

# *Website de Portefólio*



**Projeto e relatório criado por:** Afonso Viegas Vieira Semeano

**Nº de aluno:** 202001614

**Curso:** Licenciatura em Engenharia Informática

**Instituição de Ensino:** Instituto Politécnico de Setúbal

**Professor:** Joaquim Filipe

**Ano:** 3º ano, 2022/2023

**Data:** 11/04/2023

## Índice

Introdução .....	3
Descrição do Projeto .....	4
Visão Geral.....	4
Âmbito.....	4
Tecnologias Utilizadas.....	4
Metodologia de desenvolvimento.....	5
Desafios Enfrentados .....	6
Detalhes Técnicos .....	7
Páginas do Website .....	7
Solução dos desafios.....	9
Como fazer o site responsivo? .....	9
Como pode o site suportar várias línguas e ser também fácil adicionar novas línguas quando necessário? .....	10
Como ter um routing que não causa erros e que permite associar uma língua e uma página?.....	11
Como ter um website que seja minimamente bonito?.....	12
Como fazer um sistema de login e registo? .....	12
Sistema de feedback .....	15
Conclusão.....	17

## Introdução

Quando estive a pensar que tipo de projetos deveria fazer para entrar no portefólio, concluí que o primeiro projeto tem de ser um website. Agora, um website de quê? Há muitos tipos de website. Poderia fazer um website sobre jogos de tabuleiro (um hobby meu), ou sobre comida. Mas então tive uma ideia... como estou a fazer um projeto para o meu portefólio, porque não criar um website de portefólio? Ou seja, um website que exponha todos os projetos que realizei até agora.

Este website funciona como se fosse um CV. Contém uma página que faz um resumo breve sobre o meu percurso e quem eu sou, e refere os projetos que realizei até agora.

Porém, não é uma página estática que apenas mostra informação sobre mim! É também possível fazer login e registo, alterar a língua mostrada e enviar feedback para mim. Tomei a decisão de ligar o website a uma base de dados não por razões de ser prático, mas sim porque, como é o meu primeiro projeto, queria aprender o máximo de funcionalidades possível, criando-as do zero.

Este projeto começou a ser desenvolvido desde o verão de 2022, e foi sendo desenvolvido ao longo dos meses, com muito estudo das tecnologias envolvidas. Foi oficialmente dado como terminado durante o mês de Fevereiro de 2023.

## Descrição do Projeto

### Visão Geral

Este projeto consiste num website que serve como um Curriculum Vitae interactivo. A ideia central foi ter uma maneira fácil e interactiva de visualizar quem eu sou e quais os projetos que realizei.

### Âmbito

No âmbito do projeto, incluiu-se páginas tais como a página principal, a página do Sobre Mim e a página de projetos. Em cada projeto da referida página, é possível ver uma descrição. Foi também incluída a possibilidade de fazer login/registo para se poder submeter feedback. É possível alterar a língua da aplicação de português para inglês, ou vice-versa.

### Tecnologias Utilizadas

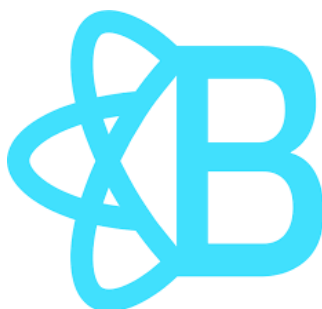
Para desenvolver o sistema, foram utilizadas as seguintes tecnologias:

**MongoDB** - Software de base de dados NoSQL orientada a documentos semelhantes a JSON com esquemas.



*Figura 1 - MongoDB*

**React-Bootstrap** - O React-Bootstrap é uma junção de React e Bootstrap, baseada em Javascript. O React oferece um estilo de programação que nos permite dividir o front-end em componentes funcionais, dinâmicos e reativos. O Bootstrap facilita na estilização desses componentes, já tendo classes CSS pré-feitas e comportamentos pré-feitos.



*Figura 2 - React-Bootstrap*

**Flask** - Framework de desenvolvimento web em Python que permite criar servidores web através de routing e manipular pedidos e respostas HTTP.



