

Capítulo I. Modelo de Negócio

A Modelagem de Negócio cada vez mais vem sendo usada como uma ferramenta principal ou de auxílio do processo de desenvolvimento de software, visando o levantamento completo dos requisitos do sistema.

Nesse capítulo trataremos de diferentes formas de modelagem de negócio:

- Organograma¹
- Modelagem de Processos
- Modelagem de Regras de Negócio

Um organograma é uma descrição da organização de uma empresa, amplamente divulgada, descrevendo as áreas da empresa e as hierarquias entre elas. O Organograma é ferramenta essencial na compreensão de uma empresa e suas linhas de poder.

A modelagem de funções de negócio permite a compreensão do funcionamento da empresa sem sofrer a intervenção da forma de organização da empresa. De certa forma, pode ser desenvolvido como tanto como um modelo da encarnação do sistema atual como quanto uma ferramenta de substituição ou complementação da análise essencial.

A modelagem de processos demonstra como funciona a empresa, passo a passo, no seu dia a dia. A partir dela pode ser possível levantar os pontos a serem automatizados de um processo e como os processos realmente realizados diferem dos processos normatizados da empresa.

A modelagem de regras de negócio permite a compreensão da empresa de forma mais detalhada que os modelos anteriores. As regras de negócio podem ser utilizadas para ajudar a levantar o modelo essencial, o modelo conceitual de dados, ou para ajudar a implementá-los. Em alguns métodos, pode até mesmo ser utilizada no lugar desses modelos.

I.1 O Organograma

Organogramas são diagramas em árvore que descrevem cargos, por meio de retângulos, e linhas hierárquicas, por meio de linhas. Alguns autores utilizam uma notação levemente mais complicada com o objetivo de descrever diferenças sutis em uma organização.

Em geral o analista não precisa levantar o organograma, pois a empresa já o possui, mas é comum que haja algumas mudanças. Na verdade, não é trabalho do analista

¹ O organograma é uma das formas mais simples e antigas de modelar uma empresa.

de sistemas construir o organograma da empresa, porém ele precisa conhecê-lo para melhor desenvolver o seu trabalho. Para isso é importante obter esse documento junto ao cliente, e verificar não só a hierarquia de cargos, mas também quem ocupa cada cargo, e como entrar em contato com cada um dos membros da organização que possa ter interesse no sistema.

A importância de conhecer o organograma da empresa se reflete tanto na modelagem propriamente dita, pois ele fornece a descrição da empresa que será convertida para objetos do modelo, como no processo de modelagem, pois a partir do organograma temos o conhecimento de cargos e responsabilidades, definindo pessoas a serem entrevistadas.

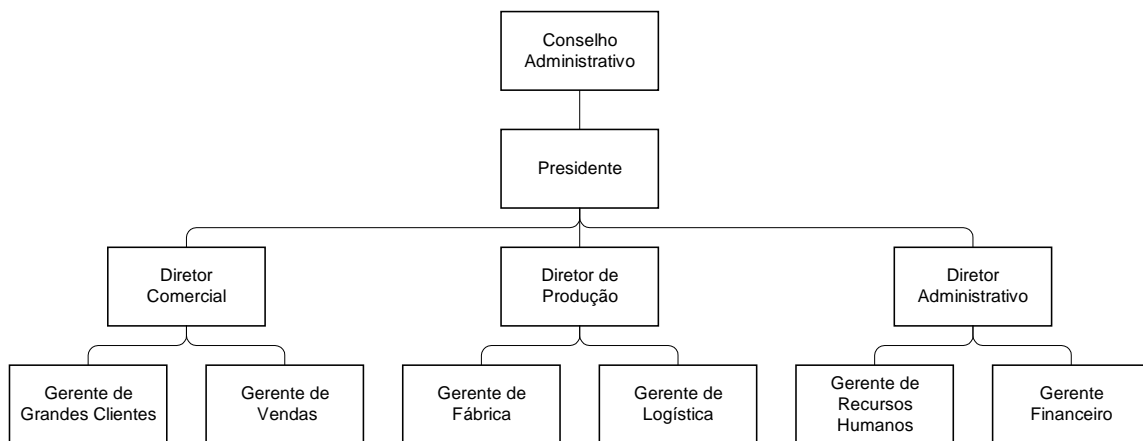


Figura 1. Um exemplo de organograma simples, contendo apenas cargos.

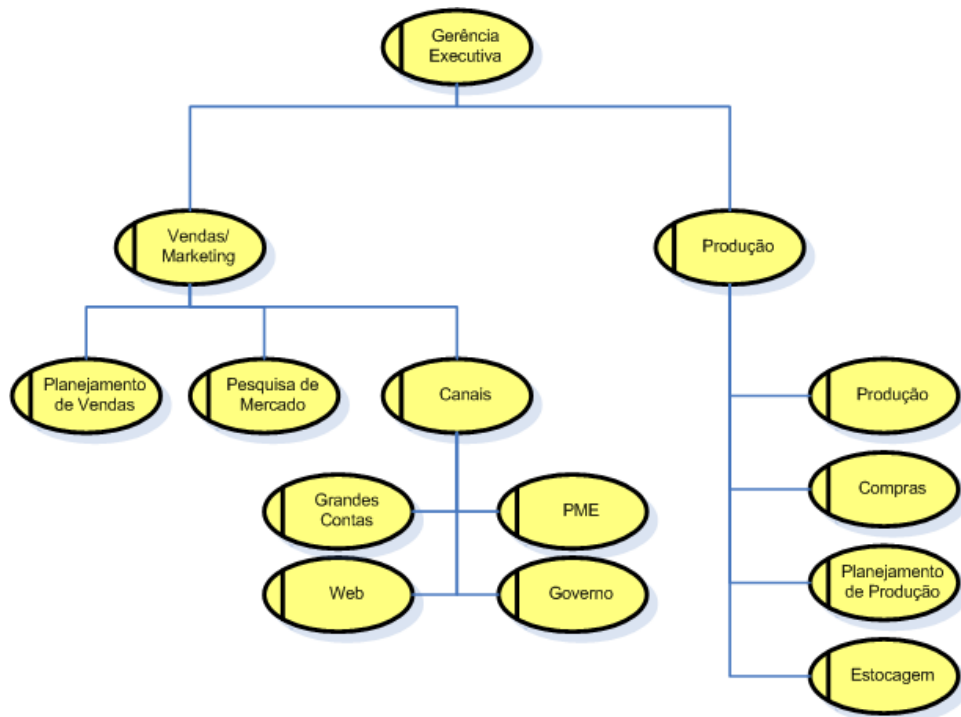


Figura 2. Exemplo de organograma simples, contendo departamentos, feito com símbolos da técnica EPC².

Um organograma pode ser utilizado para representar diferentes formas de subordinação, como a subordinação direta (onde o subordinado deve cumprir as ordens de seu chefe), a assessoria (onde o assessor fornece conselhos e pareceres) e a subordinação funcional (onde o superior pode determinar funções e métodos a outras áreas). Normalmente a subordinação direta é representada por uma linha cheia vertical, a assessoria por uma linha cheia horizontal e a subordinação funcional por uma linha pontilhada.

Ao levantar o organograma, pode ser interessante também levantar as descrições dos cargos, se elas existirem (o que não é comum, principalmente em pequenas e médias empresas).

Este texto não tratará do processo de levantamento do organograma, pois isso é mais afeito à administração. Fica, porém, o lembrete de sua importância como documento de referência ao analista.

1.2 Níveis de abstração tratados nesse capítulo

Nesse capítulo analisaremos a empresa do nível mais abstrato para o mais detalhado.

O nível mais abstrato descreve as funções de negócio da organização, sem se preocupar como essas funções são executadas ou em que ordem. Nesse nível, estamos

² A técnica EPC será vista ainda neste capítulo

preocupados em definir as responsabilidades da organização, e as entradas e saídas necessárias para cumprir essas responsabilidades.

Em um nível mais detalhado, analisamos os processos da empresa. Nesse nível estamos preocupados em como a organização executa suas funções, com que passos, e por meio de que pessoas.

Finalmente, no nível mais detalhado de todos, descreveremos as regras de negócio que regem esses processos.

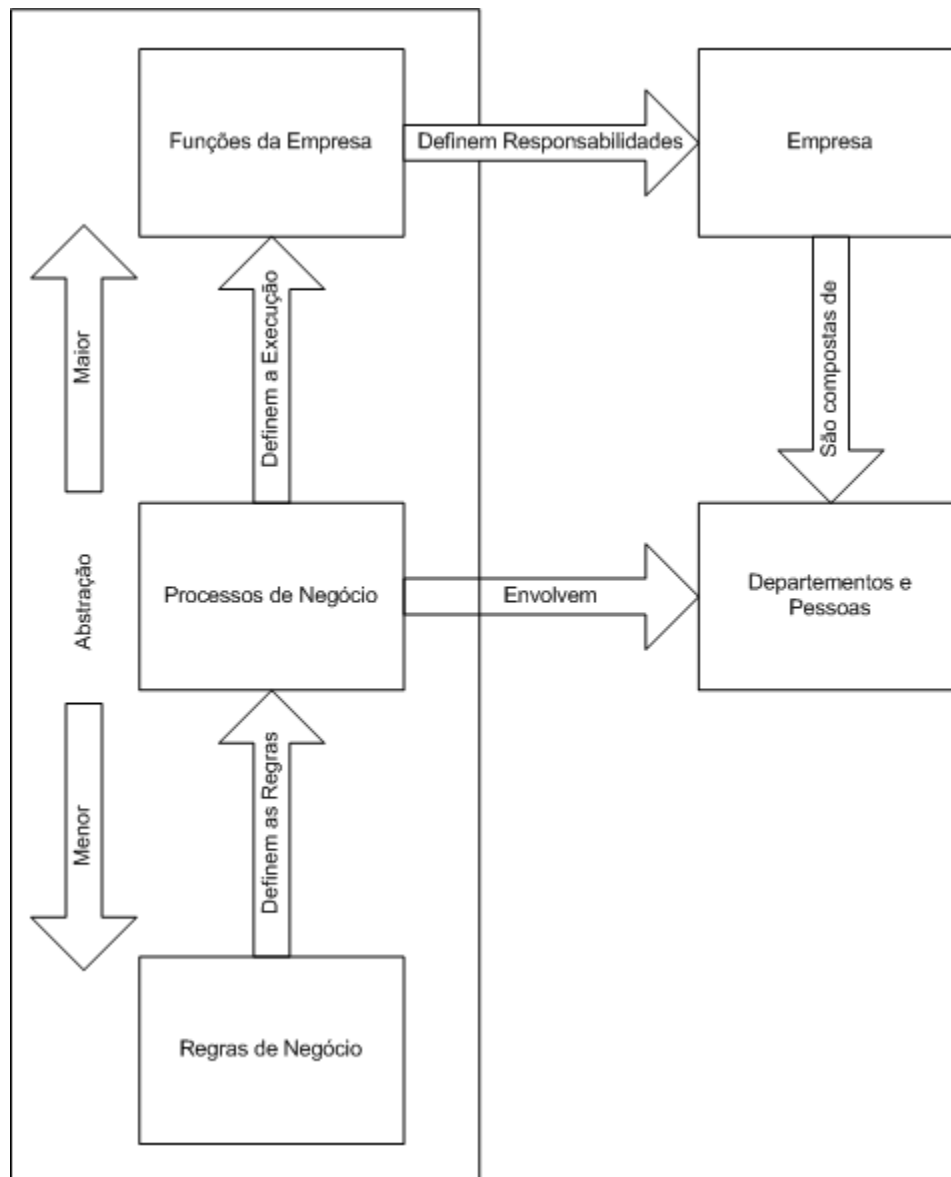


Figura 3. Níveis de abstração analisados nesse capítulo e suas relação com a empresa.

