**Instruções Gerais:**

* Esta prova é prática e deverá ser executada localmente com apoio do XAMPP (MariaDB/MySQL) e Python.
* Utilize apenas os recursos ensinados em aula: print, input, int, float, if, elif, else, while, f-string.
* HTML e PHP podem ser utilizados para criação das telas de entrada de dados.
* Não utilize bibliotecas externas nem estruturas que não tenham sido aplicadas em sala.

**Conexão do Python com o MariaDB:**

Para que o Python consiga acessar o banco de dados criado no XAMPP, utilize o seguinte código no início do seu script:

import mysql.connector

conexao = mysql.connector.connect(

host="localhost",

user="root",

password="",

database="seu\_nome\_do\_banco"

)

cursor = conexao.cursor()  
  
**PARTE 1 — MARIA DB (XAMPP)**

**1. Criação do Banco de Dados (1 ponto)**

Crie um banco de dados no XAMPP com o seu nome (ex: joao\_db).

**2. Criação das Tabelas (2 pontos)**

Crie no banco as seguintes tabelas. Utilize os comandos abaixo como modelo:

CREATE TABLE notas (

id INT AUTO\_INCREMENT PRIMARY KEY,

aluno VARCHAR(100),

materia VARCHAR(100),

nota1 FLOAT,

nota2 FLOAT,

faltas INT

);

CREATE TABLE registro\_json (

id INT AUTO\_INCREMENT PRIMARY KEY,

aluno VARCHAR(100),

dados JSON

);

**PARTE 2 — HTML + PHP**

**3. Tela de Cadastro e Consulta (HTML) + Inserção e Visualização (PHP) (2 pontos)**

Crie uma tela em HTML chamada cadastro.html com os campos:

* Nome do aluno
* Nome da matéria
* Nota 1
* Nota 2
* Número de faltas

Crie também o arquivo salvar.php para receber os dados via POST e inseri-los na tabela notas.

Além disso, desenvolva uma segunda tela chamada consulta.html que permita inserir o nome de um aluno para consulta. Essa tela deverá enviar os dados para um script consultar.php que:

* Consulta os dados da tabela notas
* Calcula a média
* Exibe o número de faltas
* Mostra a situação (Aprovado, Recuperação ou Reprovado)

⚠️ **Importante:** Cada aluno deve ser cadastrado em **4 matérias diferentes**, com 2 notas por matéria. Isso será validado no script Python.

**PARTE 3 — PYTHON (Consulta e Manipulação)**

**4. Consulta e Manipulação em Python (3 pontos)**

Desenvolva também uma tela HTML chamada consulta.html e um script PHP (consultar.php) que permita exibir, para cada aluno:

* Nome
* Matéria
* Nota 1
* Nota 2
* Média calculada
* Número de faltas
* Situação (Aprovado, Recuperação ou Reprovado)

O cálculo da média e da situação pode ser realizado diretamente no PHP ou utilizando os dados previamente manipulados e salvos pelo Python.

Além disso: Desenvolva um script Python que:

* Liste todos os alunos cadastrados no banco de dados.
* Apresente uma tela de menu simples no terminal, como por exemplo:

print("Alunos cadastrados:")

print("[0] João")

print("[1] Maria")

print("[2] Pedro")

escolha = int(input("Escolha o número correspondente ao aluno: "))

Com base na escolha do número, selecione o nome do aluno correspondente.

Verifique se o aluno possui ao menos 4 matérias diferentes.

Para cada matéria, recupere as notas e solicite o número de faltas digitado pelo usuário.

Calcule a média e classifique o desempenho da seguinte forma:

Aprovado (média = 7)

Recuperação (5 = média < 7)

Reprovado (média < 5)

Exiba os resultados de forma clara no terminal:  
  
Exemplo de saída no terminal:

Aluno: João

Matéria: Matemática | Média: 7.5 | Situação: Aprovado

Matéria: História | Média: 5.5 | Faltas: 4 | Situação: Recuperação

Matéria: Geografia | Média: 6.0 | Faltas: 7 | Situação: Reprovado por faltas

❗ Caso o aluno tenha menos de 4 matérias cadastradas, o script deve exibir uma mensagem de aviso e encerrar a execução.

⚠️ Regras de classificação:

Aprovado: média ≥ 7 e faltas ≤ 6

Recuperação: 5 ≤ média < 7 e faltas ≤ 6

Reprovado: média < 5 ou faltas > 6  
  
**PARTE 4 — PYTHON + MYSQL (Simulação NoSQL)**

**5. Simulação de NoSQL com JSON no MySQL (2 pontos)**

Utilizando apenas f-string ou concatenação, monte uma string com estrutura JSON contendo os dados do aluno e das notas cadastradas (como se fosse um documento NoSQL).

