

# ANEXOS

## **A1. BREVE RESUMO DA NOTAÇÃO BPMN**

Neste anexo são apresentadas as entidades utilizadas neste trabalho e as comumente utilizadas na modelagem de processos de negócio da notação BPMN (Business Process and Notation). O objetivo é apenas orientar o autor quanto à modelagem realizada no trabalho e não realizar uma revisão de modelagem de processo usando BPMN. Para isso o leitor pode consultar as referências apresentadas no texto a respeito do assunto. Os quadros desenvolvidos neste anexo são baseados na instrução normativa desenvolvida pelo Object Management Group (OMG) (OBJECT MANAGEMENT GROUP, 2009) e no livro de Stephen A. White, Derek Miers (WHITE e MIERS, 2008).

## Atividades

**Atividades** representam o trabalho realizado por uma organização: Elas representam um passo no processo. As **Atividades** podem ser compostas ou não.



### Tarefa

Uma tarefa é uma atividade simples que é usada para modelar o trabalho que é realizado em um processo e não é definida em mais detalhes. BPMN possui diferentes tipos de tarefas:



### Subprocesso

Trata-se de uma atividade composta cujos detalhes são definidos como um conjunto de outras atividades.



### Subprocesso encapsulado

Depende completamente do processo pai. Eles não podem conter *pools* ou *lanes*.



### Processo reutilizável

É um processo de negócio definido em outro diagrama de processo, que não depende do processo pai.

## Gateways

**Gateways** são elementos usados para controlar a divergência e convergência dos fluxos de trabalho.



### Gateway exclusivo baseado em dados – GEBD

**Divergência:** Uma decisão exclusiva tem duas ou mais saídas de fluxo de trabalho, mas somente uma delas será seguida e a decisão depende da avaliação do processo de negócio.

**Convergência:** é usado para unir fluxos de trabalho alternativos.



### Gateway exclusivo baseado em eventos - GEBE

É usado como um elemento de divergência. Esse gateway representa um ponto no processo onde somente um, de muitos fluxos pode continuar, mas baseado em um evento, não em condição baseada em dados como o GEBD.



### Gateway Paralelo

**Divergência:** é usado para criar fluxos paralelos.

**Convergência:** é usado para sincronizar múltiplos fluxos paralelos em um. O fluxo continua somente após a chegada de todos os fluxos de entrada ao *gateway*.



### Gateway Inclusivo

**Divergência:** Indica que somente uma ou mais caminhos no fluxo do processo podem ser ativados de muitos disponíveis, e a decisão é baseada em dados do processo.

**Convergência:** Indica que muitos caminhos de saída de um gateway inclusivo, usados como um elemento de divergência, podem ser sincronizados em apenas um.



### Gateway Complexo

**Divergência:** é usado para controlar pontos no processo onde existem decisões complexas que não são fáceis de gerenciar com outros tipos de *gateways*.

**Convergência:** Quando um *gateway* é usado como um aglutinante de fluxos então existirá uma expressão que determinará qual dos fluxos de entrada será necessário para o processo continuar.

## Swinlanes



Um Pool é container para um único processo. O nome do pool pode ser considerado o nome do processo. Existe pelo menos um pool em processo modelado com BPMN.



Uma *lane* é uma subdivisão de um pool e representa um papel ou área organizacional.

## Artefatos

Permite ou fornece informação adicional sobre o processo



### Anotação

Fornece informação adicional sobre o processo ao leitor dos modelos.



### Grupo

É um mecanismo visual que permite o agrupamento de atividades para o propósito de documentação ou análise dos processos.



### Objeto dado

Fornece informações sobre as entradas e saídas de uma atividade, isto é, documentos, dados e outros objetos usados e modificados durante o processo. Objetos dado não tem qualquer efeito sobre o fluxo do processo ou fluxo de informações inerentes ao processo.

## Objetos de conexão



### Sequence flow

É usado para mostrar a ordem que as atividades serão realizadas em um processo. Is used to show the order that activities will be performed in a Process. Ele é usado para representar a sequencia do fluxo objetos, onde estão presentes atividades, gateways e eventos.



