



UNIVERSIDADE SALVADOR – UNIFACS

SISTEMAS DE INFORMAÇÃO – ADS (ANÁLISE E DESENVOLVIMENTO DE SISTEMAS)

ARTHUR DE SOUZA CARVALHO - RA: 12723120130

BEATRIZ OLIVEIRA AGUIAR SIMÕES - RA: 12725161536

DANIEL SANTOS LOPES - RA: 12723130024

ENRIQUE SILVA DOS REIS - RA: 12724149018

IGOR CÉSAR QUEIROZ COELHO - RA: 12724138882

NATHAN MOREIRA DE ALMEIDA FRANÇA - RA: 12724130864

Orientador: **Eduardo Sidney da Silva Xavier**

**ATIVIDADE AVALIATIVA – SISTEMAS DISTRIBUÍDOS E MOBILE**

**PROJETO A3**

Salvador, BA

2025

ARTHUR DE SOUZA CARVALHO - RA: 12723120130  
BEATRIZ OLIVEIRA AGUIAR SIMÕES - RA: 12725161536  
DANIEL SANTOS LOPES - RA: 12723130024  
ENRIQUE SILVA DOS REIS - RA: 12724149018  
IGOR CÉSAR QUEIROZ COELHO - RA: 12724138882  
NATHAN MOREIRA DE ALMEIDA FRANÇA - RA: 12724130864  
Orientador: **Eduardo Sidney da Silva Xavier**

## **ATIVIDADE AVALIATIVA – SISTEMAS DISTRIBUÍDOS E MOBILE**

### **PROJETO A3**

Relatório documentado sobre o progresso da  
atividade avaliativa A3 da Unidade Curricular  
**Sistemas Distribuídos e Mobile**, na qual está sendo  
realizada no semestre de 2025.1.

Salvador, BA

2025

# Sumário

<b>1 - Introdução e objetivo.....</b>	<b>4</b>
<b>2 - Descrição dos requerimentos de software necessários para execução da aplicação.....</b>	<b>4</b>
<b>3 - Instruções para instalação e execução da aplicação.....</b>	<b>4</b>
3.1 - Instalação do Python.....	4
3.2 - Instalação das Dependências.....	6
3.3 - Estrutura de Pastas.....	7
3.4 - Execução do Servidor.....	7
3.5 - Acesso via Navegador.....	8
 <b>4 - Abordagem de comunicação escolhida e justificativa.....</b>	 <b>9</b>
4.1 Abordagem .....	9
4.2 Justificativa.....	10
 <b>5 - Apresentação e demonstração do uso da aplicação.....</b>	 <b>10</b>
5.1 Explicação.....	10
5.2 URL do vídeo.....	10
 <b>6 - Conclusão.....</b>	 <b>10</b>

# 1 - Introdução e objetivo

O relatório a ser lido possui o intuito de explicar a aplicação solicitada e o funcionamento, requerimentos, e instruções da mesma.

A aplicação tem como objetivo gerenciar reservas de mesas em um restaurante, permitindo o controle centralizado e em tempo real das reservas. O sistema é dividido em três perfis de usuário: **Atendente** (cria/cancela reservas), **Garçom** (confirma reservas) e **Gerente** (emite relatórios). A solução utiliza uma arquitetura cliente-servidor, com interface web e scripts de automação, garantindo praticidade, segurança e rastreabilidade das operações.

