



## **Relatório de Estágio**

Mestrado em Engenharia Informática - Computação Móvel

# **Humanportal**

**Mayelson de Sousa**

Leiria, novembro de 2020





## **Relatório de Estágio**

Mestrado em Engenharia Informática - Computação Móvel

# **Humanportal**

**Mayelson de Sousa**

Estágio realizado sob a orientação do Professor Alexandrino José Marques Gonçalves e sob supervisão do Eng. Sérgio Monteiro.

Leiria, novembro de 2020

# *À Minha Família*



# Agradecimentos

Antes de descrever as tarefas e os desafios que tive durante a realização deste estágio, não posso deixar de agradecer às pessoas que direta ou indiretamente contribuíram para o seu sucesso e para uma experiência enriquecedora.

Em primeiro lugar, à *humansoft* pela oportunidade de fazer parte da sua equipa de trabalho e por me ter proporcionado todas as condições necessárias para a concretização deste estágio. Um especial agradecimento ao supervisor do estágio, Eng. Sérgio Monteiro, pela orientação e confiança que depositou em mim. A toda a restante equipa da *humansoft*, que sempre proporcionou um ambiente de trabalho muito bom, que tive o prazer de aprender muito a nível profissional e cultural e que hoje considero como a minha segunda família.

Agradeço também ao meu orientador de estágio, professor Alexandrino Gonçalves, pela disponibilidade e orientação, e por toda ajuda, apoio e motivação que sempre me transmitiu.

Deixo também um agradecimento à Escola Superior de Tecnologia e Gestão (ESTG) do Instituto Politécnico de Leiria por me ter acolhido, a todos os colegas de curso por todo o apoio e companheirismo e a todos os docentes por todo o conhecimento que me transmitiram, desde o início do meu percurso em licenciatura a mestrado.

Um especial agradecimento a toda a equipa do programa “Bolsa fase do Instituto Politécnico de Leiria” e a todos os companheiros com quem colaborei nesse programa, pois apesar das dificuldades de colaborar e estudar, sem este programa não seria possível o sucesso que tive.

Por último, mas não menos importante, quero agradecer à minha família, em especial a minha mãe, Elsa Maria Quaresma de Sousa, aos meus amigos, destaque para Adeorel Bandeira e a minha namorada, Jussara Neto, por estarem sempre do meu lado, dando-me forças e coragem e por me apoiarem a cada passo da minha vida. Sem eles não seria possível o sucesso que tive.



# Resumo

O presente relatório foi realizado no âmbito da Unidade curricular de Estágio do Mestrado em Engenharia Informática - Computação Móvel, da Escola Superior de Tecnologia e Gestão do Instituto politécnico de Leiria.

O objetivo deste relatório é descrever todo o processo envolvido ao longo do estágio, que teve a duração de 10 meses. O estágio foi realizado na empresa *humansoft*, uma empresa que se dedica ao desenvolvimento de soluções informáticas na área da gestão de recursos humanos. Neste relatório está descrita, a caracterização da empresa onde foi realizado o estágio, a metodologia e ferramentas utilizadas, as principais tarefas realizadas e as conclusões.

O estágio tinha como principal objetivo, a reformulação e desenvolvimentos de alguns módulos do projeto *humanportal*. Este projeto consiste numa aplicação *web*, composta por 3 componentes e estes por vários módulos que oferecem à organização uma maior facilidade em todo o processo formativo, assim como permite distribuir e delegar muitas responsabilidades nos principais intervenientes da formação, tais como, formadores, formandos, coordenadores, técnicos, administradores entre outros.

Como resultado deste estágio, foi possível adquirir experiência e desenvolver competências a nível profissional através dos vários projetos integrados em ambiente empresarial.

**Palavras-chave:** *humansoft*, *humanportal*, Módulos, Componentes, Novo ORM, *React.js*.



# Abstract

This report was carried out within a curricular unit of the Master in Computer Engineering - Mobile Computing, of the School of Technology and Management of the Polytechnic of Leiria.

The goal of this report is to describe the process involved throughout the internship, which took ten months. The internship was carried out in *humansoft*, a company that aims to develop computer solutions for human resources management. This report describes the characterization of the company where the internship was conducted, the methodology and tools used, the main tasks performed and the conclusions.

The main purpose of the internship was the enhancement and development of several modules of the *humanportal* project. This online web project, made up 3 components and these by several modules, offer the organisation a greater ease in the whole formative process, as well as ease the distribution and assignment of responsibilities to the main actors of the training, such as trainers, trainees, coordinators, technicians, administrators, among others.

As a result of this internship, it was possible to acquire experience and develop a larger set of professional skills through the participation in various projects integrated in a business environment.

**Keywords:** humansoft, humanportal, Modules, Components, Novo ORM, React.js.



# Índice

Agradecimentos .....	v
Resumo .....	vii
Abstract .....	ix
Lista de Figuras .....	xiv
Lista de Tabelas .....	xvii
Lista de Siglas e Acrónimos .....	xix
<b>1. Introdução .....</b>	<b>1</b>
<b>1.1. Motivação e objetivos .....</b>	<b>1</b>
<b>1.2. Estrutura do Relatório .....</b>	<b>1</b>
<b>2. Enquadramento .....</b>	<b>3</b>
<b>2.1. Caracterização da Entidade de Acolhimento.....</b>	<b>3</b>
2.1.1. Equipa e condições de trabalho .....	3
2.1.2. Clientes .....	3
<b>2.2. humantrain.....</b>	<b>5</b>
2.2.1. Gestão Pedagógica.....	5
2.2.2. Gestão Financeira .....	6
2.2.3. Integração de sistemas .....	6
2.2.4. <i>Reporting Engine</i> .....	7
<b>2.3. humanportal.....</b>	<b>7</b>
2.3.1. Website .....	7
2.3.2. Área de formação.....	8
2.3.3. Área reservada à organização .....	9
<b>2.4. humanedu .....</b>	<b>10</b>
<b>2.5. Arquitetura .....</b>	<b>11</b>
<b>3. Metodologia e Ferramentas .....</b>	<b>13</b>
<b>3.1. Metodologia.....</b>	<b>13</b>
<b>3.2. Framework.....</b>	<b>14</b>
<b>3.3. Ferramentas .....</b>	<b>15</b>
3.3.1. humansoft CRM .....	16
3.3.2. Visual Studio Code .....	16
3.3.3. Sublime Text 3.....	17
3.3.4. HeidiSQL.....	17
3.3.5. Navegadores de <i>internet</i> .....	18
3.3.6. IIS .....	18
3.3.7. cPanel .....	18
<b>3.4. Tecnologias .....</b>	<b>19</b>
3.4.1. HTML, CSS e <i>Bootstrap</i> .....	19
3.4.2. Twig.....	20
3.4.3. JavaScript .....	20

3.4.4.	jQuery .....	21
3.4.5.	React.js .....	22
3.4.6.	PHP .....	26
3.4.7.	MySQL .....	27
3.4.8.	Git .....	27
<b>4.</b>	<b>Desenvolvimento .....</b>	<b>29</b>
<b>4.1.</b>	<b>Planeamento .....</b>	<b>29</b>
<b>4.2.</b>	<b>Componente “Website” .....</b>	<b>31</b>
4.2.1.	Site da <i>ETAP</i> .....	33
4.2.2.	Site do <i>Cencal</i> .....	34
4.2.3.	Site do <i>Cenfim</i> .....	37
4.2.4.	Site do <i>CFPSA</i> .....	38
4.2.5.	Site do <i>Modatex</i> .....	40
4.2.6.	Site do <i>Cincork</i> .....	43
<b>4.3.</b>	<b>Componente “Área de Formação” .....</b>	<b>45</b>
4.3.1.	Módulo “Ações do Coordenador” .....	45
4.3.2.	Exportação do “Cronograma para <i>Google Calendar</i> ” .....	48
4.3.3.	Reconstrução do módulo de “Testes” .....	49
4.3.4.	Reconstrução do módulo de “Trabalhos” .....	52
<b>4.4.</b>	<b>Componente “Área reservada à organização” .....</b>	<b>53</b>
4.4.1.	<i>Single Sign On</i> (SSO) - <i>Espaço Visual</i> .....	53
4.4.2.	Single Sign On (SSO) – Moodle .....	55
4.4.3.	Gestão de viaturas .....	56
4.4.4.	Autenticação.GOV .....	59
<b>5.</b>	<b>Conclusão .....</b>	<b>63</b>
<b>5.1.</b>	<b>Trabalho futuro .....</b>	<b>63</b>
	<b>Bibliografia .....</b>	<b>65</b>



# Lista de Figuras

Figura 1 - Principais clientes da <i>humansoft</i> .....	4
Figura 2 - Arquitetura dos principais produtos da <i>humansoft</i> .....	11
Figura 3 - Rotas do tipo <i>get</i> com <i>middleware</i> e <i>secret key</i> .....	14
Figura 4 - Exemplos de consultas com o ORM através do Novo ORM .....	15
Figura 5 – Exemplo de utilização de <i>Phinx</i> para adicionar nova coluna " <i>order</i> " .....	15
Figura 6 - Módulo de Gestão de pedidos de suporte .....	16
Figura 7 - <i>Visual Studio Code</i> .....	17
Figura 8 - <i>Sublime Text 3</i> .....	17
Figura 9 - <i>HeidiSQL</i> .....	18
Figura 10 - <i>cPanel dashboard</i> .....	19
Figura 11 - Número de downloads dado por <i>npm</i> package nos últimos 2 anos [19].....	23
Figura 12 - Nível de interesse de cada <i>framework</i> a nível mundial [20].....	24
Figura 13 - React.js fluxo de funcionamento [21].....	25
Figura 14 - <i>Virtual DOM</i> e <i>Real DOM</i> [22].....	26
Figura 15 - Função que devolve um determinado artigo de texto a vista.....	33
Figura 16 - Desenho inicial (esquerda) e resultado (direita) do site da ETAP .....	34
Figura 17 - Obtenção das últimas publicações do utilizador de <i>Instagram</i> .....	35
Figura 18 - Site <i>Cencal</i> página inicial .....	36
Figura 19 - Página inicial do <i>Website Cenfim</i> .....	37
Figura 20 - Partes da página inicial do <i>website CFPSA</i> .....	39
Figura 21 - Página das galerias do <i>website</i> do <i>CFPSA</i> .....	40
Figura 22 - Página inicial do <i>website</i> do <i>Modatex</i> .....	41
Figura 23 - Classe de <i>css - parallax</i> .....	42
Figura 24 - Código HTML da área de calendário de formação com efeito to <i>parallax</i> aplicado.....	42
Figura 25 - Efeito <i>parallax</i> com <i>jQuery</i> .....	42
Figura 26 - Calendário de formação <i>Modatex</i> .....	43
Figura 27 - Página inicial do <i>website</i> do <i>Cincork</i> .....	44
Figura 28 - Calendário de formação do <i>Cincork</i> .....	45

Figura 29 - Módulo “Ações do Coordenador” .....	46
Figura 30 – <i>Novo ORM</i> - Excerto de código do controlador que devolve os dados para preenchimento da tabela horizontal no módulo Ações do Coordenador.....	47
Figura 31 - <i>React.js</i> - Trecho de código que faz o render da tabela horizontal no módulo “Ações do Coordenador” .....	48
Figura 32 - Módulo cronograma- funcionalidade para exportar sessões.....	49
Figura 33 - Submódulo “Banco de Perguntas” .....	50
Figura 34 – Módulo de testes - Banco de testes .....	50
Figura 35 – Módulo de testes - ecrã de testes da ação do formador.....	51
Figura 36 - Módulo de testes – ecrã de testes da ação dos formandos .....	51
Figura 37 – Módulo de “Testes” antes da reestruturação.....	52
Figura 38 -Módulo de “Testes” depois da reestruturação .....	52
Figura 39 - Módulo de “Trabalhos” antes da reestruturação.....	53
Figura 40 – Módulo de “Trabalhos” depois da reestruturação .....	53
Figura 41 - SSO - trecho de código do registo de novo utilizador .....	54
Figura 42 - SSO - trecho de código que devolve o endereço de direcionamento para site do Espaço Visual.....	55
Figura 43 - SSO <i>Moodle</i> - opção para aceder ao <i>moodle</i> .....	56
Figura 44 - SSO <i>Moodle</i> - trecho de código do método que faz o SSO com o moodle .....	56
Figura 45 – Submódulo “Utilização de viaturas”.....	57
Figura 46 - Configuração de módulo “Gestão de viaturas” .....	58
Figura 47 - Área de viaturas disponíveis.....	58
Figura 48 - Áreas de Gestão de Requisições de viaturas.....	59
Figura 49 - Opção para autenticação através do cartão cidadão.....	60
Figura 50 - <i>Autenticação.GOV</i> - trecho de código do método “ <i>getAuthGov</i> ” .....	60
Figura 51 - <i>Autenticação.GOV</i> - trecho de código do método “ <i>ssoResponse</i> ” .....	61



# Lista de Tabelas

Tabela 1 - Cronograma das atividades efetuadas .....	30
--	----



## Lista de Siglas e Acrónimos

AJAX	<i>Asynchronous Javascript and XML</i>
API	<i>Application Programming Interface</i>
CRM	<i>Customer relationship management</i>
CRUD	<i>Create, Read, Update and Delete</i>
CSS	<i>Cascading Style Sheets</i>
CSV	<i>Comma Separated Values</i>
DNS	<i>Domain Name System</i>
DOM	<i>Document Object Model</i>
DSL	<i>Domain-Specific Language</i>
DTP	<i>Dossier Técnico Pedagógico</i>
ESTG	<i>Escola Superior de Tecnologia e Gestão</i>
GPL	<i>General Public License</i>
GUI	<i>Graphical User Interface</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IIS	<i>Internet Information Services</i>
IMAP	<i>Internet Message Access Protocol</i>
MIT	<i>Massachusetts Institute of Technology</i>
MVC	<i>Model view controller</i>
ORM	<i>Object Relational Mapper</i>
PHP	<i>Hypertext Preprocessor</i>
POP3	<i>Post Office Protocol</i>
PWS	<i>Personal Web Server</i>
REST	<i>Representational State Transfer</i>
SAML	<i>Security Assertion Markup Language</i>
SMS	<i>Short Message Service</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SOAP	<i>Simple Object Access Protocol</i>
SGBD	<i>Sistema de Gestão de Base de Dados</i>
SQL	<i>Structured Query Language</i>
TCP	<i>Transmission Control Protocol</i>
URL	<i>Uniform Resource Locator</i>

# 1. Introdução

No contexto da elaboração do trabalho final de Mestrado em Engenharia Informática - Computação Móvel, na Escola Superior de Tecnologias e Gestão (ESTG) do Instituto Politécnico de Leiria (IPLeiria), foi proporcionado a opção de realizar um estágio curricular e, por conseguinte, redigir este relatório final de estágio.

O estágio foi realizado na empresa *humansoft*, onde, durante 10 meses, foi colocado em prática todos os conhecimentos adquiridos ao longo do percurso académico principalmente nas áreas de desenvolvimento *frontend* e *backend*.

## 1.1. Motivação e objetivos

O motivo que levou à escolha da empresa *humansoft* para realizar o estágio curricular no âmbito do mestrado deveu-se principalmente ao facto de esta ter apresentado propostas de projeto interessantes para o decorrer do mesmo. Além disso, é uma empresa bem situada geograficamente e com uma equipa de trabalho experiente e enriquecedora.

A *humansoft* é uma empresa de desenvolvimento de soluções informáticas na área da gestão de recursos humanos. Foi fundada em 2005, com um projeto inovador na área de gestão e automatização de processos de formação, de simples utilização e grande capacidade de adaptação às necessidades das organizações, o que possibilitou um crescimento rápido e com sucesso.

Dando continuidade à sua estratégia de desenvolver soluções inovadoras e maximizar a satisfação do cliente através do aumento da eficiência dos processos, a *humansoft* conta na atualidade com duas principais aplicações: uma de gestão de processo formativo (*humantrain*) desenvolvida utilizando a linguagem de programação *c#*, assente de raiz na plataforma *Microsoft Sql Server*, e outra de interligação entre o *humantrain* e a *internet* (*hmanportal*), desenvolvida com base em *HTML*, *CSS*, *JavaScript*, *React.js* e *PHP*.

Com o objetivo de colocar em prática os conhecimentos adquiridos ao longo do percurso académico, aprender a conviver com o mercado profissional, adquirir novas competências, criar e desenvolver hábitos de trabalho e sentido de responsabilidade, este estágio consistiu na reformulação e desenvolvimento de alguns módulos da aplicação *humanportal* da *humansoft*, principalmente ao nível da componente *website*, área de formação e área reservada às organizações.

## 1.2. Estrutura do Relatório

O relatório encontra-se dividido em 5 capítulos, incluindo este de Introdução, na qual é feita a introdução ao tema e indicados os objetivos do estágio.

Seguidamente, o capítulo do Enquadramento, onde é feita a caracterização da entidade de acolhimento e apresentação dos seus projetos.

A seguir, o capítulo da Metodologias e Ferramentas, onde é apresentada a metodologia, a *framework*, as ferramentas e as tecnologias utilizadas ao longo do estágio.

Seguidamente, o capítulo de Desenvolvimento, onde é apresentado o planeamento das tarefas e as suas implementações.

Por fim, a Conclusão, onde é apresentada uma reflexão final do relatório, referindo os resultados obtidos e as dificuldades que surgiram ao longo do estágio. Ainda nesse capítulo são apresentadas sugestões de trabalhos futuros.

## 2. Enquadramento

Este capítulo destina-se a apresentar a empresa e a fazer o enquadramento dos projetos para o qual foram desenvolvidas as soluções.

### 2.1. Caracterização da Entidade de Acolhimento

*humansoft* - Sistema integrados de informação, Lda, é uma empresa que se dedica ao desenvolvimento de soluções informáticas na área da gestão de recursos humanos, sediada no Edifício Beira Rio, na Rua Anzebino da Cruz Saraiva N° 416 Piso 1 Esc. 27 em Leiria. Foi fundada em 2005, com um projeto inovador na área de gestão e automatização de processos de formação, de simples utilização e grande capacidade de adaptação às necessidades das organizações, o que possibilitou um crescimento rápido e com sucesso. Hoje, todas as soluções encontram-se em desenvolvimento e aperfeiçoamento de forma contínua, com um ciclo de desenvolvimento entre 1 e 2 meses, permitindo responder de forma rápida e positiva às exigências constantes do mercado.

No sentido de abranger as várias necessidades do mercado na componente dos Recursos Humanos, a *humansoft* estabeleceu parcerias com empresas como *ínCentea*, Câmara de Comércio e Indústria Luso-Alemã, *International Business Consulting*, *talenter* e *PSIquatro*, além de crescer e inovar com produtos e serviços de qualidade que são reflexos do profissionalismo que se mantêm.

#### 2.1.1. Equipa e condições de trabalho

Atualmente a equipa da *humansoft* é constituída por 8 elementos, dos quais integra gestores de projetos, *tester*, *developers* e assistentes técnicos. Desde o primeiro dia do estágio, foi possível verificar a existência de um ambiente de trabalho muito bom entre todos os colaboradores, transmitindo sempre grande motivação e disponibilidade.

No primeiro dia de estágio, a empresa disponibilizou, em primeira mão todo o material necessário para desempenhar as tarefas pretendidas, no qual se incluiu um computador portátil, monitor, rato e uma base de apoio e arrefecimento para o portátil.

#### 2.1.2. Clientes

Na Figura 1 estão apresentados os principais clientes da *humansoft*, sendo estes, entidades públicas e privadas com processos formativos.



Figura 1 - Principais clientes da humansoft

Dando continuidade à sua estratégia de desenvolver soluções inovadoras e maximizar a satisfação dos seus clientes através do aumento da eficiência dos processos, a *humansoft* conta na atualidade com uma plataforma de gestão de processo formativo (*humantrain*), uma plataforma de interligação entre o *humantrain* e a *internet* (*humanportal*) e uma plataforma de gestão escolar (*humanedu*).

## 2.2.humantrain

O *humantrain* é um sistema de gestão do processo formativo desenvolvido pela *humansoft* utilizando a linguagem de programação *c#*, assente de raiz na plataforma *Microsoft Sql Server*, que permite às entidades formadoras/empresas rentabilizarem a gestão dos seus recursos tanto ao nível pedagógico como financeiros. Nesse sentido *humantrain* conta com alguns principais módulos: gestão pedagógica, gestão financeira e integração de sistemas, *reporting engine*. Além disso conta com um motor de produção de relatórios, que permite aos utilizadores gerarem listagens e documentos em *Microsoft word e Excel*, uma ferramenta de integração entre o *humantrain* e outros sistemas, tais como SEIP, software de gestão de recursos humanos e gestão integrada (ex: *humanportal*, Primavera e PHC), ou outros sistemas assentes em base de dados.

### 2.2.1. Gestão Pedagógica

A gestão pedagógica da formação é um processo excessivamente burocrático. O *humantrain* foi concebido principalmente para colmatar este desafio, ou seja, transformar e automatizar processos complexos em tarefas simples para o utilizador. Este módulo conta com os seguintes submódulos:

- **Ficha de formando**

Permite aos utilizadores, atualizarem os dados dos formandos e formadores de forma automática através de cartão cidadão e consultar de forma rápida e prática o histórico de formação do formando e formador.

- **Ranking de formadores**

Permite aos utilizadores consultarem os melhores formadores da instituição. Este ranking é baseado numa ferramenta de questionários do *humantrain* que podem ser direcionados a vários públicos alvo, tais como: formandos, formadores e colaboradores.

- **Planificação**

Permite antecipar conflitos no processo de calendarização tais como: formandos sobrecarregados (formandos inscritos em pelo menos duas ações de formação em que uma ou mais sessões ocorrem no mesmo período), formadores, salas e material de apoio sobre alocados.

- **Controlo de Presenças**

Permite controlar de forma fácil e intuitiva todo o processo da gestão de assiduidade dos formandos.

- **Avaliação da Formação**

Permite gerir todo o processo de avaliação de cada ação de formação com diferentes parâmetros e regras. Este submódulo é bastante flexível, o que permite adaptar a realidade de cada ação de formação. Além disso, é permite configurar os parâmetros de avaliação quantitativa e qualitativa em diferentes escalas ao nível dos módulos ou ação.

### 2.2.2. Gestão Financeira

Na gestão financeira, destaca-se principalmente o processamento de apoios sociais e honorários a formadores, controlo de conta corrente de formandos, formadores e outras entidades, lançamento de despesas elegíveis de formação financiada, processamento de pagamentos e compensações à entidade patronal.

Por forma a aumentar a eficiência destes processos, este módulo conta com os seguintes submódulos:

- **Apoios sociais e honorários**  
Permite determinar através de registo de assiduidades dos formandos e formadores, os honorários e os apoios sociais a pagar, de forma autónoma.
- **Pagamentos PS2 / Home Banking**  
Permite o processamento rápido e fácil de pagamentos a formandos e formadores. Além disso, possibilita a criação automática de ficheiros PS2 e *Home Banking* para envio ao banco.
- **Lançamentos**  
Permite integrar de forma automática ou manual, todas as despesas elegíveis da formação financiada no sistema. Além disso, permite gerar todos os documentos legais para pedidos de reembolso.

Como forma de aumentar a eficiência destes processos, a *humansoft* disponibiliza um módulo que torna possível automatizar a maioria dos mesmos através da integração do *humantrain* com outros sistemas como Primavera, PHC, etc.

### 2.2.3. Integração de sistemas

O sucesso de qualquer solução informática não se resume apenas às funcionalidades exclusivas do sistema em si. Este deverá interagir e comunicar com outros sistemas da organização. Para além do *humantrain* responder às exigências das regras de negócio da formação (*core business* do sistema), incorpora um módulo que permite comunicar de forma bidirecional com qualquer outro sistema igualmente aberto.

Este módulo, permite a introdução ilimitada de configurações a outros sistemas. Atualmente o *humantrain* já interage com os seguintes sistemas:

- SAFT-PT;
- Primavera;
- PHC;
- SEIP;
- Meta4;
- PeopleNet.