*Figura 3 - Flask*

**Sass** - Linguagem de pré-processamento de CSS que permite escrever código CSS de forma mais eficiente e organizada. Útil para facilitar a personalização dos estilos do Bootstrap.



*Figura 4 - Sass*

### Metodologia de desenvolvimento

Não foi planeada nenhuma metodologia de desenvolvimento em específico, porém, como escolhi tecnologias que não conhecia antes, o processo foi algo deste género:

- Preparar o ambiente de desenvolvimento do website em React-Bootstrap, MongoDB, Flask e Sass.
- Estudar Flask, Mongo e React-Bootstrap para poder fazer um protótipo inicial.
- Ir construindo os vários componentes do website, e, sempre que tinha alguma dúvida, pesquisava sobre o assunto.
- Ao terminar o aspeto principal do site, pedi opiniões de amigos e sofreu algumas alterações devido a certas cores não estarem boas e à formatação.

## Desafios Enfrentados

Durante o processo de desenvolvimento do website, houve vários desafios que tive de enfrentar e de resolver. Mantenha a noção de que iniciei este website antes de começar o meu 3º ano de universidade, então, apesar de alguns desafios parecerem simples agora, eram complicados na altura devido a falta de conhecimento. Irei falar com mais detalhes desses desafios com screenshots no tópico [Detalhes Técnicos > Solução dos Desafios](#).

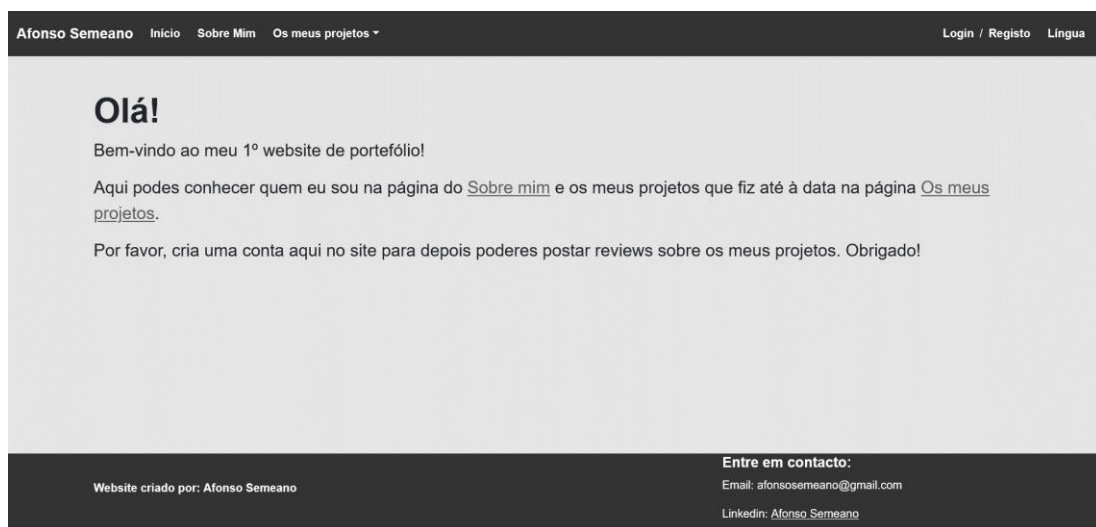
1. Como fazer o site responsivo?
2. Como pode o site suportar várias línguas e ser também fácil adicionar novas línguas quando necessário?
3. Como ter um routing que não causa erros e que permite associar uma língua e uma página?
4. Como ter um website que seja minimamente bonito?
5. Como fazer um sistema de login e registo?
6. Como fazer um sistema de feedback?

## Detalhes Técnicos

[Aqui está o link do github deste projeto](#)

Irei mostrar todas as páginas do site e irei abordar os desafios mencionados anteriormente e explicar como resolvi certos problemas através de screenshots comprovativos.

