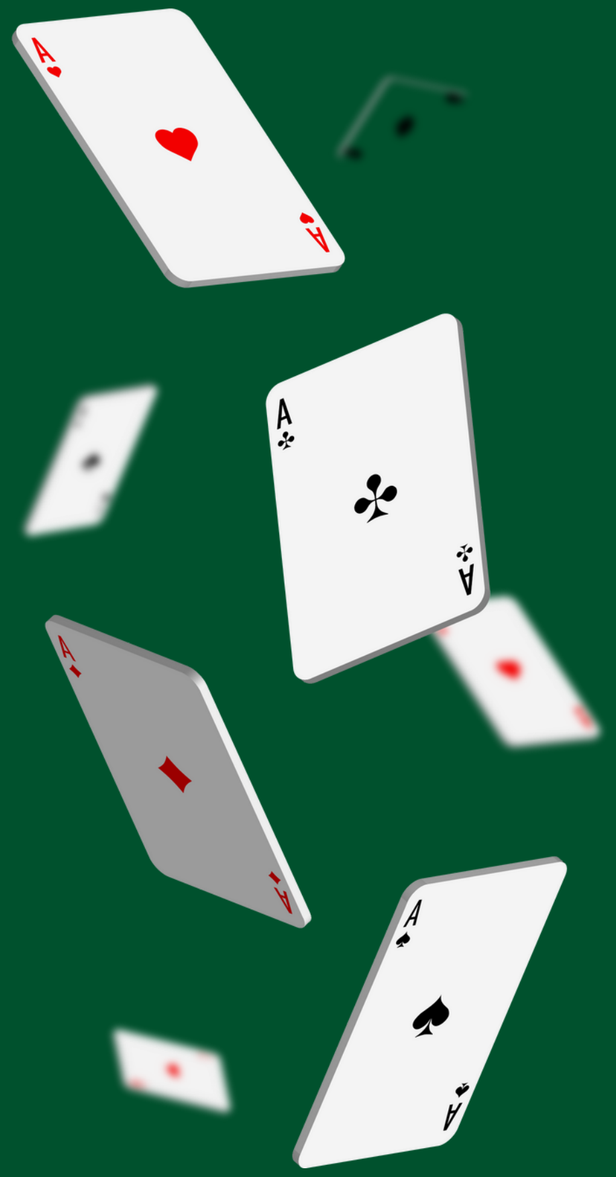
TRUCO ONLINE

Daniel Nogueira
Felipe Moreira
João Pedro Franco
Lucas Bernardes



Proposta de Desenvolvimento

Software Concorrente e Distribuído

Problema / Necessidade

"O homem é um animal político"
- Aristóteles

Essa máxima caracteriza o ser humano com um ser social que, por natureza, precisa permanecer a um

Problema / Necessidade

É inegável que da mesma forma como uma partida de futebol reune amigos no final de semana para socializarem e terem bons momentos uma partida de truco também.

Portanto, seguindo a máxima de Aristóteles uma forma do ser humano expressar sua natureza é por meio de uma partida de truco.

IDEIA DO PROJETO

Uma aplicação web que possibilita grupos de pessoas jogarem o famoso jogo de baralho truco de forma remota.

Desenvolver um truco com as regras goianas permite compartilhar novas maneiras de se jogar.

Público alvo: Jovens, estudantes universitários, pessoas que gostam do jogo ou que estão dispostas a conhecê-lo

MERCADO

O mercado global de jogos online está em constante crescimento, principalmente no período pós pandemia, ocorreu uma explosão no número competidores em plataformas online.

O truco é um dos jogos de cartas mais populares na América Latina, com uma base de jogadores apaixonados e uma rica tradição cultural associada ao jogo.

Mais de 1 milhão de downloads em outros aplicativos com o jogo truco.