## 2 - Descrição dos requerimentos de software necessários para execução da aplicação

Para executar a aplicação, são necessários:

- **Python 3.7 ou superior**  
(<https://www.python.org/downloads/>)
- **Frameworks a ser instalados:**
  - Flask
  - Flask\_Cors (Biblioteca/Extensão do framework Flask)
- **SQLite** (já incluso no Python padrão)
- **Navegador Web** (Google Chrome, Firefox, Edge, etc.)
- **Sistema Operacional:** Windows, Linux ou MacOS

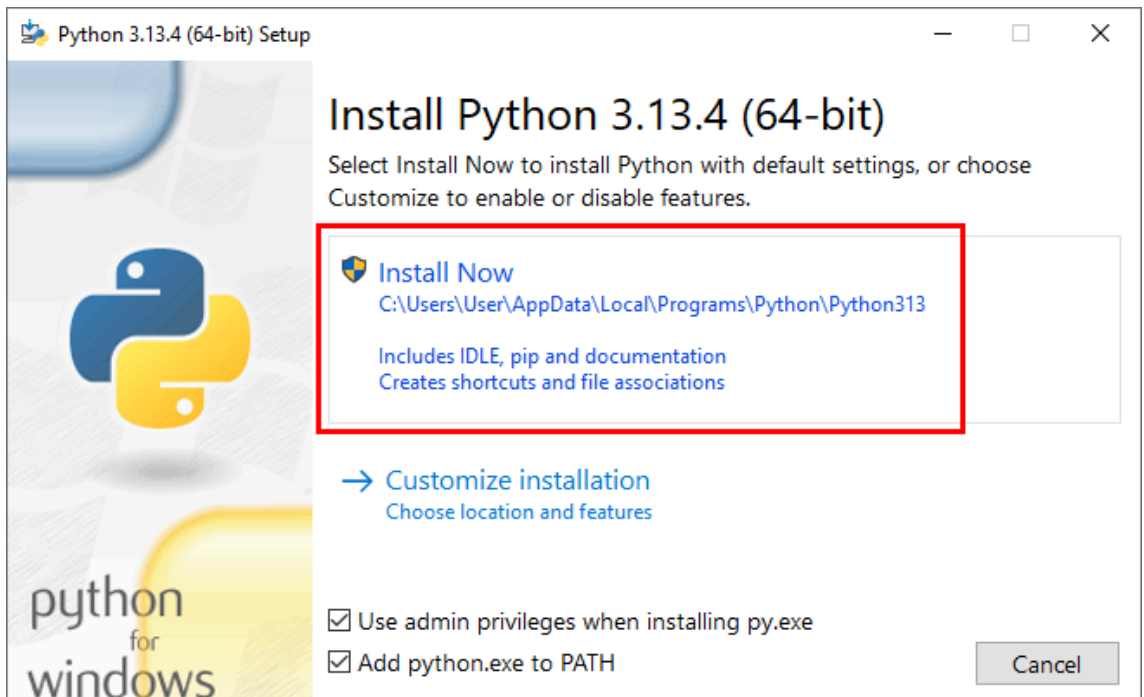
## 3 - Instruções para Instalação e Execução da aplicação

