

Guia do Projeto — Sistemas Web (HTML, CSS, JS + SQLite)

Obs. Você deve criar um arquivo MD no VsCode com o título ChatIA, você irá colocar nesses arquivos todas as suas conversas com IA para o desenvolvimento do projeto, isso faz parte da avaliação, não serão aceitos chats copiados de colegas.

Essa atividade é individual.

Serão aceitos apenas commits realizados em sala no horário da aula, commits fora desse horário terá nota zero.

1) Objetivo

Construir **um** mini-sistema web completo, **individual**, usando:

- **Front-end:** HTML5, CSS3 (Flex/Grid), JavaScript (ES6+, sem framework).
- **Back-end/API:** Python (FastAPI ou Flask), SQLite via SQLAlchemy (ou sqlite3).
- **Ferramentas:** VS Code, Copilot, Git/GitHub, Thunder Client/Insomnia.
- **Entrega:** repositório público no GitHub + relatório técnico.

Você deve escolher **um** dos três sistemas abaixo (Biblioteca, Vendas de Produtos, Escola) e seguir as especificações **do front** e **do back**. O Copilot pode gerar partes do código, mas você precisa **revisar, pedir ajustes, entender e documentar** o que foi feito.

2) Padrão Técnico (para todos)

- **Estrutura de pastas**
- /frontend
- index.html
- styles.css
- scripts.js
- /backend
- app.py (FastAPI/Flask)
- models.py

- database.py
 - seed.py
 - requirements.txt
 - README.md
 - REPORT.md
 - **API:** RESTful, retornando JSON, status codes (200, 201, 400, 404, 422, 500).
 - **SQLite:** app.db na pasta /backend.
 - **Seed:** script para inserir ~20 registros plausíveis na tabela principal.
 - **Acessibilidade:** aria-label, foco visível, contraste mínimo 4.5:1, navegação por teclado.
 - **Testes manuais:** coleção do Thunder Client/Insomnia ou arquivo .http no repo.
 - **GitHub:** repositório público dw2-<seunome>-<tema>, commits frequentes, tag v1.0.0.
-

3) Sistemas e Especificações do Front-end

3.1 Biblioteca (Catálogo e Empréstimos)

Identidade visual (cores e tipografia)

- Primária: #1E3A8A (azul marinho)
- Secundária: #F59E0B (âmbar)
- Acento: #10B981 (verde)
- Fundo: #F8FAFC (cinza muito claro)
- Texto principal: #0F172A
- Fonte sugerida: “Inter” ou “Roboto” (fallback sans-serif)

Layout

- Header fixo com logo/título “Biblioteca Escolar” + barra de busca.
- Sidebar à esquerda com filtros (Gênero, Ano, Status).
- Área principal com:
 - Card list de livros (capa, título, autor, ano, status).

- Modal “Novo Livro”.
- Modal “Empréstimo/Devolução”.

Páginas/Seções

1. **Catálogo** (home): lista de livros, busca por título/autor.
2. **Gerenciar**: formulário de cadastro/edição.
3. **Relatórios (front)**: exportar CSV/JSON da lista filtrada.

Formulário de Livro (front)

- Campos: titulo* (text, 3–90 chars), autor* (text), ano* (number 1900–ano atual), genero (select), isbn (text opcional), status* (select: “disponível”, “emprestado”).
- Validações front: required, limites de tamanho, range de ano, impedir título duplicado no array local antes da chamada.

Interações JS

- **CRUD** via fetch na API.
- **Filtro combinado** (gênero + ano + texto).
- **Ordenação** por título/ano com persistência no localStorage.
- **Paginação** (10 por página) ou **scroll infinito**.
- **Acessibilidade**: atalhos de teclado p/ abrir modal “Novo Livro” (ex.: Alt+N), foco gerenciado.

Regra de negócio (front + back)

- Impedir empréstimo se status = emprestado.
- Ao emprestar, registrar data_emprestimo (UTC), ao devolver, limpar e setar status=disponível.

3.2 Vendas de Produtos (Catálogo e Carrinho)

Identidade visual

- Primária: #0EA5E9 (azul claro)
- Secundária: #EF4444 (vermelho)
- Acento: #22C55E (verde)
- Fundo: #FFFFFF

- Texto: #111827
- Fonte sugerida: “Poppins” ou “Inter” (fallback sans-serif)

Layout

- Header com logomarca “Loja Escolar” + ícone do carrinho (badge de quantidade).
- Grid responsivo de cards de produtos (imagem, nome, preço, estoque).
- Drawer/Modal de carrinho (lista itens, subtotal, cupom).

