

# AULA 08 - CONFIGURANDO NGINX PARA PHP

Por Sediane Carmem Lunardi Hernandez

1



# AGENDA

- Instalação do PHP
- Configuração do PHP para um domínio específico

Universidade Tecnológica Federal do Paraná  
UTFPR – Câmpus Guarapuava  
Autora: Sediane Carmem Lunardi Hernandez



Esta obra está licenciada com uma Licença Creative Commons Atribuição-NãoComercial-SemDerivações 4.0 Internacional.

Esta licença permite download e compartilhamento do trabalho desde que sejam atribuídos créditos ao(s) autor(es), sem a possibilidade de alterá-lo ou utilizá-lo para fins comerciais.  
Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

This work is licensed under [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)





# DEFININDO O DOMÍNIO

- **Definindo o domínio:**

- [cafeteria.com.br](http://cafeteria.com.br)

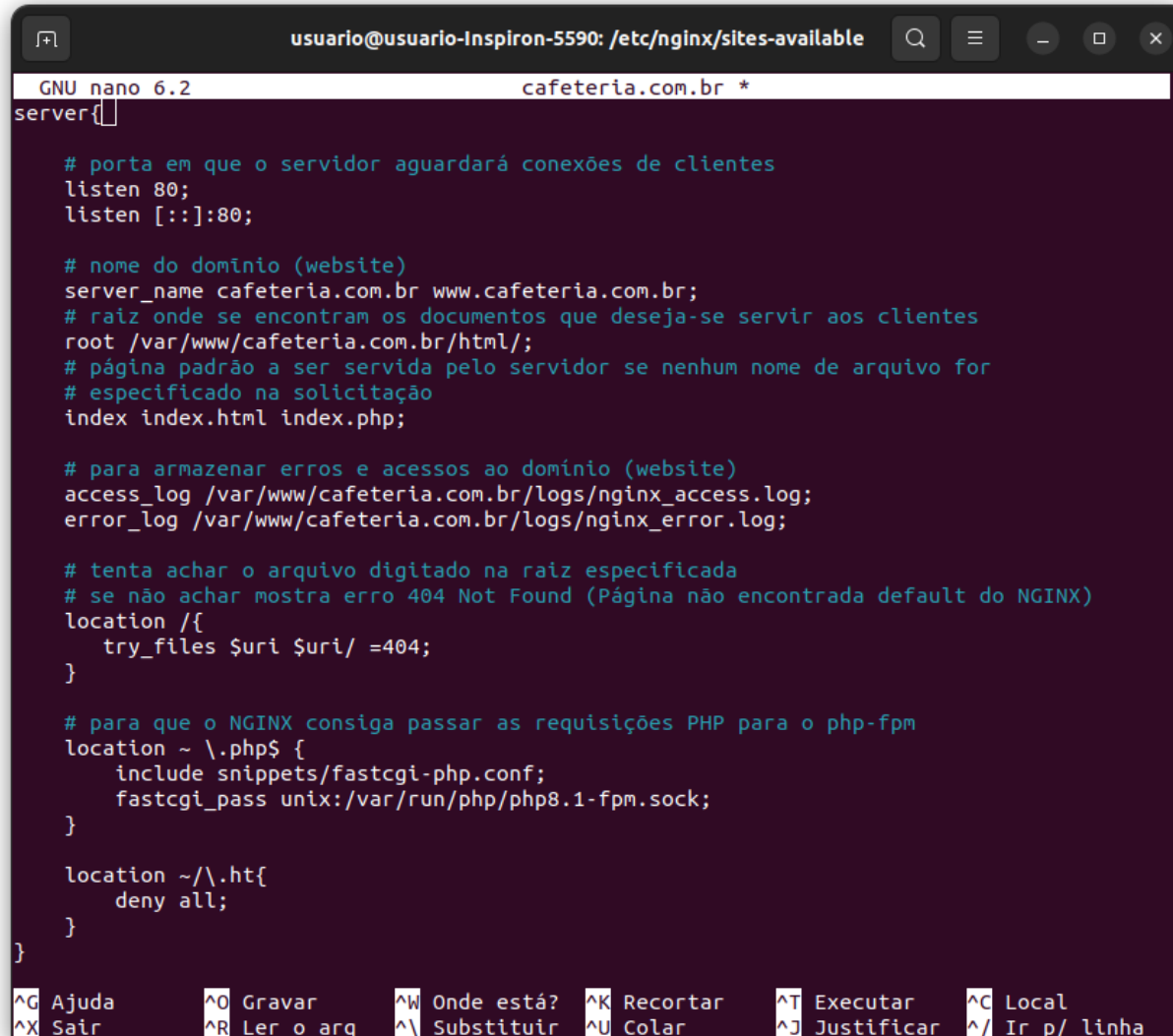
Sugiro você utilizar como parte do nome domínio o seu nome (por exemplo, [cafeteriadamaria.com.br](http://cafeteriadamaria.com.br))

