Modelo MER

Este Modelo de Entidade-Relacionamento (MER) representa um banco de dados para uma instituição educacional, com tabelas inter-relacionadas para gerenciar informações de professores, alunos, cursos, usuários e cidades.

**Tabela tbl\_professores:** Esta tabela armazena informações sobre os professores. A tabela tbl\_professores relaciona-se com a tabela tbl\_cidades através do campo tbl\_cidades\_cid\_codigo, indicando a cidade de cada professor.

**Tabela tbl\_alunos:** Esta tabela contém informações sobre os alunos. A tabela tbl\_alunos possui relacionamentos com tbl\_cidades para indicar a cidade de origem dos alunos, e com tbl\_cursos para especificar o curso em que cada aluno está matriculado.

**Tabela tbl\_usuarios:** Esta tabela armazena os dados de usuários do sistema, como administradores ou outros funcionários que possam precisar de acesso ao sistema. Não há relacionamentos diretos desta tabela com as outras tabelas visíveis no diagrama, sugerindo que os usuários são independentes das outras entidades do sistema.

**Tabela tbl\_cidades:** Esta tabela contém os dados das cidades, fundamentais para associar localidades aos professores e alunos. A tabela tbl\_cidades está relacionada tanto com a tabela tbl\_professores quanto com a tabela tbl\_alunos, permitindo que ambos professores e alunos sejam associados a uma cidade específica através do campo de chave estrangeira cid\_codigo.

**Tabela tbl\_cursos:** Esta tabela guarda os detalhes dos cursos oferecidos pela instituição. A tabela tbl\_cursos relaciona com a tabela tbl\_alunos através do campo tbl\_cursos\_cur\_codigo, que indica qual curso cada aluno está matriculado.

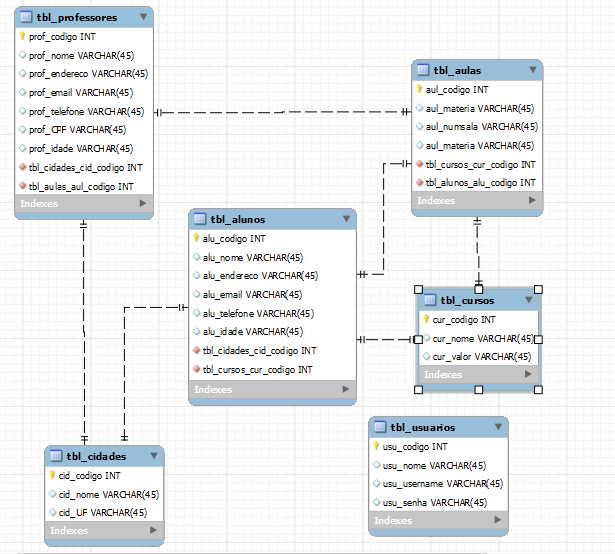
**Tabela tbl\_aulas:** Armazena informações sobre as aulas ministradas. Relaciona-se com tbl\_professores para associar aulas aos professores e com tbl\_alunos para associar aulas aos alunos.

**Professores e Aulas:** Professores estão associados às aulas que lecionam, ligando tbl\_professores a tbl\_aulas.

**Alunos, Aulas e Cursos:** Alunos assistem às aulas específicas de seus cursos, estabelecendo uma conexão entre tbl\_alunos, tbl\_aulas e tbl\_cursos.

**Professores e Alunos com Cidades:** Tanto professores quanto alunos estão associados a cidades específicas através de suas respectivas chaves estrangeiras que referenciam tbl\_cidades.

SEGUE IMAGEM DO MER



SEGUE ATUALIZAÇÕES DO ALGORITMO

**Conexão com o Banco de Dados MySQL:**

Criado código para conectar ao banco de dados MySQL usando mysql-connector-python.

Configuração para verificar a conexão e tratamento de erros de conexão.

**Criação da Interface de Login:**

Desenvolvida uma tela de login com autenticação básica.

Implementada a função para verificar as credenciais dos usuários na tabela tbl\_usuarios.

**Interface Principal com Menus:**

Criada a interface principal com menus para inserção e visualização de dados.

Os menus são: Cidades, Cursos, Professores, Alunos, Aulas, e Usuários.

