Documentação — A3 SD

*Componentes: Grupo 01*

*Alex Ramos | RA 1272117271*

*Jônatas Gomes | RA 1272117581*

*Matheus Souza | RA 1272116583*

*Robson Santos | RA 12720110737*

*Iasmim Cristina | RA 1272117595*

*David Azevedo | RA 1272119055*

*Professor: Eduardo Sidney Xavier*

*UC: Sistemas Distribuídos e Mobile*

## Dependências

# Para a execução do sistema é necessário download de algumas bibliotecas para o python.

DateTime==5.1

db==0.1.1

db-sqlite3==0.0.1

requests==2.31.0

Subprocess

time

## Inicialização do projeto

O arquivo possui duas pastas de apoio onde temos o banco de dados e outra onde temos as funções, na página inicial temos três arquivo: o arquivo nomeado de “SocketsThread.py” que é o servidor principal da aplicação, o aquivo “Vendedor.py” que é um cliente e o arquivo “Gerente.py” outro cliente. Para facilitar o entendimento apelidamos os arquivos com os nomes: Servidor, Gerente e Vendedor, respectivamente.

A aplicação pode ser iniciada de três maneiras, são elas:

1. Através de modo manual iniciando o Servidor que por sua vez dá suporte às operações e iniciando o Gerente e do Vendedor.
2. Por meio de um dos clientes (Vendedor ou Gerente), que ao detectar a ausência do servidor aplicará o método de eleição.
3. Por meio do arquivo “Inicializar.py” que possuirá algumas perguntas para o preset da aplicação, logo após as perguntas ocorrerá o start do preset.

## Funcionamento

### Divisão:

O sistema em questão possui sua modularização dividida em cinco blocos:

* **Cliente:** Código carregado apenas em (Vendedor e Gerente) onde eles utilizam para a comunicação com o server,
* **Servidor:** Código carregado em todos os arquivos, porém usado inicialmente pelo Servidor.
* **Eleição:** Código carregado apenas em (Vendedor e Gerente) que por sua vez utilizarão esses algoritmos para eleger o próximo servidor.
* **Funções:** São todos os códigos necessários para a execução das tarefas por parte dos clientes e servidores.
* **Banco:** São os códigos responsáveis pela comunicação com a base de dados, gerenciamento e persistência das informações.

### Algorítimo:

Os algoritmos foram divididos em duas subcategorias:

* **Cliente-Servidor:** Onde temos a criação e execução dos sockets, passando pelo envio e recepção de mensagens, em paralelo com as threads direcionando os usuários para atendimentos unitários.
* **Eleição:** Aqui temos a aplicação de conceitos do Algoritmo de Anel em sua variação, aplicado ao problema atual.

### Lógica:

* **Algoritmo de Anel (Variação)**

Para a aplicação do algoritmo o adaptamos para utilizar-se da lógica do maior “voto”. Onde cada máquina participante da eleição avalia se o voto que está recebendo é maior, inferior ou igual a sua indicação de voto. Nesse caso, se o voto for maior que indicação da máquina, ela irá propagar o voto que ela recebeu, caso seja menor ela irá propagar a sua indicação de voto. Então, se o voto for igual à indicação de voto, entende-se que a máquina possui capacidade de assumir como host.

## Funcionamento da eleição

Durante o início do processo de eleição cada máquina que detectar a ausência do servidor, irá tentar se ancorar em alguma porta, tendo em vista a leitura do número máximo de máquinas participantes da eleição(disponível no arquivo “NumPCs.txt”). E a leitura do IP utilizado na eleição (disponível no arquivo “IpRede.txt”, recomenda-se o uso do 127.0.0.1, para não ocorrer conflitos). Após esses passos, cada máquina irá se ancorando em sentido anti-horário partindo do primeiro IP e porta, exemplo: pc1(ip 127.0.0.1:, port:9999),pc2(ip 127.0.0.7:, port:10006),pc1(ip 127.0.0.6:, port:10005).

