**UNIFACS**

**Relatório**

**A3**

**Livia Ribeiro Jales** (1272117345)

**Gabriel Matos da Cunha** (1272122060)

**Lucas Luedy Rego** (1272116475)

**Bruno Pinheiro Oliveira** (1272117296)

**Guilherme Brito Henriques da Silva** (1272121797)

***Sistemas Distribuídos***

**Curso:** *Análise e desenvolvimento de sistemas*

**SALVADOR**

**2023**

1. **Introdução**

A intenção desse trabalho é criar uma aplicação utilizando sockets. A aplicação fará a captação de dados referente às vendas em uma rede de lojas. O objetivo principal deste trabalho é criar uma solução eficiente que permita aos vendedores registrarem suas vendas e aos gerentes consultarem as informações registradas.

A aplicação foi projetada da seguinte maneira, são dois tipos de clientes, os gerentes e os vendedores. Os vendedores têm a responsabilidade de informar ao sistema o valor de cada venda realizada, enquanto os gerentes podem consultar as informações de vendas registradas.O servidor atua como hospedeiro das informações fornecidas pelos vendedores e respondendo às consultas dos gerentes.

Uma funcionalidade importante da aplicação é a capacidade de eleger um coordenador/servidor temporário na ausência do servidor principal. Esse coordenador temporário irá assumir a responsabilidade do coordenador principal temporariamente, até que ele volte.

Não foi utilizado uma interface gráfica na aplicação, apenas textual, com o intuito de demonstrar os processos que estão ocorrendo.

Toda a aplicação foi construída em python, e construída com a arquitetura MVC, um modelo que divide aplicação em tres camadas, Model, Controller, View.

1. **Tecnologias e Bibliotecas utilizadas.**

**SQlite3 -** Uma biblioteca em linguagem C que implementa uma base de dados SQL embutida, com todas as funcionalidades, incorporado diretamente ao programa , o que permite acesso ao banco de dados sem a necessidade de um SGBD. Não requer uma instalação separada, nem precisa de um servidor dedicado.

**Biblioteca OS Python -** Biblioteca padrão para interagir com o sistema operacional em que o python está sendo executado.

**Biblioteca ‘datetime’ -** Biblioteca utilizada para formatação de data e hora.

**Biblioteca ‘sockets’ -** Biblioteca que permite que ocorra a comunicação entre os processos.

**Biblioteca ‘load\_dotenv’ -** Biblioteca utilizada para acessar as variáveis do arquivo .env. Ela busca automaticamente as variáveis definidas no arquivo .env e as adiciona ao ambiente de execução do programa.

**Biblioteca ‘re’ -** Biblioteca que verifica se a condição foi cumprida, o regex.

**Biblioteca ‘threading’ -** Biblioteca que permite a utilização de múltiplos processos no servidor.

**Biblioteca ‘uuid’ -** Biblioteca utilizada para gerar id’s únicas e universais com valor de 128 bits.

1. **Comandos de importação das bibliotecas utilizadas**

* *import sqlite3*
* *import os*
* *import datetime*
* *import socket*
* *import load\_dotenv*
* *import re*
* *import threading*
* *import uuid*

1. **Arquitetura da Aplicação**

* **root :** Se encontra o código de eleição de coordenador, o Readme e o socket do servidor.
* **Controller :** Possui o formulário do gerente, do vendedor, e do login.
* **Inicial :** Possui as configurações iniciais do banco de dados.
* **Model :** Onde fica o Banco de Dados, as consultas, informações relacionadas ao banco de dados, a lógica que permite capturar

os dados.

* **View :** É onde está a interação com o usuário.

1. **Detalhamento sobre Arquitetura e algoritmo**

A arquitetura utilizada no projeto é a MVC, que é um modelo dividido em 3 camadas. A 1 camada, a model, é referente a manipulação, validação e persistência dos dados. A camada número 2, a View, é referente a apresentação dos dados ao usuário, exibindo as informações que são dadas pelo modelo. A última camada, Controller, faz o intermédio entre o modelo e a view, recebendo e processando as ações do usuário, e a partir daí atualiza o modelo, ou a própria view caso seja necessário. O controller também é utilizado para controlar a navegação dentro da aplicação.

O algoritmo de eleição escolhido pela equipe foi o algoritmo de Bully. Ele serve para eleger um líder entre os processos, através de um id, caso um dos processos identifique a perda de contato com o líder, ele inicia uma eleição, mandando mensagem a todos os outros processos. Todos os nós respondem com seu identificador, se o processo que iniciou a eleição tiver o maior identificador entre os que se candidataram, ele se autoproclama líder e avisa aos outros, se não, ele aguarda que o processo com maior identificador inicie outra eleição. Foi necessário a colocação de uma threading nesse algoritmo para fazer com que o cliente possa atuar tanto como servidor, como cliente. Para gerar o id dos processos utilizamos o autoincremento, onde cada processo saíria com seu id único já especificado. Cada id desses, é um paralelo com a ‘força’ do processo, quanto maior, mais forte. Em caso de queda do processo coordenador, se iniciará uma eleição, A partir desse ponto, foi incluído várias condicionais no algoritmo, a fim de comparar se o algoritmo que levará a eleição, é de fato o algoritmo com maior força.

1. **Conclusão**

O trabalho consiste no desenvolvimento de uma aplicação de captação de dados de vendas em uma rede de lojas, utilizando sockets. A aplicação segue a arquitetura MVC e foi construída em Python, com o objetivo de permitir que vendedores registrem vendas e gerentes consultem as informações. A aplicação possui a capacidade de eleger um coordenador temporário na ausência do servidor principal. Foram utilizadas diversas bibliotecas, como SQLite3, sockets e threading. O algoritmo de eleição escolhido foi o algoritmo de Bully.

Além disso, destacamos o fato da aplicação ter sido desenvolvida em Python, aproveitando os benefícios da linguagem para oferecer uma solução robusta e escalável. A utilização da arquitetura MVC permitiu a modularização da aplicação, essa divisão de responsabilidade promove uma manutenção mais facilitada.

No geral, a aplicação ofereceu uma solução eficiente para o registro e consulta de vendas, demonstrando a aplicação prática dos conceitos de arquitetura e algoritmo implementados.

Através desse trabalho, foi possível compreender e aplicar conceitos fundamentais de desenvolvimento de software, como a divisão em camadas, o uso de bibliotecas e a implementação de algoritmos de eleição. Essas habilidades são valiosas para o desenvolvimento de aplicações robustas e escaláveis, garantindo a eficiência e confiabilidade dos sistemas desenvolvidos.