## Páginas do Website



*Figura 5 - Página Inicial*



*Figura 6 - Sobre Mim*



Figura 7 - Os meus projetos

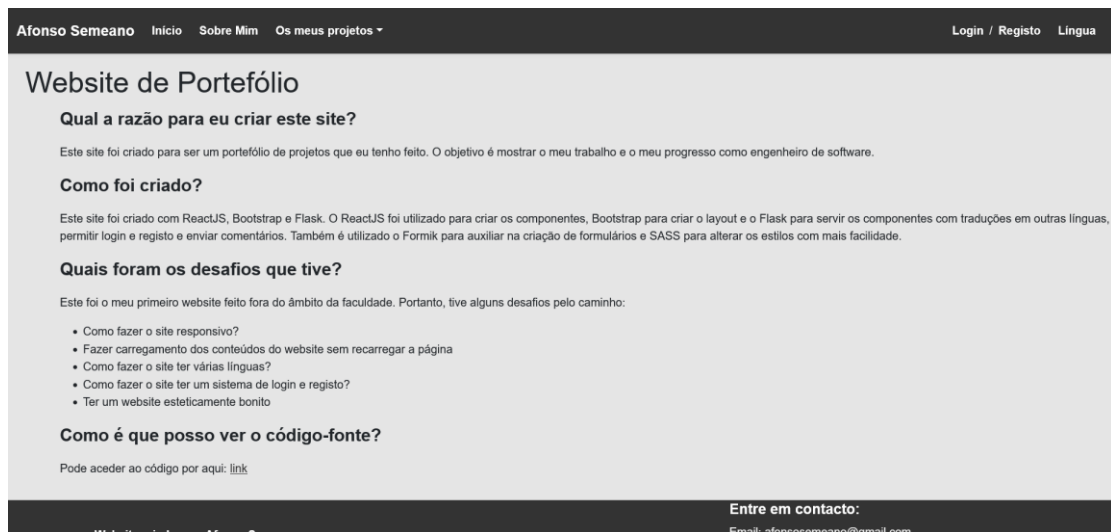


Figura 8 - Página descritiva do próprio website

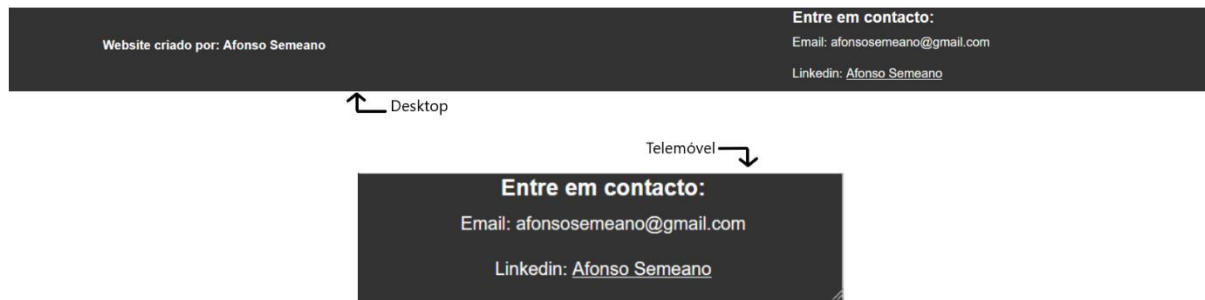


## Solução dos desafios

Vou falar agora de como resolvi os desafios que tive ao longo do projeto.

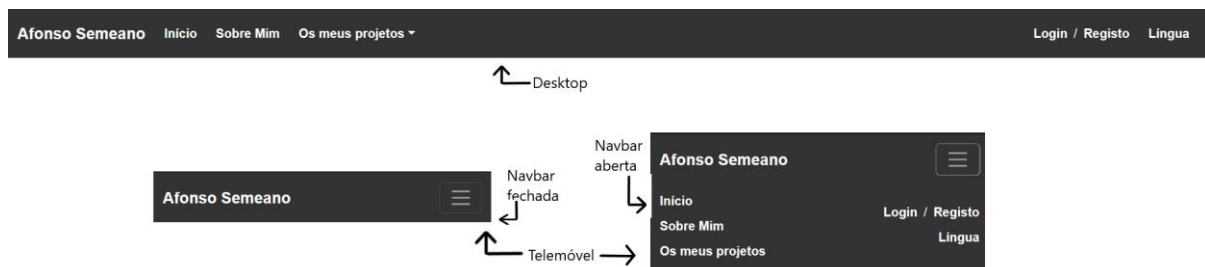
### Como fazer o site responsivo?