### 3.1 - Instalação do Python

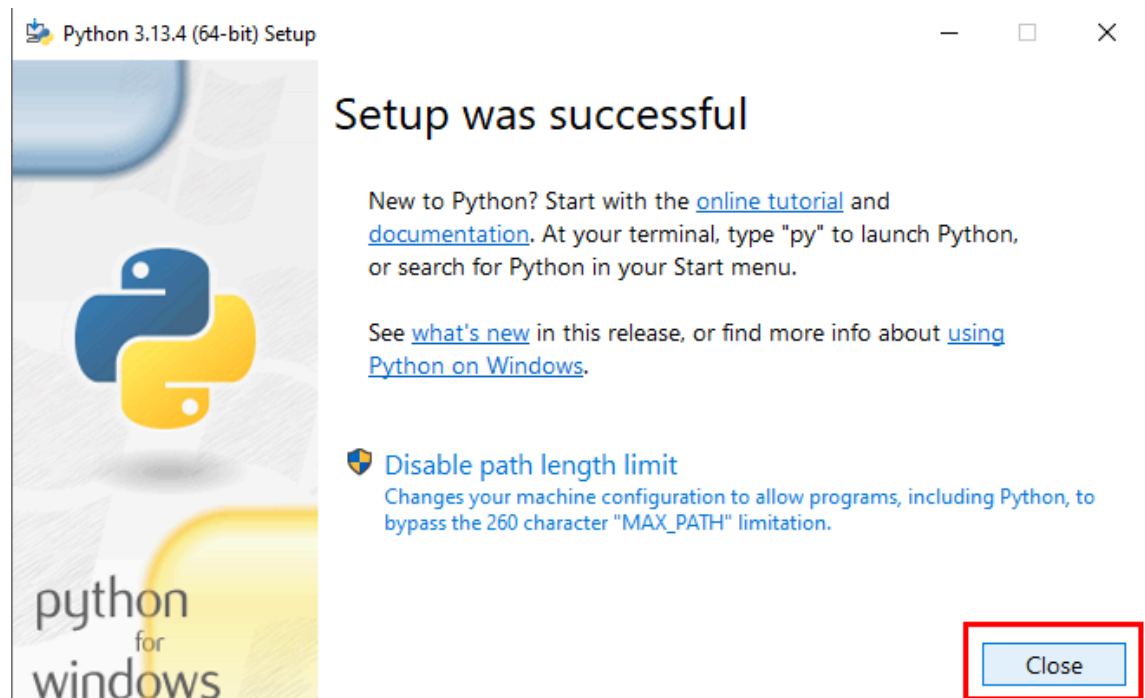
- Baixe e instale o Python 3.7+ em <https://www.python.org/downloads/>
- Durante a instalação, marque as opções "Add Python to PATH" e "Use admin privileges when installing py.exe". **(IMPORTANTE)**.



- Após marcar as opções, clique em "**Install Now**", e aguarde a conclusão da instalação.



- Ao aparecer essa janela abaixo, significa que a instalação foi concluída com sucesso. Logo, clique em "Close".



## 3.2 - Instalação das Dependências

Para instalar o framework Flask e sua extensão/biblioteca, abra o terminal (Prompt de Comando no Windows ou Terminal no Linux/Mac) e digite o seguinte comando, pressionando Enter logo em seguida para executá-lo:

***pip install flask flask\_cors requests***

```
Prompt de Comando
C:\Users\User>pip install flask flask_cors requests
Requirement already satisfied: flask in c:\users\user\appdata\local\programs\python\python313\lib\site-packages (3.1.1)
Requirement already satisfied: flask_cors in c:\users\user\appdata\local\programs\python\python313\lib\site-packages (6.0.0)
Collecting requests
  Downloading requests-2.32.4-py3-none-any.whl.metadata (4.9 kB)
Requirement already satisfied: blinker>=1.9.0 in c:\users\user\appdata\local\programs\python\python313\lib\site-packages (from flask) (1.9.0)
Requirement already satisfied: click>=8.1.3 in c:\users\user\appdata\local\programs\python\python313\lib\site-packages (from flask) (8.2.1)
Requirement already satisfied: itsdangerous>=2.2.0 in c:\users\user\appdata\local\programs\python\python313\lib\site-packages (from flask) (2.2.0)
Requirement already satisfied: jinja2>=3.1.2 in c:\users\user\appdata\local\programs\python\python313\lib\site-packages (from flask) (3.1.6)
Requirement already satisfied: markupsafe>=2.1.1 in c:\users\user\appdata\local\programs\python\python313\lib\site-packages (from flask) (3.0.2)
Requirement already satisfied: werkzeug>=3.1.0 in c:\users\user\appdata\local\programs\python\python313\lib\site-packages (from flask) (3.1.3)
Collecting charset_normalizer<4,>=2 (from requests)
  Downloading charset_normalizer-3.4.2-cp313-cp313-win_amd64.whl.metadata (36 kB)
Collecting idna<4,>=2.5 (from requests)
  Downloading idna-3.10-py3-none-any.whl.metadata (10 kB)
Collecting urllib3<3,>=1.21.1 (from requests)
  Downloading urllib3-2.4.0-py3-none-any.whl.metadata (6.5 kB)
Collecting certifi>=2017.4.17 (from requests)
  Downloading certifi-2025.4.26-py3-none-any.whl.metadata (2.5 kB)
Requirement already satisfied: colorama in c:\users\user\appdata\local\programs\python\python313\lib\site-packages (from click>=8.1.3->flask) (0.4.6)
Downloading requests-2.32.4-py3-none-any.whl (64 kB)
Downloading charset_normalizer-3.4.2-cp313-cp313-win_amd64.whl (105 kB)
Downloading idna-3.10-py3-none-any.whl (70 kB)
Downloading urllib3-2.4.0-py3-none-any.whl (128 kB)
Downloading certifi-2025.4.26-py3-none-any.whl (159 kB)
Installing collected packages: urllib3, idna, charset_normalizer, certifi, requests
Successfully installed certifi-2025.4.26 charset_normalizer-3.4.2 idna-3.10 requests-2.32.4 urllib3-2.4.0
C:\Users\User>
```

