



MESTRADO EM  
ENGENHARIA INFORMÁTICA E  
TECNOLOGIA WEB



## **Blog Pessoal – A Arte das Palavras**

### **Projeto Final – Relatório**

**Aluno:** André Costa (Nº 2302571)

**Unidade Curricular:** Programação *Web* Avançada

**Docentes:** Luís Barbosa e Ricardo Baptista

Fevereiro de 2024

## Índice

<b>1. Introdução.....</b>	<b>3</b>
1.1. Visão Geral da <i>Web App</i> Desenvolvida.....	3
1.2. Principais Funcionalidades da <i>Web App</i> .....	3
1.3. Público-Alvo e Propósito da Aplicação .....	3
1.4. Tecnologias Utilizadas .....	4
<b>2. Arquitetura e Design.....</b>	<b>5</b>
2.1. Arquitetura da Aplicação.....	5
2.2. <i>Design</i> da Interface do Utilizador .....	6
2.3. Fluxo de Navegação e Utilização .....	6
<b>3. Implementação Técnica.....</b>	<b>8</b>
3.1. Estrutura de Pastas/Ficheiros.....	8
3.2. Base de Dados .....	9
3.3. Descrição das Funcionalidades-Chave .....	10
<b>4. Deploy na Web .....</b>	<b>10</b>
<b>5. Conclusão.....</b>	<b>11</b>

## Índice de Figuras

Figura 1 - Página Inicial (Home Page) .....	7
Figura 2 - Página de Administração .....	8
Figura 3 - Base de Dados da Aplicação .....	9

## 1. Introdução

### 1.1. Visão Geral da *Web App* Desenvolvida

A *web app* desenvolvida é um *blog* pessoal de escrita, que permite que os utilizadores criem, visualizem, editem e removam as suas publicações no *blog*. Com uma interface intuitiva e responsiva, a aplicação oferece uma agradável experiência, tanto para os seus visitantes, que desejam explorar o seu conteúdo, como também para quem faz a gestão da mesma.

### 1.2. Principais Funcionalidades da *Web App*

As principais funcionalidades da *web app* desenvolvida incluem:

- Visualização das publicações na página inicial (*home page*);
- Visualização do conteúdo integral de todas as publicações;
- Registo e autenticação de utilizadores;
- Criação de novas publicações;
- Edição de publicações existentes;
- Remoção de publicações.

### 1.3. Público-Alvo e Propósito da Aplicação

Esta aplicação destina-se a dois grupos:

- **Escritores e *bloggers*:** Facilita a partilha de histórias, ideias e conhecimentos;
- **Leitores e visitantes:** Descobrem, exploram e interagem com conteúdos cativantes.

Decidi criar esta aplicação devido à minha paixão pela escrita. Acredito que a mesma é uma forma poderosa de expressarmos as nossas ideias, compartilharmos histórias e conectarmo-nos com outras pessoas. O propósito desta aplicação é proporcionar uma plataforma onde, através das palavras dos escritores, os leitores se possam encontrar e inspirar.

#### 1.4. Tecnologias Utilizadas

A aplicação foi desenvolvida utilizando as seguintes tecnologias:

- **Linguagens de Programação:**
  - **JavaScript (Node.js)** – Para o *backend* e para as interações do lado do cliente (*frontend*).
- **Frameworks e Bibliotecas:**
  - **Express.js** – Uma *framework web* para o *Node.js*, utilizada para criar APIs e manipular rotas;
  - **EJS (Embedded JavaScript)** – Linguagem de modelagem que permite gerar HTML com *JavaScript* embutido, sendo utilizada para criar as visualizações dinâmicas da aplicação;
  - **Bcryptjs** – Biblioteca utilizada para criptografar as senhas;
  - **Mongoose** – Biblioteca de modelagem de objetos *MongoDB* para o *Node.js*.

