

Politécnico do Porto  
Escola Superior de Media Artes e Design

Alexandre Condessa da Silva Costa

Estágio Helppier – Desenvolvimento web (ReactJs)

Licenciatura em Tecnologias e Sistemas de Informação para a Web

Projeto Final / Estágio

Prof. Doutor Lino Oliveira

Vila do Conde, julho de 2019

## **Agradecimentos**

Quero agradecer a todos aos quais tive colaboração para a concretização deste projeto.

Primeiramente estou grato à própria empresa por me ter acolhido.

Seguidamente aos professores pelo apoio e o trabalho que desempenharam.

Não posso deixar de agradecer aos meus amigos que tanto me apoiaram.

E finalmente um obrigado à minha família, mais especificamente aos meus pais, por sempre me apoiarem e darem-me a liberdade de ter seguido este percurso mesmo estando longe de casa.

## Resumo

Neste relatório é descrito o processo do projeto final/estágio. Decorreu na empresa Help-pier de dia 4 de março a dia 28 de junho de 2019 no âmbito da conclusão da Licenciatura em Tecnologias e Sistemas de Informação para Web, frequentado na Escola Superior de Media Artes e Design do Politécnico do Porto. É descrito não só o desenvolvimento do projeto, mas também a contextualização, a fase do pré desenvolvimento, entre outros elementos que descrevem o decorrer do estágio.

A empresa Help-pier é uma empresa que comercializa e desenvolve uma plataforma com o próprio nome. Esta plataforma consiste em criar guias de ajuda e ferramentas interativas para o website do cliente no próprio website. Durante o estágio o estagiário trabalhou juntamente com o resto da equipa de desenvolvimento da empresa.

Para este projeto final/estágio foram realizadas tarefas que serão usadas em futuras funcionalidades da aplicação Help-pier, a aplicação comercializada pela empresa. As tarefas dividiram-se em duas fases: reconstrução de componentes genéricos e o desenvolvimento de novas funcionalidades na aplicação.

A tarefa de reconstrução de componentes genéricos consistia em recriar botões, menus e outros elementos de maneira a que estes sejam reutilizáveis ao longo da aplicação, analisando todos os propósitos possíveis que cada um poderá ter.

No desenvolvimento de novas ferramentas foram realizados vários ecrãs que permitem criar testes dentro de uma Suite, ver os testes de cada uma, editá-los e eliminá-los. Estes ecrãs foram desenvolvidos de maneira a manter a estrutura dos restantes ecrãs e utilizando os componentes genéricos criados anteriormente. Os testes são guardados, editados e removidos na própria base de dados.

## **Abstract**

This report describes the final project / internship process. Held on the company Help-pier from March 4 to June 28, 2019 in the ambit of the conclusion of the Degree in Technologies and Information Systems for Web, attended at Escola Superior de Media Artes e Design (Superior School of Media Arts and Design) of the Polytechnic of Porto. It describes not only the development of the project, but also the contextuality, the pre-development phase, among other elements that describe the course of the internship.

The company Help-pier is a company that sells a platform with its own name. This platform consists on creating help guides and interactive tools for the client's website on the website itself, allowing communication with the user of the website without the user having to leave it. During the internship, the trainee worked alongside the rest of the company's development team. The tasks of the team are to develop new tools and maintain existing platform.

For this final project / internship, there were accomplished tasks that will be used in future functionalities of the application Help-pier, the application marketed by the company. The tasks were divided into two phases: the reconstruction of generic components and the development of new functionalities in the application.

The task of reconstructing generic components was to recreate buttons, menus and other elements so that they are reusable throughout the application, analysing all the possible purposes that each one may have.

In the development of new tools were realized several screens that allow to create tests within a Suite, to see the tests of each one, to edit them and to eliminate them. These screens have been developed in order to maintain the structure of the remaining screens and using the previously created generic components. The tests are saved, edited and removed in the database itself.

## Sumário

Agradecimentos.....	2
Resumo.....	3
Abstract .....	4
Sumário .....	5
Lista de ilustrações .....	6
Glossário .....	7
1. Introdução .....	9
1.1. Enquadramento.....	9
1.2. Entidade acolhedora.....	9
1.3. Objetivos do projeto .....	10
1.4. Estrutura do documento .....	10
2. Contextualização .....	11
2.1. Plataforma Helpier.....	11
2.2. Plataformas semelhantes / concorrentes .....	12
3. Gestão do trabalho.....	14
3.1. Metodologia.....	14
3.2. Planeamento e organização do trabalho .....	15
3.3. Requisitos.....	17
3.3.1. Requisitos funcionais .....	17
3.3.2. Requisitos não funcionais .....	18
3.4. Ferramentas e Tecnologias.....	18
4. Projeto.....	24
4.1. Descrição.....	24
4.2. Implementação.....	25
Conclusão.....	30