I.3 Processos de Negócio

Processos de negócio são grupos de decisões e atividades, logicamente relacionadas, requeridas para o gerenciamento de recursos da empresa³.

Podemos entender “processos de negócio como uma sequência de passos e decisões, iniciadas em resposta a um evento de negócio, que alcança um resultado específico e mensurável, tanto para o consumidor do processo como para outros interessados (*stakeholder*).”[B22] Além disso, é necessário que identifiquemos instâncias específicas dos resultados.

Não é trivial identificar processos, pois eles acontecem dentro da organização de forma esparsa, provavelmente envolvendo diversas pessoas e departamentos. Também não é trivial representar processos, pois corremos vários riscos, como fazer uma representação muito complexa ou muito simples, ser impreciso ou utilizar o método de forma errada.

Normalmente, sistemas de informação são utilizados para automatizar processos de negócios. Pode ser necessário, antes de fazer o levantamento de requisitos de um sistema, levantar como funciona o processo onde ele está inserido ou que vai substituir.

Nesse tipo de modelagem estamos preocupados com a forma em que os processos são executados dentro da empresa. Existem várias formas de se tratar a descrição de processos atualmente, variando em diferentes níveis de complexidade.

I.4 Fluxogramas

Fluxogramas são provavelmente a forma mais tradicional de modelar processos. Atualmente, fluxogramas são poucos utilizados, tendo sido substituídos por outras formas, semelhantes, que foram criadas com a finalidade de evitar problemas comuns encontrados na criação dos fluxogramas.

É importante frisar que fluxogramas podem ser utilizados em vários níveis do desenvolvimento de sistemas. Seu uso mais comum, no passado, era na especificação de algoritmos, mas esta prática está totalmente superada, sendo normalmente substituído por programas em linguagens de alto nível. Seu uso na especificação de processos também está em franca decadência, nesse caso o que se vê é a sua substituição por diagramas mais modernos (incluindo o uso de fluxogramas em diagramas de raia⁴).

I.5 EPC

EPC é a sigla em inglês para *Event Driven Process Chain* (Cadeia de Processos Dirigida por Eventos). Esse método é parte simplificada do método ARIS usada para modelagem de processo e tem grande aceitação no mundo, estando muitas vezes associado à implantação de sistemas de ERP SAP/R3.

³ Esta é a definição do BSP

⁴ swimlane diagrams

Nesse método, um processo é modelado segundo fluxo de eventos e atividades.

As principais primitivas, descritas na figura abaixo, são:

- **Atividades**, que representam funções, decisões, tarefas, processamento de informações ou outros passos do processo que precisam ser executadas e suportam um ou mais objetivos de negócio.
 - Possivelmente
 - São iniciadas ou iniciadas ou habilitadas por eventos,
 - Geram eventos
 - Consomem recursos,
 - Exigem gerenciamento, tempo, e atenção.
 - Podem representar:
 - Atividades tangíveis
 - Decisões (mentais)
 - Processamento de Informações
 - Um processo que é descrito por outro diagrama EPC.
 - Nesse caso, as ferramentas CASE fornecem alguma marca indicativa de que a função pode ser expandida.
- **Eventos**, que representam situações, ou estados do sistema, antes ou depois da execução de uma função.
 - Eventos podem ser
 - Ativadores de uma atividade
 - Ativados por uma atividade
 - “Estados comercialmente relevantes, que controlam ou influenciam a progressão subsequente de um ou mais processos de negócio”⁵
 - Um evento não consome tempo nem recursos por si só.
- **Regras**, que permitem a unificação e separação de fluxos segundo os conceitos de E, OU ou OU-exclusivo.
- **Interface de Processo**, que indica que os próximos passos são descritos por meio de um outro diagrama .

⁵ <http://www.ariscommunity.com/help/aris-express/35908/>




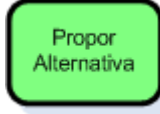



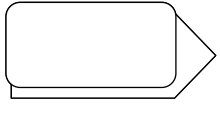
Tipo	Símbolo ARIS Express	Símbolo Tradicional
Evento		
Atividade		
Regras		
Interface de Processo		

Figura 4. Principais componentes de um EPC

EPCs (e eEPCs) são modelados semanticamente como Redes de Petri. O leitor interessado pode encontrar na Internet vários documentos tratando do assunto e demonstrando problemas semânticos no uso de algumas construções.

I.5.1 Uso Básico

No diagrama abaixo, podemos ver um exemplo simples de EPC seguindo as regras básicas de sintaxe.

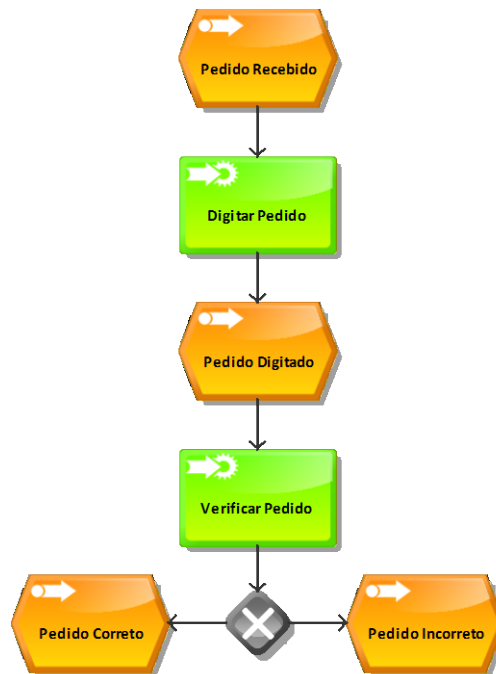


Figura 5. Exemplo de um EPC de um processo de recebimento de pedido

A seguir, descreveremos as formas básicas de uso dos componentes de um diagrama EPC.

I.5.1.1 Atividades (ou Funções)

A seguir, apresentaremos algumas regras básicas.

O nome de uma atividade é sempre da forma:

<verbo no infinitivo> <objeto direto>

Como, por exemplo, na figura a seguir:



Figura 6. Exemplo de um processo.

As atividades podem ser vistas como transformações. Elementos que descrevem uma atividade são:

- Informação que entra
- Objetivos
- Recursos

- Consumíveis, como matéria prima
- Não consumíveis, como máquinas
- Procedimentos realizados
- Capacidades necessárias
- Informação que sai
- Produto ou serviço entregue

1.5.1.2 Eventos

Os Eventos representam a mudança do estado do mundo enquanto o processo é executado⁶. Três são os tipos de eventos candidatos a serem registrados em um processo⁷:

1. A mudança de estado do mundo que faz com que o processo comece;
2. Mudanças internas de estado enquanto o processo é executado, e
3. O resultado final do processo que tem um efeito externo.

Claramente, eventos definem pré-condições para que uma atividade ocorra, e também pós-condições ao fim da atividade.

Um evento segue a sintaxe:

<Sujeito> <Verbo na voz passiva>

Ou, mais simplesmente:

<Substantivo> <Adjetivo>

Como os exemplos da figura a seguir:



Figura 7. Exemplos de eventos

Segundo a versão original de EPCs, sempre deveria haver um evento entre dois processos. Atualmente é permitido que uma sequência de processos não tenha nenhum evento entre eles. Logo, os diagramas das figuras a seguir podem ser usados como equivalentes.

⁶ Aris Design Platform

⁷ Idem

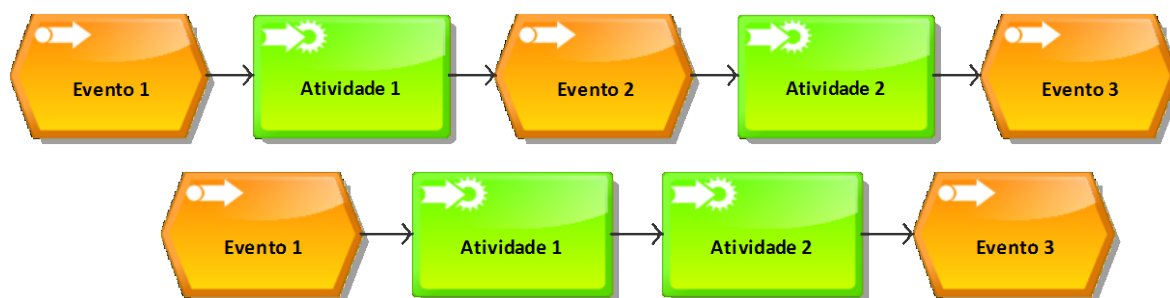


Figura 8. Duas cadeias que podem ser usadas de forma equivalente na notação atual de EPC. O evento intermediário, como não serve para indicar caminho, pode ser dispensado.

I.5.1.3 Conectores

Os conectores E, OU e OU-EXCLUSIVO funcionam de duas formas: como divisores de caminho (*split*) e como junção de caminhos (*join*). Não é possível utilizar um conector simultaneamente nas duas formas.

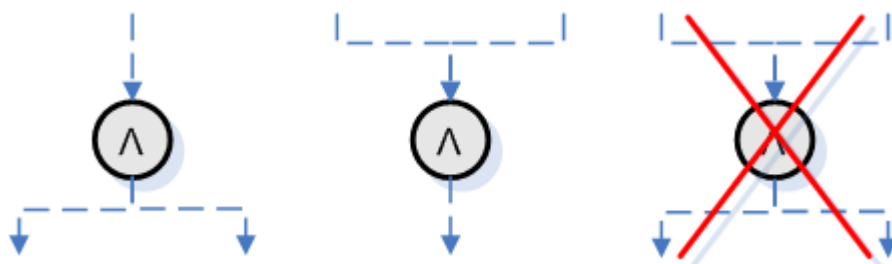


Figura 9. Um conector só deve ser utilizado na forma de divisor ou junção, e nunca das duas formas ao mesmo tempo.

De acordo com as regras sintáticas para EPCs, é possível que um processo produza um ou mais eventos simultaneamente, pelo conector E, ou não, pelos conectores OU ou OU-EXCLUSIVO. Já um evento só pode habilitar um grupo de processos simultaneamente, pelo conector E, não sendo viável que de um evento se tenha uma opção de caminho, não sendo possível a partir de um evento alcançar diretamente um conector OU ou OU-EXCLUSIVO.

As figuras a seguir apresentam as configurações permitidas e proibidas para o uso de conectores OU-EXCLUSIVO. As regras para conectores OU são equivalentes. Para conectores E não há configurações proibidas. Sempre que apresentamos 2 eventos ou 2 processos é possível utilizar mais desses componentes.

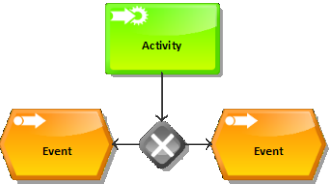
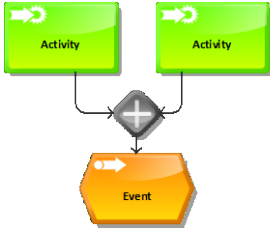
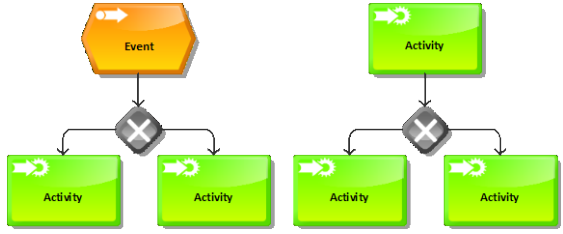
	<p>Configuração permitida para o conector OU-EXCLUSIVO (XOR), funcionando como marca da necessidade de escolher entre dois ou mais caminhos (como um if-then-else ou case). Os eventos indicam que caminho deve ser seguido. Os eventos não podem acontecer ao mesmo tempo.</p>
	<p>Configuração permitida para o conector OU-EXCLUSIVO (XOR), funcionando como ponto de convergência (join) entre dois caminhos que não serão atravessados simultaneamente.</p>
	<p>Configurações não permitidas para o conector OU-EXCLUSIVO (XOR), pois não é possível escolher que caminho seguir.</p>

Tabela 1. Configurações para o conector OU-EXCLUSIVO (XOR).

