🏗️ Estrutura Final de Diretórios

wayne-security/

│

├── backend/

│ ├── app.py

│ ├── models/

│ │ ├── \_\_init\_\_.py

│ │ ├── usuario.py

│ │ ├── recurso.py

│ │ └── logs.py

│ ├── config.py

│ ├── requirements.txt

│ └── README.md

│

├── database/

│ ├── create\_db.sql

│ └── seed\_data.sql

│

├── frontend/

│ ├── index.html

│ ├── style.css

│ ├── app.js

│ └── README.md

│

├── .gitignore

├── README.md

└── LICENSE

## 🧠 Explicação da Estrutura

| Pasta/Arquivo | Função |
| --- | --- |
| backend/ | Contém toda a API Flask e integração com o MySQL |
| models/ | Classes em POO que representam tabelas (Usuários, Recursos, Logs) |
| config.py | Define variáveis de ambiente e conexão com o banco |
| requirements.txt | Dependências Python para instalação |
| database/ | Scripts SQL para criação e inserção inicial no banco |
| frontend/ | Interface visual do sistema (HTML, CSS e JS separados) |
| .gitignore | Ignora arquivos sensíveis e ambientes virtuais |
| LICENSE | Licença do projeto (MIT recomendada) |
| README.md | Documentação principal do repositório GitHub |

🧾 Exemplo de README.md para o GitHub

# 🦇 Wayne Security System

Sistema completo de \*\*Gerenciamento de Segurança\*\* desenvolvido para as \*\*Indústrias Wayne\*\*, integrando \*\*Flask + MySQL + Frontend Web\*\*.

> Projeto Full Stack inspirado na segurança de Gotham City, com tema sóbrio e profissional — sem infantilização.

---

## 🚀 Funcionalidades Principais

### 🔐 Controle de Acesso

- Autenticação e autorização de usuários (funcionário, gerente, administrador)

- Login protegido e níveis de permissão

### 🏭 Gestão de Recursos

- Cadastro e controle de veículos, equipamentos e dispositivos de segurança

- Atualização de status e localização dos recursos

### 📊 Dashboard de Segurança

- Painel visual com indicadores de segurança e atividades recentes

- Exibição de logs de acesso e uso de recursos

---

## 🧩 Tecnologias Utilizadas

| Categoria | Tecnologia |

|------------|-------------|

| \*\*Backend\*\* | Python 3, Flask, MySQL Connector |

| \*\*Banco de Dados\*\* | MySQL 8+ |

| \*\*Frontend\*\* | HTML5, CSS3, JavaScript Vanilla |

| \*\*Outros\*\* | REST API, JSON, Fetch API |

---

## ⚙️ Instalação e Execução

### 1️⃣ Clonar o Repositório

git clone https://github.com/seu-usuario/wayne-security.git

cd wayne-security

### 2️⃣ Banco de Dados

mysql -u root -p < database/create\_db.sql

mysql -u root -p < database/seed\_data.sql

### 3️⃣ Backend (Flask)

cd backend

python -m venv venv

venv/Scripts/activate # (Windows)

source venv/bin/activate # (Linux/Mac)

pip install -r requirements.txt

flask run

### 4️⃣ Frontend

Abra o arquivo frontend/index.html no navegador  
**ou** utilize:

cd frontend

python -m http.server

Depois, acesse [**http://localhost:8000**](http://localhost:8000/)

## 📦 Requirements.txt

Flask>=2.2

Flask-SQLAlchemy>=3.0

Flask-JWT-Extended>=4.5

python-dotenv>=1.0

pymysql>=1.0

passlib[bcrypt]>=1.7

## 🧰 Endpoints Principais (API)

| Método | Rota | Descrição |
| --- | --- | --- |
| POST | /login | Autenticação de usuário |
| GET | /usuarios | Lista todos os usuários |
| POST | /usuarios | Cria um novo usuário |
| GET | /recursos | Lista recursos cadastrados |
| POST | /recursos | Adiciona um novo recurso |
| GET | /logs | Retorna logs recentes |

## 🎨 Tema Visual

* **Paleta:** Preto, cinza grafite e azul escuro metálico
* **Fonte:** Segoe UI / Roboto
* **Estilo:** Minimalista, futurista e corporativo
* **Ícones:** (Opcional) Lucide Icons ou Heroicons

## 🎨 Temática e Design

O projeto segue uma **estética sóbria e moderna**, inspirada no universo do **Batman**, com tons escuros, contrastes metálicos e elementos que remetem à tecnologia de alta segurança.

## 🧠 Tecnologias Utilizadas

* **Frontend:** HTML5, CSS3, JavaScript (ES6+)
* **Backend:** Python 3 + Flask
* **Banco de Dados:** MySQL
* **Outras:** dotenv, Flask-MySQL, Bootstrap (para estilização leve)

## 🔐 Funcionalidades Principais

* Login e autenticação de usuários
* Controle de acesso baseado em níveis (funcionário, gerente, administrador)
* Cadastro, edição e exclusão de recursos
* Dashboard com dados de segurança em tempo real

## 🧑‍💻 Desenvolvido por

**Bruce Wayne** (patrocinador)  
**Bruno Lima** (desenvolvimento técnico e documentação)

## 🪪 Licença

Este projeto é licenciado sob a **MIT License** – veja o arquivo LICENSE para mais detalhes.

# Ambientes virtuais

venv/  
**pycache**/

# Arquivos temporários

\*.log  
\*.tmp  
\*.sqlite3

# Configurações locais

.env  
.DS\_Store