#### 2.2.4. *Reporting Engine*

A *humansoft* criou de raiz um motor de configuração e produção de documentos em *Word*, *Excel* e *Pdf* e vai incluído em todas as versões do *humantrain*, o que permite que o próprio cliente configure as suas próprias listagens.

### 2.3. *humanportal*

Atualmente a presença na *internet* é praticamente de carácter obrigatório, para divulgação de novidades, eventos, bem como todas as informações relativas às organizações como entidade formadora. Nesse sentido, a *humansoft*, desenvolveu o *humanportal* para oferecer às organizações a solução completa para tornar as suas presenças na internet eficaz, simples, dinâmica e muito apelativa.

O *humanportal* é uma aplicação *web* que possibilita a interligação entre o *humantrain* e a *internet*. Esta interligação é feita através da API's (*REST Webservices*) desenvolvidas pela *humansoft*. Aliada a sincronização, a *humansoft* desenvolveu também um sistema de *cronjobs* denominada de "*Observable*", que permite despoletar determinadas ações dependendo das configurações.

Esta aplicação *web* é composta por vários componentes que oferecem à organização uma maior visibilidade no mundo virtual e ao mesmo tempo facilita todo o processo formativo, uma vez que permite distribuir e delegar muitas responsabilidades nos principais intervenientes da formação, tais como, formadores, formandos, técnicos, administrativos, entre outros.

#### 2.3.1. Website

O componente *website* permite desenvolver sites de forma dinâmica com a possibilidade de definir a estrutura de menus, artigos textos, notícias e todos os outros conteúdos. Em alternativa, o *humanportal* pode interagir com o site existente. Por exemplo, no caso de a organização pretender manter o site atual, o *humanportal* poderá publicar neste, ações de formação, eventos e praticamente todos os conteúdos do *humantrain*. A seguir estão alguns dos módulos que compõem o este componente:

- **Gestão de Menus**

Este módulo permite aos utilizadores criarem menus com o objetivo de os disponibilizarem para o site. Desta forma a *humansoft*, oferece aos seus clientes a possibilidade de criarem menus para *websites* de forma dinâmica. Entre as funcionalidades, estão disponíveis as seguintes: criação de novos menus, desativar/ativar a publicação de menus, adicionar parentes aos menus assim como ordená-los.

- **Artigos de texto**

Este módulo permite aos utilizadores criarem artigos de textos com o objetivo de os disponibilizarem para o site. Entre as funcionalidades, estão disponíveis as seguintes: criação/edição/eliminação de artigos de textos, desativar/ativar artigos de textos, visualização de resultados. Os artigos de

textos contam ainda com a funcionalidade de adição de conteúdo em destaque, onde o utilizador poderá adicionar um texto e uma imagem para destaque.

- **Conteúdos dinâmicos**

Este módulo oferece um conjunto alargado de formas de disponibilização de conteúdos no site. Está constituído por três submódulos: Tipos de conteúdos dinâmicos, Categorias de conteúdos dinâmicos e os itens do conteúdo dinâmico. Cada item está associado a uma categoria e um tipo de conteúdo dinâmico, enquanto que cada categoria está associada a um tipo de conteúdo dinâmico. Desta forma os utilizadores podem, através da sua criatividade, criar várias combinações de forma a disponibilizar todo o conteúdo que desejar no site. Por exemplo, criação de uma galeria ou a criação de uma página de descontos por categorias.

- **Imagens de *banner***

Este módulo permite aos utilizadores carregarem imagens de *banner* para o *website*. No entanto podem ser utilizados para outros fins, como por exemplo disponibilizar imagens dos acionistas do cliente. Permite fazer o upload/download de imagens e configurar o número de imagens que aparecerão no *banner* do site.

- **Explorador de ficheiros**

Este módulo, permite aos utilizadores efetuarem toda a gestão dos ficheiros do site e do *humanportal*. Entre as funcionalidades, destacam-se o upload/download de ficheiros, eliminação/renomeação e partilha de ficheiros entre utilizadores, ações de formação ou grupos do *humanportal*.

- **Notícias**

Este módulo permite aos utilizadores criarem notícias com o objetivo de os disponibilizarem para o site. Está constituído por dois submódulos: Categorias de notícias e Notícias. Entre as funcionalidades, estão disponíveis as seguintes: criação/edição/eliminação de categorias de notícias e/ou notícias, ativar/desativar categorias de notícias e/ou notícias, visualização de resultados das notícias. As notícias contam ainda com a funcionalidade de adição de conteúdo em destaque, onde o utilizador poderá adicionar um texto e uma imagem para destaque.

### 2.3.2. Área de formação

Este é um componente que torna possível a interação de todos os intervenientes da formação. Entre Formadores e Formandos, Colaboradores e Coordenadores, o processo administrativo das organizações torna-se mais simples e flexível. O processo burocrático-administrativo da formação transfere-se para múltiplos recursos humanos. Por exemplo, o formador regista o sumário e a assiduidade em sala de aula, o formando pode interagir com o formador ou pessoal administrativo através da sua área, o coordenador da formação pode acompanhar e validar a informação registada, o registo de inscrições pode ser feito online por

parte de candidatos e validado à posteriori por um administrativo etc. Além destas, este componente conta com módulos que permitem também:

- Publicar as ações a iniciar no site da organização;
- Gerir pré-inscrições dos visitantes / candidatos;
- Gerir registos de formadores externos, ou seja, bolsa de formadores;
- Editar/Atualizar os dados pessoais de todos os intervenientes da formação;
- Responder aos questionários por parte dos formandos, formadores e outros colaboradores, assim como gerir a análise de questionários;
- Controlar e registar presenças (assiduidade) e sumários;
- Gerir perguntas e testes;
- Gerir ocupação das salas;
- Gerir e validar os documentos;
- Preencher a avaliação dos formandos;
- Através de fóruns e chat, comunicar entre entidades da organização;
- Imprimir relatórios e estatísticas;
- Gerir o centro de mensagens, ou seja, consultar, enviar, responder as notificações/email;
- Envio de ficheiros diretamente para a ficha do utilizador no *humantrain* e para o DTP;
- Manipular e gerir eventos;
- Gerir *newsletter*, inclui envio em massa de *newsletter* para todos os contactos do sistema ou oriundos de outros sistemas externos;
- Gerir notícias;
- Gerir Negociação de formadores;
- Gerir pedidos de desbloqueio das avaliações dos formandos.

### 2.3.3. Área reservada à organização

O *humanportal* consegue adaptar-se às necessidades das organizações. Com este componente é possível desenvolver funcionalidades específicas da empresa.

Entre as principais funcionalidades estão:

- **Módulo de requisição de material**  
Permite gerir a comunicação entre os diversos intervenientes no processo de requisição de material de formação.
- **Módulo de integração com serviço SMS.**  
Permite otimizar recursos e melhorar os processos de envios de SMS da organização. Entre as funcionalidades, estão os envios de SMS por áreas de formação, grupos do *humanportal*, entre outros. Além de permitir aos utilizadores visualizarem o histórico de SMS.
- **Módulo de gestão de equipamentos**

Permite gerir os equipamentos da organização, tais como inserir um novo equipamento no sistema, associar a uma ação, requisitar, validar, consultar e controlar onde os equipamentos estão alocados.

- **Módulo de requisição de viaturas**

Permite gerir todo o processo de requisição da viatura, tais como: inserir viatura, requisitar, consultar viatura e requisição, validar requisição, atribuir matrícula e entregar veículo.

- **Módulo *single-sign-on***

Permite aos utilizadores autenticados no *humanportal* acederem/autenticarem-se em plataformas externas. Como por exemplo, *single-sign-on* com *moodle*.

## 2.4. *humanedu*

O *humanedu* é uma solução integrada de gestão escolar para gerir processos académicos de Escolas Profissionais e Universidades segundo o tratado de Bolonha. Permite registar todo o percurso académico e financeiro dos alunos, tais como as Candidaturas, Processo de Seriação, Matrículas, Equivalências, Avaliações, Emissão de Recibos, Controlo de Conta Corrente. Para registar esses percursos, *humanedu* conta com algumas características como:

- **Ficha do Aluno**

Permite consultar em cada momento a situação do aluno, ou seja, o percurso letivo, avaliações de todos os cursos e configurações financeiras. Além disso, permite registar equivalências.

- **Matrícula**

Permite inscrever o aluno nas unidades curriculares do ano corrente e/ou anterior, determinar equivalências, unidades extracurriculares e controlar os documentos obrigatórios a entregar. Permite ainda configurar a emissão e impressão automática dos recibos dos alunos. Estes documentos financeiros são certificados pela AT.

- **Editor de Horários**

Permite visualizar o horário dos cursos, docentes e salas, bem como alertas de sobreposição e sobrelotação.

- **Emissão de Recibos**

O processo financeiro é muito versátil. O sistema pode ser configurado por forma que, em cada momento seja possível emitir um recibo automaticamente a partir de outro ecrã. Por exemplo, durante a inscrição numa época de avaliação especial. Outro exemplo, no ecrã de registo de matrículas, após validar uma nova matrícula, o software pode imprimir de forma automática o recibo. Mesmo em casos de atraso no pagamento, o sistema pode calcular multas com base em regras pré-definidas.

O *humanedu* foi desenvolvido na mesma plataforma do *humantrain*, pelo que usufrui das mesmas potencialidades, como por exemplo, o editor de listagens que permite a configuração de documentos específicos da instituição.

## 2.5.Arquitetura

Na Figura 2 está ilustrada a arquitetura dos dois principais produtos da *humansoft* (o *humantrain* e o *humanportal*) assim como as comunicações entre os mesmos. Além disso, é possível visualizar outras comunicações entre outras plataformas.

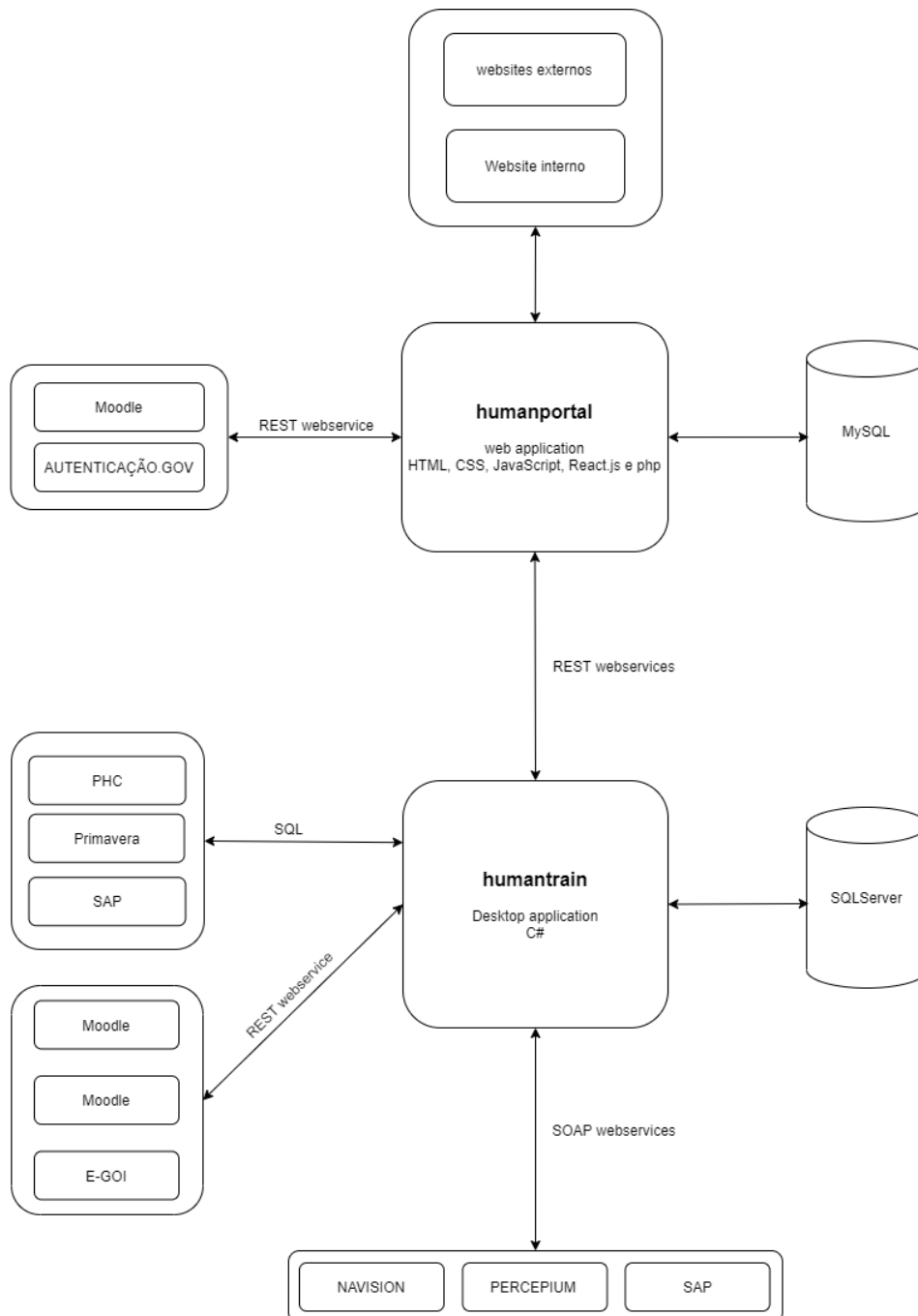


Figura 2 - Arquitetura dos principais produtos da *humansoft*



## 3. Metodologia e Ferramentas

Neste capítulo será descrita a metodologia de desenvolvimento utilizada durante o desenvolvimento dos projetos, bem como as ferramentas utilizadas para a realização das tarefas ao longo do estágio.

### 3.1. Metodologia

A metodologia adotada pela *humansoft* é baseada em *Scrum*. O *Scrum* é uma metodologia ágil usada para gestão e desenvolvimento de projetos que promove a flexibilidade, visibilidade e produtividade, além disso, é uma metodologia iterativa e incremental, que permite rever o planejamento, mudar de direção ou fazer adaptações nas iterações seguintes, assim como, permite focar nas funcionalidades prioritárias dos clientes [1].

Na *humansoft* os projetos/funcionalidades são divididos em *Sprints*. *Sprint* representa um dos pilares da metodologia adotada pela empresa e pode ser definido como o tempo de um mês ou menos quando um Incremento de produto “*Done*”, utilizável e potencialmente funcional é criado. Normalmente têm durações consistentes ao longo de um esforço de desenvolvimento e durações constantes ao longo de todo o esforço de desenvolvimento. Um novo *Sprint* só começará quando o *Sprint* anterior for concluído.

Existem três papéis importantes no *Scrum*:

- **O *Product Owner***  
É o responsável por identificar as funcionalidades a implementar e definir a sua prioridade em conjunto com todas as pessoas envolvidas no projeto, designadas por *stakeholders*.
- **O *Scrum Master***  
É o responsável por assegurar que o processo do *Scrum* seja usado corretamente e por remover impedimentos que possam impedir a equipa de desenvolvimento de concluir os objetivos de cada *Sprint*.
- ***Development team***  
É responsável por implementar as funcionalidades definidas para cada *Sprint*.

Na *humansoft*, o *Product Owner* define uma lista de funcionalidades ordenadas por prioridade designada por *Product Backlog*. Antes de iniciar um *Sprint* é realizado um *Sprint Planning Meeting* que é uma reunião na qual está presente toda a equipa *Scrum* em que é efetuado o *Sprint Backlog*, uma lista de tarefas prioritárias selecionadas do *Product Backlog* em que a equipa de desenvolvimento se compromete em realizar no *Sprint*.

Em cada dia do *Sprint* a equipa da *humansoft* realiza uma reunião diária às 10 horas da manhã, chamada de *Daily Scrum*. Esta tem como objetivo disseminar conhecimento sobre o

que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho a ser realizado no dia que se inicia.

No final de cada *Sprint* é realizada uma reunião designada por *Sprint Review Meeting* com toda a equipa e os respetivos clientes, em que é apresentado o trabalho realizado ao longo do *Sprint*, denominada de *Sprint Demo*. O mais importante é que a equipa atinja o objetivo do *Sprint*.

### 3.2. Framework

A *framework* utilizada pela *humansoft* foi desenvolvida pela própria empresa e é designada de *Novo ORM*. A *Novo ORM* é uma *framework* de desenvolvimento rápido para PHP, cujo o principal objetivo é permitir o desenvolvimento de forma estruturada e rápida. Utiliza o *Composer*<sup>1</sup> para gerir as dependências e o modelo MVC, para separar a interface de utilizador de toda a lógica da aplicação.

Com a *Novo ORM*, é possível desenvolver facilmente com o conceito de rotas. As rotas fazem um mapeamento da URL digitada no navegador para alguma ação dentro da aplicação. É possível registar rotas para qualquer método do HTTP (*get*, *post*, *put*, *patch*, *delete* ...), além disso, o sistema de rotas permite-nos trabalhar com parâmetros de rotas, que podem ser opcionais ou não, restrições de parâmetros, *middlewares* e *secret keys*. Na Figura 3 está ilustrada um exemplo de rota com restrição de parâmetros.

```
//Rota para obter utilizadores,
// é necessário estar autenticado e deve ser passado a secret key como parametro
// o resultado é json
Router::register('get/users', 'app\ctr\api\user/get')
->middleware('_sys\middleware\UrlAuth')
->param('secret_key', 'XPTO')
->responds('json');
```

Figura 3 - Rotas do tipo *get* com *middleware* e *secret key*

O sistema de *template* utilizado permite utilizar PHP juntamente com HTML sem qualquer restrição. Os ficheiros têm extensão “.*html.php*”. Um dos principais benefícios do uso desse sistema de *template* é a herança, que permite trabalhar facilmente com o conceito de *default layout / master page*. Para interação com base de dados a *framework* utiliza *ActiveRecord*<sup>2</sup>, no entanto está sendo migrada lentamente para o *ORM*.

*ORM* aplica o *Design Patter ActiveRecord* onde cada tabela de base de dados é representada no código através de uma classe “*Model*” que é usado para interagir com essa tabela. Os *Models* permitem a consulta de dados em suas tabelas, bem como, operações de

<sup>1</sup> O *Composer* é um gestor de pacotes ao nível da aplicação para a linguagem de programação PHP que fornece um formato padrão para gerir dependências do software PHP e bibliotecas necessárias.

<sup>2</sup> *ActiveRecord* é uma abordagem para acesso de dados numa base de dados. Uma tabela de base de dados ou visão (*view*) é associada (*wrapped*) à uma classe. Portanto, uma instância de um objeto é associada a um único registo (tupla) na tabela. Após a criação e gravação de um objeto, um novo registo é adicionado à tabela. Qualquer objeto carregado obtém suas informações a partir do base de dados. Quando um objeto é atualizado, o registo correspondente na tabela também é atualizado.

CRUD. Na Figura 4 estão dois exemplos de consultas simples aos dados da tabela “*courses*” através do *Novo ORM*.

```
// SELECT * FROM courses
// Corresponde à:
app\models\Course::find()->get();

/* SELECT co.course as name, co.area, ca.courseaction_ref
FROM courses co
JOIN courseactions ca ON ca.course_code = co.course_code
WHERE co.course_type = 'xpto' AND ca.web_pub = 1
ORDER BY co.area asc
LIMIT 50 */
//Corresponde à:
app\models\Course::find()->alias("co")
->select("co.course as name, co.area, ca.courseaction_ref")
->join("courseactions ca", "ca.course_code = co.course_code")
->where("co.course_type = :type")
->where("ca.web_pub", 1)
->params([":type" => 'xpto'])
->orderBy("co.area asc")
->limit(50)
->get();
```

Figura 4 - Exemplos de consultas com o ORM através do Novo ORM

Para gestão de base dados de forma automática/manual é utilizado o *Phinx*. O *Phinx* é uma ferramenta via linha de comando para gestão de migrações de base de dados. Permite executar *scripts* para realizar operações a base de dados. Um exemplo prático da utilização desta ferramenta é quando se cria uma coluna devido uma nova funcionalidade. É necessário criar esta coluna para todos os clientes que utilizará esta nova versão. Nesse sentido ao adicionar a versão no cliente, o *Phinx* executa o *script* e cria automaticamente a coluna na base de dados do cliente. Na Figura 5 está um trecho de código que representa o exemplo de um script criado através do *Phinx*.

```
<?php
use Phinx\Migration\AbstractMigration;

class DynamiccontentColumOrder extends AbstractMigration
{
    public function up()
    {
        if($this->hasTable('dynamiccontents')){
            $dynamiccontents = $this->table('dynamiccontents');
            if (!$dynamiccontents->hasColumn('order')) {
                $dynamiccontents->addColumn('order', 'integer', ['limit' => 11, 'null' => true, 'default' => null]);
            }
            $dynamiccontents->save();
        }
    }
}
```

Figura 5 – Exemplo de utilização de *Phinx* para adicionar nova coluna "order"

### 3.3.Ferramentas

Neste subcapítulo estão identificadas as ferramentas que foram utilizadas durante o desenvolvimento dos projetos ao longo do estágio.

### 3.3.1. humansoft CRM

A *humansoft* CRM é uma ferramenta de gestão de projetos, desenvolvida pela própria empresa. Esta ferramenta faz parte de um dos módulos do *humanportal* e permite:

- Efetuar a gestão de todos os pedidos de suporte (Figura 6), tais como criar e eliminar pedido de suporte, responder ao pedido de suporte atualizar os estados, consultar os pedidos em cursos, concluídos, pendentes, entre outros;
- Gerir de forma fácil as tarefas em *Sprints*, assim como consultar as estatísticas de desenvolvimento em cada *Sprint*;
- Gerir os projetos, tais como, consultar, editar, eliminar e atualizar;
- Gerir Clientes tais como, consultar, editar, eliminar e atualizar.

The screenshot displays the 'humansoft - Gestão de suporte' interface. It features a search form with the following fields and options:

- Estado:** (nenhum selecionado) ▾
- Estado int.:** (nenhum selecionado) ▾
- Visibilidade:** (todos) ▾
- Técnico:** (nenhum selecionado) ▾
- Produto:** (nenhum selecionado) ▾
- Versão:** (selecione uma opção)
- Projeto:** (selecione uma opção)
- Ver. entre:** (nenhuma) ▾ (nenhuma) ▾
- Tipo pedido:** (selecione uma opção)
- Cliente:** (selecione uma opção)
- Entre:** [ ] [ ]
- Menu/listagem:** (selecione uma opção)
- Quem?:** pedido por
- Conc. entre:** [ ] [ ]
- Pesquisa:** palavras chave

Additional filters include checkboxes for 'Sem data de conclusão', 'Tarefas planificadas', and 'Tarefas não planificadas'. A 'Pesquisar' button is located at the bottom right of the search form.

Below the search form is a table with the following columns: #, Projeto, Pedido, Técnico(s), Resolução, Data do pedido, Análise Interna, Previsto, Duração, Desconl, Estado. The table contains one row of data:

#	Projeto	Pedido	Técnico(s)	Resolução	Data do pedido	Análise Interna	Previsto	Duração	Desconl	Estado
18663	Humansoft	Opção para escolher o local, qd se regista u	Sousa	Adicionado novo campo para escolher o loc	02-01-2020	Adicionado o novo campo "local" como obri	0:0	1:0	0:0	✓

Figura 6 - Módulo de Gestão de pedidos de suporte

### 3.3.2. Visual Studio Code

O *Visual Studio Code* é um editor de código-fonte desenvolvido pela *Microsoft* para *Windows*, *Linux* e *macOS* [2]. Inclui suporte para depuração, controlo *Git* incorporado, realce de sintaxe, complementação inteligente de código, *snippets* e *refatoração* de código. É personalizável, fazendo com que os utilizadores possam mudar o tema do editor, teclas de atalho e preferências, além disso é um *software* livre e de código aberto, apesar do *download* oficial estar sob uma licença proprietária. Foi uma das ferramentas de desenvolvimento mais utilizada durante todo o estágio. Na Figura 7 é apresentada uma imagem ilustrativa deste editor.