### 3.3 - Estrutura de Pastas

Certifique-se de que os arquivos estejam organizados assim:

Servidor Restaurante(PYTHON & SQL)/

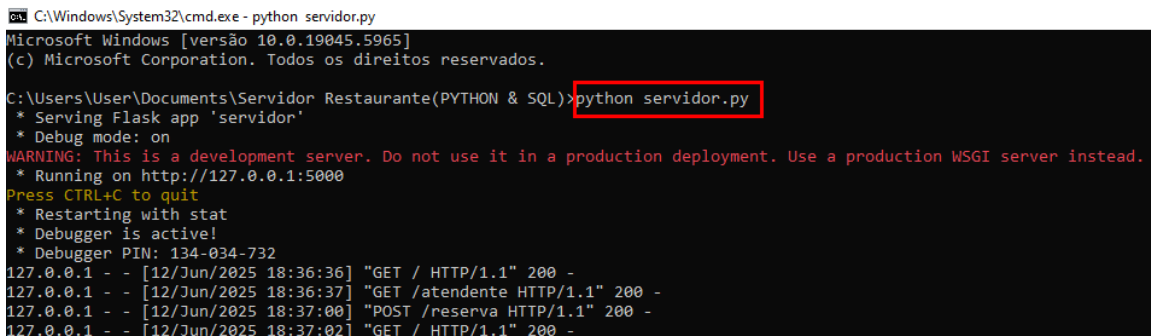
```
|
|
|— servidor.py
|— atendente.py
|— garcom.py
|— gerente.py
|— reservas.db (será criado automaticamente)
|— Templates/
|   |— index.html
|   |— atendente.html
|   |— garcom.html
|   |— gerente.html
```

### 3.4 - Execução do Servidor

No terminal, navegue até a pasta do projeto e digite o seguinte comando, pressionando Enter logo em seguida:

*python servidor.py*

O servidor Flask será iniciado em <http://localhost:5000>.



