TinyDB

BANCO DE DADOS ORIENTADO A DOCUMENTOS.

BY: DAVI LUCCIOLA

CONTEUDOS

01 INTRODUÇÃO

02 INSERINDO REGISTROS

03 CONSULTAS

04 ATUALIZANDO/REMOVENDO REGISTROS

O QUE É O TINYDB?

O TinyDB é um banco de dados **orientado a documentos** utilizado para sistemas menores que utiliza um JSON como banco de dados. Pode ser utilizado como alternativa ao SQLite3.

DOCUMENTOS:

Documentos são estruturas de dados onde cada documento pode ter sua propria estrutura. No TinyDB será um quase **um dícionario do Python**.

CRIANDO O BANCO DE DADOS

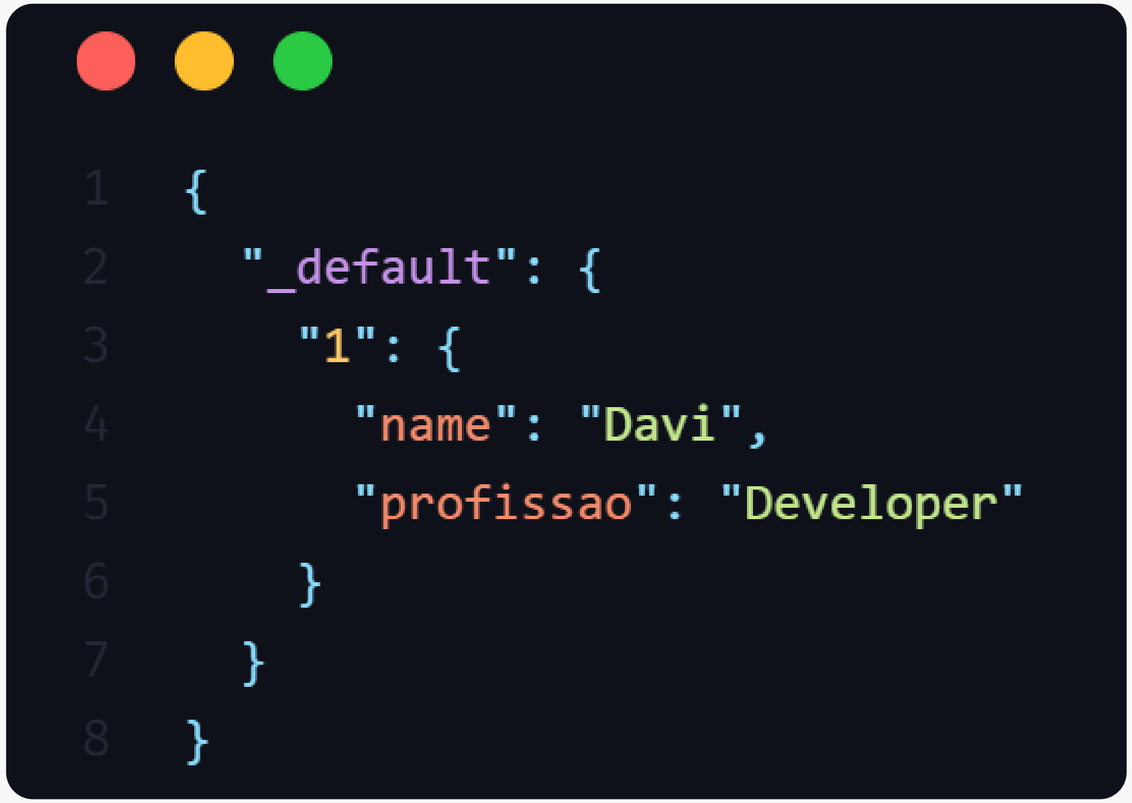
```python
from tinydb import TinyDB

db = TinyDB('./database.json', indent=4)
document = {'name': 'Davi', 'profissao': 'Developer'}

db.insert(document)
```

No código acima estamos criando o nosso banco com a classe “TinyDB”, e passamos como parametro o path do JSON que será o nosso banco de dados.

