
Documento de Requisitos do Produto

Especificação de Casos de Uso

Ferramenta para Linha de Produtos de Jogos Móveis - FLiP

Cliente: FINEP – Financiadora de Estudos e Projetos-Ministério da Ciência e Tecnologia



Responsável: Vilmar Santos



Histórico de Alterações

Data	Versão	Descrição	Autor
08/11/2006	01.00-D01	Versão inicial do documento.	Davi Pires
10/11/2006	01.00-D02	Alterado o passo 3 do UC001 e do UC003. Alterados os passos 5 e 6 do UC002 e do UC004. Para deixar o documento mais padronizado.	Vilmar Santos
13/11/2006	01.00-D03	Pequenas correções no documento.	Jorge Leal
14/11/2006	01.00-D04	Rework após revisão: \finep_flip\processo\inspecoes\200611\FLIP-UCS-REV-001.xls.	Davi Pires
20/11/2006	01.00-D05	Correção ortográfica.	Jorge Leal
06/12/2006	01.00	Documento aprovado. Pequenas alterações sugeridas nos emails de aprovação.	Jorge Leal
22/02/2007	01.01	Adição UC005, UC006, UC007 e UC008. Atualização dos diagramas de caso de uso.	Vilmar Santos
26/02/2007	01.02	Adição de UC009. Atualização dos diagramas de caso de uso, alteração do UC001 para múltiplas constantes e atributos.	Vilmar Santos
14/06/2007	01.03	Criação de um UC genérico para extração e remoção de todos os outros UC de extração. Renomear os outros UC.	Vilmar Santos
09/07/2007	01.04	Adição do [UC005] e alterações no [UC001].	Vilmar Santos
21/08/2007	02.00	Correções da revisão de inspeção.	Vilmar Santos
26/10/2007	02.01	Adição UC010, UC011, UC012, UC012, UC013, UC014, UC015, UC016, UC017, UC018, UC019	Andréa Menezes
31/10/2007	02.02	Correções da revisão de inspeção	Andréa Menezes
27/11/2007	02.03	Atualização dos requisitos do UC001, alterações da revisão da equipe de teste.	Vilmar Santos
28/01/2008	02.04	Adição do UC020 e UC021, atualização do UC001, UC014 e UC015.	Vilmar Santos
29/01/2008	02.05	Correções após revisão de inspeção	Andréa Menezes
11/02/2008	02.06	Correções após auditoria da SQA.	Vilmar Santos
11/02/2008	03.00	Correções após aprovação.	Vilmar Santos
27/02/2008	03.01	Remoção do RF038 do UC001	Vilmar Santos

Lista de Aprovadores

Nome	Cargo
Paulo Borba	Professor Colaborador UFPE
Sérgio Soares	Professor Colaborador UPE
Isabel Wanderley	Gerente de Projeto

Conteúdo

1. Introdução.....	5
1.1 Propósito.....	5
1.2 Público Alvo.....	5
1.3 Convenções, termos e abreviações.....	5
2. Atores.....	5
3. Diagrama de Casos de Uso.....	5
4. Casos de Uso.....	7
4.1 Casos de uso da ferramenta FLiPEx.....	7
[UC001] Extrair variação de código.....	7
[UC005] Extrair variações dentro de uma condição if.....	8
[UC011] Suportar clonagem de extrações.....	8
[UC012] Gerar pacote de instalação.....	9
[UC013] Criar projeto com a natureza FLiP.....	9
[UC014] Suportar extração de código pré-processado.....	10
[UC015] Suportar extração de tag pré-processada.....	11
[UC016] Criar ajuda do sistema para o usuário.....	12
[UC018] Criar ajuda do sistema para o desenvolvedor.....	12
[UC019] Suportar várias extrações com a mesma configuração.....	12
[UC020] Suportar técnicas de extração.....	13
[UC021] Extrair variação para código pré-processado.....	13
4.2 Casos de uso da ferramenta FLiPG.....	14
[UC002] Criar feature utilizando o pure::variants.....	14
[UC003] Selecionar instância da linha de produtos utilizando o pure::variants.....	15
[UC004] Associar artefato a uma expressão de features utilizando o pure::variants.....	15
[UC006] Remover feature utilizando o pure::variants.....	16
[UC007] Atualizar feature model utilizando o pure::variants.....	16
[UC008] Atualizar Configuration Knowledge.....	17
[UC009] Criar restrições sobre o Feature Model.....	17
[UC010] Gerar especificações de builds.....	17
[UC017] Suportar expressão de features.....	18
5. Referências.....	18

1. Introdução

1.1 Propósito

Este documento especifica os casos de uso da ferramenta FLIP, descrevendo os fluxos de eventos, entradas e saídas de cada caso de uso a ser implementado.

