

Especificação do Trabalho de Software Concorrente e Distribuído

Alunos:

Matheus de Moura Rosa - 202004768

Rafael Estanislau Moraes dos Santos - 202004777

Pedro Henrique Chaves Junqueira - 202000303

Mateus Silva de Sousa - 201802778

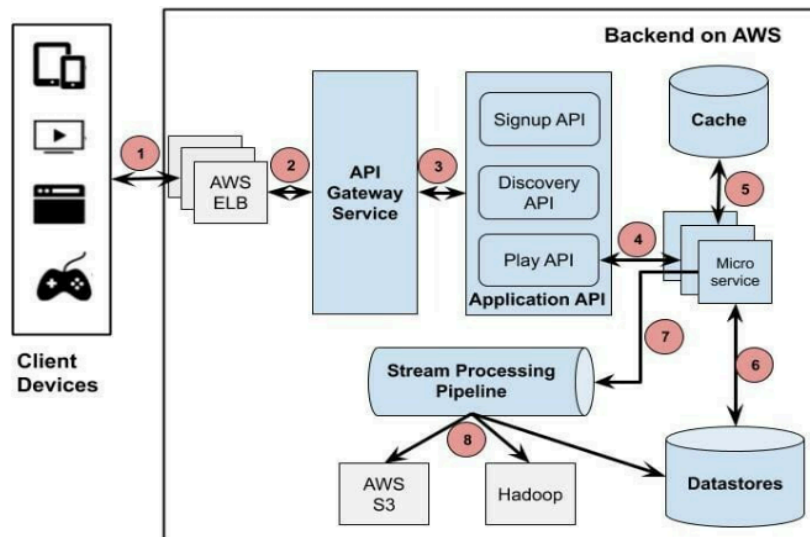
O que é o software?

A ideia é aplicar sistema de software concorrente e distribuído empregado em uma plataforma de streaming de vídeos, como uma alternativa robusta aos modelos centralizados existentes.

Como aplicar o contexto de software Concorrente e Distribuído?

Nesse contexto, cada componente do sistema, como servidores de armazenamento, servidores de transmissão e até mesmo os clientes finais, poderiam operar de forma distribuída e concorrente. Por exemplo, múltiplos servidores de streaming distribuídos geograficamente poderiam armazenar e transmitir conteúdo simultaneamente, permitindo uma melhor distribuição de carga e garantindo uma experiência de visualização mais estável para os usuários finais, especialmente em momentos de alta demanda. Além disso, uma abordagem distribuída poderia oferecer escalabilidade dinâmica, permitindo que a infraestrutura se ajuste automaticamente à demanda variável de usuários e conteúdo. Ao adotar a concorrência e a distribuição de tarefas, o sistema poderia otimizar a utilização de recursos e melhorar a tolerância a falhas, proporcionando uma plataforma de streaming mais resiliente e eficiente para os usuários.

Arquitetura exemplo do BackEnd Software



!!!!!!!!!!!!!!!!TÓPICOS SOBRE O TRABALHO!!!!!!!!!!!!!!!!!!!!

Introdução e Contextualização

Ao abranger uma variedade de jogos e gêneros, pretendemos criar uma comunidade engajada, que contemple o ambiente competitivo de esports e promova o aprendizado e o crescimento de jogadores de todos os níveis de habilidade. Queremos inspirar a próxima geração de talentos nos esports, capacitando-os com o conhecimento e as ferramentas necessárias para competir em um nível mais alto e apreciar o jogo em um nível mais profundo, desde jogadores que queiram levar os esports como carreira profissional aos que desejam apenas jogar partidas com maior nível de conhecimento e ranquear no topo de tabelas como dedicação ao seu hobby.

Proposta de Valor

Oferta de conteúdo educativo de alta qualidade para jogadores de todos os níveis. Diversidade de jogos e gêneros abordados, incluindo títulos populares como League of Legends, Valorant, CS:GO, entre outros.

Público-Alvo

Descrição do público-alvo, incluindo jogadores iniciantes, entusiastas de esports e espectadores interessados em aprender mais sobre os jogos competitivos.

Conteúdo

Explicação dos tipos de vídeos oferecidos, como tutoriais, análises de partidas, dicas avançadas e entrevistas com jogadores profissionais.