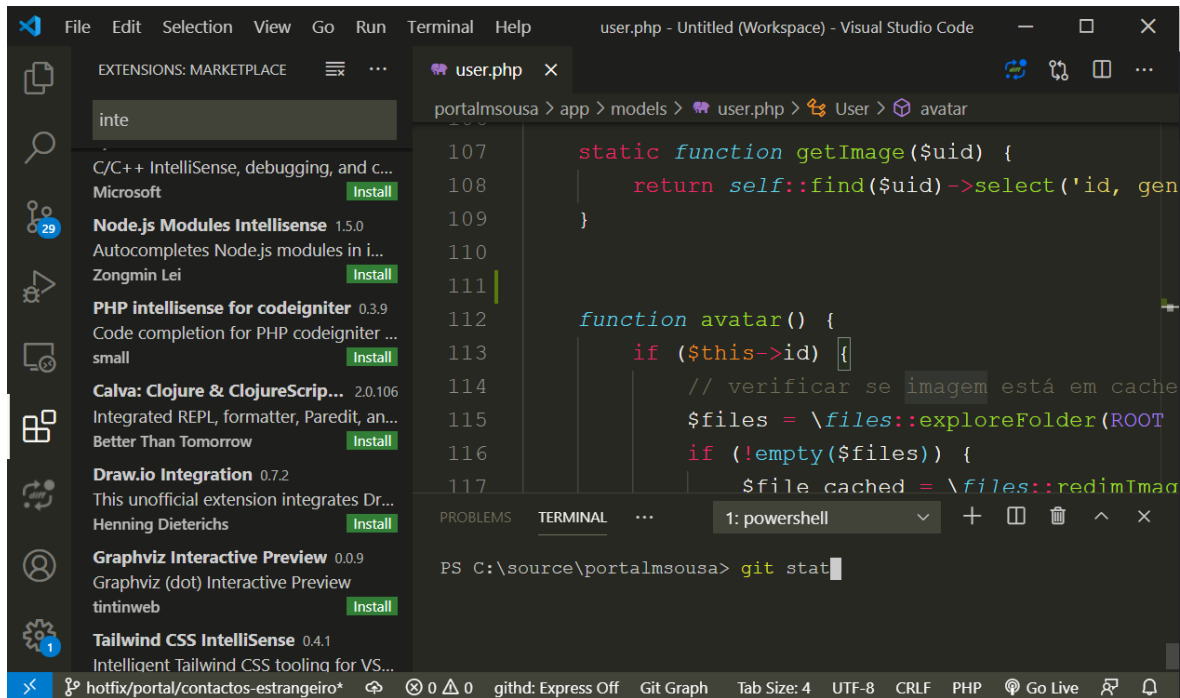


Figura 7 - Visual Studio Code

### 3.3.3. Sublime Text 3

É um *software* multiplataforma de edição de texto, utilizado por muitos desenvolvedores para editar código fonte [3]. Foi igualmente uma das ferramentas de desenvolvimento, utilizadas durante o estágio. Na Figura 8 está uma imagem ilustrativa deste editor.

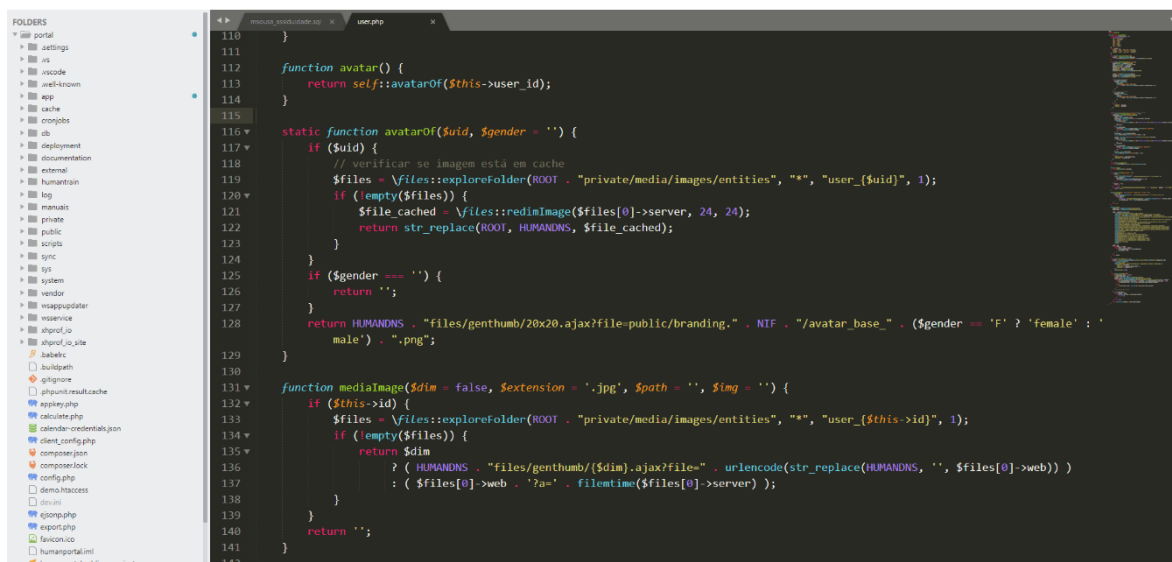


Figura 8 - Sublime Text 3

### 3.3.4. HeidiSQL

Anteriormente conhecido como *MySQL-Front* é um programa livre e cliente código-aberto para o *MySQL*, *Microsoft SQL SERVER* e *PostgreSQL* [4]. Durante o estágio, esta ferramenta foi utilizada para realizar toda a gestão de bases de dados. Na Figura 9 está uma imagem ilustrativa desta ferramenta.

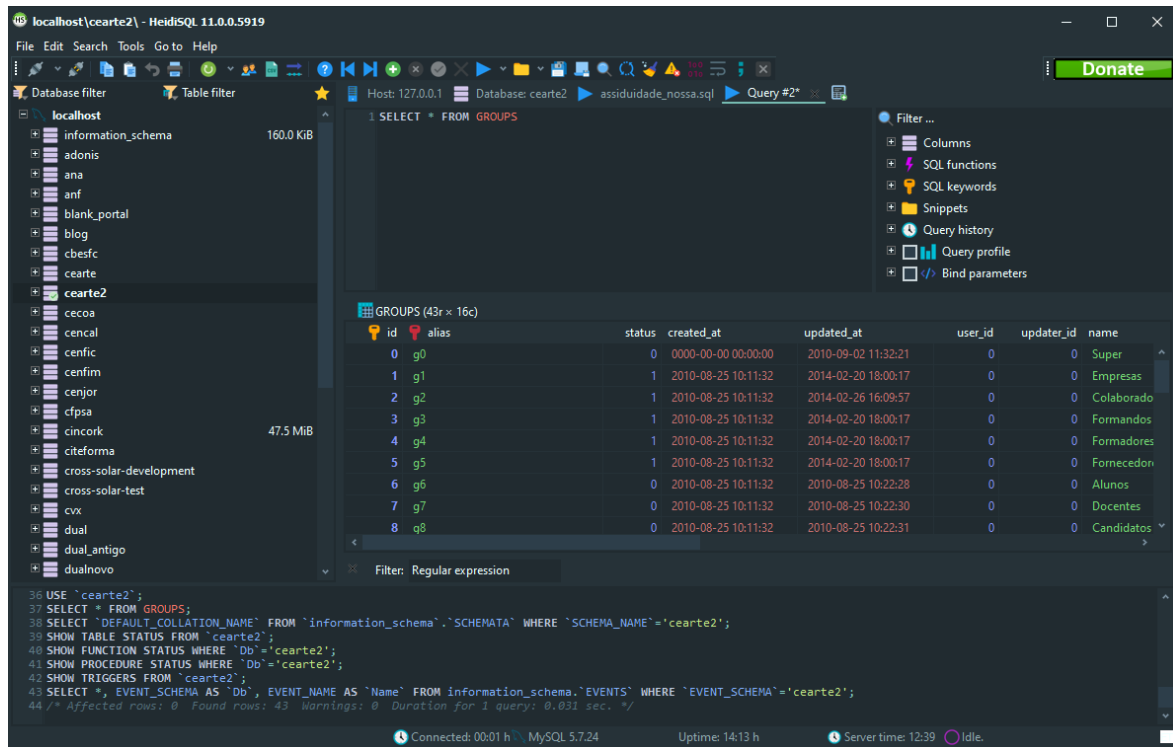


Figura 9 - HeidiSQL

### 3.3.5. Navegadores de internet

Visto que a oferta de *browser* é extensa e que todos eles não interpretam da mesma forma o código HTML e CSS, foram utilizadas alguns dos browsers mais conhecido, tais como o Google Chrome o Firefox, o Opera o Safari e o Internet Explorer, durante o estágio, para desenvolvimento e assistências dos componentes do *humanportal*.

### 3.3.6. IIS

*IIS (Internet Information Services - anteriormente denominado Internet Information Server)* é um servidor *web* criado pela *Microsoft* para servidores *windows* [5]. A primeira versão foi introduzida com o *Windows NT Server* versão 4, e passou por várias atualizações. É também o sucessor do PWS<sup>3</sup>. Esta ferramenta foi utilizada para criar o servidor de desenvolvimento, que foi utilizado durante o estágio.

### 3.3.7. cPanel

O *cPanel* é um *software* de gestão de Hospedagem *web*, com interface web na porta TCP 2082, ou na porta TCP 2083 quando em modo seguro (porta padrão).

É um sistema baseado no *Linux* que fornece uma interface gráfica e ferramentas de automação, projetadas para simplificar o processo de Hospedagem de um site da *web*. O *cPanel* utiliza uma estrutura de 3 camadas que fornece recursos para administradores, revendedores e

<sup>3</sup> PWS (Personal Web Server) foi um servidor pessoal de páginas da Microsoft. Era utilizado nos sistemas operativos Windows 95 e Windows 98.

utilizadores finais, proprietários do *website* para controlar vários aspetos de administração de site e servidor através de um navegador da *web* padrão [6].

Além do GUI, *cPanel* também tem linha de comando e acesso baseado em API que permite que fornecedores de software de terceiros, organizações de hospedagem *web* e desenvolvedores, automatizem os processos de administração de sistema padrão.

Suporta aplicações baseadas em *Apache*, *PHP*, *MySQL*, *PostgreSQL*, *Perl* e *BIND (DNS)*. Suporta serviços de email que inclui serviços *POP3*, *IMAP* e *SMTP*.

Esta ferramenta foi utilizada muitas vezes para efetuar a gestão de servidor dos clientes *humansoft*. Na Figura 10 está uma imagem ilustrativa do *dashboard* desta ferramenta.

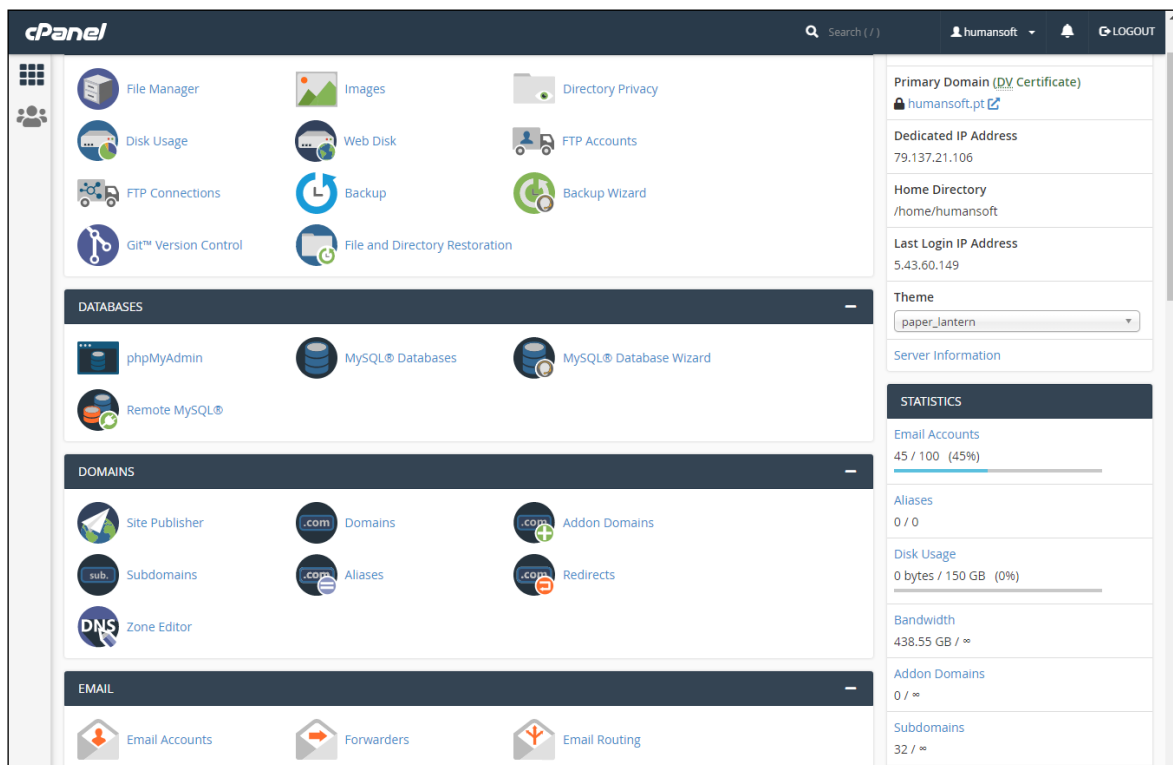


Figura 10 - *cPanel dashboard*

## 3.4. Tecnologias

Neste subcapítulo estão identificadas as tecnologias que foram utilizadas durante o desenvolvimento dos projetos ao longo do estágio.

### 3.4.1. HTML, CSS e *Bootstrap*

**HTML** é uma linguagem de marcação utilizada para publicar conteúdos tais como: texto, imagem, vídeo, áudio, ficheiros entre outros, na *web*. Esta linguagem é a base de qualquer página *web*, independentemente da sua natureza e das tecnologias envolvidas no processo de desenvolvimento. Fornece a estrutura geral de um *website* e utiliza elementos, representados por *tags*, que constituem os blocos para a construção de páginas *web* [7].

**CSS** é uma linguagem de estilo utilizada para alterar a aparência de páginas *web* construídas em linguagens de marcação, como HTML. Enquanto que o HTML forma a estrutura, o CSS foca-se no design. O CSS é responsável pela tipografia, pelos fundos, pelas cores, entre outras características de design. Uma das particularidades do CSS é a possibilidade de ter a formatação visual da página toda em ficheiros à parte de onde a estrutura está escrita, sendo que o mesmo ficheiro CSS pode ser vinculado a várias páginas, o que permite uma melhor organização de código e torna possível a um programador ou designer, alterar a aparência de várias páginas em simultâneo [8].

**Bootstrap** é uma *framework web* com código-fonte aberto para desenvolvimento de componentes de interface *frontend* para sites e aplicações *web* utilizando o HTML, CSS e *JavaScript*, baseado em modelos de *design* para a tipografia, melhorando a experiência do utilizador em um site amigável e responsivo [9].

### 3.4.2. Twig

**Twig** é sistema de *template* de código aberto, para PHP, cujo a sintaxe origina de *templates* de *Jinja* e *Django*. É licenciada pela *BSD License* e mantida por *Fabien Potencier*. A versão inicial foi criada pelo *Armin Ronacher* *SymFONY* [10].

O *Twig* conta com alguns recursos, como:

- Rápido – otimização na compilação do modelos e código PHP. A sobrecarga em comparação com o código PHP normal foi reduzida ao mínimo;
- Seguro - possui um modo *sandbox* para avaliar o código do modelo não confiável, o que permite que seja utilizado como uma linguagem de modelo para aplicações em que os utilizadores podem modificar o design do modelo;
- Flexível - é alimentado por um *lexer* e analisador flexível, o que permite que o desenvolvedor defina suas próprias *tags* e filtros personalizados e crie sua própria DSL.

A documentação relativa ao *Twig* está disponível em [11].

### 3.4.3. JavaScript

*JavaScript* (frequentemente abreviado como *JS*) é uma linguagem de programação compilada simples, interpretada, com funções de primeira classe. É uma linguagem dinâmica, baseada em protótipos e que suporta os paradigmas de programação orientada a objetos, imperativa e declarativa (incluindo programação funcional). Juntamente com HTML e CSS, o *JavaScript* é uma das três principais tecnologias da *World Wide Web*. Permite páginas da *web* interativas e, portanto, é uma parte essencial das aplicações *web*. A grande maioria dos sites utiliza, e todos os principais navegadores têm um mecanismo *JavaScript* dedicado para executá-lo. É atualmente a principal linguagem para programação *client-side* em navegadores *web* e é baseada em *ECMAScript*, padronizada pela *Ecma international* nas especificações ECMA-262 e ISO/IEC 16262 [12].

*JavaScript* conta com algumas características como:

- Suporte universal;
- Imperativa e Estruturada;
- Tipo dinâmico, ou seja, várias formas de testar o tipo de um objeto;
- Baseada em objetos;
- Avaliação em tempo de execução;
- Funções de primeira classe, ou seja, objetos que possuem propriedades e métodos, e podem ser passados como argumentos, serem atribuídos a variáveis ou retornados como qualquer outro objeto;
- Funções aninhadas, ou seja, funções definidas dentro de outras funções;
- Fechamentos, ou seja, permite que funções aninhadas sejam criadas com o escopo léxico no momento de sua definição e possui o operador “()” para invocá-las em outro momento;
- Utiliza protótipos em vez de classes para o mecanismo de herança;
- Extensões específicas do fornecedor.

A documentação relativa ao *JavaScript* está disponível em [13].

#### 3.4.4. jQuery

*jQuery* é uma biblioteca de funções *JavaScript* que interage com o HTML, desenvolvida para simplificar os *scripts* interpretados no navegador do cliente (*client-side*). *jQuery* é a mais popular das bibliotecas *JavaScript*. Esta biblioteca é um projeto da *JS Foundation*, que por sua vez faz parte da *Linux Foundation* e seu desenvolvimento é realizado pela equipa *The jQuery Team*.

*jQuery* é uma biblioteca de código aberto que utiliza a licença MIT em seu código-fonte. A sintaxe do *jQuery* foi desenvolvida para tornar mais simples a navegação do documento HTML, a seleção de elementos DOM, criar animações, manipular eventos, desenvolver aplicações AJAX e criação de plugins sobre ela. Tais facilidades permitem aos desenvolvedores criarem camadas de abstração para interações de baixo nível de modo simplificado em aplicações web dinâmicas de grande complexidade [14].

##### **Principais funcionalidades do *jQuery*:**

- Resolução da incompatibilidade entre os navegadores;
- Redução de código;
- Reutilização do código através de plugins;
- Utilização de uma vasta quantidade de plugins criados por outros desenvolvedores;
- Trabalha com AJAX e DOM;
- Implementação segura de recursos do CSS1, CSS2 e CSS3.

A documentação do *jQuery* está disponível em [15].

### 3.4.5. React.js

A crescente popularidade de *frameworks JavaScript* na comunidade originou o aparecimento de várias com o intuito de ajudar, quem desenvolve a lidar com as tarefas de programação. No entanto, a quantidade destas *frameworks* aumentou para níveis não expectáveis há uma década. Nesse sentido, torna-me mais difícil para os profissionais identificar quais destas efetivamente podem ser mais valias para as suas necessidades. Numa rápida análise pelo mercado das *frameworks JavaScript*, podemos destacar: *Vue.js*, *Angular* e *React.js*.

*Vue.js* é uma *framework* progressiva mais vocacionada para o desenvolvimento de interfaces de utilizador. Foi projetada desde a sua conceção para ser adaptável incrementalmente. A biblioteca principal é focada exclusivamente na camada visual (*view layer*), sendo fácil de integrar com outras bibliotecas ou projetos existentes. Por outro lado, o *Vue.js* é igualmente apto para o desenvolvimento de sofisticadas *Single-Page Applications* quando usado em conjunto com ferramentas atuais e bibliotecas de apoio [16].

#### Principais características:

- Progressivo - abrange todas as etapas da sua aplicação, desde as rotas até ao render para o utilizador. Desse modo, adapta-se facilmente a qualquer projeto;
- Versátil - permite a adição de funcionalidades adicionais através do carregamento de bibliotecas;
- Reativo - é baseado em quatro fundamentos basilares:
  - Elasticidade - reage expeditamente aos pedidos da aplicação gerindo em tempo real a alocação dos recursos;
  - Responsividade - reage às ações dos utilizadores; desse modo os erros são detetados de forma mais rápida e tratados com eficácia;
  - Resiliência - reage a falhas sem que o processo pare;
  - Orientado a mensagens - utiliza o envio de mensagens assíncronas para estabelecer fronteiras entre os componentes.

*Angular* é uma plataforma de aplicações *web* de código aberto e *frontend* baseado em *TypeScript* liderado pela equipa *Angular* da *Google* e pela comunidade. A versão atual do *Angular* é, na realidade, uma reimplementação do *AngularJS*, efetuado pelos seus criadores [17].

#### Principais características:

- É baseado em classes de Programação Orientada a Objetos;
- Tipo Estática, ou seja, não exige declaração do tipo de dados, pois o mesmo é definido dinamicamente para cada variável;
- Programação genérica;
- O *TypeScript* é um conjunto do *ECMAScript* 6 (ES6), e é compatível com *ECMAScript* 5 (i.e.: *JavaScript*). O *Angular* também inclui o ES6, nomeadamente:
  - Lambdas;

- Iterações;
- For/Of loops;
- *Python*;
- Reflexão.
- Carregamento dinâmico;
- Modelo de compilação assíncrono;
- A substituição de controladores e *\$escopo* com componentes e diretrizes – um componente é uma diretiva com um modelo;
- Programação reativa de suporte utilizando *RxJS*.

O *React* (também denominado *React.js* ou *ReactJS*) é uma biblioteca JavaScript de código aberto, com enfoque na criação de interfaces de utilizadores (*frontend*) em páginas web. É mantido pelo *Facebook*, *Instagram*, outras empresas, bem como uma comunidade de desenvolvedores individuais.

Em 2015, o *Facebook* anunciou o módulo *React Native*, que em conjunto com o *React.js*, possibilita o desenvolvimento de aplicações para Android e iOS utilizando componentes de interface nativos de ambas plataformas, sem precisar recorrer ao HTML.

O *React.js* tem um bom suporte documental e tutoriais onde, qualquer interessado, facilmente pode iniciar o desenvolvimento com esta tecnologia [18].

Através da plataforma *npm trends* é possível aferir das *frameworks* com o mais número de *download* nos dias de hoje, nomeadamente o *React.js* (Figura 11).

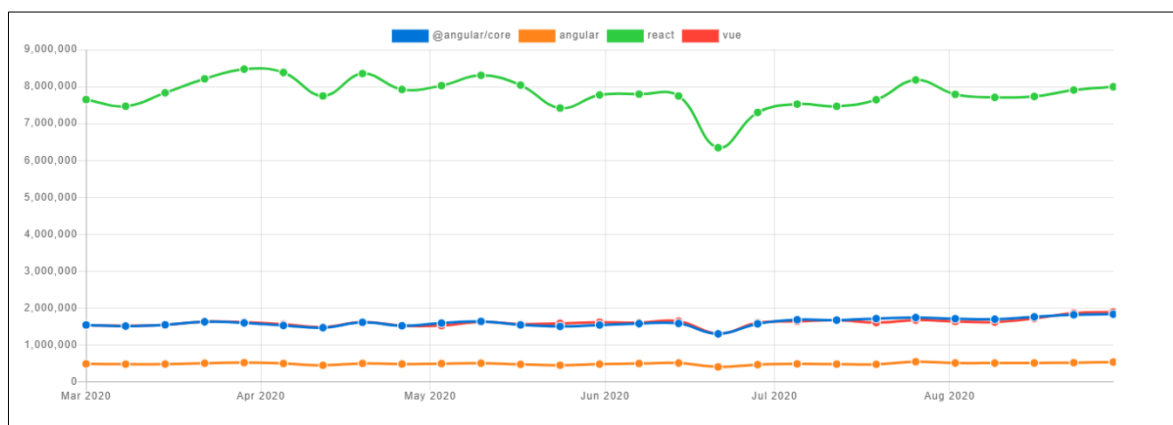


Figura 11 - Número de downloads dado por *npm* package nos últimos 2 anos [19]

Igualmente por intermédio do *Google thrend* [20], é possível verificar que o *React.js* tem um nível de interesse superior, quando comparado com as *frameworks* mencionadas anteriormente (Figura 12).