### Message flow

A entidade *Message flow* é usada para mostrar o fluxo de mensagens entre duas outras entidades ou processos.

A entidade *Message flow* representa mensagem (informação), não o controle do fluxo de informação. Nem todas as entidades *Message flows* são cumpridas para cada instância do processo e nem existe uma ordem específica para as mensagens.



### Associação

Uma associação é usada para associar informação e artefatos com objetos de fluxo.



### Evento Start

Indica a ocorrência ou início de um processo. Não existe qualquer *sequence flow* chegando a essa entidade.



### Evento intermediário

Eventos intermediários indicam que alguma coisa ocorreu ou pode ocorrer durante o curso do processo, entre o seu início e fim. Ele pode ser usado no fluxo ou anexado a borda de uma atividade.



### Evento End

Evento End indica onde um processo irá acabar. Um processo pode ter mais do que um evento End. Esses eventos não possuem qualquer *sequence flow* deixando-o.



#### None

Não especifica qualquer comportamento particular.



#### Mensagem

O processo inicia-se quando uma mensagem é recebida de outro participante do processo.



#### Timer

Indica que um processo inicia em certo tempo ou em uma data especificada.



#### None

Indica que alguma coisa ocorreu ou pode ocorrer no processo. Ele pode ser usado, somente, no fluxo do processo.



#### Mensagem

Indica que uma mensagem pode ser enviada ou recebida. Se o evento é de recepção, ele indica que o processo deve aguardar até receber uma mensagem. Esse tipo de evento pode ser usado no fluxo ou anexado a borda de uma atividade para indicar um fluxo alternativo.



#### Timer

Indica um tempo de espera no processo. Esse tipo de evento pode ser usado no fluxo do processo ou anexado a borda de uma atividade para indicar um fluxo alternativo quando acabar o tempo de



#### None

Indica que uma rota/caminho do processo chegou ao fim. Um processo acaba somente quando todas as rotas/caminhos chegam a um fim.



#### Mensagem

Indica que uma mensagem é enviada a outro processo quando o processo chega ao fim.

## Eventos

### Evento início

#### Condicional

O processo inicia quando uma condição de negócio for verdadeira.

#### Sinal

Um processo inicia quando um sinal de vindo de outro processo é recebido. Note que o sinal não é uma mensagem; mensagens tem claramente definido quem envia e quem recebe a mensagem.

#### Múltiplos

Indica que existem muitas formas de iniciar o processo. Mas somente uma delas é suficiente.

### Evento intermediário

#### Condicional

É usado quando o fluxo precisa aguardar por uma condição de negócio seja atingida. Ele pode ser usado no fluxo indicando que deve-se aguardar até uma condição de negócio seja atingida ou anexado a borda de uma atividade indicando um fluxo alternativo quando uma condição específica é atingida.

#### Sinal

É usado para enviar e receber sinais. Se ele colocado no fluxo de um processo ele pode ser enviar ou receber um sinal. Se ele é colocado anexado a borda de uma atividade, ele pode somente receber sinais e indica uma exceção ao fluxo normal que é ativado quando o sinal é recebido.

#### Múltiplos

Esse evento significa que existem múltiplos gatilhos relacionados ao evento. Todos os gatilhos devem ocorrer para que o evento aconteça.

### Evento fim

#### Sinal

Indica que um sinal é gerado quando o processo termina.

#### Múltiplos

Indica que muitos resultados podem ser alcançados ao final do processo. Todos os resultados devem ser alcançados para que o processo termine.



## Evento início



## Eventos

### Evento intermediário



#### Compensação

O evento de compensação permite manipular compensações. Quando usado em um fluxo sequencial de processo, ele indica que uma compensação é necessária para o fluxo continuar. Quando usado nas bordas de uma atividade indica que essa atividade será compensada quando o evento acontecer.



#### Link

É usado para conectar duas seções do processo.