O PROJETO E A DISCIPLINA

Distribuição: A distribuição é clara pela arquitetura, onde poderemos, por exemplo, instanciar diversas APIs e escalar horizontalmente como for necessário

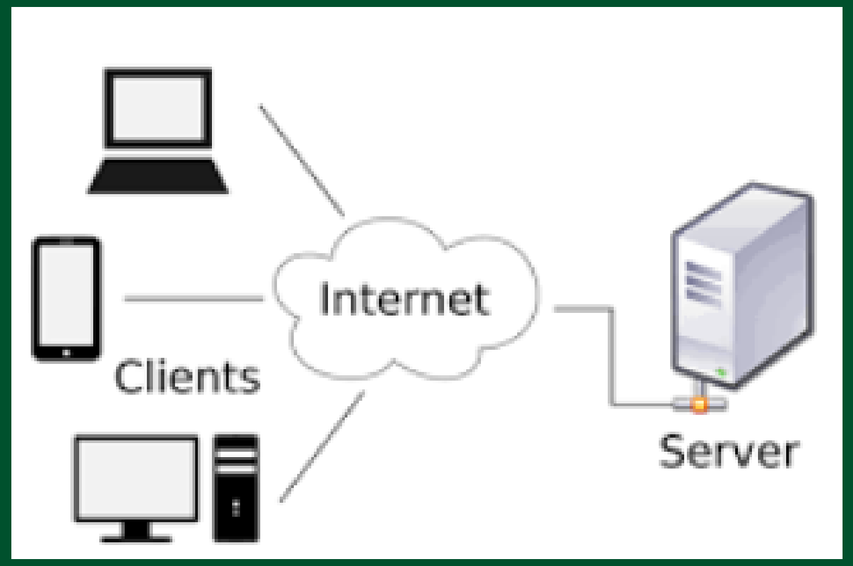
---

Concorrência: Durante a partida a concorrência se dá no monte de cartas (recurso), onde os jogadores (que funcionam como threads) só poderão ter acesso a esse recurso quando for a sua vez. Nesse caso, ainda há uma necessidade de ordenação

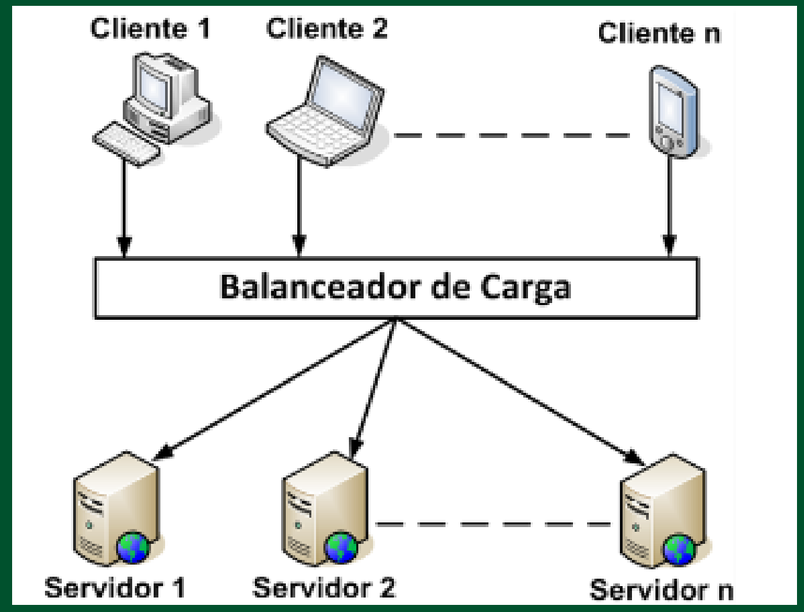
O PROJETO E A DISCIPLINA

- Hospedagem em nuvem
- Uso de conteiners, possibilitando facilidade para escalabilidade
- Cada partida é independente entre si, um mesmo servidor roda instâncias de partidas em paralelo
- Balanceador de carga
- CI/CD