## Lista de ilustrações

Figura 1 – Logótipo da Helppier

Figura 2 – Logótipo da WalkMe

Figura 3 – Logótipo da StepShot

Figura 4 – Logótipo da ScreenSteps

Figura 5 – Organização das tarefas no Jira

Figura 6 – Planeamento definido

Figura 7 – Logótipo do Visual Studio Code

Figura 8 – Logótipo do GitKraken

Figura 9 – Logótipo do Robo3T

Figura 10 – Logótipo do Jira

Figura 11 - Logótipo do JavaScript

Figura 12 - Logótipo do NodeJS

Figura 13 - Logótipo do React

Figura 14 - Logótipo do Redux

Figura 15 - Logótipo do Meteor

Figura 16 - Logótipo do MongoDB

Figura 17 - Logótipo do Storybook

Figura 18 - Logótipo do Emotion

Figura 19 - Logótipo do HTML

Figura 20 – Componente de botão genérico

Figura 21 – Componente de menu genérico

Figura 22 – Componente de barra de navegação com o menu

Figura 23 – componentes genéricos de ferramentas

Figura 24 – Conjunto de componentes genéricos para criar uma página

Figura 25 – Ecrã de lista de testes

Figura 26 – Ecrã de criação de testes

Figura 27 – Diagrama de casos de uso

Figura 28 – Estrutura da coleção de testes

Figura 29 – Diagrama de sequência (representa a criação de um teste)

Figura 30 – Diagrama de sequência (representa a atualização de um teste / editar teste)

## Glossário

- 1 **Base de dados** \_ Ferramenta de recolha e organização de informações. Contem um conjunto de informações necessárias para um software, website ou aplicação web.
- 2 **Framework** \_ Abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. Dependem do propósito e podem existir diversos frameworks diferentes para a mesma linguagem de programação, assim como existem frameworks para diferentes linguagens de programação.
- 3 **Suite de testes** \_ Coleção de testes intencionados para testar algo num software.
- 4 **Full stack** \_ Desenvolver a nível full stack quer dizer desenvolver o lado do cliente (*front-end*) e o lado do servidor (*back-end*) simultaneamente.
- 5 **Browser** \_ Um browser web é um software que permite aceder a informação na web.





## 1. Introdução

Neste capítulo é introduzido o enquadramento do projeto, a entidade acolhedora, os objetivos do projeto e a estrutura do documento.

### 1.1. Enquadramento

Este relatório vem conciliar o trabalho realizado na unidade curricular de Projeto Final / Estágio da Licenciatura em Tecnologias e Sistemas de Informação para a Web, decorrido na Escola Superior de Media Artes e Design do Politécnico do Porto. Esta unidade curricular decorreu de dia 4 de março a dia 28 de junho de 2019.

O tutor de estágio foi Marco Garcia, gestor dos desenvolvedores na empresa e a orientação por parte do professor doutor Lino Oliveira.

### 1.2. Entidade acolhedora

O estágio foi realizado na empresa Helppier, localizada no Parque de Ciências e Tecnologias da universidade do Porto (UPTEC). É uma empresa *Start-Up* (uma pequena empresa ainda no período inicial) e foi fundada em 2014.

A empresa desenvolve uma aplicação com o mesmo nome. A aplicação consiste em criar guias de ajuda e ferramentas interativas para o website do cliente, permitindo uma comunicação com o utilizador do mesmo.

### **1.3. Objetivos do projeto**

O trabalho a elaborar consiste no desenvolvimento da aplicação da empresa juntamente com os outros membros da equipa. Tem como objetivo recriar componentes genéricos que possam vir a ser usados na aplicação e a criação de novos ecrãs a nível full stack. No final da execução das tarefas o estagiário deve ter capacidade para usar ferramentas como JavaScript, ReactJS, Redux, Meteor e MongoDB.

O estagiário na empresa tem sempre o apoio e formação necessária, porem parte do mesmo desenvolver os seus conhecimentos nas ferramentas a utilizar e realizar toda a pesquisa necessária.

### **1.4. Estrutura do documento**

No capítulo 1 – Introdução é introduzido o tema do relatório

No capítulo 2 – Contextualização é abordado o estado da arte, a plataforma Helpier e plataformas semelhantes;

No capítulo 3 – Metodologia de trabalho é referido o planeamento e a organização de trabalho, os requisitos e as ferramentas utilizadas;

No capítulo 4 - Projeto é descrito o projeto e a sua implementação;

No capítulo 5 – Conclusão são descritos os principais aspetos positivos e negativos de todo o projeto, assim como a apreciação final de todo o trabalho desenvolvido e conhecimentos adquiridos.

## 2. Contextualização

A criação de guias de utilizador e instruções de passo a passo em websites e aplicações ajuda o utilizador do produto a entender não só o seu funcionamento como a mensagem que se pretende transmitir.

Existem diversos produtos que permitem criar guias para utilizadores e instruções de passo a passo.

### 2.1. Plataforma Helppier



Figura 1 – Logótipo da Helppier

A aplicação Helppier consiste em criar guias de ajuda e ferramentas interativas para o website do cliente no próprio website, permitindo uma comunicação com o utilizador do mesmo. Contem ainda outras ferramentas como estatísticas das guias, gravação de ecrã, etc. Tudo isto sem necessitar de conhecimentos de programação e sem sair da própria página, o que facilita a utilização da aplicação e também a interpretação dos tutoriais, visto que não obriga à abertura de várias janelas.

Algumas características chave da Helppier é que suporta qualquer *browser* ou dispositivo, é traduzido em 65 linguas, é fácil de utilizar devido ao seu *design* intuitivo, sem necessitar que o utilizador tenha conhecimentos de desenvolvimento web e contem diversas ferramentas que permitem analisar os guias, gerar vídeos dos tutoriais, entre outras.

