

UNIVERSIDADE ESTADUAL DO RIO GRANDE DO NORTE
DIRETORIA DE EDUCAÇÃO À DISTÂNCIA
TECNOLOGIA EM SISTEMAS PARA INTERNET

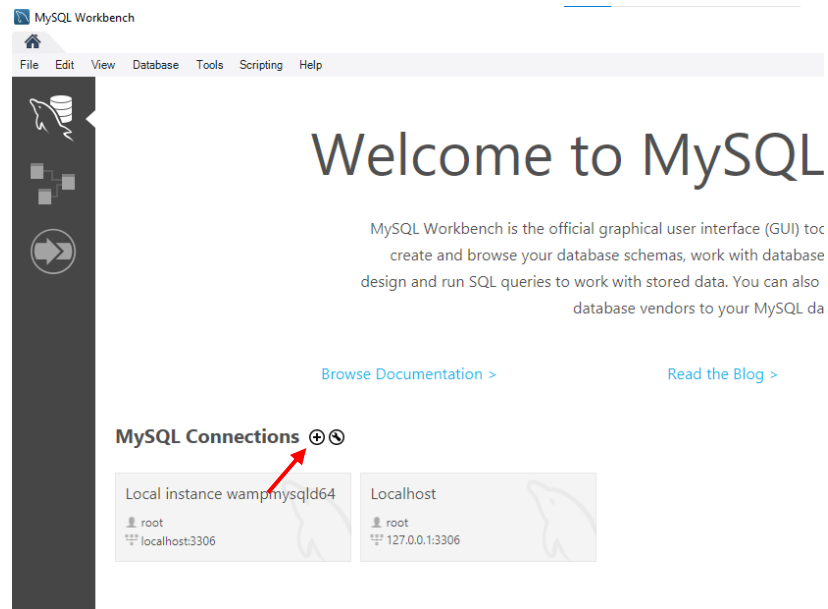
Artur Ferreira Leão Neto

Ellen Alves dos Santos Leão

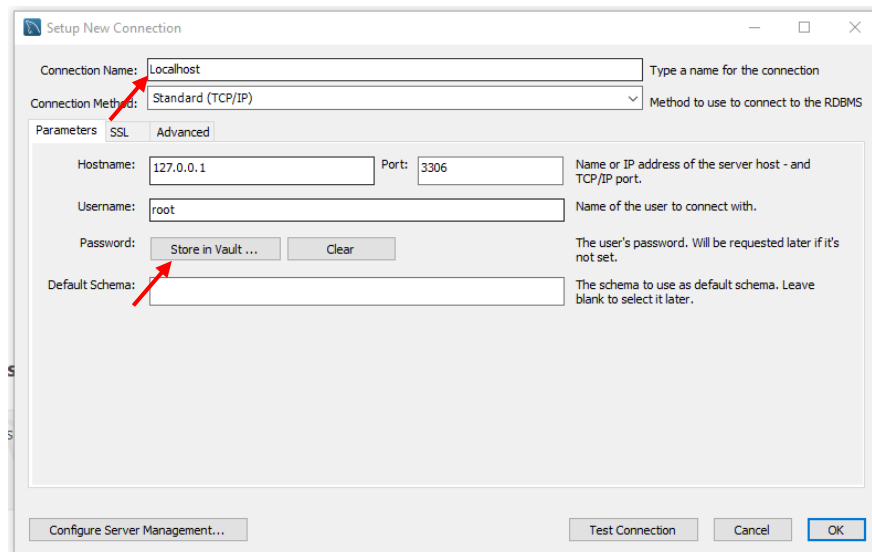
Projeto Integrador: criação e integração do banco de dados com o back-end.