CRIANDO O BANCO DE DADOS

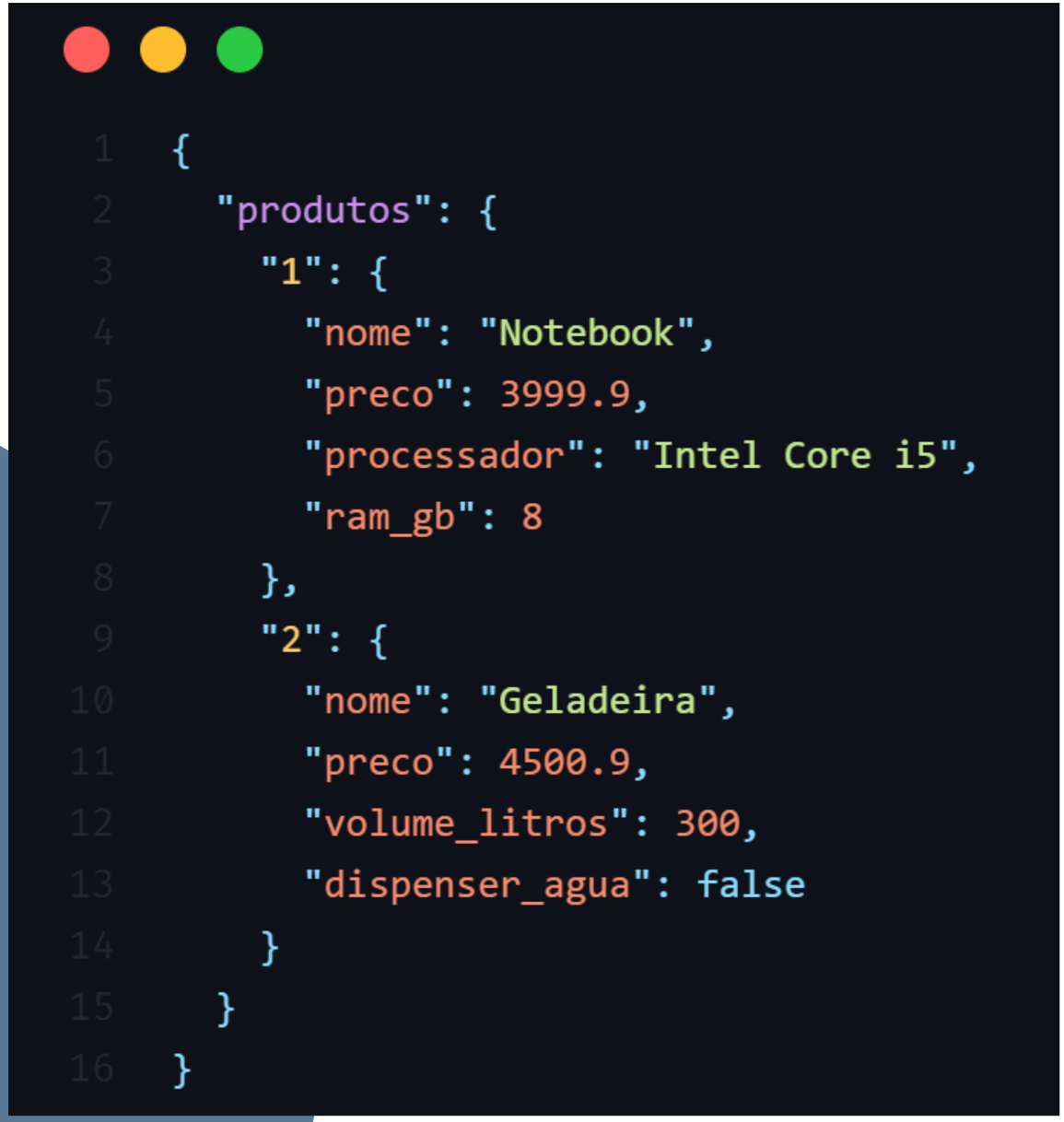
Ao lado temos um exemplo do JSON criado como banco de dados.

Cada dado será armazenado em uma “tabela”, que no JSON será representada por uma chave, sendo a tabela padrão “_default”



INSERINDO DADOS

```python
from tinydb import TinyDB

db = TinyDB('./database.json', indent=2)
table_produtos = db.table('produtos') # Podemos atribuir a tabela a uma variavel

produto1 = {'nome': 'Notebook', 'preco': 3999.90, 'processador':'Intel Core i5', 'ram_gb': 8}
produto2 = {'nome': 'Geladeira', 'preco': 4500.90, 'volume_litros': 300, 'dispenser_agua': False}

db.table('produtos').insert(produto1) # Podemos inserir acessando a tabela diretamente
table_produtos.insert(produto2) # Ou podemos inserir através da variavel da tabela
```



```
{
  "produtos": {
    "1": {
      "nome": "Notebook",
      "preco": 3999.9,
      "processador": "Intel Core i5",
      "ram_gb": 8
    },
    "2": {
      "nome": "Geladeira",
      "preco": 4500.9,
      "volume_litros": 300,
      "dispenser_agua": false
    }
  }
}
```

CONSULTANDO DADOS

```python
from tinydb import TinyDB

db = TinyDB('./database.json', indent=2)

# Todos os Produtos
db.table('produtos').all()
```