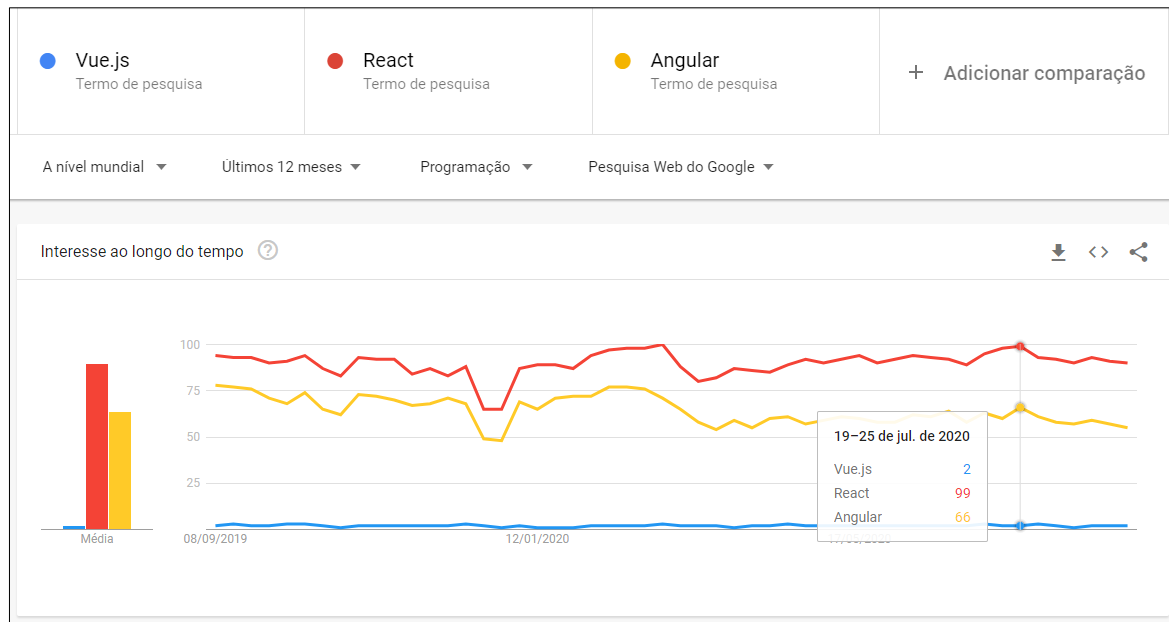


Figura 12 - Nível de interesse de cada *framework* a nível mundial [20]

Estes são apenas alguns exemplos da importância que o *React.js* tem no mercado nos dias de hoje e um dos fatores que motivou a sua utilização em alguns projetos que foram desenvolvidos durante o estágio.

### Composição do *React.js*:

- O *React.js* é composto por vários módulos;
- A própria biblioteca *React.js*, e o *React-DOM* para manipulação do DOM;
- *React-router* para implementar rotas;
- JSX, uma linguagem de marcação especial que agrega HTML em *JavaScript*;
- *React Create App*, uma interface de linha de comando que permite configurar rapidamente um projeto *React.js*;
- Bibliotecas *JavaScript* baseadas em *Axios* e *Redux*, que permitem organizar a comunicação com o *back-end*;
- *React Developer* que é uma ferramenta de desenvolvimento para os clientes *Chrome* e *Firefox*;
- *React Native* para o desenvolvimento de aplicações móveis nativas para iOS e Android.

### Principais características do *React.js*:

- Simplicidade: a abordagem baseada em componentes, o ciclo de vida bem definido e o uso de *JavaScript* simplificado tornam o *React.js* simples e prático de utilizar;
- Fácil aprendizagem: bastam conhecimentos básicos de HTML e CSS, para facilmente compreender e aprender o *React.js*;

- Abordagem nativa: pode ser utilizado no desenvolvimento de aplicações móveis (*React Native*) e por suportar a reutilização de código, simultaneamente podem ser implementadas aplicações para iOS, Android e *web*;
- *Data Binding*: utiliza a ligação de dados unidirecional e uma arquitetura de aplicação denominada Flux, que controla o fluxo de dados para os componentes por intermédio de um ponto de controlo;
- Testes: as aplicações desenvolvidas em *React.js* são muito mais fáceis de serem testadas.

### Modo de funcionamento de *React.js*

O *React.js* opera usando o conceito de *Virtual DOM*, que é uma estrutura em “árvore” com objetos simples, que potenciam o aumento da performance da aplicação, quando comparado com a execução nativa no browser (Figura 13).

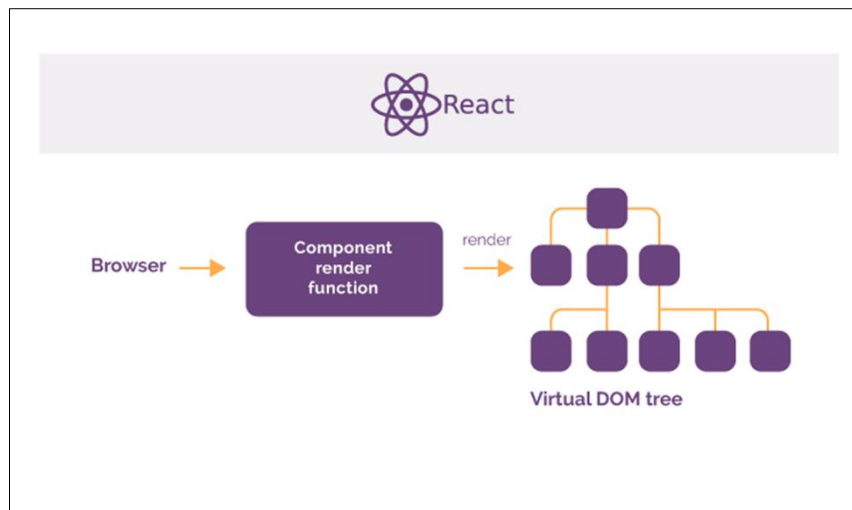


Figura 13 - *React.js* fluxo de funcionamento [21]

Aquando da construção do *Virtual DOM*, a “árvore” gerada é comparada com a árvore nativa do DOM do *browser* e apenas os componentes que necessitam de uma nova renderização são atualizados. Como efeito prático, este procedimento acelera a renderização dos componentes. Na Figura 14 é apresentado um diagrama ilustrativo deste procedimento.

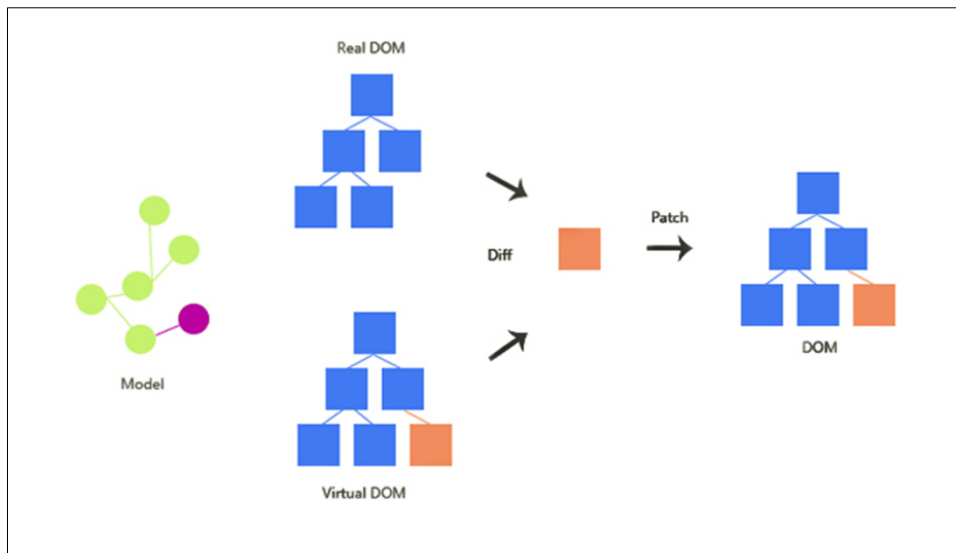


Figura 14 - *Virtual DOM* e *Real DOM* [22]

### 3.4.6. PHP

PHP é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na *World Wide Web*. Figura entre as primeiras linguagens passíveis de inserção em documentos HTML, dispensando em muitos casos o uso de arquivos externos para eventuais processamentos de dados. O código é interpretado no lado do servidor pelo módulo PHP, que também gera a página *web* a ser visualizada no lado do cliente. A linguagem evoluiu, passou a oferecer funcionalidades em linha de comando, e além disso, ganhou características adicionais, que possibilitaram usos adicionais do PHP, não relacionados a web sites. É possível instalar o PHP na maioria dos sistemas operativos, gratuitamente. Concorrente direto da tecnologia ASP pertencente à Microsoft, o PHP é utilizado em aplicações como o *MediaWiki*, *Facebook*, *Drupal*, *Joomla*, *WordPress*, *Magento* e o *Oscommerce*.

Criado por Rasmus Lerdorf em 1995, o PHP tem a produção de sua implementação principal, referência formal da linguagem, mantida por uma organização chamada *The PHP Group*. O PHP é um *software* livre, licenciado sob a *PHP License*, uma licença incompatível com a *GNU General Public License (GPL)* devido a restrições no uso do termo PHP. [23]

PHP conta com algumas características como:

- Velocidade e robustez;
- Orientação a objetos;
- Portabilidade - independência de plataforma;
- Tipo dinâmico;
- Sintaxe similar a C/C++ e o Perl;
- *Open-source*;
- *Server-side* (O cliente faz o pedido e o servidor responde em página HTML).

Toda a documentação relativo ao PHP em [24].

### 3.4.7. MySQL

O *MySQL* é um sistema de gestão de base de dados (SGBD), que utiliza a linguagem SQL como interface. É atualmente um dos sistemas de gestão de base de dados mais populares da *Oracle Corporation*, com mais de 10 milhões de instalações pelo mundo [25].

*MySQL* conta com algumas características como:

- Portabilidade (suporta praticamente qualquer plataforma atual);
- Compatibilidade (existem *drivers* ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como *Delphi, Java, C/C++, C#, Visual Basic, Python, Perl, PHP, ASP e Ruby*);
- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de nova hardware;
- Facilidade na utilização;
- É um *Software Livre* com base na GPL (entretanto, se o programa que aceder ao *Mysql* não for GPL, deverá adquirir uma licença comercial);
- Contempla a utilização de vários *Storage Engines* como *MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid* entre outros;
- Suporta controlo transacional, *Triggers, Cursors (Non-Scrollable e Non-Updatable), Stored Procedures e Functions*;
- Replicação facilmente configurável;
- Interfaces gráficas (*MySQL Toolkit*) de fácil utilização cedidos pela *MySQL Inc.*

Toda a documentação relativo ao *MySQL* em [26].

### 3.4.8. Git

*Git* é um sistema de controlo de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registar o histórico de edições de qualquer tipo de ficheiro. Foi inicialmente projetado e desenvolvido por Linus Torvalds para o desenvolvimento do *kernel Linux*, mas foi adotado por muitos outros projetos.

Cada diretório de trabalho do *Git* é um repositório com um histórico completo e habilidade total de acompanhamento das revisões, não dependente de acesso a uma rede ou a um servidor central. O *Git* também facilita a reprodutibilidade científica em uma ampla gama de disciplinas, da ecologia à bioinformática, arqueologia à zoologia.

O *Git* é um software livre, distribuído sob os termos da versão 2 da *GNU General Public License*. A sua manutenção é atualmente supervisionada por Junio Hamano. [27]

O *Git* conta com algumas características como:

- Suporta rápidas criações de ramos (*branches*) e *merges*, e inclui ferramentas específicas para visualização e navegação de históricos de desenvolvimento não lineares;

- Desenvolvimento distribuído - cada *developer* possui uma cópia local completa de todo o histórico de desenvolvimento, e as mudanças são copiadas de um único repositório para outro;
- Repositórios podem ser publicados por HTTP, FTP, *rsync*, um protocolo *Git* sobre uma porta conhecida ou por *ssh*;
- Manipulação eficiente de projetos extensos;
- Autenticação criptográfica do histórico;
- Modelo baseado em ferramentas;
- Estratégias de *merge* conectáveis;
- O lixo fica acumulado se não for limpo;
- Empacotamento periódico explícito de objetos.

Toda a documentação relativo ao *Git* em [28].

## 4. Desenvolvimento

Durante o estágio foram desenvolvidas várias atividades desde assistências à reconstrução e desenvolvimento aos módulos de componente *website*, área de formação e área reservada à organização do *humanportal*.

A maioria das atividades enquadram-se com as aprendizagens adquiridas ao nível académico. Com esses projetos, tornou-se possível elevar as experiências académicas para um nível mais profissional, desenvolvendo novas *skills* e adquirindo novos conhecimentos ao nível tecnológico e profissional.

### 4.1. Planeamento

O estágio teve início a 1 de outubro de 2019 e terminou a 31 de julho de 2020. Tendo decorrido num contexto empresarial, foi necessária a existência de um período de adaptação, com a finalidade de integração na equipa de desenvolvimento e aquisição de conhecimento do *Novo ORM*, e dos produtos da *humansoft*.

As primeiras semanas consistiram na instalação de software, de forma a criar um ambiente de desenvolvimento sólido, com todas as ferramentas de trabalho necessárias, assim como a formação interna acerca do produto *humanportal* e *humantrain*.

Este período permitiu obter uma melhor perceção acerca do modelo de negócio da empresa.

Na Tabela 1 estão ilustrados os tempos despendidos em cada um dos projetos que tiveram mais ênfase neste estágio.

Nos subcapítulos a seguir são apresentadas as atividades que foram efetuadas durante o estágio.



## 4.2. Componente “Website”

O componente *website*, como já foi dito no capítulo Enquadramento, permite desenvolver *websites* de forma dinâmica. Nesse sentido, foram efetuados vários projetos de desenvolvimento de *websites* para diferentes clientes da *humansoft*, sendo que alguns consistiram em reconstrução de *websites* já existentes, baseado num novo desenho e outros em criação de novos *websites*.

Nesses projetos foram utilizados o *Visual Studio Code* e o *Sublime Text* para edição de código fonte, o *HeidiSQL* para gestão de base dados, com intuito de simular o ambiente de produção de cada cliente.

Todos os projetos tiveram como base o desenho elaborado por um design contratado, e foram desenvolvidos tendo como base HTML, CSS, *JavaScript* e *jQuery* para desenvolvimento de *frontend* e, *MySQL* e PHP utilizando a *framework* da *humansoft*, o *Novo ORM* para desenvolvimento do *backend*.

Os desenvolvimentos dos *websites* passavam-se em criação de estruturas que permitissem aos utilizadores efetuarem toda a sua gestão, através dos módulos de componente *website*. Dessa forma os utilizadores poderiam adicionar, remover e/ou alterar as notícias, os artigos de textos, os conteúdos dinâmicos e os menus. Assim sendo, todos os projetos tinham várias áreas do *website* semelhantes e com o mesmo comportamento, diferenciando apenas nos *designs*.

O resultado de cada projeto, consistia em:

- **Header** - dividido em duas ou três áreas, sendo a primeira, a área de informações fixas, tais como, contactos gerais da empresa, e endereços das redes sociais, a segunda área de menu principal, composta por conteúdos do módulo de menu, a opção para a pesquisa no *website* e para iniciar a sessão no *humanportal* e as opções para as traduções. Ao finalizar o cabeçalho, encontra-se a área de *banner*, composta por conteúdos do módulo de conteúdos dinâmicos, desenvolvido com base no *carousel* do *Bootstrap 4*.
- **Home** - dividido em várias áreas, sendo que estava sempre presente, uma área de cursos em destaque, composta por cursos que foram selecionados e colocados em destaque no *humantrain* e que são sincronizados com o *humanportal*, outra de notícias em destaques, composta por conteúdos do módulo de notícias, outra de *newsletter*, que permitem aos utilizadores subscreverem as *newsletters* fornecidas pelo módulo de *newsletter* e poderiam ter ou não, uma ou mais áreas de módulo de artigos de textos e conteúdos dinâmicos.
- **Single Tamplate** – composta por páginas que renderizam todos os cursos, artigos de textos, notícias, conteúdos dinâmicos e páginas específicas, assim como os respetivos detalhes.
- **Footer** – dividido em duas ou três áreas, sendo que esteve sempre presente uma área menu secundário, outra de logotipos institucionais, de endereços de redes sociais,

---

políticas de privacidade, termos de e condições de utilização e site da empresa *humansoft*.

Para gerar os menus no *website*, o *Novo ORM*, conta com um método que recebe como parâmetros o “*id*” do menu “pai” e o nível da renderização. Sendo o método recursivo, este obtém automaticamente filhos de cada registo descendente até ao nível definido nos parâmetros. O método pode ser chamado através da linha de código “*app/models/Menu::render(\$elemId, \$level)*” e retorna o código HTML do menu, o que facilita no desenvolvimento dos menus principais e secundários.

No que diz respeito aos artigos de textos o *Novo ORM*, conta com um controlador geral denominado de “*articlesController*” que disponibiliza um conjunto de funções que nos permite de forma fácil aceder ao conteúdos do módulo de artigo de texto, como é o caso da função “*show*” ilustrada na Figura 15 que devolve para a vista todos os detalhes de um determinado artigo de texto.

De igual modo existem controladores como o “*newsController*” e “*dynamicControllers*” que permitem obter as notícias e conteúdos dinâmicos respetivamente. O controlador de notícias destaca-se dois métodos principais, o “*index*”, que devolve todas as notícias publicadas e o “*show*”, que devolve todos os detalhes de uma determinada notícia. Além disso, o *Novo ORM*, conta com o método “*highlights*” no modelo “*News*” que recebe como parâmetro o tipo de ordenação e a quantidade pretendida a devolver e devolve a quantidade de notícias em destaque pretendida, paginadas e pela ordem indicada no parâmetro.

Em relação aos conteúdos dinâmicos, estes, contam com vários modelos, tais como:

- “*DynamicContentType*” – representa cada tipo de conteúdo dinâmico, e pode ter uma ou mais categorias de conteúdos dinâmicos;
- “*DynamicContentCategory*” – representa cada categoria do conteúdo dinâmico, corresponde a um tipo de conteúdo dinâmico e pode ter um ou mais itens. Esse modelo conta com o método “*byType*” que recebe um determinado tipo de conteúdos dinâmicos e devolve as respetivas categorias;
- “*DynamicContent*” - representa cada item dos conteúdos dinâmicos, corresponde a uma categoria de conteúdos dinâmicos e conta com métodos como o “*byType*” e o “*byCategory*” que recebe respetivamente o tipo e a categoria do conteúdo dinâmico e devolvem os itens para os respetivos tipos ou categorias.

```

use app\models as mdl;
class _articlesController extends articlesController
{
    function show()
    {
        $this->responds("ajax", "personal", "ajax", "html");

        $this->data["article"] = app\models\Article::find()
            ->whereif("id", is_numeric($this->params->id) ? $this->params->id : false )
            ->whereif("alias", !is_numeric($this->params->id) ? $this->params->id : false)
            ->one();

        if (!$this->data["article"]->id) {
            $this->data["partial"] = 'errorpage/show404';
        }
    }
}

```

Figura 15 - Função que devolve um determinado artigo de texto a vista

A seguir são apresentados os *websites* desenvolvidos durante o estágio para diferentes clientes da *humansoft*.

#### 4.2.1. Site da ETAP

O site da ETAP (Escola Tecnológica, Artística e Profissional de Pombal) foi desenvolvido totalmente do início e foi o que teve mais níveis elevados de dificuldades no desenvolvimento de certos componentes que o constitui, com o destaque no cabeçalho do *website*, pois foi o primeiro a ser desenvolvido durante o estágio, numa altura que ainda se encontrava na fase de adaptação.

Na Figura 16 (esquerda) podem ver o desenho inicial e na Figura 16 (direita) podem verificar o resultado do *website* depois de várias alterações pedidas pelo cliente.

Neste *website* possui uma área de *banner* onde destaca-se o desenvolvimento efetuado para conseguir o efeito com o vermelho nas suas imagens, na diagonal, além disso responsivo e sempre alinhado com a área com o *background* preta das notícias em destaque.

A área das notícias, assim como as áreas dos acionistas, foram desenvolvidas com base no *Smooth Div Scroll* que no fundo é um plugin *jQuery* que permite obter o efeito de rolagem de conteúdos horizontalmente para esquerda ou para direita [29].

Hoje o site encontra-se em produção e pode ser visitado neste link: <https://etap.edu.pt/>.

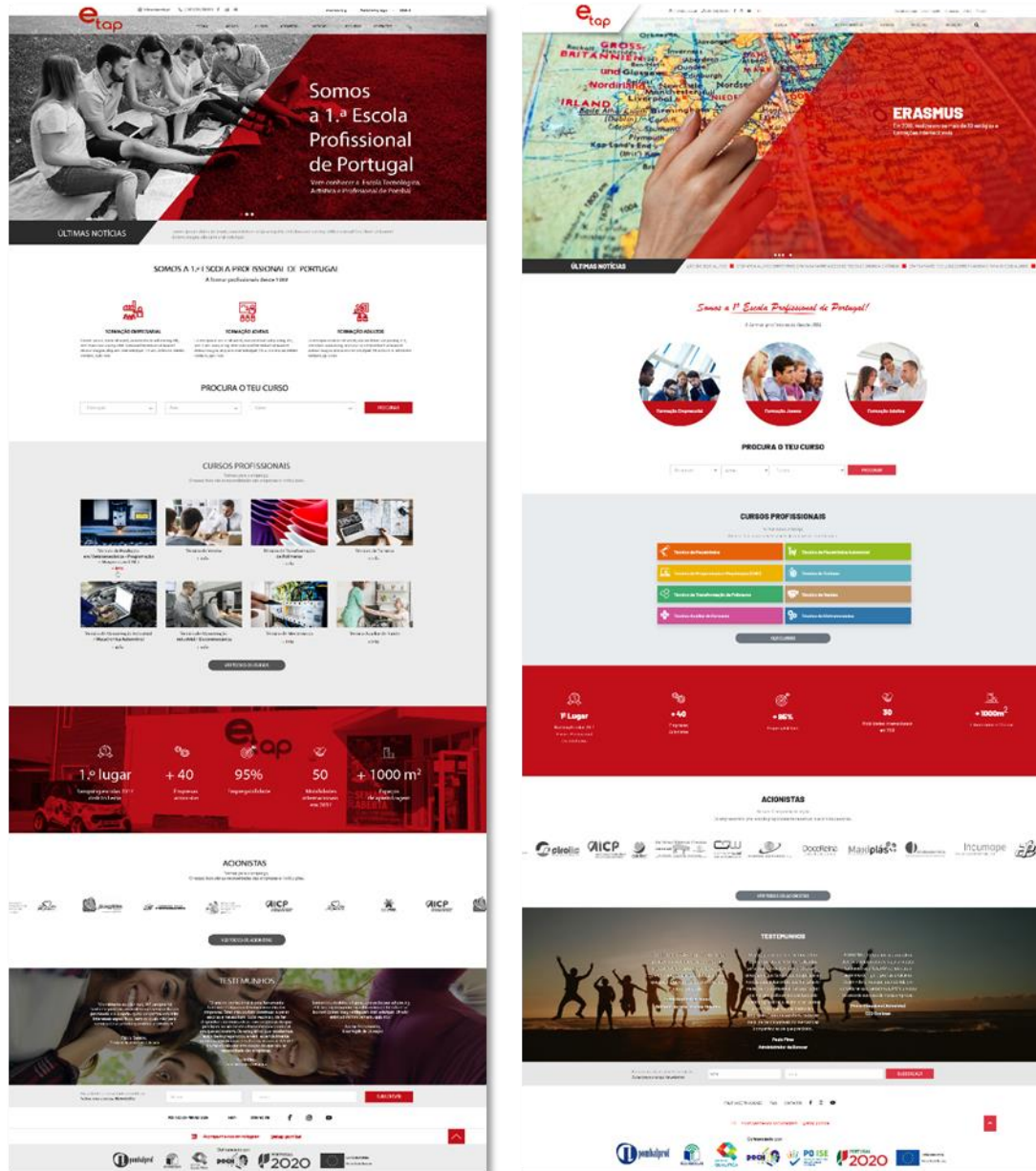


Figura 16 - Desenho inicial (esquerda) e resultado (direita) do site da ETAP

#### 4.2.2. Site do Cencal

O *site* do *Cencal* (Centro de Formação Profissional para a Indústria de cerâmica) foi a reconstrução do *website* antigo, tendo como base um novo desenho.