1. Criação do banco de dados

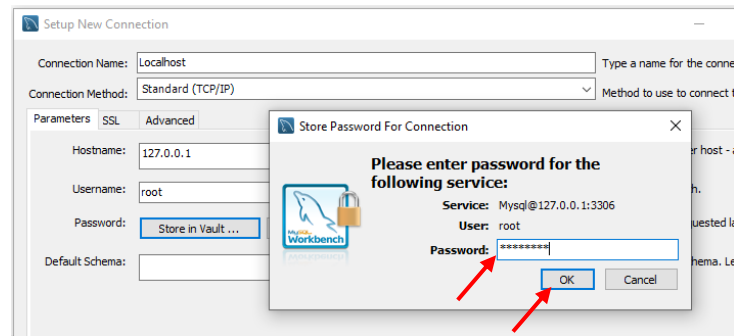
1.1. Criar nova conexão



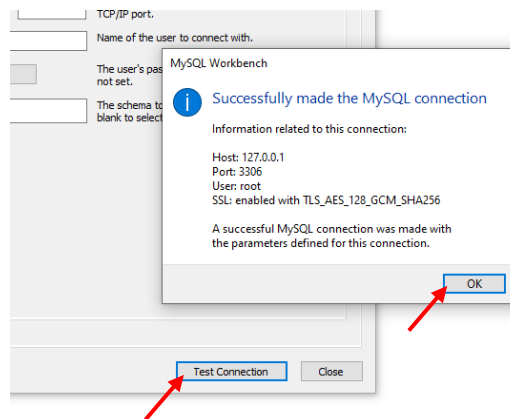
1.2. Configurar nova conexão



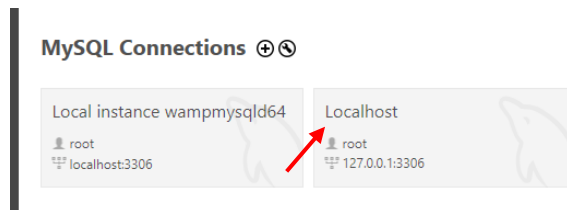
1.3. Senha para conexão



1.4. Testar conexão:



1.5. Abrir banco:



1.6. Criação do banco



1.7. Criação das tabelas (exemplo de uma tabela):

```
1 • USE pizza_lab;
2
3 • CREATE TABLE cliente(
4     id_cliente INT NOT NULL AUTO_INCREMENT,
5     nome VARCHAR(30) NOT NULL,
6     data_nascimento DATE,
7     sexo ENUM('M', 'F'),
8     telefone VARCHAR(15) NOT NULL,
9     email VARCHAR(20) NOT NULL,
10    rua VARCHAR(30),
11    numero VARCHAR(15),
12    bairro VARCHAR(30),
13    complemento VARCHAR(50),
14    PRIMARY KEY (id_cliente)
15 ) DEFAULT CHARSET = utf8mb4;
```

2. Conexão com o banco de dados

2.1. Configurações básicas do banco no arquivo setting.py da pasta “setup”:

```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.mysql",
        "NAME": "pizza_lab",
        "USER": "root",
        "PASSWORD": "Amora@2025",
        "HOST": "localhost",
        "PORT": 3306,
    }
}
```

2.2. Alteração da língua e fuso horário;

```
LANGUAGE_CODE = "pt-br"

TIME_ZONE = "America/Fortaleza"
```

2.3. Gerar classes das tabelas já existentes no banco;

```
(.venv) PS C:\Users\artur\Documents\Back\Unidade 2\pizza_lab - backup> python manage.py inspectdb
```

Resultado do comando no arquivo models.py na pasta do app:

```

class Pagamento(models.Model):
    id_pagamento = models.AutoField(primary_key=True)
    id_pedido = models.IntegerField(blank=True, null=True)
    data_pagamento = models.DateTimeField(blank=True, null=True)
    forma_pagamento = models.CharField(max_length=30, blank=True, null=True)
    valor_pago = models.DecimalField(
        max_digits=7, decimal_places=2, blank=True, null=True
    )

    class Meta:
        managed = False
        db_table = "pagamento"


class Pedido(models.Model):
    id_pedido = models.AutoField(primary_key=True)
    id_cliente = models.IntegerField(blank=True, null=True)
    data_pedido = models.DateTimeField(blank=True, null=True)
    situacao = models.CharField(max_length=30, blank=True, null=True)
    endereco_entrega = models.CharField(max_length=100, blank=True, null=True)

    class Meta:
        managed = False
        db_table = "pedido"