# INICIANDO...

- O Nginx requer um programa externo para lidar com o processamento PHP (*Hypertext Preprocessor*) e atuar como uma ponte entre o próprio interpretador PHP e o servidor web
  - Assim, o NGINX requer configuração adicional:
    - Instalação do php8.1-fpm (*PHP fastCGI process manager*)
      - diz ao Nginx para passar solicitações PHP para este software para processamento
    - Para a comunicação com o bancos de dados baseados em MySQL, instalação do php-mysql (módulo PHP que permite que o PHP se comunique com o MySQL)
- **Passo 1:** Instalar o PHP e o módulo PHP-MySQL
  - \$ sudo apt install php8.1-fpm php-mysql**
  - \$ php -v (retorna a versão do PHP)**

*PHP fastCGI process manager:*  
funciona como um gerenciador  
de processos FastCGI  
(protocolo de comunicação  
entre o servidor web e o PHP)

# CONFIGURANDO...



```
GNU nano 6.2          cafeteria.com.br *
server{

    # porta em que o servidor aguardará conexões de clientes
    listen 80;
    listen [::]:80;

    # nome do domínio (website)
    server_name cafeteria.com.br www.cafeteria.com.br;
    # raiz onde se encontram os documentos que deseja-se servir aos clientes
    root /var/www/cafeteria.com.br/html;
    # página padrão a ser servida pelo servidor se nenhum nome de arquivo for
    # especificado na solicitação
    index index.html index.php;

    # para armazenar erros e acessos ao domínio (website)
    access_log /var/www/cafeteria.com.br/logs/nginx_access.log;
    error_log /var/www/cafeteria.com.br/logs/nginx_error.log;

    # tenta achar o arquivo digitado na raiz especificada
    # se não achar mostra erro 404 Not Found (Página não encontrada default do NGINX)
    location /{
        try_files $uri $uri/ =404;
    }

    # para que o NGINX consiga passar as requisições PHP para o php-fpm
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
    }

    location ~ /\.ht{
        deny all;
    }
}
```

^G Ajuda ^O Gravar ^W Onde está? ^K Recortar ^T Executar ^C Local  
^X Sair ^R Ler o arq ^\ Substituir ^U Colar ^J Justificar ^\_ Ir p/ linha

- **Passo 2:** Editar o *server-block* do seu domínio para que o NGINX consiga servir páginas dinâmicas PHP

Aproveite e adicione as duas linhas para armazenamento de logs (não esqueça de criar o diretório logs e os arquivos sem conteúdo)

