



PEDRO BISPO DE JESUS ALMEIDA - 12724217720
CAUE DE SOUZA LUZ - 12724119349
DIOGO DA SILVA SOUZA - 12724143402
GUSTAVO LIMA DA SILVA - 12724138881
DANILO QUEIROZ NOGUEIRA - 1272419424

RELATÓRIO SISTEMAS DISTRIBUÍDOS E MOBILE

SALVADOR - BA
2025

PEDRO BISPO DE JESUS ALMEIDA - 12724217720
CAUE DE SOUZA LUZ - 12724119349
DIOGO DA SILVA SOUZA - 12724143402
GUSTAVO LIMA DA SILVA - 12724138881
DANILO QUEIROZ NOGUEIRA - 1272419424

RELATÓRIO SISTEMAS DISTRIBUÍDOS E MOBILE

RELATÓRIO ESCRITO

EDUARDO SIDNEY DA SILVA XAVIER

SALVADOR - BA

2025

RESUMO

O relatório tem como objetivo descrever as ações realizadas para a atividade A3 da matéria "Sistemas Distribuídos e Mobile".

A atividade requisita o desenvolvimento, em equipe, de um aplicativo cliente-servidor voltado para o acompanhamento de tarefas de funcionários em uma empresa. A aplicação deve permitir a interação de três perfis distintos de usuários: funcionário, supervisor e gerente, cada um com funções específicas.

O funcionário visualiza suas tarefas e informa quando uma delas é concluída. O supervisor é responsável por cadastrar tarefas e atribuí-las aos funcionários, além de acompanhar sua execução. Já o gerente pode solicitar relatórios gerenciais com dados como tarefas pendentes, tarefas cadastradas e funcionários sem pendências.

SUMÁRIO

1 INTRODUÇÃO	5
2 DESENVOLVIMENTO	6
3 PRINCIPAIS DESAFIOS	16
4 CONCLUSÃO	17

1 INTRODUÇÃO

O seguinte relatório da matéria de Sistemas Distribuídos e Mobile tem como objetivo representar a interação entre funcionários de uma mesma empresa com funções distintas. A aplicação deve permitir a interação de três perfis distintos de usuários: funcionário, supervisor e gerente, cada um com funções específicas.

O funcionário visualiza suas tarefas e informa quando uma delas é concluída. O supervisor é responsável por cadastrar tarefas e atribuí-las aos funcionários, além de acompanhar sua execução. Já o gerente pode solicitar relatórios gerenciais com dados como tarefas pendentes, tarefas cadastradas e funcionários sem pendências.

2 DESENVOLVIMENTO

Fomos desafiados a criar um software utilizando uma das tecnologias para integração e comunicação interna do programa, sendo elas: threads, sockets, APIs, RPC, RMI ou Microserviços; dissertando sobre nossa escolha. Cada uma dessas tecnologias tem um papel crucial no desenvolvimento de sistemas distribuídos e comunicação entre componentes. Aqui está um resumo das diferenças:

- 1. **Threads:** Executam múltiplas tarefas simultaneamente dentro de um único processo. Útil para melhorar a eficiência, mas não ajuda na comunicação entre sistemas distintos.
- 2. **Sockets:** Permitem comunicação direta entre dispositivos na rede. São eficientes, mas exigem bastante configuração e gerenciamento de conexões.
- 3. **APIs:** São interfaces que permitem que sistemas troquem dados e funcionalidades de forma controlada. Fáceis de integrar e amplamente utilizadas na web.
- 4. **RPC (Remote Procedure Call):** Permite que um programa execute funções de outro sistema remotamente, mas pode ser complexo de gerenciar.
- 5. **RMI (Remote Method Invocation):** Similar ao RPC, porém focado em Java. Possui forte integração com objetos remotos, mas depende da plataforma.
- 6. **Microserviços:** Arquitetura que divide um sistema em pequenos serviços independentes. Oferece escalabilidade, mas pode ser excessivamente complexa para projetos pequenos.

