



# **Projeto H – Sistema de Quiz de Múltipla Escolha**

João Geiger Piza

BCC, Turma 73

4º Período

# Descrição Do Projeto



designed by  frespk

- O Sistema serve para criar perguntas de múltipla escolha, montar quizzes com estas, aplicar os quizzes e visualizar os resultados destas aplicações;
- PÚblico que poderia utilizar: instituições educacionais, cursos, tutores, etc

# Demonstração das Funcionalidades

```
--- Menu ---  
Qual a operação desejada?  
(0) Cadastrar Pergunta;  
(1) Montar um Quiz  
(2) Aplicar um Quiz  
(3) Mostrar os resultados  
(4) Sair  
Digite a opção desejada:
```

Menu Principal

```
Digite o enunciado: Quanto é 1 + 1 ?  
Digite a alternativa A: 3  
Digite a alternativa B: 5  
Digite a alternativa C: 7  
Digite a alternativa D: 2  
Digite a alternativa correta: D  
Pergunta salva com id 3  
-----  
Mais alguma coisa?  
(0) Sim  
(1) Não  
Digite a opção desejada: |
```

Criação de uma pergunta  
(Observação: Após qualquer operação, será questionado se o usuário gostaria de realizar mais alguma coisa (Como visto acima))

# Demonstração das Funcionalidades

Digite o título do quiz: Pergunta de matemática

Digite o número de perguntas desejado: 1

0: Vai rodar?

1: Testando, vai funcionar?

2: Mais uma pergunta a ser testada, vai dar certo?

3: Quanto é  $1 + 1$  ?

Digite o id da pergunta 1: 3

Quiz salvo com id 3.

-----

Mais alguma coisa?

(0) Sim

(1) Não

Digite a opção desejada:

Criação de um quiz usando a pergunta criada no slide anterior

0: Vai rodar, 1 perguntas;

1: Teste de mais linhas, 2 perguntas;

2: Ultimo teste, 3 perguntas;

3: Pergunta de matemática, 1 perguntas;

Digite o id do quiz a ser aplicado: 3

Digite o nome de quem vai responder: Jorge

Quanto é  $1 + 1$  ?

(A) 3

(B) 5

(C) 7

(D) 2

Digite a alternativa que julgar correta: D

-----

Resultado salvo com id 4

-----

Mais alguma coisa?

(0) Sim

(1) Não

Digite a opção desejada:

Aplicação do quiz montado

# Demonstração das Funcionalidades

```
4 (quiz: 3):  
Participante: Jorge  
Nota: 1  
Data: 09/02/2026
```

```
Mais alguma coisa?  
(0) Sim  
(1) Não  
Digite a opção desejada:
```

Visualização do Resultado da aplicação realizada no slide anterior

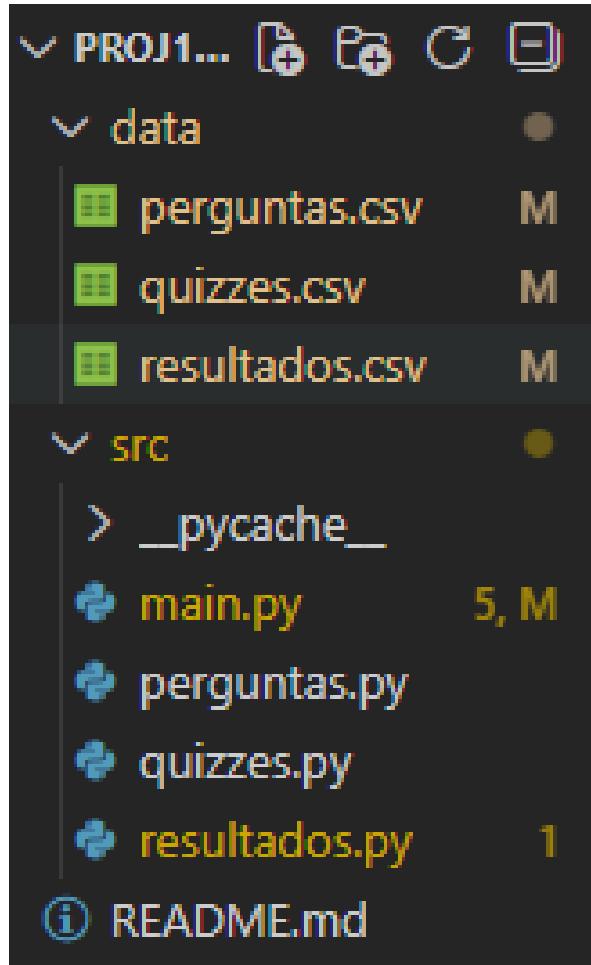
```
Mais alguma coisa?  
(0) Sim  
(1) Não  
Digite a opção desejada: 3  
Digite uma opção válida.
```

Usuário digita um número inválido em uma escolha

```
Digite a alternativa correta: F  
Digite uma alternativa válida (A, B, C, ...).  
Digite a alternativa correta:
```

Usuário digita uma alternativa inválida para ser a correta na montagem de perguntas