Conectores em branco podem ser utilizados para representar em um diagrama mais simples um conjunto de conexões muito complexas. Veja o exemplo nas figuras a seguir.

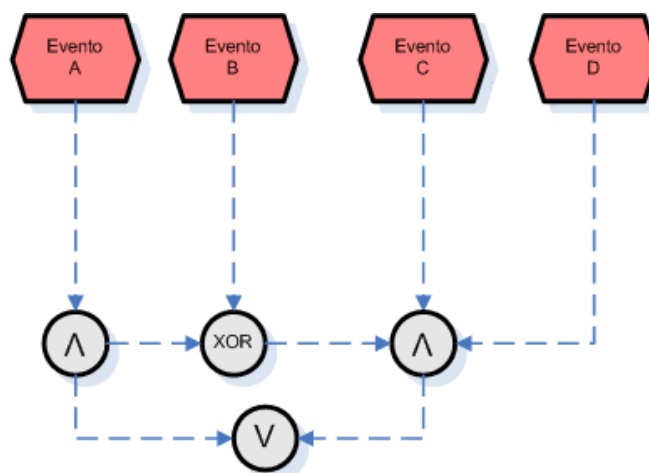


Figura 10. Exemplos de decisão complexa.

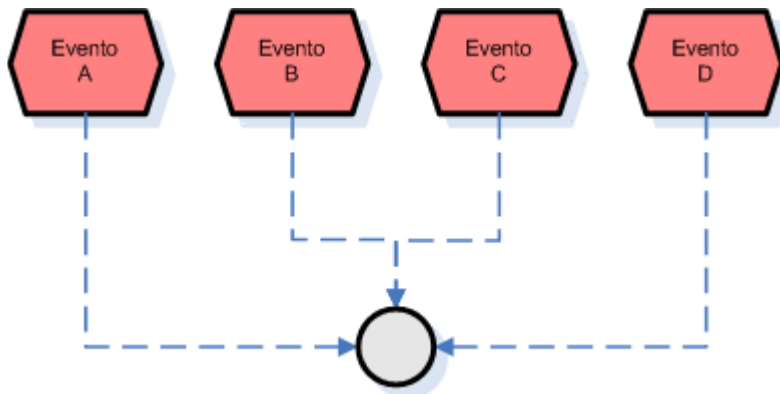


Figura 11. Exemplo de uso de conector em branco para esconder uma decisão complexa.

I.5.1.4 Interfaces de Processo

As interfaces de processos são usadas para indicar que um processo acontece quando outro processo acaba. Caminhos são sempre conectados a eventos, devendo aparecer antes ou depois deles, e recebem o nome do processo origem ou destino. Na figura a seguir é possível ver um exemplo de sua utilização.

Elas só devem ser usadas para ligar processos de um mesmo nível.

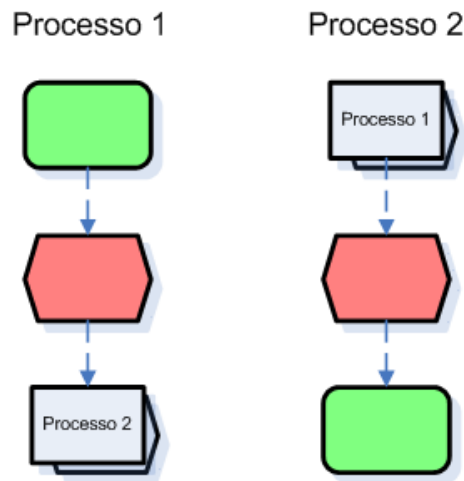


Figura 12. Como indicar que um processo é continuação lógica de outro processo. No “Processo 1” indicamos um caminho para o “Processo 2”, e nesse processo indicamos que o caminho é proveniente do “Processo 1”.

I.5.2 eEPC

eEPC é a sigla em inglês para *Extended Event Driven Process Chain* (Cadeia de Processos Dirigida por Eventos). Nessa extensão é possível declarar mais algumas informações sobre o processo sendo descrito.

Esses elementos adicionais funcionam basicamente como comentários ao processo que está sendo documentado. Assim, depois de descrito o processo pelo método

não estendido, colocamos sobre eles novos elementos documentando informações como quem realiza o processo, que informação utiliza, que produtos gera ou consome, etc...

Os principais elementos adicionais em um eEPC são:

- Unidades Organizacionais, que representam departamentos envolvidos em um processo.
- Pessoas, que representam pessoas ou papéis envolvidos em um processo.
- Informação ou dados, que representam informação utilizada ou gerada em um processo.
- Produtos ou serviços, que são gerados ou consumidos pelo processo.
- Objetivos, que representam o motivo da realização de um processo ou tarefa







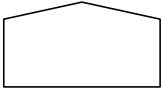
Tipo	Símbolo
Unidade Organizacional	
Informação	
Pessoa ou Cargo	
Fluxo de Informação	
Relações Organizacionais	
Produto ou Serviço	
Objetivo	

Figura 13. Elementos complementares dos diagramas eEPC.



Figura 14. Nova lista e imagens dos elementos complementares dos diagramas EPC.

A seguir, apresentamos um exemplo do mesmo diagrama EPC para demonstrar os usos dos elementos adicionais.

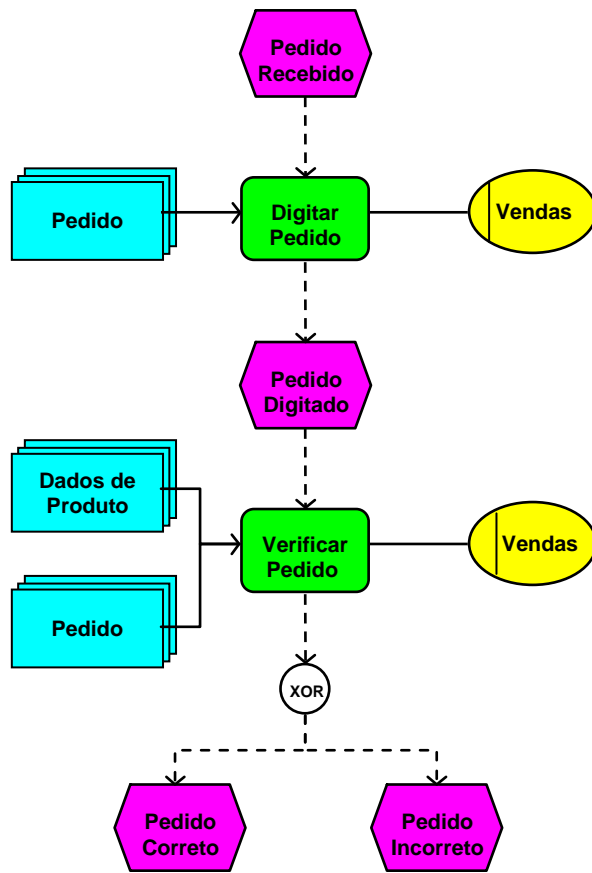


Figura 15. Exemplo de eEPC, a partir do EPC anterior. Os símbolos suplementares podem variar de acordo com a ferramenta case sendo usada.

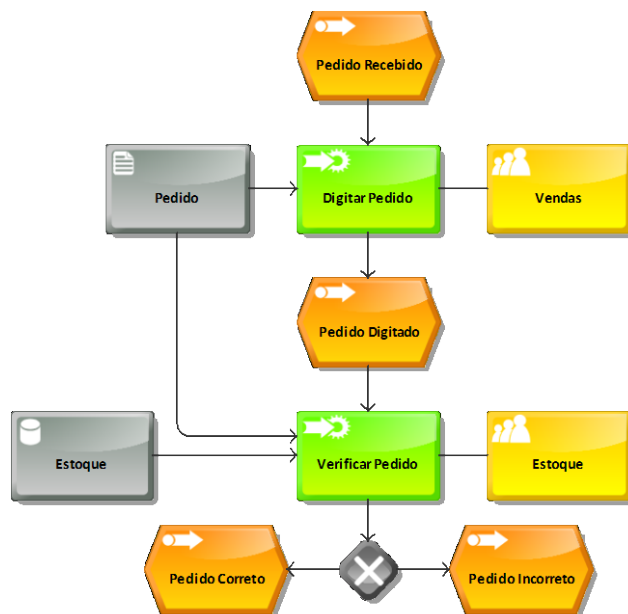


Figura 16. Desenho novo do modelo EPC

Algumas formas dos diagramas EPC estendidos apresentam várias versões. Por exemplo, o símbolo de informações apresenta pelo menos 3 versões diferentes, como na figura a seguir.

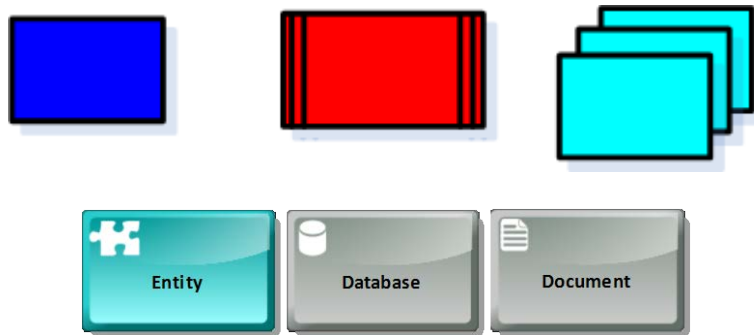


Figura 17. Diferentes versões para informação

A figura a seguir apresenta um exemplo usando alguns símbolos diferenciados.

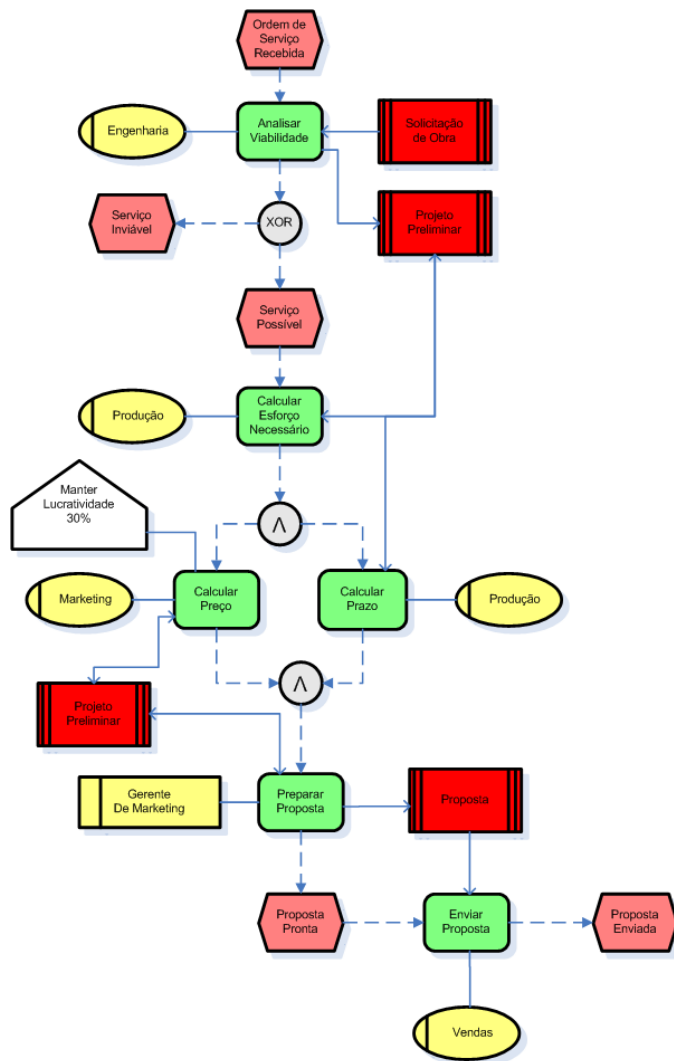


Figura 18. Um exemplo de EPC com símbolos de informação diferenciados, e ainda um objetivo indicado.

