Programa Jovem Programador

Programador de Sistemas Projeto integrador

Professor: André Ulisses da Silva E-mail: andre.ulisses@hotmail.com



TECNOLOGIA







PROJETO - Requisitos



Descrição Geral do Projeto

Os alunos irão desenvolver, em duplas ou trios, um protótipo funcional (em Portugol Studio/Web) de um sistema que registre e gerencie informações de imigrantes.

Além disso, criarão:

- O modelo de banco de dados (MER + modelo lógico)
- Um plano de testes com execução e registros

Funcionalidades Mínimas do Software

1. Cadastro do Imigrante

- Nome completo
- Nacionalidade
- Data de nascimento
- Documento de identificação

2. Informações complementares

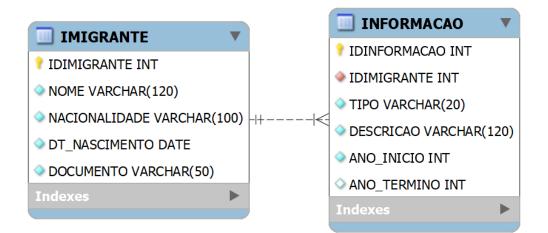
- Escolaridade (nível e país onde estudou)
- Formação profissional (cursos ou profissões)
- Empregos (empresa, cargo, cidade)

3. Serviços de apoio

- Onde tirar documentos (lista de locais)
- Onde buscar ajuda (ONGs, centros de acolhimento, cursos etc.)

PROJETO - DER







PROJETO - SQL



```
DROP DATABASE IF EXISTS DBPROJETO INTEGRADOR;
CREATE DATABASE DBPROJETO INTEGRADOR;
USE DBPROJETO INTEGRADOR;
CREATE TABLE IMIGRANTE (
 IDIMIGRANTE INT NOT NULL AUTO INCREMENT
 , NOME VARCHAR(120) NOT NULL
 , NACIONALIDADE VARCHAR(100) NOT NULL
 , DT NASCIMENTO DATE NOT NULL
 , DOCUMENTO VARCHAR(50)
 , PRIMARY KEY (IDIMIGRANTE)
CREATE TABLE INFORMACAO (
 IDINFORMACAO INT NOT NULL AUTO INCREMENT
 , IDIMIGRANTE INT NOT NULL
 , TIPO VARCHAR(20) NOT NULL
 , DESCRICAO VARCHAR(120) NOT NULL
 , ANO INICIO INT NOT NULL
 , ANO TERMINO INT
 , PRIMARY KEY (IDINFORMACAO)
 , FOREIGN KEY (IDIMIGRANTE) REFERENCES
IMIGRANTE(IDIMIGRANTE)
```

```
CREATE TABLE SERVICO_APOIO (
IDSERVICO_APOIO INT NOT NULL AUTO_INCREMENT
, TIPO VARCHAR(50) NOT NULL
, NOME VARCHAR(120) NOT NULL
, ENDERECO VARCHAR(120) NOT NULL
, TELEFONE VARCHAR(15) NOT NULL
, PRIMARY KEY (IDSERVICO_APOIO)
).
```

PYTHON – Conexão



Para conexão com o banco de dados é necessário fazer instalação do pacote de conexão com o mysql

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Downward Powershell + V II iii ··· ^ X

PS C:\Users\andre\Google Drive\SENAC\SENAC 2025-1\Jovem Programador - Implementar a Banco de Dados \Projeto Integrador\python> pip install mysql-connector-python

PYTHON – Estrutura de arquivos





main.py

Arquivo principal do projeto a partir deste será executado o projeto



imigranteTela.py

Constrói o menu com as opções de imigrante



imigrante.py

Arquivo as opções de CRUD para imigrante



conn.py

Arquivo com a conexão com o banco de dados



informacaolmigranteTela.py

Constrói o menu com as opções de Informações



requeriments.txt

Arquivo com os pacotes necessários o projeto funcionar



informacaoimigrante.py

Arquivo as opções de CRUD para Informações



script.sql

Arquivo com os comandos SQL para criar o banco de dados do projeto



servicoApoioTela.py

Constrói o menu com as opções de serviços



servicoApoio.py

Arquivo as opções de CRUD para serviços

PYTHON – main.py





PYTHON – main.py



```
from imigranteTela import MenuImigrante
from servicoApoioTela import MenuServicoApoio
import os
# metado para limpar o console
def LimparTela():
    os.system('cls' if os.name == 'nt' else 'clear')
def Menu():
    while True:
        # Limpando a tabela para mostrar novamente o menu
        LimparTela()
       print("=== Menu Principal ===")
        print("1 - Imigrante")
        print("2 - Serviço de Apoio")
       print("9 - Sair")
       opcao = input("Escolha uma opção: ")
        if opcao == '1':
           MenuImigrante()
        elif opcao == '2':
           MenuServicoApoio()
        elif opcao == '9':
            break
        else:
           print("Opção inválida.")
if __name__ == "__main__":
    Menu()
```

PYTHON – conn.py



```
import mysql.connector

def conectar():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        password="123456",
        database="DBPROJETO_INTEGRADOR"
    )
```



```
from imigrante import NovoImigrante, ListarTodosImigrantes, AtualizarImigrante, DetalheImigrante, ExcluirImigrante
import os

#metodo para limpar o console

def LimparTela():
    os.system('cls' if os.name == 'nt' else 'clear')
```

