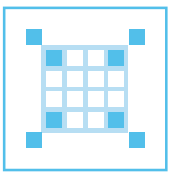

DEEC
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

Projecto de Engenharia de Software
2017/2018

Etapa 3

Implementação Mínima
Versão 1 do Software

Autoria:

Diogo M. T. Poço
2014205007
uc2014205007@student.uc.pt

João C. da C. Barreiros
2014196880
uc2014196880@student.uc.pt

João M. P. Agria
2010129833
uc2010129833@student.uc.pt

Grupo: 2

13 de Abril de 2018

Conteúdo

1 Introdução

A terceira etapa do projeto foca-se no desenvolvimento do software definido e especificado nas etapas de desenvolvimento anteriores. No final desta etapa apresentaremos a primeira versão do software, contendo uma implementação mínima de algumas características especificadas pelo cliente, o manual do utilizador e o manual de referência para o código fonte do software desenvolvido.

O conteúdo deste documento incidirá, essencialmente, sobre a forma de implementação do software e sobre as regras estipuladas e seguidas, salientando também as alterações mais importantes que se verificaram a nível do design técnico do mesmo. Serão também abordados os principais obstáculos encontrados no decorrer desta etapa, e será feita uma avaliação qualitativa do código. Será também feita uma apreciação do que foi implementado e do que ainda falta implementar, a nível dos requisitos funcionais. Convém, ainda, referir que o manual de referência e o manual do utilizador seguirão em anexo com este documento.

2 Consolidação do Design

Na etapa anterior realizámos o desenho da arquitetura proposta pela nossa equipa para a aplicação pedida, apresentando também um protótipo da interface gráfica de utilização, tendo consolidado esse trabalho num relatório. Após apreciação por parte do docente/cliente desse relatório, foi necessário proceder à consolidação desse desenho de arquitetura para melhorar o trabalho já desenvolvido. As melhorias e correções estão sublinhadas a amarelo no relatório da etapa 2 entregue em anexo. Estas foram todas as alterações realizadas:

- Apresentação de uma justificação mais satisfatória para a decisão de se utilizarem ficheiros binários para guardar as informações presentes na estrutura de dados, salientando as vantagens existentes nesta escolha.
- Explicitar concretamente em que situações é que o programa irá manipular estes ficheiros de armazenamento.

Alguns dos métodos de classes apresentados no diagrama de classes no anterior documento não foram implementados porque optámos pela utilização de métodos das classes providenciadas pelas bibliotecas do QT. Por exemplo, a classe *QList* contém métodos que facilitam a manipulação de listas ligadas e por isso alguns dos métodos que foram apresentados no diagrama de classes podem não ser implementados no código devido à sua redundância.

3 Padrão de programação

Definir um padrão de programação nesta fase do projeto permitir-nos-á criar um código-fonte consistente em todo o programa. Nesta secção apresentaremos o padrão que concordámos em estabelecer. No entanto, o código do protótipo da interface gráfica de utilização que entregámos no final desta etapa não está consistente com este padrão.

3.1 Escrita de Código

- **Nomes de classes**: Os nomes das classes iniciam-se sempre com uma letra maiúscula, e todas as palavras que se seguem também. Não se utilizam abreviações para as palavras.

  Exemplo: **class** Pagina;

- **Nomes de variáveis**: Os nomes das variáveis iniciam-se sempre com uma letra minúscula, permitindo distinguir dos nomes das classes, enquanto que todas as restantes palavras que constituam o nome devem começar com maiúscula. O nome da variável descreve a entidade a que se refere e podem existir abreviações, caso se trate de um nome longo.

  Exemplo: **QString** nomeAlbum;

- **Nomes de Constantes**: Estes devem ser escritos integralmente em maiúsculas. Para permitir a distinção entre palavras que constituam o nome, serão utilizados *underscore*. Não se utilizam abreviações em caso algum.

- **Chavetas**: Devem-se sempre colocar chavetas após um controlador de fluxo como, por exemplo, um *if*, um *elseif*, um *else*, um ciclo *for*, um ciclo *while*, entre outros. As chavetas devem abrir-se na linha seguinte ao controlador de fluxo, nunca na mesma linha que o controlador.

  Exemplo:

```
if( condição a ser avaliada )
{
        código aqui
}
```

- **Indentação**: Como se pode ver no exemplo de cima, o código dentro de chavetas deve ser indentado com tabulação, de forma a manter o espaçamento com largura consistente em todo o código.

- **Espaço entre definição de funções**: entre o fim de uma função implementada e o início da seguinte deve existir uma linha comentada com a estrutura representada por baixo deste texto. Desta forma facilitamos a leitura.

```
//================================================
```

3.2 Comentários

O código foi documentado por dois motivos distintos. O primeiro tem em vista a obtenção semi-automatizada do manual de referência com a ferramenta *Doxygen*. O *Doxygen* faz uso de comentários com uma determinada formatação para gerar o conteúdo de um manual de referência. Os comentários escritos para este efeito seguem as normas especificadas pelos criadores da ferramenta. Escolhemos comentar as linhas antecedentes de classes, atributos de classes e métodos de classes. Os comentários que documentam os ficheiros estão no início dos mesmos.

