.guilds = True
9 intents.members = True
10 intents.message_content = True
11
12 bot = commands.Bot(command_prefix="!", intents=intents)
13
14 @bot.event
15 async def on_ready():
16     print(f"✅ Bot conectado como {bot.user}")
17     init_db()
18     await bot.tree.sync()
19     print("🔄 Slash commands sincronizados com o Discord.")
20
21 import commands.admin
22 import commands.user
23
24 commands.admin.setup_admin_commands(bot)
25 commands.user.setup_user_commands(bot)
26 commands.appearance.setup_appearance_commands(bot)
27
28 bot.run("Seu_Token_Aqui")
    
```

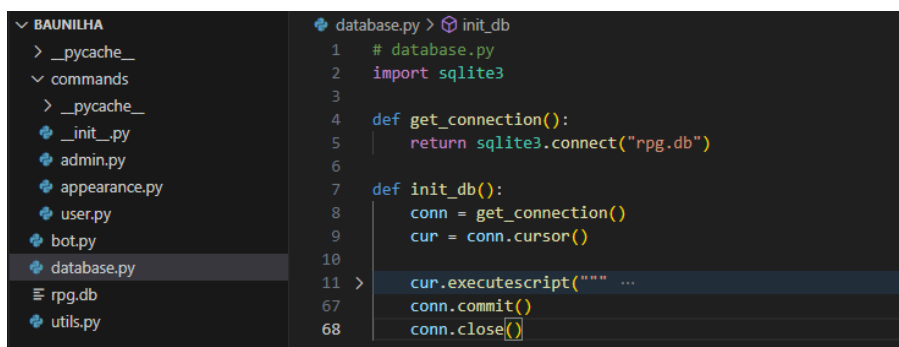
Figura 1. Estrutura de pastas e código principal do *bot* desenvolvido em *Python*.

A Figura 1 apresenta a organização das pastas e o código principal do *bot*. O arquivo `bot.py` concentra a inicialização da aplicação, definindo *intents*, sincronização de *slash commands* e importação de módulos auxiliares, como os comandos administrativos e de usuário. Essa modularização facilita a manutenção do projeto e a compreensão dos conceitos de programação envolvidos.

A Figura 2 apresenta a implementação do banco de dados no arquivo `database.py`, responsável por registrar e recuperar informações dos usuários. Essa integração evidencia como conceitos de persistência de dados podem ser aplicados de maneira prática em projetos de programação.

5.2. Tecnologias Usadas

O desenvolvimento do *bot* foi realizado na linguagem *Python*, escolhida por sua sintaxe acessível e ampla utilização tanto no meio acadêmico quanto no mercado de trabalho.



```

1  # database.py
2  import sqlite3
3
4  def get_connection():
5      return sqlite3.connect("rpg.db")
6
7  def init_db():
8      conn = get_connection()
9      cur = conn.cursor()
10
11  > cur.executescript(""" ...
67  conn.commit()
68  conn.close()
    
```

Figura 2. Arquivo `database.py` implementando a estrutura de banco de dados *SQLite* utilizada pelo *bot*.

Como ambiente de desenvolvimento, utilizou-se o *Visual Studio Code (VS Code)*, uma IDE gratuita que oferece recursos como destaque de sintaxe, terminal integrado e suporte a extensões específicas para *Python*.

A biblioteca *discord.py* foi a principal ferramenta empregada, pois permite a integração programática entre o código em *Python* e a plataforma *Discord*, possibilitando a implementação de eventos, comandos e interações em tempo real. Para garantir persistência de dados, utilizou-se o *SQLite*, banco de dados relacional leve e embutido, adequado para projetos educacionais e de pequeno porte.

5.3. Conceitos Pedagógicos Envolvidos

O uso dessas tecnologias não se restringiu ao desenvolvimento técnico, mas também foi pensado a partir de uma perspectiva pedagógica. O processo de programar *bots* promove o aprender fazendo, em que o estudante consolida conceitos teóricos ao aplicá-los em situações práticas.

Segundo Papert [Papert 1980], no construcionismo, a aprendizagem é potencializada quando o aluno constrói artefatos que têm significado em sua realidade, o que se aplica diretamente ao desenvolvimento de *bots* no *Discord*, um ambiente digital familiar aos jovens. Além disso, Kenski [Kenski 2012] destaca que as tecnologias digitais possibilitam novas formas de mediação no ensino, permitindo ao estudante interagir, experimentar e refletir sobre seus próprios processos de aprendizagem.

