

# **Programação Web - Relatório do Trabalho Final**

Data: 01/06/2025

Grupo: Gabriel Costa (51786), Miguel Santos (52013), Francisco Branco (51829)

# Universidade da Beira Interior - Engenharia Informática

## Introdução

A aplicação desenvolvida tem como objetivo gerir o consumo e a produção de energia elétrica por parte de clientes com instalações solares. O sistema destina-se a três tipos de utilizadores: **Clientes**, **Técnicos** e o **Gestor de Operações**, cada um com permissões e funcionalidades específicas.

O sistema permite aos clientes registarem instalações solares, acompanhar os seus dados de consumo e produção, e acumular créditos de energia quando a produção excede o consumo. Os técnicos são responsáveis por aprovar as instalações, e o gestor de operações pode atualizar valores, calcular créditos e enviar comunicações mensais aos clientes.

Este projeto foi implementado no âmbito da unidade curricular de Programação Web, utilizando tecnologias modernas de desenvolvimento web como **Node.js**, **Express**, **MongoDB**, **HTML**, **CSS**, **JavaScript** e **JWT para autenticação**.

---

## Implementação

### 1. Tecnologias Utilizadas

- **Backend:**
    - Node.js
    - Express
    - MongoDB (com Mongoose)
    - JWT (JSON Web Tokens) para autenticação
    - Nodemailer para envio de emails
    - Multer para upload de certificados PDF
  - **Frontend:**
    - HTML5, CSS3, JavaScript
    - Fetch API para comunicação com o backend
  - **APIs externas simuladas:**
    - API de produção e consumo (<http://localhost:4000>) para simular dados reais de energia
- 

### 2. Funcionalidades Implementadas

#### Autenticação e Gestão de Utilizadores

- Registo e login com JWT
- Validação de token e role no backend

- Três tipos de utilizadores: Cliente, Técnico e Gestor de Operações

### Funcionalidade do Cliente

- Registo de novas instalações solares com:
  - Localização
  - Capacidade (kW)
  - Marca do painel
- Visualização de todas as instalações registadas
- Acompanhamento da **produção e consumo mensal** por instalação
- Visualização do **histórico anual** mês a mês da produção e consumo total
- Dashboard adaptado a cliente (sem botões de atualização nem créditos visíveis)

### Funcionalidade do Técnico

- Listagem de instalações pendentes
- Upload de **certificado em PDF** para aprovar instalações
- Interface limpa com filtro de pesquisa por nome, localização ou marca

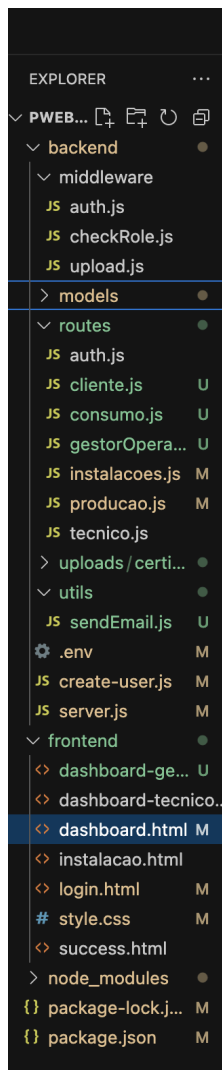
### Funcionalidade do Gestor de Operações

- Acesso a todas as instalações de todos os clientes
- Atualização manual da produção e consumo via APIs simuladas
- Visualização de:
  - Produção e consumo mensal por instalação
  - Créditos mensais e totais por instalação
  - Créditos mensais e totais por cliente
- Cálculo mensal de créditos de energia:
  - $\text{Créditos} = \text{Produção} - \text{Consumo}$  (sem valores negativos)
  - Armazenamento dos créditos e envio automático de emails informativos aos clientes

### Envio de Emails

- Implementado com nodemailer
  - Geração de emails mensais aos clientes contendo os créditos de energia
  - Integração segura com uma conta Gmail com autenticação por App Password
-

### 3. Estrutura do Projeto



- **/backend**
    - models/ – Modelos Mongoose para Utilizador, Instalação, Produção, Consumo, Créditos
    - routes/ – Rotas organizadas por tipo de utilizador e funcionalidade (auth, cliente, tecnico, gestorOperacoes, etc.)
    - middleware/ – Middleware de autenticação JWT e verificação de roles
    - utils/ – Função para envio de emails com nodemailer
    - server.js – Configuração principal do servidor Express
  - **/frontend**
    - dashboard-cliente.html, dashboard-tecnico.html, dashboard-gestor.html – Interfaces separadas por tipo de utilizador
    - style.css – Estilo comum adaptado com responsividade e design limpo
-

# Conclusão

A aplicação desenvolvida cumpre com sucesso todos os requisitos do enunciado, oferecendo um sistema funcional e dividido por permissões. A gestão energética dos clientes é realizada de forma clara, com registo de produção e consumo, cálculo e comunicação dos créditos de forma automatizada.

## Pontos fortes

- Separação clara de funcionalidades por tipo de utilizador
- Uso de APIs simuladas realistas para dados de energia
- Dashboard visualmente organizado e responsivo
- Envio automático de emails com dados reais de produção/consumo

## Limitações

- Os dados são gerados aleatoriamente pelas APIs, não sendo persistentes entre execuções
- O envio de emails depende de App Passwords da conta Google, o que pode não ser ideal em ambiente de produção
- Falta de autenticação multifator (2FA) para maior segurança

## Possíveis Melhorias Futuras

- Implementar gráficos de consumo/produktividade em tempo real
- Adicionar relatórios exportáveis em PDF para clientes
- Permitir a personalização de notificações e alertas
- Ligar a sensores ou sistemas IoT reais para recolher dados reais

---

Este projeto demonstrou a aplicação de diversos conhecimentos de desenvolvimento web fullstack, integração com APIs e gestão de dados com base em papéis e permissões de utilizadores, culminando numa solução funcional e prática.