Mais sobre a aplicação Helppier: <https://www.helppier.com/pt/>

## 2.2. Plataformas semelhantes / concorrentes

### 2.2.1. WalkMe



Figura 2 – Logótipo da WalkMe

WalkMe é uma plataforma que possibilita criar guias passo a passo no próprio website através de balões que apresentam dicas. Isso ajuda a que o utilizador execute as tarefas desejadas sem ter de sair do próprio site.

Ao contrário da Helppier, esta aplicação é mais destinada a empresas ou pessoas que tenham conhecimento em desenvolvimento de aplicações. Sendo que fornece diversas funcionalidades pode não ser entendida por todos e pode demorar algum tempo para se habituar ao software. Para além disso apenas suporta os *browsers* Firefox e Chrome, o que pode ser uma desvantagem para outros utilizadores.

Mais sobre a WalkMe: <https://www.walkme.com/>

### 2.2.2. StepShot



Figura 3 – Logótipo da StepShot

StepShot é uma ferramenta que permite criar manuais passo a passo detalhados e guias de utilizador. Permite também exportar documentos, anotar descrições em imagens, entre outras funcionalidades.

Usar StepShot garante a criação de documentação de software muito mais rápida e eficaz de que a criação de raiz.

Esta ferramenta é distinta à aplicação Helppier, pois cria documentação e manuais de utilização, em vez de mostrar tutoriais diretamente na página web.

Mais sobre StepShot: <https://stepshot.net/>

### 2.2.3. ScreenSteps



*Figura 4 – Logótipo da ScreenSteps*

ScreenSteps é um programa que permite capturar a imagem de cada passo que está envolvido no processo que se está a tentar demonstrar. Permite criar automaticamente um documento com as capturas de ecrã, exportar para vários formatos, entre outras características.

Semelhante à ferramenta Stepshot, ScreenSteps é uma maneira diferente de criar guias de utilizadores.

Mais sobre ScreenSteps: <https://www.screensteps.com/>

### 3 Gestão do trabalho

#### 3.1. Metodologia

Para a gestão e planeamento do projeto a empresa baseia-se na metodologia *Scrum*, uma metodologia de desenvolvimento de software. Nesta metodologia os projetos são divididos em ciclos chamados Sprints. Cada Sprint representa um conjunto de atividades a serem realizadas no espaço de tempo dado.

Neste caso o desenvolvimento do projeto é dividido em Sprints de 15 dias cada. No final de cada Sprint devem ser terminadas as tarefas planeadas a desenvolver durante a mesma. De seguida são realizadas breves reuniões que têm como propósito cada elemento da equipa apresentar as tarefas desenvolvidas e as novas funcionalidades implementadas durante a mesma, sendo também planeadas as seguintes tarefas a serem realizadas e a distribuição das mesmas. Para além disso é feito no início de cada dia uma Daily Scrum ou Stand Up que se baseia numa curta reunião onde a equipa se junta e cada um fala sobre as tarefas que está a realizar.

Desta maneira os elementos da equipa de desenvolvimento conseguem estar a par do trabalho que está a ser realizado pelo resto da equipa, podendo as pessoas ajudarem-se entre si facilmente.

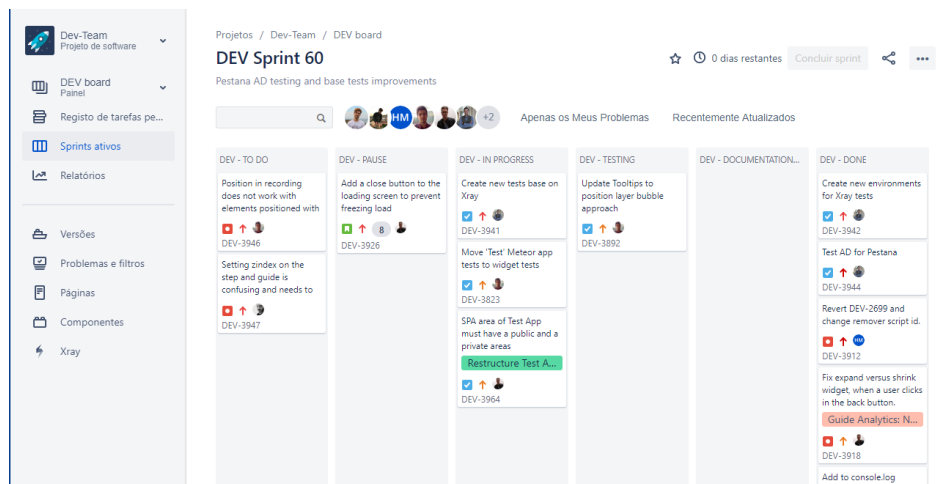


Figura 5 – Organização das tarefas utilizando a ferramenta Jira

### 3.2. Planeamento e organização do trabalho

No início do estágio foi definido um planeamento resumido que abrangesse as atividades a serem desenvolvidas de forma a que fosse possível acompanhar o trabalho do resto da equipa. Segue-se o conjunto de tarefas inicialmente enviado:

- Pesquisa das várias ferramentas utilizadas (continuo);
- de tutoriais e leitura de documentação sobre as ferramentas usadas no desenvolvimento do projeto (3 semanas);
- Análise da estrutura do projeto e do produto widget Helpfier. (1 semana);
- Estudo e compreensão do funcionamento da ferramenta Storybook. (1 semana);
- Conversão de componentes para o Storybook utilizando novas abordagens com CSS em JS utilizando Emotion (tarefa longa);
- Criação de componentes de interface em ReactJS. (tarefa longa);
- Melhorar a aplicação de Storybook com componentes ReactJS (tarefa continua);

Segue-se o planeamento definido e depois de ajustado:

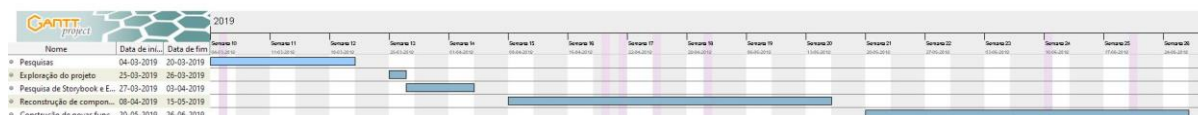


Figura 6 – Planeamento definido

Tarefas	Data de início	Data de fim
<b>Pesquisas</b>	04-03-2019	20-03-2019
Pesquisa e realização de tutoriais e leitura de documentação sobre as ferramentas utilizadas: Javascript ES6, Reactjs, Redux, entre outros;		
<b>Exploração do projeto</b>	25-03-2019	26-03-2019
Instalação e exploração do projeto: aplicação Helpier _ experimentação com <i>debugging</i> (depuração) no browser usando o DevTools (ferramenta que já vem com o browser que permite explorar o código) e exploração da estrutura do código realizado;		
<b>Pesquisa de Storybook e Emotion</b>	27-03-2019	03-04-2019
Pesquisa sobre as ferramentas Storybook e Emotion a vir utilizar na tarefa seguinte _ leitura de documentação, realização de tutoriais e adaptação do código para o projeto;		
<b>Reconstrução de componentes genéricos</b>	08-04-2019	15-05-2019
Aplicação dos conhecimentos de React a reconstruir componentes genéricos a virem ser usados no futuro na aplicação através da ferramenta Storybook e estilizado utilizando Emotion. Parte destes componentes serão realizados através de materiais RMWC (componentes como botões e menus pré feitos). Os componentes devem ser construídos de maneira a que aceitem lógica independente dos mesmos e que prevejam os seus futuros objetivos na aplicação;		
<b>Construção de novas funcionalidades</b>	20-05-2019	26-06-2019
Nesta tarefa serão desenvolvidos novos ecrãs para uma nova funcionalidade na aplicação, de maneira a reutilizar os componentes criados e manter o aspeto da aplicação. Será necessário utilizar React, Meteor, Redux e JavaScript para criar a lógica desses ecrãs, guardar documentos na base de dados, entre outras necessidades.		



### 3.3. Requisitos

Os requisitos foram recolhidos de acordo com as tarefas atribuídas.

#### 3.3.1. Requisitos funcionais

Segue-se os requisitos funcionais para o projeto desenvolvido. Estas são necessidades do projeto e definem as suas funcionalidades.

- Os componentes genéricos desenvolvidos devem ser dinâmicos e servir os propósitos previstos para o mesmo, mantendo a estrutura e de maneira a que sejam facilmente aplicados por outros membros da equipa;
- Os componentes devem ser estilizados de forma igual através de temas (componentes que distribuem as cores escolhidas pelos componentes escolhidos), podendo estes serem alterados;
- Os componentes não devem depender de lógica nem vice-versa, isto é, as suas funções não devem estar definidas diretamente nos componentes. Em vez disso os componentes devem estar prontos para receberem essas funções;
- Os ecrãs desenvolvidos devem manter a estrutura dos restantes ecrãs da aplicação;
- No ecrã de criação de testes deve permitir que o utilizador insira o nome do teste (obrigatoriamente), a descrição e a guia a ser testada;
- O ecrã de criação de testes deve poder ser acedido através do ecrã onde são mostradas as Suites, através de um menu;
- Os testes devem ser guardados na base de dados com os dados inseridos e ainda a Suite do mesmo, o site e a empresa a utilizar a aplicação, a data de criação e as definições do mesmo;
- Ao guardar o teste deve ser mostrado o ecrã com a lista de testes da mesma Suite onde este foi criado e com um botão para criar um teste nessa Suite;
- Ao clicar na Suite deve ser mostrado um ecrã com uma lista dos testes pertencentes à Suite selecionada. Este ecrã será o mesmo referido no ponto anterior;
- Cada teste deve conter um menu que permita o utilizador editar o teste e eliminá-lo
- Ao editar o teste deve mostrar o ecrã do novo teste, mas com as características atuais já escritas por defeito;

### 3.3.2. Requisitos não funcionais

Segue-se os requisitos não funcionais para o projeto desenvolvido. Estes requisitos estão relacionados a como as funcionalidades são entregues ao usuário.

- A estilização dos componentes deve ser mostrada de forma igual em qualquer browser ou dispositivo;
- As interfaces criadas devem manter a estrutura independentemente do browser e dispositivos onde está a ser corrido;
- Na utilização da aplicação esta deve estar conectada à sua base de dados;

### 3.4. Ferramentas e Tecnologias

Seguem-se as ferramentas/softwarees utilizadas na empresa:

**Visual Studio Code** é um editor de código redefinido e otimizado para construir aplicações web modernas. Suporta diversas linguagens de programação e ainda contem extensões que permitem ajudar o utilizador em linguagens ou *frameworks* específicas ou em outras situações. Neste caso foi usada a extensão **Git Lens**, que mostra quem escreveu cada ficheiro, há quanto tempo foi escrito e as mudanças que estão a ser feitas, sendo que todo o projeto está interligado com a ferramenta GitKraken.

Mais sobre Visual Studio Code: <https://code.visualstudio.com/>

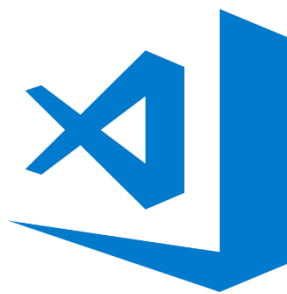


Figura 7 – Logótipo do Visual Studio Code

**GitKraken** é um software que ajuda no desenvolvimento de uma aplicação quando existem vários desenvolvedores, pois permite que cada um tenha os seus projetos, podendo alterá-los sem afetar a aplicação em si, trocar projetos com outros desenvolvedores e publicar o projeto para o produto final. Tudo de uma maneira fácil e intuitiva.

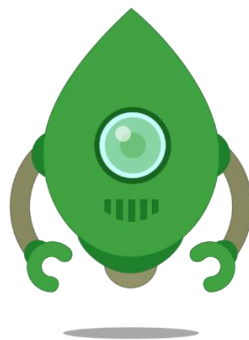
Mais sobre GitKraken: <https://www.gitkraken.com/>



*Figura 8 – Logótipo do GitKraken*

**Robo 3T** é um software que assiste no desenvolvimento de base de dados em MongoDB. Permite facilmente ver e alterar bases de dados de forma dinâmica, assim como realizar *queries* diretamente na base de dados e outras funcionalidades.

Mais sobre Robo 3T: <https://robomongo.org/>



*Figura 9 – Logótipo do Robo3T*

**Jira** é uma ferramenta online que gere a organização e monitorização das tarefas entre a equipa de desenvolvedores, acompanhando o projeto desenvolvido.

Mais sobre Jira: <https://www.atlassian.com/software/jira>



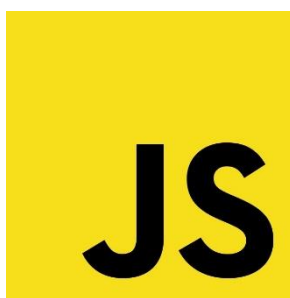
*Figura 10 – Logótipo do Jira*

De seguida são descritas as tecnologias utilizadas no decorrer do estágio:

**JavaScript** é uma linguagem de programação orientada a objetos que permite implementar lógica do lado cliente. Esta linguagem permite manipular comportamentos em páginas HTML e CSS sem que o código precise de passar pelo lado do servidor, sendo assim uma das principais linguagens no desenvolvimento de aplicações web.

Esta é a linguagem mais utilizada, pois é usando a mesma que a lógica de toda a aplicação é implementada.

Mais sobre JavaScript: <https://www.javascript.com/>



*Figura 11 - Logótipo do JavaScript*

**HTML 5** (a versão 5 de *Hyper Text Markup Language* / Linguagem de Marcação de Hipertexto) é a linguagem de marcação utilizada na construção de páginas web. Permite adicionar texto, imagens, formulários e outros elementos básicos usados páginas web, sendo a que contem o conteúdo principal da página.

Mais sobre HTML5:

<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>



*Figura 12 – Logótipo HTML*

**NodeJS** é uma plataforma que interpreta código JavaScript no lado do servidor e ajuda na criação de aplicações com servidores web.

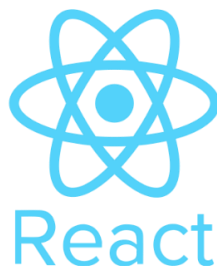
Mais sobre NodeJS: <https://nodejs.org/en/>



*Figura 13 – Logótipo de NodeJS*

**React** (ou ReactJS) é uma biblioteca JavaScript com o propósito de criar interfaces de utilizador, possibilitando que estas sejam aplicativos de página única, isto é, uma aplicação web ou website que rescreve a página dinamicamente interagindo com o utilizador sem necessitar o recarregamento da própria página.

Mais sobre ReactJS: <https://reactjs.org/>



*Figura 14 – Logótipo do React*

**Redux** é uma biblioteca JavaScript de código-aberto que permite gerir estados da aplicação, tipicamente usado juntamente com React ou bibliotecas semelhantes para construir interfaces de utilizadores.

Mais sobre Redux: <https://redux.js.org/>



Figura 15 - Logótipo do Redux

**Meteor** é uma ferramenta *framework open-source* de *javascript* que facilita a programação em *full stack* e para vários dispositivos. Integra MongoDB. No lado do cliente Meteor podem ser usadas outras *frameworks* como React.

Mais sobre Meteor: <https://www.meteor.com/>



Figura 16 - Logótipo do Meteor

**MongoDB** é uma ferramenta de base de dados orientados a documentos livres, de código aberto e multiplataforma. Sendo orientado para base de dados não relacionais (orientados a documentos livres) permite uma maior escalabilidade, fácil manipulação (visto que os dados são independentes), entre outras vantagens, ao contrário de bases de dados relacionais.