Tive de utilizar containers, columns, rows e flex, e evitar estar a utilizar pixeis fixos para determinar os tamanhos dos componentes. Tive também de esconder alguns componentes. Por exemplo, aqui está uma imagem comparativa do footer da página.



*Figura 9 - Comparação do aspeto do footer de Desktop vs Telemóvel*

A navbar também teve de sofrer alterações. Aqui está uma imagem comparativa da navbar.



*Figura 10 - Comparação do aspeto do footer de Desktop vs Telemóvel*

Como pode o site suportar várias línguas e ser também fácil adicionar novas línguas quando necessário?

Para isso, tive de criar um espaço no Mongo para poder armazenar as traduções. Cada tradução tem uma string associada, e dentro de cada string, há a sua versão em português e em inglês. Aqui está um exemplo:

```
_id: ObjectId('631f389e2d3b952dc37fd2b7')
type: "jumbotron"
welcome: Object
  pt: "Bem-vindo ao meu 1º website de portefólio!"
  en: "Welcome to my 1st portefolio website!"
aboutme: Object
  pt: "Sobre mim"
  en: "About me"
myprojects: Object
  pt: "Os meus projetos"
  en: "My Projects"
createaccount: Object
  pt: "Por favor, cria uma conta aqui no site para depois poderes postar revi..."
  en: "Please, create an account in my website so you can post reviews about ..."
meetmeone: Object
  pt: "Aqui podes conhecer quem eu sou na página do "
  en: "Here you can find about me in the page "
meetmetwo: Object
  pt: " e os meus projetos que fiz até à data na página "
  en: " and my projects I've created to date here: "
```

*Figura 11 - Exemplo da tradução da página principal no Mongo*

Depois, sempre que for necessário, fazer fetch das traduções necessárias para a página.

Abaixo está presente um exemplo de código que traduz a página inicial. Preste especial atenção à função `translate`. A função vai buscar a tradução (já obtida) utilizando o texto, que é passado na função, e a língua que é obtida nos parâmetros da URL. Ou seja, se o texto for “welcome” e a língua for português, então irá passar “welcome” e “pt”.

```
function Jumbotron(props) {
  let { lang } = useParams();

  function translate(text) {
    return props.fullTranslation[text][lang];
  }

  return (
    <div className='bg-secondary content'>
      <Container id="home" className="py-5">
        <h1 className="display-5 fw-bold">{translate("hello")}!</h1>
        <p className="fs-4">{translate("welcome")}</p>
        <p className="fs-4">
          {translate("meetmeone")}
          <a href={"/aboutme/" + lang} className='text-grey'>{translate("aboutme")}</a>
          {translate("meetmetwo")}
          <a href={"/myprojects/" + lang} className='text-grey'>{translate("myprojects")}</a>.
        </p>
        <p className="fs-4">{translate("createaccount")}</p>
      </Container>
    </div>
  );
}
```

*Bloco de Código 1 - Exemplo de tradução da página inicial*

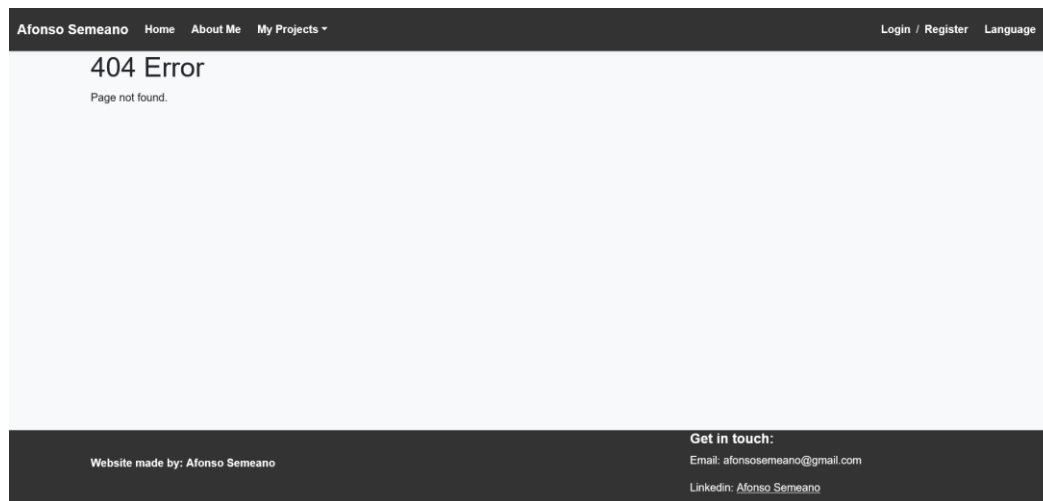
Como ter um routing que não causa erros e que permite associar uma língua e uma página?

