

## Projeto Final – “Qualifica”

O que sua equipe vai construir

Vocês vão criar um programa em Python que simula um aplicativo de aprendizagem profissional — parecido com o Duolingo, mas focado em habilidades como comunicação, lógica, tecnologia e organização no trabalho. Tudo será feito no computador, sem internet, usando apenas Python e o banco de dados SQLite3.

O sistema terá três tipos de usuários, cada um com funções diferentes:

- Aluno: faz atividades, ganha estrelas e acompanha seu progresso.
  - Professor: adiciona cursos e atividades pelo próprio programa.
  - Coordenador: vê relatórios gerais (ex: ranking, desempenho da turma).
- 

### Funcionalidades por tipo de usuário

#### 1. Login com tipo de usuário

- Ao iniciar, o usuário escolhe: “Aluno”, “Professor” ou “Coordenador”.
  - Depois digita nome e senha.
  - Se for um nome novo, o sistema pergunta o tipo e cadastra.
  - Cada tipo tem acesso a um menu diferente.
- 

#### 2. Funcionalidades do ALUNO

- Ver lista de cursos/módulos disponíveis
- Fazer atividades variadas:
  - Múltipla escolha
  - Verdadeiro ou Falso
  - Preencher lacunas
  - Ordenar etapas
  - Sequências lógicas
  - Correspondência (ligar colunas)
  - Classificação (separar categorias)
  - Escolha múltipla (várias corretas)
  - Palavra embaralhada

- Mini-cenários com decisões
  - Pedir dica antes de responder
  - Ganhar 1 estrela por acerto
  - Perder 1 vida por erro (começa com 3 vidas por curso)
  - Ver seu progresso: estrelas, vidas, atividades feitas hoje
  - Resetar seu próprio progresso
  - Ver o ranking dos 3 melhores alunos
- 

### 3. Funcionalidades do PROFESSOR

- Adicionar novo curso/módulo (ex: "Gestão de Tempo")
- Adicionar nova atividade (escolhe o tipo, escreve pergunta, opções, resposta correta e dica)
- Ver a lista de cursos e atividades cadastradas
- Não vê estrelas nem ranking — só gerencia conteúdo

Tudo é salvo no mesmo banco de dados, sem precisar editar código!

---

### 4. Funcionalidades do COORDENADOR

- Ver ranking completo da turma (todos os alunos, com estrelas e acertos)
  - Ver quantas atividades foram feitas hoje por todos
  - Ver desempenho por curso (ex: "80% acertaram o módulo de Comunicação")
  - Não faz atividades nem edita conteúdo — só acompanha
- 

### 5. Personalização extra: arquivo atividades\_extra.txt

- Além do modo professor, o sistema lê um arquivo chamado atividades\_extra.txt ao iniciar.
  - Se houver novas atividades lá, ele adiciona automaticamente no banco.
  - Isso permite ao professor preparar atividades com antecedência, sem usar o programa.
- 

### Como organizar o trabalho

- Usem funções com def para cada tarefa (ex: menu\_aluno(), adicionar\_curso(), ver\_ranking()).
- Separem o código em arquivos:
  - main.py → menu principal, login e redirecionamento por tipo de usuário
  - banco.py → conexão, criação de tabelas, leitura do atividades\_extra.txt
  - atividades.py → tipos de atividade, correção, dicas
  - usuarios.py → cadastro, verificação de tipo, login
  - progresso.py → vidas, estrelas, histórico, ranking
- Tudo deve rodar localmente, sem internet.

A tabela usuarios terá uma coluna tipo (ex: "aluno", "professor", "coordenador").

---

#### 💡 Dica final

O objetivo não é fazer algo perfeito, mas funcional, organizado e feito em equipe. Usem o que já aprenderam:

- if, elif, else
- for, while
- listas, strings, funções
- comandos básicos do SQLite3 (SELECT, INSERT, UPDATE, DELETE)