

Grupo: Quebrando códigos
Software: Skillforge

1) Overview do Projeto

1.1) Instrutor

O instrutor responsável pelo projeto é o professor Sérgio Teixeira de Carvalho, de Software Concorrente e Distribuído. Com vasta experiência em arquitetura de sistemas distribuídos e tecnologias de streaming, ele guiou o desenvolvimento da plataforma com foco em performance e escalabilidade.

1.2) Contextualização

Este projeto envolve a criação de uma plataforma de streaming de conteúdo educacional focado em jogos. Inspirando-se em plataformas populares como Netflix e Prime Video, o objetivo é proporcionar uma experiência de usuário fluida e eficiente, utilizando tecnologias de ponta para distribuição de conteúdo em escala global. A plataforma foi construída utilizando o CloudFront como CDN (Content Delivery Network) e o S3 como serviço de armazenamento de mídia, visando otimizar a entrega de vídeos e reduzir a latência.

1.3) Objetivos do Projeto

- **Utilizar uma base arquitetural similar às plataformas de streaming comerciais:** Aproveitar o conhecimento e as tecnologias aplicadas em serviços como Netflix e Prime Video para criar uma plataforma de streaming educacional.
- **Implementar requisições parciais de dados:** Desenvolver uma API capaz de calcular e fornecer apenas a quantidade de bytes necessária para exibir o timestamp desejado dos vídeos, melhorando a experiência do usuário.
- **Reduzir o tempo de carregamento dos vídeos:** Implementar técnicas que minimizem a espera para o carregamento dos vídeos, especialmente os de longa duração, através de requisições parciais.
- **Utilizar o CloudFront como CDN para escala global:** Garantir que o conteúdo esteja disponível com baixa latência em qualquer parte do mundo, utilizando o cache nas edge locations do CloudFront.

1.4) Discussões de Relevância

- **Eficiência e performance:** A abordagem de requisições parciais de dados é crucial para otimizar a performance, especialmente para vídeos longos, garantindo uma experiência de usuário satisfatória.
- **Escalabilidade global:** A utilização do CloudFront como CDN permite que a plataforma atenda usuários de diferentes regiões com baixa latência, independentemente da localização geográfica do servidor principal.

- **Tecnologia e inovação:** O projeto demonstra a aplicação prática de tecnologias modernas de streaming e armazenamento em nuvem, servindo como um estudo de caso relevante para desenvolvedores e arquitetos de sistemas.
- **Educação acessível:** Ao disponibilizar conteúdo educacional de forma eficiente e escalável, a plataforma contribui para a democratização do conhecimento, permitindo que mais pessoas acessem materiais educativos de qualidade sobre jogos, independente de sua localização.

1.5) Trabalhos relacionados

- Não foi encontrado nenhum trabalho semelhante disponibilizado em sites de código aberto como por exemplo o Github.

2) Requisitos do Projeto

2.1) Requisitos de Usuários

- **RU01 Acesso fácil ao conteúdo educacional:** Os usuários devem poder navegar e acessar o conteúdo educacional de maneira intuitiva e rápida.
- **RU02 Carregamento rápido de vídeos:** Os vídeos devem carregar rapidamente, minimizando o tempo de espera.
- **RU03 Reprodução de vídeos de alta qualidade:** A plataforma deve oferecer vídeos em alta qualidade, ajustando a resolução conforme necessário para garantir uma experiência de visualização fluida.
- **RU04 Disponibilidade global do conteúdo:** Usuários de qualquer parte do mundo devem ter acesso ao conteúdo com baixa latência.
- **RU05 Capacidade de continuar de onde parou:** Os usuários devem ser capazes de retomar a reprodução dos vídeos a partir do ponto onde pararam.

2.2) Requisitos Funcionais

- **RF01 Implementação de requisições parciais de dados:** A API deve calcular a quantidade de bytes necessária para exibir apenas o timestamp desejado dos vídeos.
- **RF02 Integração com S3 para armazenamento de mídia:** Os vídeos devem ser armazenados no S3, com acessos gerenciados de forma segura.
- **RF03 Utilização do CloudFront como CDN:** O CloudFront deve ser configurado para distribuir o conteúdo em escala global, utilizando edge locations para armazenar em cache e entregar os vídeos com baixa latência.
- **RF04 Interface de usuário intuitiva:** A plataforma deve ter uma interface amigável e fácil de usar, facilitando a navegação e a busca de conteúdo.
- **RF05 Sistema de autenticação e autorização:** Deve existir um sistema para autenticar usuários e controlar o acesso ao conteúdo.

2.3) Requisitos Não-Funcionais

- **RNF01 Desempenho:** A plataforma deve ter alta performance, garantindo tempos de resposta rápidos e carregamento eficiente de vídeos.
- **RNF02 Escalabilidade:** A solução deve ser escalável para suportar um grande número de usuários simultâneos, sem degradação na qualidade do serviço.
- **RNF03 Segurança:** Dados dos usuários e conteúdos devem ser protegidos contra acesso não autorizado. A comunicação entre cliente e servidor deve ser criptografada.
- **RNF04 Confiabilidade:** A plataforma deve ser confiável, com alta disponibilidade e mecanismos de recuperação em caso de falhas.
- **RNF05 Manutenibilidade:** O sistema deve ser fácil de manter e atualizar, permitindo a adição de novas funcionalidades e correções de bugs de forma eficiente.
- **RNF06 Compatibilidade:** A plataforma deve ser compatível com diversos dispositivos e navegadores, oferecendo uma experiência consistente para todos os usuários.

