Universidade da Madeira

Interacção Homem-Máquina
2006/2007

Projecto *Interface Killer*

Relatório Final

Elaborado por:
Grupo 8

João Gouveia	-	Engenharia Informática	-	2066502
João Pestana	-	Ensino de Informática	-	2046403
Ricardo Rodrigues	-	Engenharia Informática	-	2064903

INDÍCE

1. Inquérito e resultados 3
2. Análise de utilizadores e de tarefas 6
2.1. Mapa de papeis de utilizador 6
2.2. Contexto, características e critérios de cada papel de utilizador 6
2.2.1. GestorProjecto 7
2.2.2. VisualizadorDeScreenshots 7
2.2.3. Consultor de defeitos 7
2.2.4. Identificador de defeitos 8
2.2.5. Registador de defeitos 8
2.3. Brainstorm de casos de utilização 9
2.4. Mapas de casos de utilização 10
2.5. Matriz de suporte aos casos 11
2.6. Narrativas dos casos de utilização 12
2.7. Modelo de domínio em diagrama de classes UML 16
3. Desenho 17
3.1. Protótipo abstracto canónico 17
3.2. Mapa de navegação 18
3.3. Avaliação Heurística dos Protótipos de Baixa Fidelidade (Grupo 12 – Interface Killer) 19
4. Protótipo funcional 21
4.1. Avaliação Heurística do Protótipo Funcional (Grupo 12 – Interface Killer) 21
4.2. Protótipo Funcional do Interface Killer 22

ANEXOS:

Anexo 1: Protótipos de Baixa Fidelidade..... *PBFs_InterfaceKiller_IHM_G8.pdf*
Anexo 2: Storyboards……..…… *Storyboards_InterfaceKiller_IHM_G8.pdf*

1. Inquérito e resultados

Na primeira fase do projecto começamos por realizar um inquérito, com vista a apurar quais as metodologias de trabalho da nossa população alvo.
Passamos então a apresentar o inquérito e os dados obtidos.

Inquérito

Interface Killer: uma ferramenta de registo de defeitos de usabilidade

A ferramenta deverá pedir ao utilizador a introdução dos *screenshots* dos ecrãs em causa (num formato escolhido por si, p.e. JPEG, GIF ou BMP), pois um defeito de usabilidade é identificado com uma localização no *screenshot*. Tipicamente os utilizadores deste tipo de ferramentas recorrem ao papel e a *templates* estruturadas para anotarem e descreverem os defeitos de usabilidade que vão encontrando.
Um defeito de usabilidade é definido então por uma **localização** no ecrã; pela descrição do tipo de **feature** da interface (botão, cor, comando, diálogo, campo de texto, função, gráfico, ícone, item, etiqueta, menu, mensagem, etc.); pelo tipo de **problema** (confuso, propício a erros, distracção, não-informativo, etc.); e pelo **grau de severidade** do problema (indicado por um número, 4-crítico, 3-significativo, 2-menor, 1-nominal e ainda ?-por avaliar).

Questões:

Pode responder as questões seguintes colocando um "x" á frente de cada opção.

1- Com que frequência costuma fazer análise de usabilidade?
   a) Nunca
   b) Pelo menos uma vez na vida
   c) Pelo menos uma vez por ano
   d) 1 a 5 vezes por mês
   e) 6 a 20 vezes por mês
   f) Mais de 20 vezes por mês

   1.1 Se Respondeu "Nunca" na pergunta 1., tenha o resto de um bom dia! Obrigado ☺

2- Que sistema operativo mais utiliza?
   a) Windows
   b) Mac OS
   c) Linux
   d) Solaris
   e) Palm OS
   f) Outro:_____.

3- Conhece alguma aplicação parecida com o Interface Killer?
   a) Sim. Qual?_____.
   b) Não

4- Costuma sofrer pressão durante o seu trabalho ?
   a) Sim
   b) Não

5- Costuma trabalhar em grupo?
   a) Sim
   b) Não

6- Para que tipo de aplicações costuma fazer analise de usabilidade?
   a) Páginas web
   b) Ferramentas de trabalho
   c) Jogos
   d) Outros:_____.

7- Que tarefa desempenharia mais vezes?
   a) Navegar entre os screenshots
   b) Identificar defeitos nos screenshots
   c) Dar uma avaliação aos screenshots
   d) Todos por igual
   e) Outra:_____.

8- Como costuma organizar os screenshots?
   a) Por nome da screenshots
   b) Por data
   c) Por nome da aplicação
   d) Outro:_____.

9- Se tivesse como procurar uma determinada screenshot, o que utilizaria para fazer:
   a) Pelo nome da screenshot
   b) Pelo nome da aplicação
   c) Pelo tipo de erro
   d) Outro:_____.

10- Como classifica os erros?
   a) Por gravidade do erro
   b) Por tipo de erro
   c) Outro:_____.

11- Qual a melhor maneira de identificar erros numa imagem?
   a) Círculos
   b) Cruzes
   c) Números
   d) Outros: _____.

12- Como trata a informação após a identificação dos erros?
   a) Arquivar informação
   b) Enviar informação para terceiros
   c) As duas anteriores
   d) Outros:_____.