O PROJETO E A DISCIPLINA

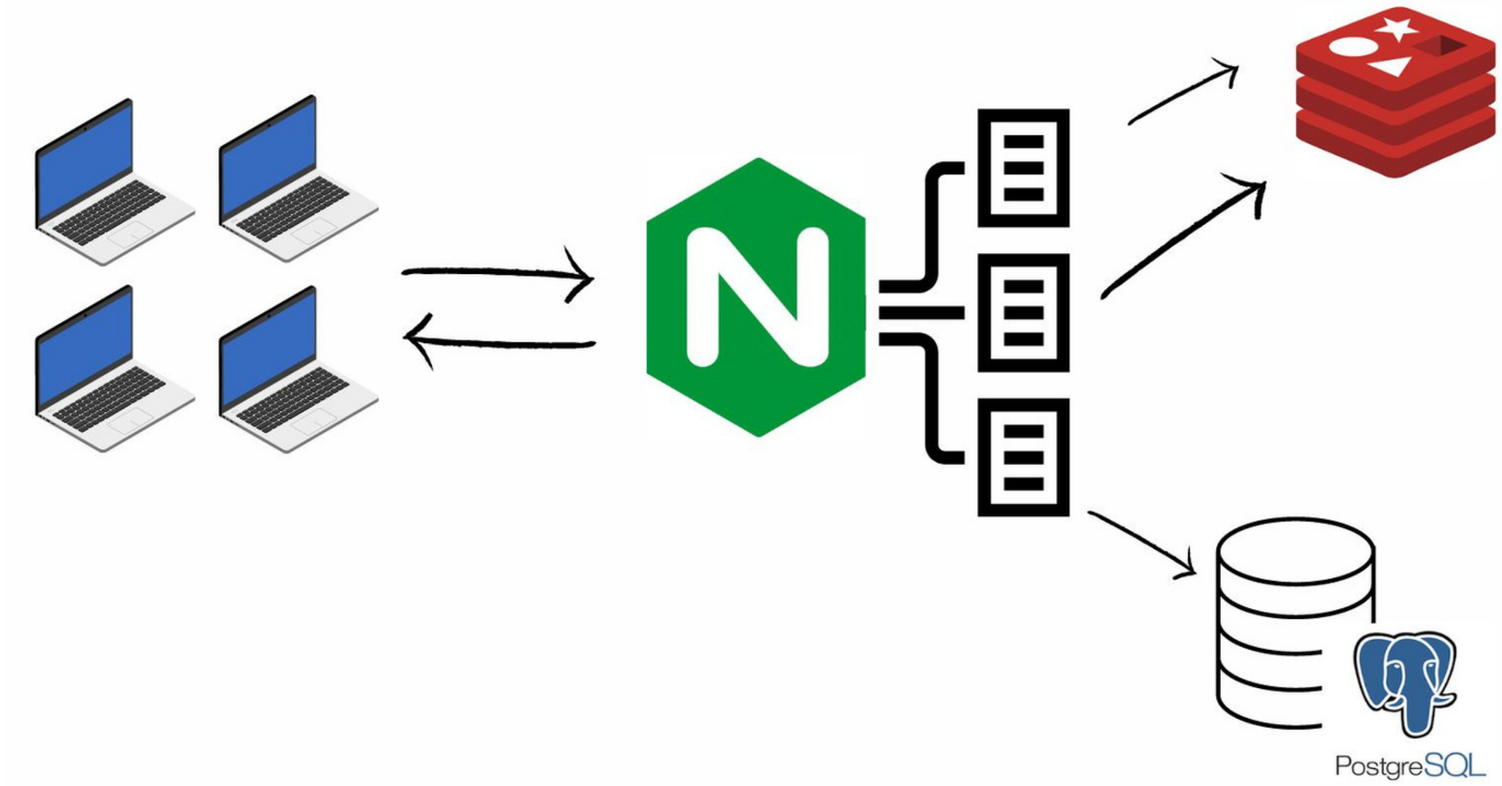


Usuários conectados



Balanceamento de carga, permitindo escalabilidade dinâmica

ARQUITETURA



ARQUITETURA

Load Balancer (NGinx)

Teremos um balanceador de carga que irá distribuir as requisições entre as N instâncias dos serviços. Assim poderemos ter quantas instâncias forem necessárias, escalando horizontalmente facilmente

API REST

Teremos uma API REST que irá ter a responsabilidade de manipular dados com maior ciclo de vida

ARQUITETURA

Banco de dados relacional

O banco de dados PostgreSQL será utilizado como banco de dados relacional para armazenamento dos dados

Banco de dados de cache (Redis)

O banco de dados Redis será utilizado para armazenar o estado das partidas, sendo ele necessário para ser uma cache compartilhada entre as instâncias da API de Lobbies

ARQUITETURA

API de Lobbies (Web Socket)

Essa API terá as conexões dos usuários e fará o transporte em tempo real dos dados da partida para os devidos jogadores, utilizando o Redis como cache

