



ISEL / DEETC

Licenciatura em Engenharia Informática e Multimédia

Produção de Conteúdos Multimédia

Projeto de Produção de Conteúdos Multimédia

Pesquisa de Fotos

Rui Jesus

1. Introdução

Este projeto tem como objetivo o desenvolvimento de uma aplicação multimédia em HTML5, incluindo APIs em JavaScript, para pesquisa e visualização de fotos digitais de uma coleção de fotografias. A aplicação deverá incluir vários tipos de informação (conteúdos) multimédia e deverá também permitir a visualização (disposição no ecrã) dos resultados da pesquisa de diversas formas. O desenvolvimento da aplicação deverá ser centrado no utilizador desde o início do projeto até ao protótipo final.

2. Google Images

Os alunos têm de desenvolver uma versão simplificada do Google Images composta por uma interface utilizador em HTML5 para permitir que o utilizador faça as pesquisas e visualize os resultados das pesquisas. A aplicação deverá ter os requisitos enumerados nos pontos seguintes.

2.1 Requisitos Mínimos

A aplicação a desenvolver tem de incluir os seguintes **requisitos mínimos** para aprovação na componente prática (nota prática máxima de 12 valores¹):

- pesquisa por **palavras-chave** e por **cor** (utilizando a cor dominante fornecida no ficheiro XML).
- na pesquisa por cor, esta deverá ser selecionada de uma lista de cores (ver Google Images);
- visualização das imagens (resultados das pesquisas) no **canvas**, organizadas em modo *grid* (grelha de imagens) de acordo com a sua relevância para a *query*.
- deverá ser avaliada com teste de usabilidade com pelo menos 8 utilizadores.

2.2 Requisitos Adicionais

A aplicação deverá incluir, para além dos **requisitos mínimos**, os seguintes **requisitos adicionais** (nota prática máxima de 17 valores²):

¹ Inclui a nota obtida nas aulas de laboratório, a avaliação do código produzido nas aulas e entregue dentro do prazo (máximo 9 valores), a aplicação a funcionar com os requisitos mínimos, a nota do relatório do projeto e a discussão final.

² Inclui os 12 valores (nota máxima) obtidos com os requisitos mínimos mais 5 valores para os requisitos adicionais, no máximo de 17 valores de nota prática.

- pesquisa por cor utilizando o histograma de 12 cores (esta funcionalidade substitui a pesquisa por cor utilizando a cor dominante fornecida no ficheiro XML);
- pesquisa por **imagem exemplo** utilizando o algoritmo de “momentos de cor” para descrever cada imagem;
- áudio para melhorar a experiência do utilizador através de sons ou músicas;
- visualização das imagens no **canvas**, organizadas em modo circular (a disposição das imagens descreve uma circunferência) de acordo com a sua relevância para a *query*.
- análise dos resultados dos testes de usabilidade no relatório.

2.3 Extras

A aplicação poderá incluir os seguintes **extras** (nota prática máxima de 20 valores³):

- na pesquisa por palavras-chave, a *query* poderá ser escolhida através da seleção numa nuvem de palavras apresentadas no ecrã.
- animações que tornem a interface utilizador mais intuitiva para o utilizador;
- alteração da aplicação de acordo com a análise do *feedback* dado pelos utilizadores através dos testes de usabilidade.

2.4 Conteúdos Multimédia

A aplicação poderá conter os seguintes conteúdos multimédia:

- fotografias digitais (fornecidas pelo docente);
- áudio (sons e músicas);
- gráficos;
- animações;
- vídeos.

2.5 Base de dados

- coleção de fotos digitais fornecidas pelo docente (ficheiro images.zip);
- meta-informação das imagens da base de dados em ficheiro XML fornecido pelo docente (database.zip).

³ Inclui os 17 valores (nota máxima) obtidos com os requisitos mínimos e requisitos adicionais, mais 3 valores para os extras, no máximo de 20 valores de nota prática.

3. Pesquisa de Imagens

Os motores de pesquisa mais conhecidos utilizam palavras-chave para encontrar num base de dados imagens descritas por palavras (e.g., Google Images). Nem sempre as palavras descrevem corretamente o conteúdo da imagem. Por isso, alguns sistemas mais recentes utilizam formas alternativas para fazer a pesquisa que utilizam o conteúdo da imagem. Neste trabalho são utilizados três tipos de pesquisa:

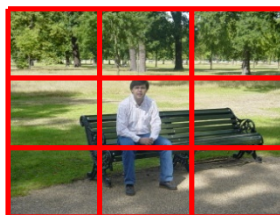
- pesquisa por **palavras-chave** - cada imagem da base de dados está classificada numa categoria (ver ficheiro XML). Para a pesquisa por palavras utilize a categoria associada a cada imagem;
- pesquisa por **cor** - cada imagem da base de dados deve ser processada de modo a calcular o número de *pixels* de um conjunto pré-definido de cores. Para esta pesquisa, calcule e utilize o **histograma de 12 cores** (as cores são indicadas pelo docente) de cada imagem;
- pesquisa por **imagem exemplo** - utilize o descritor (nome dado a um vetor de características que descrevem uma imagem), “momentos de cor no espaço HSV” para representar cada imagem neste tipo de pesquisa.