Os resultados obtidos, foram os seguintes:

RESULTADOS:

Universo deste inquérito: 7 pessoas

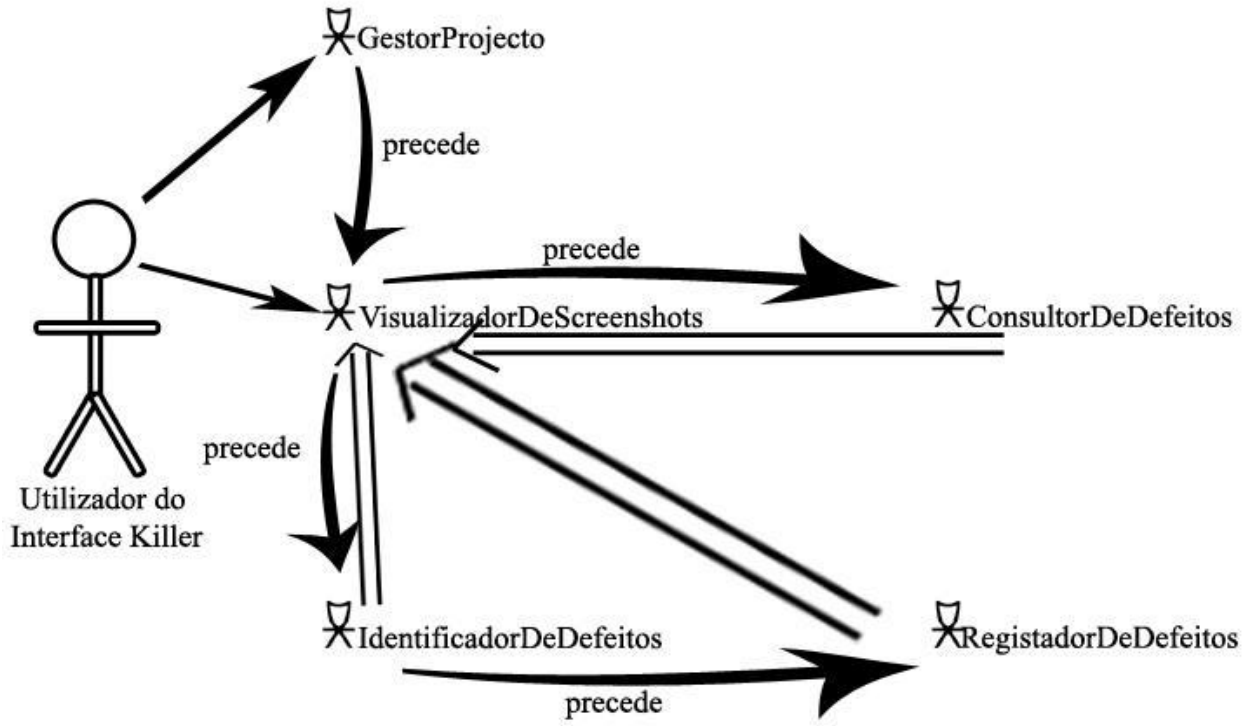
Questões	**Respostas** (alínea – número de respostas)	**Observações**
1	b) – 1; c) – 4; e) – 2	
2	a) – 6; b) – 1	
3	a) – 2; b) – 5	Das respostas "Sim", *netBeans*, *Paint* são as aplicações conhecidas que o inquirido acha parecida ao Interface Killer.
4	a) – 5; b) – 2	
5	a) – 7	
6	a) – 5; b) – 4; c) – 1	Um dos inquiridos respondeu *projectos alheios*.
7	a) – 2; b) – 2; c) – 1; d) – 2	
8	a) – 4; b) – 2; c) – 1	
9	a) – 5; b) – 2; c) – 2	
10	a) – 2; b) – 6	
11	a) – 5; c) – 3	
12	a) – 4; b) – 1; c) – 2	

De notar que, o baixo número de elementos da nossa amostra, deve-se à dificuldade em encontrar pessoas que se encontrassem dentro da área em que o projecto se insere.

2. Análise de utilizadores e de tarefas

Nesta fase, foi feita a análise de utilizadores e tarefas, isto é, analisou-se do ponto de vista do utilizador, quais os papéis que estes podiam desempenhar e que tarefas podiam efectuar, no desempenho de cada um desses papéis.

2.1. Mapa de papeis de utilizador



2.2. Contexto, características e critérios de cada papel de utilizador

2.2.1. GestorProjecto

Contexto:

- A sua função é criar projectos, abrir projectos já criados, remover projectos, e carregar ou remover as screenshots nos projectos onde irá trabalhar. Basicamente ele gere os projectos.
- O objectivo é permitir que as screenshots estejam disponíveis para a análise de usabilidade nos projectos criados, e que seja possível navegar entre as screenshots.

Características:

- Este papel é feito com alguma frequência.
- Apenas preocupa-se em gerir os projectos onde são carregadas screenshots ou removidas e também na navegação entre elas.
- Lida com alguns meios

Critérios:

- Carregamento de screenshots fácil.
- Identificação das screenshots não ambígua.
- Navegação entre as screenshots simples.