# EXPLICAÇÃO DAS DIRETIVAS DO SERVER-BLOCK

- Aqui está o que cada um desses blocos de localização e diretrizes fazem:
  - **listen** — Define em qual porta o Nginx irá escutar. Neste caso, ele irá escutar na porta 80, a porta padrão para o HTTP.
  - **root** — Define o documento root onde os arquivos servidos por este site são armazenados.
  - **index** — Define em que ordem o Nginx irá priorizar os arquivos de index para este site. É uma prática comum listar arquivos index.html com uma precedência superior aos arquivos index.php para permitir uma configuração rápida de uma página inicial de manutenção em aplicações PHP. Você pode ajustar essas configurações para melhor se adaptar às necessidades da sua aplicação.
  - **server\_name** — Define para quais nomes de domínio e/ou endereços IP este bloco de servidor deve responder. **Aponte esta diretiva para o nome de domínio do seu servidor ou endereço IP público.**
  - **location/** — O primeiro bloco de localização inclui uma diretiva `try_files`, que verifica a existência de arquivos ou diretórios que correspondam a uma requisição de URI. Se o Nginx não puder encontrar o recurso apropriado, ele irá retornar um erro 404 (o documento (i.e., recurso que pode ser uma página web) não é encontrado).
  - **location ~ \.php\$** — Este bloco de localização lida com o processamento PHP real, apontando o Nginx para o arquivo de configuração `fastcgi-php.conf` e o arquivo `php8.1-fpm.sock`, que declara qual soquete está associado ao `php-fpm`.
  - **location ~ /\.ht** — O último bloco de localização lida com os arquivos `.htaccess`, que o Nginx não processa. Ao adicionar a diretiva `deny all`, se acontecer de algum arquivo `.htaccess` ser encontrado no caminho do documento root, ele não será apresentado aos visitantes.

# CONFIGURANDO...

- **Passo 3:** Verificar se PHP está executando

```
$ systemctl status php8.1-fpm.service
```

- **Passo 4:** Verificar se os arquivos de configuração do NGINX estão corretos (sem erros de sintaxe)

```
$ sudo nginx -t
```



# TESTANDO O PHP COM O NGINX

**Passo 1:** Criar uma página chamada index.php em /var/www/**your\_domain**/html/

**Passo 2:** Digitar no arquivo criado o código apresentado na figura



```
usuario@usuario-Inspiron-5590: /var/www/cafeateria.com.br/h...
usuario@usuario-Inspiron-5590:~$ cd /var/www/cafeateria.com.br/html/
usuario@usuario-Inspiron-5590:/var/www/cafeateria.com.br/html$ ls -la
total 20
drwxr-xr-x 2 usuario usuario 4096 out 23 11:40
drwxr-xr-x 3 usuario usuario 4096 out 28 15:20 .
drwxr-xr-x 3 usuario usuario 4096 out 27 23:45 ..
-rw-r--r-- 1 usuario usuario  216 out 23 11:46 index.html
-rw-r--r-- 1 root    root      23 out 28 15:20 index.php
usuario@usuario-Inspiron-5590:/var/www/cafeateria.com.br/html$ cat index.php
<?php
    phpinfo();
?>
```

# TESTANDO O PHP COM O NGINX

- **Passo 3:** Reiniciar o servidor NGINX  
`$ sudo systemctl restart nginx`
- **Passo 4:** Acessar o seu domínio pelo navegador  
<http://cafeteria.com.br/index.php>

# MAS....

## ■ Verificar

1. se as duas linhas de logs foram adicionadas ao arquivo *server-block* do domínio (/etc/nginx/sites-available/cafeateria.com.br)
2. se o diretório de logs e os 2 arquivos (nginx\_access.log e nginx\_error.log) também foram criados

```
# para armazenar erros e acessos ao domínio (website)
access_log /var/www/cafeateria.com.br/logs/nginx_access.log;
error_log /var/www/cafeateria.com.br/logs/nginx_error.log;
```

```
usuario@usuario-Inspiron-5590:/var/www/cafeateria.com.br/html$ cd ..
usuario@usuario-Inspiron-5590:/var/www/cafeateria.com.br$ ls -la
total 20
drwxr-xr-x 4 usuario usuario 4096 nov  4 22:18 .
drwxr-xr-x 7 root    root    4096 out 27 19:53 ..
drwxr-xr-x 3 usuario usuario 4096 nov  4 22:54 html
-rw-r--r-- 1 usuario usuario  66 out 27 23:45 info.php
drwxrwxr-x 2 usuario usuario 4096 nov  4 22:19 logs
usuario@usuario-Inspiron-5590:/var/www/cafeateria.com.br$ cd logs
usuario@usuario-Inspiron-5590:/var/www/cafeateria.com.br/logs$ ls -la
total 20
drwxrwxr-x 2 usuario usuario 4096 nov  4 22:19 .
drwxr-xr-x 4 usuario usuario 4096 nov  4 22:18 ..
-rw-rw-r-- 1 usuario usuario 4732 nov  4 22:54 nginx_access.log
-rw-rw-r-- 1 usuario usuario 4073 nov  4 22:53 nginx_error.log
usuario@usuario-Inspiron-5590:/var/www/cafeateria.com.br/logs$
```