```

3. Explicação das operações CRUD

3.1. Criação das views



```

views.py
pizza_lab > views.py > ...
1  from django.views.generic import (
2      ListView,
3      CreateView,
4      UpdateView,
5      DeleteView,
6      TemplateView,
7  )
8  from django.urls import reverse_lazy
9  from .models import Pizza
10 from django.http import HttpResponseRedirect
11
12 # Criação das Views ou CRUDs
13

```

Home:

```
# View Home
class HomeView(TemplateView):
    template_name = "pizza_lab/home.html"
```

Visualizar e editar menu:

```
# View para visualizar e editar menu
class MenuListView(ListView):
    model = Pizza
    template_name = "pizza_lab/pizza_list.html"
    context_object_name = "menu"
```

Atualizar pizza:

```
# View para atualizar pizza
class MenuUpdateView(UpdateView):
    model = Pizza
    fields = ["nome", "descricao", "tamanho", "preco"]
    success_url = reverse_lazy("menu")
```

Deletar pizza:

```
# View para deletar pizza
class MenuDeleteView(DeleteView):
    model = Pizza
    success_url = reverse_lazy("menu")
```

Criar pizza:

```
# View para criar pizza
class MenuCreateView(CreateView):
    model = Pizza
    fields = ["nome", "descricao", "tamanho", "preco", "tipo"]
    success_url = reverse_lazy("menu")
```

4. Teste no postman

4.1. GET

HTTP <http://127.0.0.1:8000/gestao> Save Share </>

GET <http://127.0.0.1:8000/gestao> Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (8) Test Results 200 OK • 39 ms • 5.99 KB

HTML Preview Visualize

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1" />
6
7     <title>Pizzaria - Menu de Pizzas </title>
8
9     <link
10      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css"
11      rel="stylesheet"
12      integrity="sha384-Sg0Ja3DmI69IuzQ2PvdRZhwQ+dy64/ButbMJw1MZ8t5HZApChzRKuc4W0kG879m7"
13      crossorigin="anonymous"
14    />
15   </head>
16   <body>
17     <nav class="navbar bg-dark mb-4" data-bs-theme="dark">
```

4.2. POST

HTTP <http://127.0.0.1:8000/create> Save Share </>

POST <http://127.0.0.1:8000/create> Send

Params Authorization Headers (13) Body Scripts Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

Beautifuly

```
1 {
2   "nome": "Baiana",
3   "descricao": "Pizza com molho de tomate, pimentao, pimenta e cebola roxa",
4   "tamanho": "Grande",
5   "preco": "45,00"
6 }
```

Body Cookies (1) Headers (10) Test Results 200 OK • 38 ms • 3.35 KB

HTML Preview Visualize

```
1 <!-- prettier-ignore -->
2 <!DOCTYPE html>
3 <html lang="pt-br">
4
5   <head>
6     <meta charset="utf-8" />
7     <meta name="viewport" content="width=device-width, initial-scale=1" />
8
9     <title>Pizzaria -
10
11     Nova Pizza
```

4.3. PUT

HTTP client interface showing a PUT request to `http://127.0.0.1:8000/update/1`. The request is successful (200 OK) with a response time of 57 ms and a body size of 2.9 KB.

Headers (13):

Key	Value	Description
Content-Type	application/json	
Cookie	csrftoken=I8FVEcagDr98tGGV8Bg9AYIT2IK1...	
X-CSRFToken	I8FVEcagDr98tGGV8Bg9AYIT2IK1WGP8	

Body:

```
1 <!-- prettier-ignore -->
2 <!DOCTYPE html>
3 <html lang="pt-br">
4
5 <head>
6   <meta charset="utf-8" />
7   <meta name="viewport" content="width=device-width, initial-scale=1" />
8
9   <title>Pizzaria -
10
11     Editar Pizza
```

4.4. DELETE

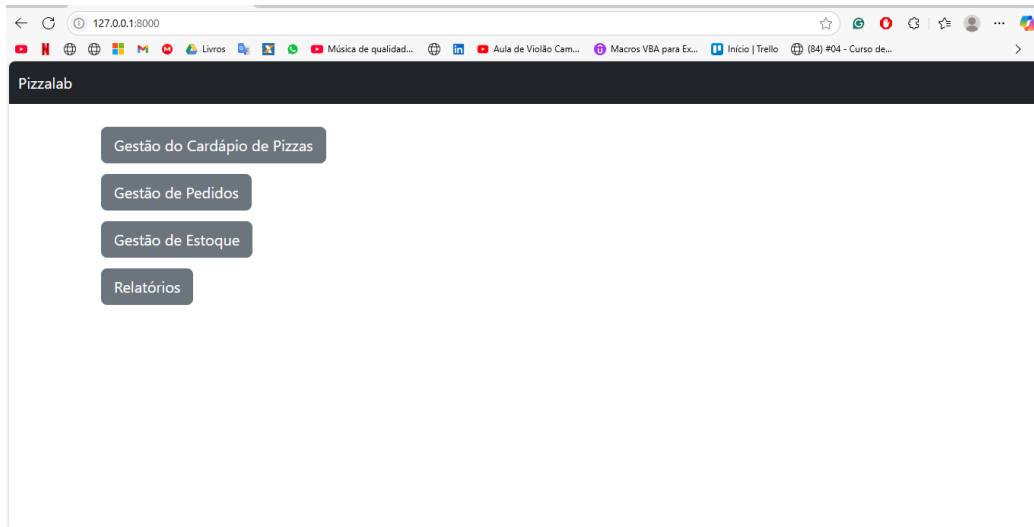
HTTP client interface showing a DELETE request to `http://127.0.0.1:8000/delete/10`. The request is successful (200 OK) with a response time of 17 ms and a body size of 5.18 KB.

Body:

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3
4 <head>
5   <meta charset="utf-8" />
6   <meta name="viewport" content="width=device-width, initial-scale=1" />
7
8   <title>Pizzaria - Menu de Pizzas </title>
9
10   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css" rel="stylesheet"
11     integrity="sha384-Sg0Ja3DmI69IuZQ2PVdRZhWQ+dy64/ButbMJw1M28t6HZApCHzRKUC4W0kG879m7"
```


5. Teste de alteração de banco de dados

Tela inicial:



Tela de Gestão de dados das pizzas

Pizzalab

[Voltar para Início](#)

Gestão do Cardápio de Pizzas

[Adicionar nova Pizza](#)

#	Nome da Pizza	Descrição	Tamanho	Preço	Tipo		
1	Margherita	Pizza clássica com molho de tomate, mussarela, tomate salada e manjerição	Média	30,00	Tradicional	Editar	Excluir
2	Calabresa	Pizza com calabresa, cebola e azeitonas	Grande	40,00	Tradicional	Editar	Excluir
3	Portuguesa	Pizza com presunto, queijo, ovos, cebola e azeitonas	Média	35,00	Tradicional	Editar	Excluir
4	Pepperoni	Pizza com fatias de pepperoni, queijo e molho especial	Grande	45,00	Especial	Editar	Excluir
5	Quatro Queijos	Pizza com mussarela, gorgonzola, parmesão e provolone	Média	38,00	Especial	Editar	Excluir
6	Frango com Catupiry	Pizza com frango desfiado, catupiry e milho	Grande	42,00	Especial	Editar	Excluir
7	Vegetariana	Pizza com vegetais como abobrinha, berinjela, pimentão e cebola	Média	30,00	Vegetariana	Editar	Excluir
8	Napolitana	Pizza com molho de tomate, anchovas, alcaparras e azeit...	Grande	38,00	Tradicional	Editar	Excluir

Para editar uma pizza existente deve-se clicar em “Editar”