1.2 Público Alvo

Esse documento se destina principalmente aos líderes técnicos e desenvolvedores do projeto.

1.3 Convenções, termos e abreviações

Esta seção explica o conceito de alguns termos importantes que serão mencionados no decorrer deste documento. Estes termos são descritos na tabela a seguir, estando apresentados por ordem alfabética.

Outros termos e abreviações, que sejam padrões do processo, estão descritos no Glossário do ProSCes (acessível a partir da página principal do ProSCes).

Termo	Descrição
Atores	Alguém ou alguma coisa (ex: máquina, outro sistema), externa à aplicação, que interage de alguma forma com o sistema.
Casos de Uso	Uma seqüência de ações realizadas pelo sistema que resulta num valor observável para um ator específico.
Requisitos funcionais	Requisitos técnicos do software que compõe o sistema, que descrevem ações que o sistema deve estar apto a executar, ou seja, o que o sistema deve fazer.
Requisitos não funcionais	Requisitos técnicos do software que compõe o sistema, que descrevem atributos que o sistema deve possuir ou restrições sob as quais ele deve operar.

2. Atores

Ator	Descrição
Usuário	Desenvolvedor de jogos móveis no contexto de uma linha de produtos.

3. Diagrama de Casos de Uso

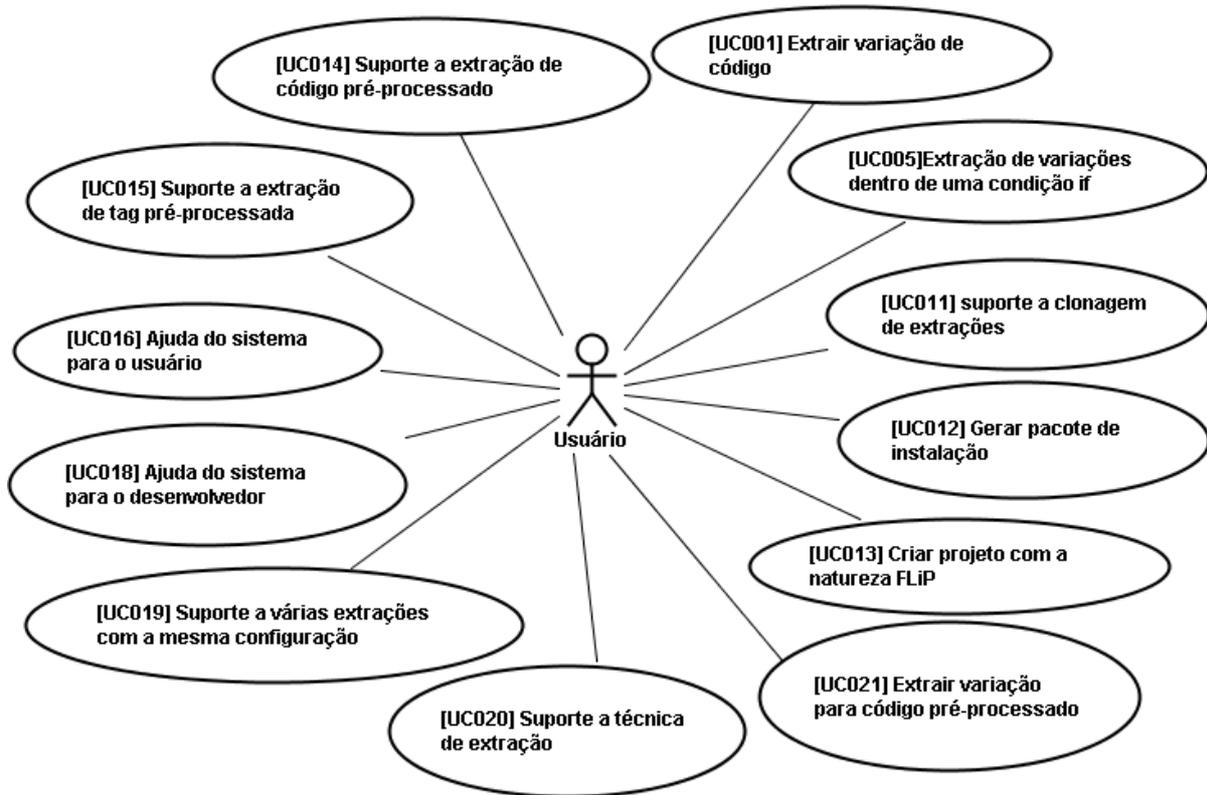


Figura 1: Diagrama de casos de uso da ferramenta FLiPex.

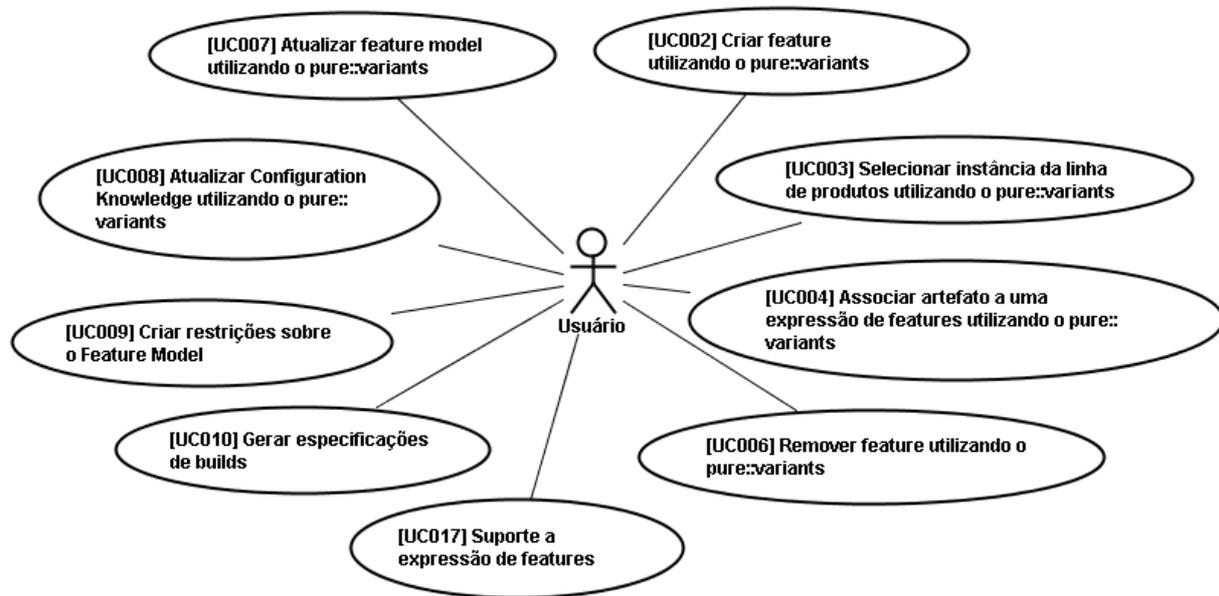


Figura 2: Diagrama de casos de uso da ferramenta FLiPg.

4. Casos de Uso

4.1 Casos de uso da ferramenta FLiPEX

Esta seção contempla os casos de uso referentes à ferramenta de extração FLiPEX.

[UC001] Extrair variação de código

Pré-condições: Nenhuma

Pós-condições: A variação deve ter sido movida para o aspecto escolhido pelo usuário e o *configuration knowledge* deve ser atualizado.

Requisitos atendidos: [RF001] [RF002] [RF003] [RF004] [RF005] [RF007] [RF010] [RF017] [RF019] [RF020] [RF021] [RF022] [RF029] [RF032] [RF033] [RF039] [RF040] [RF041]

Casos de Uso relacionados: Casos de uso da ferramenta FLiPEX.

Fluxo de eventos principal

1. Usuário seleciona o código a ser extraído e requisita a operação de extração ao sistema.
2. O sistema valida o código a ser extraído, e retorna uma lista de extrações aplicáveis [UC020] [FE01].
3. Usuário seleciona a opção de extração desejada [UC019] e clica o botão “*next*”.
4. O sistema apresenta ao usuário uma tela para a criação de uma *feature*.
5. Usuário cria nova *feature* a ser associada ao código extraído e clica o botão “*next*” [FA01] [UC017] [UC011] [UC009] [FA03].
6. O sistema apresenta ao usuário uma tela para a criação de um aspecto.
7. Usuário cria um novo aspecto para o qual o código selecionado é extraído e clica para finalizar a extração [FA02].
8. O sistema extrai a variação para o aspecto especificado [UC008] [FE02].

Fluxos Alternativos

[FA01] Usuário deseja utilizar uma *feature* já existente

1. O sistema exibe uma árvore com as *features* já existentes.
2. Usuário seleciona a *feature* com a qual o código selecionado vai ser associado.
3. O fluxo retorna para o passo 6 do fluxo principal.

[FA02] Usuário deseja utilizar um aspecto já existente

1. O sistema exibe a lista dos aspectos já existentes.
2. Usuário seleciona o aspecto para o qual o código selecionado vai ser extraído.
3. O fluxo retorna para o passo 8 do fluxo principal.

[FA03] Usuário deseja atualizar uma *feature* existente

1. O sistema exibe a lista das *features existentes*.
2. Usuário seleciona a *feature que deseja atualizar* e pressiona o botão *edit*.
3. Usuário muda o tipo da feature e pressiona o botão *ok*.

Fluxos de Exceção

[FE01] Usuário selecionou um código inválido para extração

1. O sistema informa ao usuário que ele não selecionou um código inválido para extração.
2. O fluxo retorna para o passo 1 do fluxo principal.

[FE02] Extração não foi realizada com sucesso

1. O sistema informa ao usuário que a extração não foi realizada com sucesso.
2. O fluxo retorna para o passo 1 do fluxo principal.

[UC005] Extrair variações dentro de uma condição *if*

Pré-condições: Nenhuma

Pós-condições: A variação deve ter sido movida para o aspecto escolhido pelo usuário, e as especificações de build e *configuration knowledge* devem ser atualizadas.

Requisitos atendidos: [RF010]

Casos de Uso relacionados: [UC001]

Fluxo de eventos principal

1. Usuário seleciona o código a ser extraído e realiza um *refactoring extract method* no eclipse.
2. O arquivo é atualizado.
3. Usuário seleciona método gerado pelo *refactoring* do passo 1 e executa o UC001 a partir do passo 2.

[UC011] Suportar clonagem de extrações

Pré-condições: Usuário ter selecionado um trecho de código e um extrator.

Pós-condições: A lista de *features* selecionadas deve conter a *feature* inicialmente selecionada e as alternativas que o usuário escolher.

Requisitos atendidos: [RF023]

Casos de Uso relacionados: [UC001]

Fluxo de eventos principal

1. Usuário executa fluxo de eventos principal do [UC001] do passo 1 ao 4.
2. Usuário seleciona o *checkbox clone*.
3. Sistema atualiza a página de features
4. Usuário seleciona a *feature* desejada.

5. O sistema mostra uma lista contendo a *feature* selecionada e suas alternativas é mostrada na tela.
6. O Usuário pode desmarcar o *checkbox* das *features* que ele não deseja utilizar e retorna ao passo 6 do [UC001].

Fluxos Alternativos

[FA01] Usuário deseja clonar após selecionar a *feature*.

1. Usuário seleciona a *feature* desejada.
2. O sistema atualiza a página de *features*.
3. Usuário seleciona o *checkbox clone*.
4. O fluxo retorna para o passo 4 do fluxo principal.

[UC012] Gerar pacote de instalação

Pré-condições: Nenhuma

Pós-condições: O FLiP está corretamente instalado na plataforma *eclipse* do usuário

Requisitos atendidos: [RF024]

Casos de Uso relacionados: NA.

Fluxo de eventos principal

1. Usuário seleciona a opção *Help->Software updates->Find and Install* no menu do *eclipse*.
2. O sistema mostra uma tela para inicializar a instalação.
3. Usuário seleciona a opção *Search for new features to install* e pressiona o botão *next*.
4. O sistema mostra a tela de procura por novas *features*.
5. Usuário pressiona o botão *New Remote Site...* e digita FLiP no campo *name* e a URL onde se encontra o projeto e pressiona o botão *ok* e posteriormente o botão *finish*.
6. O sistema exibe uma tela com os *plugins* encontrados.
7. Usuário seleciona o que deseja instalar e pressiona o botão *finish*.
8. Sistema instala o FLiP corretamente.

[UC013] Criar projeto com a natureza FLiP

Pré-condições: Nenhuma

Pós-condições: Um projeto com a natureza FLiP deverá estar criado.

Requisitos atendidos: [RF025]

Casos de Uso relacionados: NA.

Fluxo de eventos principal

1. Usuário seleciona a opção *File->New->Other* no menu do *eclipse* [FA01]
2. O sistema apresenta uma tela para a escolha do tipo de projeto

3. Usuário seleciona a opção FLiP Project e seleciona o botão *next*
4. O sistema mostra a tela do wizard para criação do projeto FLiP
5. Usuário digita o nome do projeto e pressiona o botão *finish*.
6. Sistema cria o projeto corretamente

Fluxos Alternativos

[FA01] Usuário deseja criar um projeto a partir do *workspace*.

1. Usuário clica com o botão direito do *mouse* no *workspace*.
2. Usuário seleciona a opção *New->Other*.
3. Fluxo retorna ao passo 2 do fluxo principal.

[UC014] Suportar extração de código pré-processado

Pré-condições: Nenhuma.

Pós-condições: A variação deve ter sido movida para o aspecto escolhido pelo usuário e o *configuration knowledge* deve ser atualizado.

Requisitos atendidos: [RF026]

Casos de Uso relacionados: [UC001]

Fluxo de eventos principal

1. Usuário seleciona o código pré-processado a ser extraído e requisita a operação de extração de código ao sistema.
2. O sistema valida o código a ser extraído, e retorna uma lista de *tags* identificadas e para cada *tag* uma lista de extrações aplicáveis. [FE01]
3. Usuário seleciona para cada *tag* uma opção de extração desejada e pressiona o botão *next*.
4. O sistema mostra uma tela para associar as *tags* as *features*.
5. Usuário clica na tabela de *tags* para selecionar *features* que serão associadas às *tags* e seleciona uma *feature* existente que será exibida pelo sistema em uma árvore [FA01] [FA02] e pressiona o botão *ok*.
6. O sistema atualiza a página de associação entre *tag* e *feature*
7. Usuário repete os passos 5 a 6 até que todas as *tags* possuam uma *feature* associada e pressiona o botão *next*.
8. O sistema mostra uma tela para associar as *features* escolhidas com aspectos.
9. Usuário clica na tabela de *features* para selecionar os aspectos e seleciona um aspecto existente da lista exibida pelo sistema [FA03] e pressiona o botão *ok*.
10. O sistema atualiza a página de associação entre *feature* e aspecto.
11. Usuário repete os passos 8 a 9 até que todas as *features* possuam um aspecto associado e pressiona o botão *finish*.
12. O sistema extrai a variação para o aspecto especificado [FE02].

Fluxos Alternativos

[FA01] Usuário deseja criar uma nova *feature*

1. Usuário pressiona o botão *new*.
2. Sistema exibe tela de criação de nova *feature*.
3. Usuário digita o nome da *feature*, seleciona o tipo de parâmetro e a *feature* a qual estará associada e pressiona o botão *ok*.
4. O fluxo retorna para o passo 5 do fluxo principal.

[FA02] Usuário deseja escrever uma expressão de *features*

1. O usuário escreve uma expressão de *features*.
2. O fluxo retorna para o passo 5 do fluxo principal.

[FA03] Usuário deseja criar um novo aspecto

1. Usuário pressiona o botão *new aspect*.
2. Usuário digita nome do novo aspecto.
3. Usuário pressiona o botão *finish*.
4. O fluxo retorna para o passo 8 do fluxo principal.

Fluxos de Exceção

[FE01] Usuário selecionou um código inválido para extração

3. O sistema informa ao usuário que ele não selecionou um código inválido para extração.
4. O fluxo retorna para o passo 1 do fluxo principal.

[FE02] Extração não foi realizada com sucesso

3. O sistema informa ao usuário que a extração não foi realizada com sucesso.
4. O fluxo retorna para o passo 1 do fluxo principal.

[UC015] Suportar extração de *tag* pré-processada

Pré-condições: Nenhuma.

Pós-condições: A variação deve ter sido movida para o aspecto escolhido pelo usuário e o *configuration knowledge* deve ser atualizado.

Requisitos atendidos: [RF028]

Casos de Uso relacionados: [UC014]

Fluxo de eventos principal

1. Usuário seleciona o projeto ou a classe no qual deseja realizar a operação e requisita a operação de extração de *tag* pré-processada ao sistema.

2. O sistema seleciona automaticamente um trecho de código pré-processado.
3. Usuário executa [UC014] a partir do passo 3.
4. Os passos 3 e 4 são repetidos até que a seleção automática tenha chegado ao fim ou o usuário cancele toda a operação.

[UC016] Criar ajuda do sistema para o usuário

Pré-condições: Nenhuma.

Pós-condições: O sistema de ajuda para usuário da ferramenta deve ter sido exibido.

Requisitos atendidos: [RF035]

Casos de Uso relacionados: NA.

Fluxo de eventos principal

1. Usuário pressiona o botão *help* no *wizard*.
2. O sistema exibe a tela de ajuda.

[UC018] Criar ajuda do sistema para o desenvolvedor

Pré-condições: Nenhuma.

Pós-condições: O sistema de ajuda para o desenvolvedor da ferramenta deve ter sido exibido.

Requisitos atendidos: [RF036]

Casos de Uso relacionados: NA.

Fluxo de eventos principal

1. Usuário seleciona a opção *Help* -> *Help contents* no menu do *eclipse*.
2. O sistema exibirá a tela de ajuda do eclipse.
3. O Usuário deverá selecionar a opção FLiP no *menu*.
4. O sistema exibe a tela de ajuda do FLiP.

[UC019] Suportar várias extrações com a mesma configuração

Pré-condições: Usuário ter executado uma extração anteriormente.

Pós-condições: Uma extração com a configuração selecionada deve ser realizada.

Requisitos atendidos: [RF016]

Casos de Uso relacionados: [UC001]

Fluxo de eventos principal

1. Usuário executa Fluxo de eventos principal do [UC001] do passo 1 ao 4.
2. Usuário seleciona a aba de extrações anteriores.

3. O sistema exibe a tela de extrações anteriores
4. Usuário seleciona a extração desejada e pressiona o botão *finish*.
5. Sistema extrai o código para o aspecto destino.

[UC020] Suportar técnicas de extração

Pré-condições: Nenhuma.

Pós-condições: Uma extração com deve ter sido selecionada.

Requisitos atendidos: [RF043]

Casos de Uso relacionados: [UC001] [UC021]

Fluxo de eventos principal

1. Usuário executa Fluxo de eventos principal do [UC001] ou [UC021] do passo 1 ao 3.
2. Usuário seleciona a técnica que deseja aplicar.
3. O sistema lista para o usuário as opções de extração para a técnica escolhida.
4. Usuário retorna a execução do [UC001] ou [UC021] a partir do passo 4.

[UC021] Extrair variação para código pré-processado

Pré-condições: Nenhuma

Pós-condições: A variação deve ter sido transformada e o *configuration knowledge* deve ser atualizado.

Requisitos atendidos: [RF042] [RF030] [RF031] [RF034]

Casos de Uso relacionados: [UC001] e [UC020].

Fluxo de eventos principal

1. Usuário seleciona o código a ser extraído e requisita a operação de extração ao sistema.
2. O sistema valida o código a ser extraído, e retorna uma lista de extrações aplicáveis [UC020] [FE01].
3. Usuário seleciona a opção de extração para código pré-processado [UC019] e clica o botão “*next*”.
4. O sistema mostra ao usuário a tela para criação de features.
5. Usuário cria nova *feature* a ser associada ao código extraído [FA01] [UC017] [UC011] [UC009] [FA02].
6. Sistema apresenta o texto pré-processado para que o usuário possa editá-lo.
7. Usuário realiza as alterações desejadas no código pré-processado e clica em *finish*.
8. O sistema extrai a variação para a classe original do código [UC008] [FE02].

Fluxos Alternativos

[FA01] Usuário deseja utilizar uma *feature* já existente

1. O sistema exibe uma árvore com as *features* já existentes.
2. Usuário seleciona a *feature* com a qual o código selecionado vai ser associado.
1. O fluxo retorna para o passo 6 do fluxo principal.

[FA02] Usuário deseja atualizar uma *feature* existente

1. O sistema exibe a lista das *features existentes*.
2. Usuário seleciona a *feature que deseja atualizar* e pressiona o botão *edit*.
3. Usuário muda o tipo da *feature* e pressiona o botão *ok*.

Fluxos de Exceção

[FE01] Usuário selecionou um código inválido para extração

1. O sistema informa ao usuário que ele não selecionou um código inválido para extração.
2. O fluxo retorna para o passo 1 do fluxo principal.

[FE02] Extração não foi realizada com sucesso

1. O sistema informa ao usuário que a extração não foi realizada com sucesso.
2. O fluxo retorna para o passo 1 do fluxo principal.

4.2 Casos de uso da ferramenta FLiPG

Esta seção contempla os casos de uso referentes à ferramenta de gerenciamento FLiPG.

[UC002] Criar *feature* utilizando o `pure::variants`

Pré-condições: Projeto de variação criado pelo `pure::variants5`

Pós-condições: *Feature* foi criada e *feature model* atualizado com sucesso.

Requisitos atendidos: [RF006]

Casos de Uso relacionados: NA.

Fluxo de eventos principal

1. Usuário clica com o botão direito a *feature* a partir da qual a nova *feature* será criada.
2. O sistema exibe um menu para o usuário.
3. Seleciona a opção *new* no menu e clica a opção *Generic Feature*
4. O sistema mostra uma tela para criação da *feature*.
5. Usuário preenche os campos [FA01] e clica o botão *finish*.
 1. *Unique name**
 2. *Visible name*
 3. *Class/Type*

4. *Variation type*
5. *Description*
6. Sistema cria a feature atualizando o *feature model*.

* Campo de preenchimento obrigatório

Fluxos Alternativos

[FA01] Usuário deseja adicionar relações, atributos, restrições e limitações a feature

1. O Usuário clica o botão *Next* no *wizard* de criação de *feature*
2. Cada tela mostrará as opções para realizar cada uma das ações listadas
 1. Criar relação
 2. Criar atributos
 3. Criar restrições
 4. Criar limitações
3. Usuário preenche os campos que desejar
4. O fluxo retorna para o passo 4 do fluxo principal.

[UC003] Selecionar instância da linha de produtos utilizando o `pure::variants`

Pré-condições: *Feature model* da linha de produtos criado.

Pós-condições: Instância da linha de produtos criada com sucesso.

Requisitos atendidos: [RF008]

Casos de Uso relacionados: NA.

Fluxo de eventos principal

1. Usuário cria um arquivo *.vdm* para edição no ambiente eclipse.
2. Usuário seleciona features para geração da instância
3. O usuário clica o botão *check model* para validar a seleção [FE01]
4. Usuário recebe uma mensagem do sistema indicando que a seleção é válida

Fluxos de Exceção

[FE01] Usuário realizou uma seleção inválida

1. O usuário recebe uma mensagem indicando que a seleção não foi válida
2. O fluxo retorna para o passo 2 do fluxo principal

[UC004] Associar artefato a uma expressão de features utilizando o `pure::variants`

Pré-condições: *Family model* da linha de produtos criado com pelo menos um componente (artefato).

Pós-condições: Associação do artefato com uma expressão de features criada com sucesso.

Requisitos atendidos: [RF009]

Casos de Uso relacionados: NA.

Fluxo de eventos principal

1. Usuário abre o arquivo *.cfm* da linha de produtos para edição no ambiente eclipse e abre as propriedades de um componente
2. Sistema mostra tela para adicionar restrição.
3. Usuário adiciona uma restrição na aba *Restrictions* e entra no modo de edição da restrição
4. Sistema mostra tela para a edição da restrição.
5. Usuário cria a expressão da restrição utilizando a interface provida pelo *pure::variants* e clica o botão *ok* para finalizar a tela de criação da expressão
6. O sistema mostra a tela de edição de componentes.
7. Usuário clica o botão *ok* para finalizar a edição do componente
8. Sistema atualiza o componente corretamente.

[UC006] Remover *feature* utilizando o *pure::variants*

Pré-condições: *Feature* existir no *feature model*.