Ou seja, um sistema de routing que mostre uma página 404 caso a página ou língua não existam e que permita sempre dois elementos: o nome da página e a língua da página.

Portanto, o routing é composto desta forma: /portfoliowebsite/pt. Primeiro o nome da página, e depois a abreviatura da língua.

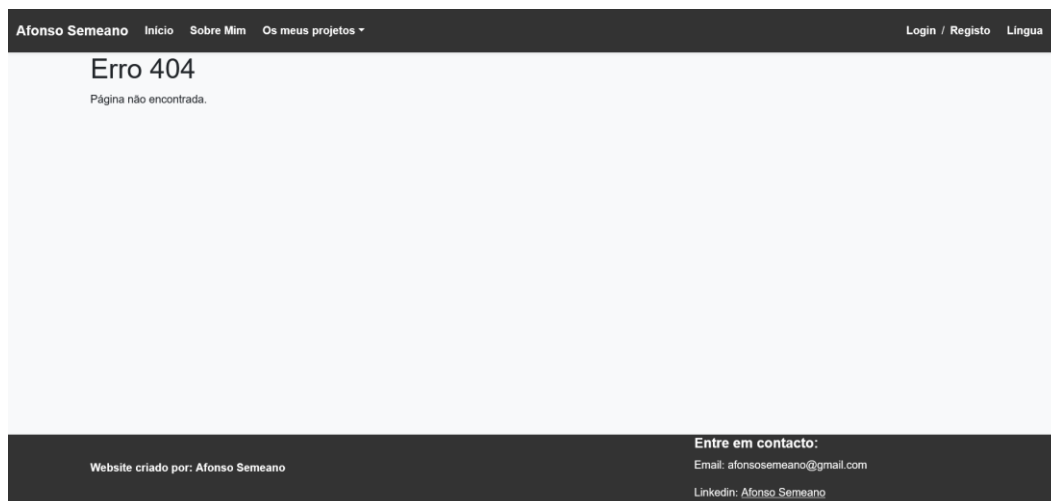
Caso não encontre uma página válida ou uma língua válida, irá retornar um erro 404. Porém, se a língua não for válida, então terá que retornar um erro 404 em inglês, e essa foi a parte desafiante: de como retornar uma página 404 numa língua válida.

Se a língua for inválida, retornará esta página:



*Figura 12 - Exemplo de página 404 caso a língua seja inválida*

Porém, se a página não for válida, mas a língua for válida, então irá retornar um 404 na língua correspondente. Por exemplo:



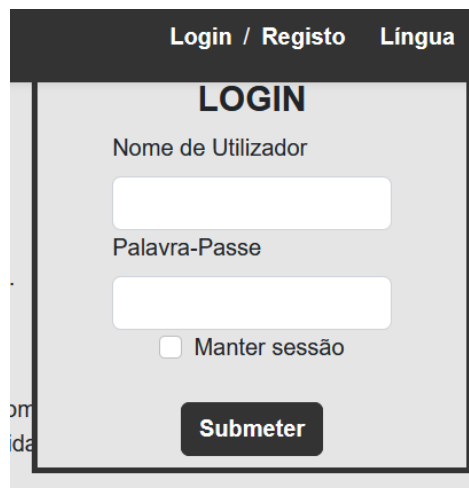
*Figura 13 - Exemplo de página 404 caso a página seja inválida*

### Como ter um website que seja minimamente bonito?

Para isto, tive que fazer alguns testes com conhecidos meus. E no final tinha sempre algumas críticas, nomeadamente relativamente às cores e à posição de certos elementos. Considero que o site melhorou significativamente.

