| | |
|---|---|
| funec RIACHO | Projeto integrador multidisciplinar. |
| Prof. Rubens Palhares | Curso: Ensino médio integrado com Técnico em Informática |

# Projeto Integrado: Sistema de Gestão de Aluno

Objetivo Geral:

Desenvolver um sistema completo de cadastro, consulta e análise de desempenho de alunos utilizando:

Python para lógica e testes iniciais

MySQL como banco de dados

PHP + HTML/CSS para interface web

---

**Etapas do Projeto e Distribuição por Grupos**

1 Desenvolvimento Inicial em Python

Status: Já realizado

**O que foi feito:**

Cadastro de alunos (nome e nota)

Validação de dados (nome não vazio, nota entre 0 e 10)

Listagem, busca e estatísticas (média e aprovados)

---

## Grupo Responsável: Grupo 1

***Próximo passo: Migrar a lógica para o banco de dados MySQL***

2 Migração para MySQL

Plataforma: ***Visual Studio Code*** com extensão ***MySQL***

**Tarefas:**

1. Abrir novo terminal no VS Code

2. Conectar ao MySQL usando: `***mysql -u usuario -p***`

3. Criar o banco de dados: `CREATE DATABASE gestao_alunos;`

4. Criar a tabela alunos com:

**SQL**

```
CREATE TABLE alunos (
   id INT AUTO_INCREMENT PRIMARY KEY,
   nome VARCHAR(100) NOT NULL,
   nota FLOAT NOT NULL CHECK (nota >= 0 AND nota <= 10)
);
```

5. Testar inserções: `INSERT INTO alunos (nome, nota) VALUES ('João', 7.5);`

6. Testar consultas: `SELECT * FROM alunos;`

Saída Esperada:  Banco de dados pronto para integração

---

## Grupo Responsável: Grupo 2

**3 Integração Python + MySQL**

Plataforma: Visual Studio Code

Tarefas:

1. Criar novo arquivo Python no VS Code

2. Instalar conector: `pip install mysql-connector-python`

3. Implementar conexão:

PYTHON

```
import mysql.connector
def conectar():
   return mysql.connector.connect(
      host="localhost",
      user="usuario",
```

```
    password="senha",
    database="gestao_alunos"
)
```

4. Adaptar funções para usar MySQL:

PYTHON

```
def cadastrar_aluno(nome, nota):
    conexao = conectar()
    cursor = conexao.cursor()
    cursor.execute("INSERT INTO alunos (nome, nota) VALUES (%s, %s)", (nome, nota))
    conexao.commit()
    conexao.close()
```

Saída Esperada: Script Python funcionando com MySQL

---

## Grupo Responsável: Grupo 3

4 Consultas e Estatísticas via SQL

Plataforma: Visual Studio Code

**Tarefas:**

1. Criar novo arquivo Python para consultas

2. Implementar funções de relatório:

PYTHON

```
def media_geral():
    conexao = conectar()
    cursor = conexao.cursor()
    cursor.execute("SELECT AVG(nota) FROM alunos")
    return cursor.fetchone()[0]
def listar_aprovados():
    conexao = conectar()
```

```
    cursor = conexao.cursor()

    cursor.execute("SELECT nome, nota FROM alunos WHERE nota >= 6")

    return cursor.fetchall()
```

3. Testar exibição dos resultados

Saída Esperada: Relatórios funcionais do banco

---

**Grupo Responsável: Grupo 4**

5 Interface Web HTML + CSS

Plataforma: Visual Studio Code

**Tarefas:**

1. Criar pasta `**web**` no projeto

2. Criar arquivos:

***`index.php` (menu principal)***

***`cadastro.php` (formulário)***

***`listar.php` (lista alunos)***

***`estatisticas.php` (relatórios)***

3. Desenvolver estrutura HTML básica

4. Criar arquivo `style.css` para estilização

Exemplo index.php:

**HTML**

```
<!DOCTYPE html>

<html>

<head>

    <link rel="stylesheet" href="style.css">

</head>

<body>
```

```
    <h1>Gestão de Alunos</h1>
    <nav>
        <a href="cadastro.php">Cadastrar</a>
        <a href="listar.php">Listar</a>
        <a href="estatisticas.php">Estatísticas</a>
    </nav>
</body>
</html>
```

Saída Esperada: Páginas web estáticas estilizadas

---

## Grupo Responsável: Grupo 5

6 Lógica Web em PHP + MySQL

Plataforma: Visual Studio Code

**Tarefas:**

1. Configurar servidor local (Live Server ou XAMPP)

2. Implementar conexão PHP:

PHP

```
<?php
$conexao = new mysqli("localhost", "usuario", "senha", "gestao_alunos");
if ($conexao->connect_error) {
    die("Erro de conexão: " . $conexao->connect_error);
}
?>
```

3. Processar formulário em `cadastro.php`:

**PHP**

```
<?php
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    require "conexao.php";
    $nome = $_POST["nome"];
    $nota = $_POST["nota"];
    $stmt = $conexao->prepare("INSERT INTO alunos (nome, nota) VALUES (?, ?)");
    $stmt->bind_param("sd", $nome, $nota);
    $stmt->execute();
    echo "Aluno cadastrado!";
}
?>
```

Saída Esperada: Sistema web completo e funciona

---

## Conclusão e Integração Final

1. Testar fluxo completo:

- Cadastrar aluno via formulário web
- Verificar inserção no banco
- Visualizar listagem e estatísticas

2. Checklist de entrega:

- Banco de dados populado
- Backend Python funcionando
- Interface web responsiva
- Relatórios operacionais

---

2. Cada grupo foca em sua etapa

3. Reuniões para integração entre grupos sempre verificar o código do grupo anterior.

4. Validar cada etapa antes de avançar

3. Realizar testes integrados

4. Preparar apresentação final