### Evento fim



#### Compensação

Indica que o processo acabou e que uma compensação é necessária.



#### Cancelamento

É usado somente em transações de sub-processos e indica que a transação deve ser cancelada.



#### Erro

Indica que um erro conhecido é gerado quando o processo termina.



#### Fim imediato

Esse evento termina o processo imediatamente. Quando uma de fluxo de processo chega a seu fim, indicando que o processo terminou completamente.

## **A2. BREVE RESUMO DA NOTAÇÃO SysML**

Neste anexo são apresentadas as entidades utilizadas neste trabalho e as comumente utilizadas na modelagem de sistemas da linguagem SysML (*System Modeling Language*). O objetivo é apenas orientar o autor quanto à modelagem SysML realizada no trabalho e não realizar uma revisão completa da mesma. Para isso o leitor pode consultar as referências apresentadas no texto a respeito do assunto. Os quadros desenvolvidos neste anexo são baseados na instrução normativa desenvolvida pelo *Object Management Group* (OMG) (OBJECT MANAGEMENT GROUP, 2010) e no livro de Tim Weilkiens (WEILKIENS, 2006).

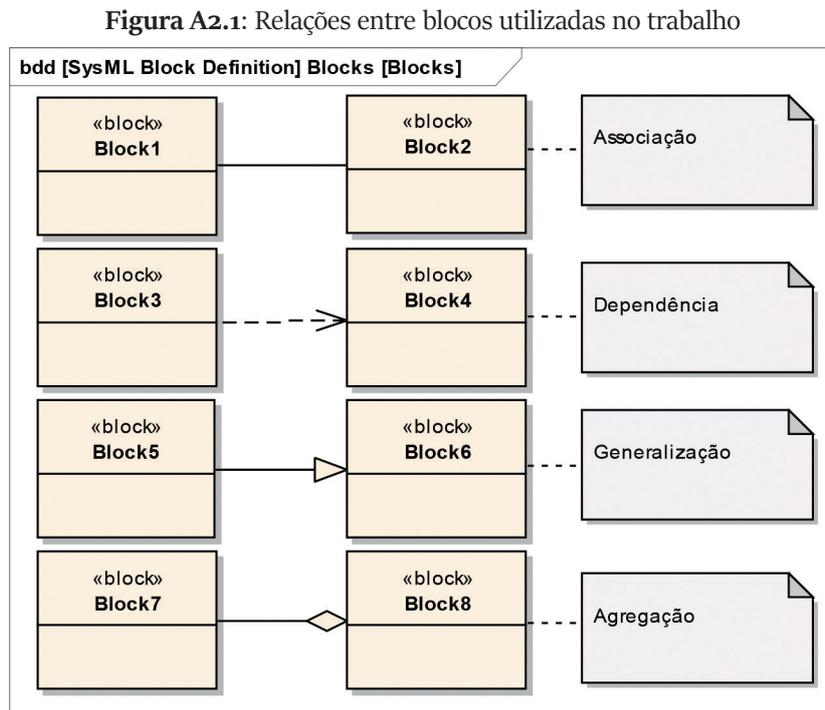
### **DIAGRAMA DE BLOCOS**

Os elementos centrais segundo o paradigma de modelagem orientada a objetos são classes e objetos. Estreitamente relacionados às classes estão os componentes na linguagem UML. Desde que os dois termos estão historicamente relacionados ao desenvolvimento de software, SysML não os usa dessa forma. Todos os conceitos e objetos estáticos são blocos em SysML (OBJECT MANAGEMENT GROUP, 2010). Eles têm a função de descrever a estrutura do sistema e identificar a organização de suas partes. Pode apresentar o fluxo de informações entre componentes do sistema e definições de interface por meio de portas. O Quadro A2.1 apresenta os principais elementos de um diagrama de blocos.

