

Aluno/a: _____ Mat: _____

Terceira Avaliação - 2023-1

1. Descrição do Relatório Técnico

O projeto constitui de uma atividade que precisa ser completada através de duas tarefas: (i) implementação e (ii) relatório. A primeira tarefa envolve-se com a construção do projeto que deve ser acompanhado de um relatório (conforme instruções) no formato de Artigo Técnico modelo SBRC. Os artigos podem ser escritos em português ou inglês e devem ser submetidos somente no formato PDF. Cada relatório não tem limite de páginas. Os trabalhos devem ser formatados seguindo o modelo de artigos da SBRC, disponível em [\[https://www.sbc.org.br/documentos-da-sbc/category/169-templates-para-artigos-e-capitulos-de-livros\]](https://www.sbc.org.br/documentos-da-sbc/category/169-templates-para-artigos-e-capitulos-de-livros) (Modelos para Publicação de Artigos). O relatório deverá conter os tópicos abaixo discriminados:

- **Título, Autores:** dispostos na parte superior da página, de modo que logo em seguida venha o resumo do projeto.
- **Implementação e Testes:** importante considerar a escrita para alguém não familiar com o programa e que o mesmo consiga entender o que foi feito. Na sequência, resultados obtidos da execução do programa devem ser listados e apresentados. Como mencionado anteriormente, a forma mais apropriada para apresentar os dados coletados é no formato de tabela.
- **Análise e Resultados da Implementação:** nesta seção, os resultados apresentados anteriormente serão analisados, discutidos e explicados, ou seja, algo que faça sentido deverá ser apresentado. Ela poderá se referir aos propósitos do experimento. Lembre-se também de explicar anomalias encontradas nos resultados. Tendências nos resultados devem ser evidenciadas bem como explicadas. De modo geral, esta seção oferece ao leitor uma explanação de como o programa agiu e se comportou segundo os propósitos descritos e contemplados na programação. Gráficos são excelentes ferramentas para ilustrar os resultados e apoiar a escrita.

- **Conclusão:** deve reforçar a informação já apresentada pelo estudante, bem como apresentar resumidamente o experimento. Adicionalmente, incluir uma ou duas sentenças que respondam às seguintes perguntas:
 - Qual o propósito do experimento?
 - Tal propósito foi alcançado?
 - Quais foram os erros e dificuldades? Quais foram superados e como?
 - Quais foram os resultados do experimento?
 - Que informação foi descoberta com a execução dos programas?
 - O que foi aprendido com a realização dos experimentos?
 - O que se pode fazer para melhorar?
- **Apêndice:** deverá conter fragmento dos códigos fontes que foi produzido como parte do experimento e considerado pelos autores pontos-chaves de sua solução e suporte para explicação do texto.

2. Descrição do Projeto

A segunda tarefa refere-se ao entendimento, projeto, implementação e execução do programa a ser desenvolvido que trata de assuntos cobertos em sala de aula e na teoria.

Rede de Distribuição de Conteúdo - CDN (*Content Delivery Network*) é uma Rede de Distribuição de Conteúdo é um grupo de servidores que permite que os conteúdos da *Internet* estejam facilmente disponíveis, com rapidez e segurança. A rede é a responsável por melhorar a experiência do usuário enquanto usa os recursos dela de forma eficiente. Algumas marcas conhecidas de CDN são Cloudflare, Akamai, Incapsula e MaxCDN. Benefícios de usar uma CDN: (a) Melhorias na velocidade; (b) Melhor disponibilidade do conteúdo; (c) custo-benefício; Uma CDN é importante tanto para os usuários quanto para o provedor de conteúdo.

Problema de *Streaming* - *Streaming* é a tecnologia que permite consumir filmes, séries e músicas em qualquer lugar, é bastante popular e acessível e vem ajudando a combater a pirataria.