Na página inicial deste *website* destaca-se duas áreas, a área das formações e das notícias em destaque. Estas áreas foram conseguidas graças ao *slick slider* que é um *plugin jQuery* para criar carrosséis/*sliders* totalmente personalizáveis, responsivos e amigáveis para aplicações em HTML [30].

No *footer*, destaca-se também, uma área de imagens, desenvolvida através da integração com o *Instagram*. O desenvolvimento desta área consistiu em obter as últimas publicações do utilizador do *Cencal* no *Instagram* e disponibilizá-las no *website*. Para conseguir essas publicações foi utilizada a API da *Instagram* [31], na Figura 17 está um excerto de código em

PHP que possibilitou este desenvolvimento. Ainda nesta área foi utilizada o *Smooth Div Scroll* para conseguir o efeito de rolagem de conteúdos horizontalmente da direita para esquerda.

```
// get recent instagram publications of user
if (isset($_SESSION['images_instagram']) && !empty($_SESSION['images_instagram'])) {
    $images = $_SESSION['images_instagram'];
} else {
    $images = [];
    $link = "https://api.instagram.com/v1/users/self/media/recent?access_token=".INSTAGRAM_ACCESS_TOKEN;
    $result = sys\lib\Curl::exec($link);
    $results = json_decode($result);

    if (is_array($results->data) && count($results->data) > 0) {
        foreach ($results->data as $data) {
            $images[] = $data->images->standard_resolution->url;
        }
    }
    $_SESSION['images_instagram'] = $images;
}
```

Figura 17 - Obtenção das últimas publicações do utilizador de *Instagram*

Hoje o site encontra-se em produção e pode ser visitado no endereço <https://cencal.pt/>. Na Figura 18 está ilustrada a página inicial do site do *Cencal*.

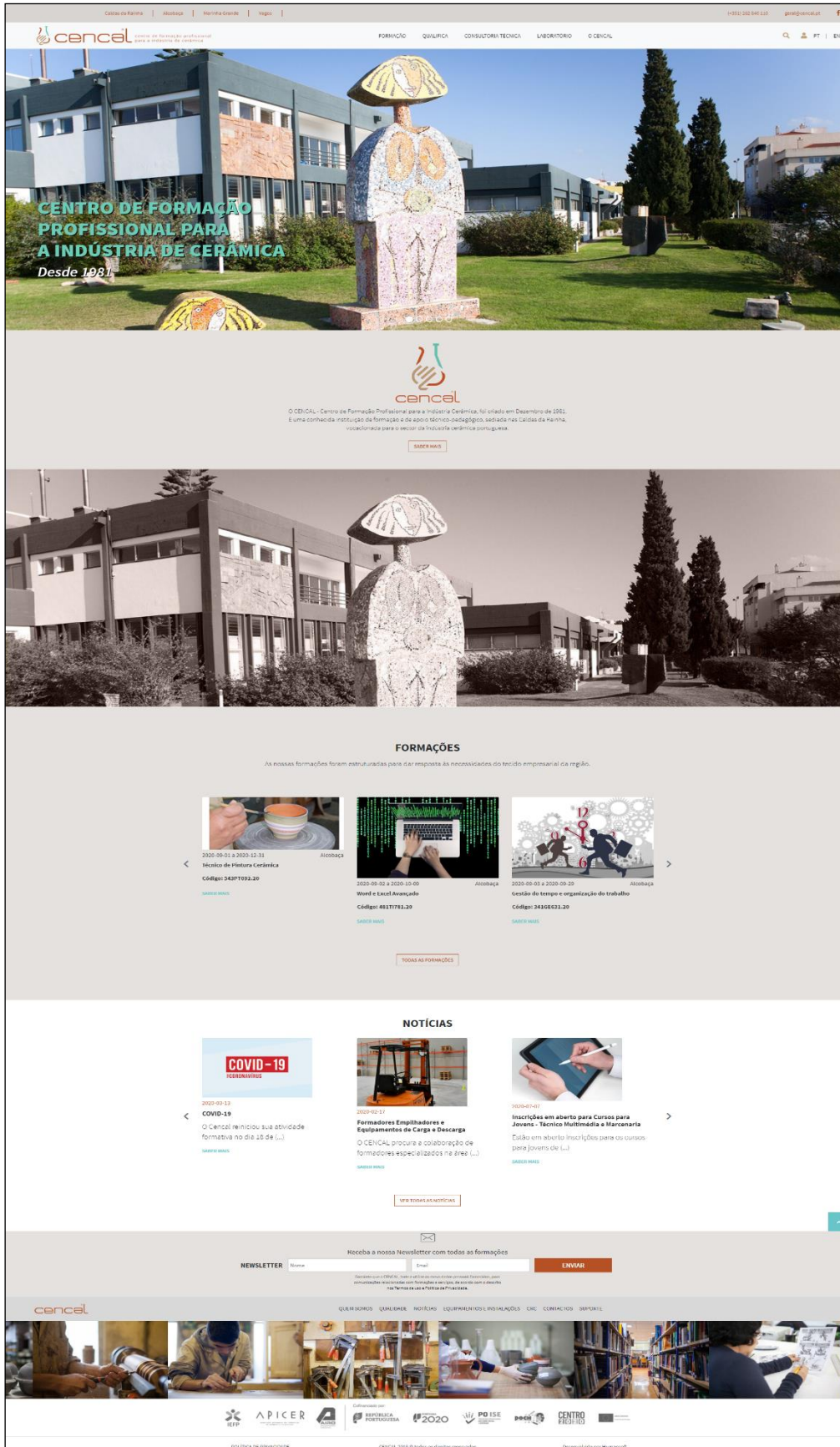


Figura 18 - Site Cencal página inicial

### 4.2.3. Site do *Cenfim*

O site do *Cenfim* (Centro de Formação Profissional da Indústria Metalúrgica e Metalomecânica) foi a tarefa que durou mais tempo, devido aos constantes pedidos do cliente em alterações de *layout* de quase todas as páginas.

Neste *website*, ilustrado na Figura 19, destaca-se na página principal, a área das notícias em destaque, que é uma área que foi desenvolvido com base no *plugin slick slider* que, como já foi dito anteriormente permite criar carrosséis/sliders totalmente personalizáveis, responsivos e amigáveis para aplicações.

Hoje o site encontra-se em produção e pode ser visitado neste link: <https://www.cenfim.pt/>.

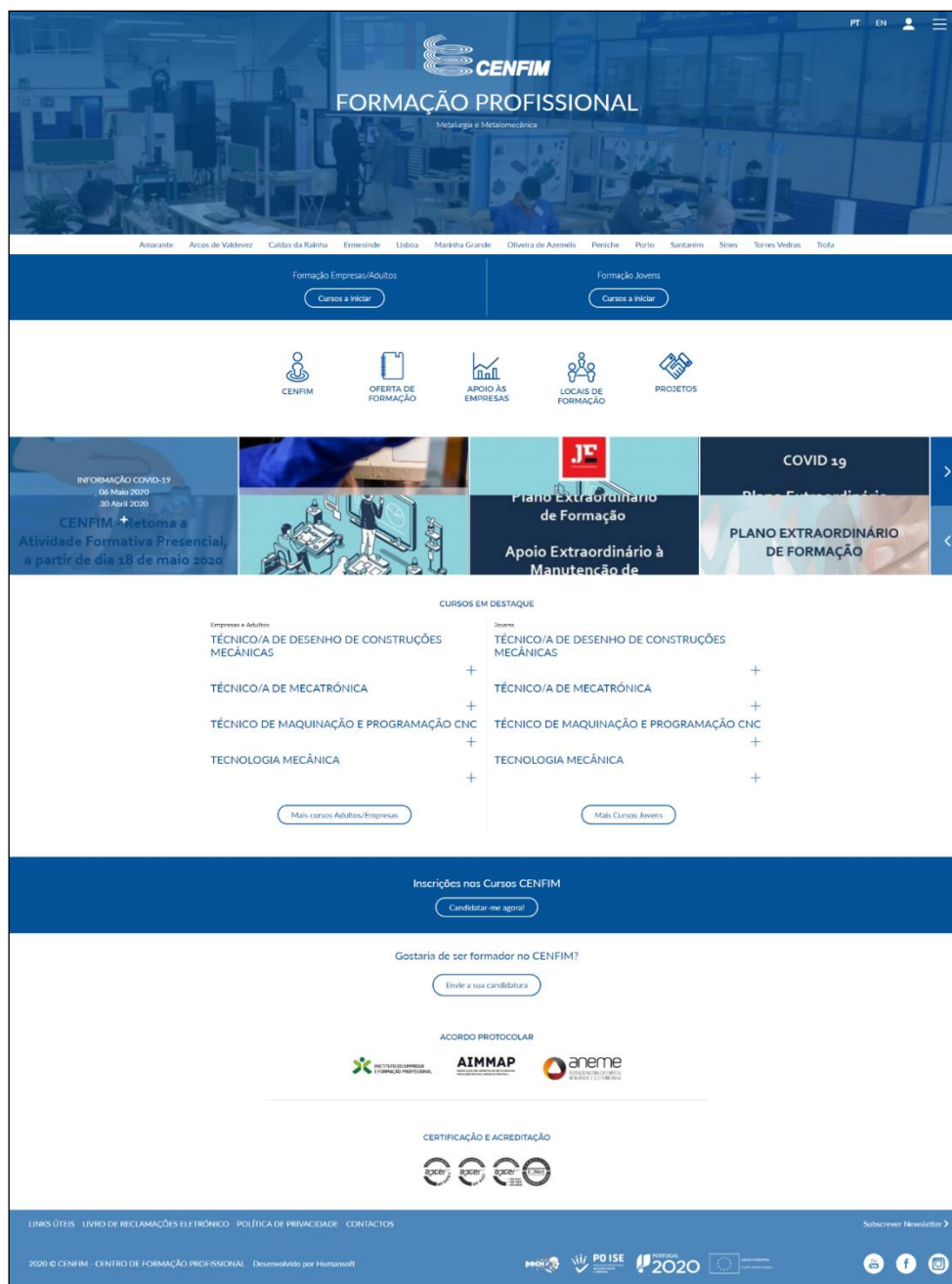


Figura 19 - Página inicial do Website *Cenfim*

#### 4.2.4. Site do CFPSA

O site do CFPSA (Centro de Formação Profissional para o Sector Alimentar), ilustrado na Figura 20, foi desenvolvido totalmente de raiz, no entanto, os controladores que devolviam os conteúdos à vista, foram reaproveitados, sendo que foi necessário ajustar e otimizar todos os controladores. Além disso foi necessário também a criação de alguns novos controladores específicos para o cliente.

Na página inicial destaca-se uma segunda área no cabeçalho, onde contém os *banners* dinâmicos, desenvolvidos com base no *carousel* do *Bootstrap 4*.

No conteúdo principal da página inicial destaca-se várias áreas, como área de acesso rápido aos tipos de formação, os cursos, e a área de membro (*humanportal*), a área dos cursos em destaques, notícias em destaques desenvolvidas tendo como base o *plugin slick slider*, multimédias e testemunhos, que no fundo são conteúdos dinâmicos do tipo multimédia e testemunhos respetivamente.

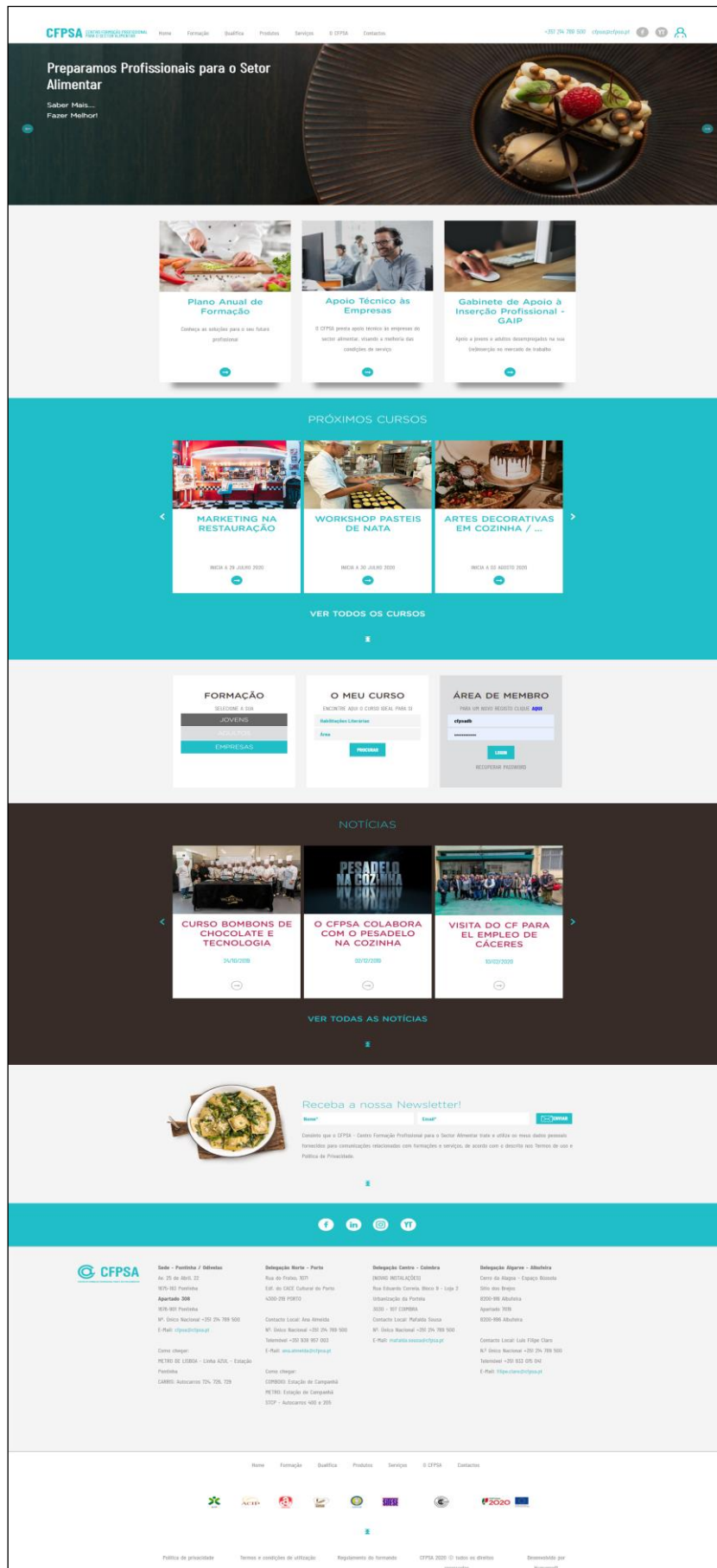


Figura 20 - Partes da página inicial do website CFPSA

Destaca-se ainda a página das galerias, ilustrada na Figura 21, que no fundo foi a página do *website* que mais dificuldades foram sentidas. Foi desenvolvida com base no *slick slider*, no entanto foi necessário muita adaptação do *plugin slick slider* para conseguir o resultado desejado pelo cliente. Hoje, o site encontra-se em produção e pode ser visitado neste link: <https://www.cfpsa.pt/>.

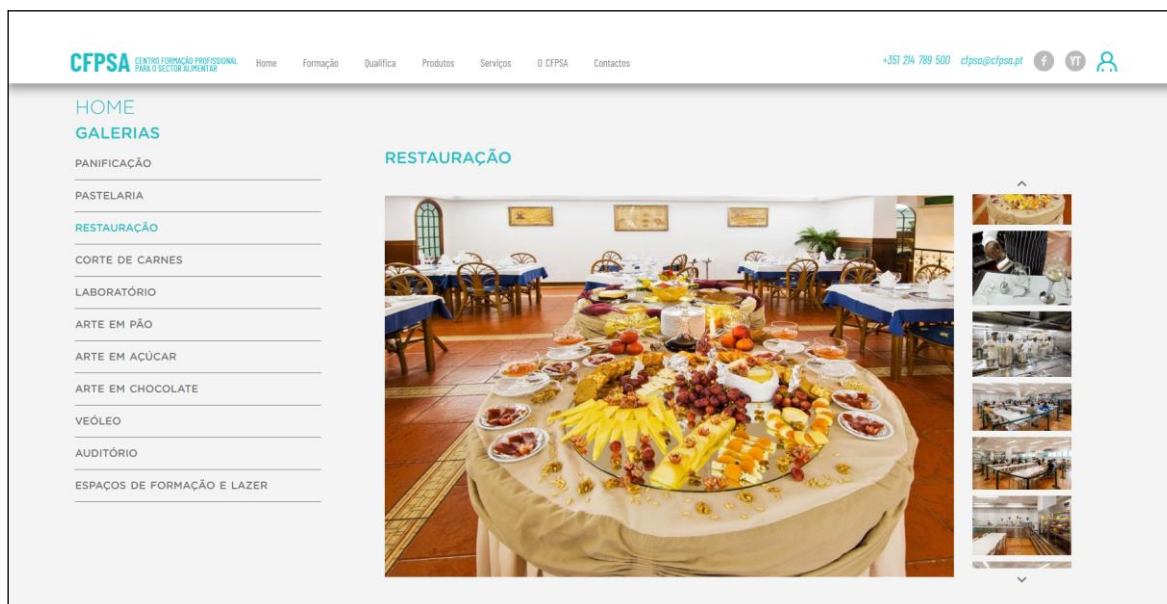


Figura 21 - Página das galerias do *website* do CFPSA

#### 4.2.5. Site do Modatex

No site do *Modatex* (Centro de Formação Profissional da Indústria Têxtil, Vestuário, Confeção e Lanifícios), desenvolvido igualmente de raiz (Figura 22), destacam-se os *banners* dinâmicos, desenvolvidos com base no *carousel* do *Bootstrap 4*. Ainda na página inicial, destaca-se o efeito *parallax*, que é uma técnica de *web design* onde o *background* se movimenta em um ritmo mais lento que o primeiro plano. Isso resulta em um efeito *3D* à medida que os visitantes rolam pela página, adicionando uma sensação de profundidade e criando uma experiência mais imersiva [32].

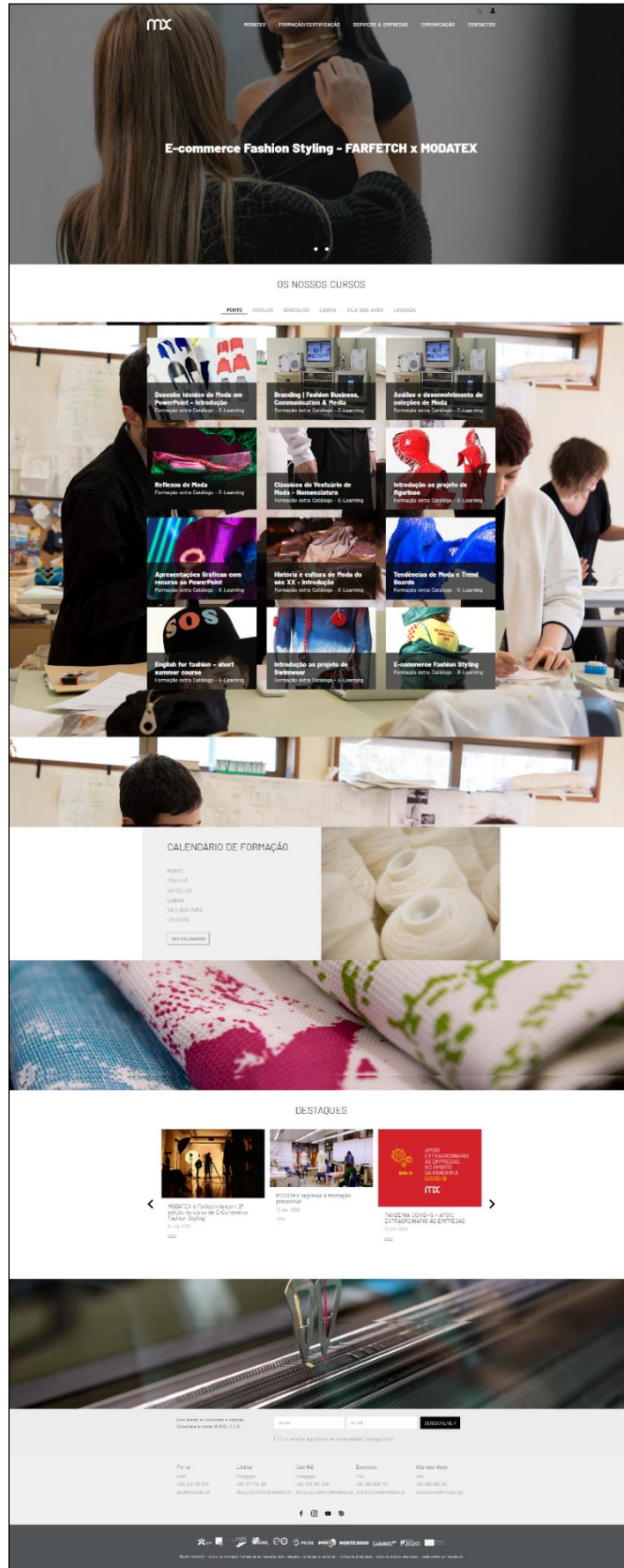


Figura 22 - Página inicial do website do Modatex

Para conseguir o efeito *parallax* foi desenvolvido uma classe de CSS com o nome de “*parallax*” (Figura 23), que foi utilizado posteriormente no conteúdo HTML (destaque na Figura 24) e foram efetuados alguns ajustes no *jQuery* e *JavaScript* para obter o movimento em velocidades diferentes (Figura 25).

```
.parallax {
  background-size: cover;
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: center bottom;
  padding-bottom: 400px;
}
```

Figura 23 - Classe de css - *parallax*

```
<!-- Calendário de formação -->
<div class="container-fluid mw-100 parallax" id="parallax1"
  style="background-image: url(<?=$parallax_imgs[1]->web ? : TEMPLATEDNS.'img/parallax1.jpeg' ?>);">
  <div class="container-fluid bck2">
    <div class="container">
      <div class="row">
        <div class="col-md-6 py-5 pl7 bck6">
          <p class="fs30 c3 mb-5 text-uppercase"><?=$formation_calendar->name; ?></p>
          <div class="c7 lh18 my-4"><?=$formation_calendar->high_text; ?></div>
          <div class="lh30">
            <?php foreach ($locals as $local): ?>
              <p><a class="c-link c7 text-uppercase" href="/courses/calendar.html?courseactions.1c
                <?php endforeach ?>
            </div>
          <a class="btn btn-c7 mt-4" href="/courses/calendar.html">VER CALENDÁRIO <i class="c7 la la-a
        </div>
      <div class="col-md-6 cover-bck calendar-img" style="background-image: url(<?=$formation_calenda
    </div>
  </div>
</div>
```

Figura 24 - Código HTML da área de calendário de formação com efeito to *parallax* aplicado

```
<script>
$(window).scroll(function() {
  var wScroll = $(this).scrollTop();
  let caption = $('#carousel-caption');
  if (wScroll > caption.offset().top - $(window).height()) {
    caption.css({'transform': 'translate(0px, '+ wScroll / 4 +'%)'});
  }
  let banner = $('#main-banner .carousel-item');
  if (wScroll > banner.offset().top - $(window).height()) {
    let offset = wScroll - banner.offset().top;
    banner.css({'background-position': 'center '+ (Math.abs(offset * 0.05))+'px'});
  }
  let parallax2 = $('#list-courses');
  if (wScroll > parallax2.offset().top - $(window).height()) {
    let offset = wScroll - parallax2.offset().top;
    parallax2.css({'background-position': 'center '+ (Math.abs(offset * 0.05))+'px'})
  }
});
</script>
```

Figura 25 - Efeito *parallax* com *jQuery*

Destaca-se ainda a página de calendário de formação, onde são apresentadas as formações no calendário desenvolvido com base no *FullCalendar* que é *plugin jQuery* para apresentar

dados no calendário [33]. Na Figura 26 encontra-se ilustrado uma parte do calendário de formação. Hoje, o site encontra-se em produção e pode ser visitado neste link: <https://www.modatex.pt/>



Figura 26 - Calendário de formação *Modatex*

#### 4.2.6. Site do *Cincork*

O site *Cincork* (Centro de Formação Profissional da Indústria de Cortiça), foi a reconstrução do site antigo, tendo como base um novo desenho. Na página inicial deste *website* (Figura 27) destaca-se uma área no cabeçalho, onde contém os *banners* dinâmicos, desenvolvidos com base no *carousel* do *Bootstrap 4*. Além disso, destaca-se ainda duas áreas no conteúdo principal desta mesma página, a área das formações e das notícias em destaque. Estas áreas foram conseguidas graças ao *slick slider*.