# POR QUE PROFESSORA?







# RESPOSTA

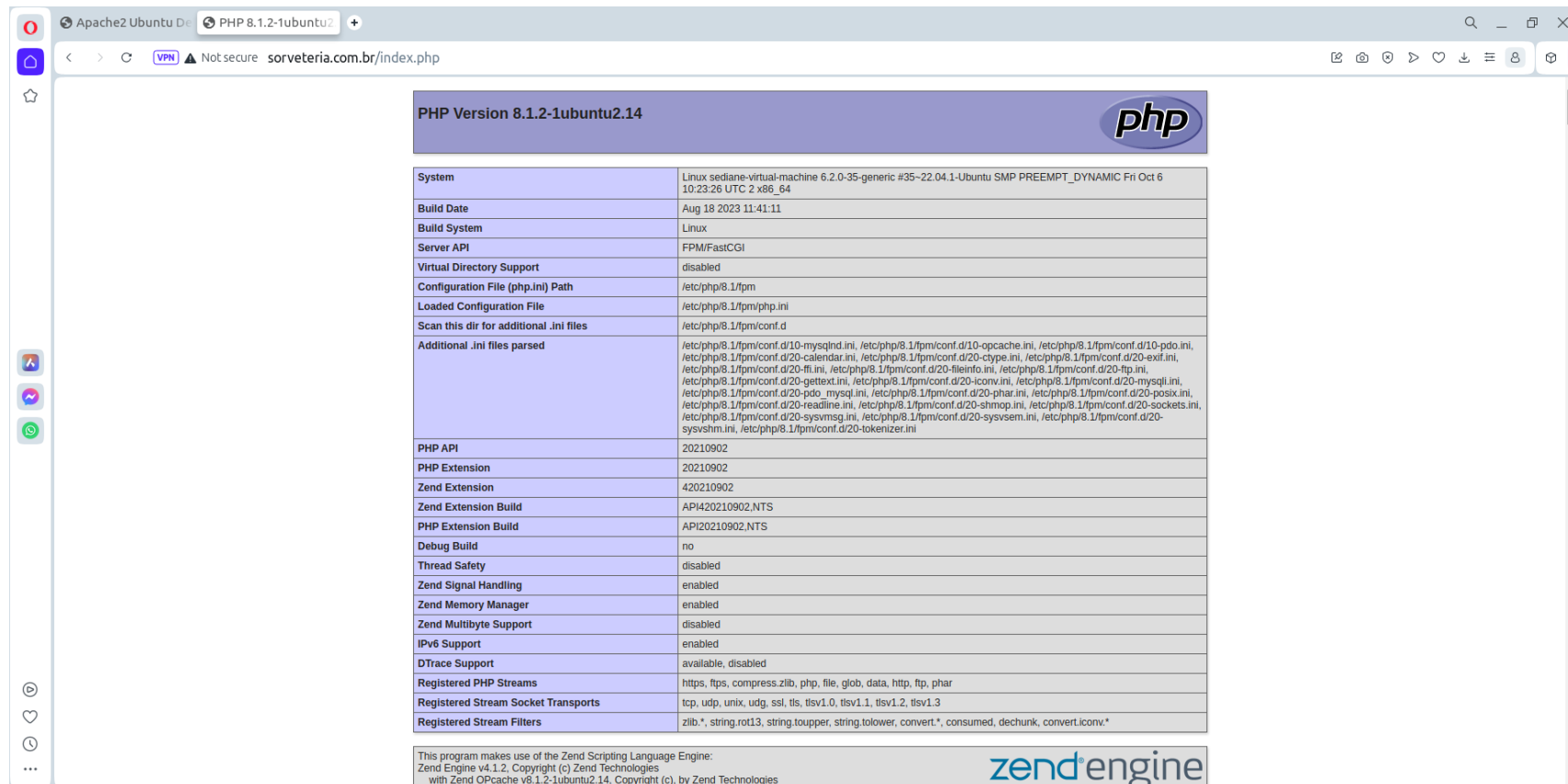
- Porque depois iremos acessar os logs de acesso e todas as requisições serão armazenadas, especialmente no arquivo `nginx_acces.log`.
- Extremamente importante para a realização das próximas atividades relacionadas a logs

