

Projeto Final

1 Introdução e Objetivos

O projeto final da disciplina de vocês é a especificação de um sistema de gerenciamento de preços de produtos. O objetivo principal é que o sistema de vocês permita a consulta do melhor mercado para realizar a compra de uma lista de produtos, baseado nas compras já realizadas por outros usuários (crowdsourcing¹). Imagine a situação em que você ao sair do mercado compartilhe as informações de preço dos produtos presentes na nota fiscal daquela compra. Simplesmente com uma foto da nota fiscal (ninguém gosta de ficar digitando longos textos manualmente :-)), vocês usarão uma ferramenta de OCR (Reconhecimento Óptico de Caracteres)² para reconhecer a informação textual da imagem com as informações dos produtos disponíveis na nota fiscal.³ Este projeto permitirá a vocês exercitarem todo o conteúdo visto durante o curso, incluindo desde o processo de desenvolvimento (análise, projeto e implementação), até conceitos e técnicas de programação POO, testes unitários, etc. Além disso vocês terão contanto com uma ferramenta de reconhecimento visual de texto em imagens e de serialização (a serialização consiste em salvar o estado de um objeto em um formato que possa ser armazenado na memória e posteriormente pode ser lido e usado pelo seu programa.) É importante ressaltar que seu programa deverá ser feito em linguagem C++.

2 Especificação do Sistema e Funcionalidade Básica

O seu sistema terá diferentes usuários que podem compartilhar ou buscar informações sobre os preços. Como dito anteriormente usaremos uma ferramenta de reconhecimento de texto em imagens para retirar os tipos de produtos e seus preços. Para isso vamos considerar imagens de notas fiscais “simplificadas” (no estilo das duas imagens mostradas na figura 1). Um pequeno exemplo de como utilizar a biblioteca OCR para retirar a informação textual é fornecido no arquivo <https://www.dcc.ufmg.br/~renato.martins/courses/DCC004/projeto/projeto-ocr-dcc004.zip>. Veja também como instalar e usar a Tesseract consultando o Apêndice ao final deste documento.

O seu sistema deverá permitir:

- Cadastro de Usuários: Deve ser implementado um gerenciador básico de contas de usuários no sistema. Essas contas de usuários podem ser de usuários que façam a gerência do sistema (usuários administradores),

¹<https://pt.wikipedia.org/wiki/Crowdsourcing>

²https://en.wikipedia.org/wiki/Optical_character_recognition

³Importante: Vocês não terão que programar o OCR, ele será fornecido para vocês usando a biblioteca <https://github.com/tesseract-ocr/tesseract>, como mostrado Apêndice deste documento.

Nota Fiscal #1 DCC004	
1 Sabao	3.00/U
2 Oleo	2.00/L
3 Agua sem gas	2.80/L
4 Agua com gas	5.00/L
5 Vassoura	10.00/U
6 Vinho branco	10.00/U
7 Cebola	5.00/kg

Nota Fiscal #2 DCC004	
1 Abacaxi	3.00/kg
2 Agua com gas	2.00/L
3 Macarrao	4.00/U
4 Leite	2.89/L
7 Pimenta	4.00/U
8 Suco de uva	5.00/L
9 logurte	1.00/U
10 Pao	7.00/kg

Figura 1: Exemplos de notas fiscais simplificadas de compras para a utilização com OCR.

forneçam as informações de preços (usuários contribuintes) ou que queiram fazer uma pesquisa sobre os preços (usuários consumidores). Observe que um usuário consumidor pode também ser um usuário contribuinte em diferentes momentos de utilização do sistema.

- Cadastro de Mercados: Todo usuário pode cadastrar um novo mercado no sistema, desde que este mercado já não tenha sido previamente cadastrado no sistema previamente. Os mercados são identificados pelo nome e CEP do endereço. Cada mercado contém uma lista de estoque de produtos que estão disponíveis.
- Cadastro de Produtos: Cada mercado contém uma lista de produtos disponíveis e seus respectivos preços. Ao utilizar o OCR para realizar a identificação dos produtos comprados, você terá uma lista de produtos e seu respectivo preço para aquele mercado. Caso o produto do mercado em questão ainda não fazer parte do estoque do mercado, o sistema terá que cadastrar esse novo produto. Caso o produto do mercado em questão já faça parte do mercado, o sistema deve simplesmente realizar a média ponderada do preço do produto no mercado. *Dica: mantenha uma lista com o número de vezes que o preço de um produto foi identificado via OCR para aquele mercado.*
- Tipos de Produtos/Estoque: Os produtos de cada mercado são variados e identificados pelo seu nome (não acentuado). Os produtos podem ser de vários tipos, como por exemplo: limpeza, vestuário ou alimentício (além de muitos outros). Para o caso do tipo de produtos alimentícios, podemos ter produtos líquidos (preço por litro) ou sólidos (preço por kilo e preço por unidade). Os produtos típicos do seu sistema são aqueles comumente encontrados nos mercados como frutas (banana, maçã, manga, cupuaçu,...), pastas (macarrão, massas, ...), carnes (suína, bovina, frango,...), leguminosas (feijão, lentilha,...), legumes (batata, alface, alho poro, pimentão,...). Você deve definir uma hierarquia de classes usando os conceitos de OO usando classes concretas, abstratas e interfaces para modelar os seus produtos. Podem usar da criatividade :-)
- Realização de Consulta de Preços: O seu sistema deve permitir a realização da consulta do melhor mercado para a realização da compra de uma lista de produtos que o usuário consumidor deseja comprar. Para isso você deve imprimir uma lista para o usuário mostrando o mercado mais conveniente para cada produto. Ao final da lista, você deve indicar para o usuário o melhor mercado para a compra daquela lista, baseado no número de produtos com menor preço.
- Salvar/Ler Estado do Sistema: Você deve salvar em disco o estado do sistema de todos os objetos criados (contas de usuários, supermercados, produtos, ...) usando as classes de serialização da biblioteca Boost⁴. O seu programa deverá ler os objetos serializados ao ser iniciado (caso o sistema já tenha sido utilizado).
- Funcionalidade Extra: o grupo poderá ainda propor, caso deseje, alguma funcionalidade não básica, que irá representar o aspecto de inovação do projeto. Como esta etapa não é obrigatória, ela irá compor uma pontuação extra em relação ao trabalho prático aqui proposto.

