

ANÁLISE E IMPLEMENTAÇÃO DA ESPECIFICAÇÃO

**"EAT & DRINK"**

MÓDULO FICHEIROS

Curso: Licenciatura em Engenharia Informática

Unidade Curricular: Sistemas de Informação Distribuídos

Docente: Maria Albuquerque

Grupo:

António Raimundo | 54408

Diogo Peres | 54411

José Serro | 54415

Nuno Coelho | 54378

24/03/2014

# Análise do módulo

A seguinte análise será feita utilizando uma escala consoante o documento que nos foi entregue e disponibilizado no e-learning. Será feita uma análise por vários pontos importantes como a Legibilidade, a Completude das Especificações e a Qualidade da solução.

# Legibilidade

Quanto à legibilidade dos diagramas não houve dificuldades nenhumas. Os diagramas estão bem apresentados e, completamente, legíveis.

Foi detectado a ausência do diagrama de *Use-Case*, o que faz com que complique a leitura / percepção da especificação, uma vez que este é o diagrama que traduz de uma forma simples as principais funcionalidades do módulo.

As opções são claras e coerentes, embora se tenha identificado algumas decisões que poderiam ser melhoradas, mas nada que implique o mau funcionamento do sistema.

Foi, também, detectado uma falta de descrição / explicação em texto dos diagramas o que dificulta a compreensão dos mesmos.

Os diagramas são fáceis de interpretar e em geral estão agradáveis de analisar. Os formulários disponibilizados dão uma melhor interpretação do sistema junto com os diagramas de sequência.

Em conclusão, utilizando uma escala de 1 (mau) a 5 (excelente), fizemos a seguinte análise à Legibilidade dos diagramas:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pontos / Escalas: | **1** | **2** | **3** | **4** | **5** |
| **Facilidade de leitura** |  |  |  |  | X |
| **Opções / Justificações** |  |  | X |  |  |
| **Interpretação / Dúvidas** |  |  |  | X |  |

# Completude das Especificações / Problemas

Quanto à completude das especificações foram detetados vários problemas. Foram identificadas várias funções que deveriam ser métodos e vários métodos que deveriam ser funções. Há funções / métodos com ausência de parâmetros / argumentos. Há métodos que necessitam de informações que não estão definidas nas classes (exemplo: ***uploadFotografia()*** foi preciso adicionar o ID do comentário para que se possa adicionar na base de dados a fotografia ao comentário correto).

Há falta de interacção entre as classes de swing (interfaces visuais). Também foi detectado acesso à base de dados pouco informativos e muito incompletos (exemplo: método ***save()***da classe de acesso à base de dados).

Um problema especifico do segundo diagrama de sequência: não faz sentido pesquisar todos os comentários quando quer-se adicionar uma nova fotografia se já foram, supostamente e como indicado nos formulários, carregados anteriormente e visto que é preciso saber o comentário previamente antes de adicionar uma fotografia ao mesmo.)

Devido a estes e outros problemas foram modificados métodos, funções e interacções da especificação proposta para que o sistema funcione.

Em conclusão, utilizando uma escala de 1 (mau) a 5 (excelente), fizemos a seguinte análise quantitativa:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pontos / Escalas: | **1** | **2** | **3** | **4** | **5** |
| **Ausência de métodos** |  |  |  | X |  |
| **Ausência de parâmetros** | X |  |  |  |  |
| **Interação entre formulários** |  | X |  |  |  |
| **Acessos à base de dados** | X |  |  |  |  |

# Qualidade da solução

Quanto à qualidade da solução, o utilizador à partida terá dificuldades ao utilizar o programa, pois falta informação mais detalhada nos formulários.

Em termos de modularidade, o código não será redundante e está bem dividido, por vezes até em demasia. Está bem encapsulado e gere-se de forma autónoma.

Finalmente em termos de flexibilidade, há alguns problemas na comunicação com outros componentes, mas são facilmente corrigidos.

Em conclusão, utilizando uma escala de 1 (mau) a 5 (excelente), fizemos a seguinte análise quantitativa:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pontos / Escalas: | **1** | **2** | **3** | **4** | **5** |
| **Usabilidade** |  | X |  |  |  |
| **Modularidade** |  |  |  | X |  |
| **Encapsulamento** |  |  |  | X |  |
| **Flexibilidade** |  |  | X |  |  |

# Implementação da especificação

Após a análise da especificação dada, foi implementado o módulo ficheiros (comentários e fotografias) consoante a especificação dada pelo grupo 5-dia.

Como já havíamos visto anteriormente existem vários problemas com a completude da especificação. Esses problemas foram ultrapassados com a implementação / alteração de métodos e funções da especificação.

Não foi feita a implementação da funcionalidade da foto de perfil do utilizador, por não haver nenhum formulário/interface para a mesma. Esta decisão foi tomada em reunião com o coordenador da UC.

Foi suposto que a avaliação dos estabelecimentos e pratos é feita numa escala de 0 a 5, com base no formulário/interface, pois não foi especificado no relatório do grupo 5-dia.

De notar que por falta de descrição/explicação dos diagramas, na ausência de detalhe o grupo optou por resolver o problema da maneira mais coerente possível, de modo que as funcionalidades não se alterassem.

# Detalhes da Implementação (Problema / Solução adoptada)

**save()** : boolean (todos os métodos save() )

O método não leva argumentos quando é chamado para guardar dados na base de dados, no entanto é necessários argumentos. Optou-se por passar os dados necessários para guardar na base de dados.

**2.º diagrama de sequência /Adicionar foto a comentário**

Não faz sentido pesquisar todos os comentários existentes na base de dados antes de adicionar uma foto a um comentário, porque os comentários já estão na pesquisa feita anteriormente. Esta pesquisa extra foi descartada, não tendo influência na funcionalidade

**uploadFotografia(File file):** boolean

O método não leva os argumentos necessários para funcionar. Foi preciso adicionar as informações do comentário onde a foto vai ser inserida para que a inserção na base de dados seja feita com sucesso.

**"TextFields"** da Interface gráfica

Foram alterados alguns *TextFields* para *JComboboxs*, pois não faz muito sentido o utilizador ter de adivinhar que estabelecimentos/pratos/utilizadores existem. Esta alteração foi decidida em reunião com o coordenador da UC.

**Comentários**

Os comentários não têm data, logo não se percebe porquê a data no comentário no formulário/interface de comentário.

Adicionar / Editar / Ver / Eliminar Comentário

**save(TipoComentario tipo, int id, String comentario, int nota):** boolean

Não existe *ID* de comentário. O argumento foi alterado, adicionando-se todas as informações do comentário, nomeadamente o email do utilizador, o *ID* do Prato e o *ID* do estabelecimento.

**Existência Fotografia no comentário**

Para verificar se o comentário tem fotografia é preciso fazer dois *selects* à base de dados. Como as *Querys* à base de dados não foram especificadas no relatório de especificação, optou-se por criar outro método para verificar a existência de fotografia no comentário, não afectando a funcionalidade.

**Diagrama de sequência para ver fotografia**

Não existe diagrama de sequência para ver fotografia. Foi utilizada a mesma interface para a adicionar uma fotografia e o processo foi definido pelo grupo da forma mais simples possível, não alterando as funcionalidades da especificação.