

Como gerenciar sua equipe no Discord integrando com o Jira

Bruno Castro

Nayane Galeno

**salutations my good
chum**



Sabemos que a área de TIC é repleta de termos e pode ser desafiador acompanhar e absorver tudo.

Por isso, qualquer dúvida que surgir, fiquem à vontade para perguntar!

Tópicos abordados

- Jira
- Discord
- Funcionamento do bot
- Do que vamos precisar?
- Setup do projeto no Jira
- Setup do projeto no Discord
- Eventos do Discord
- Jira API
- Criação do projeto
- E depois?

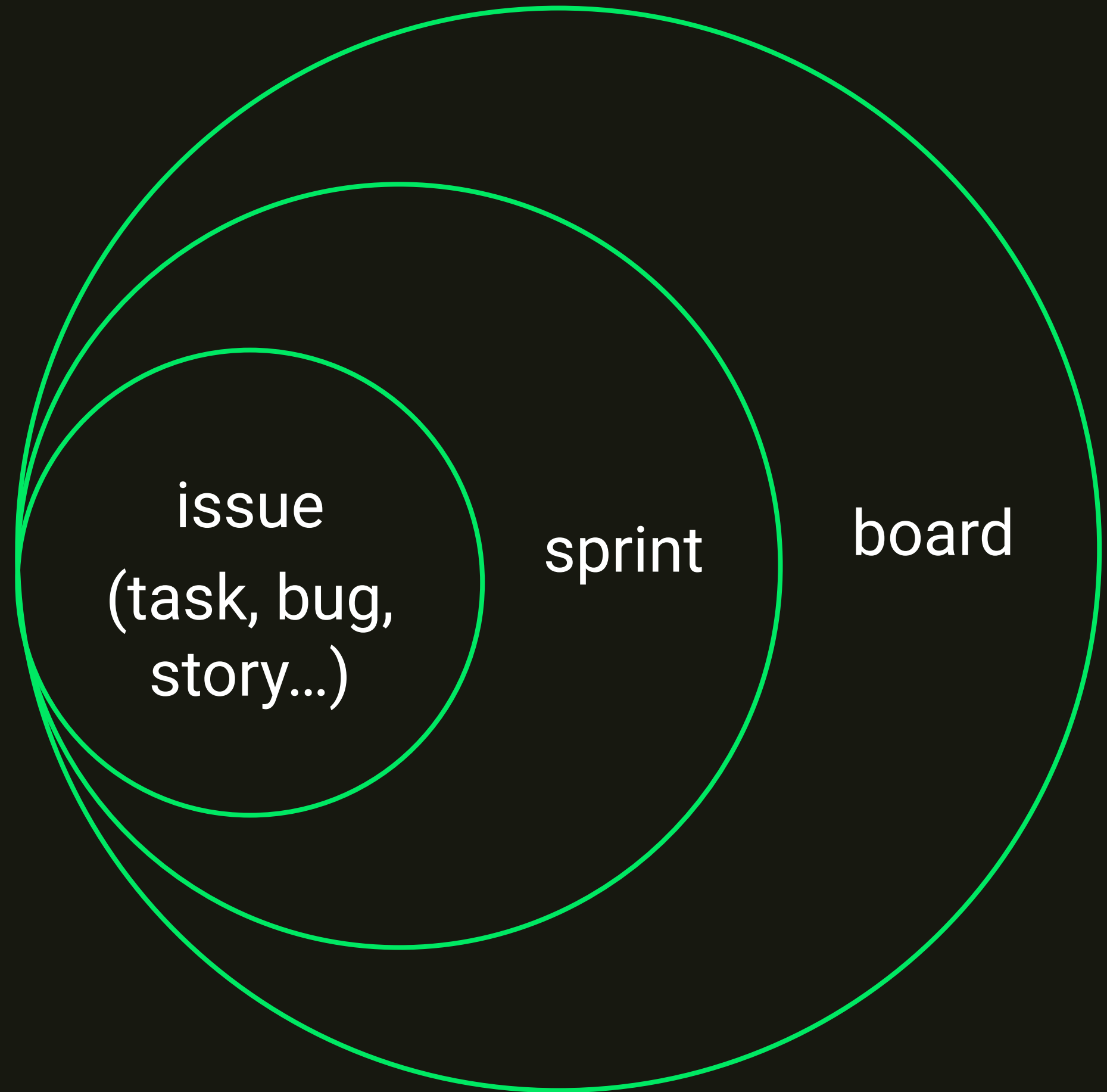
Discord

- Plataforma de comunicação online