Páginas/Seções

1. **Catálogo:** grid de produtos com busca por nome.
2. **Carrinho:** drawer/modal com itens.
3. **Admin (front):** formulário de produto (somente enquanto não há autenticação).

Formulário de Produto (front)

- Campos: nome* (3–60), descricao (textarea), preco* (decimal ≥ 0.01), estoque* (int ≥ 0), categoria* (select), sku (opcional).
- Validações front: required, min/max, número positivo, preço com 2 casas.

Interações JS

- **CRUD** via fetch.
- **Carrinho** no localStorage (adicionar/remover/atualizar).
- **Cupom:** aplicar código “ALUNO10” (10% off) no front e validar no back.
- **Ordenação** por preço (asc/desc) e por nome.
- **Acessibilidade:** botões com aria-pressed e leitura clara para screen readers.

Regra de negócio (front + back)

- Não permitir adicionar ao carrinho se estoque=0.
- Ao confirmar “pedido” (mock), reduzir estoque na API e registrar total_final (com cupom, se válido).

3.3 Escola (Turmas e Matrículas)

Identidade visual

- Primária: #2563EB (azul)
- Secundária: #10B981 (verde)
- Acento: #F97316 (laranja)
- Fundo: #F1F5F9 (cinza claro)
- Texto: #0B1220
- Fonte sugerida: “Roboto” ou “Inter” (sans-serif)

Layout

- Header com “Gestão Escolar” + busca por aluno.
- Duas colunas: à esquerda filtro/estatísticas rápidas; à direita listagem principal.
- Modais para **Novo Aluno** e **Nova Matrícula**.

Páginas/Seções

1. **Alunos**: listagem com filtros (turma, status).
2. **Turmas**: listagem de turmas com capacidade e ocupação.
3. **Relatórios (front)**: exportar CSV/JSON de alunos ou matrículas.

Formulário de Aluno (front)

- Campos: nome* (3–80), data_nascimento* (date), email (pattern), status* (ativo/inativo), turma_id (select opcional).
- Validações front: required, data válida ≥ 5 anos atrás, email regex, status.

Interações JS

- **CRUD** via fetch.
- **Filtro combinado** (turma + status + texto).
- **Ordenação** por nome/idade.
- **Indicadores**: total de alunos, ativos, por turma.
- **Acessibilidade**: foco visível, legenda em tabelas, aria-live para feedback de operações.

Regra de negócio (front + back)

- Ao matricular, **não exceder capacidade** da turma.
- Alterar status do aluno para “ativo” ao ser matriculado.

4) Especificações do Back-end (para os três)

Use **FastAPI** + **SQLAlchemy** + **SQLite** (ou Flask + sqlite3/SQLAlchemy). Entidades mínimas e endpoints:

Biblioteca

- **Entidade Livro:** id, titulo, autor, ano, genero, isbn?, status, data_emprestimo?
- **Endpoints**
 - GET /livros?search=&genero=&ano=&status=
 - POST /livros (validações)
 - PUT /livros/{id}
 - DELETE /livros/{id}
 - POST /livros/{id}/emprestar (valida status; seta data)
 - POST /livros/{id}/devolver (valida status; limpa data)

Vendas de Produtos

- **Entidade Produto:** id, nome, descricao?, preco, estoque, categoria, sku?
- **Entidade Pedido** (opcional simplificada): id, total_final, data
- **Endpoints**
 - GET /produtos?search=&categoria=&sort=
 - POST /produtos
 - PUT /produtos/{id}
 - DELETE /produtos/{id}
 - POST /carrinho/confirmar (body com itens; valida estoque; aplica cupom “ALUNO10”; baixa estoque; cria pedido)

Escola

- **Entidade Turma:** id, nome, capacidade
- **Entidade Aluno:** id, nome, data_nascimento, email?, status, turma_id?
- **Endpoints**
 - GET /alunos?search=&turma_id=&status=

- POST /alunos
- PUT /alunos/{id}
- DELETE /alunos/{id}
- GET /turmas
- POST /turmas
- POST /matriculas (body: aluno_id, turma_id → valida capacidade; seta aluno.status=ativo, aluno.turma_id)

Regras gerais

- Validações coerentes no back-end (espelhando as do front).
- Respostas com mensagens claras de erro.
- Seeds de dados plausíveis (20 registros).

