

# Mini Projeto 1

Tecnologias Dinâmicas para a Internet

Ana Filipa Ferreira | 89442

Repositório: <https://github.com/mcmm-mi/tdi-mp1-ana-ferreira>

## Índice

Introdução.....	3
1. Design e estrutura da aplicação.....	3
2. Implementação técnica .....	5
2.1. Tecnologias utilizadas.....	5
2.2. Arquitetura de pastas .....	6
3. Decisões tomadas.....	6
4. Principais desafios .....	7
5. Obstáculos não ultrapassados.....	7
Conclusão .....	8
Webgrafia.....	8

## Introdução

O projeto *Potterpedia*, tal como o nome indica, debateu-se sobre a temática da saga da escritora J. K. Rowling. Este nome surgiu após pensar no *Website* como uma enciclopédia do Harry Potter, por isso fez todo o sentido fazer a junção das duas palavras. Além disso, a razão de escolha desta temática prendeu-se sobretudo com o facto de ser algo bastante interessante e que, ao mesmo tempo, acabava por satisfazer os requisitos propostos pelos docentes.

Deste modo, foi criado um *website* onde o utilizador consegue emergir no mundo da magia através da visualização dos feitiços lecionados em Hogwarts e até mesmo ser selecionado para uma das casas da escola de magia. Adicionalmente, é possível visualizar as personagens da saga, bem como obter mais detalhes sobre estas, pode-se ainda ver as características de cada casa e gerar aleatoriamente algumas das frases mais icónicas da saga.

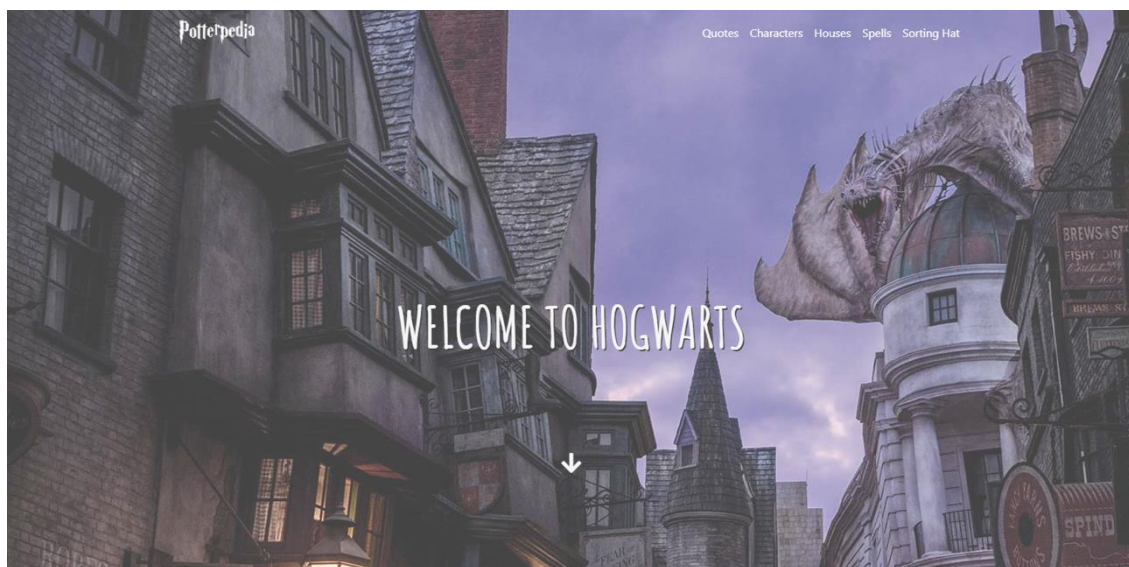


Figura 1 Página Inicial do Website

### 1. Design e estrutura da aplicação

Primeiramente, é preciso realçar o facto que o *design* foi surgindo conforme a construção de cada página, porém acabou por pensado de forma a que existisse harmonia entre as páginas. Isto deve-se essencialmente à questão de não ter existido muito tempo para refletir ou até investir nesta questão, tal como aconteceu em outros projetos. Para existir harmonia foi essencial a criação de uma paleta de cores, sendo que esta foi baseada numa das casas de Hogwarts.

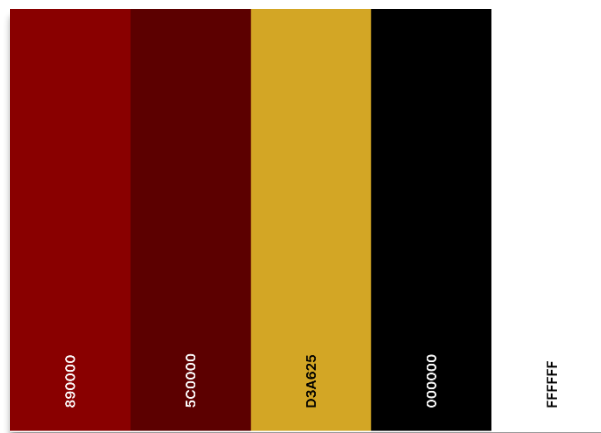


Figura 2 Paleta de cores do Website

Deste modo, o *design* pensado prendeu-se com algo mais *clean* mas que, ao mesmo tempo, dá a ideia de se tratar de um *Website* alusivo à saga Harry Potter, tanto devido ao uso de imagens, como ao tema mais escuro com cores características desta. Por conseguinte, em todas as páginas, excetuando na *Quotes* e no *Sorting Hat*, existe um *banner* que identifica a página em questão e só depois deste é que aparece o conteúdo informativo. No caso da página das frases é apenas utilizada uma imagem de fundo e, por outro lado, na do *Sorting Hat* é utilizada uma cor sólida que vai alterando consoante a casa sorteada.

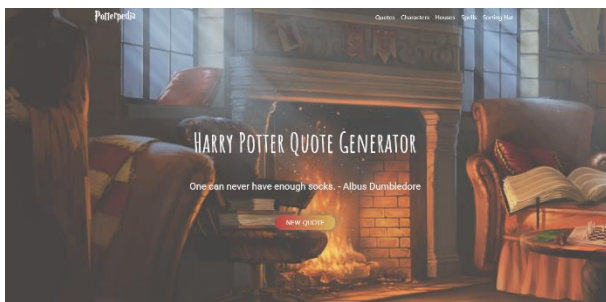


Figura 5 Página Quotes

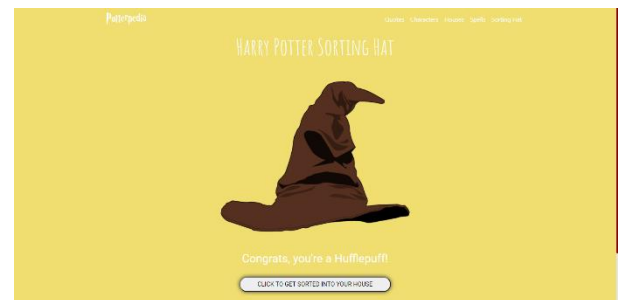


Figura 4 Página Sorting Hat



Figura 3 Página Characters

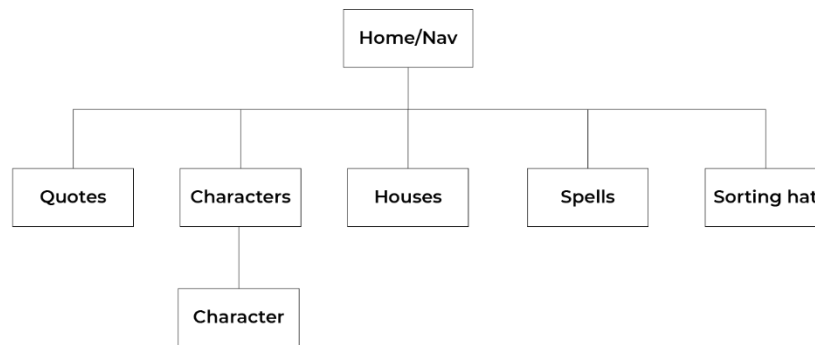


Figura 6 Mapa do Website

No que diz respeito à estrutura da aplicação, foi escolhida uma navegação simples que fosse possível de ser entendida por qualquer utilizador, desta forma, tanto a partir da página principal, como da *navbar* é possível aceder às principais páginas do *Website*, tal como se pode observar pela figura 6. Explicando com mais detalhe as páginas, nas *Quotes* é possível gerar aleatoriamente uma frase icónica da saga através de um botão. De seguida, na página *Characters* pode-se visualizar todas as personagens, bem como ordená-las pelo nome, de forma ascendente ou descendente, e fazer uma pesquisa pelo nome de cada um. Esta página dá ainda acesso a uma página *master detail* de qualquer personagem que for seleccionada, onde é possível obter informações mais detalhadas sobre a mesma. Na página *Houses* é possível observar quatro *cards* com um pequeno texto sobre cada uma das casas existentes em Hogwarts. Já na página *Spells* pode-se observar todos os feitiços que existem, bem como ser informado sobre o que cada um faz, além disso, é também possível ordenar de forma ascendente e descendente cada um e também fazer uma pesquisa pelo nome. Por último, existe a página *Sorting Hat* onde o utilizador pode ser sorteado aleatoriamente para uma casa, de forma a ter uma experiência mais completa.

## 2. Implementação técnica

### 2.1. Tecnologias utilizadas

Conforme os objetivos definidos para a unidade curricular de Tecnologias Dinâmicas para a internet, com base em *React*, foi desenvolvido um *Website* com recurso a duas APIs diferentes, nomeadamente a *Potter API* e a *HP-API*. Adicionalmente, foi utilizado o *Redux* para retornar os dados das duas APIs e, em termos de design, foi utilizado o *Reactstrap* em conjunto com o *Bootstrap*, recorrendo também a *Styled Components*.

## 2.2. Arquitetura de pastas

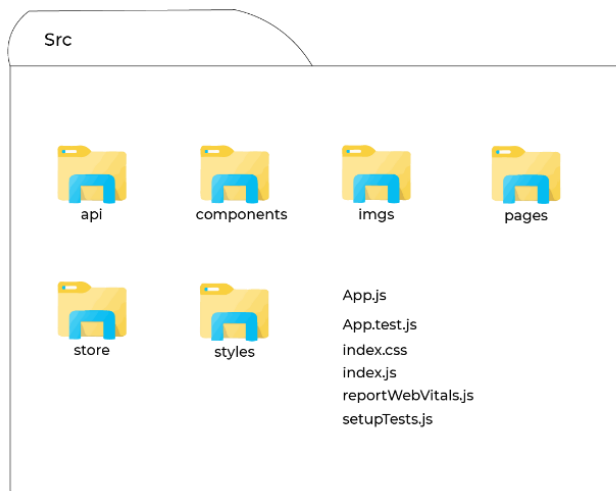


Figura 7 Esquema da pasta src

Apesar do projeto ser composto por outras pastas, igualmente importantes para o seu funcionamento, aquilo que foi feito durante todo o projeto concentrou-se maioritariamente na pasta *src*.

Por conseguinte, dentro desta pasta, tal como se consegue observar na figura 7, encontram-se todos os componentes do projeto. Esta encontra-se organizada em seis pastas diferentes, juntamente com outros

componentes do tipo *JavaScript* e *CSS*, dos quais se destacam o *index.js*, que renderiza o *JSX* no ficheiro de *HTML*, e o *app.js* que tem duas funcionalidades, servir de *Provider* para a *store* e atribuir links às diversas páginas do *Website* através do *Router*.

Seguidamente, é crucial abordar também algumas das pastas dentro da pasta *src*, de forma a esclarecer o seu conteúdo. Deste modo, na pasta *api* é onde começa o *fetch* dos dados, no sentido de ser aí que são chamados os links das respetivas APIs. A pasta *components* foi utilizada para conter todas os componentes necessários do *Website*, isto é, foi nesta pasta que foram construídos os componentes, recorrendo a *JSX*. Por outro lado, a pasta *pages* é aquela onde são ligados os componentes mencionados anteriormente para construir a página que aparece para o utilizador. Por fim, a última pasta considerada importante é a *store*, uma vez que é esta que contém a lógica do *Redux*. A pasta está dividida em três subpastas (*characters*, *houses* e *spells*), sendo que cada uma destas contém as pastas *actions* e *reducers*. As informações destas três pastas principais acabam por se ligar dentro do *index.js* que se encontra dentro da pasta *store*. Esta é a mais relevante pois é a que permite que os dados das APIs sejam retornados devidamente.

## 3. Decisões tomadas

Devido ao curto espaço de tempo que existia para a realização do projeto, tiveram de existir diversas decisões ao nível do que deveria ou não ser implementado e como o fazer.

Por conseguinte, uma das primeiras coisas que ficou decidida foi o uso do *Redux* para recolha dos dados vindos das APIs selecionadas, apesar de no momento não compreender bem o seu funcionamento percebi que iria facilitar o trabalho futuramente.



Para além disso, e após analisar o que as APIs retornavam, foi feita uma seleção dos dados considerados interessantes a serem exibidos no *Website*, uma vez que nem todos faziam sentido aparecer, tendo em conta o objetivo que tinha em mente.

Por último, optei por apresentar os conteúdos das personagens e dos feitiços com *infinite scrolling*, para dar ao utilizador uma experiência diferente, deste modo o utilizador não precisa de andar a carregar em botões para ter acesso a toda a informação. Para compensar isso, foi criado um botão que aparece no canto inferior direito que dá acesso direto ao topo caso o utilizador necessite de usar algum dos filtros.

## 4. Principais desafios

Tal como já aconteceu no desenvolvimento de projetos anteriores, durante a realização deste existiram, também, alguns desafios, sobretudo devido ao uso de uma *Framework* (ReactJS) com a qual ainda se teve pouco contacto, o que leva a maiores dificuldades. Deste modo, teve de existir uma maior autonomia na procura de soluções e documentações que pudessem ajudar no decorrer do projeto.

Assim, um dos maiores desafios foi, sem dúvida, a escolha de uma API que tanto pudesse satisfazer os objetivos propostos, como fosse de um tema interessante. Ora, existiam muitas que tinham um número limitado de pedidos ou onde era necessário pagar para se puderem utilizar, porém, ao aprofundar a pesquisa e com a ajuda do professor foi possível chegar a duas APIs que cabiam perfeitamente nos requisitos tanto pessoais como do projeto.

Adicionalmente, apesar de ter sido abordado em questão de sala de aula, o uso do *redux* foi um desafio que teve de ser ultrapassado. De forma mais detalhada, o principal desafio deste foi perceber o seu funcionamento e onde era o melhor sítio para o colocar a funcionar dentro do projeto, porém, após diversas pesquisas e consultando o que foi feito em aula foi possível chegar a uma solução que funciona perfeitamente.

Por último, tanto a realização da ordenação das personagens e dos feitiços por nome como a construção da *search bar* foram um grande desafio. Isto aconteceu, principalmente, por ter tentado recorrer maioritariamente ao *redux* para a realização destas duas funcionalidades, quando a solução passava por algo mais simples.

## 5. Obstáculos não ultrapassados

Existiram diversos obstáculos que não foram possíveis de ultrapassar, no entanto destaco apenas aquele que considero ser o maior. Assim, este está ligado à parte de *design*, uma vez que pretendia seguir a linha de design de uma das maiores páginas de *fan club* de Harry Potter (*Wizarding World*) e ter um fundo animado que simbolizasse a magia, no entanto devido a questões de tempo apenas o consegui fazer na página que exibe os feitiços.

## Conclusão

Concluindo, este foi um projeto bastante desafiante em diversos aspetos, entre os quais saliento o tempo para o realizar e a utilização de informação externa que não depende de quem realiza o projeto. Porém, e apesar do produto final poder ter sido mais apurado, foi um bom desafio para estimular a autonomia e aplicar os conhecimentos adquiridos em contexto de aula. Além disso, existiu a possibilidade de entrar em contacto com novas tecnologias durante as pesquisas realizadas, o que acabou por ser uma mais valia para o desenvolvimento pessoal.

## Webgrafia

Abramov, D. (2020). Redux. Retrieved from <https://redux.js.org/>

Fraser, B. (n.d.). HP-API. Retrieved November 8, 2020, from <http://hp-api.herokuapp.com/>

Prakash, S. (2017). Navbar changing to add shadow on Scroll. Retrieved November 9, 2020, from <https://stackoverflow.com/questions/40967682/navbar-changing-to-add-shadow-on-scroll>

Reactstrap. (n.d.). Retrieved November 5, 2020, from <https://reactstrap.github.io/>

Spencer, K., & Glavina, A. (2017). Harry Potter API. Retrieved November 8, 2020, from <https://www.potterapi.com/>