- Inicialmente voltada para gamers, mas agora utilizada por uma ampla gama de comunidades e grupos
- Chat por texto, voz e vídeo
- Permite a criação de servidores personalizados para interações entre membros
- Canais categorizados por temas, permissões personalizadas para usuários e bots que automatizam funções

Jira

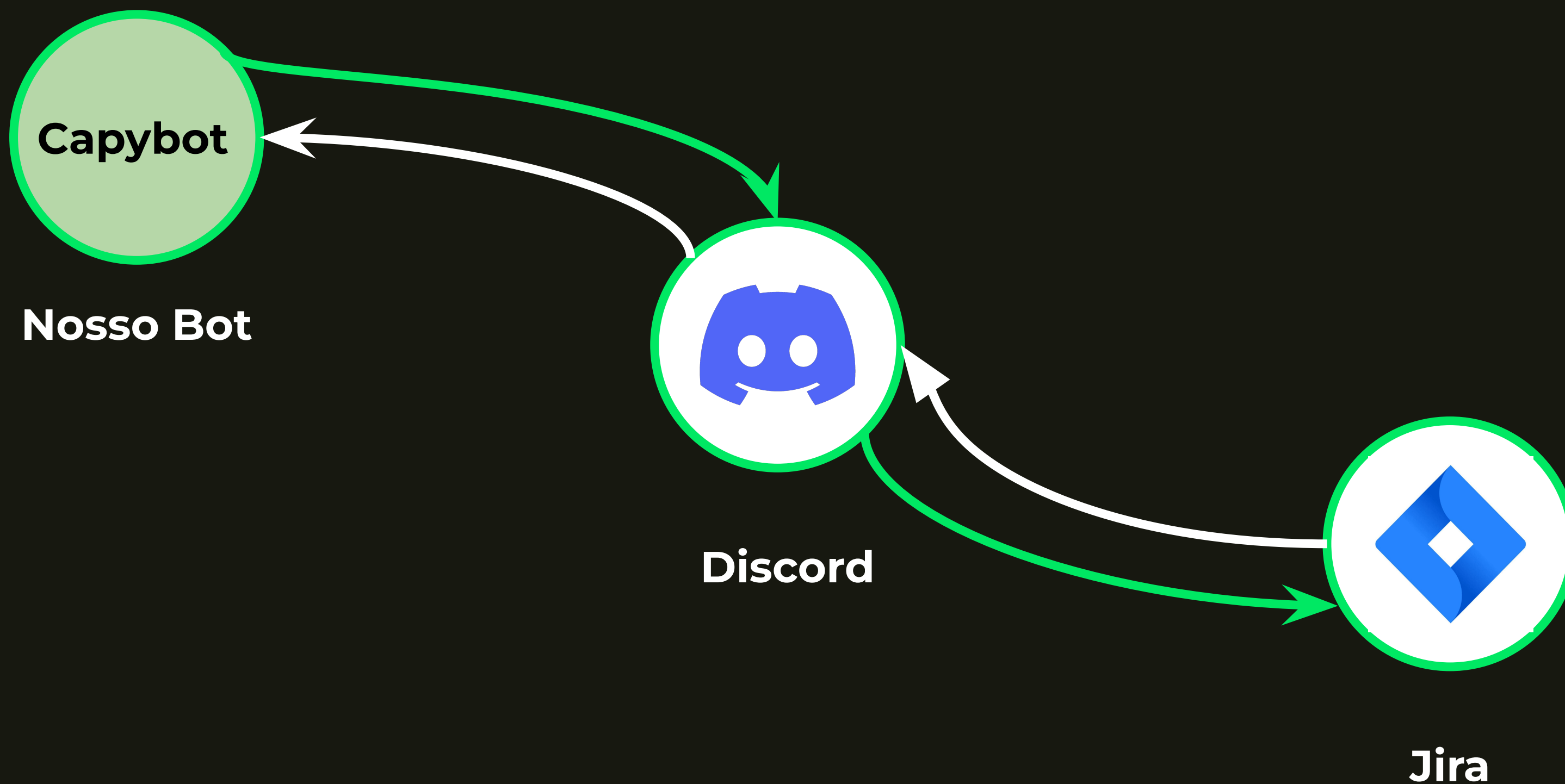
- Sistema de gerenciamento de projetos ágeis (diversas áreas)
- Scrum ou Kanban
- criação e acompanhamento de tarefas
- gestão de sprints
- organização de backlog
- acompanhamento de bugs e problemas
- relatórios e painéis customizáveis