5) Peculiaridades obrigatórias (3 de 10)

Implemente qualquer **3** (vale para qualquer sistema):

1. Acessibilidade real (tabindex, aria, foco; descreva no relatório).
2. Validações custom no front e back (ex.: regras de faixa etária, estoque, ano).
3. Filtro avançado (múltiplos critérios) sem recarregar.
4. Ordenação persistida (localStorage).
5. Paginação ou scroll infinito.
6. Export CSV/JSON da lista atual.
7. Seed script com dados plausíveis.
8. Testes de API (coleção Insomnia/ThunderClient ou .http).
9. Tratamento de erros com toasts/feedback visual + HTTP codes.
10. Tema claro/escuro com persistência.

6) Roteiro de execução (quatro encontros)

Dia 1: criar repo, base do back (FastAPI + SQLite + /health), base do front (layout, header, lista, modais), commit inicial.

Dia 2: CRUD principal + 1 peculiaridade, testes de API, commits.

Dia 3: filtros/ordenção, acessibilidade, mais peculiaridades, REPORT.md em progresso.

Dia 4: seed, refino, prints/gifs, README/REPORT final, tag v1.0.0.

7) Git/GitHub (obrigatório)

- Repositório público: dw2-<seunome>-<tema>.
 - Commits pequenos e frequentes (mensagens claras).
 - Inclua: código, seed, coleção de API, README, REPORT, prints/gifs.
 - Crie **release** v1.0.0.
-

8) Relatório Técnico (REPORT.md)

Inclua:

- **Arquitetura** (diagrama simples + fluxo req→API→ORM→SQLite→resposta).
- **Tecnologias e versões** (Python, FastAPI/Flask, SQLAlchemy, SQLite, libs JS, extensões VSCode; cite o que o Copilot sugeriu).
- **Prompts do Copilot** (mínimo 6), com trechos aceitos/editados e **por quê**.
- **Peculiaridades implementadas** (quais e como).
- **Validações**: exemplos no front e no back.
- **Acessibilidade**: o que foi aplicado.
- **Como rodar** (passo a passo + prints de sucesso).
- **Limitações e melhorias futuras**.

Divisão dos trabalhos

NOME	TRABALHO
Daniel Godoy de Oliveira Silva	BIBLIOTECA
Nicolas Correa	BIBLIOTECA
Hannah Sophia Reis Franco	BIBLIOTECA
Anthony Rodrigues Guerra	BIBLIOTECA
Jhenifer Rodrigues Salatti	BIBLIOTECA
Marcelle dos Santos da Silva Perez	BIBLIOTECA
Maria Beatriz Oliveira de Souza	BIBLIOTECA
Robson Gabriel da Silva Rangel	BIBLIOTECA
Yuri Kaian Ribeiro Costa	BIBLIOTECA
Arthur Luiz Monteiro Macedo	ESCOLA
Wendel Gabriel Teodoro Munhos	ESCOLA
Gabrielly Gonçalves Custódio	ESCOLA
Thales de Társis Cezare Filho	ESCOLA
Victor Gabriel de Paula Quintiliano	ESCOLA
Vinicius Oliveira da Cunha Claro	ESCOLA
Danilo César Silva de Almeida	ESCOLA
Samuel Xavier Mubarak	ESCOLA
Daniel Abreu Costa	VENDAS
Evelyn Forti Simão	VENDAS
Laura Constantino Saab	VENDAS
Luccas Ribeiro	VENDAS
Pedro Gabriel Machi Dantas	VENDAS
Eliseu Rogério Fagundes Serrano	VENDAS
Giulia Oliveira Cei	VENDAS
Henzo Marques dos Santos Souza	VENDAS
Lorrany Eloisa dos Santos Andrade	VENDAS
Marxson Moreira da Cruz	VENDAS
Anna Beatriz Ramirez Pereira	BIBLIOTECA
Arthur Martins Moraes Alves	ESCOLA
Connie Francis Batista Ribeiro	
Leite	BIBLIOTECA
Jhony Samuel Freitas	ESCOLA
Laura Mazon Zaguette	BIBLIOTECA
Leonardo Heyer Lourenço	ESCOLA
Neo Lucca Viana e Silva	VENDAS