Apesar de *humanportal* possuir módulos que permitem inscrição de formandos em ações de formações e de formadores na bolsa de formadores e que facilitam a gestão desses processos, uma vez que essas inscrições são sincronizadas do portal para o *humantrain*, este cliente preferiu que os respetivos formulários dos módulos tivessem comportamento específico, ou seja, que esses dados fossem enviados apenas para os emails indicados.

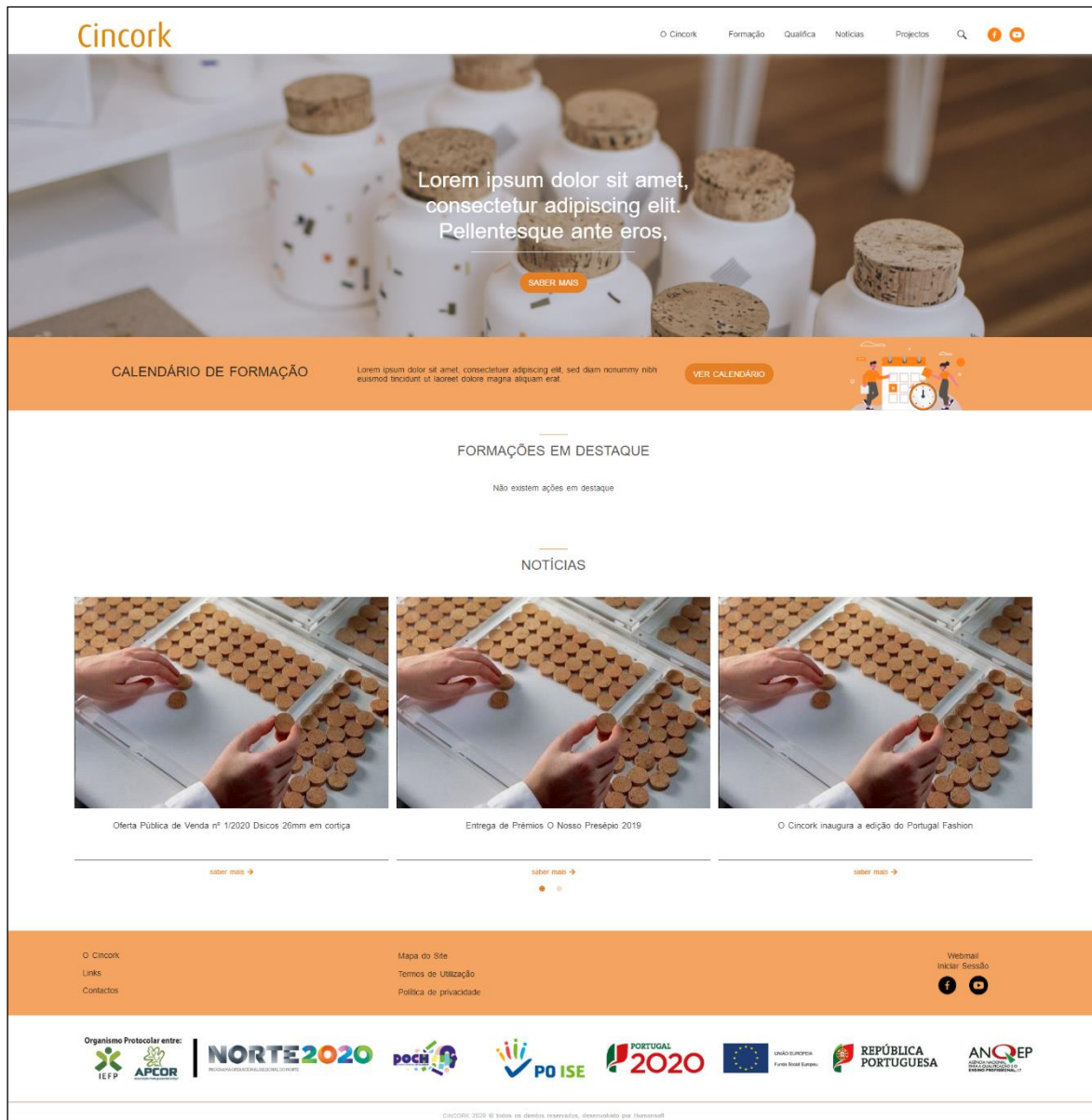


Figura 27 - Página inicial do website do Cincork

A semelhança do website do *Modatex*, este, também se destaca em possuir uma página da agenda, onde são apresentadas as formações no calendário desenvolvido também com base no *FullCalendar*. Na Figura 28 encontra-se ilustrado uma parte dessa página.

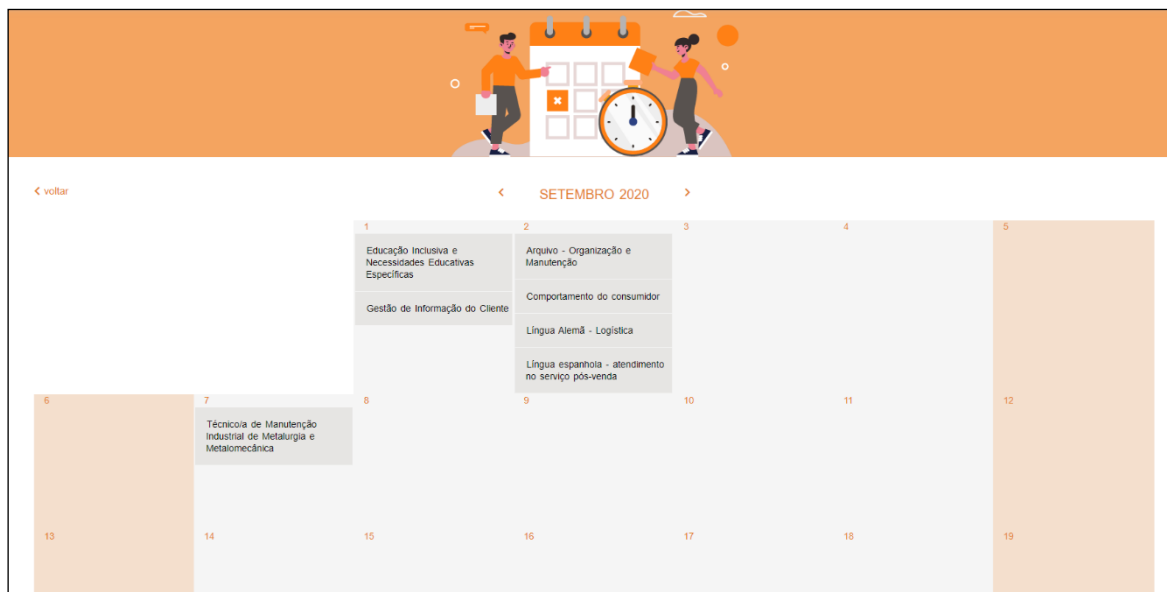


Figura 28 - Calendário de formação do *Cincork*

### 4.3. Componente “Área de Formação”

Este é um componente como já foi dito no capítulo Enquadramento, que torna possível a interação de todos os intervenientes da formação. Nesse sentido, foram desenvolvidos para este componente, alguns módulos novos, assim como reestruturação e adições de funcionalidades de módulos já existentes.

O *back-end* de todo o desenvolvimento a seguir, descritos, foram desenvolvidas em PHP utilizando a *framework* da *humansoft*, ou seja, o Novo ORM. Já o *front-end*, foram desenvolvidos na maioria em HTML, CSS e *Javascript*, e em alguns casos em *React.js*.

#### 4.3.1. Módulo “Ações do Coordenador”

Este é um módulo de consulta, onde permite ao coordenador visualizar todas as informações acerca da assiduidade e avaliação dos formandos para uma determinada ação.

O módulo, ilustrado na Figura 29, conta com uma área onde estão as informações sobre a ação e outra área onde encontra-se uma tabela horizontal. A tabela encontra-se dividida em duas principais colunas, a primeira onde estão os módulos e a segunda, onde encontram-se as informações das assiduidades e avaliações de cada formando em cada módulo. A primeira coluna é fixa enquanto que a segunda possui uma *scrollbar* na horizontal que permite visualizar os parâmetros de avaliação de todos os formandos para cada módulo.

Assiduidade / Aprovação																									
Curso/Ação Fotografia 18 19																									
18_19_a_f_18_19_a_1 [2018-10-01 - 2019-07-31] a Fare																									
Módulo	Nota	Obs.	F.J.	F.L.	%Assid.	Nota	Obs.	F.J.	F.L.	%Assid.	Nota	Obs.	F.J.	F.L.	%Assid.	Nota	Obs.	F.J.	F.L.	%Assid.					
1 - História de Fotografia	0.00	-	16	0	0%	15.00	-	0	0	100%	15.00	-	0	0	100%	15.00	-	8	0	50%	16.00	-	0	0	
Assiduidade / Pontualidade	0.00				4.28%	20.00				4.28%	0.00				4.28%	10.00				4.28%	20.00				4.28%
Domínio / Aplicação de Saberes	0.00				4.29%	15.00				4.29%	15.00				4.29%	15.00				4.29%	15.00				4.29%
Participação / Iniciativa	0.00				4.29%	12.00				4.29%	13.00				4.29%	12.00				4.29%	12.00				4.29%
Atitude / Comportamento	0.00				4.28%	20.00				4.28%	20.00				4.28%	20.00				4.28%	20.00				4.28%
Criatividade	0.00				4.28%	15.00				4.28%	15.00				4.28%	15.00				4.28%	15.00				4.28%
Capacidade de Expressão	0.00				4.29%	12.00				4.29%	16.00				4.29%	16.00				4.29%	15.00				4.29%
Autonomia	0.00				4.29%	20.00				4.29%	20.00				4.29%	20.00				4.29%	20.00				4.29%
Projeto de Módulo	0.00				70.00%	14.00				70.00%	15.00				70.00%	15.00				70.00%	15.00				70.00%
2 - A Máquina Fotográfica	11.00	-	4	0	83.3%	16.00	-	0	0	100%	13.00	-	0	0	100%	13.00	-	0	0	100%	15.00	-	0	0	4
3 - Iluminação	11.00	-	0	6	81.3%	15.00	-	0	0	100%	14.00	-	0	4	87.5%	14.00	-	0	2	93.8%	14.00	-	0	10	
4 - Intr. Fotografia Retrato	14.00	-	0	0	100%	15.00	-	0	4	87.5%	14.00	-	0	0	100%	15.00	-	0	0	100%	14.00	-	0	0	
5 - Pós Produção Fotográfica I (Photoshop)	1.00	-	0	15	53.1%	15.00	-	0	0	100%	15.00	-	0	0	100%	16.00	-	0	4	87.5%	13.00	-	0	0	
6 - Intr. à Fotografia Publicitária	9.00	●	0	0	100%	16.00	●	0	0	100%	15.00	●	0	4	87.5%	15.00	●	0	0	100%	14.00	●	0	4	
7 - Fotografia Criativa	2.00	●	0	12	50%	16.00	●	0	0	100%	14.00	●	0	4	83.3%	18.00	●	0	4	83.3%	16.00	●	0	4	
12 - Portfolio, Pitching e Empreendedorismo	5.00	●	8	4	62.5%	16.00	●	0	4	87.5%	13.00	●	0	24	25%	17.00	●	0	8	75%	14.00	●	0	20	
13 - Desenvolvimento	-	-	2	0	0%	-	-	0	0	100%	-	-	-	-	-	-	-	0	0	100%	-	-	0	0	
TOTAL	-	-	30	85.5	65%	-	-	0	12.5	96.2%	-	-	0	81.5	74.8%	-	-	8	62	78.8%	-	-	0	83.5	

Figura 29 - Módulo “Ações do Coordenador”

No *back-end*, foi utilizado o Novo ORM, para obtenção dos dados. Primeiramente obteve-se os formandos inscritos na ação, assim como seus dados pessoais a apresentar no ecrã, em seguida, os dados da ação, do curso e os módulos que o constituem. Seguidamente obteve-se as informações das avaliações e das assiduidades de cada formando para cada módulo. Esses dados foram enviados em apenas dois *arrays* associativos, os “*trainees*” que continha todas as informações do formando e o “*modules*” que continha os módulos e as informações das avaliações e assiduidades. Na Figura 30 está um excerto do código que devolve os dados para preenchimento da tabela horizontal.

```

$courseaction_id = $this->params->id ?: 0;
$strainees = app\models\Entity::find()->alias('en')
->select('u.name, en.entity_code, en.table, en.versao_rowid, u.gender')
->select("case when m.disapproved_absences = 1 then 'r_faltas' else '' end as status")
->select("case when m.canceled = 1 then 'quit' else '' end as status")
->join("matriculations m", "en.entity_code = m.entity_code and en.group_id = 3")
->join('users as u', 'u.id = en.user_id')
->where('m.courseaction_id', $courseaction_id)
->where('u.name is not null and m.confirmed = 1')
->orderBy("u.name asc")
->get();

$courseaction = app\models\CourseAction::where('courseaction_id', $courseaction_id)->one();
$course = $courseaction->parent('Course')->one();

$modules = app\models\CourseModule::find()->alias('cm')
->select('cm.id, cs.coursemodule_id, cm.level, concat(cm.level, " - ", cm.module) as module')
->join('coursesections as cs', 'cs.coursemodule_id = cm.id')
->where('cs.courseaction_id = :courseaction_id', [':courseaction_id' => $courseaction_id])
->groupby('cm.id')
->orderBy('nivel(cm.level), cm.level asc')
->map('coursemodule_id');

$modules[0] = app\models\CourseModule::create([
    'id' => 0,
    'coursemodule_id' => 0,
    'level' => 0,
    'module' => 'TOTAL'
]);

$evaluations = app\models\EvaluationFinalTotal::find()->alias('eft')
->select('eft.coursemodule_id, round(eft.average, 2) as classification, eft.trainee_code as entity_code, eft.obs')
->join('evaluationfinal as ef',
    'ef.courseaction_id = eft.courseaction_id and
    ef.trainee_code = eft.trainee_code and
    ef.level = eft.level'
)
->join("entities en", "en.entity_code = eft.trainee_code")
->join("users u", "u.id = en.user_id")
->where('eft.courseaction_id', $courseaction_id)
->get();

foreach ($evaluations as $eval) {
    $modules[$eval->coursemodule_id]->addChildren($eval, 'trainees', $eval->entity_code);
}

```

Figura 30 – *Novo ORM* - Excerto de código do controlador que devolve os dados para preenchimento da tabela horizontal no módulo Ações do Coordenador

O *front-end* deste módulo foi desenvolvido em *React.js*, permitindo aos utilizadores terem uma experiência na consulta e análise das avaliações e assiduidades dos seus formandos num único ecrã e de uma forma estruturada, apelativa e fluída. Conta ainda com uma permissão para que sejam dadas apenas aos coordenadores dos clientes da *humansoft* que o adquirem.

O *React.js* além de conseguir um resultado de excelência, com uma presença muito forte em termos de usabilidade e fluidez, permitiu um desenvolvimento efetivo deste módulo, com menos codificação quando comparado com um desenvolvimento em *JavaScript* e *jQuery*, o que torna a sua manutenção mais simples. Na Figura 31 está um trecho do código desenvolvido em *React.js* que faz o render da tabela horizontal.

```

render() {
  // percorrer todos os módulos
  const table_body = hp.store.modules.map((module, i) => {
    // percorrer todos os formandos
    const tds = hp.store.trainees.map((trainee, j) => {
      const evall = module.trainees ? module.trainees[trainee.entity_code] : null;
      var assid = "-";
      var just = "-";
      var injust = "-";
      var search = "";
      const status = module.coursemodule_id != 0 ? trainee.status : "";
      const params = {
        title: __('attrs_details')
      }

      if(hp.store.assids && hp.store.assids[module.coursemodule_id] && hp.store.assids[module.coursemodule_id][trainee.entity_code]){
        const value = hp.store.assids[module.coursemodule_id][trainee.entity_code];
        if(value && value.assid >= 0){
          assid = value.assid + __('attrs_percent') + " ";
          search = < i className="fa fa-search-plus" onClick={() => Modal.show(
            <AbsenteeismDetails
              coursemodule_id={module.coursemodule_id}
              entity_code={trainee.entity_code}
            />,
            params
          )} title={__( 'attrs_details_attendance')} >
        </i>
        } else {
          assid = "-";
          search = ""
        }
        just = value.just;
        injust = value.injust;
      }
      const td_obs = evall && evall.obs ? <em className="fa fa-info-circle" title={evall.obs} /> : "-";
      return [
        <td key={'a' + j}>{evall && evall.classification ? evall.classification : "-"}</td>,
        <td key={'b' + j}>{td_obs}</td>,
        <td key={'c' + j}>{just}</td>,
        <td key={'d' + j}>{injust}</td>,
        <td className='ht-separator right' key={'e' + j}>
        { assid }
        { search }
      </td>
    ]
  })
}

```

Figura 31 - *React.js* - Trecho de código que faz o render da tabela horizontal no módulo “Ações do Coordenador”

### 4.3.2. Exportação do “Cronograma para *Google Calendar*”

A exportação do cronograma (ilustrada na Figura 32) para google calendar é uma funcionalidade que se foi desenvolvida no módulo do “Cronograma” do *humanportal* com o objetivo de exportar as sessões do utilizador para o calendário da *google*. O utilizador tem a possibilidade de escolher as datas em que pretende exportar, assim como selecionar os diferentes grupos a que pertença para exportação das suas respetivas sessões.

O *front-end* foi desenvolvido em HTML e CSS. No entanto a *modal* faz parte do *template* do *humanportal*. Sendo que apenas foi necessário definir o título, o conteúdo principal e a ação para o botão “confirmar”.

O *back-end* foi desenvolvido com base no Novo ORM e *Google Calendar API*, que no fundo é uma API desenvolvida pela *google* que permite obter, criar e modificar eventos de calendário da *google*, bem como trabalhar com muitos outros objetos relacionados ao calendário, como por exemplo controlos de acesso. [34]

O utilizador da primeira vez que exporta as sessões, depois de confirmar a exportação no *humanportal* é redirecionado para uma página da *google* onde terá de iniciar sessão com a conta da *google* a que se deseja importar as sessões e em seguida permitir que o *humanportal* faça alterações ao calendário da *google*, após isso as sessões são importadas no calendário da *google* e são possíveis visualizá-las um novo calendário no calendário da *google*, menu esquerdo com o nome “Portal” com a cor vermelha.

Para conseguir esta funcionalidade, foi utilizada a classe “*Google\_Client*”, com as configurações (obtidas na conta do *developer* da *google*) específicas para cada cliente da *humansoft*. É feito um pedido do *token* de acesso a conta *google*, caso seja a primeira vez a efetuar a exportação ou caso o *token* esteja expirado. É obtida as sessões conforme os parâmetros selecionados pelo utilizador e caso tenha sessões é criado/atualizado um evento no calendário do *google* para cada sessão. Além disso é registada na base de dados do *humanportal* as informações dos eventos que foram criados/atualizados no calendário do *google*. Esta funcionalidade apenas está disponível nos portais de clientes que tenham a respetiva permissão e do ficheiro de configuração específico de *Google Calendar API*.

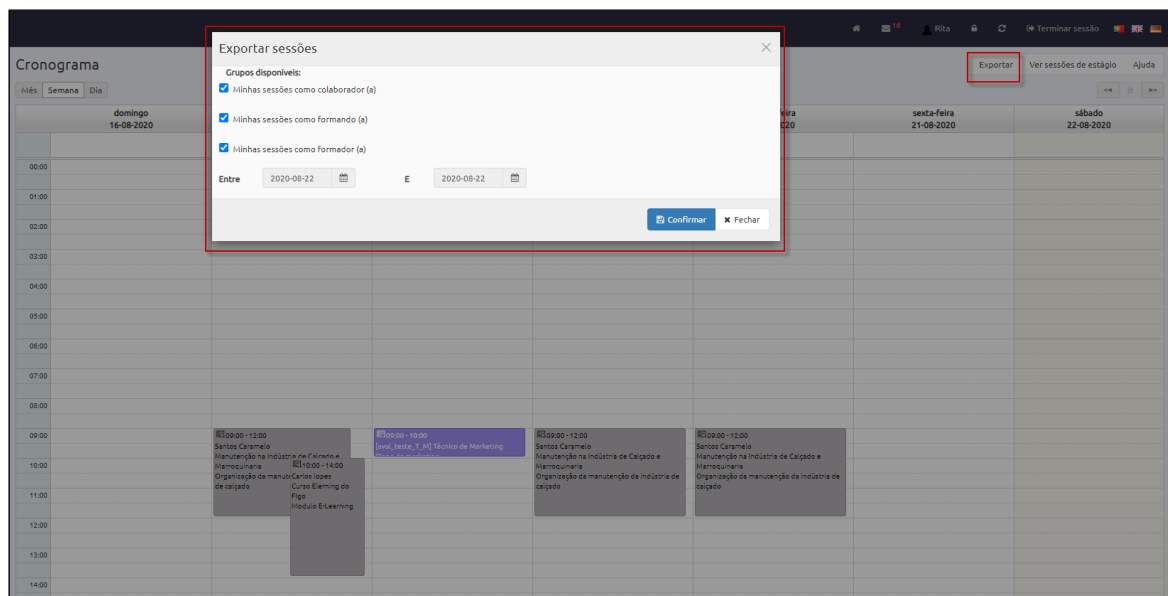


Figura 32 - Módulo cronograma- funcionalidade para exportar sessões

### 4.3.3. Reconstrução do módulo de “Testes”

O módulo de testes é um módulo que permite aos formadores/técnicos responsáveis/coordenadores de cada ação efetuarem toda a gestão dos testes, desde a criação até a correção. Além disso permite aos formandos responderem ao teste assim como visualizar a correção e a avaliação obtida. Este módulo está composto por dois submódulos denominados de “Banco de Perguntas” e “Banco de Testes”.

O “Banco de Perguntas” é constituído apenas por um ecrã (Figura 33), onde os intervenientes da ação podem criar perguntas, associar respostas possíveis, atribuir cotações, associar a pergunta a ação, ao tipo de curso, adicionar áudio/vídeo. Essas perguntas estarão disponíveis para os respetivos intervenientes no “Banco de testes”.

#	Pergunta	Nº de respostas possíveis	Áreas de formação	Cursos
128	Introduza uma descrição	Resposta Aberta		
127	Introduza uma descrição	Resposta Aberta		
126	Quanto é 2 + 2	3		
117	Quanto é 1 + 1	3		
116	Introduza uma descrição	1		
113	Esta pergunta tem um vídeo	Resposta Aberta		
112	Introduza uma descrição	3		
111	Como se chama o guarda redes da equipa principal da seleção nacional?	Resposta Aberta		
110	Qual é o nome completo do futebolista português conhecido como CR	Resposta Aberta		• N.º do T. 10309

**Figura 33 - Submódulo “Banco de Perguntas”**

O “Banco de Testes”, está constituído por 3 ecrãs, o primeiro com o mesmo nome desse submódulo (Figura 34) onde os intervenientes responsáveis da ação, criam os testes, associam as perguntas aos testes, e por fim associa-os as determinadas ações/cursos/áreas/tipos de cursos. Ainda neste ecrã, os responsáveis têm a possibilidade de alterar/apagar os testes, assim como pré-visualizar o teste criado.