Jira - scrum



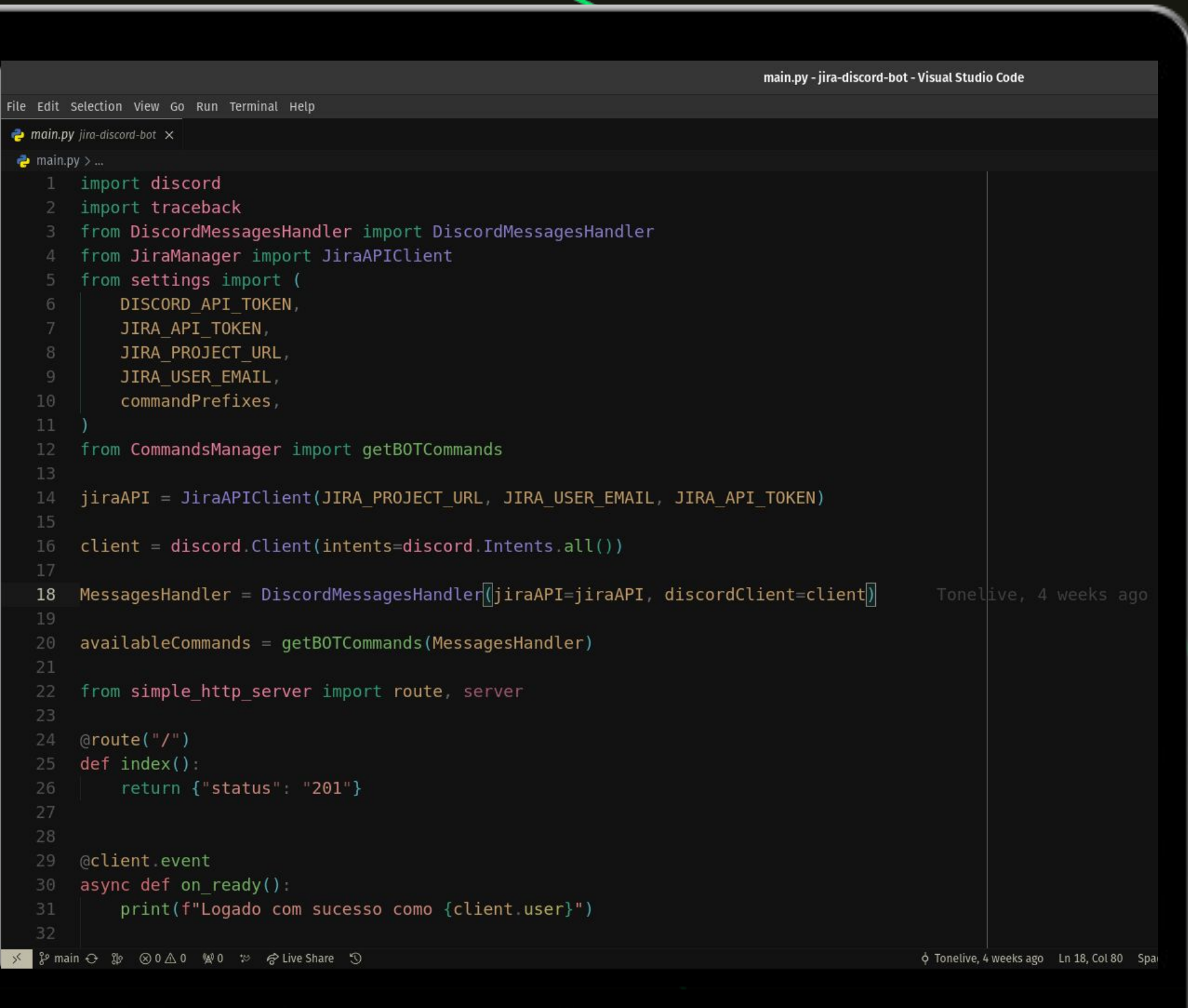
Funcionamento do bot

Quando solicitado via comando do Discord, busca informações no Jira e responde com um relatório



Do que vamos precisar?

- IDE favorita
- Python 3.10
- Conta no Discord
- Conta no Jira
 - Não pode ser criada com o Google (Não nos perguntem o por que)
- Vamos fornecer uma conta previamente criada e configurada para agilizar o processo.



```
main.py - jira-discord-bot - Visual Studio Code
File Edit Selection View Go Run Terminal Help
main.py jira-discord-bot X
main.py > ...
1 import discord
2 import traceback
3 from DiscordMessagesHandler import DiscordMessagesHandler
4 from JiraManager import JiraAPIClient
5 from settings import (
6     DISCORD_API_TOKEN,
7     JIRA_API_TOKEN,
8     JIRA_PROJECT_URL,
9     JIRA_USER_EMAIL,
10     commandPrefixes,
11 )
12 from CommandsManager import getBOTCommands
13
14 jiraAPI = JiraAPIClient(JIRA_PROJECT_URL, JIRA_USER_EMAIL, JIRA_API_TOKEN)
15
16 client = discord.Client(intents=discord.Intents.all())
17
18 MessagesHandler = DiscordMessagesHandler(jiraAPI=jiraAPI, discordClient=client)
19
20 availableCommands = getBOTCommands(MessagesHandler)
21
22 from simple_http_server import route, server
23
24 @route("/")
25 def index():
26     return {"status": "201"}
27
28
29 @client.event
30 async def on_ready():
31     print(f"Logado com sucesso como {client.user}")
32
```