Pós-condições: *Feature* foi removida e *feature model* atualizado com sucesso.

Requisitos atendidos: [RF018]

Casos de Uso relacionados: NA.

Fluxo de eventos principal

1. Usuário clica com o botão direito na *feature* que deseja remover.
2. Sistema apresenta um menu para o usuário.
3. Usuário seleciona a opção *delete* no menu e clica em *Yes* na caixa de diálogo de confirmação da operação.
4. Sistema remove a *feature* corretamente.

[UC007] Atualizar *feature model* utilizando o *pure::variants*

Pré-condições: *Feature model* existir.

Pós-condições: *Feature model* atualizado com sucesso.

Requisitos atendidos: [RF011]

Casos de Uso relacionados: NA.

Fluxo de eventos principal

1. Usuário seleciona uma *feature* que deseja atualizar e seleciona a opção para editar a *feature*.
2. Sistema mostra tela com os campos que podem ser atualizados.
3. Usuário muda os campos que desejar e confirma as atualizações.
4. Sistema atualiza o *feature model*

[UC008] Atualizar *Configuration Knowledge*

Pré-condições: *Configuration Knowledge* existir.

Pós-condições: *Configuration Knowledge* atualizado com sucesso.

Requisitos atendidos: [RF012]

Casos de Uso relacionados: NA.

Fluxo de eventos principal

1. Usuário executa fluxo de eventos principal do [UC001] do passo 1 ao 8.
2. *Configuration Knowledge* é atualizado pelo sistema utilizando as seguintes informações:
 1. Nome do aspecto destino
 2. Nome da(s) *feature(s)* utilizadas.
3. *Configuration Knowledge* é atualizado pelo sistema criando o componente que guarda o aspecto*.
4. *Configuration Knowledge* é atualizado pelo sistema criando uma relação entre a *feature* e o componente*.

* As atualizações só terão efeito se o usuário finalizar a extração.

[UC009] Criar restrições sobre o *Feature Model*

Pré-condições: *Feature model* existir.

Pós-condições: Restrição criada com sucesso.

Requisitos atendidos: [RF013]

Casos de Uso relacionados: NA.

Fluxo de eventos principal

1. Usuário seleciona uma *feature* que deseja atualizar e seleciona a opção para editar a *feature*.
2. O sistema mostra uma tela com informações sobre a *feature*
3. Usuário seleciona a opção de editar as restrições.
4. O sistema mostra uma tela para a edição de restrições.
5. Usuário cria as restrições que desejar e confirma as atualizações*.
6. Sistema atualiza as restrições da *feature*.

* As atualizações só terão efeito se o usuário finalizar a extração.

[UC010] Gerar especificações de *builds*

Pré-condições: Existir pelo menos um “.vdm” com uma seleção válida.

Pós-condições: Especificação criada com sucesso.

Requisitos atendidos: [RF014] [RF015]

Casos de Uso relacionados: NA.

Fluxo de eventos principal

1. Usuário seleciona um projeto de natureza FLiP . O projeto selecionado deve ter configurado obrigatoriamente:
 1. Arquivo de *Feature model*
 2. Arquivo de *Configuration Knowledge*
 3. Diretório de *output*
 4. Diretório do sistema de *build*.
2. O sistema mostra o projeto selecionado.
3. Usuário seleciona a opção de gerar especificações de build.
4. Sistema indica ao usuário o fim da geração.

[UC017] Suportar expressão de *features*

Pré-condições: Nenhuma

Pós-condições: Uma expressão de *features* válida deve ter sido criada.

Requisitos atendidos: [RF027]

Casos de Uso relacionados: [UC01]

Fluxo de eventos principal

1. Usuário seleciona a opção *feature expression* e digita a expressão de *features* e pressiona o botão *validate*.
2. O sistema valida a expressão de *features*. [FE01]

Fluxos de Exceção

[FE01] Usuário digitou uma expressão inválida

1. O sistema exibe uma mensagem de erro.
2. O fluxo retorna para o passo 2 do fluxo principal.

5. Referências

- [1] Documento de Requisitos do Cliente; Versão 03.03;
finep_flip\produtos\flip\documentos\requisitos\FLIP-REQ.doc;
- [2] <http://www.pure-systems.com/english.3.0.html>