- **Ferramentas e Pacotes Adicionais:**

- **Dotenv** – Módulo do *Node.js* que carrega as variáveis de ambiente a partir de um arquivo “.env”, de forma a garantir a segurança das credenciais;
- **Express-ejs-layouts** – *Middleware* para o *Express.js* que simplifica a utilização de *layouts* com EJS;
- **Method-override** – *Middleware* para o *Express.js* que permite utilizar métodos HTTP, como o “PUT” e o “DELETE”, em formulários HTML;
- **Connect-mongo** – Pacote que permite armazenar as sessões do *Express* no *MongoDB*;
- **Express-session** – *Middleware* para o *Express.js*, utilizado para gerir as sessões do utilizador.

## 2. Arquitetura e *Design*

### 2.1. Arquitetura da Aplicação

A arquitetura da aplicação segue o padrão MVC (*Model-View-Controller*):

- **Modelo (*Model*)** – Representado nos ficheiros da pasta “*models*”, define a estrutura dos dados da aplicação, tais como os utilizadores e as publicações (*posts*). Utiliza o *Mongoose* para interagir com a base de dados *MongoDB*;

- **Visualização (*View*)** – Representado pelos ficheiros “.ejs”, existentes na pasta “*views*”, são responsáveis pela apresentação da interface do utilizador. Estes ficheiros são renderizados dinamicamente pelo *Express.js*;
- **Controlador (*Controller*)** – Representado pelos ficheiros da pasta “*routes*”, controlam o fluxo de solicitações HTTP. Lidam, igualmente, com as operações de criação, leitura, edição e remoção (CRUD) dos dados, assim como a autenticação dos utilizadores.

## 2.2. *Design* da Interface do Utilizador

O *design* da interface do utilizador é intuitivo e fácil de utilizar, com uma navegação clara e elementos bem organizados. São utilizados *layouts* responsivos, de forma a garantir uma experiência consistente em diferentes dispositivos e tamanhos de ecrã. A interface do utilizador é limpa e minimalista, estando o foco no conteúdo, facilitando assim a leitura e a interação dos utilizadores.

## 2.3. Fluxo de Navegação e Utilização

O fluxo de navegação começa na página inicial (*home page*), onde os utilizadores podem ver uma lista das publicações (*posts*) mais recentes. Ao clicarem no título de uma publicação, os utilizadores são direcionados para a página da mesma, onde podem ler todo o seu conteúdo.