I.5.3 Diagramas de Resumo

Também é possível desenhar diagramas representando o resumo de um processo, como o exemplo a seguir.

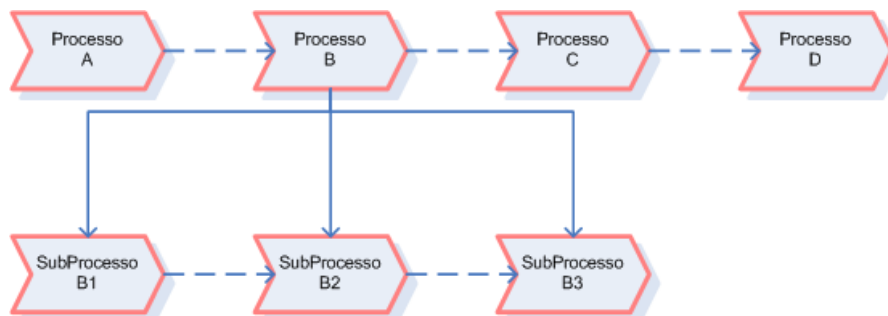


Figura 19. Um resumo de processo. A linha tracejada indica fluxo de controle, a linha cheia indica divisão hierárquica.

I.5.4 Formas diferentes dos diagramas

É possível desenhar os EPCes de diferentes formas. Uma dessas formas permite que iniciemos apenas com as comunicações entre unidades, como é mostrado na figura a seguir. Isto permite um conhecimento inicial do processo que pode ser muito útil na sua discussão durante reuniões ou entrevistas.

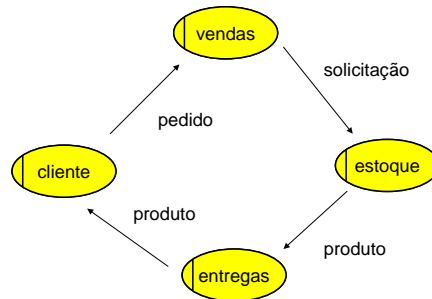


Figura 20. EPCe baseado apenas em unidades da organização.

Os mesmos símbolos também podem ser usados para construir organogramas.

I.5.5 EPC e 5W2H

Um evento indica quando (*when*) algum processo, função ou tarefa deve ser iniciado.

Uma função ou tarefa indica o quê (*what*) deve ser feito.

Uma unidade organizacional indica quem (*who*) deve fazer.

I.5.5.1 Passos para construir modelos EPC/EPCe[B23]

Identifique os eventos que iniciam as funções, que servem como gatilhos para o processo se iniciar. Normalmente vem de “fora para dentro” do processo.

Identifique as funções do processo, associando-as aos eventos que as iniciam e sua sequência.

Decomponha as funções, verificando se são ações lógicas simples ou compostas, executadas por uma ou mais pessoas (ou ainda um sistema de computador). Verifique também se a função é uma transação isolada ou pode ser dividida em partes, se pode ser interrompida em um momento específico e se existe um evento que a interrompa ou que a faça funcionar novamente.

Analise os eventos novamente, definindo-os e refinando-os se necessário. Garanta que são necessários e suficientes para iniciar a função. Analise se existem casos especiais nos quais as funções acontecem ou não. Use operadores lógicos para montar as relações entre os eventos.

Identifique os eventos de finalização e as saídas (tanto de material quanto de informação). Procure identificar quem processos e pessoas no resto da organização que dependem do processo sendo analisado.

EPCs podem ser muito pequenos ou enormes, dependendo unicamente do tamanho do processo que está sendo mapeado.

I.5.6 Regras de ouro de EPCs[B23]

- Não existem nós isolados
- Funções e eventos têm apenas uma entrada e uma saída
- Operadores lógicos contêm vários fluxos de entrada e um de saída, ou um único fluxo de entrada e vários de saída.
- Conexões entre operadores lógicos são acíclicas.
- Dois nós só podem possuir um único link entre eles
- Existe um evento inicial e um evento final
- Eventos não tomam decisões, logo só possuem uma saída.
- Evite o uso de OU como junção, pois sua semântica não é clara.
- Cuidado com modelos complexos, que podem levar a *deadlocks*, principalmente com o uso do operador E.
- Mantenha o diagrama estruturado. Sempre que dividir um caminho com XOR ou E, feche o caminho com o mesmo símbolo.

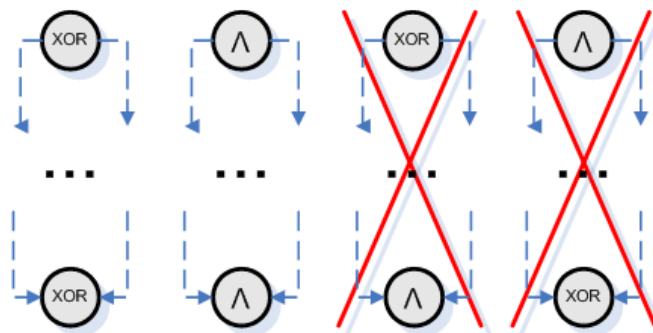


Figura 21. Manter os diagramas estruturados significa fechar cada bloco de caminhos com um conector do mesmo tipo que abriu esse bloco. Isso evitará deadlocks.

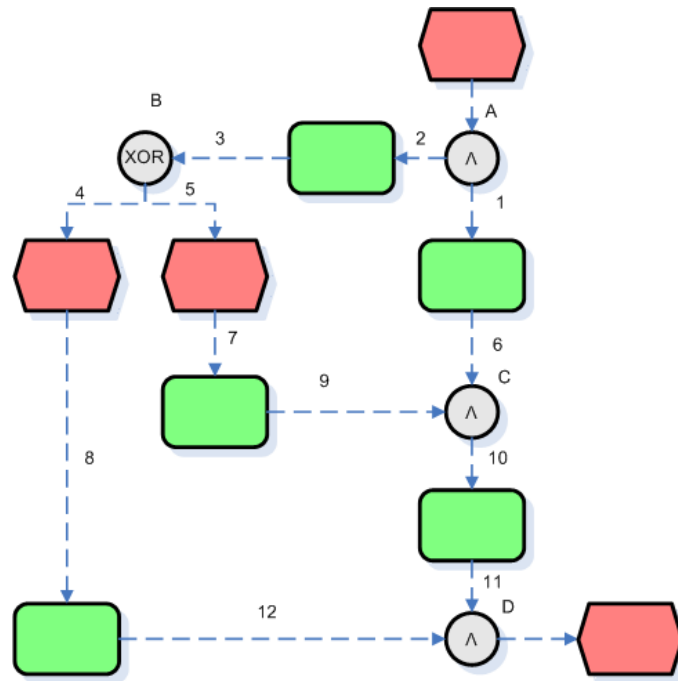


Figura 22. O modelo acima, demonstra um *deadlock*. Como comentário, os números identificam as setas e as letras identificam os conectores⁸. No conector XOR (B), um dos caminhos 4 ou 5 será escolhido. Porém, se for escolhido o caminho 5, o sistema para de funcionar no conector E (D), pois esse E nunca será satisfeito (já que 4-8-12 não será percorrido). Caso seja escolhido o caminho 4, o conector E (C) nunca será satisfeito (pois 5-7-9 não será percorrido), e o sistema vai parar em (C), esperando 9, e (D), esperando 11.

I.5.7 EPC e Loops/Laços

Por diferentes motivos, laços arbitrários não funcionam bem com EPCs. Alguns textos explicitamente proíbem laços, outros não apresentam exemplos, desconsiderando-os implicitamente. Porém, é possível descrever laços em EPML, a linguagem XML para descrever EPCs.

Fazemos as seguintes recomendações quanto ao uso de laços nos diagramas EPC:

- Evite os laços
 - Use apenas os laços estritamente necessários
- Use apenas laços simples, baseados em XOR
 - Busque usar laços apenas em processos de correção e similares.

⁸ Isso não faz parte da notação, está sendo usado apenas para explicar o *deadlock*.

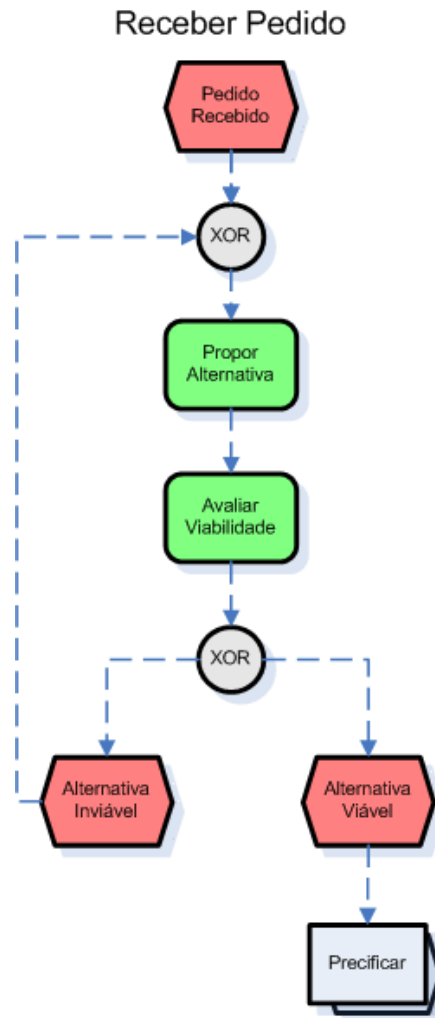


Figura 23. Um EPC válido, com um loop simples baseado em XOR. Também é dado um exemplo de como conectar esse EPC a um outro processo.

I.5.8 Algumas sentenças em EPC

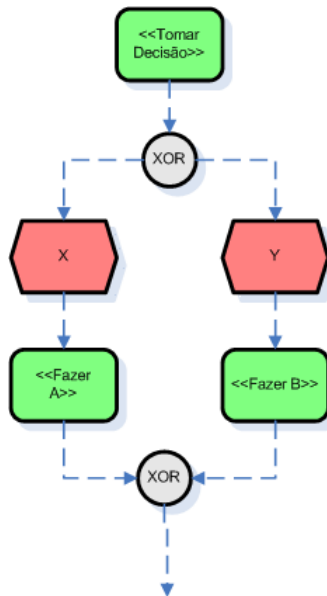


Figura 24. Descrição EPC para “Se X então Fazer A, Se Y, então Fazer B”

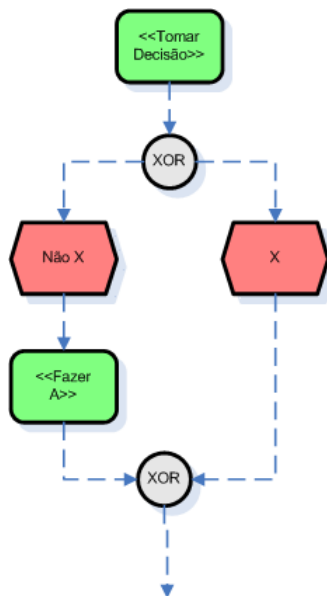


Figura 25. Descrição EPC para “Se Não X então Fazer A”