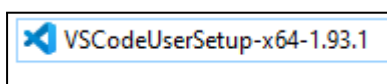
- **Por que threads e sockets são a melhor escolha para um pequeno projeto empresarial?**
- Threads e sockets representam uma abordagem poderosa e flexível para a comunicação entre processos, especialmente adequada para pequenos projetos empresariais que exigem controle direto sobre a troca de dados e a execução simultânea de tarefas. Veja por que essa escolha é vantajosa:
- **Facilidade de controle:** O uso direto de sockets e threads permite que os desenvolvedores tenham domínio completo sobre o fluxo de comunicação e o comportamento concorrente do sistema, tornando mais fácil adaptar soluções a necessidades específicas.
- **Desempenho e leveza:** Essa abordagem tende a consumir menos recursos, ideal para projetos com infraestrutura limitada, proporcionando uma comunicação rápida e eficiente.
- **Execução simultânea:** O uso de threads possibilita que múltiplas operações ocorram em paralelo, como o atendimento simultâneo de vários clientes, sem bloquear o servidor principal.
- **Compatibilidade com diversos ambientes:** Sockets são amplamente suportados em diferentes linguagens e plataformas, facilitando a implementação tanto em aplicações locais quanto distribuídas.
- **Baixa dependência de bibliotecas externas:** Diferente de frameworks mais complexos, o uso de threads e sockets geralmente depende apenas das bibliotecas nativas da linguagem, o que reduz a complexidade da manutenção e da instalação.
- Em um pequeno projeto empresarial, onde os recursos são limitados e há a necessidade de soluções diretas e personalizáveis, threads e sockets se destacam por sua simplicidade, eficiência e controle granular. Eles permitem construir sistemas robustos com comunicação eficiente, mantendo a flexibilidade para futuras adaptações.

Para fazer a instalação e execução do programa, elaboramos o seguinte passo-a-passo:

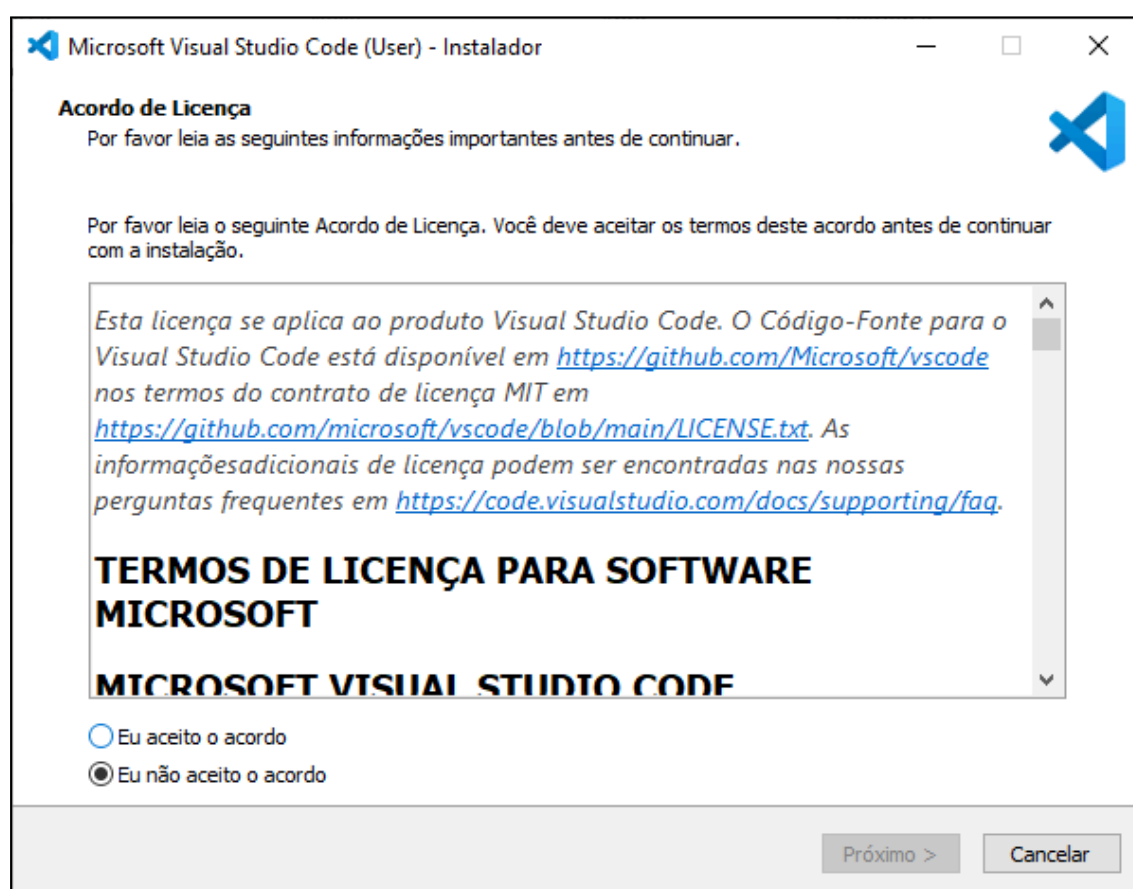
1 – INSTALAÇÃO DO VSCODE:

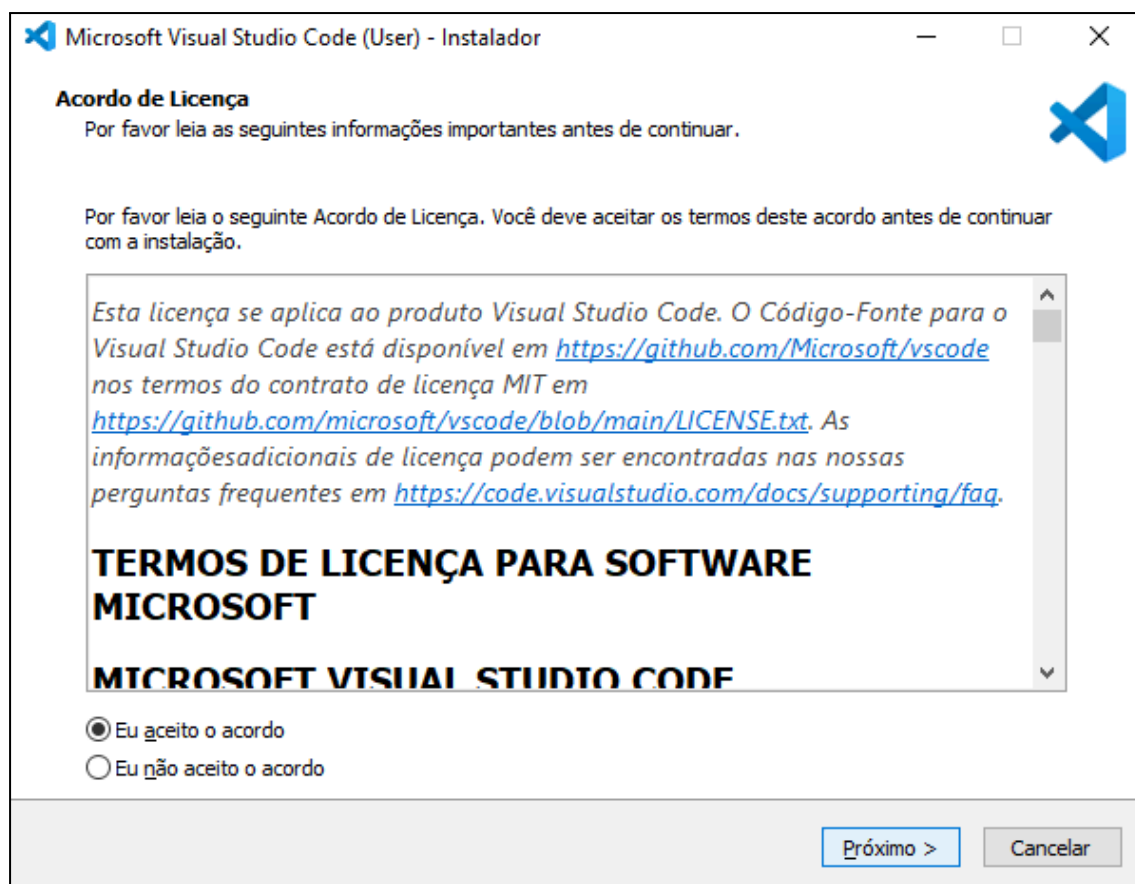
Acesse o link: <https://code.visualstudio.com>

2 – EXECUTE O EXE:

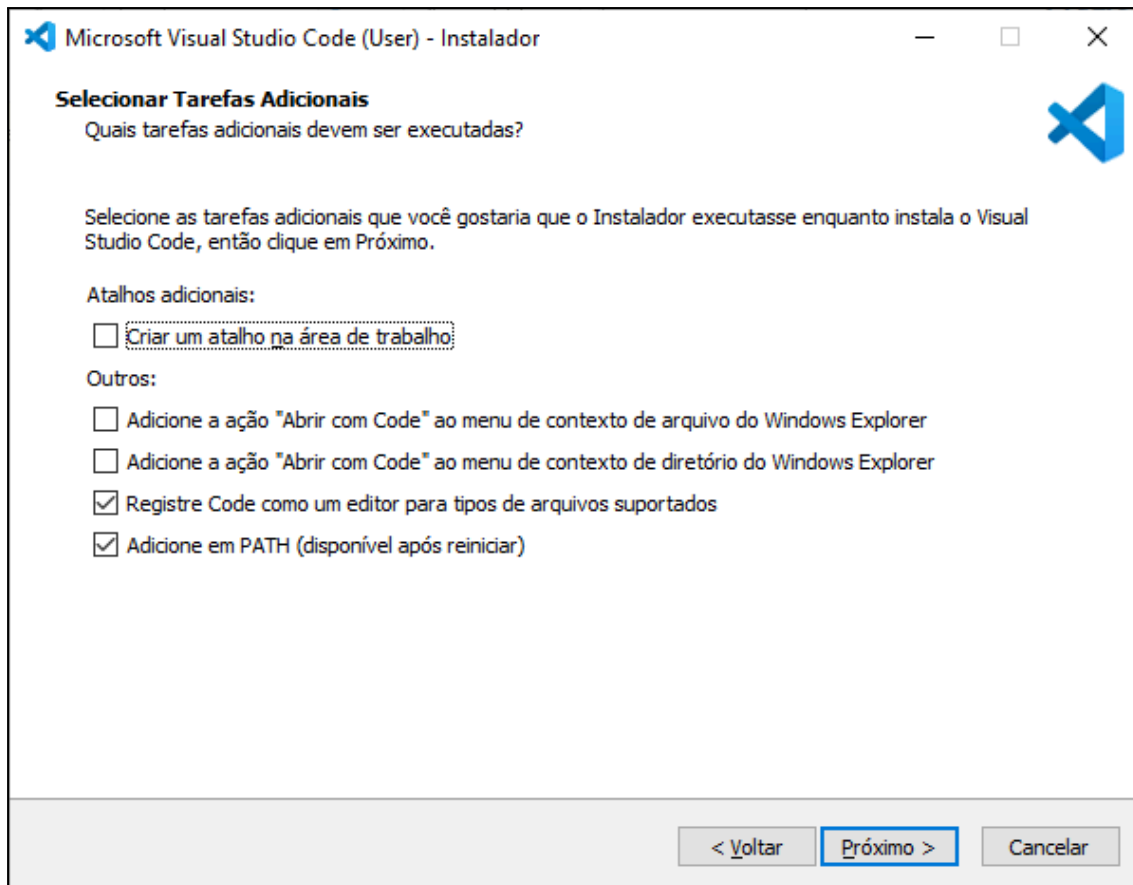


3 – CLIQUE EM **Eu aceito o acordo** e depois em Próximo:

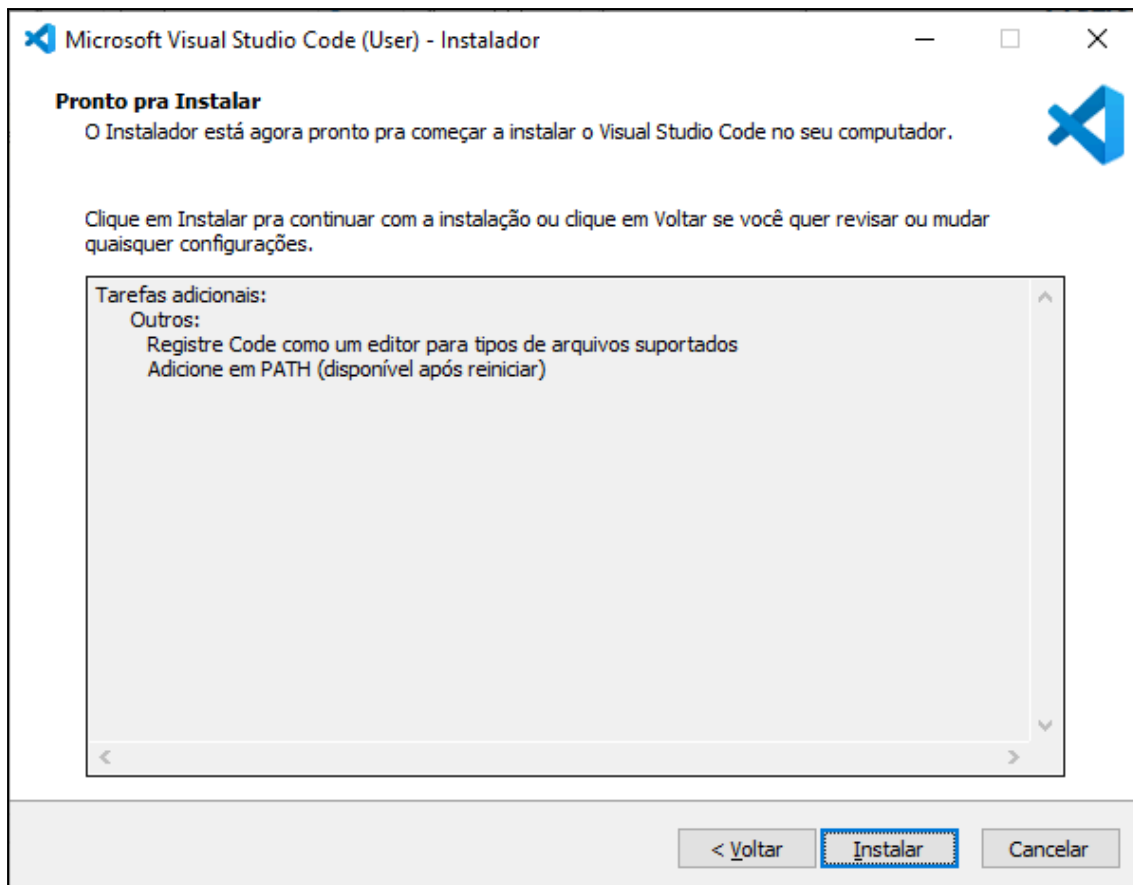




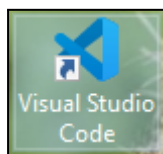
4 – CLICAR EM PROXIMO NOVAMENTE



5 – CLICAR EM INSTALAR



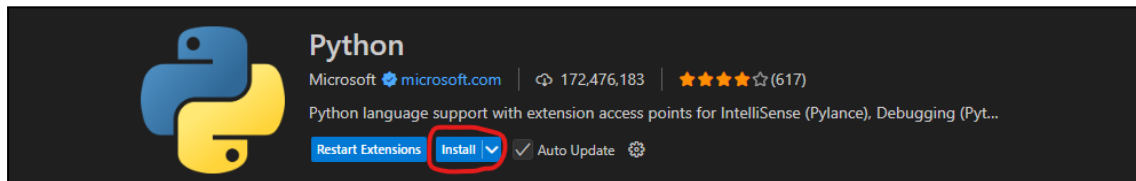
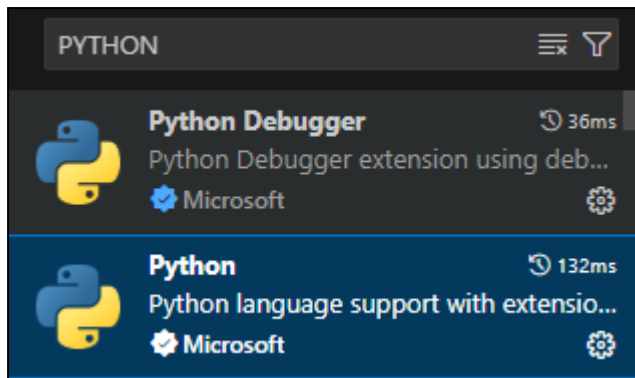
6 – ABRIR O VSCODE



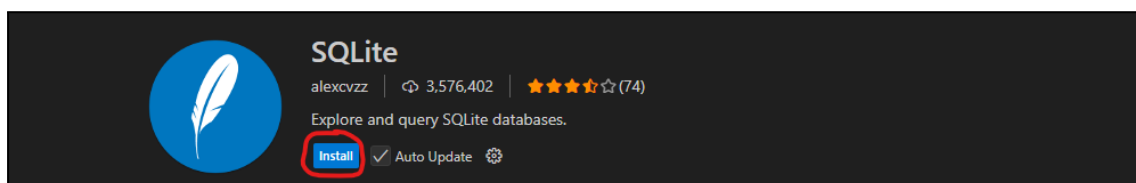
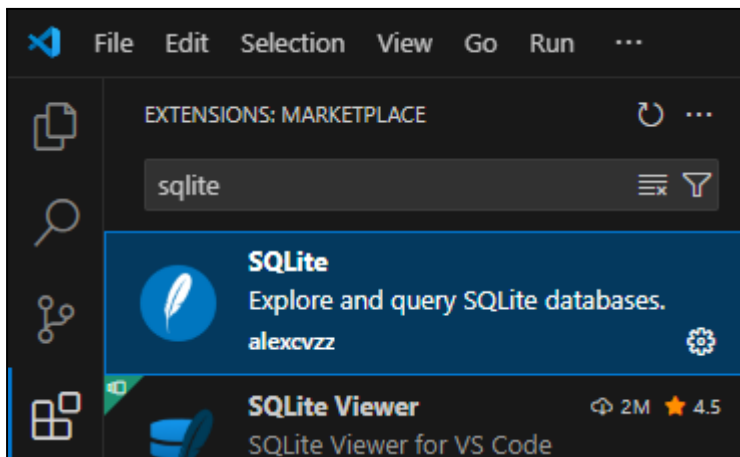
7 – DENTRO DO VSCODE, CLICAR NO ÍCONE DE EXTENSÕES