# Organização do Código e Módulos



main.py: lida com o menu e a interatividade com o usuário

perguntas.py: define a classe Pergunta e lida com salvar e carregar estas em/de perguntas.csv

quizzes.py: define a classe Quiz e lida com salvar e carregar estes em/de quizzes.csv

resultados.py: define a classe Resultado e lida com salvar e carregar estes em/de resultados.csv

# Classes e Orientação a Objetos

```
class Resultado:  
    """ Classe que lida com resultados """  
    def __init__(self, idr, quiz, nome_participante, nota, data):  
        self.idr = idr  
        self.quiz = quiz  
        self.nome_participante = nome_participante  
        self.nota = nota  
        self.data = data  
    def para_csv(self):  
        """ Transforma um Resultado em uma linha csv """  
        return [  
            str(self.idr),  
            str(self.quiz.idq),  
            self.nome_participante,  
            str(self.nota),  
            self.data  
        ]
```

Classe Resultado, que representa um resultado obtido de uma aplicação de um quiz. Contém um objeto da classe Quiz (self.quiz). O método para\_csv mostrado serve para facilmente converter uma entidade de Resultado em uma linha a ser salva em um arquivo csv.

# Classes e Orientação a Objetos

```
class Quiz:  
    """ Classe que lida com quizzes """  
    def __init__(self, idq, titulo, lista_perguntas):  
        self.idq = idq  
        self.titulo = titulo  
        self.lista_perguntas = lista_perguntas
```

```
@property  
def titulo(self):  
    """ Getter de titulo """  
    return self._titulo  
@titulo.setter  
def titulo(self, titulo):  
    """ Pode levantar ValueError caso titulo não seja uma string ou seja uma vazia """  
    if not isinstance(titulo, str) or not titulo.strip():  
        raise ValueError('titulo invalido')  
    self._titulo = titulo
```

Classe Quiz, que representa os quizzes criados pelo usuário, usando as perguntas já no sistema. Possui uma lista de entidades da classe Pergunta. As propriedades mostradas servem para averiguar se o atributo titulo foi definido corretamente (String não vazia)

# Leitura e Escrita de Arquivos

```
0,0,tester,0,19/01/2026  
1,1,tester,2,19/01/2026  
2,2,tester,3,19/01/2026  
3,3,Jorge,0,09/02/2026  
4,3,Jorge,1,09/02/2026
```

Arquivo resultados.csv

```
0,Vai rodar,0  
1,Teste de mais linhas,0,1  
2,Ultimo teste,0,1,2  
3,Pergunta de matemática,3
```

Arquivo quizzes.csv

```
0,Vai rodar?,Sim,Não,Claro,Negativo,A  
1,"Testando, vai funcionar?",Claro,Nem ferrando,Com certeza,Será?,C  
2,"Mais uma pergunta a ser testada, vai dar certo?",Obvio,Nunca,Será?  
3,Quanto é 1 + 1 ?,3,5,7,2,D
```

Arquivo perguntas.csv

# Leitura e Escrita de Arquivos

```
if not lista_arq_perguntas:  
    return []  
quizzes = []  
numero_linha = 0  
with open(PATH_CSV, newline = '', encoding = 'utf-8') as arq:  
    leitor = csv.reader(arq, delimiter = ',')  
    for linha in leitor:  
        numero_linha += 1  
        if not linha or all(cedula.strip() == '' for cedula in linha):  
            continue  
        lista_ids_perguntas = []  
        lista_perguntas = []  
        if len(linha) < TAMANHO_MINIMO:  
            raise ValueError(f'(Quizzes) Linha {numero_linha} está errada')  
        idq = int(linha[0].strip())
```

Leitura do arquivo quizzes.csv

```
with open(PATH_CSV, 'a', newline = '', encoding = 'utf-8') as arq:  
    escritor = csv.writer(arq)  
    escritor.writerow(quiz.para_csv())
```

Escrita no arquivo quizzes.csv

Os dados são carregados no início do programa, averiguando se estão no formato correto ou não.

Os dados são salvos ao adicionar algo, garantindo que mesmo que o programa seja encerrado de uma forma não esperada, os dados continuarão salvos.

# Desafios e Aprendizados

A principal dificuldade foi lidar com atributos que eram entidades de outra classe na hora de salvar e carregar dos arquivos csv, pois se fossem salvos de forma simples as linhas ficariam gigantes e confusas, logo uma outra solução envolvendo um fator identificador destas foi utilizada.

O projeto me ensinou muito sobre programação utilizando Python, me permitindo ter maior domínio da linguagem e suas mecânicas como um todo.

Se eu pudesse recomeçar, com certeza iria criar um módulo separado apenas para lidar com arquivos, o que facilitaria ao todo a utilização destas funções e tornaria o código melhor modularizado.

# Possíveis Melhorias Futuras

Implementação de uma GUI, tornando o programa mais fácil de ser utilizado e compreendido, além de esteticamente belo, criação de perguntas com número customizável de alternativas, criação de contas de dois tipos diferentes (alunos, que só respondem a quizzes, e professores que podem criar perguntas e montar quizzes, além de aplicá-los) e cálculo da nota média de um quiz específico.

Para evoluir o código, imagino que seria interessante, como citado no slide anterior, criar um módulo exclusivo para lidar com arquivos. Além disso, separar o módulo main em 2: Um com as funções do menu, e o outro para execução, permitindo o código ficar melhor no geral.