O segundo motivo que nos levou a documentar o código com comentários é que estes são uma ferramenta bastante útil. Permitem que outros programadores ou analistas percebam o algoritmo e a implementação realizada pelo criador do código. Sendo este software realizado numa equipa, os comentários tornam-se importantes para tornar a fase de integração dos diversos componentes realizados por cada elemento mais fácil e rápida. Os comentários com objetivo de facilitar a compreensão do código são adicionados ou em frente à linha de código a que dizem respeito, caso caibam numa única linha, ou nas linhas que antecedem a linha ou o bloco de código a que dizem respeito.

4 Implementação

A implementação do software de acordo com as especificações propostas e validadas pelo docente/cliente começa com a distribuição de tarefas pelos membros da equipa. Após cada um ter completado a sua parte, interligamos os componentes criados por cada um e aferimos se os requisitos estão a ser cumpridos. Nesta fase de desenvolvimento, as funcionalidade não estão completamente implementadas e não cumprem os parâmetros de qualidade exigidos. Alcançaremos esse objetivo apenas na etapa final. A tabela que se segue tem os requisitos identificados na primeira etapa deste trabalho e a prioridade atribuída pela equipa na altura e um comentário sobre o estado da implementação desse requisito nesta primeira versão do software.

	Requisitos Funcionais	Comentários	Prioridade
1	Gerir álbuns		1
1.1	Adicionar novos	A aplicação permite a criação de novos álbuns, com introdução de informação nova na estrutura de dados e criação de pasta na diretoria escolhida pelo utilizador	1
1.2	Modificar dados	Ainda não está implementada a edição de dados de álbuns	2
1.3	Eliminar	O utilizador pode eliminar um álbum à sua escolha e a aplicação remove a informação do mesmo da estrutura de dados e a diretoria é apagada, bem como todo o seu conteúdo	1
2 2.1 2.2 2.3	Pesquisar fotos por palava-chave por pessoas por datas	Apesar da sua elevada prioridade atribuída, ainda não foi implementada a funcionalidade de pesquisa de fotos.	1

3	Fazer browsing na árvore de diretórios	A navegação na árvore de diretórios está implementada	2
3.1	por miniaturas	na interface gráfica	3
3.2	por nome		3
3.3	por lista detalhada		3
4	Gerir fotos		1
4.1	Adicionar	Está implementada a adição de fotos. No entanto, só é possível escolher uma selecção de fotos dentro da mesma pasta. Ainda não está implementada a escolha de uma pasta e a inserção recursiva.	1
4.2	Copiar	Funcionalidade não implementada	3
4.3	Modificar dados	Funcionalidade não implementada	3
4.4	Eliminar	Funcionalidade implementada, mas a necessitar de otimização. Ainda não é possível eliminar mais do que uma foto seleccionada de cada vez.	1
4.5	Mover entre páginas do mesmo álbum ou páginas de álbuns diferentes	Funcionalidade não implementada	3
5	Gerir uma lista de pessoas	Apesar da sua elevada prioridade atribuída, ainda não foi implementada nenhuma funcionalidade de gestão da lista de pessoas	1
5.1	Adicionar		1
5.2	Alterar dados		2
6	Gerir páginas		1
6.1	Adicionar novas	Funcionalidade implementada	1
6.2	Mover entre álbuns	Funcionalidade não implementada	3
6.3	Copiar	Funcionalidade não implementada	3
6.4	Eliminar	Funcionalidade implementada	1
6.5	Modificar dados	Funcionalidade não implementada	2
7	Visualizar		1
7.1	Álbuns	Funcionalidade implementada	1
7.2	Páginas	Funcionalidade implementada	1
7.3	Fotos	Funcionalidade implementada	1
7.4	Pessoas e dados dos mesmos	Funcionalidade não implementada	1

8	Criar Slideshows	Funcionalidade não implementada	4
8.1	Com possibilidade de escolha de música	Funcionalidade não implementada	5
8.2	Com configuração do tempo entre duas fotos	Funcionalidade não implementada	5
8.3	Com possibilidade de apresentação organizada de fotos	Funcionalidade não implementada	5
8.3.1	Por ordem alfabética do nome	Funcionalidade não implementada	5
8.3.2	Por ordem alfabética inversa do nome	Funcionalidade não implementada	5
8.3.3	Da data mais antiga para a data mais recente	Funcionalidade não implementada	5
8.4.4	Da data mais recente para a data mais antiga	Funcionalidade não implementada	5
9	Imprimir fotos	Funcionalidade não implementada	2
10	Confirmar para apagar fotos, páginas ou álbuns	Funcionalidade implementada	3
10.1	Confirmação dupla para álbuns e páginas	Funcionalidade implementada	3
10.2	Confirmação simples para fotos e pessoas	Funcionalidade implementada	3
11	Organizar fotos de uma página	Funcionalidades implementads na interface gráficaa	4
11.1	Por ordem alfabética do nome	Funcionalidade implementada	5
11.2	Por ordem alfabética inversa do nome	Funcionalidade implementada	5
11.3	Da data mais antiga para a data mais recente	Funcionalidade implementada	5
11.4	Da data mais recente para a data mais antiga	Funcionalidade implementada	5

Como se pode verificar, muitas das funcionalidades a que atribuímos uma elevada prioridade ainda não foram implementadas. No entanto, a aplicação implementada até ao final desta etapa já permite alcançar a sua função objetivo de arquivar fotos em álbuns e pastas. Na quarta e última fase serão implementados todos os requisitos funcionais.