3.1 Histograma de cores

Para a pesquisa de cor, utilize o histograma da imagem com 12 cores. Para obter o histograma calcule o número de pixels na imagem para cada uma das 12 cores. Cada imagem é representada por 12 números que representam o número de ocorrências de cada uma das cores na imagem.

3.2 Momentos de cor no espaço HSV

Utilize o descritor, “momentos de cor no espaço HSV” na pesquisa por imagem exemplo. O algoritmo que permite calcular o descritor começa por dividir a imagem em 9 partes iguais (ver figura em baixo) e depois faz a extração da média e da variância de cada parte.



Cada imagem é representada por um vetor de 54 elementos ($9 \text{ partes} * 3 \text{ componentes de cor} * 2 \text{ medidas estatísticas}$).

3.3 Pesquisa de imagens baseada no conteúdo

No caso em que as imagens são descritas por palavras-chave, para realizar a pesquisa por palavras implemente uma pesquisa no XML pela palavra relativa à categoria da imagem.

No caso das pesquisas por cor e imagem exemplo, em primeiro lugar, é preciso analisar o conteúdo da imagem. Uma aplicação de pesquisa de imagens baseada em conteúdo é composta por:

- sistema para **definir queries** - para a pesquisa por cor, o utilizador escolhe uma cor como *query* e para a pesquisa por imagem exemplo, o utilizador escolhe uma imagem;
- sistema para **extrair características** - algoritmos necessários para extrair os descritores das imagens (histograma de cores para pesquisa por cor e momentos de cor para a pesquisa por imagem exemplo);
- **motor de pesquisa** - o motor de pesquisa tem como tarefa calcular uma lista ordenada de imagens, de acordo com um critério que relacione a semelhança entre a *query* e as restantes imagens da base de dados. No caso da pesquisa por cor, o motor deve ordenar todas as imagens de acordo com o *bin* do histograma que representa a cor definida como *query*. Para as pesquisas por imagem exemplo, o motor de pesquisa deve ordenar as imagens de acordo com a distância de Manhattan obtida entre a imagem *query* e as imagens da base de dados.

O processamento das imagens e o cálculo da lista ordenada pode ser demorado. Por isso, é geralmente realizado *offline* e armazenado numa base de dados antes da aplicação ser colocada *online*. A primeira parte do projeto consiste em construir esta base de dados com a informação organizada de modo a que o sistema seja mais eficaz a fazer as pesquisas. Para organizar a informação utilize o XML. Mas, em vez guardar o XML num ficheiro no servidor vamos gravar no *localStorage*.

No caso da pesquisa por cor, grave no *localStorage* a informação organizada da seguinte forma, para cada categoria:

```
<images>
  <image class="yellow">
    <path> images/beach/img1 </path>
  </image>
```

```

<image class="yellow">
  <path> images/beach/img10 </path>
</image>
...
<image class="blue">
  <path> images/beach/img2 </path>
</image>
<image class="blue">
  <path> images/beach/img10 </path>
</image>
...
</images>

```

A *key* para gravar a *string* anterior no *localStorage* é o nome da categoria. As imagens são organizadas de acordo com o número de *pixels* de cada cor. A *string* anterior deve conter para cada categoria e para cada cor as 30 imagens com maior número de *pixels*.

No caso da pesquisa por imagem exemplo, grave no *localStorage* a informação organizada da seguinte forma, para cada imagem:

```

<images>
  <image class="Manhattan">
    <path> images/beach/img1 </path>
  </image>
  <image class="Manhattan">
    <path> images/beach/img10 </path>
  </image>
  ...
  <image class="Manhattan">
    <path> images/beach/img2 </path>
  </image>
</images>

```

A *key* para gravar a *string* anterior no *localStorage* é o nome da imagem (*path*). As imagens são organizadas de acordo com a distância de Manhattan. A *string* anterior deve conter para cada imagem as 30 imagens mais semelhantes de acordo com a distância de Manhattan.

4. Desenvolvimento Centrado no Utilizador

A aplicação deverá ser concebida e implementada incluindo o utilizador no processo:

- **Análise**
 - Definição das ideias iniciais e os requisitos (fornecidos pelo docente);
 - Definição e caracterização dos utilizadores (público-alvo);
 - Definição e caracterização das tarefas do utilizador;
 - Cronograma.
- **Design**
 - Estrutura do *site*;
 - Esboços iniciais dos *layouts* da aplicação;
 - Diagrama de classes (objetos).
- **Implementação**
 - Implementação da aplicação em HTML5/JavaScript.
- **Avaliação**
 - Testes de usabilidade do protótipo implementado. Utilize o questionário desenvolvido no trabalho de laboratório 3 nos testes de usabilidade.

5. Submissão do Trabalho e do Relatório

A entrega do trabalho inclui o relatório em formato digital através da plataforma Moodle e o código fonte em HTML5/JavaScript em DVD (caso não seja possível enviar através do Moodle). O relatório deverá incluir uma introdução ao trabalho, todo o processo de desenvolvimento da aplicação, resultados obtidos, análise da informação recolhida através dos testes de usabilidade, discussão das questões mais relevantes e conclusões. Este projeto terá de ser entregue até ao dia 19 de Fevereiro de 2017. As discussões dos trabalhos práticos serão realizadas nos dias 20, 21 e 22 Fevereiro de 2017. Os alunos podem antecipar a discussão combinando uma data com o docente.