2.2.2. VisualizadorDeScreenshots

Contexto:

- A sua função é navegar entre os screenshots.
- Lida com vários screenshots.
- Normalmente trabalha sem grandes pressões.
- Escolhe quais as screenshots que deseja visualizar .

Características:

- Não é necessário muita experiência para navegar e visualizar screenshots .
- Navegação e visualização repetitivas, cada vez que é usada a aplicação.
- Tolerante ás más escolhas de visualização.

Critérios:

- Ser rápido a navegar entre screenshots.
- Não existir ambiguidades na escolha da screenshot a visualizar.

2.2.3. Consultor de defeitos

Contexto:

- A sua função é consultar defeitos já registados.
- O objectivo é consultar quais os defeitos que cada screenshot tem
- Pode trabalhar em grupo.
- Visualiza as screenshots sobre as quais consulta os defeitos.
- Consulta detalhadamente cada defeito

Características:

- Este papel é feito com alguma frequência.
- Consultar os defeitos de uma screenshot poderá ser repetitivo
- Apenas consulta, não interage muito com o sistema

- É necessário ter alguma experiência de análise de usabilidade.

Critérios:
- Era desejado que as consultas sejam rápidas e não ambíguas.

2.2.4. Identificador de defeitos

Contexto:
- A sua função é identificar defeitos na screenshot.
- O objectivo é marcar na screenshot o defeitos, sem se preocupar com a sua descrição.
- Pode trabalhar em grupo.
- Visualiza a screenshot sobre a qual identifica os defeitos.

Características:
- Este papel é feito com alguma frequência.
- Vai interagir muito, caso existam muitos defeitos.
- É necessário ter alguma experiência de análise de usabilidade.

Critérios:
- Era desejado identificar o defeito quando clicamos, na screenshot, o local da identificação.
- É rápido e de fácil utilização.

2.2.5. Registador de defeitos

Contexto:
- A sua função é registar defeitos.
- O objectivo é descrever detalhadamente o defeito.
- A identificação do defeito precede ao registo do defeito.
- O ambiente de trabalho é um formulário onde será preenchido as características e detalhes do defeito.
- Pode trabalhar em grupo.

Características:
- Não é muito frequente ao contrário da visualização de screenshots.
- É necessário preencher um formulário para cada defeito a registar.
- É esperado alguma experiência em análise de usabilidade.
- Utilizador já sabe razoavelmente bem qual o defeito a registar e os atributos necessários a preencher.

Critérios:
- É prioritário ter pelo menos os campos de tipo de defeito e grau de severidade.
- É necessário que seja objectivo, sem ambiguidades.

2.3. Brainstorm de casos de utilização

Após a realização de brainstorm com vista a apurar quais os casos de utilização mais relevantes obtivemos os seguintes resultados:

1. VisualizandoScreenshot
2. IdentificandoDefeito
3. CarregandoScreenshot
4. RegistandoDefeito
5. NavegandoScreenshot
6. ConsultandoDefeito
7. EditandoDefeito
8. RemovendoScreenshot
9. CriandoProjecto
10. GravandoProjecto
11. AbrindoProjecto
12. GravandoTrabalhoScreenshot
13. TrabalhandoNoProjecto
14. EliminandoDefeito
15. RemovendoProjecto

Em seguida fez-se uma tabela com vista a apurar qual a importância e frequência de cada um dos casos.