Nesse sentido, a construção do *bot* possibilitou exercitar conteúdos-chave como lógica de programação, modularização, manipulação de dados e integração com APIs, aproximando teoria e prática de forma motivadora.

5.4. Execução em Nuvem e Manutenção do *Bot*

Durante o processo de implementação, foi necessário gerar um arquivo `requirements.txt` para listar todas as dependências usadas no projeto, o que foi feito com o comando:

```
pipreqs . --encoding=utf-8 --force
```

Esse arquivo é fundamental para implantar o *bot* em serviços de hospedagem na nuvem, como a *Square Cloud*, pois permite que o ambiente de execução instale automaticamente as bibliotecas necessárias, garantindo a portabilidade e a reprodutibilidade do projeto.

A hospedagem em nuvem também resolve uma limitação prática: manter o *bot* rodando 24 horas por dia em um computador pessoal pode sobrecarregar o hardware, gerar alto consumo de energia e reduzir a vida útil da máquina devido ao funcionamento contínuo. Além disso, quedas de energia ou interrupções de internet podem comprometer a estabilidade do *bot*. Dessa forma, a utilização de serviços online se torna uma alternativa mais eficiente, confiável e sustentável.

6. Resultados

Com base nos objetivos estabelecidos neste trabalho, buscou-se avaliar como o uso da plataforma *Discord* e o desenvolvimento de *bots* podem contribuir para o processo de ensino-aprendizagem em cursos técnicos de informática. Os resultados apresentados a seguir são fruto da análise do questionário aplicado aos estudantes e têm o propósito de relacionar as percepções deles às dificuldades e oportunidades identificadas durante a pesquisa.

6.1. Nível de Confiança em Programar

Ao serem questionados sobre o nível de confiança em programar, os estudantes responderam em uma escala de 1 a 5. A Figura 3 mostra a distribuição dessas respostas, permitindo observar a percepção geral dos participantes em relação às próprias habilidades.

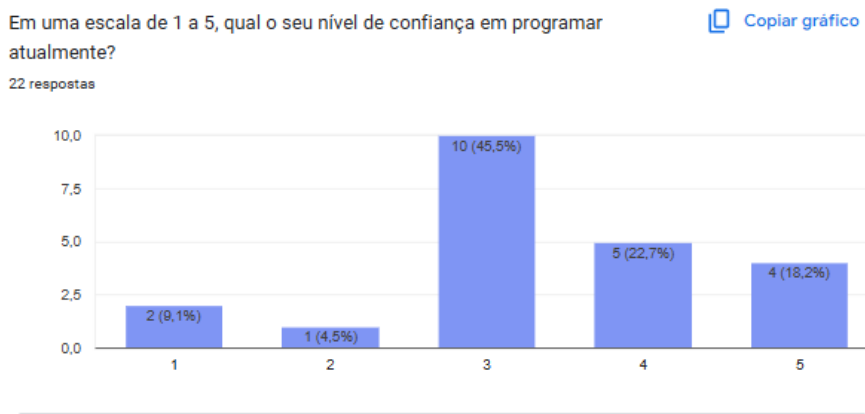


Figura 3. Nível de confiança em programar atualmente.

A análise revela que a maior parte dos alunos (45,5%) indicou nível 3, seguido de 22,7% no nível 4 e 18,2% no nível 5. Apenas 13,6% se posicionaram nos níveis 1 e 2. Esses dados indicam que a maioria se encontra em um estágio intermediário de confiança, com uma parcela já demonstrando autoconfiança elevada.

6.2. Dificuldades em Programação

Também foi investigado quais partes da programação os alunos consideram mais desafiadoras. A Figura 4 apresenta os resultados dessa questão.

Qual parte da programação você considera mais difícil?

22 respostas

 Copiar gráfico

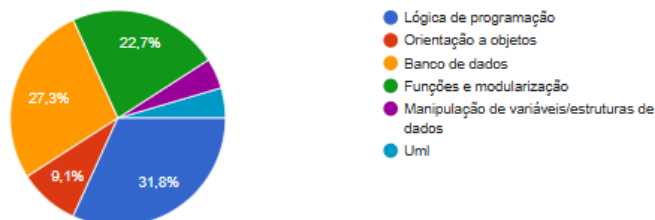


Figura 4. Partes da programação consideradas mais difíceis.