### Como fazer um sistema de login e registo?

Era necessário que houvesse uma encriptação mínima das passwords e que armazenasse essa informação na base de dados. Resolvi o problema da encriptação utilizando **bcrypt**. Já o resto do sistema de login e registo tive de ser eu a criar do zero, e a verificar tudo sozinho, sem usar bibliotecas externas.



The image shows a web form for login. At the top, there is a dark header bar with the text "Login / Registo" and "Língua" in white. Below the header, the word "LOGIN" is centered in bold. The form contains two input fields: "Nome de Utilizador" (Username) and "Palavra-Passe" (Password). Below the password field is a checkbox labeled "Manter sessão" (Keep session). At the bottom of the form is a dark button labeled "Submeter" (Submit).

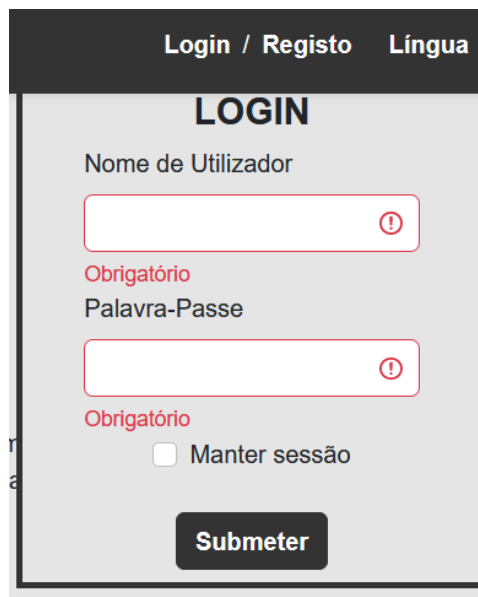
Figura 14 - Caixa de Login



The image shows a web form for registration. At the top, there is a dark header bar with the text "Login / Registo" and "Língua" in white. Below the header, the word "REGISTO" is centered in bold. The form contains two input fields: "Nome de Utilizador" (Username) and "Palavra-Passe" (Password). The username field has a placeholder text "e.g TheRandomPerson". Below the password field is a checkbox labeled "Manter sessão" (Keep session). At the bottom of the form is a dark button labeled "Submeter" (Submit).

Figura 15 - Caixa de Registo

Foi implementado deteção de diversos erros:



The image shows a login form titled "LOGIN" with a header "Login / Registo" and "Língua". The form contains two input fields: "Nome de Utilizador" and "Palavra-Passe". Both fields are empty and have a red error icon (an exclamation mark inside a circle) to their right. Below the "Palavra-Passe" field, there is a red label "Obrigatório" and a checkbox labeled "Manter sessão". At the bottom of the form is a black button labeled "Submeter".

Figura 16 - Campos obrigatórios no Login



The image shows the same login form as in Figure 16, but with the "Nome de Utilizador" field filled with the text "sasdddsa" and the "Palavra-Passe" field filled with six dots. The "Manter sessão" checkbox is still present. Below the input fields, there is a red error message: "Nome de utilizador não encontrado, por favor registe-se." At the bottom of the form is a black button labeled "Submeter".

