

Projeto 2: Biblioteca de imagens

LABI

André Patacas, Gil Teixeira, Sofia Vaz, Luis Andrade



Projeto 2: Biblioteca de imagens

DETI

LABI

André Patacas, Gil Teixeira, Sofia Vaz, Luis Andrade

08/05/2019

Resumo

Este relatório pretende descrever uma biblioteca de imagens pesquisável desenvolvida. Nesta, insere-se o tipo de imagem que se procura, que por sua vez é um dos dados da base de dados que tem todas as imagens. Depois, as imagens pedidas pelo utilizador são apresentadas por ordem de relevância.

Conteúdo

1	Introdução	1
2	Metodologia	2
2.1	Definição de tarefas	2
2.2	Backend	2
2.2.1	DbCommunicator.py	2
3	Resultados	4
4	Análise	5
5	Conclusões	6

Capítulo 1

Introdução

O frontend da aplicação foi feito em HTML, CSS e JavaScript, enquanto que o backend foi feito em python3. Esta aplicação foi feita no âmbito de Laboratórios de Informática no ano letivo 2018/2019. A adicionar às especificações básicas pedidas, segundo o guião sobre regras do segundo projeto, construiu-se ainda suporte para pydocs, para haja uma explicação mais detalhada de cada classe e método do nosso projeto. Este documento está dividido em quatro capítulos. Depois desta introdução, no Capítulo 2 é apresentada a metodologia seguida, tendo em conta o trabalho faseado de cada um, no Capítulo 3 são apresentados os resultados obtidos, sendo estes discutidos no Capítulo 4. Finalmente, no Capítulo 5 são apresentadas as conclusões do trabalho.

Capítulo 2

Metodologia

2.1 Definição de tarefas

Esta fase consistiu em, antes de começar sequer a trabalhar, definir quais membros do grupo fariam o quê, o que pode ser consultado na table seguinte:

Tabela 2.1: Divisão geral de tarefas

Aluno	Tarefa Atribuída
André Patacas	backend
Gil Teixeira	frontend
Sofia Vaz	relatório
Luis Andrade	testes unitários e funcionais

2.2 Backend

2.2.1 DbCommunicator.py

__init__ Este método inicializa o objeto que será adicionado à base de dados.

get_dims_and_color Este método devolve os dados da imagem à base de dados, sendo estes a cor e as dimensões, como o nome indica. Estes dados serão, depois, adicionados à base de dados.

request_characteristics Este método liga-se ao *website* fornecido pelos docentes, devolvendo um array em que cada posição é um dicionário que tem os seguintes atributos:

- nome

- classe
- *bounding box*, ou seja, a "caixa" pela qual a imagem será cortada
- "confiança" com a qual o servidor conseguiu classificar a imagem, isto é, de 0 a 1, sendo 1 o valor máximo, a certeza de que a imagem foi classificada corretamente.

add Este método adiciona cada imagem que foi adicionada até agora à base de dados. É de notar que se uma imagem tiver várias classes, isto é, se uma imagem tiver mais do que um objeto que consta na lista, então a imagem ficará replicada na base de dados, cada uma da base de dados tendo uma classe diferente. Assim, a base de dados terá os seguintes dados para cada objeto:

- nome
- altura
- largura
- quantidade de vermelho
- quantidade de verde
- quantidade de azul
- *boundig box*
- chance, de 0 a 1, da imagem ter sido classificada corretamente

É de notar que alguns destes dados são guardados no seu formato original e num formato *cropped*, sendo este *cropping* feito com uma codificação em md5 e um *digest*.

remove Ao executar este método, uma imagem dada será removida da base de dados.

get Este método devolve uma imagem baseado nos dados pelos quais a imagem for procurada. É de notar que a imagem apenas poderá ser removida da base de dados se esta existir.

request Faz com que os dados obtidos no método `request_characteristics` sejam inseridos num *array*, sendo que este já não terá nenhum dicionário associado. É importante fazer isto pois será mais fácil adicionar dados via *array* a uma base de dados do que via dicionário.

debugging Ambos os métodos `__clear_all_caution__` e `populate` são usados para efeitos de *debugging*, não sendo usados na aplicação em si.

Capítulo 3

Resultados

Descreve os resultados obtidos.

Capítulo 4

Análise

Analisa os resultados.

Capítulo 5

Conclusões

Apresenta conclusões.

Contribuições dos autores

Resumir aqui o que cada autor fez no trabalho. Usar abreviaturas para identificar os autores, por exemplo AS para António Silva. No fim indicar a percentagem de contribuição de cada autor.

Acrónimos