8 – PESQUISAR POR **PYTHON** e clicar em **INSTALAR**



9 – Instalar a extensão **SQLITE**



10 – Baixar o **PYTHON** no navegador e executar o **EXE**

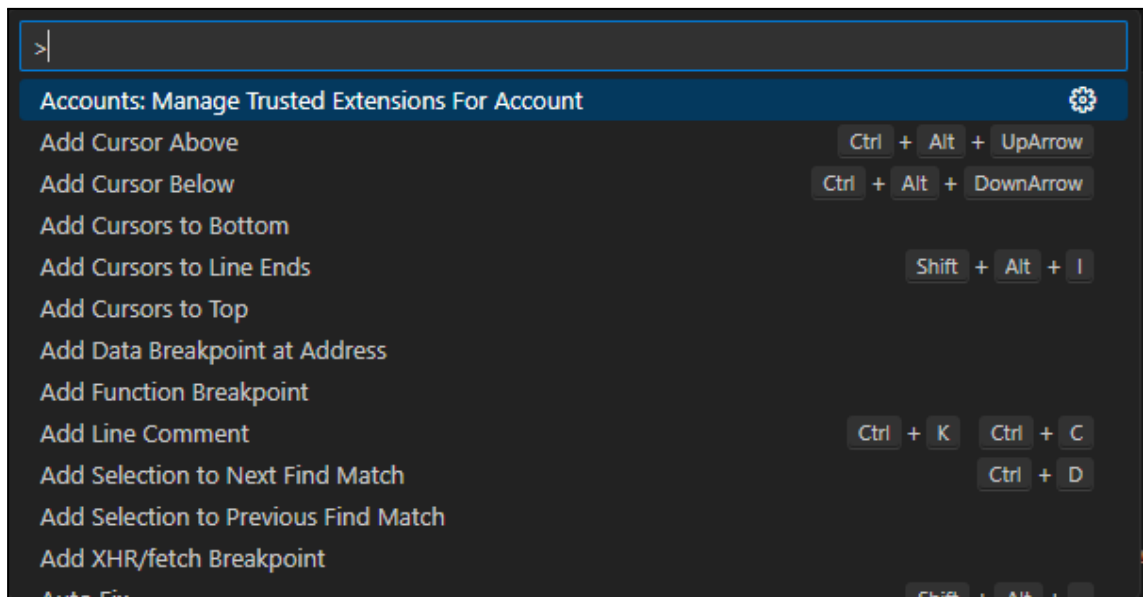
<https://www.python.org/downloads/>



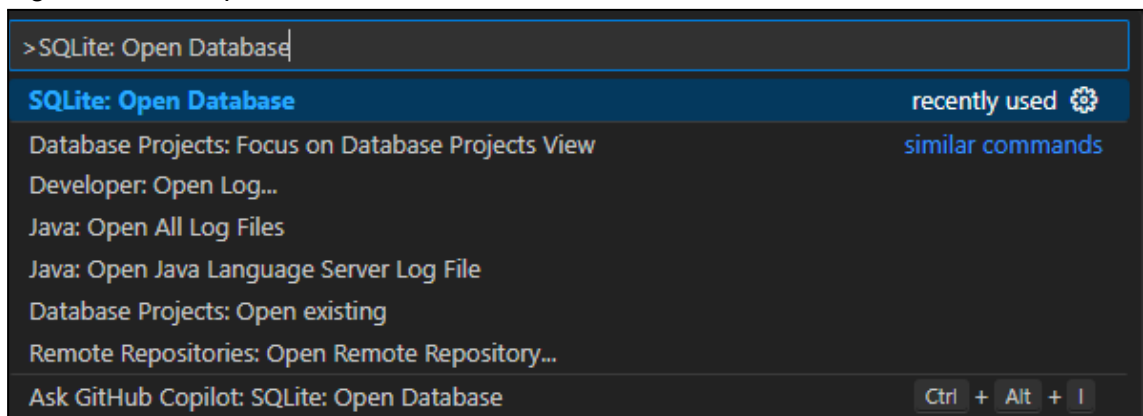
python-3.13.0-amd64

11 – Logo após a ativação do servidor, para demonstração da atualização do banco de dados, seguir os seguintes passos:

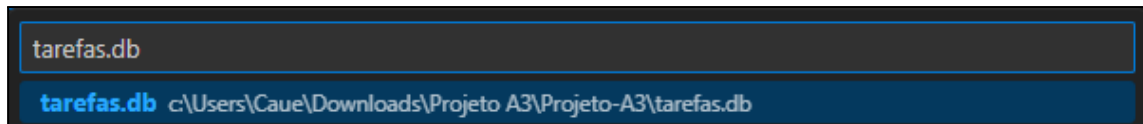
- Apertar no teclado CTRL + SHIFT+ P



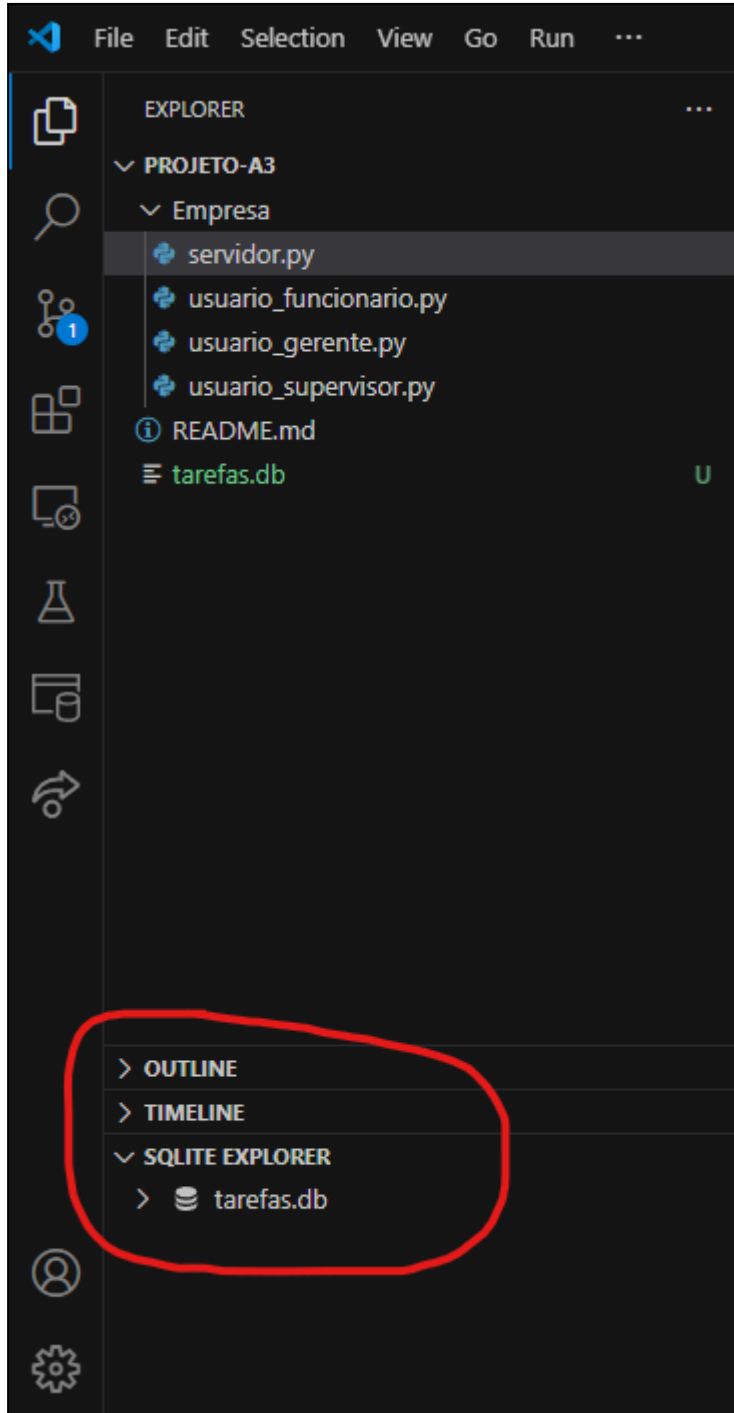
- Digitar SQLite: Open Database



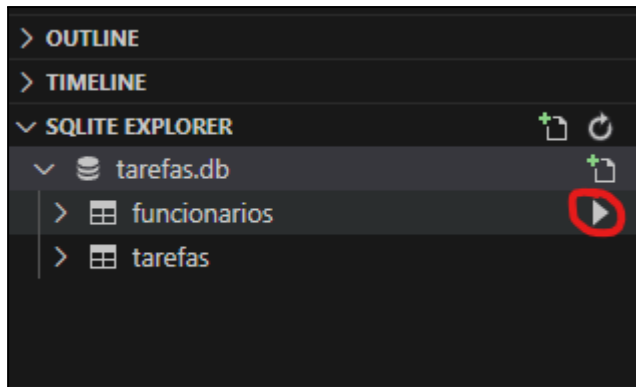
- Selecionar o arquivo tarefas.db



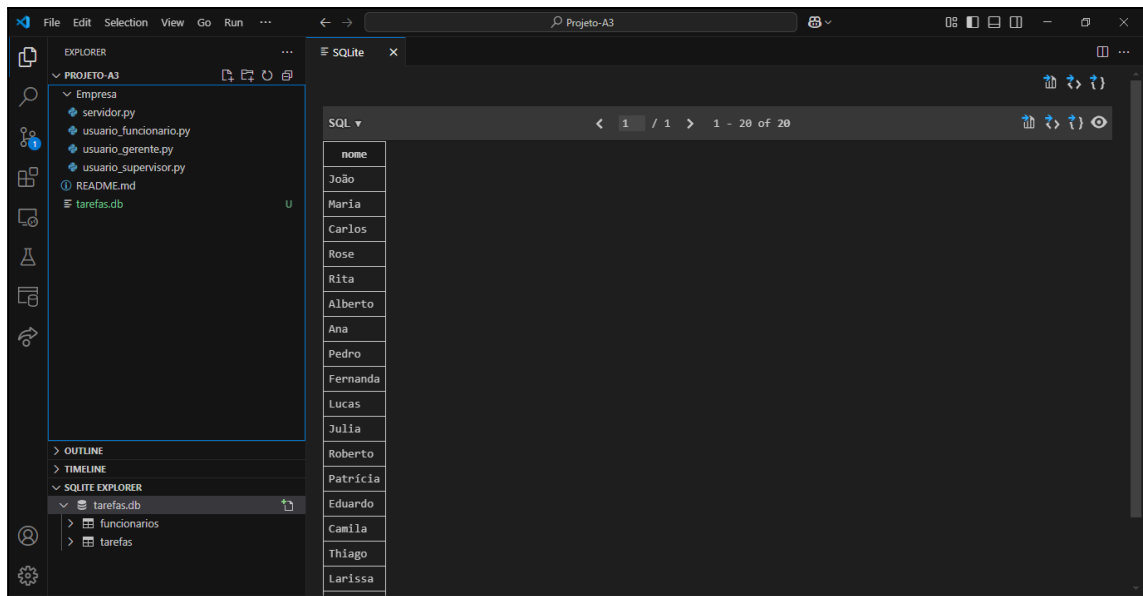
- Irá abrir no canto inferior esquerdo um novo explorer



- Clicar no **tarefas.db** e visualizar as tabelas no ícone de **play**



- Será aberta uma nova janela com os dados contidos na tabela



2.1 PRINCIPAIS DIFICULDADES

Durante a etapa de configuração do servidor, foi proposta a implementação de uma função específica para processar o recebimento dos dados. No entanto, surgiram dificuldades significativas na recuperação das informações armazenadas no banco de dados, especialmente ao tentar listar as tarefas pendentes, sem obter os resultados esperados.

Para diagnosticar o problema, foram inseridos pontos de depuração (debugs) que permitiram verificar os dados e os tipos de informações que estavam sendo enviados e recebidos pelo servidor. Essa estratégia foi fundamental para identificar inconsistências e, com isso, realizar os ajustes necessários tanto no código quanto na estrutura do banco de dados. Após essas correções, o funcionamento do sistema foi normalizado, possibilitando o correto retorno das tarefas pendentes.

3 CONCLUSÃO