I.5.9 Exemplo

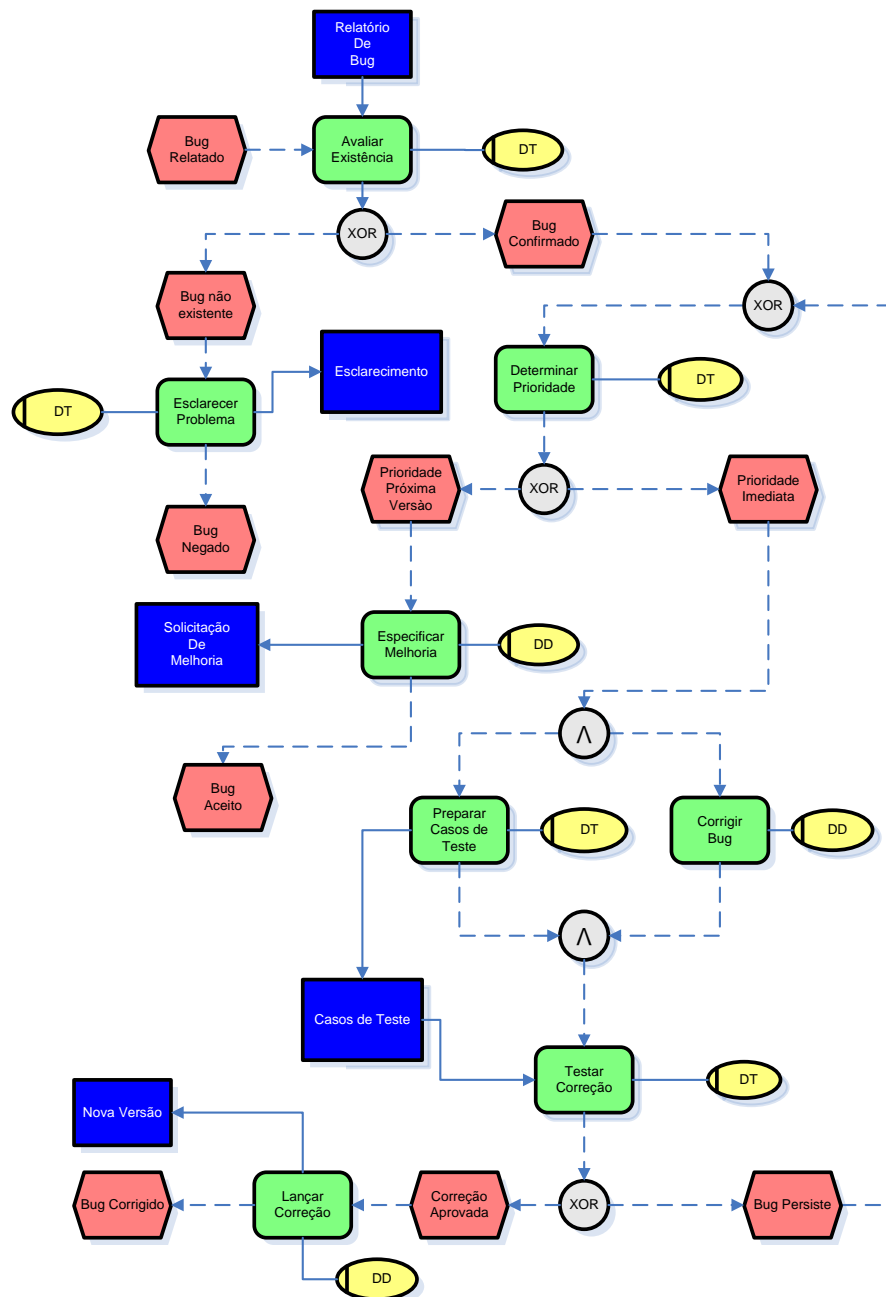


Figura 26 EPC para o texto apresentado em I.4.4

I.6 Diagramas de Atividade (UML 2.0)

O Diagrama de Atividade (DA) é uma das formas que UML [B24] propõe para modelar os aspectos dinâmicos de um sistema, sendo basicamente um tipo de avançado de fluxograma mostrando como o controle e dados fluem entre atividades. A partir da versão 2.0 um DA tem a semântica de uma Rede de Petri.

Segundo o padrão da UML 2.0, modelar uma atividade tem como ênfase modelar coordenação de comportamentos de mais baixo nível, as ações. Isso significa modelar a seqüência e a condição em que esses comportamentos ocorrem.

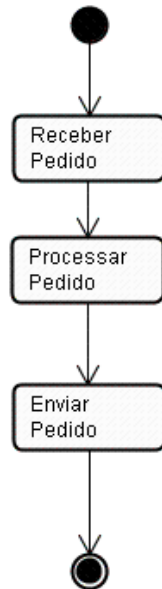


Figura 27. Exemplo de um Diagrama de Atividades simples, com uma seqüência de três ações.

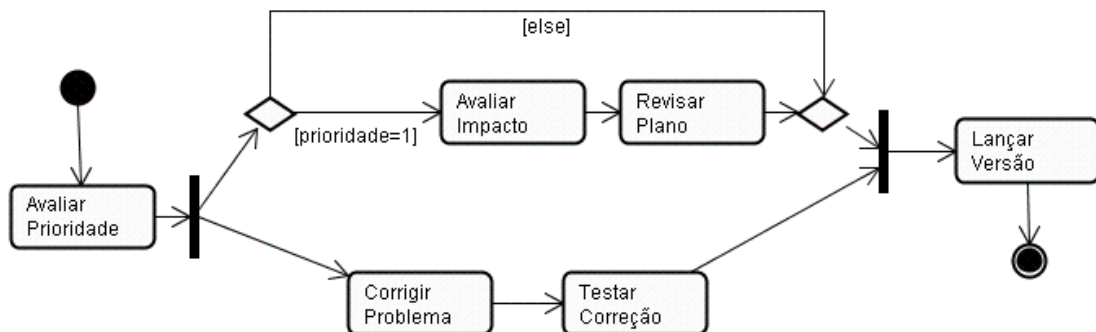


Figura 28. Exemplo de um diagrama de atividades mais complexo, contendo decisões e paralelismo.

I.6.1 Nós, vértices

Um DA é um grafo composto de nós e arestas direcionadas (isto é, que partem de um nó e chegam em outro nó, como uma seta). Os nós representam os passos em uma atividade e podem ser nós executáveis (ou ações), nós de controle ou objetos. As arestas representam fluxos, de dados ou de controles. Na Figura 50 podemos ver três tipos de nós (nó de início de atividade, nó de fim de atividade e três ações) e um tipo de vértice (de fluxo de controle).

A partir da especificação UML 2.0, Diagramas de Atividades são semelhantes a Redes de Petri. O significado disso pode ser descrito da seguinte forma: em qualquer momento, um ou mais nós de um DA podem estar marcados com “tokens”, isto é,

“marcas”. Uma marca pode ser entendida, de forma geral, como dizendo que aquele nó “está ativo” ou “é o passo corrente”.

A semântica de uma atividade é baseada no fluxo de tokens. Um token contém um objeto, um dado ou uma indicação de controle e está presente em um nó específico da atividade. Quando um nó começa a executar, ele aceita tokens de todos ou apenas alguns das arestas que apontam para ele. Quando termina a execução de um nó o token é removido desse nó e oferecido para todos ou apenas alguns das arestas que partem dele.

Por exemplo, na Figura 52 podemos ver um token caminhando entre os vários nós do diagrama, de acordo com

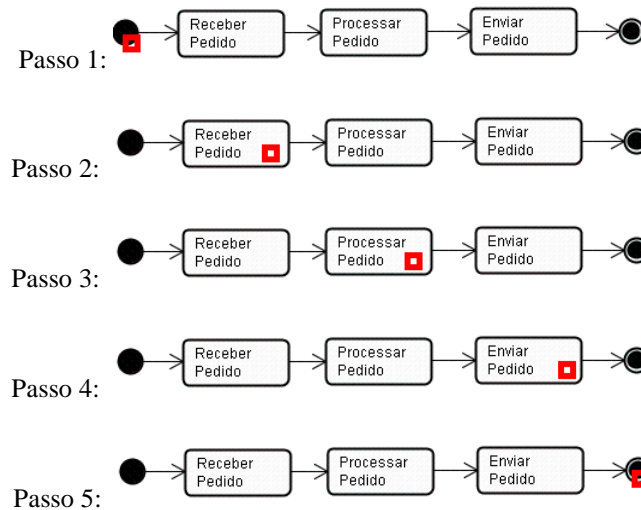


Figura 29. Um “token” vai caminhando no Diagrama de Atividades, indicando o nó ativo a cada instante.

I.6.2 Ações, Objetos e Fluxos

O componente básico do Diagrama de Atividades é a **ação**⁹. Uma ação representa um passo simples que não é mais decomposto dentro do modelo da atividade sendo descrita. É importante notar que uma ação é simples apenas do ponto de vista da atividade sendo modelada naquele instante, mas pode implicar em muitos outros passos detalhados (e muitos outros níveis de atividades) em outros níveis de abstração mais detalhados. Em especial, uma ação pode representar uma chamada para outra atividade. As atividades, dessa forma, são reutilizáveis, mas as ações só existem no contexto da atividade em que aparecem.

Uma ação é descrita em um DA na forma de um retângulo com os cantos arredondados e um rótulo, com seu nome, no meio.

⁹ As ações são um conceito novo em UML 2.0, tomando o lugar dos conceitos *ActionState*, *CallState* e *SubactivityState* existentes na UML 1.5.

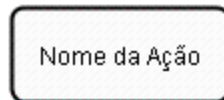


Figura 30. Símbolo para uma ação (com texto).

Pré e pós condições podem ser indicados usando notas (comentários) ligados a ação, com as palavras chaves `<<localPrecondition>>` e `<<localPoscondition>>`, como mostra o exemplo a seguir. Uma pré-condição sempre deve ser verdade antes da ação iniciar, enquanto uma pós-condição sempre será verdade após a ação ser executada.

Em relação a semântica do DA, uma ação só se inicia quando todas as arestas que chegam nela contém tokens. Quando a execução da ação está completa, ela oferece tokens para todas as arestas saindo dela. É interessante notar que se vários *tokens* estão disponíveis em uma mesma aresta, todos são consumidos.

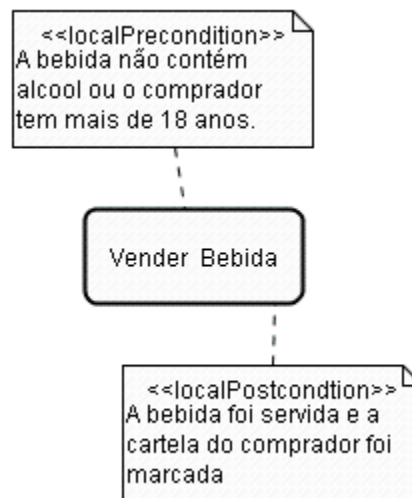


Figura 31. Exemplo de uso de comentários para indicar pré e pós condições.

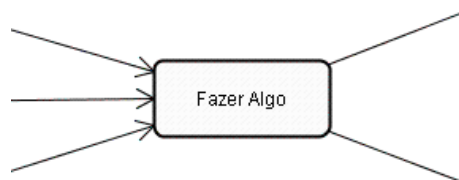


Figura 32. A ação “Fazer Algo” só se inicia quando todas as setas que chegam nela estão habilitadas. Quando ela termina, habilita todas as setas que partem.



Figura 33. A descrição da ação “Fazer Algo”, feita na figura anterior, equivale a usar um *join* e um *fork* na sua entrada e saída, respectivamente.

Uma atividade é então um conjunto (ou sequência) de ações, coordenadas de alguma forma. As arestas, ou fluxos, servem para indicar como as ações são seqüenciadas. Existem dois tipos de fluxos: fluxos de controle e fluxos de dados/objetos.

Os fluxos são indicados por meio de setas. Os fluxos de controle são indicados por setas simples e transportam apenas “tokens” de controle, como na Figura 57.

