

/projeto-python-fullstack

```
├── frontend/
│   ├── public/
│   │   └── index.html
│   ├── css/
│   │   └── style.css
│   ├── js/
│   │   └── main.js
│   └── assets/
│       └── logo.png
├── backend/
│   ├── app/
│   │   ├── main.py
│   │   ├── models.py
│   │   ├── routes.py
│   │   ├── auth.py
│   │   └── utils.py
│   ├── requirements.txt
│   └── Dockerfile
├── database/
│   ├── init.sql
│   ├── seed.sql
│   └── postgres.dockerfile
├── modelagem-arquitetura/
│   ├── uml/
│   │   ├── casos_uso.png
│   │   ├── classes.png
│   │   ├── sequencia.png
│   │   └── atividades.png
│   ├── banco_dados/
│   │   ├── MER.png
│   │   └── DER.png
│   ├── arquitetura.md
│   └── prototipos/
│       └── wireframes_figma.png
├── materiais-suporte/
│   ├── checklist_planejamento.pdf
│   ├── apostila_git.pdf
│   └── exemplos_codigo/
│       └── utilitarios.py
├── projeto-final/
│   ├── imagens/
│   │   ├── tela_login.png
│   │   └── tela_dashboard.png
```

```
| | | — fluxo_usuarios.png
| | | — explicacao.md
| — docker-compose.yml
| — README.md
```

(projetos node, php e csharp seguem a mesma estrutura, com ajustes para backend e banco de dados específicos)

```
/projeto-node-fullstack
| — frontend/ [...]
| — backend/ [...]
| — database/
| | — init.sql
| | — seed.sql
| | — mysql.dockerfile
| — modelagem-arquitetura/ [...]
| — materiais-suporte/ [...]
| — projeto-final/ [...]
| — docker-compose.yml
| — README.md
```

```
/projeto-php-fullstack
| — frontend/ [...]
| — backend/ [...]
| — database/
| | — init.sql
| | — mysql.dockerfile
| — modelagem-arquitetura/ [...]
| — materiais-suporte/ [...]
| — projeto-final/ [...]
| — docker-compose.yml
| — README.md
```

```
/projeto-csharp-fullstack
| — frontend/ [...]
| — backend/ [...]
| — database/
| | — init.sql
| | — sqlserver.dockerfile
| — modelagem-arquitetura/ [...]
| — materiais-suporte/ [...]
| — projeto-final/ [...]
```

```
|— docker-compose.yml
|— README.md
```

# Estrutura e arquivos principais em cada projeto

---

## 1. Projeto Python Fullstack

/projeto-python-fullstack

```
|— frontend/
|   |— public/
|   |   |— index.html      # Página principal HTML
|   |— css/
|   |   |— style.css       # Estilos CSS
|   |— js/
|   |   |— main.js        # Scripts JS para frontend
|   |— assets/
|   |   |— logo.png       # Imagens e assets
|— backend/
|   |— app/
|   |   |— main.py        # Ponto de entrada da aplicação (FastAPI ou Django)
|   |   |— models.py      # Modelos ORM do banco
|   |   |— routes.py      # Definição das rotas/endpoints
|   |   |— auth.py        # Autenticação (JWT, OAuth etc.)
|   |   |— utils.py       # Funções auxiliares
|   |— requirements.txt   # Dependências Python
|   |— Dockerfile        # Containerização backend Python
|— database/
|   |— init.sql           # Script para criação das tabelas
|   |— seed.sql           # Dados iniciais para popular o banco
|   |— postgres.dockerfile # Dockerfile para rodar banco PostgreSQL
|— modelagem-arquitetura/
|   |— uml/
|   |   |— casos_uso.png  # Diagrama casos de uso
|   |   |— classes.png    # Diagrama de classes
```

```

├─── sequencia.png      # Diagrama de sequência
├─── atividades.png     # Diagrama de atividades
├─── banco_dados/
│   ├─── MER.png       # Modelo entidade relacionamento
│   ├─── DER.png       # Diagrama entidade relacionamento
├─── arquitetura.md     # Documento de arquitetura (MVC, camadas etc)
├─── prototipos/
│   └─── wireframes_figma.png # Protótipos de tela / wireframes
├─── materiais-suporte/
│   ├─── checklist_planejamento.pdf # Lista de verificação do planejamento
│   ├─── apostila_git.pdf           # Apostila sobre Git e GitHub
│   ├─── exemplos_codigo/
│   │   └─── utilitarios.py         # Exemplos de funções úteis
├─── projeto-final/
│   ├─── imagens/
│   │   ├─── tela_login.png        # Imagem da tela de login
│   │   ├─── tela_dashboard.png    # Imagem da dashboard
│   │   └─── fluxo_usuarios.png     # Diagrama de fluxo do usuário
│   └─── explicacao.md              # Documentação explicativa do projeto final
├─── docker-compose.yml             # Orquestração de containers: frontend, backend, banco
└─── README.md                      # Documentação geral do projeto