**Quadro A2.1:** Principais elementos de um diagrama de blocos (OBJECT MANAGEMENT GROUP, 2010).

<b>Nome do elemento</b>	<b>Definição</b>	<b>Origem</b>
Blocos (Blocks)	Correspondem a uma extensão do conceito de classe da UML.	SysML
Operações (Operations)	São elementos que definem o comportamento do bloco.	SysML
Propriedades (Properties):	São atributos do bloco	SysML
Restrições (Constraints)	Propriedades restritivas que o componente deve obedecer.	SysML
Atores (Actors):	Correspondem a usuários ou outros sistemas que interagem de forma ativa ou passiva com o sistema modelado.	UML
Relações (Relationship):	Relacionamentos entre os blocos de um modelo	UML
Associação (Association):	Indica uma relação entre tipos de instâncias	UML
Composição (Composition)	Relação entre dois blocos onde um bloco contém outro bloco	UML
Generalização (Generalization):	Relaciona um classificador mais específico com um classificador mais geral. Também é conhecida como herança.	
Dependência (Dependency)	Define uma relação em que o modelo de um elemento requer outro modelo de elemento para sua especificação ou implementação.	
Agregação (Aggregation)	Representa uma relação “has-a”, representando que um objeto, representando o “todo”, tem objetos que são suas partes. Agregação é um tipo especial de associação	
Porta (Ports):	Especificam serviços que o bloco oferece ao ambiente ou serviços que o bloco espera do ambiente	
Portas de fluxo (Flow ports)	Especificam entradas e saídas de itens que proporcionam fluxo entre blocos e ambiente. São pontos de interações através de dados, material ou energia que entra ou sai do próprio bloco.	

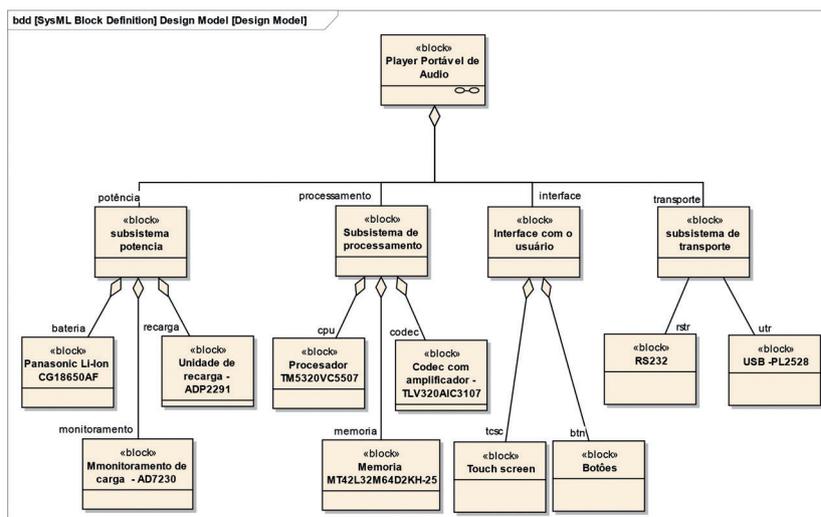
A Figura A2.1 mostra graficamente as relações usadas nesse trabalho entre blocos:



Um exemplo de diagrama de blocos é apresentado na Figura A2.2. Nele, um sistema portátil de áudio é representado por blocos que simulam os componentes do sistema (subsistemas). Cada subsistema é responsável por realizar uma tarefa específica. O diagrama contém subsistemas para suprimento de potência, processamento e playback de sinal sonoro, realizando interfaces com outros dispositivos e o usuário.

Dessa forma, em um estágio, muitas vezes bastante incipiente de desenvolvimento é possível realizar diversas análises técnicas e propor requisitos que se cumpridos garantiram o sucesso técnico do sistema desenvolvido. Além disso, os modelos desenvolvidos podem passar a fazer parte de um legado da empresa que pode ser reutilizado sempre que necessário.