#	Nome	Descrição	Criado em	Tipos de Curso	Áreas de formação	Cursos	Ações de formação
42	teste msousa 1	teste msousa	2020-07-17 15:51:47			• Formação para 6.2.0.4	
35	Teste modelo	teste modelo description	2020-07-11 14:13:45			• Formação para 6.2.0.4	
29	Teste para audio/vídeo	Teste para testar a inserção de audio e vídeo	2020-05-26 12:31:16				
26	Primeiro teste para versão 6.3.3.3	msousa teste da versão 6.3.3.3. Feito para responder em 5 minutos e sem consulta	2020-05-22 12:49:49			• [10309-T-HT]	
23	Msousa Teste com pergunta de texto livre	Novo teste para testar o comportamento de teste com perguntas apenas de texto livre	2020-05-07 18:35:05			• [10299-T-P80]	
15	My first test	msousa tests	2020-04-20 17:59:50			• Formação para 6.2.0.4	

**Figura 34 – Módulo de testes - Banco de testes**

O segundo ecrã (Figura 35), denominado de “Testes”, onde os formadores acedem aos testes associados anteriormente a uma determinada ação no “Banco de testes”. Este ecrã, permite ao formador da ação, aplicar o teste a ação (fica disponível para formandos responderem), visualizar o teste, alterar por exemplo a data que o teste irá estar disponível, ou o número de perguntas a apresentar no teste e se aleatório ou não. Além disso, permite ao formador, eliminar o teste (continua no banco de teste, mas fica indisponível para formandos), aceder as respostas dos formandos aos testes, onde terá a possibilidade de visualizar a nota, assim como corrigir ou reabrir teste para formandos.

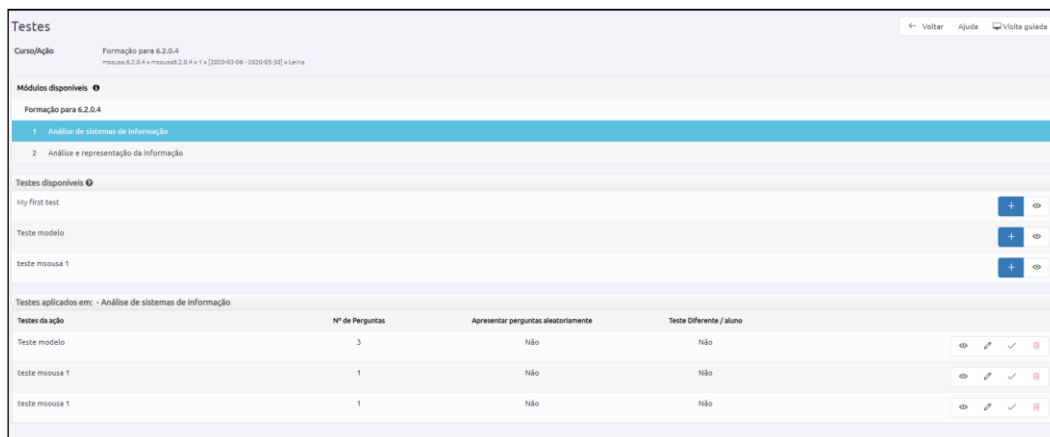


Figura 35 – Módulo de testes - ecrã de testes da ação do formador

Por fim o terceiro ecrã (Figura 36), onde os formandos têm acesso aos testes disponíveis, e têm a possibilidade de responderem aos testes, assim como visualizar as notas de cada teste.



Figura 36 - Módulo de testes – ecrã de testes da ação dos formandos

Todos esses ecrãs descritos anteriormente, foram reestruturados. Destacam-se, as seguintes tarefas efetuadas:

- No “Banco de Testes” foram efetuadas as alterações para que cada um tivesse acesso apenas os seus testes; ainda nesse ecrã, foram feitas alterações para que ao associar um teste a uma determinada ação, apresentasse apenas as ações em que o utilizador está negociado;
- No “Banco de Perguntas” foram efetuadas as alterações para que cada um tivesse acesso apenas as suas perguntas;
- No “Banco de Perguntas” foram efetuadas alterações para criar perguntas do tipo vídeo/áudio (associar ficheiro de vídeo/áudio);
- No “Banco de Testes” foram efetuadas as alterações para permitir associar perguntas do tipo vídeo/áudio aos testes;
- No ecrã onde os formandos respondem os testes foram efetuadas as alterações para apresentar os vídeos/áudio e permitir reprodução dos mesmos;
- O ecrã dos testes disponíveis do formador/formando, foi totalmente reescrito, pois inicialmente os testes apenas poderiam ser aplicados a ação. Com o novo desenvolvimento, os testes passaram ser aplicados ao nível da ação ou a um

determinado módulo. Na Figura 37 e na Figura 38 estão ilustrados, este ecrã antes e depois da reestruturação respetivamente;

- No ecrã de respostas dos formandos aos testes, foi adicionada a paginação e pesquisa;
- Foi desenvolvido o sistema de alerta para envio de mensagem sempre que há um novo teste disponível para responder.

O *front-end* dos ecrãs citados anteriormente foram desenvolvidos em HTML, CSS e JavaScript. Já o *back-end* desses ecrãs foram desenvolvidos em PHP, com o recurso ao *Novo ORM*.



Figura 37 – Módulo de “Testes” antes da reestruturação

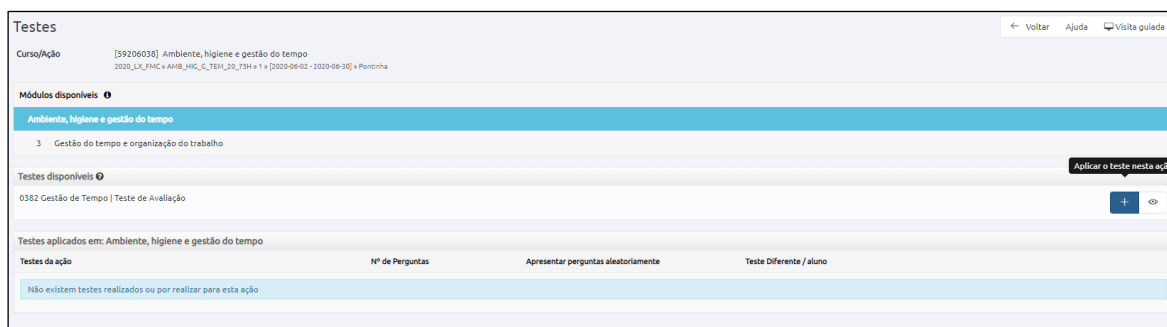


Figura 38 -Módulo de “Testes” depois da reestruturação

#### 4.3.4. Reconstrução do módulo de “Trabalhos”

O módulo de trabalhos é um módulo que permite aos formadores/técnicos responsáveis/coordenadores de cada ação efetuarem toda a gestão dos trabalhos, desde a criação até a correção. Além disso permite aos formandos entregarem os trabalhos assim como efetuarem o download do trabalho entregue e das correções de trabalho.

Este módulo foi totalmente reescrito, pois inicialmente os trabalhos apenas poderiam ser aplicados a ação. Com o novo desenvolvimento, os trabalhos passaram ser aplicados ao nível da ação ou a um determinado módulo. Na Figura 39 e na Figura 40 está ilustrado este ecrã, antes e depois da reestruturação.

O *front-end* dos ecrãs citados anteriormente foram desenvolvidos em HTML, CSS e JavaScript. Já o *back-end* desses ecrãs foram desenvolvidos em PHP, com o recurso ao *Novo ORM*.

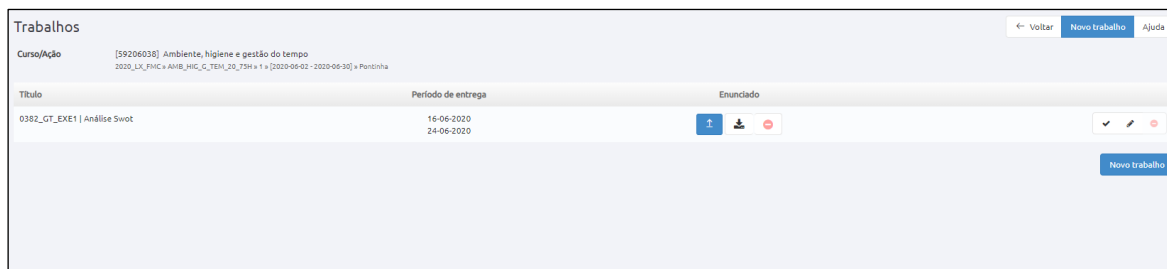


Figura 39 - Módulo de “Trabalhos” antes da reestruturação

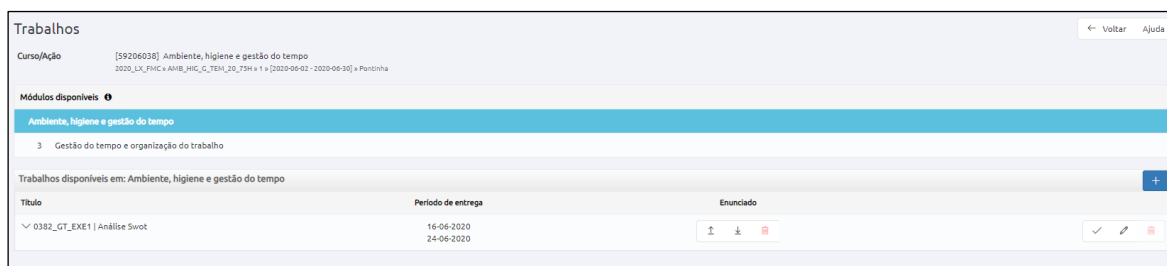


Figura 40 – Módulo de “Trabalhos” depois da reestruturação

## 4.4. Componente “Área reservada à organização”

Este é um componente que, como já foi dito no capítulo Enquadramento, permite criar módulos específicos para as organizações. Assim sendo, foram desenvolvidos para este componente, alguns módulos novos, assim como reestruturação e adições de funcionalidades de módulos já existentes.

### 4.4.1. Single Sign On (SSO) - Espaço Visual

Este módulo foi desenvolvido ao pedido do cliente *Espaço Visual*, e permite aos utilizadores autenticados no *humanportal* dos *Espaço Visual* autenticarem-se no site do espaço visual (*Opigno LMS*) automaticamente.

Através do serviço de *cronjob* da *humansoft*, foi criada o método “*registerNewUser*”, que é chamada através do “*observable*” sempre que é efetuada uma nova inscrição numa ação. Este método recebe como parâmetro o “*id*” da inscrição e devolve *true* ou *false*, conforme teve sucesso ou não ao registar o utilizador no site do *Espaço Visual*. Com o “*id*” da inscrição, é feita uma *query* através do Novo ORM, para obter os dados do formando inscrito, a seguir com a identificação do formando, obtém-se todas as inscrições do mesmo. Por fim, é efetuado um primeiro pedido para obtenção do *token* de autenticação ao site do *Espaço Visual*, e em seguida, são enviados os dados do formando, como email e nome, e todas as inscrições do mesmo para o site do *Espaço Visual*. Caso o formando já esteja registado no site do *Espaço Visual* então apenas são atualizadas as inscrições do mesmo (Figura 41).

```

static function registerNewUser($module, $matriculation_id)
{
    // obtém o formando através do id da inscrição
    $trainee = self::getUserInfo($matriculation_id);
    Log::file("single-sign-on", __FILE__, __LINE__, "Início de registo da inscrição - matriculation_id: " . $matriculation_id . " email: " . $trainee->email . " name: " . $trainee->name . " entity_code: " . $trainee->entity_code);
    if ($trainee->entity_code || $trainee->entity_id) {
        if ($trainee->email) {
            // obtém todas as inscrições do formando
            $courseaction_ids = self::getUserMatriculations($trainee);
            if (count($courseaction_ids) > 0) {
                $link_to_access_token = URL_TOKEN;
                //obtem o token de autenticação
                $token = Curl::exec($link_to_access_token);
                if (!empty($token)) {
                    $url = URL_NEW_USER;
                    $payload = [
                        "_links": [
                            "type": {
                                "href": "http://www.espaco-visual.pt/rest/type/user/user"
                            }
                        ],
                        "name":[{"value":"" . $trainee->name . ""}],
                        "mail":[{"value":"" . $trainee->email . ""}],
                        "status":[{"value": "1"}],
                        "field_courses_ids":[{"value":"" . implode(", ", $courseaction_ids) . ""}]
                    ];
                    $header = [
                        'Accept: application/json',
                        'Content-Type: application/hal+json',
                        'X-CSRF-Token: ' . $token
                    ];
                    // pedido de registo do novo user e atualização das inscrições atuais
                    $response = self::executeCurl($url, $payload, $header);
                    //se a conta ja existe entao, atualiza as inscrições do formando
                    if ($response->status_code == 422) {
                        self::updateUser($trainee->email, $courseaction_ids);
                    }
                    Log::file("single-sign-on-success", __FILE__, __LINE__, "Registo do utilizador " . $trainee->name . " para o single sign on com sucesso! Ação: " . implode(", ", $courseaction_ids));
                    return true;
                }
            }
        }
    }
}

```

Figura 41 - SSO - trecho de código do registo de novo utilizador

Nas “Ações de formando” foi desenvolvida uma nova opção “*Elearning*” que permite aos formandos tirarem partido do desenvolvimento acima descrito. Esta opção é apenas um endereço, que contém como parâmetro o “*id*” da ação. Ao clicar no endereço é chamado método “*singleSignOn*” no controlador “*\_elarningsController*” específico do *Espaço Visual*. Neste método, é obtido novamente o *token* de autenticação no site do *Espaço Visual*, e com este, é feito um novo pedido de endereço para redirecionamento do formando para a sua área no site do *Espaço Visual* para a ação selecionada, em caso de sucesso, caso contrário, e se o formando ainda não estiver autenticado é feito o registo do formando e das suas inscrições, e por fim é efetuado novamente o pedido do endereço de direcionamento para o site (Figura 42).

```

class _elearningsController extends elearningsController
{
    function singleSignOn()
    {
        $this->responds("html", "xml", "ajax", "ajax");
        if ($this->session->ses && in_array(3, $this->session->ag)) {
            $entity = $this->session->entity(3, true);
            $email = $entity->email1 ?: $entity->email2;
            $courseaction_id = app\models\Courseaction::find($this->params->id)->one()->coursea
            if (!empty($email)) {
                $payload = '{
                    "email":"' . $email . '",
                    "course_id":"' . $courseaction_id . '"
                }';
                $link_to_access_token = URL_TOKEN;
                $token = Curl::exec($link_to_access_token);
                if (!empty($token)) {
                    $url = URL_SSO;
                    $header = array(
                        'Accept: application/json',
                        'Content-Type: application/json',
                        'X-CSRF-Token: ' . $token
                    );
                    $response = $this->executeCurl($url, $payload, $header);
                    if ($response->status_code == 404) {

```

Figura 42 - SSO - trecho de código que devolve o endereço de direcionamento para site do Espaço Visual

#### 4.4.2. Single Sign On (SSO) – Moodle

Este módulo foi desenvolvido ao pedido do cliente *AMA* (Agência para a Modernização Administrativa), e permite aos utilizadores autenticados no *humanportal* da *AMA* autenticarem-se no *moodle* da *AMA* automaticamente.

Para conseguir este módulo foram necessárias configurações do lado do *moodle*, assim como desenvolvimento do lado do *humanportal*.

Do lado do *moodle*, foram efetuadas as seguintes tarefas:

- Análise do *plugin* que mais se encaixava nas necessidades da *humansoft*. Após esta análise, foi selecionado o *plugin User key authentication* [35], que permite autenticar-se no *moodle*, utilizando um endereço de *login* baseada em chave de única de utilizador.
- Instalação do *plugin* selecionado na plataforma *moodle* no menu “Administração do Site” e de seguida no submenu “Módulos”.
- Ativação do *plugin* e as respetivas configurações. Destaca-se a configuração do campo “*Mapping field*”, onde foram definidas as chaves únicas (*username*, *email* ou *idnumber*) permitidas para autenticação.
- Criação do serviço externo com o nome *humansoft*, que permite a comunicação entre o *humanportal* e o *moodle*.
- Associação do utilizador de *humansoft* a lista dos utilizadores autorizados do serviço externo *humansoft*.
- Associação da função “*auth\_userkey\_request\_login\_url*” ao serviço externo.

Do lado do *humanportal*, foram efetuados os desenvolvimentos de forma a que pudesse ser utilizado para futuros clientes. Nesse sentido, foi criada uma opção (Figura 43) no “*header*” do *humanportal*, através de html e CSS, onde apenas está visível para clientes que tenham permissões de acesso ao *moodle*.

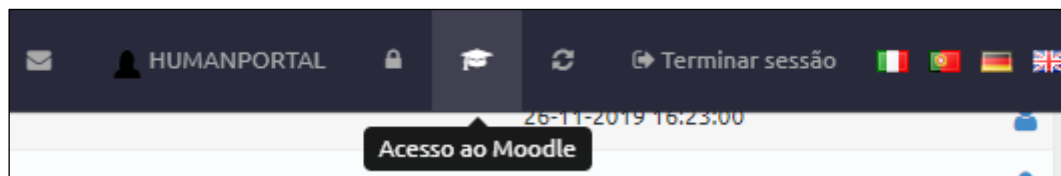


Figura 43 - SSO Moodle - opção para aceder ao moodle

Associada a esta opção, foi criado um controlador geral com o nome “*moodleController*”, onde foram criados vários métodos, dos quais destaca-se o “*singleSignOn*”, que permite autenticar o utilizador do *humanportal* no *moodle*. Este método utiliza as credenciais de administrador de *humansoft* no *moodle* para obter um *token* através do serviço externo do *humansoft* criado no *moodle*. Após isso é feito novamente um pedido ao *moodle*, através de *webservice*, onde são passados os parâmetros como, o “*token*”, a função “*auth\_userkey\_request\_login\_url*” e o valor (*username*, *email* ou *idnumber* do utilizador autenticado) consoante o “*Mapping field*” configurado no *plugin User key authentication* e em caso de sucesso é retornado o endereço que posteriormente irá redirecionar o utilizador para o *moodle*. Na Figura 44 está ilustrada um trecho de código deste método.

```

* E redireciona para a página de um curso em específico ou para a área do utilizador
* MOODLE_MAPPINGFIELD: campos em que é possível a autenticação, utilizado para encontrar o utilizador no moodle
*/
function singleSignOn() {
    $this->responds("html", "xml", "ajax", "ajax");
    if (defined('MOODLE_URL') && defined('MOODLE_MAPPINGFIELD') && $this->session->ses) {
        if ($token = self::getToken()) {
            $entity = $this->session->entity(3, true);
            switch (MOODLE_MAPPINGFIELD) {
                case 'username':
                    $value = $this->session->user(TRUE)->username;
                    break;
                case 'idnumber':
                    $value = $entity->entity_code ? $entity->entity_code_2;
                    break;
                default:
                    $value = $entity->email ? $entity->email;
                    break;
            }
            $params = [
                'wstoken' => $token,
                'wsfunction' => "auth_userkey_request_login_url",
                'moodlewsrestformat' => 'json',
                "user[\".MOODLE_MAPPINGFIELD.\"]" => $value,
            ];
            $url = MOODLE_URL.'webservice/rest/server.php';

            $response = json_decode(Curl::exec($url, $params));
            if ($response->loginurl) {
                $urlToGo = $response->loginurl;
                $moodle_course = $this->params->id ? self::getCourse($url, $token, $this->params->id) : false;
                /* Se foi encontrado o curso, então redireciona para a página do curso em específico, caso contrário redireciona para a área de utilizador*/
                if ($moodle_course->id) {
                    $urlToGo = $urlToGo."&wantsurl=/course/view.php?id=".$moodle_course->id;
                }
                $this->redirect_to($urlToGo);
            }
        }
    }
}

```

Figura 44 - SSO Moodle - trecho de código do método que faz o SSO com o moodle

#### 4.4.3. Gestão de viaturas

Este módulo foi pedido pelo cliente *CFPSA*, que sentiu a necessidade de gerir e controlar de forma simples, as suas viaturas. Nesse sentido, foi criado o módulo Gestão de viaturas. Este

módulo, como foi dito no capítulo Enquadramento, permite criar, editar consultar ou eliminar viaturas, assim como, requisitar e consultar, validar/rejeitar as requisições.

O módulo conta com 4 momentos principais:

- Momento 1 – Requisição da viatura, onde o utilizador, colaboradores do CFPSA, efetuam uma requisição de viatura, indicando algumas informações como por exemplo o destino e as datas;
- Momento 2 - Validação por um responsável, onde este, pode validar ou rejeitar a requisição. Em caso da rejeição é enviado um email ao requisitante com a indicação de que foi rejeitada e o motivo da rejeição. Por outro lado, caso seja validada é enviado um email ao armazém a informar da nova requisição e outro email para o requisitante com a indicação de que foi validade e que aguarda matrícula;
- Momento 3 – Atribuição da matrícula, onde um dos utilizadores que possui o grupo “Armazém” irá associar uma matrícula à requisição e é enviado um email ao requisitante a informar que a requisição teve sucesso e poderá levantar a viatura nas datas requeridas;
- Momento 4 – O último momento, onde o requisitante, depois de utilizar a viatura deverá entregá-la e adicionar algumas informações como por exemplo os quilómetros finais da viatura.

O módulo desenvolvido consistiu em dois *submódulos*, um designado de “Utilização de viaturas” e outro de “Gerir viaturas” constituído por 3 áreas. O *submódulo* “Utilização de viaturas” (Figura 45) é onde decorre o Momento 1 da requisição. Este *submódulo* foi desenvolvido com base no plugin *FullCalendar*, onde, em conjunto com o Novo ORM, permite consultar e efetuar as requisições de viaturas.

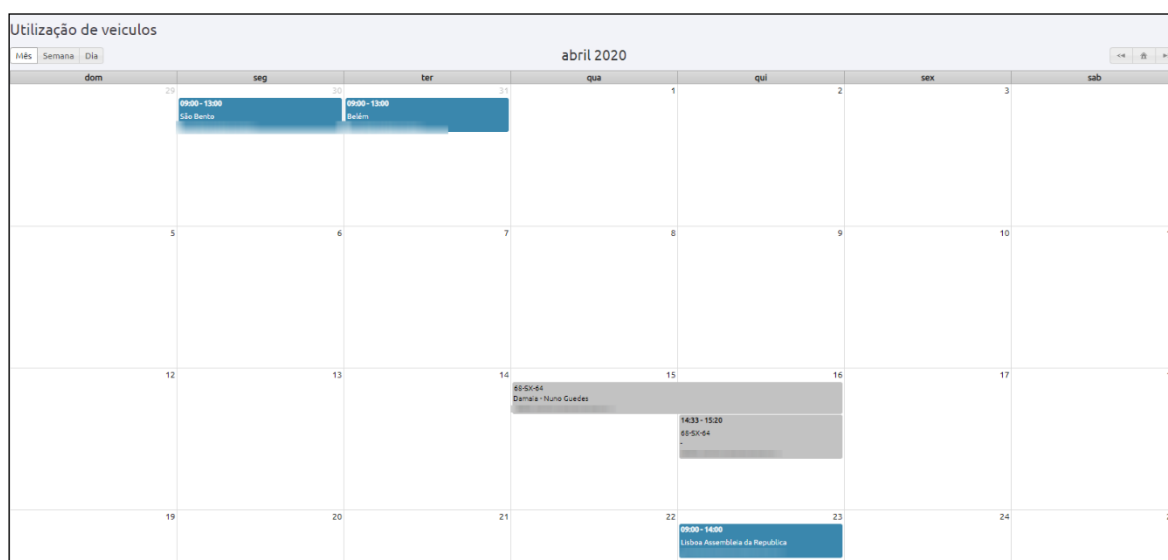


Figura 45 – Submódulo “Utilização de viaturas”

No *submódulo* “Gerir viaturas”, encontra-se uma primeira área constituída pelo título e a opção de configuração deste módulo. Ao clicar nesta opção o utilizador tem acesso a um conjunto de configurações como, “Responsável pela validação”, “Responsável pela atribuição

da matrícula” entre outros, como ilustrado na Figura 46. Este ecrã de configuração foi desenvolvido com base em *React.js* e o *Novo ORM*.

