

# Sistemas Distribuídos: Relatório Avaliação 2 *Middleware*

Alunos: Eduardo, Eric, Hugo, Jonas, Victor

Temas: Principal - Concorrência e Coordenação de Processos, secundário - Tolerância a falhas

Assunto do Livro: Capítulo Sistemas de Arquivos Distribuídos

---

Projeto de sistema de edição colaborativa de arquivos inspirado no *Google Docs*

Descrição do sistema e sua arquitetura

- O presente trabalho consiste em uma implementação com *Middleware* de um sistema distribuído que contempla a ideia de servidores distribuídos de arquivos. Para o mesmo, foram utilizados exclusivamente bibliotecas e frameworks do ecossistema *Python*: especialmente o *middleware Pyro5* e o *Flask*, *framework* para servidores *HTTP*. O código pode ser acesso clicando aqui neste link.

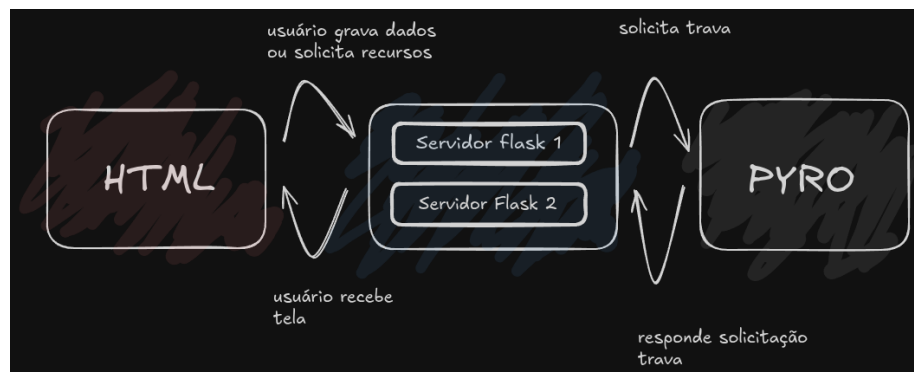


Figure 1: Diagrama Explicativo de arquitetura

- Como explicitado na imagem acima, o servidor *Pyro* gerencia o acesso aos recursos e garante o acesso seguro de usuários aos arquivos a que se deseja editar, e centraliza todo o controle de acesso. A escolha desse tipo de arquitetura foi deliberada, a fim de usar o *middleware* para as tarefas mais decisivas, a fim de demonstrar a importância desse tipo de módulo em sistemas distribuídos de uma forma geral. Os servidores *Flask* foram projetados para serem leves e desempenharem o mínimo possível, retornando páginas *HTML*, tecnologia utilizada para as interfaces de usuário, e fornecendo a interface de interação do mesmo com o servidor.

Os mesmos trabalham em conjunto para o mesmo propósito de forma complementar.

#### **Justificativa da característica escolhida e como foi implementada**

- O tema principal a ser abordado seria o acesso concorrente a arquivos com sucesso, garantindo funcionamento normal do sistema, assim como tolerância a possíveis falhas. Após a implementação de um sistema que corretamente suporta a coordenação de acesso de arquivos por partes de usuários, evitando condições de corrida devido ao uso de *threads*, foi implementada a funcionalidade de espelhamento de arquivos, funcionalidade garantida através do funcionamento do sistema através de dois servidores separados que devem ser executados ao mesmo tempo. O espelhamento de arquivos é bidirecional, permitindo que usuários em servidores separados possam trabalhar em conjunto colaborativamente em edição de arquivos.

#### **Estratégias adotadas, limitações e testes realizados**

- O maior desafio foi fornecer uma interface de usuário que pudesse interagir com um sistema complexo distribuído. Esse desafio foi vencido em grande parte através da escolha do uso de uma arquitetura *web* simples, o *MVC*, termo comumente utilizado para descrever aplicações *web* com renderização de HTML inteiramente no servidor, e com armazenamento de estado centralizado no mesmo. No entanto, a interface de usuário não permite edições concorrentes, simulando um *lock* compartilhado: apenas um usuário pode editar, porém muitos podem visualizar (não só o conteúdo do arquivo de forma estática, mas também visualizar as alterações sendo feitas pelo usuário editor, com intervalo de tempo de atualização de poucos segundos).
- O sistema não foi testado com arquivos de texto grandes o suficiente para observar problemas de desempenho. No entanto, é importante destacar a limitação de servidores HTTP para este tipo de aplicação (COULOURIS, 2013), possivelmente sendo um possível gargalo no projeto em uma situação de necessidade de escalabilidade.