Figura A2.2: Diagrama de blocos representando um sistema portátil de áudio



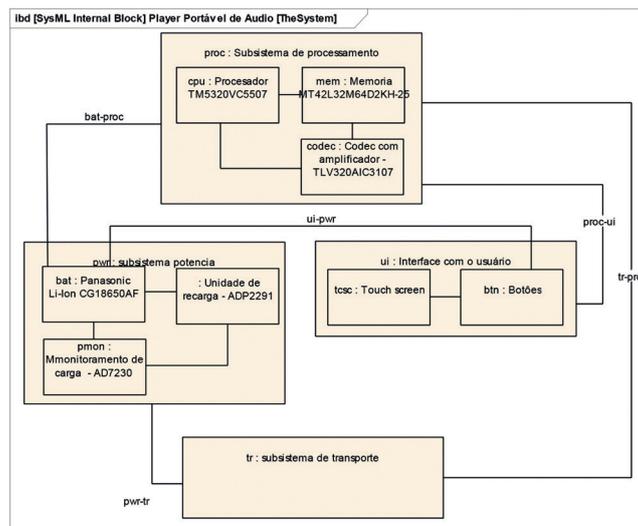
De maneira geral o diagrama de blocos (*Block Definition Diagram*) pode apresentar todos os elementos que compreendem o sistema, além dos usuários, redes inerentes ao sistema, software, e requisitos.

## INTERNAL BLOCK DIAGRAM

As definições de modelos e diagramas de estrutura de cada bloco previamente definido no diagrama de blocos podem ser detalhados por meio de diagrama interno de blocos (*Block Internal Diagram*) ou BID. Os elementos que compõem o *BID* são similares ao Diagrama de Blocos, restringindo apenas alguns tipos de associações entre os elementos.

A Figura A2.3 apresenta um exemplo de *Internal Block Diagram* para um dos blocos do *Block Definition Diagram* da Figura A2.2 (diagrama “pai”). Nota-se que os elementos relativos ao bloco “pai” são herdados para o diagrama interno, tais como portas e definições de atributos, operações e mensagens.

Figura A2.2: Diagrama interno de blocos do sistema portátil de áudio (SPARX-SYSTEMS, 2011)

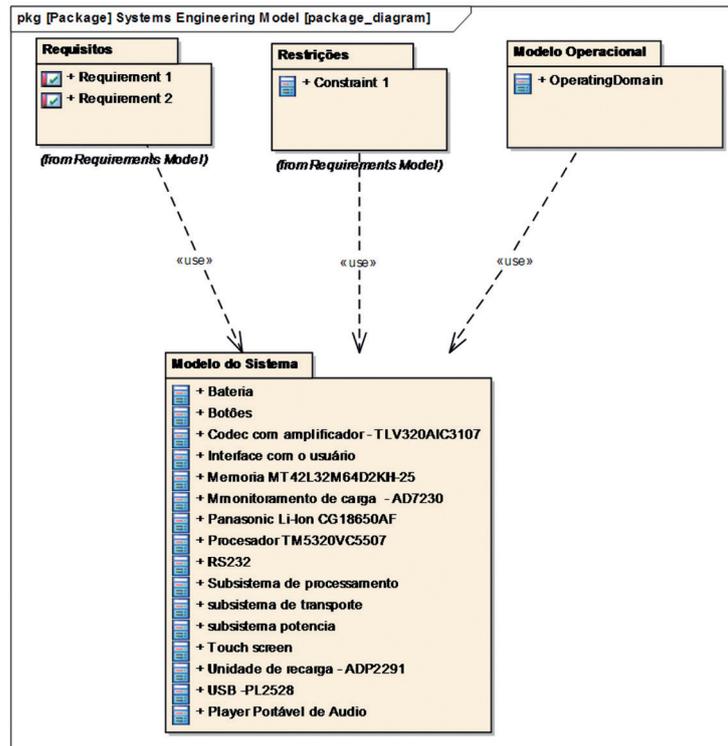


Nesse exemplo, o diagrama (Figura A2.2) é detalhado mostrando-se como cada subsistema é estruturado. Esse diagrama também descreve as relações entre as partes, que descreve como elas estão funcionalmente ligadas umas as outras - por exemplo, A CPU, Memória e CoDec (Codificador/Decodificador) estão juntos no subsistema de processamento.