Desafio deste Projeto - Este projeto compreende o desenvolvimento de uma rede de distribuição de *streaming* segundo a arquitetura a seguir (Figura 1). Na arquitetura temos os seguintes componentes:

1. **Servidor de Stream:** É o responsável pelo conteúdo e, consequentemente, é o criador do *stream*. Um *stream* é um fluxo contínuo de dados. Os fluxos são criados por uma taxa de X quadros por segundo.
2. **Orquestrador:** É o principal elemento dentro da arquitetura. Na arquitetura ele é representado como um único componente, mas analisando-o com mais detalhes ele assumirá as seguintes funções:
 - a. Atender a vários servidores de Stream. Os servidores devem conectar ao orquestrador através de uma conexão e após a conexão, o orquestrador deve criar novos servidores de vídeo para redistribuir os *streams*.
 - b. Servidores de Vídeo. Um servidor de vídeo é implantado dentro de uma borda da rede. O objetivo dele é evitar que o servidor de stream seja sobrecarregado com muitas conexões. Sua missão é atender a clientes que estão próximos deles. Cabe ao Orquestrador informar aos clientes quais os servidores de vídeo são os mais adequados.
 - c. Redireciona vários streams a vários clientes de vídeo (**CV**). Os clientes não sabem da localização dos Servidores de Streaming, ele são redirecionados para servidores de vídeos que possuem a melhor condição de atendê-los, tais servidores são *proxies*, ou seja, atuam para melhorar a qualidade da experiência, assim, evitam a sobrecarga dos servidores primários (origem dos dados).

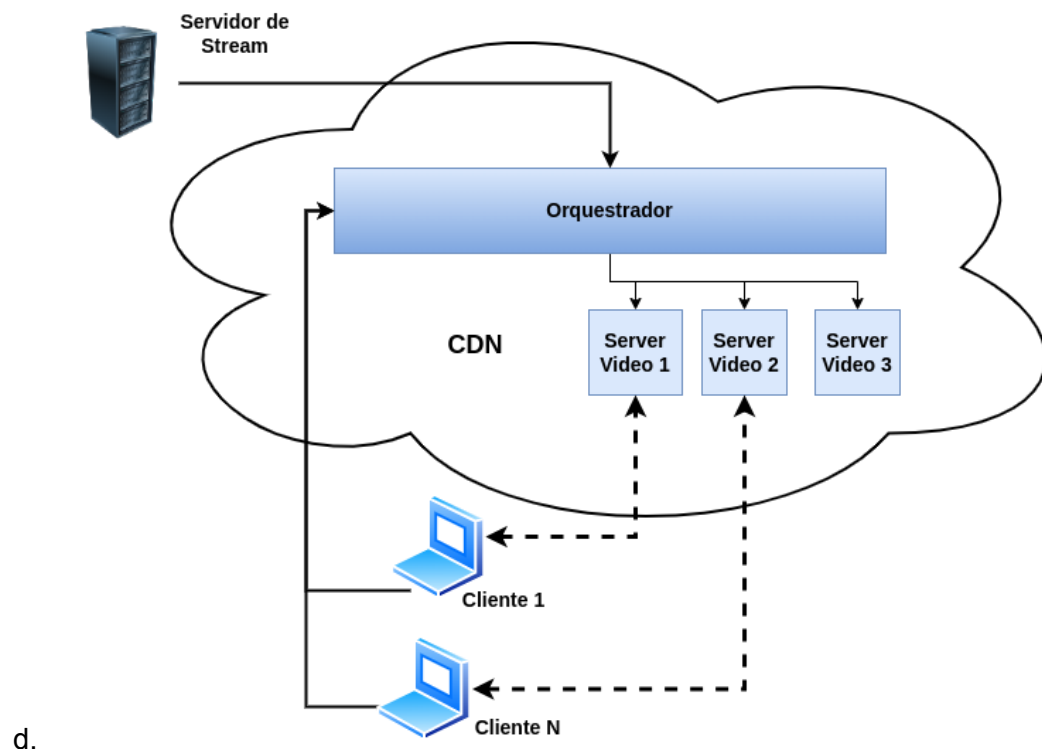


Figura 1: Arquitetura Geral

Resolução do Problema -

Segunda a nossa arquitetura, podemos ter vários servidores de *streaming* (**SS**) e clientes. Todos eles se conectam ao mesmo orquestrador. O orquestrador é responsável por certas áreas da rede.