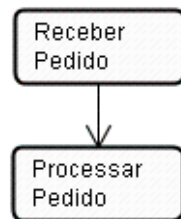
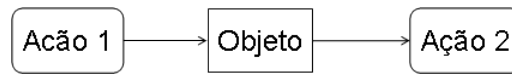
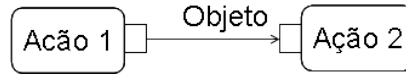


Figura 34 Notação para um fluxo de controle, usando apenas uma seta simples.

Um fluxo de objetos é representado por meio de três notações possíveis, apresentadas na Figura 58. Na primeira notação o objeto pertence ao fluxo. Na segunda notação, aparecem os “pins”, que podem ser de entrada ou de saída, alternativamente. Por um fluxo de objetos só pode passar um “token” de objeto. Na última notação, o objeto é “elidido”.



(notação com um objeto explícito)



(notação usando “pins”)



(notação com o objeto indicado)

Figura 35. Notações alternativas para fluxos de dados ou objetos.
Na primeira o objeto aparece explicitamente. Na segunda, são usados “pins”, as pequenas caixas junto as ações, e a seta recebe um rótulo com o nome do objeto ou dado. Na terceira o objeto é só indicado, mas não identificado.

Um fluxo de objeto entrando em uma ação representa um dado ou objeto necessário para a ação executar.

Na notação anterior apresentamos mais um tipo de nó, o objeto. Um objeto pode ser usado de muitas formas, sendo a primeira delas representar um objeto que passa por um fluxo de dados. Nesse caso, o objeto guarda apenas um token. Mais adiante no texto veremos outros uso para nós do tipo objeto.

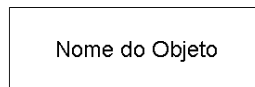


Figura 36. Representação de um objeto

Uma ação, para iniciar, deve ter disponíveis tokens em todos os seus fluxos de entrada, sejam de dados ou de controle¹⁰. Ao terminar, deixa tokens disponíveis em todos os seus fluxos de saída¹¹.

Cada fluxos pode conter uma **expressão de guarda**, ou simplesmente uma **guarda**. Uma guarda é uma expressão lógica que indica se a seta pode ou não ser usada. A partir da versão 2.0, setas do DA não podem mais indicar eventos ou ações. As expressões de guarda são particularmente usadas junto aos nós de decisão.

Os fluxos apresentam algumas outras características que apresentaremos mais adiante no texto.

¹⁰ Funcionando na prática como uma sincronização (join) implícita.

¹¹ Funcionando na prática como uma paralelização (fork) implícita.

I.6.3 Controle

Para caracterizar o comportamento de um diagrama de atividades é necessário controlar a seqüência de execução das ações de alguma forma. O DA fornece dois mecanismos básicos para isso: a escolha de um entre vários caminhos possíveis e o seguimento de vários caminhos possíveis simultaneamente (paralelismo). Esses comportamentos, e outros necessários a uma melhor descrição de uma atividade, são descritos por nós de controle.

Os nós de controle podem ser de sete tipos, conforme a tabela a seguir:






Tipo do Nó	Gráfico correspondente
Início	
Fim de Fluxo	
Fim de Atividade	
Fork, ou início de paralelismo	
Join, ou sincronismo	
Decisão	
Merge, unificação	

Tabela 2. Tipos de nós de controle e seus gráficos

Um DA se inicia no “nó de **início**”. Podem existir vários desses nós, e todos são ativados simultaneamente. Um nó de início é indicado por um círculo preto (Tabela 3). De acordo com o padrão 2.1, um token é colocado em cada nó de início quando a atividade inicia.

Os nós de “**fim de fluxo**” e “**fim de atividade**” servem, respectivamente, para indicar o fim de um fluxo de controle ou o fim de toda a atividade. Quando um token chega a um nó de fim de fluxo, o token é destruído e o caminho seguido por ele acaba. Quando um token chega a um nó de fim de atividade todos os tokens são destruídos e atividade é encerrada.

É possível indicar a possibilidade de tomar caminhos diferentes em função de uma **decisão**. Isso é indicado por um losango, sendo que cada caminho possível deve receber como rótulo uma expressão de guarda que indique a decisão. A partir da versão 2.0 o nó de decisão passou a ser obrigatório, não sendo possível representar uma decisão diretamente a partir de uma ação. Pela semântica do DA, o nó de decisão recebe um token e o envia pela aresta que contém uma expressão de guarda que seja verdadeira. Segundo o padrão, apenas uma expressão de guarda será escolhida. Caso várias expressões de guarda sejam verdadeiras, não existe uma semântica definida.

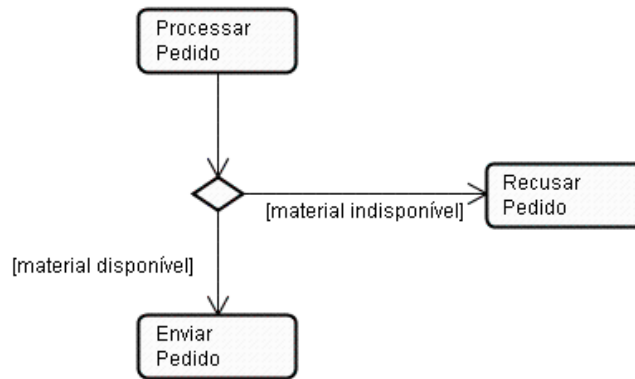


Figura 37. Fragmento de diagrama de atividade mostrando uma decisão.

Os nós “fork” e “join” permitem a execução paralela de ações, e sua possível sincronização (*fork* e *join*) futura. É importante lembrar que uma sincronização (*join*) sempre espera por todas as arestas terem tokens disponíveis, e as consome simultaneamente (o que evita *deadlocks*).

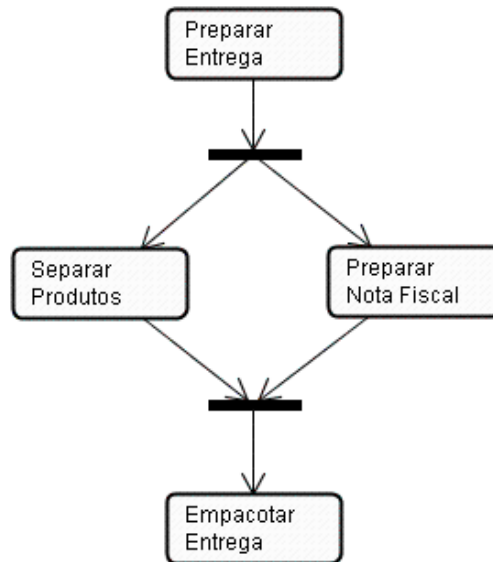


Figura 38. Exemplo de caminhos em paralelo e sincronização.
“Empacotar entrega” só poderá executar após “Separar produtos” e “Preparar Nota Fiscal” completarem.

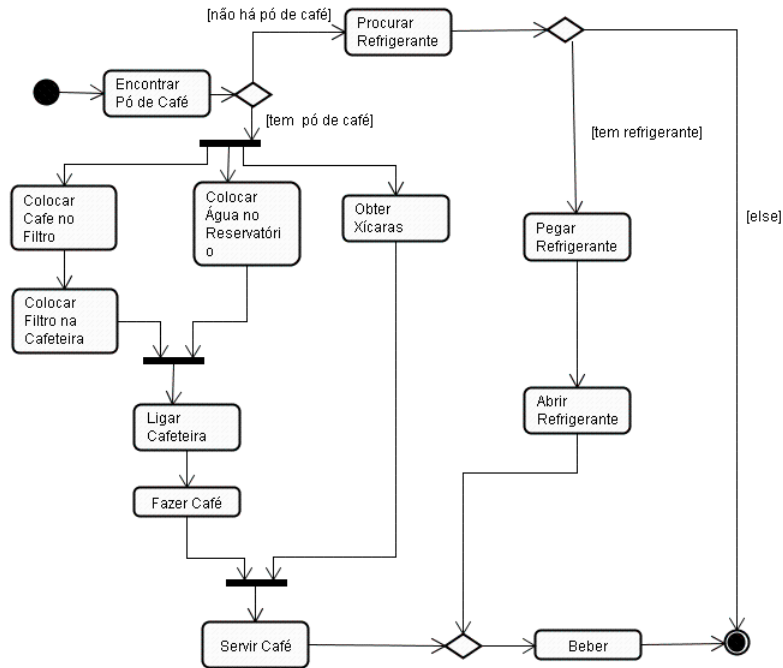


Figura 39. Exemplo de diagrama de atividades descrevendo o processo de tomar um café ou um refresco em um intervalo de trabalho, segundo a versão 2.1 do padrão UML. Note a necessidade de um nó de unificação antes da ação “Beber”. Caso os fluxos que entram nesse nó entrassem direto na ação “Beber”, essa nunca seria executada (*deadlock*), pois seria necessário que os dois fluxos contivessem tokens, o que nunca poderá acontecer, já que apenas um dos caminhos será escolhido.

I.6.4 Interrupções, Exceções e Sinais

Em alguns sistemas é importante modelar a capacidade de receber e emitir sinais. Por exemplo, durante o processamento de um pedido, a empresa pode receber um sinal que indica o seu cancelamento.

Outra capacidade importante é a modelagem de exceções. Na figura a seguir mostramos uma exceção que interrompe um processo ao recebimento de um sinal de cancelamento.

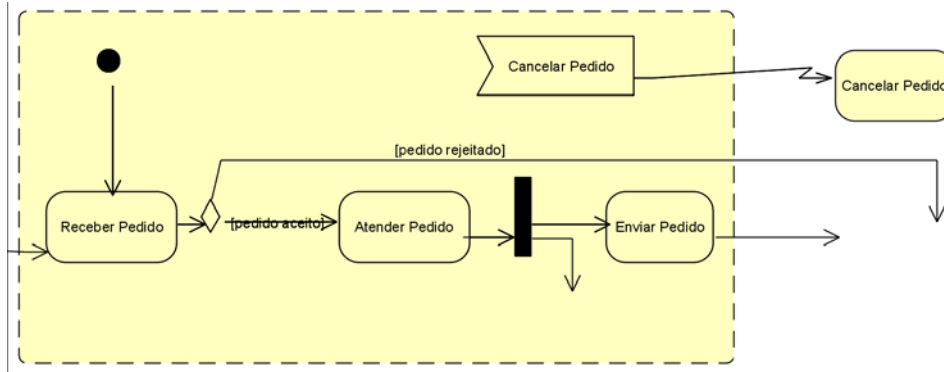


Figura 40. O detalhe de diagrama (UML 2.0) acima apresenta uma área onde pode acontecer uma interrupção, indicada pela ação de recepção de eventos “Cancelar Pedido”. Caso isso aconteça, será lançada uma exceção que invocará a ação “Cancelar Pedido”. A exceção é indicada pela seta com formato de raio. Uma exceção interrompe todas as ações dentro da área de interrupção.



Figura 41. Símbolos para envio de sinais e recepção de eventos (sinais) em UML 1.5

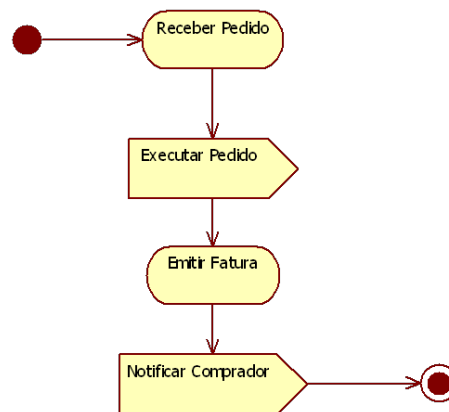


Figura 42. Exemplo de envio de sinais (UML 2.0).