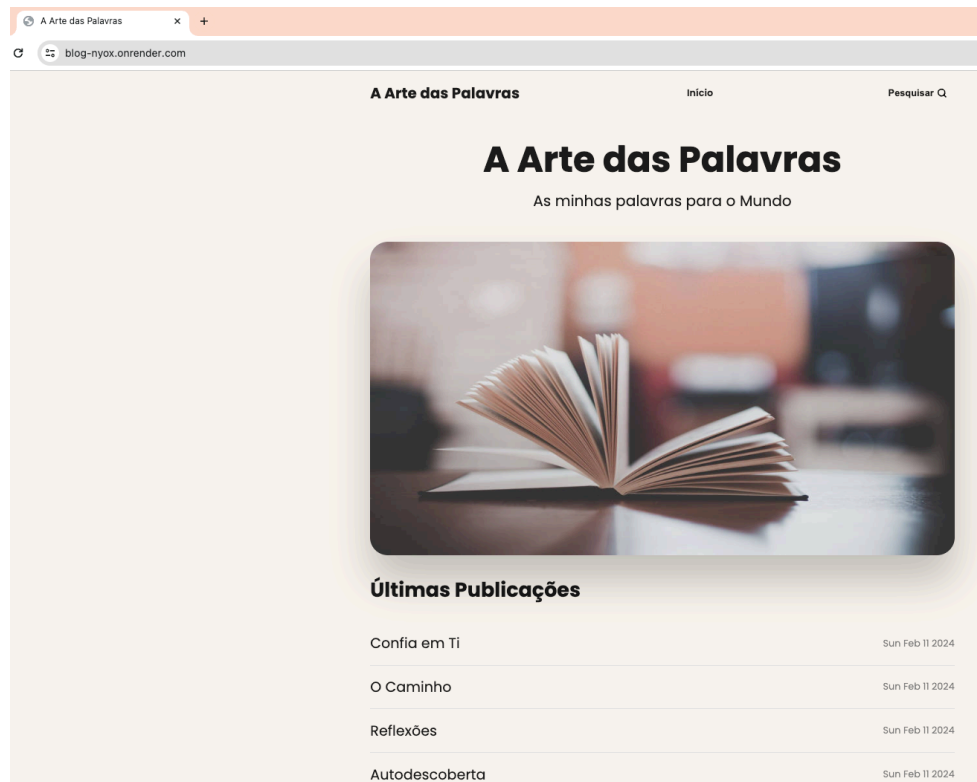


Figura 1 - Página Inicial (*Home Page*)

Adicionalmente, existe uma página de administração onde os utilizadores se podem registar ou iniciar sessão (*login*). Após iniciarem sessão, os utilizadores têm acesso a um “*dashboard*” onde podem criar, editar e remover as suas publicações. O processo de criação, edição e remoção das publicações é simples e direto, proporcionando desta forma uma experiência eficiente para os utilizadores.

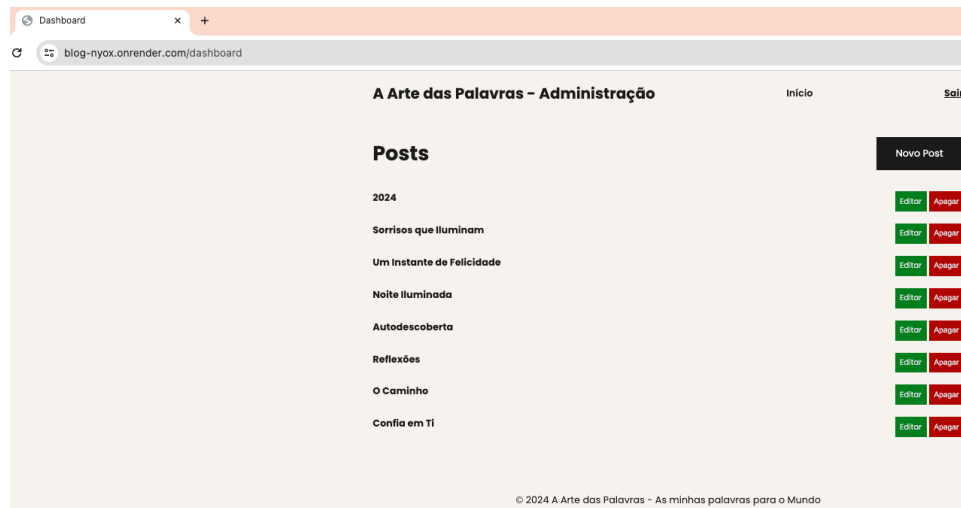


Figura 2 - Página de Administração

## 3. Implementação Técnica

### 3.1. Estrutura de Pastas/Ficheiros

- ***design\_elements***: Contém os ficheiros “estáticos” (CSS e *JavaScript*);
- ***server***: Contém os ficheiros relacionados com a lógica do servidor, que inclui configurações, modelos, rotas e *middleware*;
- ***views***: Contém os ficheiros de visualização EJS, que renderizam a interface do utilizador;
- ***views/admin***: Contém os ficheiros com as visualizações específicas do painel de administração;
- ***views/layouts***: Contém os ficheiros com os *layouts* EJS, partilhados pelas várias visualizações;



- **views/sections:** Contém os ficheiros com a secções reutilizáveis, tais como o cabeçalho (*header*) e o rodapé (*footer*).

### 3.2. Base de Dados

A aplicação utiliza o *MongoDB Atlas* como base de dados, com o *Mongoose* como ORM (*Object-Relational Mapping*) para interagir com a base de dados a partir do *Node.js*.