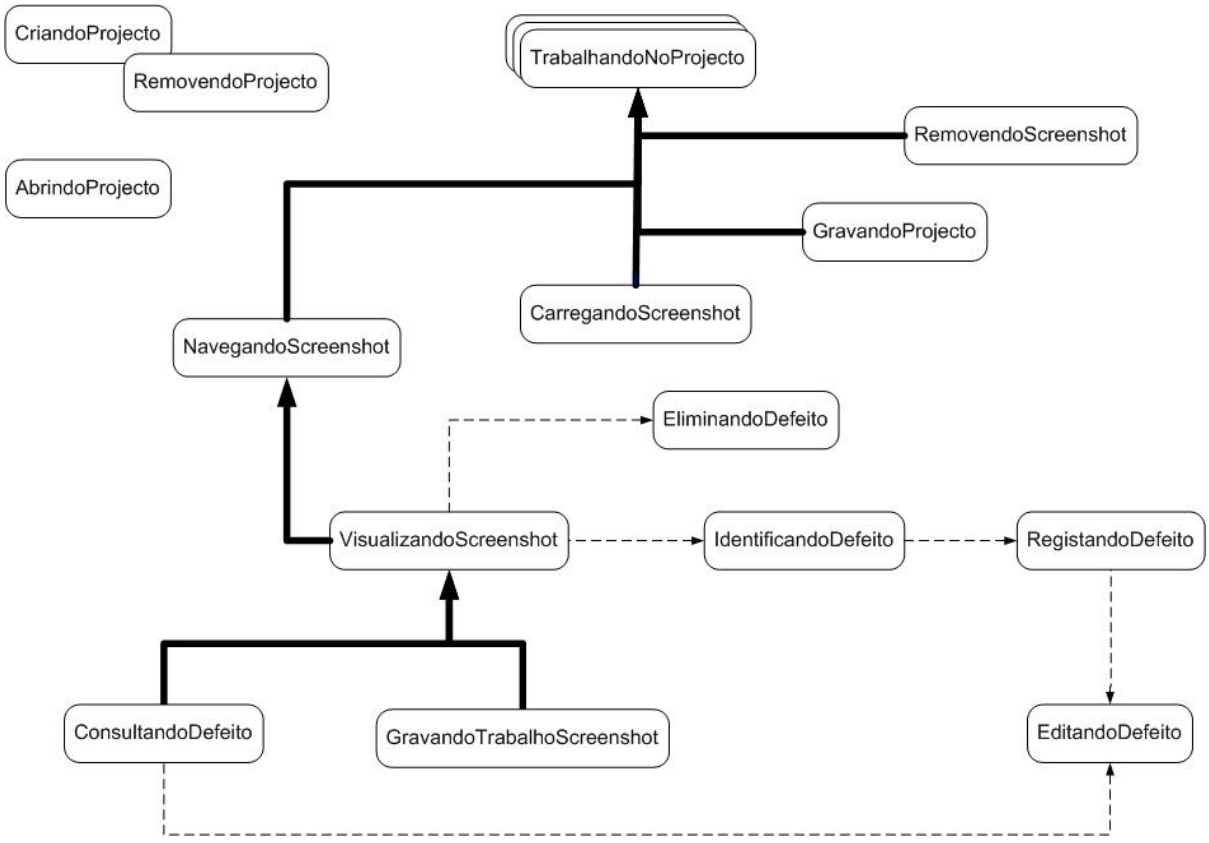
Casos de uso	Frequência (F)	Importância (I)	F&I
1	9	9	18
2	8	10	18
3	7	9	16
4	7	7	14
5	9	5	14
6	7	7	14
7	5	6	11
8	5	6	11
9	6	7	13
10	8	9	17
11	9	7	16
12	10	8	18
13	10	7	17
14	4	6	10
15	3	7	10

Nota: F e I são dadas por um valor de 1 a 10, sendo a mais alta a que tem mais peso em termos de frequência e importância.

Com bases nestes resultados, ordenamos os casos de utilização como se apresenta na próxima tabela:

1º. VisualizandoScreenshot; IdentificandoDefeito; GravandoTrabalhoScreenshot.	**5º.** CriandoProjecto.
2º. GravandoProjecto; TrabalhandoNoProjecto.	**6º.** EditandoDefeito; RemovendoScreenshot.
3º. CarregandoScreenshot; AbrindoProjecto.	**7º.** EliminandoDefeito; RemovendoProjecto.
4º. RegistantoDefeito; NavegandoScreenshot; ConsultandoDefeito.	

2.4. Mapas de casos de utilização



2.5. Matriz de suporte aos casos

Nesta secção, identificamos quais os casos de utilização que cada papel pode desempenhar

	Papel 1	Papel 2	Papel 3	Papel 4	Papel 5
Caso 1		X	X	X	
Caso 2				X	
Caso 3	X				
Caso 4					X
Caso 5	X	X			
Caso 6			X		X
Caso 7			X		X
Caso 8	X				
Caso 9	X				
Caso 10	X				
Caso 11	X				
Caso 12		X			
Caso 13	X				
Caso 14			X		
Caso 15	X				

Lista de Casos:

1. VisualizandoScreenshot
2. IdentificandoDefeito
3. CarregandoScreenshot
4. RegistandoDefeito
5. NavegandoScreenshot
6. ConsultandoDefeito
7. EditandoDefeito
8. RemovendoScreenshot
9. CriandoProjecto
10. GravandoProjecto
11. AbrindoProjecto
12. GravandoTrabalhoScreenshot
13. TrabalhandoNoProjecto
14. EliminandoDefeito
15. RemovendoProjecto

Lista de Papeis:

1. GestorProjecto
2. VisualizadorDeScreenshots
3. ConsultorDeDefeitos
4. IdentificadorDeDefeitos
5. RegistadorDeDefeitos

2.6. Narrativas dos casos de utilização

Caso 1: VisualizandoScreenshot	
Intenção do Utilizador	**Responsabilidade do Sistema**
Escolher screenshot	
	Mostrar screenshot

Caso 2: IdentificandoDefeito	
Intenção do Utilizador	**Responsabilidade do Sistema**
Escolher ferramenta de identificação	
	Adaptar-se á ferramenta
Indicar localização (x,y) do defeito	
	Mostrar screenshot com o defeito marcado
	Dar opção de detalhar o defeito (Se no momento ou mais tarde)
Escolher opção	
	Guardar posição do defeito na screenshot

Caso 3: CarregandoScreenshot	
Intenção do Utilizador	**Responsabilidade do Sistema**
Escolher screenshot externamente	
	Carregar screenshot no projecto activo
	Pedir o nome da screenshot
Introduzir o nome da screenshot (ou não, ficando por defeito o nome do ficheiro da imagem)	
	Actualizar projecto

Caso 4: RegistandoDefeito	
Intenção do Utilizador	**Responsabilidade do Sistema**
	Apresentar formulário
Preencher detalhes relativos ao defeito	