Primeiro passo

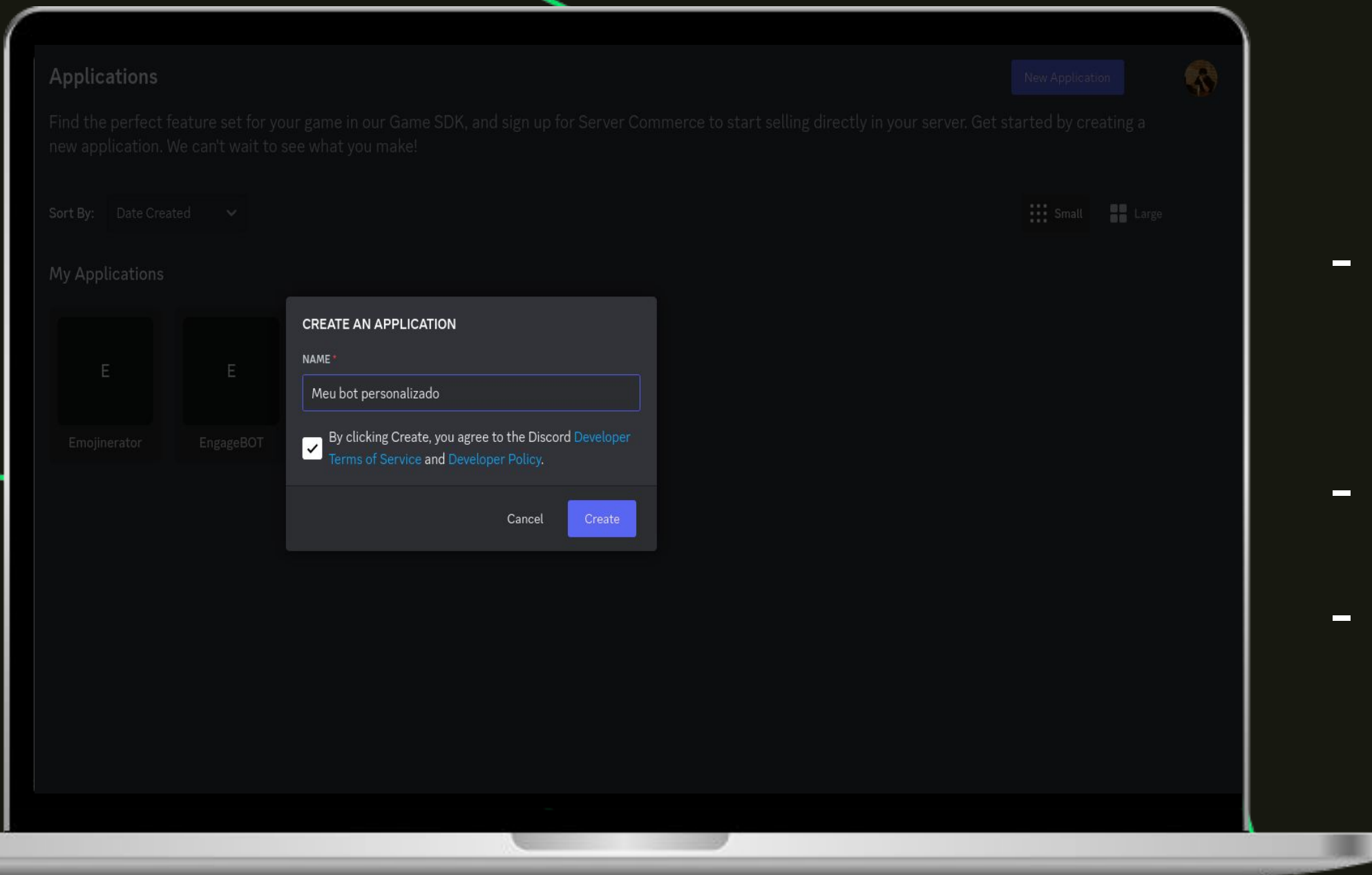
Acesse o repositório do projeto e entre no canal do Discord que está no README.

Com isso vocês terão acesso a todos os links importantes na guia “Links”

