

**Como Elaborar um Relatório com o Estilo (Formato) APA (American Psychological Association)**

**Rafael Ascenção e Tiago Guerreiro**

Curso Técnico de Gestão e Programação de Sistemas Informáticos

Escola Profissional Bento de Jesus Caraça, Delegação do Barreiro

P.S.I.: Programação e Sistemas de Informação

Coordenador Marcelo Simão

05 de fevereiro de 2025

**Introdução**

O **Sistema de Gestão de Ginásio** é um projeto desenvolvido em **Python** com utilização do **SQLite3** para armazenamento de dados. O sistema permite realizar operações fundamentais em um banco de dados, como **criação, leitura, atualização e remoção de registros** (CRUD).

Este projeto foi elaborado com o objetivo de facilitar a gestão de um ginásio, permitindo o cadastro de membros, planos de assinatura e gestão eficiente das informações.

**Estrutura da Base de Dados**

A estrutura do banco de dados do sistema de gestão de ginásio é composta por duas tabelas principais: **MEMBROS** e **PLANOS**. A relação entre elas é feita por meio da chave estrangeira **PLANO\_ID**, que conecta um membro a um plano específico.

### **Tabela MEMBROS**

Armazena informações sobre os membros cadastrados no ginásio.

|  |  |  |
| --- | --- | --- |
| **Coluna** | **Tipo de Dado** | **Restrições** |
| ID | INTEGER | PRIMARY KEY, AUTOINCREMENT |
| NOME | TEXT | NOT NULL |
| IDADE | INTEGER | NOT NULL |
| PLANO\_ID | INTEGER | FOREIGN KEY REFERENCIANDO PLANOS(ID) |

**Descrição:**

O **ID** é a chave primária e é gerado automaticamente (autoincremento).

O **NOME** do membro é obrigatório e armazenado como **TEXT**.

A **IDADE** também é obrigatória e armazenada como **INTEGER**.

O **PLANO\_ID** referencia a tabela **PLANOS**, associando o membro a um plano específico.

### **Tabela PLANOS**

Armazena os diferentes planos de assinatura disponíveis no ginásio.

|  |  |  |
| --- | --- | --- |
| **Coluna** | **Tipo de Dado** | **Restrições** |
| ID | INTEGER | PRIMARY KEY, AUTOINCREMENT |
| NOME\_PLANO | TEXT | NOT NULL |
| PRECO | INTEGER | NOT NULL |

**Descrição:**

O **ID** é a chave primária e gerado automaticamente.

O **NOME\_PLANO** armazena o nome do plano (exemplo: "Plano Básico", "Plano Intermediário", "Plano Premium").

O **PRECO** define o valor do plano em **INTEGER**.

### **Relacionamento entre as Tabelas**

A chave estrangeira **PLANO\_ID** na tabela **MEMBROS** referencia a chave primária **ID** da tabela **PLANOS**. Isso significa que cada membro pode estar associado a um único plano, garantindo integridade referencial.

# **Sistema de Gestão de Ginásio**

## **Introdução**

O **Sistema de Gestão de Ginásio** é um projeto desenvolvido em **Python** com utilização do **SQLite3** para armazenamento de dados. O sistema permite realizar operações fundamentais em um banco de dados, como **criação, leitura, atualização e remoção de registros** (CRUD). Este projeto foi elaborado para facilitar a gestão de um ginásio, permitindo o cadastro de membros, planos de assinatura e gerenciamento eficiente das informações.

## **Estrutura do Projeto**

O sistema é composto por diferentes módulos que executam tarefas específicas:

### **Estrutura dos Arquivos**

**app.py**: Arquivo principal que executa o sistema e exibe o menu interativo.

**create.py**: Responsável por criar novos registros no banco de dados.

**read.py**: Permite visualizar registros existentes.

**update.py**: Atualiza e adiciona novos registros.

**delete.py**: Exclui registros do banco de dados.

**create\_membros.py**: Cria a tabela de membros e estrutura os dados associados.

**create\_planos.py**: Define a estrutura dos planos de assinatura e insere opções iniciais no banco de dados.

## **Procedimentos de Uso**

### **Requisitos**

Python 3.x

SQLite3

### **Etapas de Execução**

Clone ou baixe o repositório:

git clone <https://github.com/tiagoyxz/projeto-final.git>

Navegue até a pasta do projeto:

cd projeto-final/SRC/APP

Execute o arquivo principal:

python app.py

Escolha uma das opções do menu:

Criar novos registros

Ler registros existentes

Atualizar registros

Excluir registros

Sair

## **Funcionalidades**

O sistema permite realizar as seguintes operações:

**Gerenciamento de Membros**: Cadastro, consulta, atualização e remoção de membros.

**Gestão de Planos**: Criação e manutenção de diferentes planos de assinatura.

**Banco de Dados Estruturado**: O banco de dados é armazenado localmente via SQLite3, garantindo a persistência dos dados.

## **Estrutura do Banco de Dados**

O banco de dados SQLite armazena informações em duas tabelas principais:

### **Tabela Membros**

**ID**: Identificador único (chave primária, autoincrementado).

**Nome**: Nome do membro.

**Idade**: Idade do membro.

**Plano\_ID**: Identificador do plano associado (chave estrangeira referenciando a tabela Planos).

### **Tabela Planos**

**ID**: Identificador único (chave primária, autoincrementado).

**Nome do Plano**: Nome do plano (exemplo: "Plano Básico").

**Preço**: Valor do plano.

## **Tecnologias Utilizadas**

**Python 3.x**

**SQLite3** para manipulação do banco de dados

**Biblioteca OS** para interação com o terminal

## **Considerações Finais**

O sistema de gestão de ginásio apresenta uma solução eficiente para a organização de membros e planos de assinatura. Com sua estrutura modular, facilita a expansão e aprimoramento conforme novas necessidades surgirem. A utilização do SQLite3 garante um armazenamento seguro e acessível dos dados.

### **Referências**

American Psychological Association. (2020). *Publication manual of the American Psychological Association* (7th ed.). APA Publishing.