	Pedir confirmação
Confirmar (ou anular)	
	Guardar dados

Caso 5: NavegandoScreenshots	
Intenção do Utilizador	**Responsabilidade do Sistema**
	Apresentar screenshots disponíveis do projecto activo
Seleccionar screenshot	
	Marcar screenshot

Caso 6: ConsultandoDefeito	
Intenção do Utilizador	**Responsabilidade do Sistema**
Escolhe defeito a visualizar	
	Mostrar informação do defeito

Caso 7: EditandoDefeito	
Intenção do Utilizador	**Responsabilidade do Sistema**
Alterar detalhes do defeito	
	Pedir confirmação
Confirmar (ou anular)	
	Guardar dados

Caso 8: RemoveScreenshot	
Intenção do Utilizador	**Responsabilidade do Sistema**
	Mostrar screenshots disponíveis do projecto activo
Seleccionar screenshot	
	Marcar screenshot
	Pedir confirmação
Confirmar (ou anular)	
	Apagar screenshot (ou sem acção)

Caso 9: CriandoProjecto	
Intenção do Utilizador	**Responsabilidade do Sistema**
	Pedir nome do projecto
Introduzir nome do projecto	
	Pedir localização onde criar o projecto

Introduzir localização	
	Guardar

Caso 10: GravandoProjecto	
Intenção do Utilizador	**Responsabilidade do Sistema**
Escolher gravar o projecto (ou escolher o caminho onde gravar)	
	Guardar projecto activo

Caso 11: AbrindoProjecto	
Intenção do Utilizador	**Responsabilidade do Sistema**
Escolher projecto (ou introduzir localização)	
	Abrir projecto

Caso 12: GravandoTrabalhoScreenshot	
Intenção do Utilizador	**Responsabilidade do Sistema**
Escolher gravar	
	Gravar screenshot activa

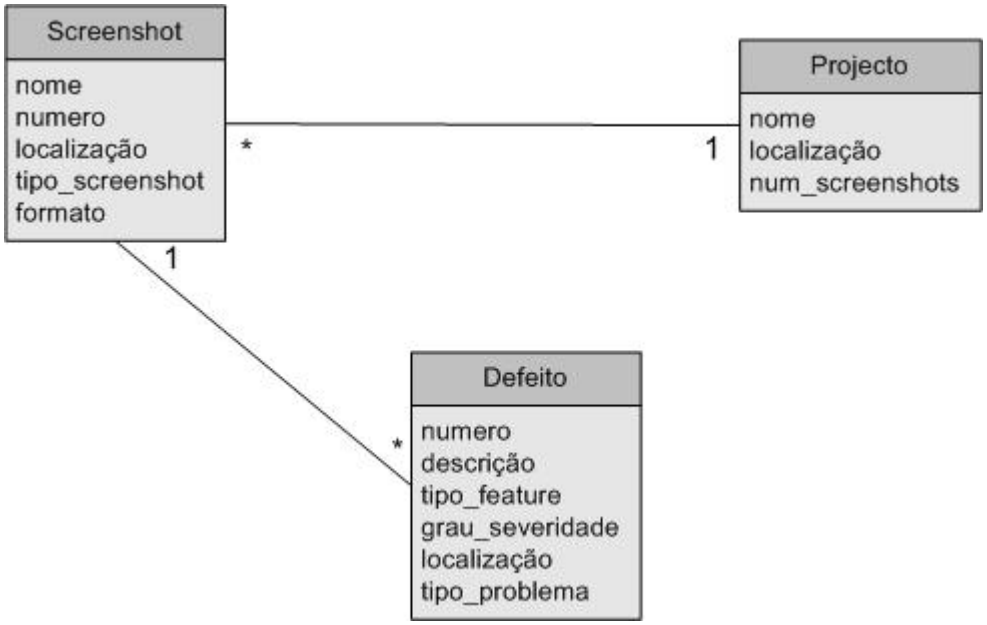
Caso 13: TrabalhandoNoProjecto	
Intenção do Utilizador	**Responsabilidade do Sistema**
	Mostrar tarefas disponíveis a executar sobre o projecto no geral
Escolher tarefa	
	Executar tarefa

Caso 14: EliminandoDefeito	
Intenção do Utilizador	**Responsabilidade do Sistema**
Seleccionar defeito da screenshot por eliminar	
	Pedir confirmação
Confirmar (ou anular)	
	Guardar

Caso 15: RemovendoProjecto	
Intenção do Utilizador	**Responsabilidade do Sistema**
Seleccionar projecto e pedir para remover	
	Pedir confirmação
Confirmar (ou anular)	

	Apagar projecto (ou sem acção)

