

Sistemas Distribuídos

Trabalho Final

1 – Introdução.

O trabalho em si tratava-se da criação de um sistema, obviamente distribuído para uma livraria, onde os clientes poderiam acessar o mesmo e realizar operações como: cadastrar-se, comprar um livro, ver os livros disponíveis, sendo que algumas dessas operações só seriam realizadas por clientes cadastrados.

Porém a ideia inicial para o trabalho não era essa, nossa primeira ideia foi desenvolver um jogo distribuído usando a biblioteca PYRO do python, contudo essa possibilidade foi vetada, o que nos deu muito pouco tempo para pensar em um novo sistema, e devido a isso veio a ideia do sistema para a livraria.

O sistema encontra-se praticamente completo, faltando apenas alguns detalhes que por falta de tempo não foram terminados. O mesmo é dividido em duas partes um servidor e um cliente.

2 – Estrutura do sistema.

Como falado anteriormente o sistema é dividido em duas classes principais um cliente e um servidor.

2.1 – Servidor.

O servidor do sistema foi implementado na linguagem Java, sendo um servidor multithread, ou seja, capaz de receber várias requisições ao mesmo tempo.

O servidor é responsável por atender e tratar todas as requisições dos clientes, nele é feita a conexão com o banco de dados para o armazenamento de informações dos clientes e dos livros. Assim a cada requisição o servidor realiza a operação e retorna para o cliente que solicitou uma resposta de acordo com o resultado da operação.

Contudo, ele é responsável por gerenciar, interpretar e aplicar toda a regra de negócio do sistema.

2.2 – Cliente.

O cliente do sistema foi implementado em python, o mesmo foi criado com alguns cuidados para não deixar a interação do usuário com o sistema muito dependente do cliente. Sendo assim, o mesmo tem um interface simples aonde basicamente o usuário digita o que deseja e passa os dados referentes a operação desejada, e após confirmada a mesma será enviada como requisição para o servidor e aguardará uma resposta do mesmo para que possa continuar sua execução.

Todavia, o cliente não possui nenhuma regra de negócio, sendo que basicamente tudo é feito no servidor, tornando assim a interação do usuário com o sistema menos cansativa e menos propícia a erros ou ações indesejadas.

3 – Comunicação e Ferramentas utilizadas.

A comunicação entre o servidor e o cliente foi realizada utilizando a API Sockets, tanto a do Python como a do Java. A comunicação em si foi um dos maiores problemas para a criação do sistema. Pois ao enviar uma requisição o cliente deve aguardar uma resposta do servidor e somente após a mesma continuar a executar.

E obtivemos alguns erros por parte da própria API, erros estes que foram em grande parte consertados, mas alguns ainda existem no sistema.

Na parte de representação externas dos dados foi usado o Middleware JSON, sendo que por ser prático de usar e de modo legível, tornou-se opcional a forma de mostrar as respostas ao usuário, pois poderíamos tratar a resposta vinda do servidor e então depois de tratada repassar ao usuário ou deixar de acordo com o próprio JASON. A ultima opção foi a escolhida, visto que tínhamos pouco tempo e o nosso Middleware fazia isso de forma legível ao usuário.

Utilizamos também criptografia na hora de armazenar os dados, porém essa parte não foi integrada ao projeto na apresentação, pois encontrava-se em um projeto que estava sendo testado em outra máquina, e devido a falta de tempo não foi incrementada.

4 – Conclusão.

No momento o sistema encontra-se praticamente completo (salientamos novamente que o tempo foi nosso inimigo nessa parte), porém é um trabalho futuro terminá-lo. Assim como incrementar mais funcionalidades no mesmo, torna a comunicação mais simples e rápida, assim como adicionar a parte de criptografia que já foi feita, e basicamente reavaliar algumas possíveis melhorias no mesmo.

Contudo, é possível usar o sistema, por mais que o mesmo esteja um pouco incompleto algumas funcionalidades estão funcionando. As demais não estão pois como comentado anteriormente não foi possuí concluí-las.