BUSCANDO DADOS

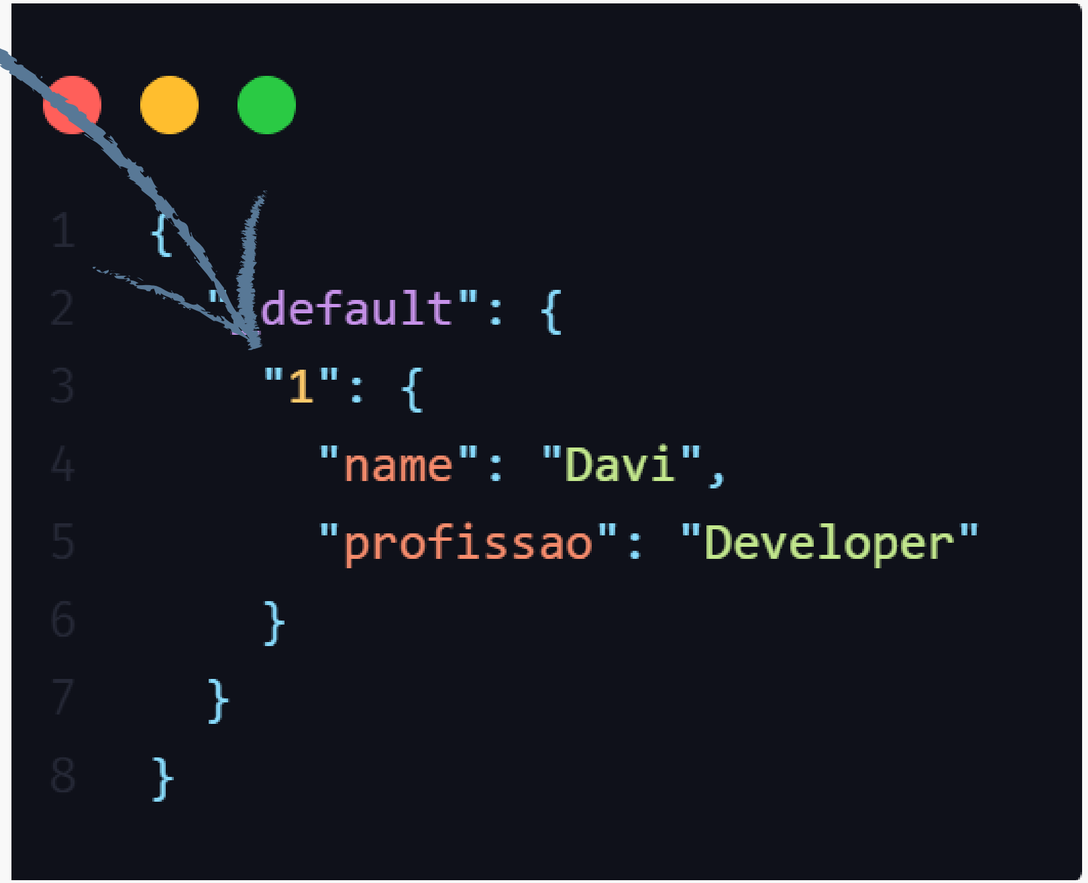
```python
from tinydb import Query

Produto = Query()

# Filtrando por atributo
db.table('produtos').search(Produto.preco < 4000)

# Buscando um dado especifico por chaves
db.table('produtos').get(Produto['name'] == 'Geladeira')
```

BUSCANDO DADOS

```python
from tinydb import Query

Produto = Query()

# Operador 'E' - 'and' do python
db.table('produtos').search(Produto.nome.search('Note') & Produto.preco < 4000)

# Operador 'OU' - 'or' do python
db.table('produtos').search(Produto.nome == 'Geladeira' | Produto.nome == 'Fogão')
```

DOC ID

Quando buscamos uma informação com os métodos “search” ou “get”, o TinyDB nos trará um Documento, esse documento terá um “doc_id”, que será o identificador unico daquele documento em nosso banco de dados.



ATUALIZANDO DADOS

```python
from tinydb import TinyDB, Query

db = TinyDB('./database.json', indent=2)
table_produtos = db.table('produtos')

Produto = Query()
notebook = table_produtos.get(Produto.nome == 'Notebook')
table_produtos.update({'preco': 2000}, doc_ids=[notebook.doc_id])
```

REMOVENDO DADOS

```python
from tinydb import TinyDB, Query

db = TinyDB('./database.json', indent=2)
table_produtos = db.table('produtos')

Produto = Query()
geladeira = table_produtos.get(Produto.nome == 'Geladeira')
table_produtos.remove(doc_ids=[geladeira.doc_id])
```

OBRIGADO!