3) Fundamentos de SD relacionados ao projeto

3.1) Princípios de sistemas distribuídos

- **Escalabilidade:** O sistema deve ser capaz de lidar com um número crescente de usuários e dados sem perder performance. A utilização do CloudFront como CDN e o armazenamento no S3 ajudam a alcançar esse objetivo.
- **Tolerância a Falhas:** A plataforma deve continuar funcionando mesmo quando partes do sistema falham. A replicação de dados e o uso de múltiplas edge locations no CloudFront ajudam a garantir alta disponibilidade.
- **Desempenho:** É crucial que o sistema responda rapidamente às requisições dos usuários. O buffer implementado para os vídeos e a distribuição dos servidores através da CDN melhoram significativamente o tempo de resposta e a qualidade do streaming.

3.2) Fundamentos de arquitetura de sistemas distribuídos e dos estilos arquiteturais

- **Arquitetura Cliente-Servidor:** A plataforma segue um modelo cliente-servidor, onde o cliente faz requisições à API para acessar o conteúdo armazenado nos servidores.
- **Arquitetura em Camadas:** O sistema é organizado em camadas, separando a lógica de apresentação (interface do usuário), lógica de aplicação (API), e a lógica de dados (armazenamento em S3 e distribuição via CloudFront).

3.3) Fundamentos de paradigmas de comunicação em sistemas distribuídos

- **Comunicação Assíncrona:** A API deve suportar requisições assíncronas, permitindo que o cliente continue operando enquanto aguarda a resposta do servidor.

3.4) Robustez em sistemas distribuídos: nomeação, coordenação, consenso, consistência e replicação, e tolerância a falhas

- **Nomeação:** Utilizamos URLs únicas e consistentes para acessar os recursos, garantindo que o conteúdo esteja sempre acessível de maneira previsível.
- **Coordenação e Consenso:** Para garantir que os servidores CDN estejam sempre atualizados com o conteúdo mais recente, utilizamos mecanismos de coordenação e consenso, como algoritmos de consenso distribuído.
- **Consistência e Replicação:** Os dados são replicados em múltiplas localizações para garantir alta disponibilidade e consistência. O CloudFront cuida da replicação dos vídeos nas edge locations.
- **Tolerância a Falhas:** O sistema é projetado para ser tolerante a falhas, utilizando replicação de dados e estratégias de failover para minimizar o impacto de falhas em componentes individuais.

4) Resultados

4.1) Design arquitetural

Camada de Apresentação (Frontend):

- **Interface de Usuário:** Desenvolvida com React, a interface oferece uma experiência de navegação intuitiva e responsiva, permitindo aos usuários acessar facilmente o conteúdo educacional.
- **Player de Vídeo:** Um player customizado que suporta buffering eficiente e controle de qualidade de vídeo, garantindo uma reprodução suave e de alta qualidade.

Camada de Aplicação (Backend):

- **API RESTful:** Implementada utilizando Node.js com Express, a API gerencia requisições de usuários, autenticação, autorização e controle de acesso aos vídeos armazenados no S3.

Camada de Dados:

- **Armazenamento em Nuvem (S3):** Os vídeos são armazenados no Amazon S3, garantindo escalabilidade e segurança. O S3 facilita a gestão de grandes volumes de dados e oferece alta durabilidade e disponibilidade.
- **Content Delivery Network (CDN):** O Amazon CloudFront distribui os vídeos globalmente, utilizando edge locations para reduzir a latência e melhorar a performance de entrega de conteúdo.

4.2) Design dos dados

Estrutura de Armazenamento:

- **Bucket S3:** Os vídeos são armazenados em buckets do S3, organizados por categorias e níveis de acesso. Cada vídeo é dividido em segmentos para facilitar o buffering e a entrega eficiente.

5) Limitações, trabalhos futuros e perspectivas do Projeto

5.1) Limitações

Embora a plataforma de streaming educacional tenha sido projetada com foco em performance e escalabilidade, algumas limitações ainda podem ser observadas:

- **1. Latência em Regiões Remotas:** Apesar do uso do CloudFront, algumas regiões muito remotas podem ainda experimentar latências maiores devido à distância física das edge locations.
- **2. Dependência de Serviços da AWS:** A plataforma depende fortemente dos serviços da AWS, como S3 e CloudFront. Isso pode limitar a flexibilidade e aumentar os custos, especialmente se a escala de uso aumentar significativamente.

5.2) Trabalhos Futuros

- **1. Implementação de Comunidade:** Desenvolver uma seção dedicada à comunidade dentro da plataforma, onde os usuários podem interagir, compartilhar experiências, e colaborar em projetos relacionados a jogos.
- **2. Sistema de Comentários:** Adicionar a funcionalidade de comentários nos vídeos e cursos, permitindo que os usuários façam perguntas, compartilhem opiniões e troquem ideias sobre o conteúdo educacional.
- **3. Funcionalidade de Interação:** Incorporar funcionalidades de interação, como fóruns de discussão, grupos de estudo e eventos ao vivo, onde os usuários podem participar de webinars e sessões de Q&A com instrutores.
- **4. Gamificação:** Introduzir elementos de gamificação, como badges, níveis e recompensas, para incentivar a participação ativa e o engajamento dos usuários na comunidade e nos cursos.
- **5. Perfis de Usuários:** Permitir que os usuários criem perfis personalizados, onde possam exibir suas conquistas, cursos concluídos e áreas de interesse, facilitando a conexão com outros membros da comunidade com interesses semelhantes.

