



# Whatsapp



NDS-03

WhatsApp group



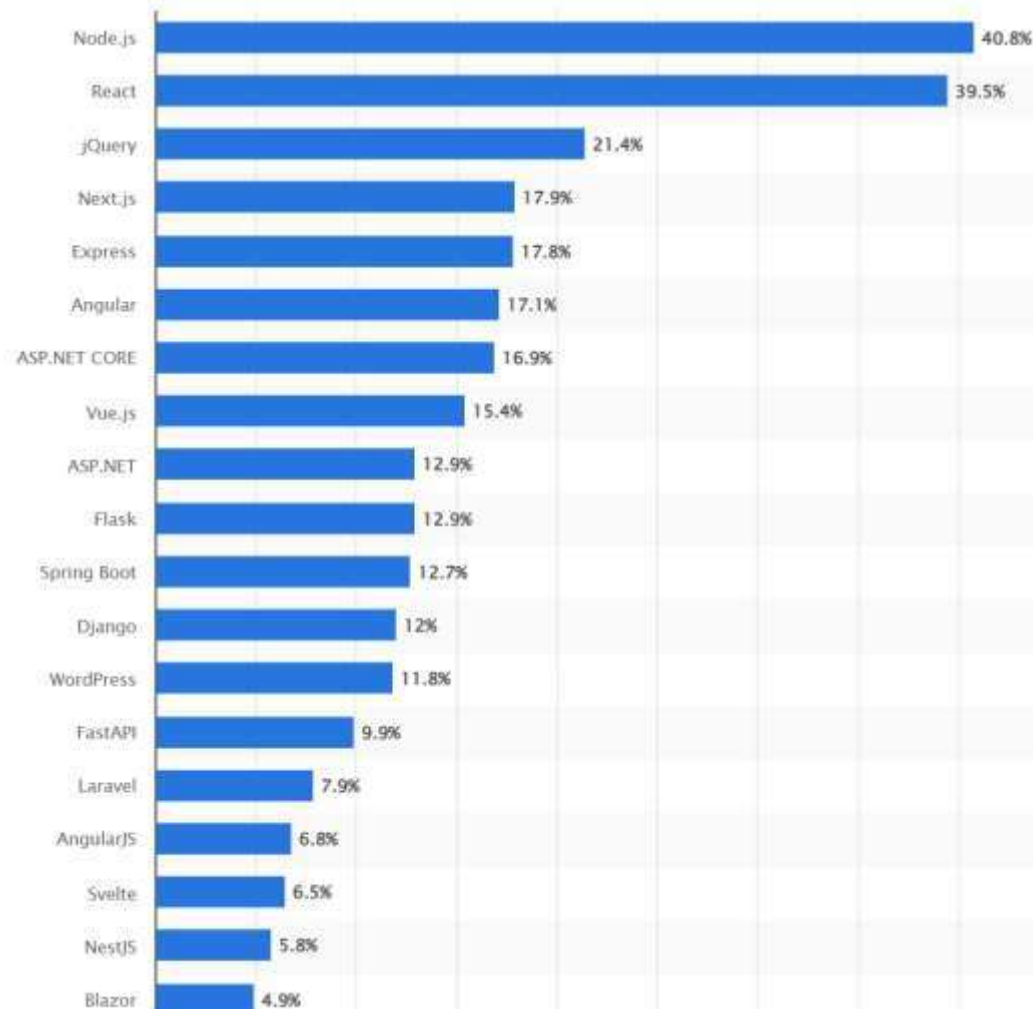
Scan this QR code using the WhatsApp camera to join  
this group



@fpftech.educacional

Technology &amp; Telecommunications &gt; Software

## Most used web frameworks among developers worldwide, as of 2024



## DOWNLOAD



PDF



XLS



PNG



PPT

## Source

[→ Show sources information](#)[→ Show publisher information](#)[→ Use Ask Statista Research Service](#)

## Release date

July 2024

## Region

Worldwide

## Survey time period

May 19 to June 20, 2024

## Number of respondents

48,503 respondents

## Special properties

Software developers

## Method of interview

Online survey

## Supplementary notes

Original question: "Which web frameworks

# Resolução do Exercício de presença

	<b>CAPACITAÇÃO EM TECNOLOGIA, GESTÃO E COMPUTAÇÃO</b>  <b>CURSO: Desenvolvimento de Software</b>	
---	--	--

<b>Aluno(a):</b>	<b>Data:</b> <b>07/05/2025</b>
<b>Professor(a): Jonatas S. Lopes</b>	

Exercício valendo presença (TEMA: Python e Banco de Dados)

## 1. Em python, o que esse código está fazendo?

```
for key, item in fruits.items():  
    print(f"Esta é a {chave} com o valor {valor}")
```

- a) Ele está iterando uma lista de frutas e imprimindo o índice e o nome de cada fruta.
- b) Ele está iterando os itens do dicionário "fruits" e printa cada chave e valor.
- c) Ele está acessando apenas os valores do dicionário "fruits" e imprimindo-os em ordem alfabética.
- d) Ele está criando um dicionário com as chaves e valores de "fruits" e imprimindo cada item.

## 2. Qual a palavra reservada em python que cria uma função nomeada?

# Resolução do Exercício de presença

```
gabarito = ['b', 'a', 'c', 'a', 'd', 'd', 'd', 'a', 'b', 'a', 'c',  
'a', 'b', 'a', 'd', 'c', 'b', 'd', 'd', 'b']
```

```
total = len(gabarito) #len=largura, comprimento
```

```
respostas = ['a', 'b', ...]
```

```
acertos = sum(1 for c, t in zip(gabarito, respostas) if c.lower()  
== t.lower())
```

```
# zip combina 'gabarito' e 'respostas', filtrando valores iguais e  
mesmo índice = ['c', 'a', 'd', ...]
```

```
# dessa nova lista filtrada... Todos viram número 1 = [1, 1, 1, 1, 1, 1, ...]
```

```
# realiza-se a soma de todos os elementos e salva essa soma na variável
```

```
acertos
```

```
print(f"{acertos}/{total}")
```

# Resolução do Exercício de presença

## Em forma de função?

```
def comparar_teste(teste):
```

```
    gabarito = ['b', 'a', 'c', 'a', 'd', 'd', 'd', 'a', 'b', 'a', 'c', 'a',  
    'b', 'a', 'd', 'c', 'b', 'd', 'd', 'b']
```

```
    iguais = sum(1 for c, t in zip(gabarito, teste) if c == t.lower())
```

```
    total = len(gabarito)
```

```
    porcentagem = int((iguais / total) * 100)
```

```
    return f"{iguais}/{total} = {porcentagem}%"
```

**Agora, chame a função dentro de um print passando suas respostas**





# Entendendo o arquivo models.py

É onde é definida os modelos de dados da aplicação. Um modelo é uma classe Python que herda da biblioteca `django.db.models.Model` e cada modelo mapeia diretamente para uma tabela no banco de dados.

```
models.py

from django.db import models

# Create your models here.

class Cliente(models.Model):

    name = models.CharField(
        db_column='tx_name',
        max_length=100,
        null=False
    )

    age = models.IntegerField(
        db_column='tx_age',
        null=False
    )
```



cliente	
tx_name	varchar(100)
tx_age	integer
id	integer

Tabela do  
banco de  
dados

# Salvar no banco postgres

No Terminal do seu projeto, digite:

```
ects\djangoProject> python manage.py showmigrations
```

[ ] ainda não aplicadas  
[ X ] aplicadas no banco postgres ->

```
[ ] 0001_initial
[ ] 0002_remove_content_type_name
sessions
[ ] 0001_initial
teste
(no migrations)
```

Nosso projeto ainda não tem ->



# Salvar no banco postgres

Migrações são como receita de bolo que o Django vai seguir e criar no DB (postgres)

Para criar sua migração para o banco de dados digite:

```
ts\djangoProject> python manage.py makemigrations
```

Ele mostra os modelos criados nas migrações (receita):

```
(.venv) PS C:\Users\jonatas.lopes\PycharmProjects\djangoProject> python manage.py makemigrations
Migrations for 'teste':
  teste\migrations\0001_initial.py
    + Create model Client
    + Create model Employee
    + Create model Product
    + Create model Sale
(.venv) PS C:\Users\jonatas.lopes\PycharmProjects\djangoProject>
```

# Salvar no banco postgres

Agora, com a migração criada, podemos agora aplicar (rodar) a migração e efetivamente criar no banco de dados postgres:

```
ts\djangoProject> python manage.py migrate
```

Ele aplicou todas as migrações para o banco.

```
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying auth.0010_alter_group_name_max_length... OK
Applying auth.0011_update_proxy_permissions... OK
Applying auth.0012_alter_user_first_name_max_length... OK
Applying sessions.0001_initial... OK
Applying teste.0001_initial... OK
(.venv) PS C:\Users\jonatas.lopes\PycharmProjects\djangoProject>
```

# Salvar no banco postgres

Podemos agora digitar novamente o “showmigrations” pra ver a mudança

```
ects\djangoProject> python manage.py showmigrations
```

```
[X] 0009_alter_user_last_name_max_length
[X] 0010_alter_group_name_max_length
[X] 0011_update_proxy_permissions
[X] 0012_alter_user_first_name_max_length
contenttypes
[X] 0001_initial
[X] 0002_remove_content_type_name
sessions
[X] 0001_initial
teste
[X] 0001_initial
(.venv) PS C:\Users\jonatas.lopez\PycharmProjects\djangoProject>
```

[ X ] todas aplicadas no banco

# Obrigado!



    @fpftech.educacional