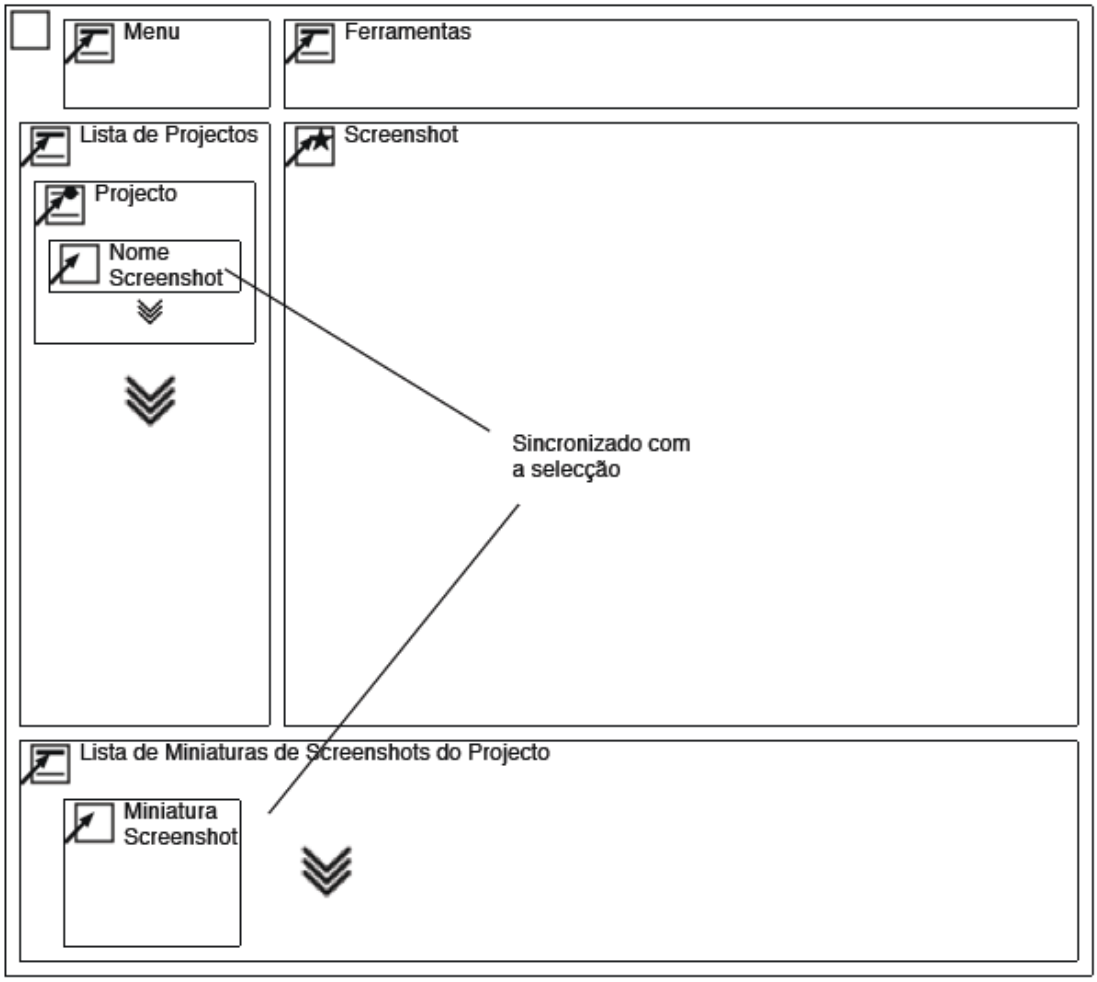
2.7. Modelo de domínio em diagrama de classes UML