ExcluirImigrante(id)

```
# Menu principal com as opções de imigrante
def MenuImigrante():
    while True:
        # Limpando a tela
        LimparTela()
        # Criando o menu de opções
        print("=== Menu Imigrante ===")
        print("1 - Cadastrar Imigrante")
        print("2 - Listar Clientes")
        print("3 - Ver detalhes do Imigrante")
        print("4 - Atualizar Imigrante")
        print("5 - Deletar Imigrante")
        print("9 - Sair")

        opcao = input("Escolha uma opção: ")
```

```
elif opcao == '9':
| break

else:
| print("Opção inválida.")
```

```
if opcao == '1':
           # chamando o metodo para criar novo imigrante
           NovoImigrante()
 elif opcao == '2':
     # listando todos os imigrantes
     ListarTodosImigrantes()
     print("")
     input("pressione enter para voltar para o menu")
elif opcao == '3':
    # listando todos os imigrantes para escolher um e ver o detalhe do imigrante
    ListarTodosImigrantes()
    print("")
    id = input("Informe o ID do imigrante para ver detalhes: ")
    # chamando metodos para mostrar os dados do imigrante selecionado
    DetalheImigrante(id)
 elif opcao == '4':
     # listando todos os imigrantes para escolher o que vai ser atualizado
     ListarTodosImigrantes()
     print("")
     id = input("ID do imigrante para atualizar: ")
     print("")
     # chamando metodos para atualizar os dados
     AtualizarImigrante(id)
elif opcao == '5':
    # listando todos os imigrantes para escolher o que vai ser atualizado
    ListarTodosImigrantes()
    print("")
    id = input("ID do imigrante para excluir: ")
    print("")
    # chamando metodos para excluir os dados
```



```
metodo para criar um novo imigrante
def NovoImigrante():
   print("+----
   print("| Cadastrar novo Imigrante
   print("+----
   print("")
   # solicitando dados do imigrante
   nome = input("Nome: ")
   nacionalidade = input("Nacionalidade: ")
   dataNascimento = input("Data de Nascimento: ")
   documento = input("Documento de identificacao: ")
   # Criando conexão com o banco de dados
   conn = conectar()
   cur = conn.cursor()
   # enviando o comando para o banco de dados
   cur.execute("INSERT INTO IMIGRANTE(NOME, NACIONALIDADE, DT NASCIMENTO, DOCUMENTO) VALUES (%s, %s, %s, %s)"
              , (nome, nacionalidade, dataNascimento, documento))
   conn.commit()
   print("")
   print("Imigrante cadastrado com sucesso!")
   input("pressione enter para voltar para o menu")
   # fechando conexão com o banco de dados
   cur.close()
   conn.close()
```



```
# metodo para lista todos os imigrante
def ListarTodosImigrantes():
   print("+----
   print("| Lista de Imigrantes
   print("+----
   print("")
   # Criando conexão com o banco de dados
   conn = conectar()
   cur = conn.cursor()
   # Eviando comando para o banco de dados
   cur.execute("SELECT IDIMIGRANTE, NOME, NACIONALIDADE, DT_NASCIMENTO, DOCUMENTO FROM IMIGRANTE")
   #Percorrento o resultado do comando SQL e listando em tela
   for id, nome, nacionalidade, dataNascimento, documento in cur.fetchall():
      print(f"{id} - {nome} - {nacionalidade} - {dataNascimento} - {documento}")
   # Fechando comando com o banco de dados
   cur.close()
   conn.close()
```



```
# metodo para Mostrar detalhe do imigrante
def DetalheImigrante(id):
   # Criando conexão com o banco de dados
   conn = conectar()
   cur = conn.cursor()
   # Eviando comando para o banco de dados
   cur.execute("SELECT IDIMIGRANTE, NOME, NACIONALIDADE, DT_NASCIMENTO, DOCUMENTO FROM IMIGRANTE WHERE IDIMIGRANTE = %s",
   for id, nome, nacionalidade, dataNascimento, documento in cur.fetchall():
       print(f"| Detalhe do Imigrantes {nome}")
       print("+----
       print(f"id: {id}")
       print(f"Nome: {nome}")
       print(f"Data de Nacimento: {dataNascimento}")
       print(f"Nacionalidade: {nacionalidade}")
       print(f"Documento: {documento}")
       print("")
   print("")
   input("pressione enter para voltar para o menu")
   # Fechando comando com o banco de dados
   cur.close()
   conn.close()
```



```
def AtualizarImigrante(id):
    # Criando conexão com o banco de dados
    conn = conectar()
    cur = conn.cursor()
    # solicitando novos dados do imigrante
    print(" Informa os novos dados")
    nome = input("Nome: ")
    nacionalidade = input("Nacionalidade: ")
    dataNascimento = input("Data de Nascimento: ")
    documento = input("Documento de identificacao: ")
    # Eviando comando para o banco de dados
    cur.execute("UPDATE IMIGRANTE SET NOME = %s, NACIONALIDADE = %s, DT NASCIMENTO = %s, DOCUMENTO = %s WHERE IDIMIGRANTE
                , (nome, nacionalidade, dataNascimento, documento, id,))
    conn.commit()
    print("")
    print("Imigrante atualizado com sucesso!")
    input("pressione enter para voltar para o menu")
    # Fechando comando com o banco de dados
    cur.close()
    conn.close()
```



```
def ExcluirImigrante(id):
    # Criando conexão com o banco de dados
    conn = conectar()
    cur = conn.cursor()

# Eviando comando para o banco de dados
    cur.execute("DELETE FROM IMIGRANTE WHERE IDIMIGRANTE = %s", (id,))
    conn.commit()

print("")
print("Imigrante excluido com sucesso!")
input("pressione enter para voltar para o menu")
# Fechando comando com o banco de dados
    cur.close()
    conn.close()
```