<https://github.com/CapybaHub/jira-discord-bot>

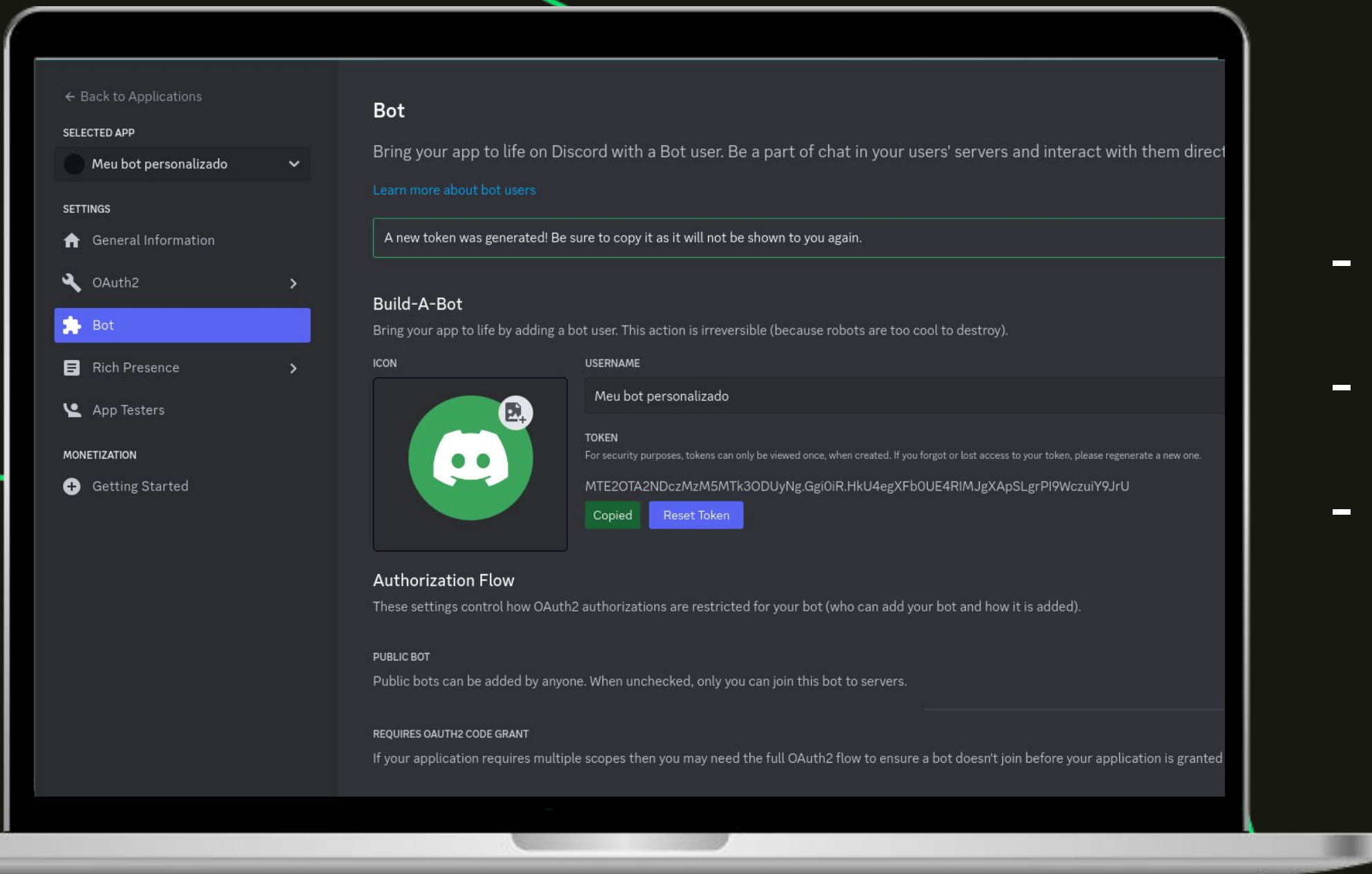


Discord Developer



- Acesse o link do Discord Developer disponível na aba Links
 - Faça login com a sua conta
 - Crie uma nova aplicação
- OBS: Utilizem nomes diferentes