Mais sobre MongoDB: <https://www.mongodb.com/>



Figura 17 - Logótipo do MongoDB

**Storybook** é uma ferramenta de open source que permite desenvolver componentes de interfaces de utilizador isoladamente. O seu propósito é ajudar na criação de componentes sem que estes tenham de depender de nada.

Mais sobre Storybook: <https://storybook.js.org/>



*Figura 18 - Logótipo do Storybook*

**Emotion** é uma ferramenta que permite desenvolver estilizações CSS em JavaScript, possibilitando atribuir valores a variáveis e facilitando a estilização dos componentes a desenvolver.

Mais sobre Emotion: <https://emotion.sh/docs/introduction>



*Figura 19 - Logótipo do Emotion*

## 4. Projeto

### 4.1. Descrição

O estágio dividiu-se em duas tarefas distintas: A recriação de componentes genéricos e a construção de novas páginas na aplicação.

O objetivo desta primeira tarefa é recriar os componentes (utilizando materiais de Interface de Utilizador) de maneira a que esses mesmos componentes (botões, menus, páginas, etc.) sejam dinâmicos e tenham vários comportamentos que dependam de cada situação. Por exemplo, o componente do botão é chamado várias vezes, mas este tem títulos diferentes e ações distintas dependendo de onde aparece. Apesar disso o botão é o mesmo componente. É por isso que é importante analisar todas as situações em que cada componente pode vir a ser utilizado. No final da tarefa deve ser fácil para os outros membros da equipa entenderem o componente e adaptarem-no para o propósito que pretendam que este tenha.

Também faz parte desta tarefa estilizar cada componente de várias maneiras para que estes se adequem aos vários temas da aplicação.

Nesta tarefa foram utilizadas as ferramentas ReactJS, JavaScript, HTM, Emotion e Storybook.

A segunda tarefa consistiu na criação de novos ecrãs para uma nova ferramenta no *widget* a ser desenvolvida. Esta ferramenta permite ao utilizador criar testes que verifiquem o estado dos guias (tutoriais criados através da aplicação) quando corridos nos vários browsers e dispositivos.

Durante a tarefa foram utilizados os componentes recriados na tarefa anterior. Nos ecrãs criados o utilizador poderá criar e visualizar testes com as características desejadas, assim como guardá-los, editá-los e eliminá-los da base de dados. Esta tarefa exigiu muita pesquisa pois era importante seguir a estrutura do código já desenvolvido para mantê-lo simples e fácil de entender para os outros desenvolvedores da empresa.

Nesta tarefa foram utilizadas as ferramentas ReactJS, JavaScript, HTML, Redux, Meteor e MongoDB.



## 4.2. Implementação

A tarefa iniciou com a recriação de um botão usado várias vezes. Este botão inicialmente era um componente diferente para diferentes ocasiões. Para simplificar não só a aparência visual da aplicação, mas também o código e a sua organização, foi criado um componente de um botão cujo título pode ser facilmente alterado ao chamá-lo. Este pode conter também um ícone à escolha que se adequa à função do botão.

Ao chamar esse componente é possível também chamar qualquer função, podendo esta ser independente do componente ou até estar presente noutro ficheiro.



Figura 20 – Componente de botão genérico

De seguida, foi construído um menu genérico. Este permite ter um título à escolha, os itens desejados, podendo ser cada um representado por um ícone e por um nome à escolha. Cada item executa uma função que fosse pedida exteriormente. Esse menu pode então ser inserido dentro de outro componente desejado.

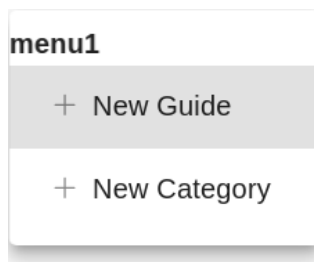


Figura 21 – Componente de menu genérico



Figura 22 – Componente de barra de navegação com o menu

De seguida foram desenvolvidos mais componentes, seguindo a mesma lógica dos anteriores.

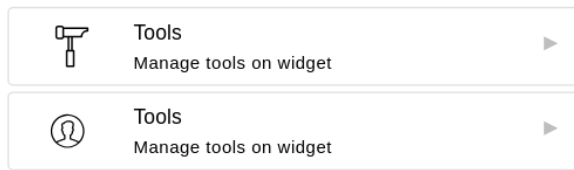


Figura 23 – componentes genéricos de ferramentas

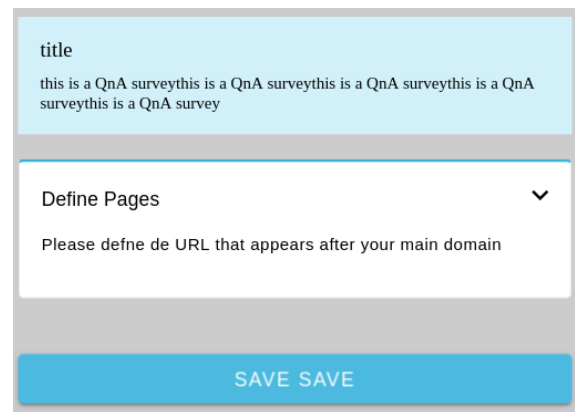


Figura 24 – Conjunto de componentes genéricos para criar uma página

Nesta tarefa do projeto foram adquiridos e aprofundados conhecimentos sobre React e JavaScript ES6.