Enquanto isso, a primeira máquina tentará comunicar com outro pc para enviar o seu voto, ao ser a primeira máquina na rede. A comunicação ocorrerá no sentido horário com a aplicação de três tentativas por máquina caso não haja resposta, ela seguirá para a máquina subsequente.

Na medida que as máquinas vão trocando seus votos (que para fins didáticos foram gerados de modo aleatório dando a oportunidade de qualquer máquina da eleição ser o próximo servidor) podem ocorrer três situações:

* **Está sozinho na eleição:** Caso a máquina tente comunicação com todas as máquinas e não haja retorno, ela verificará se o servidor principal está no ar, caso não esteja ela assumirá esse papel.
* **Alguma máquina cair na eleição:** Caso ocorra de alguma máquina cair ou houver demora excessiva na resposta, podem acontecer duas coisas: ou a máquina tentará enviar para outro PC (isso a partir da segunda rodada da eleição) ou irá bater o time out e ela tentará outra eleição.
* **Vencedor da eleição:** Se o PC for o campeão da eleição irá ocorrer dois fluxos a priori. A máquina irá testar a possibilidade de o servidor primário está no ar, caso ele não esteja, a máquina enviará o aviso para seu nó que por sua vez enviará o aviso para seu nó e assim até chegar na última máquina. Durante esse processo, a máquina campeã entrará em modo servidor possibilitando as outras máquinas o retorno das suas atividades normais até que o servidor principal retome as atividades.

Após esse processo a máquina campeã está no ar, porém, ela possui um comando de desligamento acionado assim que o servidor principal retorna às suas funções, fazendo com que essa máquina assuma novamente o papel de cliente.

## Informações Adicionais

Para a execução da eleição com mais de oito máquinas faz-se necessário a modificação do arquivo “NumPCs.txt” (disponível na mesma pasta dos clientes e servidores).

Para a execução da eleição em um ip diferente faz-se necessário a modificação do arquivo “IpRede.txt”(disponível na mesma pasta dos clientes e servidores).

O sistema possui validação de CPF, caso deseje realizar algum cadastro faz-se necessário a utilização de um CPF válido (Recomenda-se o uso desse site <https://www.4devs.com.br/gerador_de_cpf>).

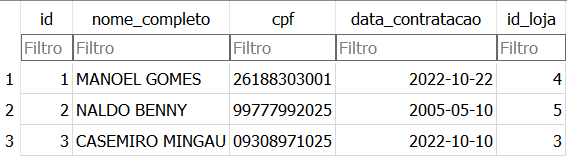
O sistema possui um algoritmo de busca por CEP. Caso deseje registrar alguma loja faz-se necessário a utilização de um CEP válido (Recomenda-se o uso desse site <https://www.geradordecep.com.br>).

Para a inicialização da eleição com máquinas já logadas faz-se necessário o input do usuário para o start do processo.

A aplicação possui um arquivo de nome “Inicializar.py” onde é possível realizar os preset’s para a inicialização da aplicação ou para a realização do teste de eleição.

A aplicação possui uma base de dados com alguns valores já preenchidos para teste, são eles:

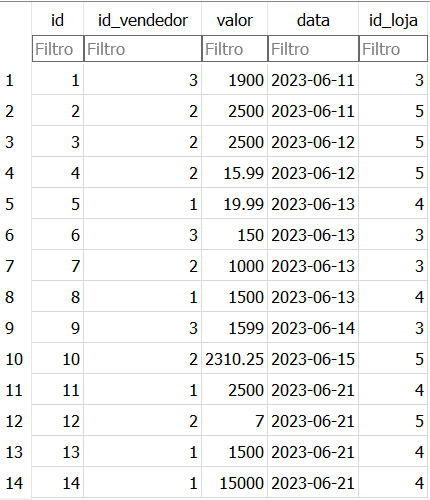
Vendedores:



Lojas:



Vendas:



Caso opte pela não utilização da base de dados, faz-se necessário a exclusão do arquivo “referencias.db” no caminho “\Banco\”.