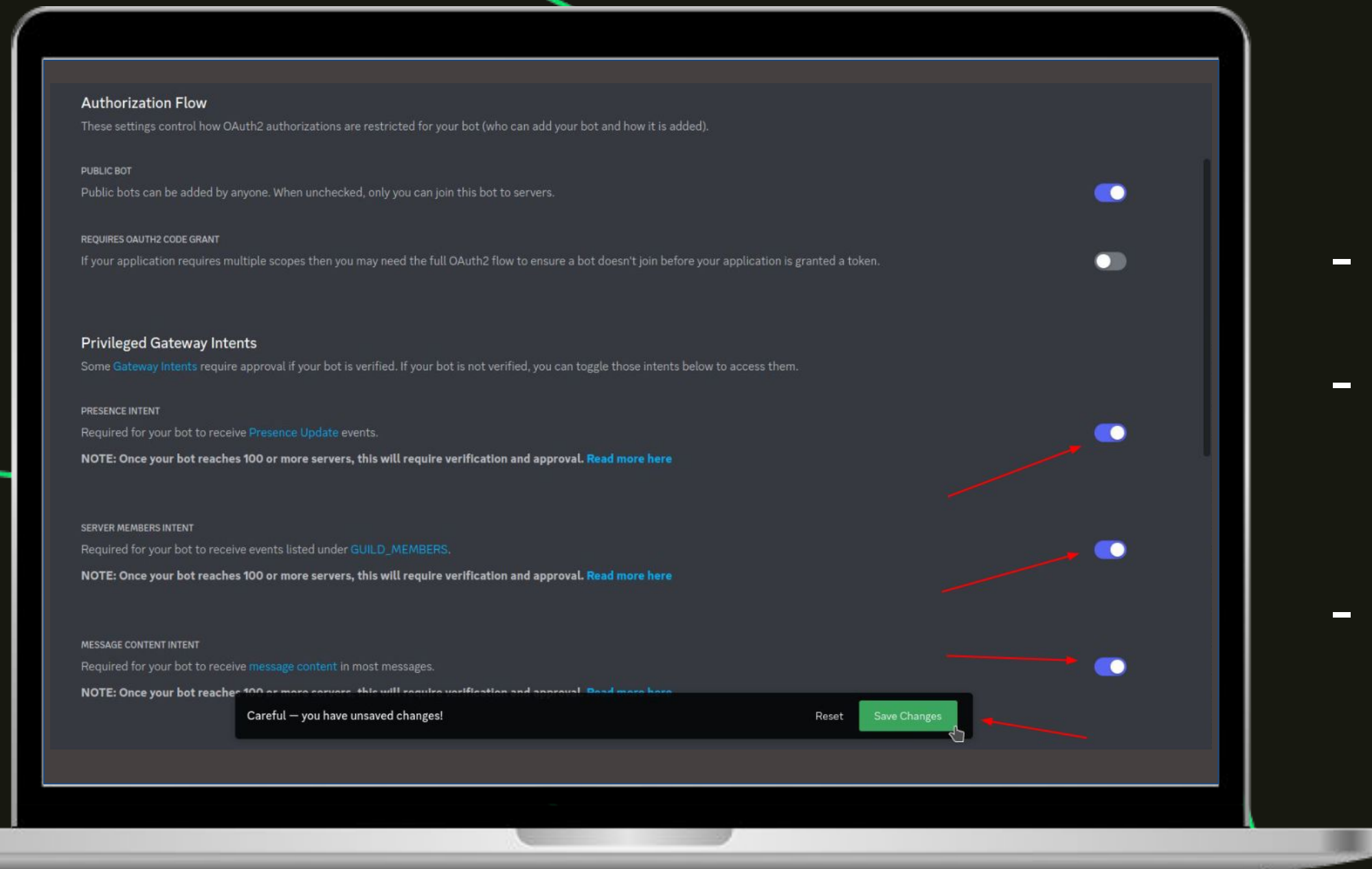
Discord Developer



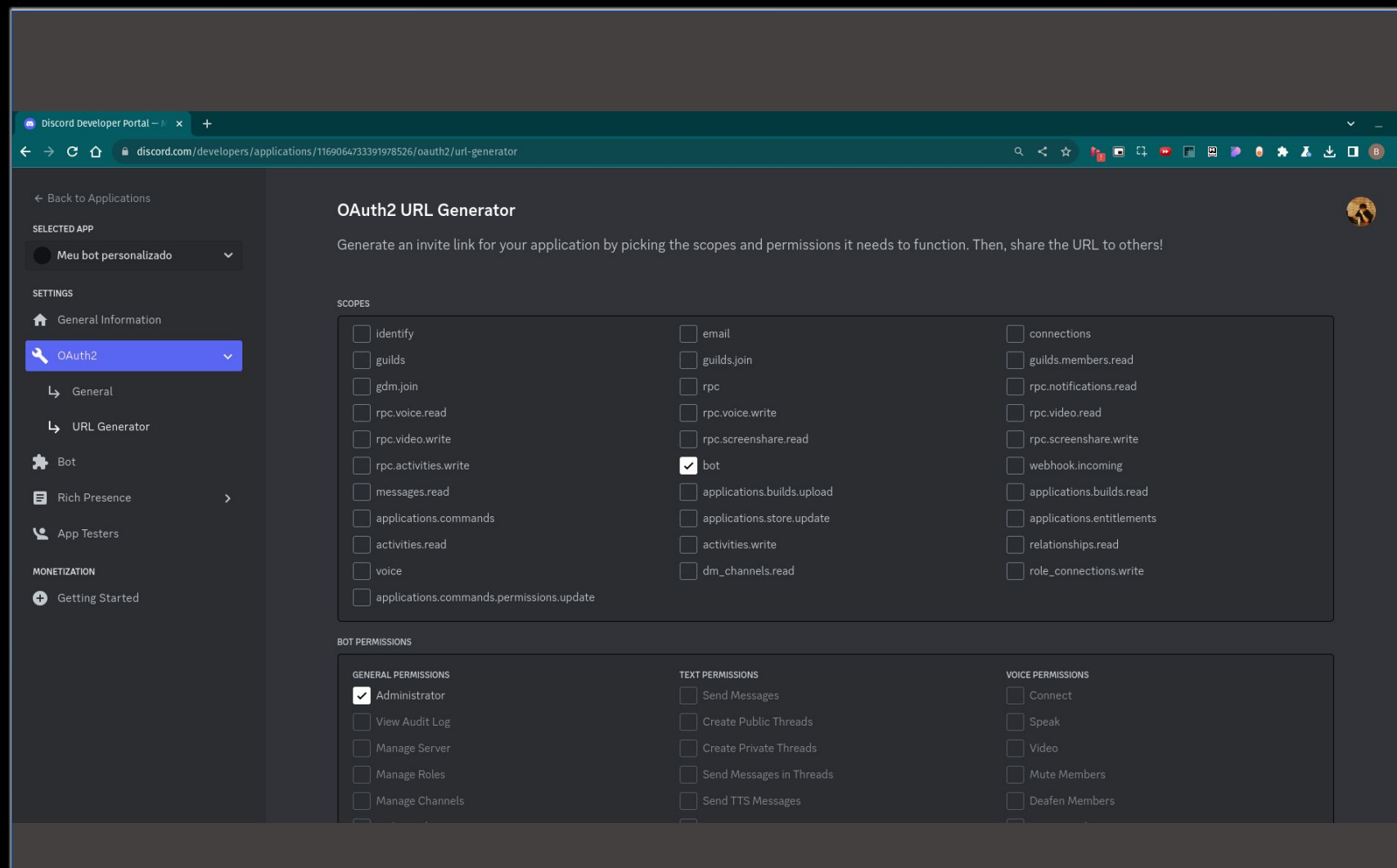
- Navegue até a aba BOT
- Faça o reset do seu Token
- Salve esse token para uso posterior