Figura 17 - Nome de utilizador não encontrado no Login

Figura 18 - Campos obrigatórios no Registo

Figura 19 - Nome de utilizador já existe no Registo

Todas as passwords são encriptadas. Aqui está um comprovativo:

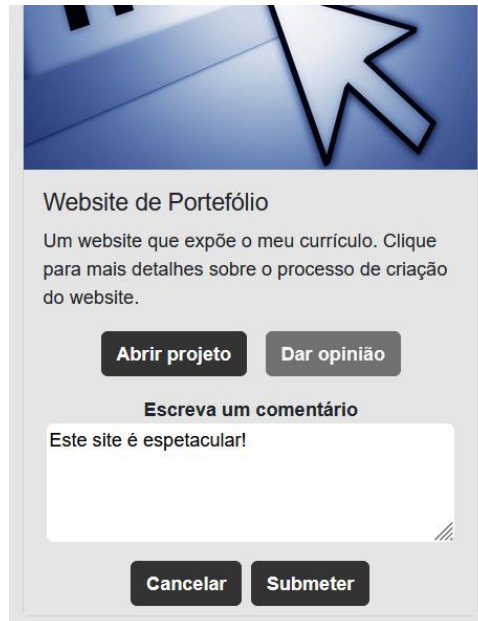
```
_id: ObjectId('6362c2ebaf78f1d06cc914b6')
username: "FakeUse"
passwordEnc: "$2b$12$NdTmokoytROXDaMbmK5qB04VN6hf6aZVBsOk/S1hXw.B4Bx.0HvIS"

_id: ObjectId('6362d950af78f1d06cc914ba')
username: "NewUser"
passwordEnc: "$2b$12$b3uJGLyWDCRHEZbc0CZ4uzjJKgJgp8LLhTlGRrB6rSZ0B9StDu8m"
```

Figura 20 - Comprovativo de encriptação das passwords

## Sistema de feedback

Criei um sistema de feedback bastante simples em que, cada janela da página de projetos tem a possibilidade de o utilizador escrever um feedback sobre o projeto específico. Aqui está um exemplo:



The screenshot shows a feedback form for a project titled "Website de Portefólio". The form includes a description of the project, two buttons ("Abrir projeto" and "Dar opinião"), a text input field for a comment, and two buttons ("Cancelar" and "Submeter"). The comment field contains the text "Este site é espetacular!".

Website de Portefólio

Um website que expõe o meu currículo. Clique para mais detalhes sobre o processo de criação do website.

**Abrir projeto** **Dar opinião**

**Escreva um comentário**

Este site é espetacular!

**Cancelar** **Submeter**

*Figura 21 - Janela de feedback do projeto*

Quando o utilizador submete um feedback, esse feedback é enviado para o Mongo, e depois eu visualizo o feedback na base de dados.

Também coloquei algumas verificações. Exemplos:



The screenshot shows the same feedback form as Figure 21, but with a validation message displayed below the comment field: "Por favor, escreva algo para dar a sua opinião".

Website de Portefólio

Um website que expõe o meu currículo. Clique para mais detalhes sobre o processo de criação do website.

**Abrir projeto** **Dar opinião**

**Escreva um comentário**

Por favor, escreva algo para dar a sua opinião

**Cancelar** **Submeter**

*Figura 22 - Verificação de se o utilizador escreveu algo*



**Website de Portefólio**

Um website que expõe o meu currículo. Clique para mais detalhes sobre o processo de criação do website.

**Abrir projeto** **Dar opinião**

**Escreva um comentário**

sadasd

Por favor, faça login para dar a sua opinião

**Cancelar** **Submeter**

*Figura 23 - Verificação de se o utilizador está logado*



## Conclusão

Depois de ter criado o site e de estar quase a terminar o curso de Engenharia Informática, eu posso dizer que conseguiria fazer este site bastante mais rapidamente e de forma mais eficiente se o fosse fazer uma segunda vez... isto porque já tenho conhecimento das tecnologias utilizadas e dos princípios de criação de websites com Engenharia de Software Aplicada.

Assim, posso dizer que o tempo que demorei a realizar este site é de cerca de 100h.

Este tempo inclui:

- Estudo das tecnologias
- Tempo gasto a programar
- Tempo gasto a resolver bugs (que eu sei muito bem que fiquei preso durante horas em certos bugs)

Acho que é uma estimativa bastante justa para um site desta dimensão e para alguém com a minha experiência.

A nota que dou a mim mesmo de 0 a 5 é de 4. E dou um 4 porque creio que a interface gráfica poderia ter sido melhorada... falta adicionar mais animações, melhorar a formatação do texto.

Assinatura do responsável pela validação:

X

---

Joaquim Filipe  
Professor