3 Documentação

Um relatório deve ser entregue ao final de forma a deixar claro as decisões tomadas relativas aos casos e detalhes que porventura estejam omissos no enunciado. Adicione também comentários gerais sobre o trabalho e as principais dificuldades encontradas. Além disso, dentre outras coisas o seu relatório deve conter:

- Introdução: descrição geral sobre o funcionamento do seu programa;
- Implementação: descrição sobre a implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas, o funcionamento das principais funções, bem como decisões tomadas. Possível descrição de extras implementados. É importante você informar neste momento como os conceitos de OO foram aplicados no seu sistema;
- Testes: proponha testes para validar o funcionamento do sistema e apresente aqui os resultados. Pode por exemplo apresentar prints dos menus, das saídas, etc., sempre relacionando com as partes correspondentes do código.
- Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
- Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso. Não tem problema olhar códigos na Internet, contanto que você não copie :)

⁴https://www.boost.org/doc/libs/1_68_0/libs/serialization/doc/index.html

4 Cometários Gerais

- Comece a fazer este trabalho logo: o prazo para terminá-lo está tão longe quanto jamais poderá estar! :)
- O programa fonte deve ser claramente comentado;
- O trabalho deverá ser feito pelo grupo de 4 alunos da turma;
- Todos os integrantes do grupo devem conhecer a totalidade do código e ter condições de explicar sua função;
- Deve ser fornecido com o código um arquivo Makefile com as opções “make” e “make run”;
- A entrega final deverá ser feita utilizando GitHub: o grupo deve criar um repositório no GitHub e me adicionar como colaborador (usuário: renatojmsdh). A correção será feita até o último commit antes da data limite para entrega;
- Mais detalhes sobre a entrega poderão ser fornecidos posteriormente.

5 Avaliação

A avaliação do projeto será dividida nas seguintes partes.

5.1 Entrega Parcial

- Data entrega: **26/11/2018**
- Definição e proposição das Users Stories (6 pts)

5.2 Entrega Final

- Data entrega: **06/12/2018** e apresentação em aula no **07/12/2018**
- Documentação: clareza, organização, etc (5 pts);
- Funcionamento correto: compila, executa, etc (10 pts);
- Uso correto de boas práticas e dos conceitos de OO (5 pts);
- Testes: proposição de testes, comportamento adequado do sistema dado comportamento inesperado do usuário, etc (4 pts);

6 Apêndice – Instruções para usar o OCR

Instale a biblioteca Tesseract⁵ com os seguintes comandos em um computador com distribuição GNU/Linux Ubuntu (ou na sua máquina virtual):

```
1 sudo apt install tesseract-ocr
2 sudo apt install tesseract-ocr-dev
3 sudo apt install tesseract-ocr-por
```

Um exemplo da utilização desta biblioteca para extrair o texto de imagens é fornecido no arquivo <https://www.dcc.ufmg.br/~renato.martins/courses/DCC004/projeto/projeto-ocr-dcc004.zip>:

```
1 #include <iostream>
2 #include <cstdlib>
3
4 int main() {
5     std::string nota = "data/nota-1.png";
6     std::string textoNota = "data/textoNota-1";
7     std::string programa = "tesseract " + nota + " " + textoNota + " -l por";
8     const char *comando = programa.c_str();
9     system(comando);
10    return 0;
11 }
```

Baixe o arquivo e compile usando `make` e em seguida execute com o comando `make run`. Um arquivo `textoNota-1.txt` será criado dentro da diretório `data` contendo a informação textual da imagem `data/nota-1.png`.

⁵<https://github.com/tesseract-ocr/tesseract>