Plataforma e Tecnologia

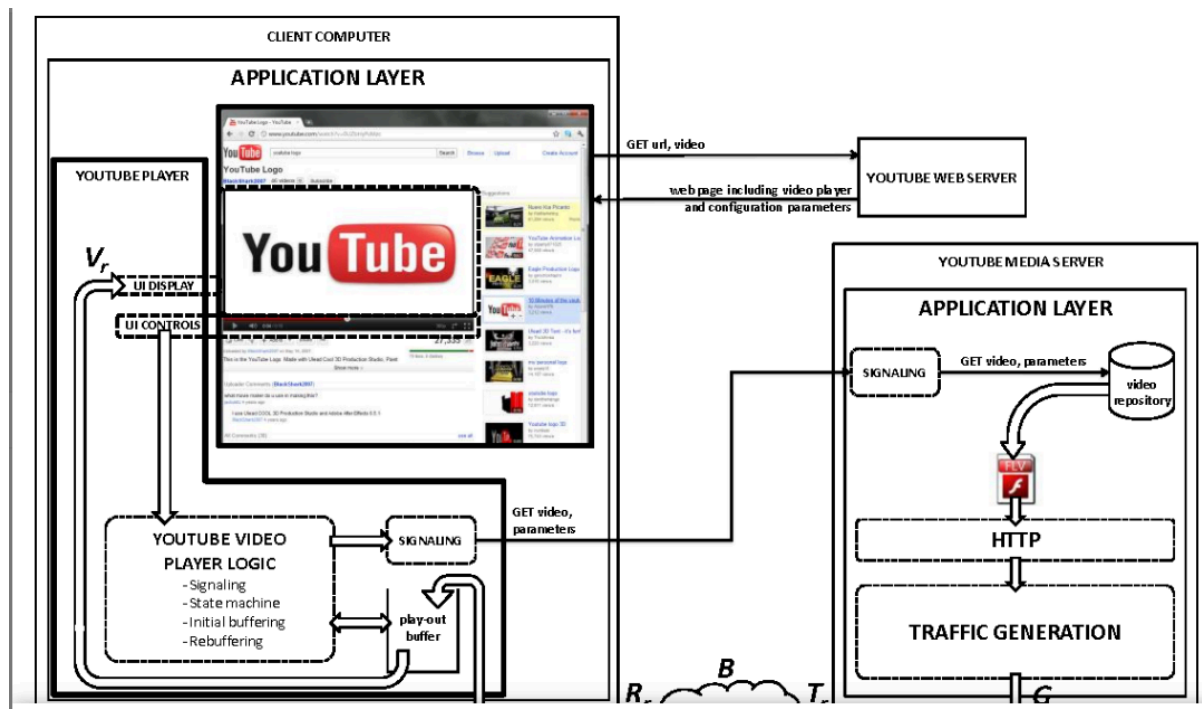
Descrição da plataforma do site, destacando recursos como divisão por níveis de conhecimento de jogo, interatividade e tecnologias utilizadas para melhorar a experiência do usuário.

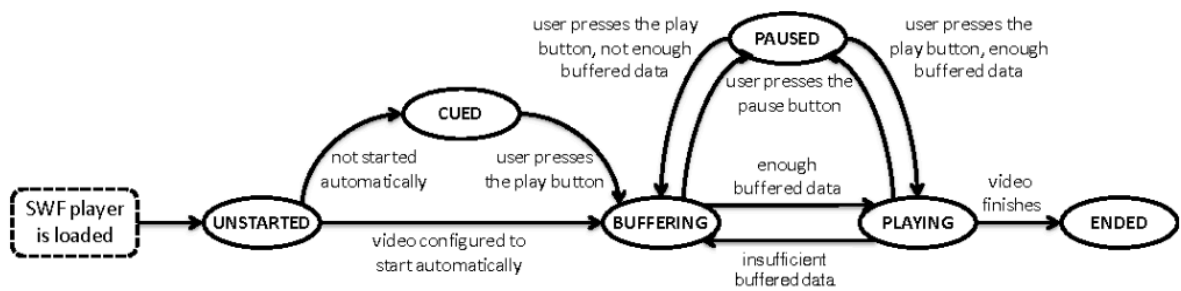
Monetização

Estratégia de monetização, incluindo assinaturas, publicidade e parcerias com marcas de jogos.

Equipe

Apresentação da equipe por trás do site, podemos incluir jogadores experientes para a criação de conteúdos, editores de vídeos e desenvolvedores do site.





Sugestão de funcionamento:

Divisão do Vídeo em Chunks: O vídeo é dividido em pequenas partes, chamadas de chunks, que podem ser de alguns segundos de duração.

Solicitação Assíncrona de Chunks: Quando o vídeo começa a ser reproduzido, o player faz uma solicitação assíncrona para o servidor para obter o próximo chunk do vídeo.

Download do Chunk: O servidor responde à solicitação com o chunk solicitado, que é então baixado pelo player em segundo plano.

Armazenamento em Buffer: O chunk baixado é temporariamente armazenado em um buffer no dispositivo do usuário.

Reprodução Contínua: Enquanto o vídeo é reproduzido, o player verifica se há chunks suficientes no buffer para reprodução contínua.

Pré-baixamento de Chunks: Para evitar interrupções na reprodução, o player pode pré-baixar alguns chunks adicionais, à medida que o vídeo é reproduzido, para garantir que haja sempre chunks suficientes no buffer.