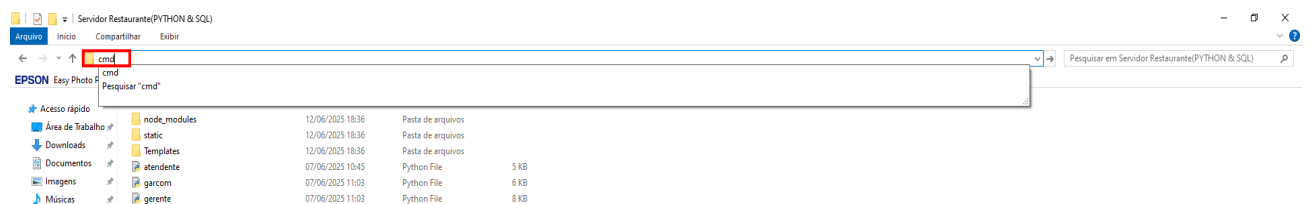
```
C:\Windows\System32\cmd.exe - python servidor.py
Microsoft Windows [versão 10.0.19045.5965]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\User\Documents\Servidor Restaurante(PYTHON & SQL)>python servidor.py
* Serving Flask app 'servidor'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 134-034-732
127.0.0.1 - - [12/Jun/2025 18:36:36] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 18:36:37] "GET /atendente HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 18:37:00] "POST /reserva HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 18:37:02] "GET / HTTP/1.1" 200 -
```

```
C:\Windows\System32\cmd.exe - python servidor.py
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 134-034-732
127.0.0.1 - - [12/Jun/2025 22:05:51] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:05:52] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [12/Jun/2025 22:05:55] "GET /atendente HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:06:34] "POST /reserva HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:06:35] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:06:37] "GET /garcom HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:06:37] "GET /reservas-disponiveis HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:06:52] "POST /confirmar/11 HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:06:52] "GET /reservas-disponiveis HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:06:53] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:06:55] "GET /gerente HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:06:56] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:06:57] "GET /gerente HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:07:04] "GET /relatorio/garcom/Vanessa%20Santos HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:07:05] "GET /relatorio/garcom/Vanessa%20Santos HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:07:08] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:07:10] "GET /atendente HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:07:12] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:07:13] "GET /gerente HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:07:20] "GET /relatorio/mesa/14 HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:07:43] "GET /relatorio/periodo?inicio=2025-06-14&fim=2025-09-13 HTTP/1.1" 200 -
127.0.0.1 - - [12/Jun/2025 22:07:55] "GET / HTTP/1.1" 200 -
```

## Visão do funcionamento do sistema no terminal

**OBSERVAÇÃO:** Uma forma rápida e simples de realizar essa navegação é clicando uma vez na caixa de endereço da pasta do projeto e digitar **cmd**. Após isso, pressione Enter, e a janela do terminal abrirá automaticamente.