# A SAÍDA SERÁ

Apache2 Ubuntu De PHP 8.1.2-1ubuntu2


Not secure sorveteria.com.br/index.php

## PHP Version 8.1.2-1ubuntu2.14



System	Linux sediane-virtual-machine 6.2.0-35-generic #35-22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Fri Oct 6 10:23:26 UTC 2 x86_64
Build Date	Aug 18 2023 11:41:11
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.1/fpm
Loaded Configuration File	/etc/php/8.1/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.1/fpm/conf.d
Additional .ini files parsed	/etc/php/8.1/fpm/conf.d/10-mysqld.ini, /etc/php/8.1/fpm/conf.d/10-opcache.ini, /etc/php/8.1/fpm/conf.d/10-pdo.ini, /etc/php/8.1/fpm/conf.d/20-calendar.ini, /etc/php/8.1/fpm/conf.d/20-ctype.ini, /etc/php/8.1/fpm/conf.d/20-exif.ini, /etc/php/8.1/fpm/conf.d/20-ffi.ini, /etc/php/8.1/fpm/conf.d/20-fileinfo.ini, /etc/php/8.1/fpm/conf.d/20-ftp.ini, /etc/php/8.1/fpm/conf.d/20-gettext.ini, /etc/php/8.1/fpm/conf.d/20-iconv.ini, /etc/php/8.1/fpm/conf.d/20-mysqli.ini, /etc/php/8.1/fpm/conf.d/20-pdo_mysql.ini, /etc/php/8.1/fpm/conf.d/20-phar.ini, /etc/php/8.1/fpm/conf.d/20-posix.ini, /etc/php/8.1/fpm/conf.d/20-readline.ini, /etc/php/8.1/fpm/conf.d/20-shmop.ini, /etc/php/8.1/fpm/conf.d/20-sockets.ini, /etc/php/8.1/fpm/conf.d/20-sysvmsg.ini, /etc/php/8.1/fpm/conf.d/20-sysvsem.ini, /etc/php/8.1/fpm/conf.d/20-sysvshm.ini, /etc/php/8.1/fpm/conf.d/20-tokenizer.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902.NTS
PHP Extension Build	API20210902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v4.1.2. Copyright (c) Zend Technologies  
with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies



# POR FIM...

- Agora que você verificou que o NGINX consegue manusear corretamente páginas php você pode excluir a página `index.php` criada (opcional)

```
:/var/www/cafeteria.com.br/html$ rm index.php
```



# PÁGINAS DINÂMICAS...





# CONSIDERAÇÕES INICIAIS

- Embora você tenha instalado php-mysql no início deste tutorial, instalar o `mysql-server`. Para isso, basta digitar:

```
$ sudo apt install mysql-server
```

# TESTANDO A CONEXÃO DO PHP COM O BANCO DE DADOS

- **Passo 1:** Criar um banco de dados de teste no MySQL

```
$ sudo mysql  
mysql> CREATE DATABASE minha_base_de_dados;
```

- **Passo 2:** Criar um usuário com uma senha para acessar a base de dados criada