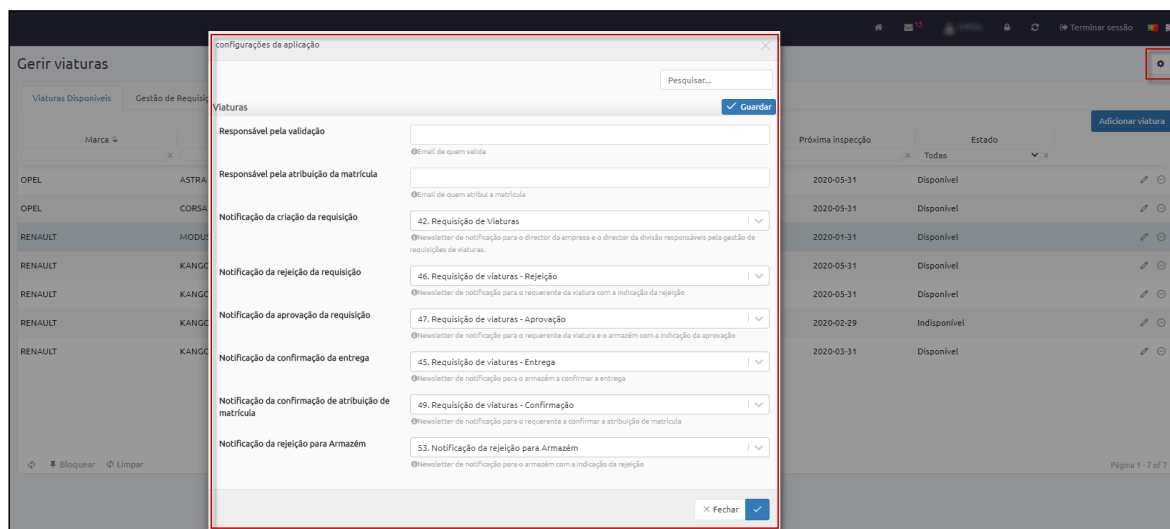


Figura 46 - Configuração de módulo “Gestão de viaturas”

A segunda, a área de viaturas disponíveis, onde estão listadas todas as viaturas inseridas (ver Figura 47). Ainda nesta área o utilizador poderá adicionar uma viatura, assim como eliminar ou alterar viaturas. O *front-end* desta área foi desenvolvido utilizando HTML, CSS e o *plugin jqGrid* desenvolvido pela *humansoft* com base no componente *jqGrid* do *jQuery*, que é no fundo um *plugin* que permite listar dados em tabelas, e permite fácil pesquisa por campos e exportação de dados dependendo das configurações desenvolvidas para criar a tabela. O *back-end* foi desenvolvido com base no Novo ORM, utilizando o “*CustomGrid*” que é um modelo que devolve todos os registos para a tabela conforme as configurações definidas, assim como as ações por exemplo editar, e eliminar, para cada registo da tabela.

Marca	Modelo	Matrícula	Km efetuados	Data de aquisição	Próxima Inspeção	Estado
OPEL	ASTRA	68-5X-64	21197 km	2017-05-05	2020-05-31	Disponível
OPEL	CORSA	68-5X-63	81077 km	2017-05-05	2020-05-31	Disponível
RENAULT	MODUS	95-41-ZL	263541 km	2005-01-25	2020-01-31	Disponível
RENAULT	KANGOO	80-80-ZZ	349218 km	2005-05-24	2020-05-31	Disponível
RENAULT	KANGOO	97-90-ZZ	349432 km	2005-05-25	2020-05-31	Disponível
RENAULT	KANGOO	23-26-ZH	0 km	2005-02-07	2020-02-29	Indisponível
RENAULT	KANGOO	66-65-RG	64022 km	2001-03-08	2020-03-31	Disponível

Figura 47 - Área de viaturas disponíveis

Por último encontra-se a área de “Gestão de Requisições de Viaturas” ilustrada na Figura 48, onde se encontram listadas todas as requisições efetuadas, desenvolvidas igualmente com o recurso ao Novo ORM, HTML, CSS e o *plugin jqGrid*. Esta área está disponível apenas aos utilizadores com certas permissões, como utilizadores que pertencem por exemplo ao grupo de “Armazém”, “Administradores” ou “Diretores”.

#	Viaturas	Km efetuados	Destino	Nome	Data da Requisição	Data de Início	Data de Fim	Estado
386	95-41-ZL	15km	Av. da República e Conde Valbom		2020-07-01 11:54:14	2020-07-07 09:00:00	2020-07-07 13:00:00	Reserva Fechada
384	95-41-ZL	23km	J.F. Arroios		2020-06-30 14:50:38	2020-07-01 10:30:00	2020-07-01 12:30:00	Reserva Fechada
383	68-5X-64	24km	Rua do Cruzeiro - Ajuda		2020-06-30 12:39:01	2020-06-30 13:00:00	2020-06-30 14:00:00	Reserva Fechada
382	68-5X-63	538km	Águeda		2020-06-29 15:38:54	2020-07-06 17:30:00	2020-07-08 09:00:00	Reserva Fechada
381	97-90-ZZ	32km	Farmácia Tojal		2020-06-29 10:03:07	2020-06-29 10:00:00	2020-06-29 12:00:00	Reserva Fechada
380	97-90-ZZ	189km	Niontjo e Castanheira do Ribatejo		2020-06-25 14:49:12	2020-06-29 17:30:00	2020-06-30 13:30:00	Reserva Fechada
379	95-41-ZL	22km	J.F. Arroios		2020-06-25 12:18:54	2020-06-28 14:30:00	2020-06-28 16:30:00	Reserva Fechada
378	95-41-ZL	22km	Charneca do Lumiar, Lisboa		2020-06-23 14:24:01	2020-06-29 09:30:00	2020-06-29 12:30:00	Reserva Fechada
377	97-90-ZZ	13km			2020-06-23 08:53:50	2020-06-23 10:30:00	2020-06-23 12:00:00	Reserva Fechada
376	N/A	0km	IEFP - Xabregas		2020-06-23 07:50:32	2020-06-23 10:00:00	2020-06-23 14:00:00	Aguarda Validação
375	97-90-ZZ	133km	Almada e Amora		2020-06-22 17:56:45	2020-06-23 17:45:00	2020-06-24 12:30:00	Reserva Fechada
374	95-41-ZL	22km	Travessa do Desterro, Lixa		2020-06-22 14:43:29	2020-06-23 14:00:00	2020-06-23 16:00:00	Reserva Fechada

Figura 48 - Áreas de Gestão de Requisições de viaturas

#### 4.4.4. Autenticação.GOV

Este módulo foi desenvolvido ao pedido do cliente *AMA* (Agência para a Modernização Administrativa) e permite aos utilizadores do *humanportal* efetuarem login através do seu cartão de cidadão. Para o conseguir, foi necessário fornecer à *AMA*, informações sobre a configuração de um fornecedor de serviços no *Autenticação.GOV*. Neste sentido, foram fornecidos:

- *CSR (Certificate Signing Request)* para gerar um certificado de utilização exclusiva nos pedidos SAML enviados ao *Autenticação.GOV*;
- *Issuer / Identificador do humanportal* – para utilizar como identificador do *humanportal*;
- *ProviderName / Designação do Organismo* - para ser exibido ao utilizador durante o processo de autenticação;
- Método para *logout* estático – utilizado para realizar o *logout* no *humanportal* – fornecido um *url*;
- Método para *logout dinâmico* - utilizado para realizar o *logout* no *humanportal* – fornecido no pedido SAML;
- Email técnico - para notificações técnicas.

Após a criação dos serviços no *Autenticação.GOV*, foram necessários vários desenvolvimentos do lado do portal. Nesse sentido, foi criada uma opção (Figura 49) no ecrã do login do *humanportal*, que apenas está visível para clientes que tenham permissões de *Autenticação.GOV*.

Introduza os seus dados de login para entrar.

Nome de utilizador \*

Palavra-passe \*

 autentique-se aqui  
AUTENTICAÇÃO.GOV.PT

Não tem acesso? Registe-se aqui.  
Recuperar palavra-passe.

Entrar

Figura 49 - Opção para autenticação através do cartão cidadão

Associada a esta opção, foram criados vários métodos no controlador “*sessionsController*”, dos quais se destaca o “*getAuthGov*”, “*ssoResponse*”, “*getAuthGovLogout*”. Ao selecionar a opção para autenticação através da *Autenticação.GOV*, é efetuado um pedido *ajax* ao método “*getAuthGov*” (Figura 50), onde é construído e assinado o pedido SAML a ser enviado para realizar a autenticação do lado da *Autenticação.GOV*. Em resposta ao pedido efetuado a *Autenticação.GOV* é chamado o método “*ssoResponse*” (Figura 51), onde é processado a resposta também em SAML, efetuada a validação e autenticação do utilizador no *humanportal*.

```

47 function getAuthGov() {
48     $this->responds('json');
49     if (!defined('AUTH_GOV')) { !file_exists($file = ROOT . 'private/auth.gov/settings-' . AUTH_GOV . '.php') } {
50         sys\kern\Messages::warning(Lang::get('sessions_authgov_denied'));
51         return [];
52     }
53     $settings = new sys\lib\saml\Settings(include_once($file));
54     $spdata = (object)$settings->getSPData();
55     $idpdata = (object)$settings->getIdPData();
56     $xml_request_template =
57     <?xml version="1.0" encoding="UTF-8" ?>
58     <samlp:AuthnRequest
59         xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
60         Version="2.0"
61         IssueInstant="{timestamp}"
62         Destination="{target}"
63         ProtocolBinding="{binding}"
64         AssertionConsumerServiceURL="{respondsTo}"
65         ProviderName="{entityId}"
66     >
67     <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://www.humansoft.pt</saml:Issuer>
68     <samlp:Extensions
69         <fa:RequestedAttributes xmlns:fa="http://autenticacao.cartaodecidadao.pt/atributos">
70         {attributes}
71         </fa:RequestedAttributes>
72         <fa:FAAALevel xmlns:fa="http://autenticacao.cartaodecidadao.pt/atributos">3</fa:FAAALevel>
73     </samlp:Extensions>
74     </samlp:AuthnRequest>;
75     $attributes = array_map(function($attr) {
76         return sys\lib\saml\Utils::parse('<fa:RequestedAttribute Name="{name}" NameFormat="{nameFormat}" isRequired="{isRequired}" />', [
77             'name' => $attr['name'],
78             'nameFormat' => $attr['nameFormat'],
79             'isRequired' => $attr['isRequired'] && $attr['isRequired'] ? 'True' : 'False'
80         ]);
81     },
82     $spdata->attributeConsumingService['requestedAttributes'];
83     );
84     $xml_request = sys\lib\saml\Utils::addSign($xml_request, $settings->getSPKey(), $settings->getSPcert());
85     $xml_request = base64_encode($xml_request);
86     Log::file('autenticacao.gov', 'sessionscontroller', 'getAuthGov', $xml_request);
87     return [
88         'saml_request' => $xml_request,

```

Figura 50 - *Autenticação.GOV* - trecho de código do método “*getAuthGov*”

```

function ssoResponse() {
    $this->responds('json');
    if (!defined('AUTH_GOV') || !file_exists($file = ROOT . 'private/auth.gov/settings-' . AUTH_GOV . '.php')) {
        sys\kern\Messages::warning(Lang::get('sessions_authgov_notauth'));
        $this->redirect_to('/sessions/login.personal');
    }
    $settings = include_once($file);
    $auth = new sys\lib\saml\Auth($settings);
    try {
        $auth->processResponse($this->params->request_id ?: null);
    }
    catch(Exception $ex) {
        sys\kern\Messages::warning(Lang::get('sessions_authgov_error'));
        $this->redirect_to('/sessions/login.personal');
    }
    if (!$auth->isAuthenticated()) {
        sys\kern\Messages::warning(Lang::get('sessions_authgov_notauth'));
        $this->redirect_to('/sessions/login.personal');
    }
    // verificar se a sessão é a mesma
    $relay = sys\lib\Ssl::decrypt($this->params->RelayState, '_humansoft#', true);
    list($nif, $session_id) = explode('.', $relay, 2);
    if ($nif != NIF || $session_id <> session_id()) {
        sys\kern\Messages::warning(Lang::get('sessions_authgov_expired'));
        $this->redirect_to('/sessions/login.personal');
    }
    // validar o utilizador
    $attributes = [];
    foreach($auth->getAttributes() as $name => $value) {
        preg_match('/\w+/', $name, $matches);
        $attributes[ $matches[0] ] = $value[0];
    }
    Log::file('autenticacao.gov.login', 'sessionscontroller', 'ssoResponse', var_export($attributes, true));

    $user = app\models\User::where('nif', $attributes['NIF'])->select('username')->orderBy('enabled desc')->one();
    if (!$user->username) {
        $user = app\models\User::where("id_doc_type = 'CC' and id_doc like concat('CC', '%')", ['cc' => $attributes['NIC']])
            ->select('username')
            ->orderBy('enabled desc')
            ->one();
    }
    if (!$user->username) {
        sys\kern\Messages::warning(Lang::get('sessions_authgov_notfound'));
        $this->redirect_to('/sessions/login.personal');
    }
    //sys\kern\Messages::success(Lang::get('sessions_authgov_success'));
    $_POST['trusted_login'] = true;
    $_SERVER['LOGON_USER'] = $user->username;
    Cookie::set('sso_gov', 1);
    $this->input->responds_to = 'personal';
    $this->_startNormal();
}

```

Figura 51 - Autenticação.GOV - trecho de código do método “ssoResponse”

Todos os utilizadores que se autenticam através da aplicação *Autenticação.GOV*, ao realizar o *logout* no *humanportal* é efetuado um pedido a *Autenticação.GOV* através do método “*getAuthGovLogout*”, onde é construído e assinado o pedido SAML a ser enviado para realizar o *logout* do lado da *Autenticação.GOV*. Em resposta ao pedido efetuado é chamado o método “*end*” que realiza o *logout* do lado do *humanportal*.



## 5. Conclusão

Os objetivos desse estágio, incidiram essencialmente na reformulação e desenvolvimento de alguns módulos do projeto *humanportal*, com o propósito de abranger as várias necessidades do mercado na componente dos recursos humanos, assim como crescer e inovar com produtos e serviços de qualidade.

Depois do estágio realizado, em grande parte descrito neste relatório, considero que os objetivos propostos foram cumpridos. Durante os 10 meses do estágio, as experiências e competências adquiridas foram inúmeras, sendo igualmente de realçar o desenvolvimento de *soft skills*, principalmente ao nível profissional.

O desenvolvimento do módulo da integração com o moodle, da *Autenticação.GOV* e ações de coordenadores, foram projetos bastante desafiantes que permitiram adquirir conhecimentos em várias tecnologias, e foram igualmente os projetos em que foram sentidas mais dificuldades, pois foi necessária muita pesquisa, e a documentação disponível, por vezes é escassa, como é o caso da *Autenticação.GOV*. Contudo, o estudo do *React.js*, para o desenvolvimento do módulo de ações de coordenadores, por exemplo, trouxe competências que satisfazem as necessidades da empresa, pois esta tecnologia foi selecionada para futuros desenvolvimentos/reformulações de módulos de *humanportal*. Merece igualmente relevo o enriquecimento da experiência dada a aprendizagem da tecnologia SAML 2.0 no módulo da *Autenticação.GOV*.

Por fim, pode-se confirmar que o estágio na *humansoft* foi, uma experiência muito enriquecedora, tanto ao nível da obtenção de competências ao nível cultural e experiência profissional, o que permitiu ultrapassar com sucesso, todas as dificuldades e obstáculos que surgiram durante o percurso.

Parte do trabalho efetuado no âmbito deste estágio, resultou na escrita de um artigo científico que foi aceite e apresentado como *Full Paper* na *CISTI'2020 - 15<sup>th</sup> Iberian Conference on Information Systems and Technologies* [36], estando no momento a ser redigido um segundo artigo para posterior submissão.

### 5.1. Trabalho futuro

Por forma a melhorar as experiências dos seus utilizadores e aumentar o reconhecimento da sua importância do *humanportal*, no mercado da componente dos recursos humanos, existem diversos desenvolvimentos e reformulações que poderão ser realizadas:

- **Análise e implementação de outro explorador de ficheiro**

O atual explorador de ficheiro tem muitos problemas, um estudo de um melhor *plugin* para este efeito seria, algo muito satisfatório, uma vez que é um dos módulos que muitos clientes que possuem a componente website, utilizam, e que mais *feedback* negativo é transmitido.

- **Desenvolvimento de módulos recorrendo ao *React.js*,**  
A semelhança de alguns módulos que já foram desenvolvidos ou reformulados com base nessa tecnologia, os resultados são de excelência, com um grau de satisfação dos clientes notório, tendo os mesmos destacado o rápido acesso. Além disso torna-se muito mais simples a manutenção, uma vez que o *React.js* é baseado em componentes.
- **Reformulação do módulo das *keys* e das permissões.**  
Estes dois módulos, são dos mais importantes no *humanportal*, no entanto, os clientes não tiram partido desses módulos devido uma utilização muito desafiante e complicada de se entender.
- **Disponibilização de pesquisa “on the fly” nos menus.**  
Uma vez que o *humanportal*, conta com muitos módulos, o que faz com que tenha uma vasta quantidade de menus, um dos trabalhos futuros, seria adicionar uma pesquisa que filtrasse os menus, desta forma facilitava muito a utilização dos menus, como também, ganhava-se mais tempo na sua utilização.
- **Desenvolvimento de configurações em cada ecrã principal de cada módulo.**

Uma das dificuldades que são notórias, também ao nível das configurações, pois o *humanportal* possui várias configurações de módulos, sendo que não estão organizadas da melhor forma, o que torna difícil, chegar a uma determinada configuração. Esta disponibilização de configuração em cada ecrã principal do módulo irá trazer mais valias para a *humansoft* que ganhará muito mais tempo, uma vez que irá diminuir o número de assistências para definição de uma determinada configuração.

---

## Bibliografia

- [1] L. Gonçalves, “O que é Scrum, tudo o que precisa de saber sobre esta metodologia,” [Online]. Available: <https://luis-goncalves.com/pt-pt/o-que-e-scrum/>. [Accessed 19 08 2020].
- [2] “Getting Started,” [Online]. Available: <https://code.visualstudio.com/docs>. [Accessed 20 09 2020].
- [3] S. Woollahra, “Documentation - Sublime Text,” [Online]. Available: <https://www.sublimetext.com/docs/3/>. [Accessed 10 09 2020].
- [4] “HeidiSQL,” [Online]. Available: <https://www.heidisql.com/>. [Accessed 20 09 2020].
- [5] “Internet Information Services,” [Online]. Available: [https://en.wikipedia.org/wiki/Internet\\_Information\\_Services](https://en.wikipedia.org/wiki/Internet_Information_Services). [Accessed 10 09 2020].
- [6] “cPanel & WHM Features,” [Online]. Available: <https://cpanel.net/products/cpanel-whm-features/>. [Accessed 10 09 2020].
- [7] “HTML: Linguagem de Marcação de Hipertexto,” [Online]. Available: <https://developer.mozilla.org/pt-PT/docs/Web/HTML>. [Accessed 18 09 2020].
- [8] “CSS,” [Online]. Available: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. [Accessed 18 09 2020].
- [9] “Build fast, responsive sites with Bootstrap,” [Online]. Available: <https://getbootstrap.com/docs/4.5/getting-started/introduction/>. [Accessed 18 09 2020].
- [10] “Introduction - Documentation Twig,” [Online]. Available: <https://twig.symfony.com/doc/2.x/intro.html>. [Accessed 20 08 2020].
- [11] “Twig,” [Online]. Available: <https://twig.symfony.com/doc/2.x/index.html>. [Accessed 18 09 2020].
- [12] “JavaScript,” [Online]. Available: <https://pt.wikipedia.org/wiki/JavaScript>. [Accessed 20 08 2020].
- [13] “JavaScript,” [Online]. Available: <https://developer.mozilla.org/pt-PT/docs/Web/JavaScript>. [Accessed 18 09 2020].

- 
- [14] “jQuery,” [Online]. Available: <https://pt.wikipedia.org/wiki/JQuery>. [Accessed 20 08 2020].
- [15] “jQuery API,” [Online]. Available: <https://api.jquery.com/>. [Accessed 18 09 2020].
- [16] “Introduction - Vue.js,” [Online]. Available: <https://vuejs.org/v2/guide/>. [Accessed 10 10 2020].
- [17] “Introduction to the Angular Docs,” [Online]. Available: <https://angular.io/docs>. [Accessed 10 10 2020].
- [18] “React - Getting Started,” [Online]. Available: <https://reactjs.org/docs/getting-started.html>. [Accessed 18 09 2020].
- [19] “@angular/core vs angular vs react vs vue,” [Online]. Available: <https://www.npmtrends.com/@angular/core-vs-angular-vs-react-vs-vue>. [Accessed 20 08 2020].
- [20] “Google Trends,” [Online]. Available: <https://trends.google.com/trends/explore?cat=31&q=Vue.js,React,Angular>. [Accessed 20 08 2020].
- [21] “The Best JS Frameworks for Front End,” [Online]. Available: <https://rubygarage.org/blog/best-javascript-frameworks-for-front-end>. [Accessed 20 08 2020].
- [22] “React.js and Front-End Development,” [Online]. Available: <https://dzone.com/articles/why-choose-react-for-front-end-development>. [Accessed 20 08 2020].
- [23] “PHP,” [Online]. Available: <https://pt.wikipedia.org/wiki/PHP>. [Accessed 20 08 2020].
- [24] “PHP Manual,” [Online]. Available: <https://www.php.net/manual/en/index.php>. [Accessed 18 09 2020].
- [25] “MySQL,” [Online]. Available: <https://pt.wikipedia.org/wiki/MySQL>. [Accessed 26 08 2020].
- [26] “MySQL Documentation,” [Online]. Available: <https://dev.mysql.com/doc/>. [Accessed 18 09 2020].
- [27] “Git,” [Online]. Available: <https://pt.wikipedia.org/wiki/Git>. [Accessed 26 08 2020].

- 
- [28] “Book,” [Online]. Available: <https://git-scm.com/book/en/v2>. [Accessed 18 09 2020].
- [29] “Demo for JQuery Smooth Div Scroll,” 30 11 2019. [Online]. Available: <http://www.htmldrive.net/items/demo/554/JQuery-Smooth-Div-Scroll>.
- [30] “Slick,” 18 12 2019. [Online]. Available: <https://kenwheeler.github.io/slick/>.
- [31] “Instagram,” 25 11 2019. [Online]. Available: <https://www.instagram.com/developer/>.
- [32] “O que é o Efeito de Rolagem Parallax? Definição e Exemplos,” 16 06 2020. [Online]. Available: <https://pt.wix.com/blog/2019/09/o-que-e-efeito-rolagem-parallax-exemplos/>.
- [33] FullCalendar, “JavaScript Calendar,” [Online]. Available: <https://fullcalendar.io/>. [Accessed 07 06 2020].
- [34] Google, “Google Calendar API,” 03 05 2019. [Online]. Available: <https://developers.google.com/calendar/concepts/events-calendars>. [Accessed 22 08 2020].
- [35] “Authentication: User key authentication,” [Online]. Available: <https://moodle.org/plugins/pluginversion.php?id=16532>. [Accessed 15 01 2020].
- [36] Sousa, M. and Gonçalves, A. "humanportal – A React.js case study", in proc. of CISTI'2020 - 15th Iberian Conference on Information Systems and Technologies, Sevilha, 2020. doi: 10.23919/CISTI49556.2020.91