I.6.5 Diagramas de Raias

Qualquer diagrama que passe a idéia de um fluxo de execução, como um fluxograma ou um diagrama de atividades, pode ser construído dentro de um espaço que modele alguma partição dessas atividades. Normalmente o que se faz é utilizar colunas (ou linhas) para modelar os agentes que realizam a atividade específica, dando a impressão de “raias de natação” ao diagrama (*swimlanes*).

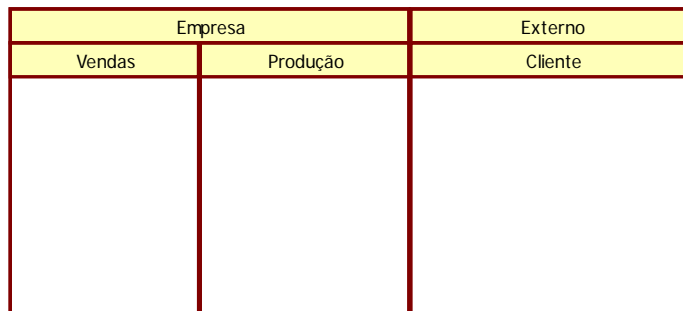


Figura 43. Exemplo de raias com partição hierárquica.



Figura 44. Exemplo de raias com partições múltiplas.

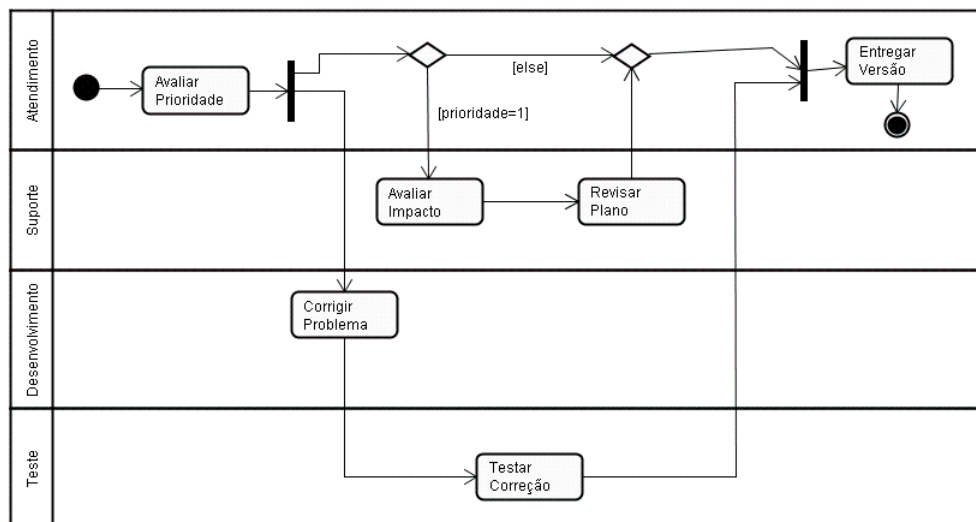


Figura 45. Exemplo de diagrama de atividades com raias

A seguir apresentamos dois diagramas interessantes se comparados. Ambos descrevem um mesmo comportamento final, porém com implementações diferentes. Deixamos ao leitor o desafio de ver como as seqüências de ações realizadas serão sempre as mesmas.

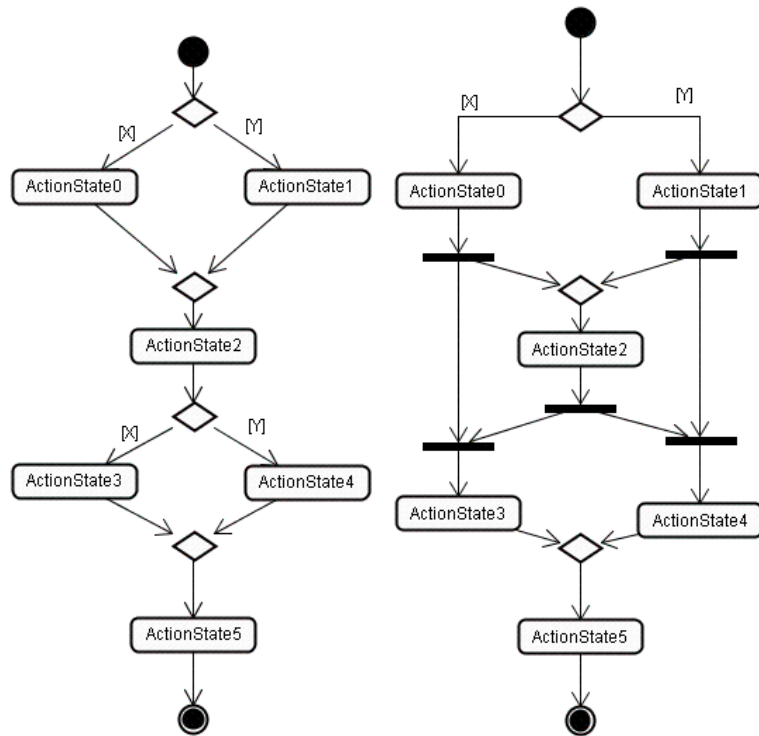


Figura 46 Os dois Diagramas de Atividade acima apresentarão a mesma seqüência de ações para uma mesma entrada, porém utilizando implementações diferentes.

I.6.6 Exemplo

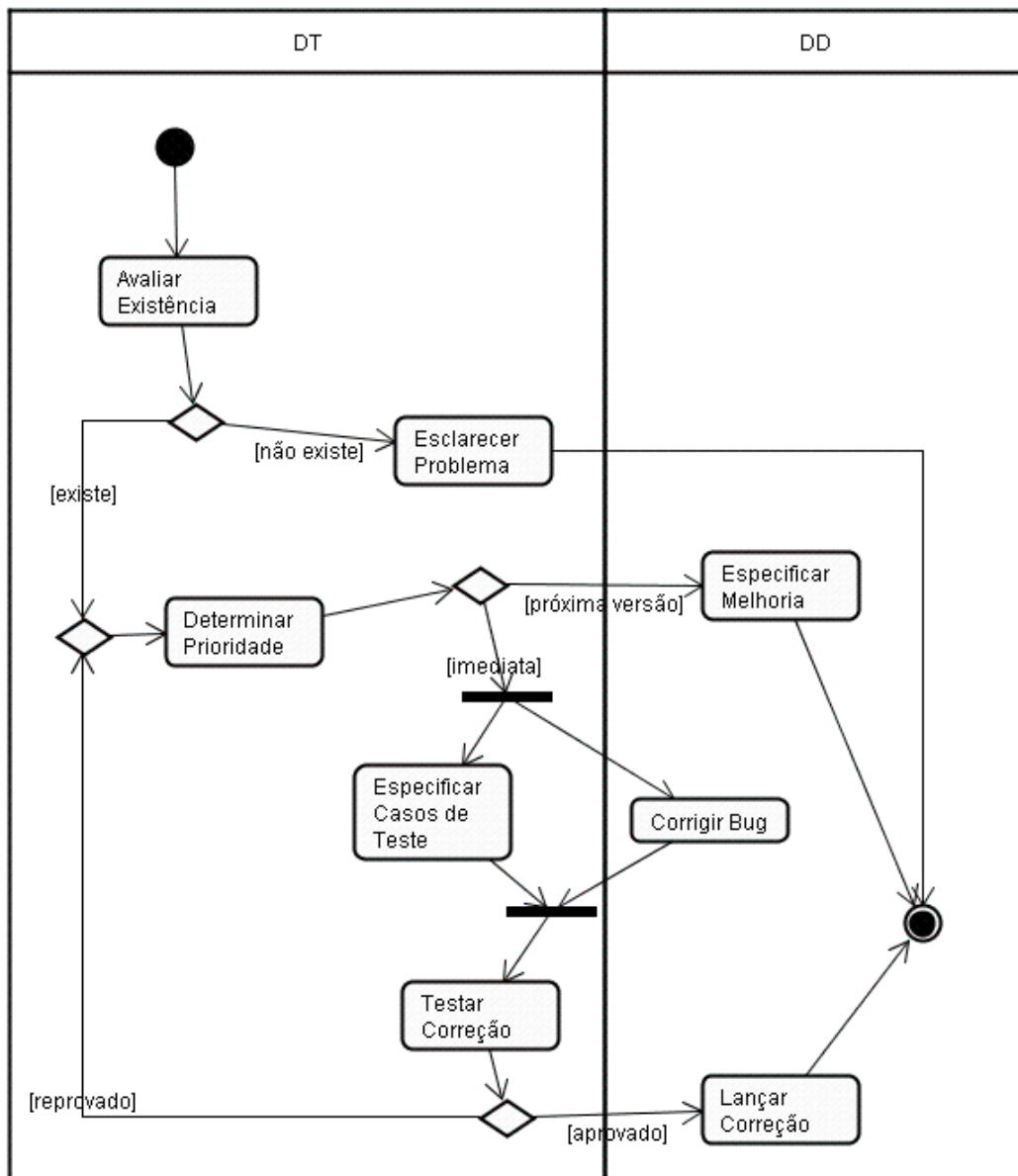


Figura 47. Diagrama de Atividades para o texto da Figura I.4.4

I.7 Regras de Negócio

“Uma regra de negócio é uma sentença que define algum aspecto do negócio. Tem o objetivo de afirmar a estrutura do negócio ou de controlar ou influenciar o comportamento do mesmo. Regras de negócio são atômicas, isto é, não podem ser quebradas”[B26].

Regras de negócio são muito estudadas hoje em dia. A maioria dos importantes autores americanos se reuniu em um grupo conhecido com “Business Rule Group”¹², que produziu alguns documentos [B26;B27] que forneceram a estrutura básica que estamos apresentando aqui. Todas as definições são ou versões dos originais em inglês.

Uma regra de negócio pode ser de três categorias.

- **Declarações Estruturais**, um conceito ou a declaração de um fato que expressa algum aspecto da estrutura da organização. Podem ser termos simples ou fatos relacionando esses termos. Normalmente são descritas por meio de um diagrama de entidades e relacionamentos¹³.
- **Declarações de Ação**, que descrevem aspectos dinâmicos do negócio, sendo uma expressão de uma restrição ou de uma condição que controla as ações de uma organização.
- **Derivações**, a declaração de um conhecimento que é derivado a partir de outro.

1.7.1 Declarações Estruturais

Inclui os termos de negócios e fatos relacionando termos.

1.7.1.1 Definição de Termos de Negócios

“O elemento básico de uma regra de negócio é a linguagem utilizada para expressá-la. A definição das palavras utilizadas nessa linguagem descreve como as pessoas pensam e falam” [B26].

Normalmente um projeto só começa realmente a progredir quando a equipe de desenvolvimento compreende o discurso dos clientes. Para isso, nas primeiras entrevistas ou reuniões, é feito um esforço para levantar um glossário de termos, e, mais tarde, muitos desses termos podem aparecer no Modelo Conceitual de Dados do sistema.

Um exemplo tirado de um sistema governamental:

- **Contribuinte**: é a pessoa física ou jurídica responsável pelo pagamento de um imposto.

A definição de um termo muitas vezes envolve uma relação com outros termos, como no exemplo acima.

Um termo pode ser um **tipo** (uma classe) ou um **literal** (uma instância). No caso de regras de negócio costumamos trabalhar principalmente com tipos. Um tipo especial de tipo é um **sensor**, que representa algo que detecta e reporta valores que mudam constantemente no ambiente (mundo externo). Existe um sensor especial, o **relógio**, que relata a passagem do tempo, cujo valor é sempre o instante corrente (data e hora corrente).

Os termos definidos são reunidos em um glossário e em um dicionário de dado.

¹² Originalmente “The Guide Business Rules Project”

¹³ Não sendo necessariamente igual ao modelo conceitual de dados, mas podendo fornecer elementos para esse.