Conforme mostra a figura, lógica de programação (31,8%) e banco de dados (27,3%) foram os tópicos mais apontados. Em seguida, aparecem manipulação de variáveis e estruturas de dados (22,7%). Já funções/modularização (9,1%), orientação a objetos (4,5%) e UML (4,5%) foram menos mencionadas. Esses resultados reforçam a necessidade de metodologias práticas que auxiliem na compreensão da lógica e da persistência de dados.

6.3. Uso do *Discord* pelos Estudantes

Por fim, analisou-se o uso do *Discord* pelos alunos. A Figura 5 apresenta a proporção de participantes que já utilizaram a plataforma.

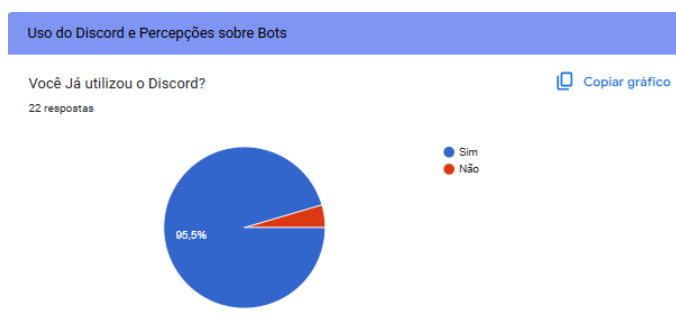


Figura 5. Proporção de estudantes que já utilizaram o *Discord*.

Nota-se que a ampla maioria (95,5%) declarou já ter utilizado o *Discord*, contra apenas 4,5% que nunca utilizaram. Além disso, a frequência de uso está apresentada na Figura 6.

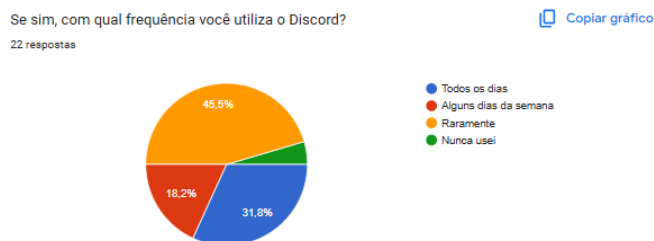


Figura 6. Frequência de uso do *Discord* pelos estudantes.

Os dados revelam que 31,8% dos alunos utilizam o *Discord* diariamente, 45,5% alguns dias da semana, 18,2% raramente e apenas 4,5% nunca. Esses resultados confirmam que o *Discord* já faz parte da rotina dos estudantes e, portanto, apresenta alto potencial para uso pedagógico.

6.4. Discussão

Os resultados obtidos permitem compreender melhor o perfil dos estudantes e suas percepções em relação à aprendizagem de programação. A predominância de níveis intermediários de confiança em programar indica que os alunos já possuem alguma familiaridade com os conceitos básicos, mas ainda enfrentam barreiras para avançar em tópicos mais complexos. Essa constatação está alinhada aos achados de Pereira e Bertagnolli [Pereira and Bertagnolli 2020], que apontam a lógica de programação e a persistência de dados como as maiores dificuldades entre estudantes de cursos técnicos e superiores.

Além disso, o fato de a ampla maioria dos alunos já utilizar a plataforma *Discord* confirma seu potencial de aplicação pedagógica. Como defendido por Kenski [Kenski 2012], metodologias que incorporam ferramentas do cotidiano dos alunos favorecem a aprendizagem significativa ao diminuir a distância entre contextos formais e informais. Essa familiaridade pode ser um fator estratégico para aumentar a motivação e o engajamento no processo de ensino.

A predominância de respostas que apontam lógica e banco de dados como áreas mais difíceis sugere que estratégias tradicionais de ensino podem não estar atendendo plenamente as necessidades de aprendizagem. Segundo Vygotsky [Vygotsky 1998], o aprendizado é potencializado em ambientes colaborativos e mediados socialmente — características inerentes à plataforma *Discord*.

6.5. Limitações e Implicações Práticas

É importante ressaltar que este estudo possui limitações. A amostra foi reduzida (22 participantes) e composta por alunos de um único campus, o que restringe a generalização dos resultados. Além disso, os dados coletados baseiam-se exclusivamente em percepções autorrelatadas, sem aplicação de instrumentos objetivos para mensurar desempenho em programação.