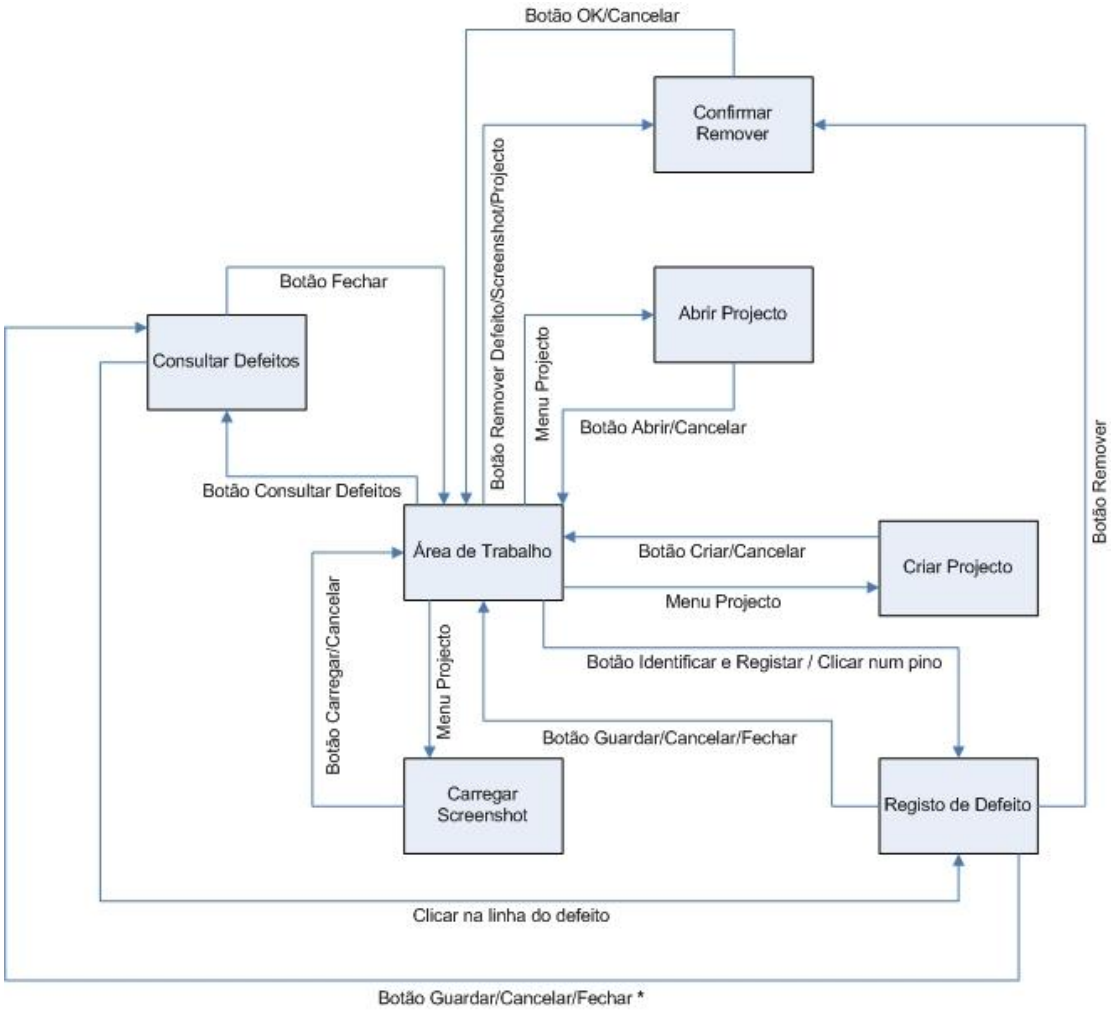
3. Desenho

Para esta fase, desenhou-se os PAC's, protótipos abstractos canónicos, mapas de navegação e *storyboards* com vista à construção de um primeiro protótipo de baixa fidelidade (pbf).
Após o término do desenho o pbf foi sujeito a uma avaliação heurística por parte de outro grupo. Em anexo estão os nossos pbf e storyboards, bem como os pbf do outro grupo, aos quais efectuamos uma avaliação heurística.

3.1. Protótipo abstracto canónico



3.2. Mapa de navegação



* Se a janela anterior tenha sido a Consultar Defeitos.

3.3. Avaliação Heurística dos Protótipos de Baixa Fidelidade (Grupo 12 – Interface Killer)

Como já se referiu no fim o protótipo foi sujeito a uma avaliação heurística. Apresentamos então a avaliação heurística que efectuamos na aula prática, aos pbf do grupo 12. Os pbf foram entregues anteriormente e por isso, não podemos voltar a coloca-los neste relatório.

No primeiro PBF avaliado:

- Heurística 6
  - pois temos de nos lembrar onde se encontram os comandos. Por exemplo remover a screenshot é necessário.

- Heurística 7
  - Porque não é eficiente ter uma screenshot com vários títulos para cada defeito que conter. Não será possível visualizar os defeitos de uma screenshot todos de uma vez.
  - Porque é difícil navegar entre screenshots. Se tivermos 50 screenshots , como navegar entre elas? Ou escolher qual a que trabalhar.
    Sugestão: Uma lista de miniaturas de screenshots, ou uma lista com os nomes das screenshots era mais eficiente para mudar o ambiente de trabalho.

- Heurística 8
  - pois o tamanho da screenshot pode ser demasiado pequena para poder identificar erros nela. Sugestão : Reduzir o tamanho das outras componentes, ou adicionar uma barra extensível para aumentar ou diminuir a área da screenshot.

No segundo PBF avaliado:

- Não conseguimos avaliar a heurística deste PBF por ser demasiado abstracto.

E por último, no terceiro PBF avaliado:

- Heurística 1
  - Porque há uma perca de contexto, por estarmos sempre a mudar de ecrã. O ideal era ser possível executar operações como mover, guardar, identificar defeito, etc., sem termos que mudar de ecrã.

- Heurística 7
    - pois a navegação de screenshots é pouco eficiente.
      Sugestão: Uma lista de miniaturas de screenshots era mais eficiente para mudar o ambiente de trabalho.

4. Protótipo funcional

Para esta fase, começou-se por fazer uma primeira versão do protótipo funcional, em MXML, e realizou-se uma avaliação heurística ao protótipo do grupo 12, a qual passamos a apresentar.

4.1. Avaliação Heurística do Protótipo Funcional (Grupo 12 – Interface Killer)

As heurísticas que eram transgredidas no protótipo do grupo 12 foram as seguintes:

- Heurística 1
  - Na descrição do erro, quando seleccionamos uma das opções das Combo Box, não é possível verificar a opção, visto que o título que aparece é o nome do botão e não o do seleccionado.

- Heurística 4
  - Termo usado para defeito devia ser apenas em neste caso termos defeito e erros.
  - Têm botões em inglês e outros em português

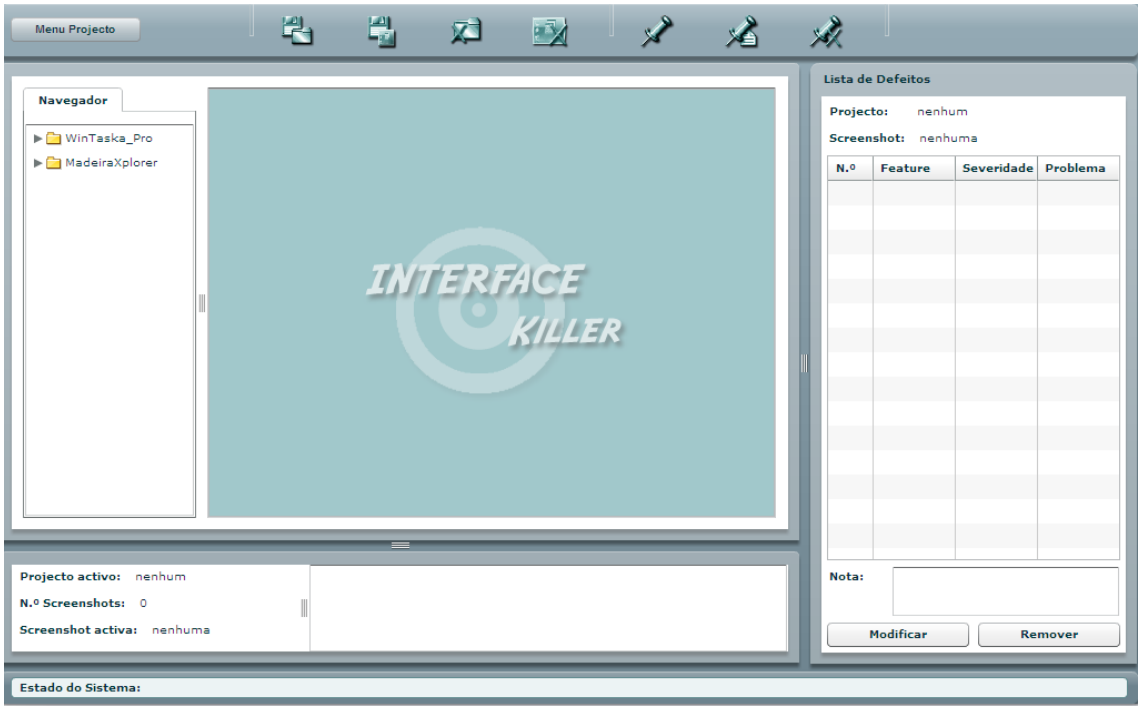
- Heurística 6
  - Se a lista de erros for demasiado longa, será necessário recordar onde se encontra o erro; Se tivesse organizado por screenshot seria mais fácil de encontrar.

- Heurística 7
  - Se uma screenshot tiver mais que 1 defeito não é eficiente estarmos a repetir a mesma screenshot para identificar outro erro.
  - Não há uma maneira rápida de remover defeitos e marcar erro, pois é sempre necessário ir escolher o menu opção.

- Heurística 8
  - A área de visão das screenshots pode ser demasiado pequena. Não é possível aumentar essa área para uma melhor percepção do erro.

- Heurística 10
  - Não tem ajudas rápidas, tipo tooltips; sempre que se tem uma dúvida temos que ir ao menu about.

4.2. Protótipo Funcional do Interface Killer



No nosso protótipo funcional do *Interface Killer*, não foram implementadas algumas das funcionalidades que desejaríamos. Abaixo listamos algumas das mais relevantes:

- Era desejável associar uma lista de consulta de defeitos por *screenshot*.

- Associar cada defeito a um *pin* na *screenshot*. Sendo o *pin* representado por uma imagem de um *pin* em formato .png, e contendo um número identificador. Número esse que corresponde ao defeito na lista.

- Deveria ser disponibilizado uma janela de registo ao clicarmos numa das linhas da lista de defeitos. Sendo possível assim editar, alterar ou remover.

- A ferramenta de identificação de defeitos (a ferramenta que identifica rapidamente defeitos sem ter que registar os campos no momento) não está implementada. Sendo ainda apenas possível identificar defeitos através da ferramenta "Identificar e Registar Defeitos".

- Ainda não está implementada a representação na *screenshot* dos defeitos sob a forma de imagens de *pins*. Era desejável que estes estivessem representados com um numero de defeito e cor dependendo do grau de severidade (*nominal* – verde, *minor* – amarelo, *major* – laranja, *critical* – vermelho). Caso este não tivesse um grau de severidade avaliado, a cor do *pin* seria cinzenta. A solução provavelmente passará pelo uso de um *Canvas*, onde estaria sobreposto á *screenshot* e lá poderíamos colocar os *pins*.

- A função de carregar uma *screenshot* num projecto chama o browser do sistema do utilizador, dando a escolher um tipo de imagem nos formatos (.jpg, .jpeg, .gif, .png). Mas ainda só é possível carregar imagens previamente introduzidas no directório "*images"* do nosso projecto Interface Killer. Seria desejável que fosse feito o *upload* de imagens do sistema do utilizador para o directório *"images"* do IK.

- Seria desejável que o *Interface Killer* gravasse os defeitos e que os associasse a determinado projecto e *screenshot*.