

# Bet Bot - System Design Document

1 Semestre - 2023

## Sumário

1. Introdução .....	1
2. Visão geral do sistema .....	2
3. Arquitetura do sistema.....	2

## 1. Introdução

O propósito desse SDD (system design document) é alcançar o melhor entendimento sobre o sistema desenvolvido, dessa maneira, trazendo luz a cada componente do programa. Assim facilitando o desenvolvimento e manutenção para os programadores.

O bot irá ajudar o usuário a acompanhar o mundo de apostas esportivas no futebol, seu objetivo é que o usuário receba notificações diárias sobre os jogos e odds das ligas que ele acompanha, sendo possível adicionar mais ligas, excluir as que estão em sua lista, etc ...

O SSD explicará basicamente o que cada função dentro do repositório irá fazer, trazendo comparações com outros sistemas e explicando o porquê do uso de determinada ferramenta, arquitetura ou escolha.

Até o exato momento o bot não é totalmente funcional de acordo com seu objetivo. É preciso desenvolver e consertar a junção entre o scraping que traz as informações ao bot. O bot é funcional por si só, assim como o sistema de scraping também é.

## 2. Visão geral do sistema

O sistema é inspirado em grupos de bet no telegram, o bot não tem nenhuma pretensão de ser um robô de dicas, o projeto foi inspirado para trazer rapidez para aqueles que querem saber os jogos que ocorrerão diariamente, assim sendo uma espécie de assistente virtual aos apostadores.

Atualmente se encontra voltado ao mercado brasileiro por estar totalmente em português, porém é possível adicionar mais línguas ao bot, assim como mais funcionalidades.

## 3. Arquitetura do sistema

Nessa sessão irei falar mais sobre as escolhas de cada ferramenta, biblioteca e banco de dados.

### **Telegram Bot:**

O código tem duas bibliotecas relacionados ao bot, essas são: node-telegram-bot-api e @types/node-telegram-bot-api. A escolha dessa biblioteca foi por conta da simplicidade e ótima documentação da biblioteca.

Utiliza a chave de API do bot, tornando as interações com os usuários responsivas, dinâmicas e apropriadas conforme o comando enviado.

### **Database:**

PostgreSQL é o único banco de dados usado atualmente, sua função é guardar as principais informações dos usuários para que assim o bot consiga enviar as informações adequadas para cada caso.

O uso do PostgreSQL tem influência pela confiabilidade e consistência de informação que o banco de dados traz. Além da facilidade da administração do banco de dados.

### **ORM:**

É usado o Prisma como ORM, serve como ferramenta para tornar a lógica do projeto em algo seguro e conciso no banco de dados, executa as queries e leva as informações até a API.

A escolha do uso do prisma é por conta da simplicidade e confiabilidade para o desenvolvimento da API.

### **Typescript:**

O principal motivo pelo uso do typescript é a tipagem, organização e a indicação de erros que a linguagem garante, assegurar o retorno de um tipo específico em cada função garante que a API funcione da maneira correta.

## **4. Design de dados**

A estrutura de dados do PostgreSQL foi criada por esse modelo do prisma:

```
model User{
  id String @id @default(uuid())
  telegramId String @unique
  name String
  leagues String[]
}
```

Cada informação é usada para registrar o user e sua lista de campeonatos que este deseja acompanhar.

Os comandos do bot são basicamente um CRUD que atualiza a lista do usuário, e com essas informações o scraping é feito.