Restrições

1. Um SS transmite somente para o orquestrador;
2. O Orquestrador faz o buffer de cada SS e retransmite para cada SV
3. O SV transmite para 0 ou mais Clientes. O SV possui uma quantidade máxima de clientes, essa informação é estabelecida no momento de sua criação.
4. Cada interação entre o Orquestrador e seus clientes (SS, SV, CV) é realizada em uma Thread separada.
5. O Orquestrador deve divulgar apenas seu endereço IP e sua porta de comunicação, o resto de toda comunicação e interação com os outros componentes, deve ser negociada entre eles (*outbound*). Esse ponto do projeto é importante, pois assim, a transparência de localização é implantada. Dessa forma, vemos o orquestrador como o cérebro do projeto, ele é quem sabe onde estão todos os elementos do sistema.
6. É responsabilidade do Orquestrador é direcionar os CV para os SV. Para isso o Orquestrador deverá respeitar as regras:
 - a. escolher o SV que tem a menor latência entre ele e o cliente;
 - b. respeitar a quantidade máxima de clientes atendida por cada SV.
 - c. caso as regras a. e b. não sejam atendidas, então o cliente receberá a notificação para:
 - i. desistir;
 - ii. esperar até que seja notificado pelo orquestrador de que um SV está livre, quando isso ocorrer os clientes automaticamente receberá o vídeo.

Requisitos

1. O sistema tem apenas 1 Orquestrador;

2. O sistema terá apenas 1 SS;
3. Inicialmente o Orquestrador criará 3 SV;
4. Total de CV 6;

Implementação

Os códigos devem ser implementados usando Python. Para a versão distribuída deve se usar Docker e a comunicação pode ser implementada por RPC, Socket ou MOM. Todos os processos serão abrigados na mesma máquina, mas cada em um *containers* diferente. A Tabela a seguir apresenta as especificações que devem ser seguidas.

Especificação	Opções
Sistema operacional	Windows ou Linux
Arquitetura de comunicação	TCP/IP
Linguagem de programação	Python, versão > 3.0
Comunicação entre processos	Sockets, RPC ou MOM
Interface de uso	Modo texto
Interface de teste	Modo texto
Dinâmica dos teste:	Crie 3 clientes; depois, gradativamente vai removendo um cliente e criando o outro.
Docker e Recursos	Cada Docker deve rodar em um Alpine/Outro. O Orquestrador ficará com 50% da CPU e Memória. Cada SV deve usar no máximo 10% da CPU. Por fim, cada CV deve usar no máximo 5% da CPU. Divide a memória para garantir que o Orquestrador tenha 50% e o resto seja dividido entre os outros.

Resultados

1. Gerencie o sistema de forma a mostrar o escalonamento de cada cliente;

Pontuação

Entrega contando que cada item contempla todos os pontos requisitados	Valor (pontos)
Relatório em PDF	2
Servidor de Stream	2
Orquestrador	2
Servidor de Vídeo	2
Clientes	2

Penalização

Ausência	Penalidade
Ausência do relatório	trabalho rejeitado
Ausência do código fonte	trabalho rejeitado
Não utilização do Python	trabalho rejeitado
Código fonte não roda conforme o relatório	trabalho rejeitado
Não apresentação do trabalho	trabalho rejeitado

Realização e Entrega

Observe os itens abaixo, eles são obrigatórios, o não atendimento implica em eliminação da 3ª avaliação, logo o aluno deve atender a todos os pontos para validar o trabalho. São eles:

1. Trabalho individual.
2. Envio somente pelo SIGAA.
3. Data de entrega até 02/08/2023 às 23:55.
4. Data de apresentação: 07/08 e 09/08.
5. O relatório e os códigos devem ser enviados em um único arquivo compactado (ZIP ou TAR.GZ) contendo:
 - a. Relatório
 - b. Arquivos fontes dos códigos

6. O código fonte deve conter comentários informativos. Para cada função ou método espera-se um texto explicativo sobre o bloco de código, uma descrição dos parâmetros e o que ele deve retornar.
7. Os trabalhos deverão ser apresentados, este requisito pode ser dispensado pelo professor em qualquer momento para todos os grupos de forma indiscriminada.
8. As apresentações serão em ordem alfabética nos dias programados.

Bom Trabalho!