O desenvolvimento deste projeto permitiu aplicar na prática os conhecimentos adquiridos na disciplina de Sistemas Distribuídos e Mobile, promovendo a compreensão dos principais conceitos relacionados à arquitetura cliente-servidor, comunicação entre processos e persistência de dados em banco relacional. A criação da aplicação para acompanhamento de tarefas corporativas exigiu o planejamento e implementação de funcionalidades específicas para três perfis distintos de usuários: funcionário, supervisor e gerente, cada um com suas responsabilidades e formas de interação com o sistema.

Durante o processo de configuração do servidor, foi proposta a implementação de uma função dedicada ao processamento dos dados recebidos. Entretanto, dificuldades surgiram ao tentar recuperar as tarefas pendentes no banco de dados, uma vez que as consultas não retornavam os resultados esperados. Através da utilização de mecanismos de depuração, foi possível inspecionar os dados trafegados entre cliente e servidor e identificar incompatibilidades de tipos. Com base nessas informações, ajustes precisos foram realizados tanto no código quanto na estrutura do banco de dados, resultando na resolução do problema e no funcionamento adequado da aplicação.

O projeto, portanto, não apenas consolidou os conteúdos teóricos da disciplina, mas também proporcionou vivências práticas de resolução de problemas, trabalho em equipe e uso de ferramentas essenciais para o desenvolvimento de sistemas distribuídos. A experiência contribuiu significativamente para o amadurecimento técnico dos integrantes e para a preparação profissional em contextos reais de desenvolvimento de software.