A segunda tarefa foi iniciada com a criação de um ecrã que permitisse o utilizador criar um teste dentro de uma Suite. Para a realização da tarefa foram disponibilizados os *mockups* e explicado o que era necessário ser realizado.

A página em si consiste num formulário que permite ao utilizador escolher o nome do teste, a descrição e o guia a ser testado. Após o preenchimento do formulário e ao criar o teste este deve ser registado na base de dados com toda a informação introduzida.

O segundo ecrã a ser desenvolvido foi a lista dos testes para cada Suite. Seguindo a mesma estrutura do ecrã da lista das Suites o ecrã deve mostrar um botão para criar um teste na Suite selecionada e a lista dos testes dessa Suite com um menu por cada teste que permita editar ou eliminar o teste selecionado.

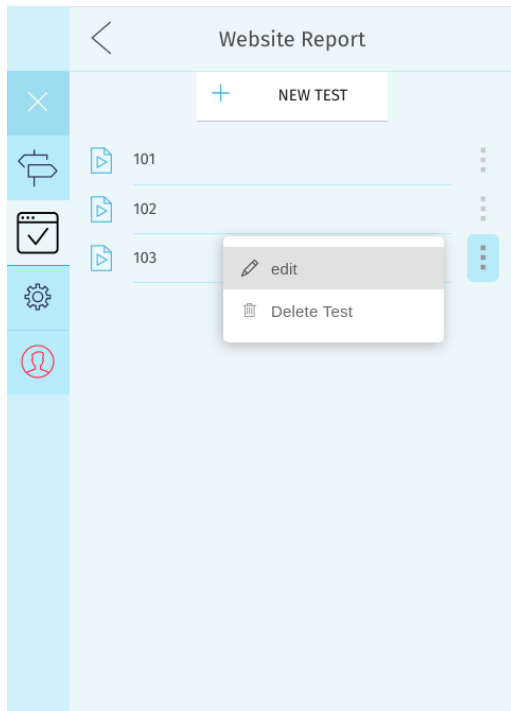


Figura 25 – Ecrã de lista de testes

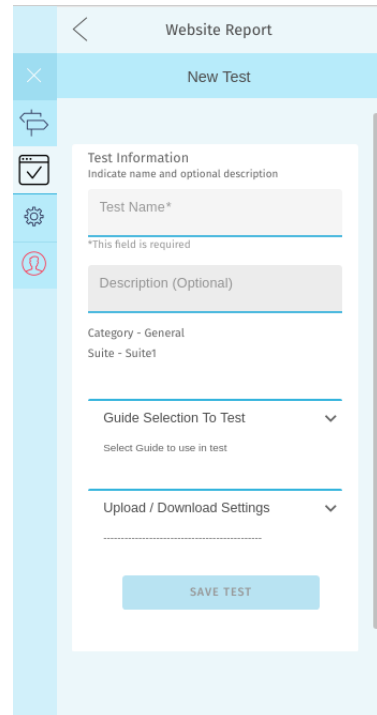


Figura 26 – Ecrã de criação de testes

No projeto desenvolvido apenas se alterou uma coleção de dados, esta chamada “testes”. A coleção contém documentos, cada um representando um teste, contendo toda a informação guardada.

Para executar as funções e alterar a base de dados foi necessário criar métodos Meteor, isto é, funções em JavaScript que alteram a base de dados utilizando MongoDB, podendo introduzir, editar e eliminar documentos na coleção de testes. Estas funções são declaradas no servidor e chamadas através do lado do cliente.

No final da realização das tarefas foram desenvolvidos vários diagramas que descrevam o que foi feito, assim como uma breve explicação do trabalho desenvolvido. Foram então criados os seguintes diagramas:

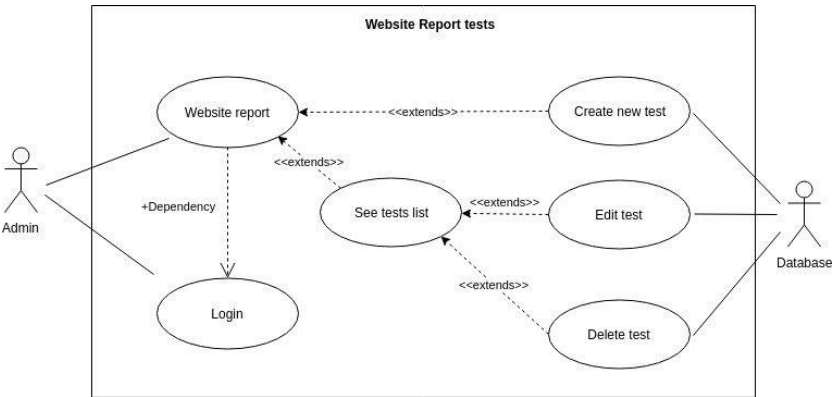


Figura 27 – Diagrama de casos de uso

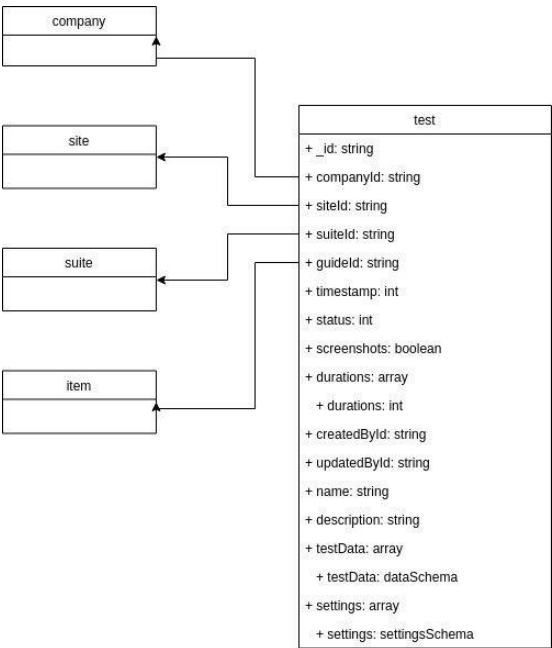


Figura 28 – Estrutura da coleção de testes

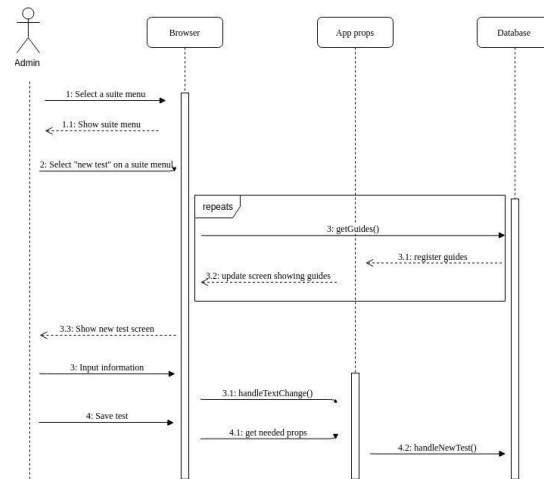


Figura 29 – Diagrama de sequência (representa a criação de um teste)

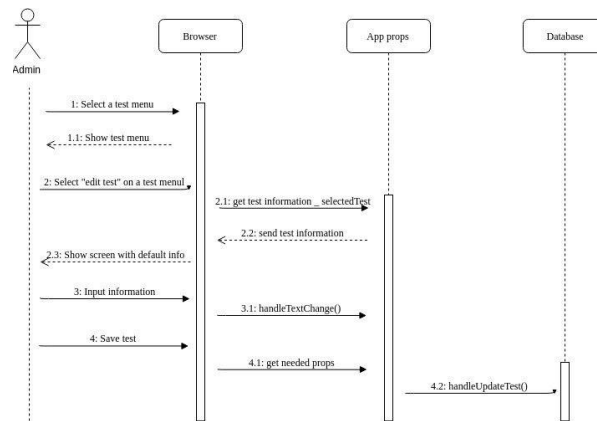


Figura 30 – Diagrama de sequência (representa a atualização de um teste / editar teste)

O dia a dia na empresa resume-se à execução das tarefas, muita pesquisa sobre as ferramentas utilizadas e leitura e compreensão do código existente da aplicação, de forma a entender a sua lógica e conseguir adaptá-lo. Durante todo o estágio o estagiário teve o apoio dos outros membros da equipa de desenvolvimento sempre que foi necessário

## Conclusão

Para um estudante do ramo de sistemas de informação a primeira experiência numa empresa é uma fase de aprendizagem bastante distinta, pois constrói-se algo para um público já existente e trabalha-se com uma equipa já com conhecimentos mais avançados na matéria. Por essa razão esta experiência foi algo bastante distinto de qualquer outra unidade curricular, sendo que foram adquiridos conhecimentos estando num ambiente real, destacando-se conhecimentos sobre métodos de trabalho de empresas do ramo.

O projeto desenvolvido baseou-se no desenvolvimento de novas funcionalidades para o produto Helppier (aplicação comercializada pela empresa acolhedora), assim como investigar e recriar componentes genéricos que pudessem ser utilizados ao longo da aplicação.

Durante a realização das tarefas foi necessária muita pesquisa e apoio da restante equipa para pesquisar as ferramentas utilizadas e entender o código já desenvolvido de maneira a adaptá-lo às novas funcionalidades desenvolvidas. Porém, ao longo do projeto, sentiu-se cada vez mais facilidade ao realizar as tarefas e estas foram concluídas como esperado. No futuro as tarefas realizadas serão aproveitadas pela empresa para continuar a desenvolver a ferramenta iniciada e reutilizar os componentes desenvolvidos.

Concluindo, na unidade curricular Projeto Final / Estágio foram adquiridos muitos conhecimentos que ajudarão na futura vida profissional, como o uso das diversas ferramentas utilizadas e desenvolver aplicações web a nível *full stack* usando novas tecnologias.

#### REFERÊNCIAS BIBLIOGRÁFICAS

Sobre a empresa helppier e concorrentes: <https://www.owler.com/company/helppier>

Helppier e WalkMe (diferenças): <https://www.helppier.com/en/walkme-alternative/>

Sobre JavaScript: [https://www.w3schools.com/whatis/whatis\\_js.asp](https://www.w3schools.com/whatis/whatis_js.asp)