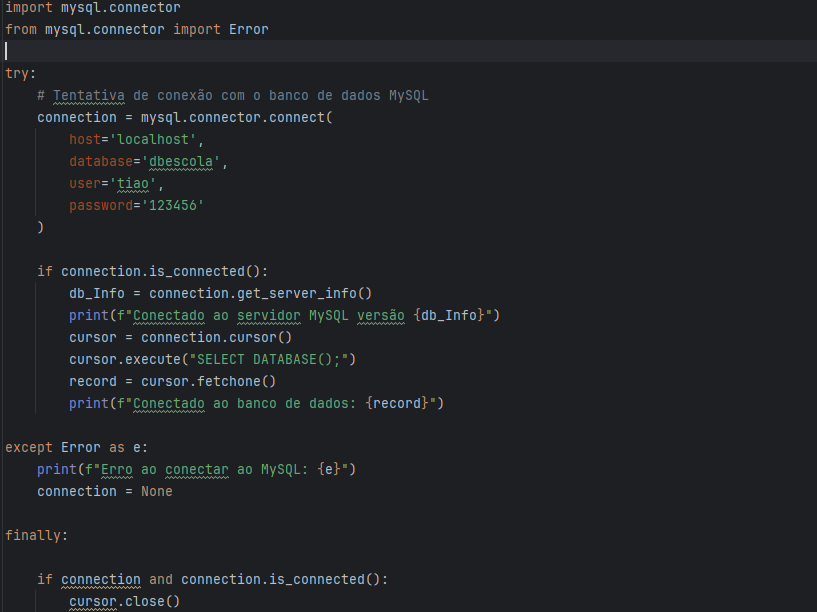
Cada menu tem um botão específico para inserir dados na tabela correspondente.

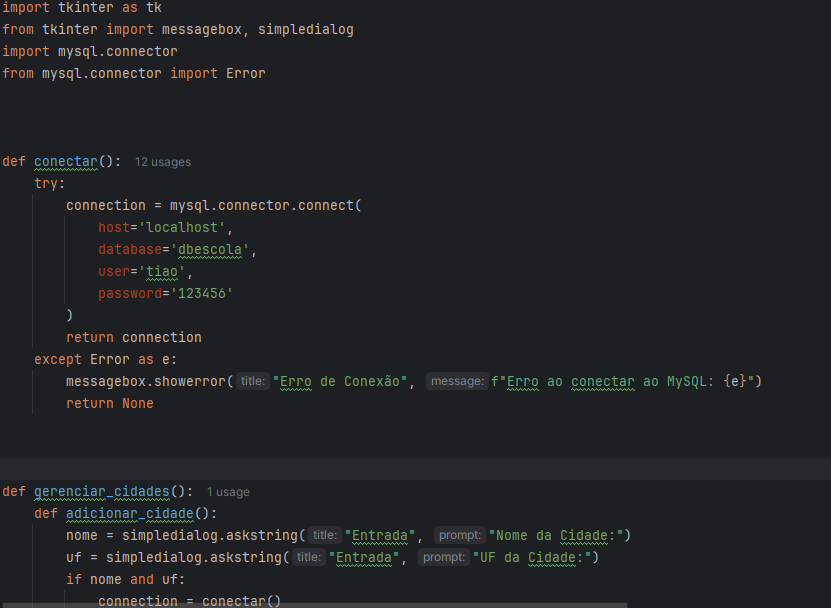
**Inserção de Dados:**

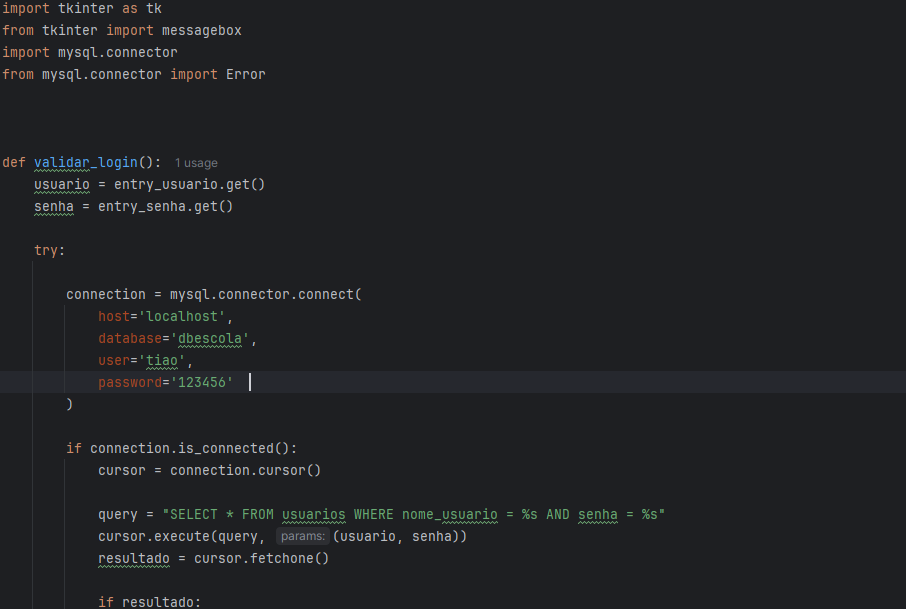
Adicionados botões e formulários para inserir dados em cada tabela do banco:

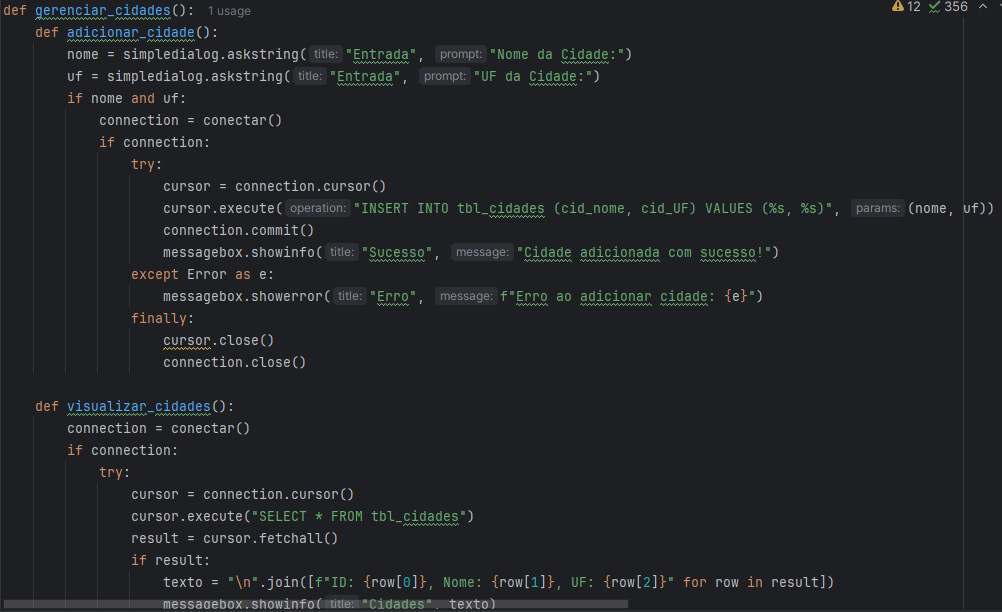
* + tbl\_cidades
  + tbl\_cursos
  + tbl\_professores
  + tbl\_alunos
  + tbl\_aulas
  + tbl\_usuarios

SEGUE ALGUMAS IMAGENS DO CÓDIGO









Relatório do Projeto Atualizado

Nome: João Gabriel Castanheira

Professor: Carmona

Data: 24/03/24

Introdução

Neste projeto, desenvolvi um Sistema de Gestão de Usuários utilizando Python. A proposta foi criar uma ferramenta prática e acessível para adicionar, editar e excluir usuários. Para isso, utilizei a biblioteca tkinter para construir a interface gráfica e sqlite3 para gerenciar o banco de dados.

Estrutura do Código

O código está organizado em algumas classes principais, cada uma com uma função específica:

Banco: Esta classe é responsável por gerenciar a conexão com o banco de dados e criar a tabela de usuários, garantindo que todas as informações sejam armazenadas de maneira estruturada.

Usuarios: Aqui, implementei as funções que manipulam os dados dos usuários, como adicionar, editar e excluir informações.

PaginaPrincipal: É a interface que aparece ao abrir o aplicativo, servindo como ponto de partida para as interações do usuário.

AdicionarUsuario, AlterarUsuario, ExcluirUsuario, GerenciarUsuarios: Essas classes tratam das operações específicas de cada funcionalidade, permitindo uma navegação intuitiva pelo sistema.

Funcionalidades

As principais funcionalidades do sistema incluem:

Adicionar Usuário: Permite o cadastro de novos usuários, preenchendo informações essenciais.

Alterar Usuário: Oferece a possibilidade de editar informações de usuários já cadastrados, garantindo que os dados estejam sempre atualizados.

Excluir Usuário: Facilita a remoção de usuários do sistema, caso necessário.

Gerenciar Usuários: Exibe uma lista de todos os usuários cadastrados, permitindo a seleção e edição de dados conforme necessário.

Detalhes Técnicos

O sistema utiliza SQLite como banco de dados, armazenando as informações dos usuários em uma tabela que inclui campos como ID, nome, telefone, email, usuário, senha e cidade. A interface foi construída com tkinter, que se mostrou bastante intuitiva e fácil de usar, proporcionando uma boa experiência ao usuário.

Segue imagens:

