

Licenciatura Engenharia Informática e Multimédia

Programação de Conteúdos Multimédia - PCM

Relatório Projeto Final

Docente: Eng. Rui Jesus

Trabalho realizado por:

Fábio Dias, nº 42921

**1. Introdução**

Para este projecto final, foi-nos pedido a realização de um clone do Google Images, com base nos ficheiros disponibilizados pelo docente.

Foi realizada uma versão com a procura de imagens a partir de palavras-chave, assim como, por semelhança de cores, utilizando a cor dominante de cada imagem.

**2. Desenvolvimento**

**2.1. HTML**

As partes fundamentais de este projecto são a barra de pesquisa, onde se colocam as palavra-chave a serem pesquisadas; a barra de cores, onde estão as cores dominantes que o browser mostra; e o *canvas*, onde são apresentadas as imagens, tanto por palavra-chave como por cores.

Utilizando o ficheiro *html* fornecido pelo docente, no *head* importei o Bootstrap 5, tal como se pode observar na Fig. 1. Para o *body*, criei um *header* (Fig. 2), onde inseri o logotipo deste clone do Google Images, uma barra de pesquisa com as cores todas alinhadas e um botão para a pesquisa.

Mais abaixo, mas ainda dentro do *body*, encontra-se o *canvas*, onde as imagens são apresentadas. Por fim, a função *main*, explicada mais à frente, é chamada quando o browser acaba de carregar esta página. Tudo isto pode ser observado na Fig. 3.

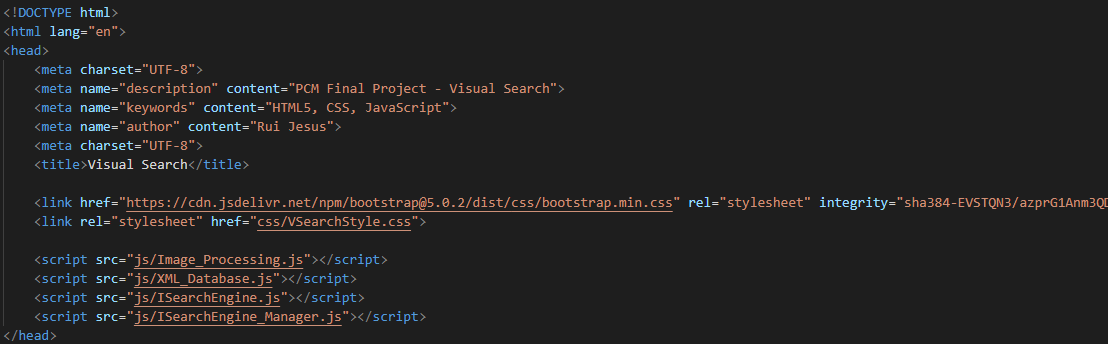


Figura - Head no HTML



Figura - Body, Header no HTML

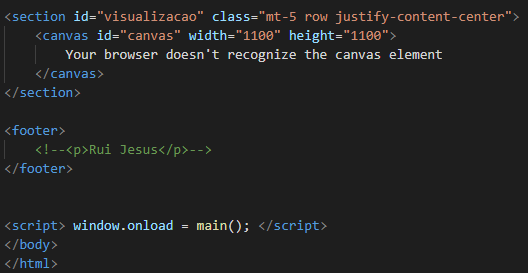


Figura - Canvas no HTML

**2.2.** **Javascript**

O fluxo do programa começa no *script ISearchEngine\_Manager*. Aqui, no seu método *main* (Fig. 4), é instanciada e iniciada a classe *ISearchEngine* que possui a lista de imagens, assim como as categorias, cores e cada valor dos componentes RGB.

O método *init* possui o *databaseProcessing* (Fig. 5), e deve ser chamado apenas uma vez e, de seguida, comentado.

Este método processa cada imagem e chama o método que permite a criação dos ficheiros XML e a colocação do mesmo no *Local Storage*. Dado que só é necessário correr uma vez, deve-se comentar para poupar processamento, e para permitir o funcionamento normal da aplicação.

Ainda no *main* (Fig. 4), são adicionados vários *Event Listener*s, um para o clique no botão de pesquisa por palavra-chave, e um por cada uma das cores exibidas. Estes chamam os métodos *searchKeywords* e *searchColors*, respectivamente.

O método *searchKeywords* (Fig. 6) pesquisa, no XML local, as imagens que pertencem à categoria passada como argumento, o array das imagens amostradas é limpo e são inseridas as novas imagens encontradas, depois é chamado o método *gridView* que apresenta as imagens com os devidos espaçamentos e alinhamentos. *searchColors* (Fig. 7) pesquisa no XML do *Local Storage* as primeiras imagens que tenham a cor dominante semelhante à cor seleccionada, depois, tal como no *searchKeywords*, é limpo o array, são inseridas estas novas imagens e é chamado o *gridView*.



Figura - Método main, no ISearchEngine\_Manager

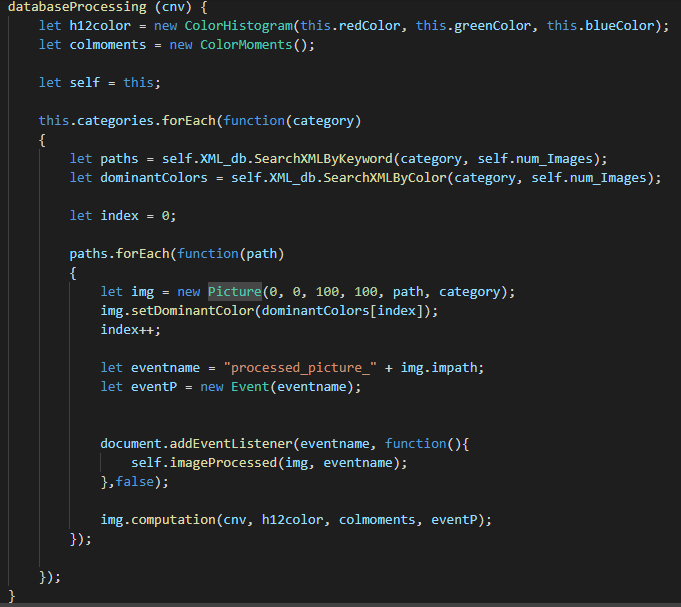


Figura - Método databaseProcessing, no ISearch\_Engine

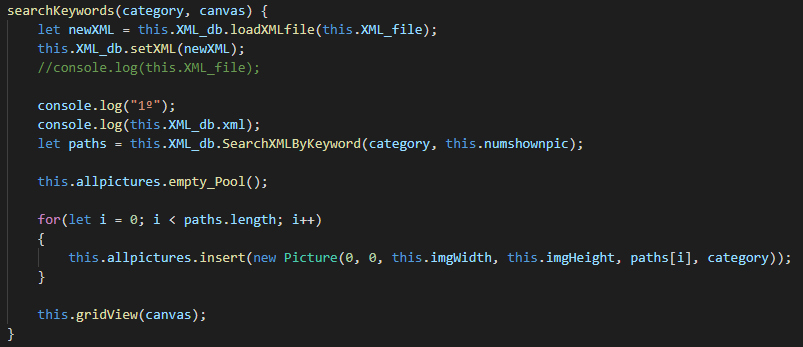


Figura - Método searchKeywords, no ISearch\_Engine

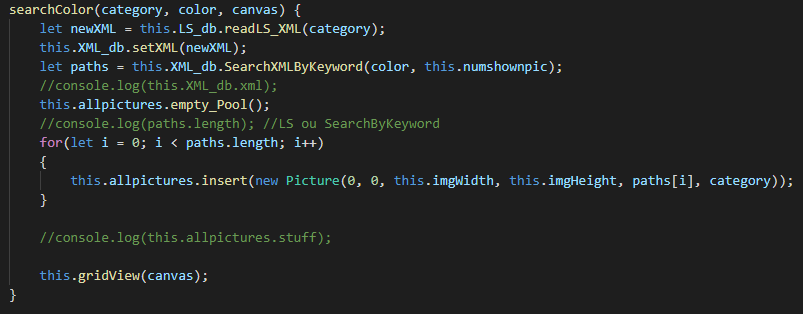


Figura - Método searchColor, no ISearch\_Engine

Com esta aplicação implementada, é possível observar o seguinte resultado.

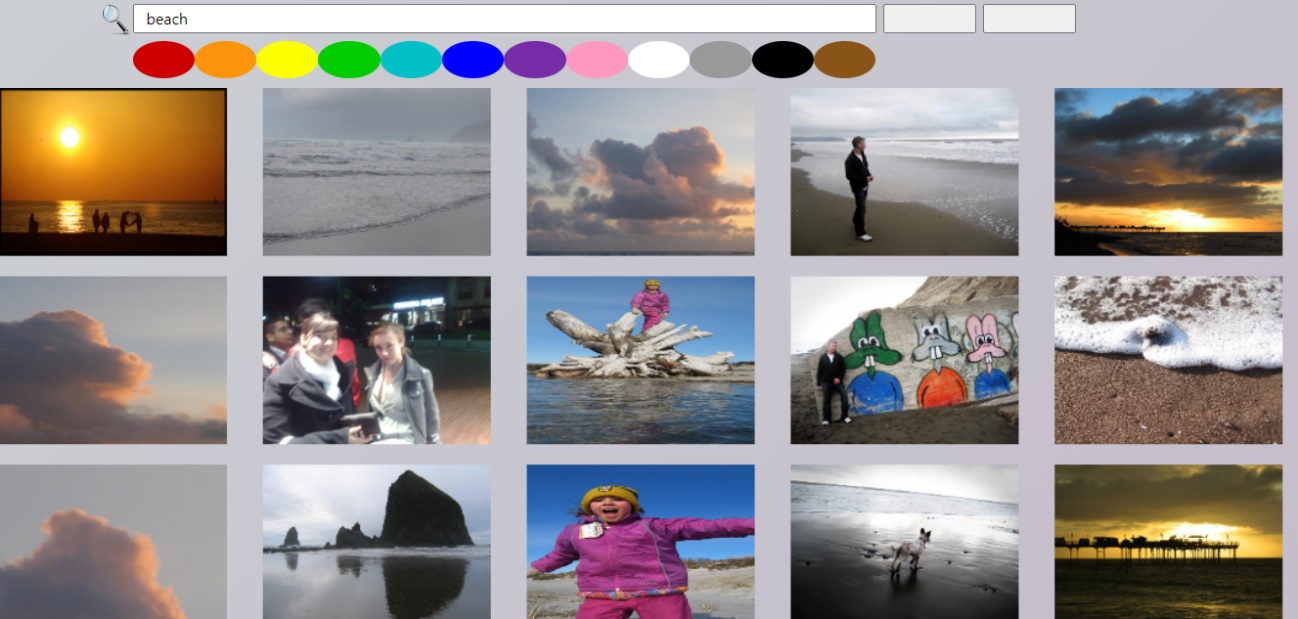


Figura - Interface Gráfica da aplicação

**3. Conclusão**

Embora este clone do Google Images seja um projecto interessante, tive de apontar para a versão mais simples devido à falta de tempo útil para o desenvolvimento deste programa. Tinha algumas ideias de como tornar a sua interface gráfica mais apelativa, como o canvas estar centrado e possuir um botão que variava entre o *light theme* e o *dark theme*, tema claro e escuro, mas acabou por cair no desuso.

Mesmo assim, foi possível observar uma aplicação que é capaz de pesquisar imagens por base em ficheiros XML, tanto locais, como no *Local Storage*. Para além disso, foi possível filtrar, por entre milhares de imagens, por *queries* diferentes, como palavras-chave e cor,