Mesmo com essas limitações, os achados indicam que o uso de ambientes digitais amplamente utilizados pelos estudantes pode ser uma estratégia relevante para apoiar práticas pedagógicas. O *Discord* apresenta potencial para ser incorporado a projetos educacionais que envolvam lógica de programação, integração de APIs e aprendizagem cola-

borativa. Estudos futuros podem ampliar a amostra e aplicar instrumentos experimentais para mensurar o impacto dessa abordagem de forma mais robusta.

7. Considerações Finais

Este trabalho apresentou uma proposta de uso da plataforma *Discord* como ferramenta pedagógica complementar no ensino de programação em cursos técnicos de informática. Por meio de um questionário aplicado a 22 estudantes do IFPR – Campus Paranavaí, buscou-se compreender as percepções dos alunos sobre suas dificuldades em programação e sua familiaridade com a plataforma.

Os resultados evidenciaram que a maioria dos participantes apresenta um nível intermediário de confiança em programar e que lógica de programação e banco de dados são os conteúdos mais desafiadores. Além disso, observou-se que o *Discord* já faz parte do cotidiano desses alunos, o que indica um alto potencial para integrar práticas pedagógicas que aliem conteúdos técnicos e ambientes digitais familiares.

A proposta técnica de desenvolvimento de um *bot* no Discord demonstra como recursos interativos e colaborativos podem ser utilizados para criar experiências de aprendizagem mais dinâmicas e engajadoras. Embora os estudantes não tenham participado diretamente do desenvolvimento, a ferramenta serve como base para futuras aplicações práticas.

Como limitações, destaca-se o tamanho reduzido da amostra e a natureza perceptiva dos dados coletados. Para trabalhos futuros, sugere-se a ampliação da pesquisa, a implementação de intervenções pedagógicas com participação ativa dos alunos e a utilização de métricas de desempenho para avaliar impactos no aprendizado. Dessa forma, será possível fortalecer a base empírica sobre o uso de plataformas digitais, como o Discord, no ensino de programação.

Referências

- Almeida, C. (2021). Inteligência artificial e personalização da aprendizagem em ambientes digitais. In *Anais do Simpósio Brasileiro de Informática na Educação*.
- Ausubel, D. P. (1982). *A aprendizagem significativa: a teoria de David Ausubel*. Moraes, São Paulo.
- Compartilha, M. (2025). Metodologia de ensino. Disponível em: <https://www.modernacompartilha.com.br/metodologia-de-ensino>. Acesso em: 16 set. 2025.
- Deitel, H. M. and Deitel, P. J. (2013). *Java: Como programar*. Pearson, São Paulo, 10 edition.
- discord.py (2025). discord.py documentation. Disponível em: <https://discordpy.readthedocs.io/>. Acesso em: 20 set. 2025.
- Kenski, V. M. (2012). *Educação e tecnologias: o novo ritmo da informação*. Papirus, Campinas, 8 edition.
- Moraes, A. (2022). Plataformas digitais e engajamento no ensino de programação. *Educação e Tecnologia*.

- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York.
- Pereira, L. F. and Bertagnolli, S. C. (2020). Dificuldades no ensino-aprendizagem de programação: uma revisão sistemática. *RENOTE*, 18(1).
- Piaget, J. (1975). *A formação do símbolo na criança: imitação, jogo e sonho, imagem e representação*. Zahar, Rio de Janeiro.
- Prensky, M. (2001). *Digital Game-Based Learning*. McGraw-Hill, New York.
- Rapptz (2015). discord.py. Disponível em: <https://github.com/Rapptz/discord.py>. Acesso em: 20 set. 2025.
- Raymond, E. (1999). *The Cathedral and the Bazaar*. O'Reilly Media.
- Silva, J. and Pereira, M. (2023). Gamificação no ensino técnico: estratégias e resultados práticos. *Revista Brasileira de Educação Tecnológica*.
- Statista (2023). Most popular communication apps worldwide 2023. Disponível em: <https://www.statista.com/>.
- Valente, J. A. (1999). *O computador na sociedade do conhecimento*. UNICAMP/NIED, Campinas.
- Van Rossum, G. and Drake, F. L. (2014). *The Python Language Reference Manual*. Amsterdam.
- Vygotsky, L. S. (1998). *A formação social da mente*. Martins Fontes, São Paulo, 6 edition.