**IPV – Instituto Politécnico de Viseu**

**ESTGV – Escola Superior de Tecnologia e Gestão de Viseu**

**Departamento de Informática**

****

**Relatório de Aplicações para a Internet**

**Licenciatura em Engenharia Informática**

**Realizado**

**por**

**Artur Santos pv20251**

**Luís Anselmo pv20245**

**Alexandre Moreira pv20223**

**Orientador**

**ESTGV: Steven Abrantes**

**Viseu, 2021**

**IPV – Instituto Politécnico de Viseu**

**ESTGV – Escola Superior de Tecnologia e Gestão de Viseu**

**Departamento de Informática**

**Relatório de Aplicações para a Internet**

**Licenciatura em Engenharia Informática**

**Realizado**

**por**

**Artur Santos pv20251**

**Luís Anselmo pv20245**

**Alexandre Moreira pv20223**

**Orientadores**

**ESTGV: Steven Abrantes**

**Viseu, 2021**

Índice

[1. Introdução 4](#_Toc94608880)

[2. HTML 5](#_Toc94608881)

[2.1. <head> 5](#_Toc94608882)

[2.2. <header> 5](#_Toc94608883)

[2.3. <main> 6](#_Toc94608884)

[2.4. <footer> 8](#_Toc94608885)

[2.5. Overlay 9](#_Toc94608886)

[2.6. Scripts Finais 9](#_Toc94608887)

[3. JavaScript 10](#_Toc94608888)

[4. React 14](#_Toc94608889)

[5. Conclusão 18](#_Toc94608890)

[6. Documentação 18](#_Toc94608891)

# 1. Introdução

Neste trabalho prático iremos demonstrar tudo oque aprendemos ao longo do semestre nesta cadeira.

Neste trabalho pretende-se criar um site a nossa escolha utilizando os diversos conteúdos que nos foram lecionados ao longo do semestre. O nosso grupo decidiu que iriamos fazer um portefólio sobre todos os elementos do grupo

Neste trabalho, o utilizador pode ver o portefólio assim como algumas breves informações sobre os elementos do grupo e também comunicar com os mesmos.

# 2. HTML

# 2.1. <head>

Ao longo do programa todas as páginas escritas em html têm o mesmo *<head>*, onde chamamos o ficheiro relativo a CSS e definimos o nome da página.

# 2.2. <header>

Assim como no *<head>*, o *<header>* é igual para todas as páginas apenas mudando o link da própria página no menu de navegação. Na imagem a cima podemos ver que ao carregar o body, o programa automaticamente inicializa a função ‘*startTime*’ permitindo carregar imediatamente a hora local, no seguimento de inserir um relógio que permanecerá no canto superior da página.

# 2.3. <main>

Text

Description automatically generated

Na imagem a cima podemos ver o conteúdo existente no *<main>*  na pagina inicial do site. Neste ponto podemos observar a existência de ‘*sliders*’, quer para imagens quer para a informação relativa aos criadores. Para alem dos ‘*sliders*’ referidos, temos também alguns icons que permitem o utilizador visitar alguns sites relacionados com a página e os seus criadores.

Text

Description automatically generated

No caso da página de contactos, todo o código desenvolvido esta presente no ficheiro relativo a ‘react’ (react.jsx)

Text

Description automatically generated

Finalmente na última página, ‘*about*’, o utilizador pode ver mais alguns detalhes os criadores do site.

# 2.4. <footer>

Text

Description automatically generated

Como pode ser visto na imagem apresentada, o ‘*footer*’ apresenta o nome dos criadores do site assim como as redes sociais dos mesmos. A rotação dos utilizadores apresentados esta diretamente ligada aos ‘*sliders*’ anteriores.

# 2.5. Overlay

Text

Description automatically generated

Com ajuda de alguma pesquisa e com o objetivo de deixar a página inicial mais apelativa, ao iniciar o site o utilizador será recebido por alguns *overlays* animados.

# 2.6. Scripts Finais



Finalmente, no fim de todas as páginas html são apresentados diversos scripts que relativos à componente JavaScript e React, assim com utilização de animações (GSAP) e de icons (ionicons)

# 3. JavaScript

Text

Description automatically generated

A função apresentada em cima é relativa ao menu de navegação. Quando o utilizador visualiza o website num dispositivo mais pequeno, o menu da página é substituído por um “hamburger menu”.

Text

Description automatically generated

A função mostrada na figura acima é a função responsável pelo relógio que é carregada no body de cada página (referido no header).

Text

Description automatically generated

Todas as funções apresentadas acima são responsáveis pelo funcionamento dos *slideshows* das imagens assim como do texto da home page e o footer.

Text

Description automatically generated

Text

Description automatically generated

JavaScrip associado às animações (GSAP).

Ver mais informações no final do relatório em “documentação”

# 4. React

A screenshot of a computer

Description automatically generated with medium confidence

A partir do inicio do programa podemos ver que iniciamos um criador de elemento, assim como a class principal no caso do nosso programa. Apos definirmos a class, inicializamos o construtor onde definimos os *Hoocks* que nos permitem usar *state* sem necessitar de escrever outra class. Em seguimento definimos as variáveis que vamos utilizar para guardar os dados sobre o contacto.

Text

Description automatically generated

De seguida, começamos a nossa primeira e única função *submitFunc*. Nesta função definimos uma lista com todas as variáveis que iremos utilizar, assim como a alocação dessa mesma lista no armazenamento local.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

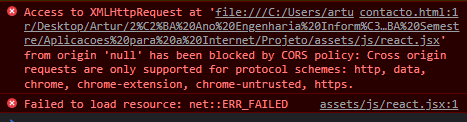
Nas imagens seguintes podemos ver os diversos *inputs* que têm como função registar e guardar os dados que o utilizador coloca na área de contacto assim como o botão final que permite ao utilizador submeter o formulário.

A screenshot of a computer

Description automatically generated with medium confidence

Nesta imagem final, podemos ver por fim o construtor que faz com que todo o código presente em react possa ser renderizado na nossa página *HTML.*

**Nota:** Devido a utilização de um ficheiro **.jsx** local, o react apenas funcionará se o site estiver diretamente ligado a um servidor online (use por exemplo a extensão “Live Server” do Visual Studio Code)



Após alguma pesquisa, verificámos que este erro é bastante comum quando se está na fase de desenvolvimento de um website em ambiente local. O facto de estarmos a fazer um “request” da “data” de um ficheiro externo JavaScript, no caso, o formulário, faz ativar este erro, basicamente bloqueando o acesso ao ficheiro react.jsx.

Como solução, o website deve ser acedido através de um servidor (http request). Use por exemplo a extensão “Live Server” do Visual Studio Code.

De qualquer maneira, o Website está perfeitamente operacional dentro das normas.

Ver mais informação em “documentação”.

# 5. Conclusão

Após diversas horas de trabalho e pesquisa, conseguimos realizar todas as funções que nos foram pedidas.

Este programa, apesar do grau relativamente alto de conhecimento necessário para o desenvolver, acabou por ser um trabalho interessante de realizar, obrigando-nos a utilizar algumas funcionalidades ainda por explorar assim como pensar de maneiras inovadoras.

Este projeto serviu essencialmente para enriquecer o nosso conhecimento a nível de *CSS* mas sobretudo nível de *React* e *JavaScript*, sendo assim um trabalho que nos possibilitou ter uma visão mais ampla relativamente ao processo de desenvolvimento Web.

# 6. Documentação

* <https://reactjs.org/docs/forms.html>
* <https://greensock.com/gsap/>
* <https://ionic.io/ionicons>
* <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/CORS>