

# **PLANO DETALHADO – CRIAÇÃO DO BOT DE DISCORD**

**PLANO COMPLETO PARA DESENVOLVER UM BOT AVANÇADO DE DISCORD**

## **OBJETIVO:**

Criar um bot profissional com tickets, punições, registros, botões, logs e banco de dados, do zero ao avançado.

## **VISÃO GERAL:**

O cronograma abaixo é pensado para estudo consistente, sem excesso e sem enrolação. Focado no essencial.

## **CRONOGRAMA (15 DIAS)**

### **DIA 1 – Fundamentos de JavaScript (Parte 1)**

- Variáveis, tipos, operadores, conversões.
- Exercícios: Criar variáveis, manipular strings e números.

### **DIA 2 – Fundamentos de JavaScript (Parte 2)**

- Funções, escopo, retorno, parâmetros.
- Exercícios: Criar funções simples e modularizar código.

### **DIA 3 – Estruturas de Dados**

- Arrays, objetos, loops, condicionais.
- Exercícios: Criar lista de usuários e simular dados.

### **DIA 4 – Node.js Básico**

- npm, módulos, importar e exportar arquivos.
- Criar primeiro projeto Node.

### **DIA 5 – Preparação do Bot**

- Criar bot no Discord Developers.
- Instalar discord.js.
- Fazer o bot ligar e responder /ping.

## DIA 6 – Slash Commands

- Criar estrutura de comandos.
- Criar dois comandos reais e organizar pastas.

## DIA 7 – Interações Avançadas

- Botões, selects, modais.
- Criar um painel simples de testes.

## DIA 8 – Sistema de Tickets (Parte 1)

- Criar painel com botão.
- Gerar canais privados automaticamente.

## DIA 9 – Sistema de Tickets (Parte 2)

- Botão de fechar ticket.
- Logs ao fechar.

## DIA 10 – Sistema de Punições

- Comandos: ban, mute, warn.
- Criar logs automáticos.

## DIA 11 – Sistema de Registro

- Botão “Registrar”.
- Dar cargo automaticamente.

## DIA 12 – Banco de Dados (MongoDB)

- Instalar mongoose.
- Criar conexão e primeiro model.

## DIA 13 – Salvando Informações

- Salvar punições.
- Salvar tickets e configurações.

## DIA 14 – Organização Profissional

- Separar eventos, comandos, serviços.
- Criar utilitários e limpar o código.

## DIA 15 – Finalização e Revisão

- Testar tudo.
- Criar README.
- Subir para o GitHub.

## INSTRUÇÕES IMPORTANTES:

1. Nunca suba o token do bot para o GitHub.
2. Use pastas organizadas: /src, /commands, /events, /services.
3. Teste sempre pequena parte antes de juntar tudo.
4. Não pule a etapa de JavaScript, ela evita travamentos futuros.
5. Comece simples, só adicione funções quando o básico estiver sólido.

PRONTO.

Este cronograma é direto, objetivo e pensado para você evoluir sem depender de tutoriais.