## PACKAGE DIAGRAM

Pacotes (*packages*) são utilizados para agrupar blocos ou outros elementos sobre um controle único denominado *namespace*. É bastante utilizado no agrupamento de blocos relacionados, podendo ser apresentados sobre um nível mais elevado dos diagramas do projeto. À medida que um sistema complexo denota uma quantidade muito grande de blocos, os chamados *packages* são bastante versáteis na organização do projeto, sendo válido também para promover o reuso dos elementos de modelagem. Podem ser apresentados também como um sistema ou subsistema, organizando em cada *package* um conjunto de todos os blocos relativos ao mesmo. Na Figura A2.3 apresenta-se um diagrama de pacotes em que são mostrados elementos utilizados e organizados em pacotes dos diagramas apresentados anteriormente (*Modelo do Sistema*). Ao mesmo tempo, representa-se que requisitos e restrições são usados aliados ao um modelo operacional (*Modelo Operacional*) para a definição do sistema mencionado anteriormente.

Figura A2.3: Exemplo de Diagrama de Pacotes

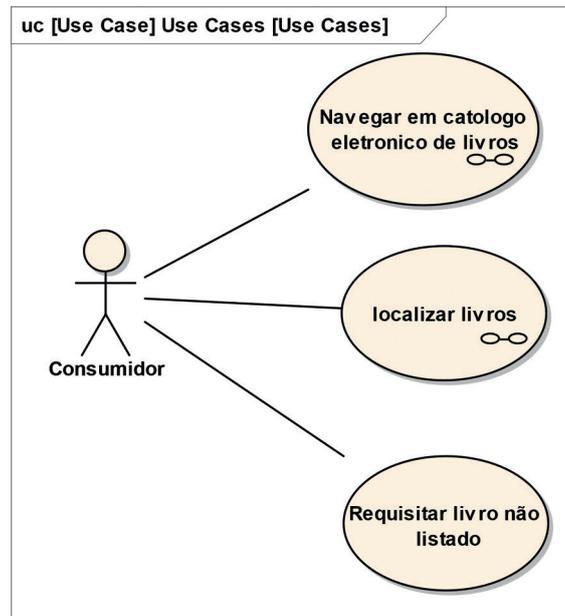


## USE CASE DIAGRAM

O *Use Case Diagram* descreve o uso de um sistema do ponto de vista dos atores (entidades com as quais o sistema interage) e as funcionalidades requeridas por esses atores ao sistema. Ao fornecer essas funcionalidades aos atores, o sistema cumpre em parte, sua missão, pois ele agora tem funções que atendem os atores e possivelmente alguns *stakeholders*.

O exemplo da Figura A2.4 representa alguns casos (funcionalidades) requeridos por um consumidor de livros que realizar compras com o auxílio de um sistema de busca na internet. No exemplo o consumidor precisa navegar no catalogo eletrônico de livros, localizar aquele que mais lhe agrada e quando for o caso requisitar livros não listados no acervo.

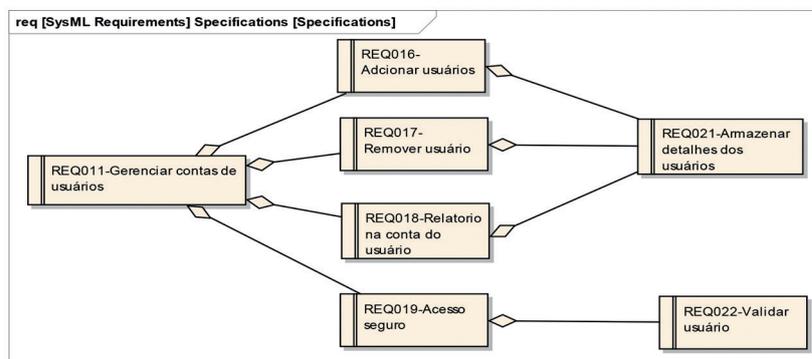
Figura A2.4: Diagrama de Casos de *Uso* (SPARX-SYSTEMS, 2011)