Exemplo de como montar a string manualmente:

json\_simulado = f'{{"materias": {{"Matematica": {{"nota1": 7.0, "nota2": 8.5}}, "Historia": {{"nota1": 6.0, "nota2": 6.0}}, "Geografia": {{"nota1": 5.0, "nota2": 4.5}}, "Ciencias": {{"nota1": 9.0, "nota2": 8.5}} }} }}'  
  
Exemplo de estrutura esperada:

{

"materias": {

"Matematica": {"nota1": 7.0, "nota2": 8.5},

"Historia": {"nota1": 6.0, "nota2": 6.0},

"Geografia": {"nota1": 5.0, "nota2": 4.5},

"Ciencias": {"nota1": 9.0, "nota2": 8.5}

}

}  
  
  
  
Auxílio.

Modelo de tabelas (MySQL/MariaDB)  
CREATE TABLE notas (

id INT AUTO\_INCREMENT PRIMARY KEY,

aluno VARCHAR(100),

materia VARCHAR(100),

nota1 FLOAT,

nota2 FLOAT

);  
  
Exemplo do trecho principal em Python  
import mysql.connector

conexao = mysql.connector.connect(

host="localhost",

user="root",

password="",

database="avaliacao"

)

cursor = conexao.cursor()

# Lista todos os alunos e matérias cadastradas

cursor.execute("SELECT DISTINCT aluno, materia FROM notas")

registros = cursor.fetchall()

print("Alunos e matérias disponíveis:")

for i, (aluno, materia) in enumerate(registros):

print(f"[{i}] {aluno} - {materia}")

escolha = int(input("Escolha o número do aluno/matéria que deseja consultar: "))

aluno\_escolhido, materia\_escolhida = registros[escolha]

# Busca as notas para aquele aluno/matéria

cursor.execute("SELECT nota1, nota2 FROM notas WHERE aluno = %s AND materia = %s", (aluno\_escolhido, materia\_escolhida))

notas = cursor.fetchall()

# Calcula média

for nota1, nota2 in notas:

media = (nota1 + nota2) / 2

if media >= 7:

status = "Aprovado"

elif media >= 5:

status = "Recuperação"

else:

status = "Reprovado"

print(f"\nAluno: {aluno\_escolhido}")

print(f"Matéria: {materia\_escolhida}")

print(f"Nota 1: {nota1} | Nota 2: {nota2}")

print(f"Média: {media:.2f} | Situação: {status}")

Exemplo de validação em Python:  
# Verifica quantas matérias o aluno tem

cursor.execute("SELECT DISTINCT materia FROM notas WHERE aluno = %s", (aluno\_escolhido,))

materias = cursor.fetchall()

if len(materias) < 4:

print(f"\nO aluno {aluno\_escolhido} ainda não possui 4 matérias cadastradas.")

print("Cadastre as matérias faltantes antes de continuar.")

else:

print(f"\nO aluno {aluno\_escolhido} possui {len(materias)} matérias cadastradas. Calculando médias...\n")

for materia in materias:

cursor.execute("SELECT nota1, nota2 FROM notas WHERE aluno = %s AND materia = %s", (aluno\_escolhido, materia[0]))

nota1, nota2 = cursor.fetchone()

media = (nota1 + nota2) / 2

if media >= 7:

status = "Aprovado"

elif media >= 5:

status = "Recuperação"

else:

status = "Reprovado"

print(f"Matéria: {materia[0]} | Média: {media:.2f} | Situação: {status}")  
  
<form action="salvar.php" method="post">

Nome: <input type="text" name="aluno"><br>

Matéria: <input type="text" name="materia"><br>

Nota 1: <input type="text" name="nota1"><br>

Nota 2: <input type="text" name="nota2"><br>

<input type="submit" value="Cadastrar">

</form>  
  
<?php

$conn = new mysqli("localhost", "root", "", "avaliacao");

$aluno = $\_POST['aluno'];

$materia = $\_POST['materia'];

$nota1 = $\_POST['nota1'];

$nota2 = $\_POST['nota2'];

$sql = "INSERT INTO notas (aluno, materia, nota1, nota2)

VALUES ('$aluno', '$materia', $nota1, $nota2)";

$conn->query($sql);

echo "Cadastro realizado com sucesso!";

?>  
  
CREATE DATABASE IF NOT EXISTS avaliacao;

USE avaliacao;

CREATE TABLE IF NOT EXISTS notas (

id INT AUTO\_INCREMENT PRIMARY KEY,

aluno VARCHAR(100),

materia VARCHAR(100),

nota1 FLOAT,

nota2 FLOAT

);  
  
  
  
CREATE TABLE IF NOT EXISTS registro\_json (

id INT AUTO\_INCREMENT PRIMARY KEY,

aluno VARCHAR(100),

dados JSON

);