Os dados são estruturados em coleções, que incluem os utilizadores e as publicações (*posts*), conforme definido nos modelos “*User.js*” e “*Post.js*”.

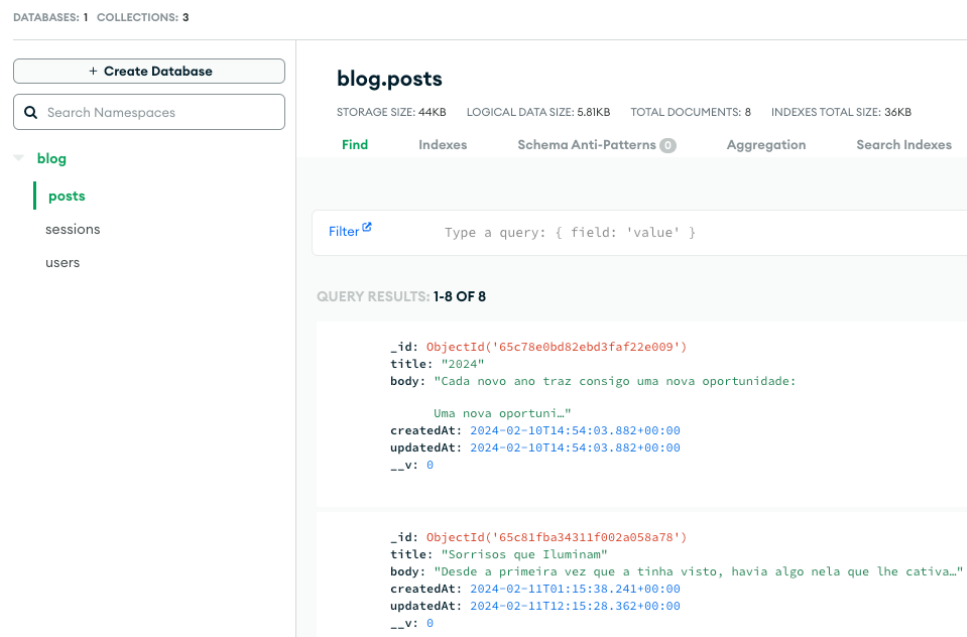


Figura 3 - Base de Dados da Aplicação

**NOTA:** Foi efetuado um *export* da base de dados, encontrando-se os ficheiros na pasta “DBs”, dentro da pasta do projeto.

### 3.3. Descrição das Funcionalidades-Chave

- **Registo de utilizador e autenticação:** Implementado com rotas e *middleware*, de forma a lidar com as solicitações de registo, *login* e autenticação dos utilizadores. Para a autenticação dos utilizadores no painel de administração é utilizado o JWT (*JSON Web Token*);
- **Gestão de publicações (*posts*):** As funcionalidades CRUD para as publicações são geridas através de rotas e controladores, permitindo assim aos utilizadores criarem, visualizarem, editarem e excluïrem publicações.

## 4. Deploy na Web

O *deploy* da aplicação na *web* foi efetuado na plataforma de hospedagem *Render*, permitindo assim que a aplicação seja acedida pelo público em geral de forma rápida e segura.

**Link de acesso à página inicial (*home page*):**

<https://blog-nyox.onrender.com/>

**Link de acesso ao painel de administração:**

<https://blog-nyox.onrender.com/admin>

## 5. Conclusão

Este projeto representou um desafio estimulante, onde me deparei com alguns obstáculos ao longo do caminho, fruto do pouco conhecimento de algumas das tecnologias utilizadas. No entanto, considero o resultado final bastante positivo, já que a aplicação desenvolvida cumpre os requisitos solicitados, e proporciona uma experiência fluída e funcional.

A realização deste trabalho permitiu-me expandir tanto os meus conhecimentos como as minhas habilidades.

***Link do repositório no *GitHub*:***

[https://github.com/andrecosta14/PWA\\_ProjetoFinal\\_Blog](https://github.com/andrecosta14/PWA_ProjetoFinal_Blog)