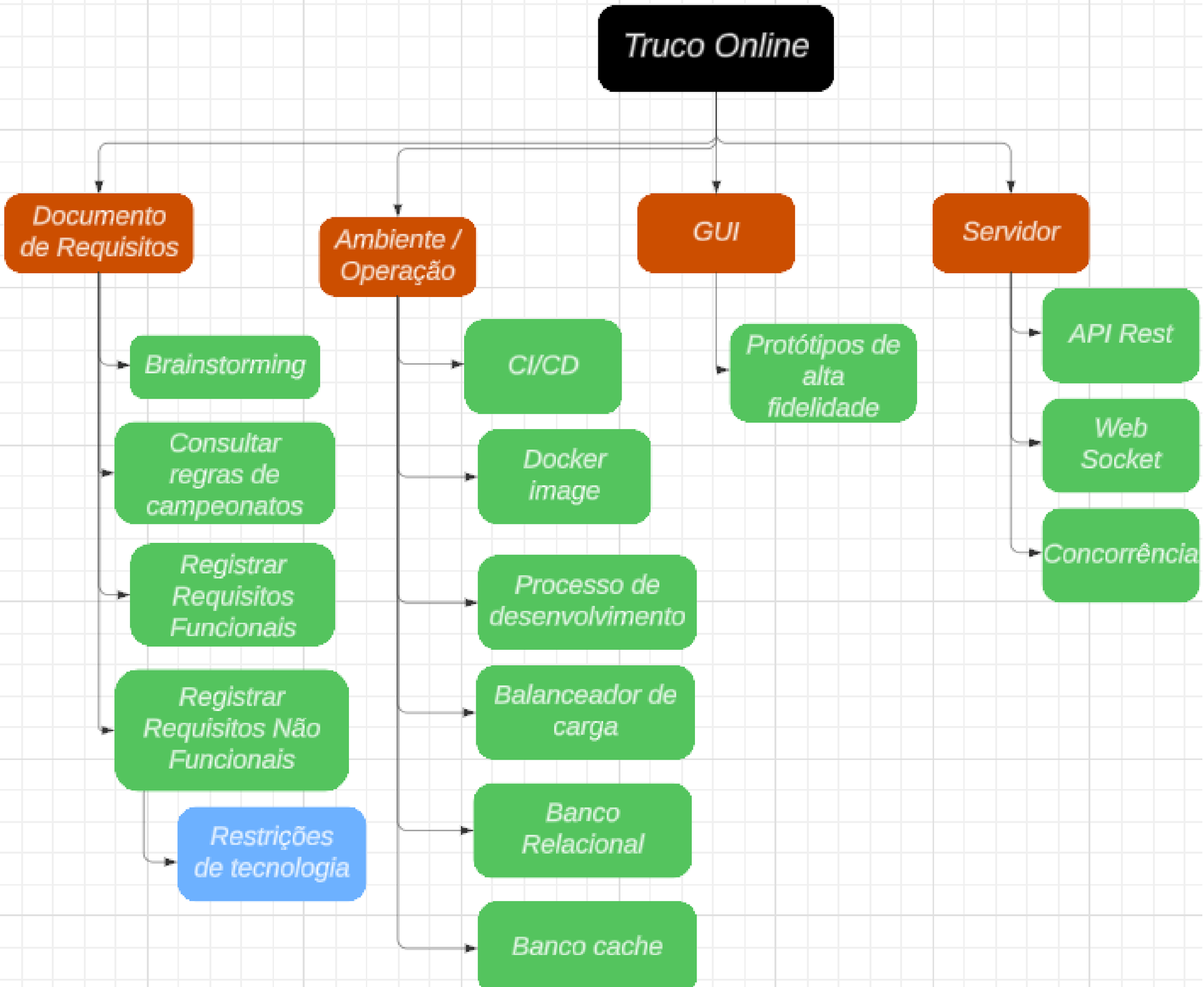
Frontend

O frontend será uma página web simples para ser acessada através de qualquer navegador

Equipe

- João - Líder, Gerente de projeto, desenvolvedor backend
- Daniel - Desenvolvedor backend
- Lucas - Desenvolvedor frontend
- Felipe - Desenvolvedor backend

Tarefas (WBS)


Truco Online
Documento de Requisitos
Ambiente / Operação
GUI
Servidor
Brainstorming
Consultar regras de campeonatos
Registrar Requisitos Funcionais
Registrar Requisitos Não Funcionais
Restrições de tecnologia
CI/CD
Docker image
Processo de desenvolvimento
Balanceador de carga
Banco Relacional
Banco cache
Protótipos de alta fidelidade
API Rest
Web Socket
Concorrência

OBRIGADO!