```

---

## 2. Projeto Node.js Fullstack

/projeto-node-fullstack

```

├─── frontend/           # Igual estrutura frontend do Python
│   ├─── public/
│   ├─── css/
│   ├─── js/
│   └─── assets/
├─── backend/
│   ├─── src/
│   │   ├─── controllers/ # Lógica de negócio e controle
│   │   ├─── models/      # Modelos do banco (ORM/ODM)
│   │   ├─── routes/      # Definição das rotas API
│   │   └─── app.js        # Arquivo principal Express
│   ├─── package.json     # Dependências e scripts Node.js
│   └─── Dockerfile       # Container backend Node.js

```

```
|
|— database/
|   |— init.sql      # Script criação banco (MySQL ou PostgreSQL)
|   |— seed.sql      # Script para popular banco
|   |— mysql.dockerfile # Dockerfile para banco MySQL
|
|— modelagem-arquitetura/ # Mesma organização com diagramas e protótipos
|— materiais-suporte/
|— projeto-final/
|— docker-compose.yml
|— README.md
```

---

### 3. Projeto PHP Fullstack

```
/projeto-php-fullstack
|— frontend/      # Igual frontend geral
|— backend/
|   |— public/     # DocumentRoot para web server
|   |— routes/     # Arquivos de rotas
|   |— app/
|       |— Http/Controllers/ # Controllers MVC
|       |— Models/          # Modelos ORM/Eloquent
|       |— Views/           # Views (Blade ou similares)
|   |— composer.json      # Dependências PHP
|   |— Dockerfile
|
|— database/
|   |— init.sql
|   |— mysql.dockerfile
|
|— modelagem-arquitetura/
|— materiais-suporte/
|— projeto-final/
|— docker-compose.yml
|— README.md
```

---

### 4. Projeto C# Fullstack (ASP.NET Core)

```
/projeto-csharp-fullstack
|— frontend/      # Igual frontend geral
```

```
├── backend/
│   ├── Controllers/    # API controllers
│   ├── Models/        # Classes de modelo
│   ├── Services/      # Serviços da aplicação
│   ├── Program.cs      # Ponto de entrada
│   ├── Startup.cs     # Configuração da aplicação
│   ├── appsettings.json # Configurações
│   └── Dockerfile
├── database/
│   ├── init.sql
│   └── sqlserver.dockerfile # Dockerfile para banco SQL Server
├── modelagem-arquitetura/
├── materiais-suporte/
├── projeto-final/
├── docker-compose.yml
└── README.md
```

---

## Observações

- Todos os projetos têm o `docker-compose.yml` para facilitar o desenvolvimento integrado e deploy.
- Os arquivos `.sql` em `database/` permitem criar e popular o banco, bem como executar migrações manuais.
- `README.md` é fundamental para documentar o funcionamento e como executar o projeto.
- As pastas de modelagem, protótipos, materiais e projeto final são para dar suporte didático e organizacional, mostrando o “porquê” de cada escolha.