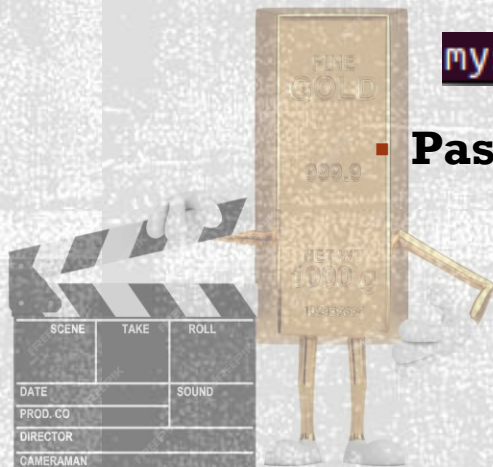
```
mysql> CREATE USER 'usuario'@'%' IDENTIFIED WITH mysql_native_password BY 'password' ;
```

- **Passo 3:** Dar permissão ao usuário para manipular a base de dados e impedir que ele crie ou altere outros bancos de dados do servidor

```
mysql> GRANT ALL ON minha_base_de_dados.* TO 'usuario'@'%';
```

- **Passo 4:** Sair do shell (terminal) do PHP

```
mysql> exit  
Bye
```



**Atenção:** Substituir `usuario` pelo seu nome do seu usuário e `password` por uma senha

**Passo 5:** Entrar no mysql e confirmar a base de dados criada `minha_base_de_dados`

```
$ sudo mysql
```

```
$ SHOW DATABASES;
```



```
mysql> SHOW DATABASES;
+-----+
| Database                |
+-----+
| information_schema      |
| minha_base_de_dados    |
| performance_schema      |
+-----+
3 rows in set (0,01 sec)

mysql> 
```

**Passo 6:** Criar uma tabela `todo_list` na `minha_base_de_dados`

```
mysql> CREATE TABLE minha_base_de_dados.todo_list (  
-> item_id INT AUTO_INCREMENT,  
-> content VARCHAR(255),  
-> PRIMARY KEY(item_id)  
-> );  
Query OK, 0 rows affected (0,04 sec)
```

**Passo 7:** Inserir valores na tabela com o seguinte comando (pode repetir várias vezes)

```
mysql> insert into minha_base_de_dados.todo_list (content) values ("Açúcar");  
Query OK, 1 row affected (0,02 sec)
```

```
mysql> insert into minha_base_de_dados.todo_list (content) values ("Leite");  
Query OK, 1 row affected (0,01 sec)
```

**Passo 8:** Para verificar se os itens foram inseridos na tabela:

```
mysql> select * from minha_base_de_dados.todo_list;  
+-----+-----+  
| item_id | content |  
+-----+-----+  
|      4 | Açúcar |  
|      5 | Leite  |  
|      6 | Café   |  
+-----+-----+  
3 rows in set (0,00 sec)
```