Pizzalab

Editar Pizza

Nome*

Margherita

Descricao

Pizza clássica com molho de tomate, mussarela, tomate salada e manjeriçao

Tamanho

Média

Preco

30,00

Salvar

Alterado preço para R\$35,00 e informação refletida no banco de dados:

MySQL Workbench

Localhost x

File Edit View Query Database Server Tools Scripting Help

Navigator

Query 1 auth_user SQL File 4 x

1 • select * from pizza where id_pizza = 1

Limit to 1000 rows

Result Grid

Filter Rows

id_pizza nome descricao tamanho preco tipo

1 Margherita Pizza clássica com molho de tomate, mussarela, ... Média 35.00 Tradicional

Columns

id_pizza nome descricao tamanho preco tipo

Table: pizza

Columns:

id_pizza int(11) PK

nome varchar(60)

descricao varchar(150)

tamanho varchar(30)

preco decimal(5,2)

tipo varchar(30)

Object Info Session

Query Completed

Criando nova Pizza ao acessar a opção “Adicionar nova Pizza”:

Pizzalab

Nova Pizza

Nome*

Nordestina

Descricao

Pizza especial com carne de sol, nata e queijo qualho

Tamanho

Grande

Preco

48,00

Tipo

Especial

Salvar

Informação adicionada no banco de dados:

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane lists various databases, with 'pizza' selected. Below it, the 'Columns' pane shows the structure of the 'pizza' table. The main area displays a query: `select * from pizza where id_pizza = 13`. The 'Result Grid' shows one row of data for the pizza with id 13.

id_pizza	nome	descricao	tamanho	preco	tipo
13	Nordestina	Pizza especial com carne de sol, nata e queijo q...	Grande	48,00	Especial

Table: pizza
Columns:
id_pizza int AI PK
nome varchar(60)
descricao varchar(150)
tamanho varchar(30)
preco decimal(5,2)
tipo varchar(30)

Excluindo a pizza do cardápio:

Pizzalab

Excluir Pizza?

Tem certeza que deseja excluir a Pizza **Havaiana** do cardápio?

ExcluirCancelar

Ao confirmar em “Excluir” a informação foi retirada do banco de dados:

MySQL Workbench

Localhost x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- django_admin_log
- django_content_type
- django_migrations
- django_session
- entrega
- funcionario
- ingrediente
- item_pedido
- log_pizza
- pagamento
- pedido
- pizza**

Columns

- id_pizza
- nome
- descricao
- tamanho
- preco
- tipo

Administration

Schemas

Information

Table: **pizza**

Columns:

- id_pizza int(11) PK
- nome varchar(60)
- descricao varchar(150)
- tamanho varchar(20)
- preco decimal(5,2)
- tipo varchar(30)

Query 1

auth_user SQL File 4" x

Limit to 1000 rows

1 • select * from pizza

Result Grid

	id_pizza	nome	descricao	tamanho	preco	tipo
1	1	Margherita	Pizza clássica com molho de tomate, mussarela, ...	Média	35.00	Tradicional
2	2	Calabresa	Pizza com calabresa, cebola e azeitonas	Grande	40.00	Tradicional
3	3	Portuguesa	Pizza com presunto, queijo, ovos, cebola e azei...	Média	35.00	Tradicional
4	4	Pepperoni	Pizza com fatias de pepperoni, queijo e molho e...	Grande	45.00	Especial
5	5	Quatro Queijos	Pizza com mussarela, gorgonzola, parmesão e p...	Média	38.00	Especial
6	6	Frango com Catupiry	Pizza com frango desfiado, catupiry e molho	Grande	42.00	Especial
7	7	Vegetariana	Pizza com vegetais como abobrinha, berinjela, p...	Média	30.00	Vegetariana
8	8	Napolitana	Pizza com molho de tomate, anchovas, alcaparr...	Pequena	28.00	Tradicional
13	13	Nordestina	Pizza especial com carne de sol, nata e queijo q...	Grande	48.00	Especial

Apply Revert