



# Proposta de Roteiro de Teste

API para estatísticas de jogo online



---

# Índice

O que eu vou testar

Requisitos funcionais e  
não funcionais

Linguagem

Objetivos da aplicação

Modelagem

Roteiro

Diário

Ferramentas

Datas



# O que eu vou testar

API REST do backend de uma aplicação para unificação e cruzamento de estatísticas de jogo digital





# Requisitos funcionais e não funcionais

Identificação	Nome	Descrição
RF01	Buscar jogos	Listar os jogos de um jogador específico, por uma quantidade de no máximo 100 jogos
RF02	Buscar jogadores relacionados	Listar os jogadores que já jogaram com um jogador específico, tanto como parceiros quanto inimigos
RF03	Listar taxa de vitórias contra jogadores.	Listar os jogadores que já foram relacionados com um jogador específico, e saber a taxa de vitórias e derrotas contra eles e taxa de vitórias e derrotas em conjunto com eles
RF04	Listar taxa de vitórias geral	Buscar a taxa de vitórias gerais de jogos de um jogador específico.
RF05	Listar taxa de derrotas geral	Buscar a taxa de derrotas gerais de jogos de um jogador específico.
RF06	Listar taxa de mortes, assassinatos e assistências	Listar taxa de mortes, assassinatos e assistências dos jogos de um jogador específico
RF07	Listar taxa de escolha de campeões	Listar a taxa de escolha de campeões, dos jogos de um jogador específico
RF08	Listar partidas por dia	Listar quantas partidas em um dia jogou um jogador específico.



# Requisitos não funcionais

**Identificação**

**Nome**

**Descrição**

RNF01

Acesso a API externa

O acesso a API externa deve estar com disponibilidade de acesso em 99% do tempo.



# Linguagem

Javascript (Typescript) - Utilizando Node.js com fastify





# Quais são os objetivos da aplicação?

Listagem de estatísticas do jogo online League of Legends, buscando os dados da API oficial da empresa distribuidora do jogo.







# Modelagem

Até o momento não. A primeira versão da aplicação vai devolver os dados calculados por demanda em memória do servidor através de um bot da plataforma Discord (formato serverless).







# Técnica de testes

- **Testes unitários:** testam o comportamento do código, testam as funções e regras de negócio de forma individual e interdependente de atores externos. São os testes mais básicos e um padrão cada vez mais presente no dia a dia de desenvolvimento devido a sua importância.
  1. **Funções e Lógica de Negócios:** Testar as funções e lógica de negócios que não dependam diretamente da camada de rede ou da estrutura da API.
  2. **Validações de Entrada:** Verificar se as funções de validação de entrada estão funcionando corretamente.
  3. **Tratamento de Erros:** Testar como a API lida com erros e exceções. Certifica de que os erros sejam tratados de maneira apropriada, com mensagens de erro adequadas e códigos de status HTTP corretos.
  4. **Mapeamento de Rotas:** Testar se as rotas estão mapeando para as funções corretas e se os dados estão sendo passados corretamente.



# Roteiro

Os testes serão feitos com a aplicação já pronta, seguindo os requisitos funcionais em ordem e qualquer outra especificação que seja interessante validar conforme necessário durante o desenvolvimento dos casos de teste unitário;





# Diário de Testes

O diário vai ter os registros em uma planilha do excel.





# Ferramentas de teste

Será utilizado o Jest para criação dos testes na api, e integração dos testes na pipeline de deploy (Github actions)





# Datas

- Os testes serão criados a partir do dia 9/10/2023, e espera-se que todos os casos de testes para os requisitos funcionais e não funcionais estejam finalizados até dia 01/11/2023.
- Entrega do diário de testes no dia 07/11/2023.





# Proposta de Roteiro de Teste

API para estatísticas de jogo online