**Passo 9:** Sair do shell do MySQL

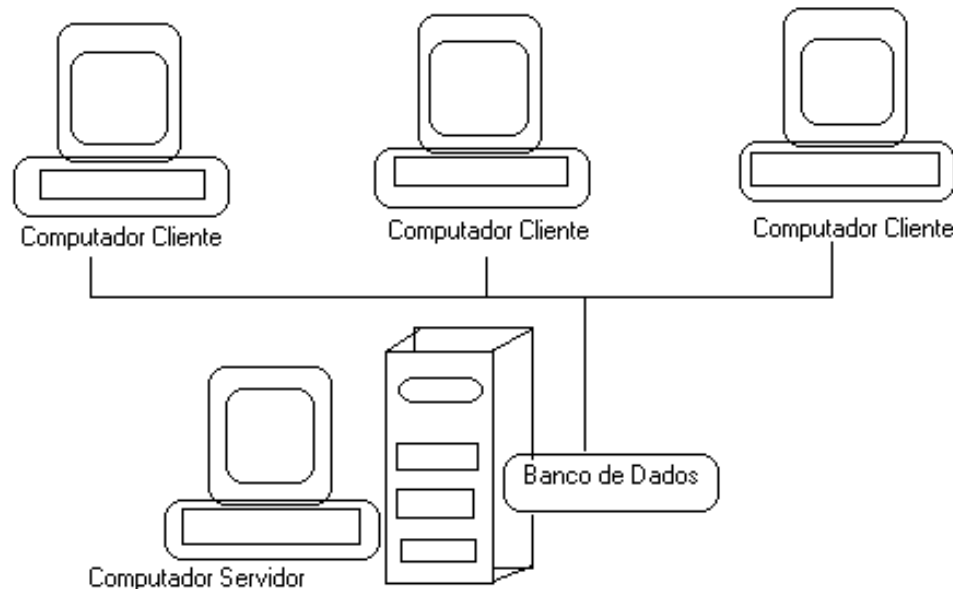
```
mysql> exit  
Bye
```



# CONECTANDO PHP AO MYSQL

**Passo 1:** Criar um arquivo .php para essa conexão (a página PHP deve ficar no diretório html do domínio escolhido para que seja servida aos clientes web - *browsers*)

```
/var/www/cafeteria.com.br/html$ pico todo_list.php
```



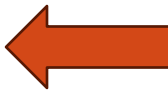
# CONECTANDO PHP AO MYSQL (CONT.)

**Passo 2:** Colar o código abaixo no arquivo

**/var/www/cafeateria.com.br/html/todo\_list.php**

```
<?php
$user = "user";
$password = "password";
$database = "database";
$table = "todo_list";

try {
    $db = new PDO("mysql:host=localhost;dbname=$database", $user, $password);
    echo "<h2>TODO</h2><ol>";
    foreach($db->query("SELECT content FROM $table") as $row) {
        echo "<li>" . $row['content'] . "</li>";
    }
    echo "</ol>";
} catch (PDOException $e) {
    print "Error!: " . $e->getMessage() . "<br/>";
    die();
}
```

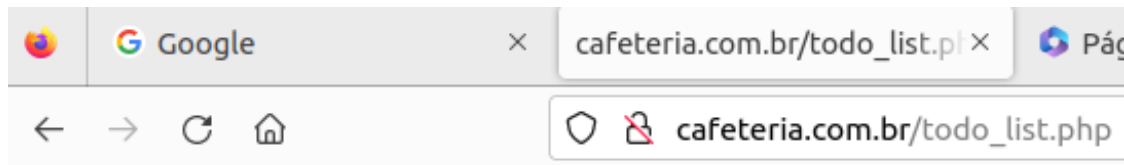


**modificar**

O arquivo se conecta a base de dados criada e mostra os itens inseridos

# POR FIM...

- **Passo 3:** No navegador digite:  
[http://cafeteria.com.br/todo\\_list.php](http://cafeteria.com.br/todo_list.php)

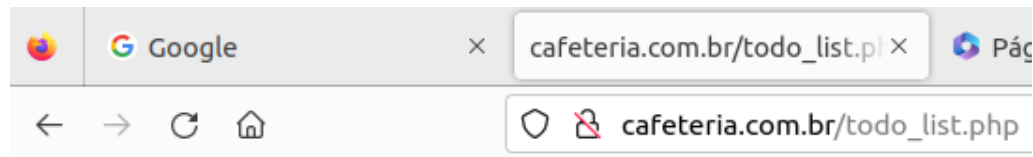


## TODO

1. Açúcar
2. Leite
3. Café

# POR FIM...

- **Passo 4:** Adicionando mais item a base de dados e exibindo código fonte (código PHP não é mostrado > processamento acontece no servidor):



## TODO

1. Açúcar
2. Leite
3. Café
4. Bolacha
5. Bolo







**PARA IR ALÉM...**



# TRABALHANDO COM FORMULÁRIOS

- Criar dois arquivos:
  1. `mensagem.php` (formulário que solicita o registro do nome, e-mail, telefone e mensagem)
  2. `resposta.php` (mostra mensagem que os dados do formulário foram recebidos)

# TRABALHANDO COM FORMULÁRIOS (CONT.)

**Observação:** a ideia aqui não é explicar programação PHP, mas sim só como fazer o NGINX servir páginas dinâmicas com o PHP