## DIAGRAMA DE REQUISITOS (REQUIREMENT DIAGRAM)

O diagrama de requisitos (*Requirement Diagram*) apresenta as relações entre os diversos requisitos existentes de um sistema. Estas relações podem ser utilizadas para decomposições ou para a rastreabilidade dos requisitos do sistema. Este diagrama proporciona ao analista de sistemas um controle da derivação e rastreabilidade dos requisitos, conforme ilustrado na Figura A2.5. Por meio deste diagrama é possível estabelecer regras de manipulação dos requisitos e apresentação deles de forma integrada.

Figura A2.5: Exemplo de *Diagrama de requisitos SysML*.



As relações presentes no Diagrama de requisitos são agrupadas em relações de dependência, tais como cópia, satisfação, derivação, verificação e refinamento, onde cada uma delas possui características específicas, mostradas no Quadro A2.5.

**Quadro A2.5:** Relações entre requisitos segundo a linguagem SysML (OBJECT MANAGEMENT GROUP, 2010)

<b>Relações</b>	<b>Características</b>
Derivação	Organiza os requisitos numa hierarquia mostrando qual requisito derivou outro. Conseqüentemente, se o requisito “pai” é modificado, o requisito “filho” no mínimo precisa ser reavaliado.
Cópia	Ilustra a relação entre o requisito de um fornecedor e uma cópia criada no cliente, cuja cópia é do tipo read-only. Criando um ambiente de compartilhamento de requisitos.
Satisfação	Ilustra a relação entre um requisito e um elemento de modelo que preenche o requisito
Verificação	Ilustra a relação entre um requisito e um caso de teste que determina se o sistema obedece ou não ao requisito
Refinamento	São todas as relações cujos requisitos têm adicionados mais informações a partir da base de um requisito.

### **A3 ROTEIRO DAS ENTREVISTAS**

#### **1. Objetivo**

Obter a confirmação dos usuários sobre a representatividade dos modelos de processo desenvolvidos. Ainda, procura-se com esse questionário coletar informações sobre os processos representados nele, tais como, potenciais problemas para a execução dos processos, sugestões de melhorias e o impacto nos processos desses problemas.

#### **2. Justificativa**

Atualmente, o autor trabalha na área onde foi realizada a demonstração do método. Devido a isso, optou-se por buscar informações muitas vezes de forma não estruturada para construção da modelagem de processos aproveitando-se da dinâmica do dia para observar, *in loco*, como os processos são executados. Contudo, era necessária uma verificação final acerca dos modelos de processos construídos e obter uma confirmação de que a modelagem realizada representa de fato aquilo que é executado.

### 3. Perguntas que direcionaram a entrevistas

1. Você enxerga o que você faz no seu dia a dia neste mapa de processo?

Obs: mostrar o mapa desenvolvido nesse momento explicando como os processos acontecem.

2. Gostaria de acrescentar algum aspecto que o mapa de processo não contempla atualmente?

3. Qual a sua maior preocupação para executar o seu trabalho?

4. Por quê? Qual o impacto disso na execução do seu trabalho?

5. Qual é a sua sugestão para resolver essa situação?

### 4. Entrevistados (alguns *stakeholders* do processo)

O conjunto de indivíduos que participaram dessa pesquisa são todos os indivíduos que participam diretamente dos processos estudados. Eles são apresentados no Quadro A2.1.

**Quadro A2.1:** Descrição dos entrevistados durante a demonstração do método

Entrevistados	Idade (Anos)	Experiência(Anos)	Formação
Designer	45	7	Design
Designer	30	3	Design
Engenheiro de produto	33	8	Engenharia Mecânica
Projetistas de moldes	55	30	Engenharia Mecânica
Planejador de processos de manufatura	50	20	Engenharia Mecânica
Programador CAM	40	20	Engenharia Mecânica