Discord Developer

- Desça a página
- Ative as opções de acesso a conteúdo privilegiado
- Salve as alterações



Discord Developer



- Acesse a aba OAuth2
- Selecione o escopo como BOT
- Selecione as permissões como “Administrator”
- Role até o final da página e copie o link.

Acessando esse LINK você poderá adicionar seu BOT em um canal do Discord.

Esses são os passos necessários para conseguir acesso programático ao Discord

Agora iremos testar este acesso ao bot com um código básico.

1. Clone o repositório e acesse a branch “python-br”.
2. Abra a pasta “passo_1” na sua IDE favorita.

Este código utiliza a biblioteca “Discord.py”.
Definiremos um prefixo para cada BOT.

**A documentação do Discord.py
também está no canal de Links.**

**Tire um tempo para tentar implementar
algum outro comando simples.**

Segundo passo

Agora que já temos nosso BOT do Discord funcionando corretamente, precisamos buscar dados do Jira de forma programática.

Para facilitar, estaremos disponibilizando os dados de uma conta do Jira já previamente criada e configurada com algumas tasks e sprints no quadro.

Para criar seu próprio quadro, basta criar uma conta no Jira e para gerar um token de acesso programático acesse o Link no Discord.

**Agora vamos implementar o código para
buscar os dados da API do Jira.**

Abra a pasta “passo_2”

**Tendo os conhecimentos individuais,
começamos a integrá-los**

Abra a pasta “passo_3”

**Agora que já temos os conteúdos básicos,
nossa missão é chegar ao bot final**

**Desenvolva um BOT que entregue um
report da sprint atual.**

DESAFIO

**Faça com que o relatório seja enviado
todo o dia no mesmo horário:
9h e 17h**

E agora ?

**O BOT é um projeto OpenSource
Ou seja, você pode contribuir, caso queira.**

**Para utilizá-lo na sua empresa, faça um Fork
ou somente clone o projeto atual e atualize
os envs.**

Obrigado!

Saiba mais sobre a Capyba no nosso site:

www.capyba.com

Trabalhamos com Python, deixe seu currículo na aba “Carrers”

www.capyba.com/careers