**/var/www/cafeteria.com.br/html/mensagem.php**

```
<?php
if (isset($_POST['BTEnvia'])) {

//Variaveis de POST, Alterar somente se necessário
//=====
$nome = $_POST['nome'];
$email = $_POST['email'];
$telefone = $_POST['telefone'];
$mensagem = $_POST['mensagem'];
//=====

//REMETENTE --> ESTE EMAIL TEM QUE SER VALIDO DO DOMINIO
//=====
$email_remetente = "email@doseudominio"; // deve ser uma conta de email do
seu dominio
//=====

//Configurações do email, ajustar conforme necessidade
//=====
$email_destinatario = "email@querecebe"; // pode ser qualquer email que
receberá as mensagens
$email_reply = "$email";
$email_assunto = "Contato formmail"; // Este será o assunto da mensagem
//=====

//Monta o Corpo da Mensagem
//=====
$email_conteudo = "Nome = $nome \n";
$email_conteudo .= "Email = $email \n";
$email_conteudo .= "Telefone = $telefone \n";
$email_conteudo .= "Mensagem = $mensagem \n";
//=====

//Seta os Headers (Alterar somente caso necessario)
//=====
$email_headers = implode ( "\n",array ( "From: $email_remetente", "Reply-
To: $email_reply", "Return-Path: $email_remetente","MIME-Version: 1.0","X-
Priority: 3","Content-Type: text/html; charset=UTF-8" ) );
//=====

//Enviando o email
//=====
if (mail ($email_destinatario, $email_assunto, nl2br($email_conteudo),
$email_headers)){
echo "</b>E-Mail enviado com sucesso!</b>";
}
else{
echo "</b>Falha no envio do E-Mail!</b>"; }
//=====
}
?>
```

# CÓDIGOS

```
<form action="resposta.php" method="POST">
<p>
Nome:<br />
<input type="text" size="30" name="nome">
</p>
<p>
E-mail:<br />
<input type="text" size="30" name="email">
</p>
<p>
Telefone:<br />
<input type="text" size="35" name="telefone">
</p>
<p>
Mensagem:<br />
<input type="text" size="35" name="mensagem">
</p>
<p>
<input type="submit" name="BTEnvia" value="Enviar">
<input type="reset" name="BTApaga" value="Apagar">
</p>
```



Quando o formulário for enviado  
a página resposta.php é  
mostrada

**/var/www/cafeteria.com.br/html/resposta.php**

```
<?php

echo "Dados recebidos com sucesso!"

?>
```



# SAÍDA

cafeateria.co X Bem-vindo a p Página Inic Conl

← → ↻ 🏠 [cafeateria.com.br/mensagem.php](https://cafeateria.com.br/mensagem.php)

Nome:

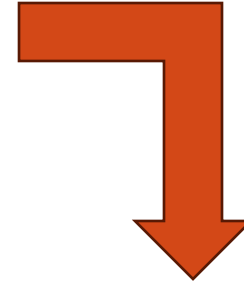
E-mail:

Telefone:

Mensagem:

Enviar

Apagar



Caixa de entrac X PowerPoint | M X Configuração N X

← → ↻ 🏠 [cafeateria.com.br/resposta.php](https://cafeateria.com.br/resposta.php)

Dados enviados com sucesso!

# ATIVIDADE

# TAREFA

1. Configurar mais um domínio para páginas dinâmicas utilizando este tutorial.





# FOI DIVERTIDO...

- Até a próxima...



# REFERÊNCIAS

- [https://computingforgeeks.com/install-nginx-with-php-fpm-on-ubuntu-linux/?expand\\_article=1](https://computingforgeeks.com/install-nginx-with-php-fpm-on-ubuntu-linux/?expand_article=1)
- <https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-on-ubuntu-20-04-pt>
- <https://king.host/wiki/base-de-conhecimento/envio-pelo-site/>
- Algumas figuras foram retiradas de <https://br.freepik.com>.

