



**UNIFACS**  
LAUREATE INTERNATIONAL UNIVERSITIES®

## A3 de Sistemas Distribuídos

### Relatório da equipe

#### Integrantes:

NOME	RA
Pedro Henrique de Oliveira Carvalho	12724111265
Victor Eduardo Teles Chaves	1272419530
Kleber Araujo Rodrigues	12724135299
Ana Priscila Silva Oliveira	1272411739

#### Contribuições:

PAPÉIS	RESPONSÁVEL
Servidor	Pedro Henrique de Oliveira Carvalho
Banco	Pedro Henrique de Oliveira Carvalho
Atendente	Kleber Araujo Rodrigues
Garçom	Ana Priscila Silva Oliveira
Gerente	Victor Eduardo Teles Chaves
Documentação	Victor Eduardo Teles Chaves Kleber Araujo Rodrigues
Video	Ana Priscila Silva Oliveira Pedro Henrique de Oliveira Carvalho

## Requerimento de Softwares:

COMPONENTE	VERSÃO MINIMA RECOMENDADA	DESCRIÇÃO
Python	3.8 ou superior	Linguagem de programação usada no servidor e clientes
Biblioteca <b>socket</b>	Padrão no Python	Usada para comunicação entre clientes e servidor
SQLite3	Padrão no Python	Banco de dados local e leve, usado para registrar reservas
VS Code	Qualquer versão	Ambiente de desenvolvimento/testes

## Instruções de Instalação e Execução

### Pré-requisitos:

- Ter o Python 3 instalado
- Estar em um sistema com terminal (Linux, Windows, WLS ou MAC)

### Passo a passo:

#### 1. Clone do repositório:

Comando:

```
git clone https://github.com/Pedrohxxz/A3-de-Sistemas-Distribuidos.git
```

#### 2. Inicie o servidor:

Comando:

```
python3 server/servidor.py
```

### 3. Em outros terminais, execute os clientes:

Comando:

```
python3 cliente_atendente/cliente_atendente.py python3  
cliente_garcom/cliente_garcom.py python3  
cliente_gerente/cliente_gerente.py
```

Os três clientes se conectam ao mesmo servidor via socket TCP e se comunicam em tempo real.

## Justificativa da Abordagem de Comunicação

Optamos por uma **arquitetura cliente-servidor com sockets TCP** por ser uma abordagem eficiente, leve e direta para aplicações em redes locais. A comunicação TCP garante confiabilidade na entrega das mensagens, o que é essencial em um sistema de reservas — onde cada confirmação ou cancelamento precisa ser preciso e consistente.

A escolha de **sockets nativos do Python** foi feita para evitar dependências externas e facilitar o entendimento do funcionamento interno da comunicação entre processos. Além disso, essa abordagem permite escalar facilmente o projeto, adicionando mais clientes ou funções ao servidor sem grandes mudanças estruturais.

O uso de **SQLite como banco de dados local** reforça a simplicidade e autonomia da aplicação, mantendo os dados organizados e persistentes sem necessidade de servidores de banco externos.

Responsável: Eduardo Sidney da Silva Xavier

