

# Divisão do Projeto (4 pessoas)

Tema: biblioteca

## 1. César — Backend (API + Regras de Negócio)

Responsável principal pela API

Tarefas:

- Escolher o framework backend (ex.: **Flask** / **FastAPI** / **Express**)
- Criar a estrutura do projeto
- Implementar os endpoints obrigatórios:
  - `GET /items`
  - `POST /items`
  - `PUT /items/:id`
  - `PATCH /items/:id/status`
  - `DELETE /items/:id`
- Implementar **todas as validações obrigatórias**:
  - título  $\geq 3$  caracteres
  - tipo válido
  - status válido
  - valor  $\geq 0$  (se existir)
- Criar arquivo de **configuração** (`config.js` ou `settings.py`) com:
  - tipos permitidos
  - status permitidos
- Padronizar respostas de erro e sucesso

- Testar a API (Postman / Insomnia)

**Entrega:** API funcional + validações + documentação básica das rotas

---

## 2. Lídia — Frontend (UI + Layout + UX)

**Responsável pela interface e experiência do usuário**

**Tarefas:**

- Criar o projeto frontend (React ou Vue)
- Configurar **TailwindCSS**
- Criar a **tela única** do sistema:
  - formulário (criar / editar item)
  - lista ou tabela de itens
- Criar componentes:
  - Formulário
  - Lista/Tabela
  - Botões (Editar, Remover, Mudar Status)
- Implementar estados visuais:
  - loading
  - mensagens de erro
  - feedback de sucesso (“Item cadastrado!”)

**Entrega:** Interface completa, organizada e responsiva (mesmo sem dados reais no início)

---

## 3. Godofredo — Integração Front ↔ Backend

**Responsável por conectar tudo e garantir funcionamento real**

**Tarefas:**

- Implementar consumo da API usando **fetch ou axios**
- Integrar:
  - listagem de itens
  - criação
  - edição
  - remoção
  - mudança de status
- Implementar filtros:
  - por tipo
  - por status
- Tratar erros da API no frontend
- Configurar **CORS** no backend (em conjunto com César)
- Garantir que o front funcione com dados reais

**Entrega:** Sistema totalmente funcional consumindo a API

---

## 4. Ezaely — Persistência + Organização do Projeto

**Responsável pelos dados, estrutura e qualidade**

**Tarefas:**

- Implementar a **persistência**:
  - `items.json` (opção recomendada)
  - leitura e escrita segura no arquivo
- Garantir que os dados **não se percam ao reiniciar**
- Criar estrutura de pastas organizada (back e front)

- Escrever o **README.md** do projeto:
  - como rodar backend
  - como rodar frontend
  - lista de endpoints
  - regras de validação
- Gerar os **prints/gif** do sistema funcionando
- Conferir se todos os requisitos do enunciado foram atendidos

**Entrega:** Projeto organizado, dados persistentes e documentação completa