O código desenvolvido nesta primeira versão do software teve já em vista o seu desenvolvimento incremental posterior, pelo que, apesar de ainda haver muitas funcionalidades por implementar e

outras por melhorar, o núcleo de base estará relativamente robusto. A robustez do código pode ser quantificada pelas seguintes características presentes na aplicação:

- **Validação de Dados**: todos os dados de entrada fornecidos pelo utilizador por meio da interface gráfica e pelos ficheiros quando a aplicação inicializa têm de ser validados. A inserção de dados inválidos é respondida com notificações no caso de ser o utilizador que os fornece. No caso em que os ficheiros forneçam informação desatualizada, esta é descartada. Imagine que o utilizador, durante a utilização da aplicação, cria vários álbuns, cada um com páginas e fotos lá dentro. O utilizador fecha a aplicação e, por algum motivo, um ou vários álbuns, páginas ou fotos criadas são apagados. Quando a aplicação for iniciada novamente, os ficheiros conterão informação desatualizada.

- **Modularidade**: O programa está segmentado em módulos (classes) que gerem diferentes aspetos do sistema, providenciando robustez, na medida em que a falha de um módulo não compromete todo o sistema, e é mais simples localizar e eliminar erros de programação.

É importante notar que existem, ainda, alguns métodos passíveis a alterações. Com efeito, existem funções que foram pensadas na etapa de *design*, que não irão ser implementadas como havia sido projetado. Como já foi referido anteriormente, a utilização das bibliotecas de classes do Qt facilita muito a manipulação da estrutura de dados.

Nesta fase foi dada baixa prioridade à robustez do código devido à falta de tempo. Por isso, é possível existirem várias funções que podem provocar falhas no software. Assim, admitimos que a qualidade do código nesta etapa apresenta-se ainda significativamente reduzida, sendo uma das prioridades da próxima etapa melhorar substancialmente essa qualidade.

5 Principais obstáculos

Durante esta etapa de desenvolvimento surgiram alguns obstáculos que a equipa de desenvolvimento procurou ultrapassar. O fator mais limitativo para a equipa foi a limitação de tempo para o desenvolvimento do software e respetiva documentação. Os seguintes fatores contribuíram para a falta de tempo:

- Falta de conhecimentos sobre o ambiente de programação QT. A inexperiência de dois elementos da equipa na utilização desta ferramenta levantou alguns problemas pontuais, que atrasaram o processo. A inexperiência dos mesmos dois a nível de programação de interfaces gráficas também provocou algum atraso na implementação. Houve assim, a necessidade de um período de familiarização.

- Falta de conhecimentos a nível do software *Doxygen*, para a construção automatizada do manual de referências.

- Dificuldades sentidas durante a conjugação do código com a interatividade do GUI.

- Dificuldades de integração das contribuições de cada elemento da equipa de desenvolvimento no projeto. Este facto limita fortemente a produtividade da equipa. A falta de experiência de realização de projetos de software em equipa provoca, por vezes, a distribuição e gestão ineficiente de tarefas e tempo.

Para ultrapassar estes obstáculos, houve um esforço elevado para aprender eficazmente a trabalhar com as ferramentas utilizadas e, também, uma repartição de tarefas pelos diferentes membros da equipa. No entanto, sendo que, tanto o manual de referência, como o manual do utilizador necessitam que o software esteja já operacional, estes dois foram feitos com tempo demasiado reduzido, não permitindo assegurar a qualidade desejada dos mesmos.

É de notar também que houve alguma dispersão na implementação das funcionalidades. Isto é, algumas funcionalidades não essenciais foram implementadas, ainda que outras mais importantes não o tivessem sido. Todos estas barreiras provocaram um agravamento à falta de tempo, o que por sua vez levou à redução da qualidade da documentação e da implementação do código.

6 Conclusão

O docente referiu em diversas ocasiões que a segunda etapa do projeto era o ponto fulcral para a implementação do produto final. Foi visível durante este terceira fase que um bom *design* facilita a organização do trabalho. Dadas as várias cadeiras que cada um dos elementos da equipa de desenvolvimento frequenta e os trabalhos que estas acarretam, o trabalho ficou substancialmente atrasado. O grupo ficou satisfeito com os requisitos funcionais implementados mas esperava ter tempo para implementar mais do que conseguiu. Na próxima etapa iremos melhorar substancialmente a qualidade do nosso código, em termos de comentários e de robustez, e completar o manual de referência e o manual de utilizador.