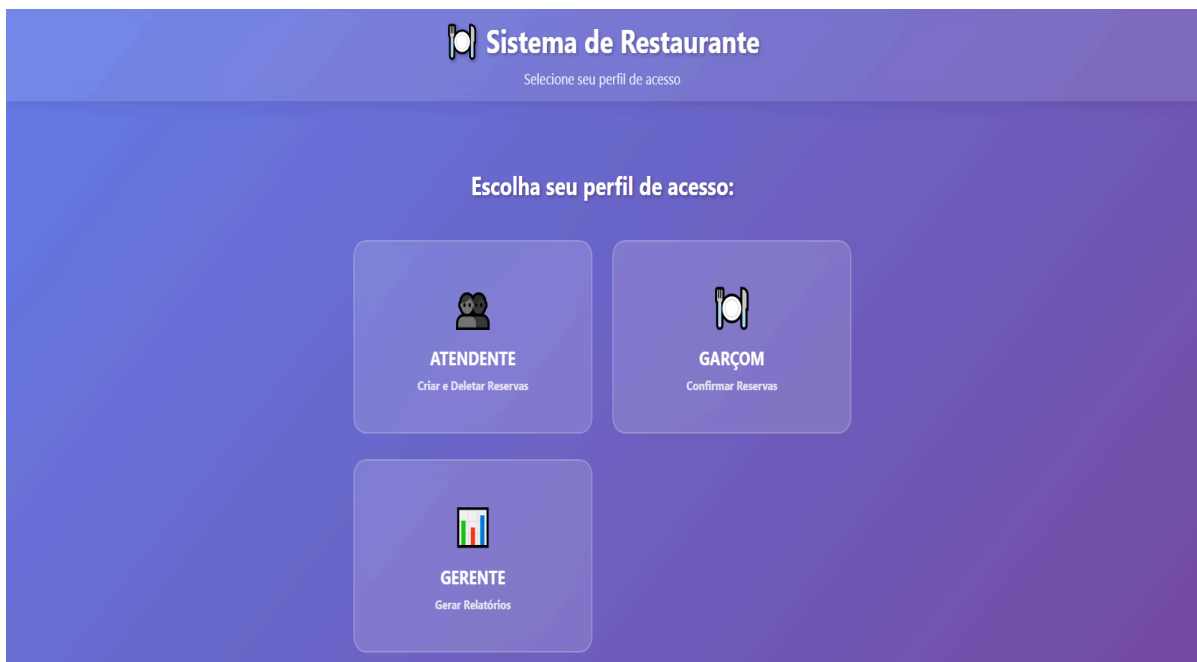
## 3.5 - Acesso via Navegador

Abra o navegador e acesse:

<http://localhost:5000>

Escolha o perfil desejado (Atendente, Garçom ou Gerente) para acessar as funcionalidades.





**OBSERVAÇÃO:** Caso haja algum erro inesperado durante o uso da aplicação, é possível que o framework Flask instalado pelo usuário esteja desatualizado. Para atualizá-lo, abra o Prompt de Comando normalmente (sem navegar até a pasta do projeto), e execute o comando:

***pip install --upgrade flask***

```
C:\Users\User>pip install --upgrade flask
Defaulting to user installation because normal site-packages is not writeable
```

Após isso, feche a guia no navegador onde foi acessada a aplicação, e abra novamente, fazendo mais uma vez o mesmo processo de execução do servidor.

## 4 - Abordagem de comunicação escolhida e justificativa

### 4.1 Abordagem

A comunicação entre clientes (web ou scripts) e o servidor é feita com a abordagem **API-Rest**, juntamente com o protocolo de comunicação **HTTP**. As informações são trocadas em formato **JSON**. O servidor **Flask** expõe rotas para criação, cancelamento, confirmação de reservas e geração de relatórios, que podem ser acessadas tanto por interfaces web quanto por scripts Python.

### 4.2 Justificativa

- **Padronização:** HTTP/REST e JSON são padrões amplamente aceitos, facilitando integração com diferentes plataformas. Além disso, a implementação da linguagem Python com a abordagem API se tornou essencial para o projeto devido a familiaridade que possuem com o desenvolvimento de lógicas de negócio e servidores, e suporte a frameworks necessários para o funcionamento da aplicação.
- **Facilidade de uso:** Permite que qualquer cliente (web, desktop, mobile ou script) interaja com o sistema de forma simples.
- **Escalabilidade:** A arquitetura REST facilita a manutenção e expansão do sistema.
- **Segurança e rastreabilidade:** Todas as operações são centralizadas e registradas no banco de dados.
- **Flexibilidade:** O mesmo backend atende múltiplos perfis e interfaces.

## 5 - Apresentação e demonstração do uso da aplicação

### 5.1 - Explicação

Em um vídeo criado pela equipe desenvolvedora do projeto, o uso da aplicação é demonstrado, além de os integrantes explicarem brevemente sobre o funcionamento e os requerimentos da aplicação.

### 5.2 - URL do vídeo

Segue abaixo o vídeo de apresentação da aplicação:

<https://www.youtube.com/watch?v=TvtX-rq3bxQ>

## 6 - Conclusão

Durante o semestre, foi ensinado aos estudantes sobre não apenas os conceitos básicos de Sistemas distribuídos e mobile, como também, a importância dos mesmos e as mais diversas situações e analogias no qual esse campo da TI é mencionado e utilizado. Foi analisado as consequências que vários tipos de projetos teriam caso não houvesse um fator crucial, que é a comunicação entre processos por meio de redes de comunicações.

A equipe e os demais estudantes puderam notar que, com as áreas da TI em suas constantes expansões e evoluções, a utilização de sistemas distribuídos continuará indispensável, seja na prática, ou até mesmo em simples conceituações durante todo um projeto.