

📁 Estrutura Atualizada

meu_projeto_fastapi/

```
meu_projeto_fastapi/  
├── .gitignore  
├── main.py  
├── README.md  
├── requirements.txt  
└── venv/
```

📁 .gitignore

```
venv/  
__pycache__/  
*.pyc
```

📄 README.md

```
# Projeto FastAPI com MySQL
```

Este projeto utiliza FastAPI, Uvicorn e MySQL como banco de dados.

📄 main.py (com BaseModel + conexão MySQL + rotas básicas)

```
from fastapi import FastAPI, HTTPException  
from pydantic import BaseModel  
import mysql.connector  
from mysql.connector import Error  
  
app = FastAPI()  
  
# Modelo para requisições POST/PUT  
class Usuario(BaseModel):  
    nome: str  
    email: str  
  
# Função para conectar ao banco  
def conectar_banco():  
    try:  
        conexao = mysql.connector.connect(  

```

```

        host="localhost",
        user="root",
        password="sua_senha",    # <- Troque por sua senha real
        database="meubanco"      # <- Troque pelo seu banco
    )
    return conexao
except Error as e:
    print("Erro ao conectar:", e)
    return None

@app.get("/")
def home():
    return {"mensagem": "Olá, mundo com FastAPI + MySQL"}

@app.post("/usuarios")
def criar_usuario(usuario: Usuario):
    conexao = conectar_banco()
    if conexao is None:
        raise HTTPException(status_code=500, detail="Erro na conexão com o banco de dados")

    try:
        cursor = conexao.cursor()
        query = "INSERT INTO usuarios (nome, email) VALUES (%s, %s)"
        cursor.execute(query, (usuario.nome, usuario.email))
        conexao.commit()
        return {"mensagem": "Usuário criado com sucesso!"}
    except Error as e:
        raise HTTPException(status_code=400, detail=f"Erro ao inserir no banco: {e}")
    finally:
        cursor.close()
        conexao.close()

@app.put("/usuarios/{id}")
def atualizar_usuario(id: int, usuario: Usuario):
    conexao = conectar_banco()
    if conexao is None:
        raise HTTPException(status_code=500, detail="Erro na conexão com o banco de dados")
    try:
        cursor = conexao.cursor()
        query = "UPDATE usuarios SET nome=%s, email=%s WHERE id=%s"
        cursor.execute(query, (usuario.nome, usuario.email, id))
        conexao.commit()
        return {"mensagem": "Usuário atualizado com sucesso!"}
    except Error as e:
        raise HTTPException(status_code=400, detail=f"Erro ao atualizar: {e}")

```

```

        finally:
            cursor.close()
            conexao.close()

@app.delete("/usuarios/{id}")
def deletar_usuario(id: int):
    conexao = conectar_banco()
    if conexao is None:
        raise HTTPException(status_code=500, detail="Erro na conexão com o banco de dados")
    try:
        cursor = conexao.cursor()
        query = "DELETE FROM usuarios WHERE id=%s"
        cursor.execute(query, (id,))
        conexao.commit()
        return {"mensagem": "Usuário deletado com sucesso!"}
    except Error as e:
        raise HTTPException(status_code=400, detail=f"Erro ao deletar: {e}")
    finally:
        cursor.close()
        conexao.close()

```

Testando com o Postman

- **POST** <http://localhost:8000/usuarios>

```

{
  "nome": "Maria",
  "email": "maria@teste.com"
}

```

PUT <http://localhost:8000/usuarios/1>

```

{
  "nome": "Maria Atualizada",
  "email": "nova@teste.com"
}

```

DELETE <http://localhost:8000/usuarios/1>