1.7.1.2 Declaração de Fatos

A partir dos termos, devemos construir sentenças que descrevam o negócio a partir das relações entre termos e da estrutura criada por essas relações. Normalmente esses fatos são caracterizados nas entrevistas e reuniões, a partir de declarações do cliente de como o negócio funciona.

Fatos relacionando termos são bastante fáceis de serem encontrados. Muitas vezes encontramos primeiro um fato e depois analisamos o significado dos seus termos.

As declarações estruturais podem declarar atributos, generalizações ou participações. Uma participação, por sua vez, pode ser um papel, uma agregação ou uma associação simples.

Exemplos:

Tipo de Fato	Exemplo
Atributo	DRE é um atributo de aluno
Generalização	Aluno de graduação é um tipo de Aluno
Papel	Um aluno pode ser representante de classe
Agregação	Uma turma precisa ter alunos
Associação simples	Um aluno deve fazer provas

Tabela 3. Tipos de Fatos e exemplos

1.7.1.3 Declarações estruturais e o modelo de dados

Termos e fatos estabelecem a estrutura de um negócio. Esta estrutura é normalmente descrita em um modelo de dados. Assim, a maior parte do trabalho inicial com as regras de negócio estruturais pode ser descrita por meio de um modelo de dados conceitual, tema tratado no próximo capítulo.

1.7.1.4 Exemplos

A seguir damos alguns exemplos de declarações estruturais, para um sistema acadêmico imaginário:

- Alunos são pessoas matriculadas em um curso
- Um Aluno de Graduação é um tipo de Aluno
- Um Aluno de Pós-Graduação é um tipo de Aluno
- Um Curso é construído por um conjunto de Cadeiras
- Uma Cadeira é ministrada por um Professor
- Um Professor é um tipo de Funcionário
- Durante um Período, Alunos podem se matricular em Cadeiras

1.7.2 Declarações de Ações (ou Restrições)

Representam as restrições ou condições que as instituições colocam em suas ações. Podem aparecer por força de lei, prática de mercado, de decisão da própria empresa ou ainda outros motivos.

Exemplos:

- Um aluno deve ter um DRE
- Um aluno não pode se registrar em dois cursos que acontecem no mesmo horário

Uma Declaração de Ação pode ser uma condição, uma restrição de integridade ou uma autorização. Uma condição diz que se alguma coisa for verdade, então outra regra deve ser aplicada (se - então). Uma restrição de integridade é uma declaração que deve sempre ser verdade. Uma autorização dá uma prerrogativa ou privilégio a um termo, normalmente permitindo que uma pessoa possa realizar uma ação.

Declarações de ações podem ser de uma variedade de outros tipos. Recomendamos a leitura de

Uma declaração de ação pode dizer que precisa acontecer (controle) ou o que pode acontecer (influência).

1.7.3 Derivações

São regras que mostram como transformar conhecimento em uma forma para outro tipo de conhecimento, possivelmente em outra forma, incluindo leis da natureza[B27]. Geralmente são regras ou procedimentos de cálculo ou manipulação de dados.

Exemplo:

- O valor a ser pago do imposto predial é 3% do valor venal do imóvel.
- A lista de devedores inclui todos os devedores a mais de dois anos.

1.7.4 Regras e Eventos

Cada regra de negócio tem a possibilidade de ser violada em um ou mais eventos do sistema.

A questão do que é um evento será respondida mais a frente nesse texto, porém, no momento, basta entendermos que um evento é algo que exige que alteremos os fatos conhecidos pelo sistema (ou seja, um evento exige a alteração de algum dado na base de dados) ou que consultemos esses fatos a fim de informá-los, de alguma forma processada, ao usuário.

Analisando uma regra podemos ver que tipos de eventos são prováveis de necessitarem de alguma atenção do sistema. Por exemplo, suponha que um sistema de vendas para uma empresa possua as regras[B28]:

- Um aluno cliente solicita um produto
- Um vendedor é designado para atender um cliente permanentemente

A descrição acima faz com que nos perguntemos:

- O que acontece quando um cliente faz seu primeiro pedido(e não tem ainda um vendedor associado)?
- O que acontece quando um vendedor se desliga da empresa.

Em todo evento, um conjunto de regras deve ser ativado e cada erro deve ser reportado ao usuário.

I.7.5 Descrevendo Regras de Negócio

Existem muitas formas de descrever regras de negócio, de acordo com o grau de formalismo e a necessidade de execução (ou compreensão por serem humanos) que desejamos. A tabela a seguir define quatro formas básicas de descrição.

Fragmento de conversação de negócios	Versão em linguagem natural	Versão em uma linguagem de especificação de regras	Versão em uma linguagem de implementação de regras
Pode não ser relevante	Relevante	Relevante	Executável
Pode não ser atômica	Atômica	Atômica	Pode ser procedural
Pode não ser declarativa	Declarativa	Declarativa	
Pode não ser precisa	Não totalmente precisa	Precisa	
Pode ser incompleta	Pode ser incompleta	Completa	
Pode não ser confiável	Confiável	Confiável	
Pode não ser autêntica	Autêntica	Autêntica	
Pode ser redundante	Pode ser redundante	Única	
Pode ser inconsistente	Pode ser inconsistente	Consistente	

Tabela 4. Formas de descrever regras de negócio e suas características adaptado de [B29].

Halle[B29] propõe uma classificação e um conjunto de *templates* que pode ser utilizado para descrever regras de negócio (aqui adaptado para português), apresentado na tabela a seguir.

Classificação	Descrição Detalhada	Template
Termo	Nome ou sentença com uma definição acordada que pode ser: <ul style="list-style-type: none"> • Conceito, classe, entidade, • Propriedade, detalhe, atributo, • Valor, • Conjunto de valores 	<termo> é definido como <texto>
Fato	Sentença que relaciona termos em observações relevantes ao negócio <ul style="list-style-type: none"> • Relacionamentos entre entidades • Relacionamentos entre entidades e atributos • Relacionamentos de herança 	<termo1> é um <termo2> <termo1> <verbo> <termo2> <termo1> é composto de <termo2> <termo1> é um papel de <termo2> <termo1> tem a propriedade <termo2>
Computação	Sentença fornecendo um algoritmo para calcular o valor de um termo	<termo> é calculado como <formula>
Restrição obrigatória	Sentença que indica restrições que devem ser verdade em informações fornecidas ao sistema (input)	<termo1> deve ter <no mínimo, no máximo, exatamente n> <termo2> <termo1> deve ser <comparação> <termo2> ou <valor> ou <lista de valores> <termo> deve ser um de <lista de valores> <termo> não pode star em <lista de valores> se <regra> então <restrição>
Guideline	Sentença que indica restrições que deveriam ser verdade em informações fornecidas ao sistema (input)	<termo1> deveria ter <no mínimo, no máximo, exatamente n> <termo2> <termo1> deveria ser <comparação> <termo2> ou <valor> ou <lista de valores> <termo> deveria ser um de <lista de valores> <termo> não poderia star em <lista de valores> se <regra> então <restrição>
Conhecimento inferido	Sentenças que expressam circunstâncias que levam a novas informações	se <termo1> <operador> <termo2, valor, ou lista de valores> ... então <termo3> <operador> <termo4> Onde operador pode ser: =, <, =<, >=, <,>, contém, não contém, tem no máximo, tem no mínimo, tem exatamente, etc.
Iniciador de ação	Sentença expressando condições que levam ao início de uma ação	se <termo1> <operador> <termo2, valor, ou lista de valores> ... então <ação>

Tabela 5. Classificação para regras, definição e templates adaptada de (Halle, 2002).

Obviamente, como fizemos aqui, a forma textual é bastante útil. Uma maneira bastante comum é utilizar Diagramas de Entidades e Relacionamentos[B27].

Deve ficar claro, porém, que a utilização de DER para definir regras de negócio não é igual à utilização de DER para definir o modelo conceitual de dados de um sistema.

DERs, porém, não representam todas as características das regras de negócio. Ross [B30], propôs uma notação mais complexa, incluindo (muitos) novos símbolos em um DER. Essa notação, porém, foge do escopo desse texto.

Vejamos a descrição de duas regras para uma livraria (**Figura 71**):

- Um cliente deve pedir livros.
- Livros são entregues por distribuidoras.

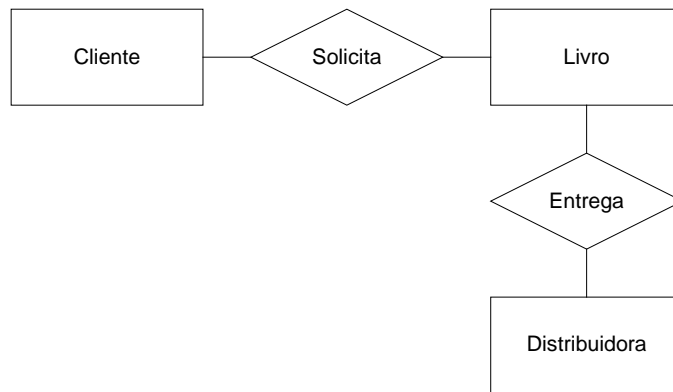


Figura 48. Exemplo de duas regras de negócio descritas utilizando uma notação de DER simplificada.

Alguns autores fazem uma descrição regra a regra, como na **Figura 72**.

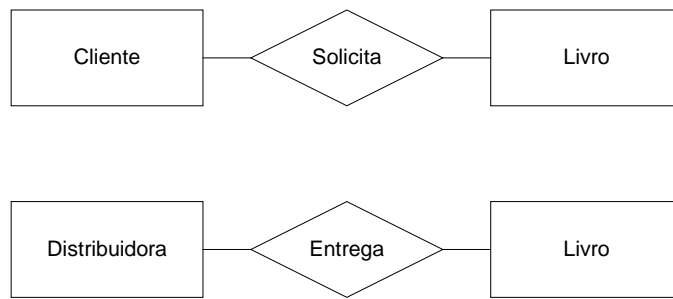


Figura 49. Descrição das regras uma a uma.

As regras de negócio serão utilizadas nos próximos passos para definir modelos mais formais do sistema. Mas podemos, por exemplo, definir a cardinalidade dos termos em cada relacionamento descrito, como na **Figura 73** e no texto a seguir:

- Um cliente pede um ou mais livros,
- Livros podem ser pedidos por nenhum, um ou mais clientes,
- Uma distribuidora entrega um ou mais livros,
- Um livro é entregue por apenas uma distribuidora.

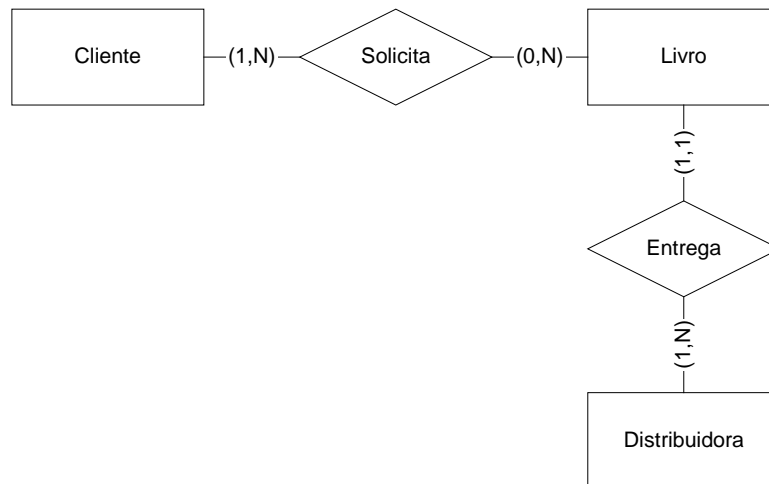


Figura 50. Regras com cardinalidades.

É interessante notar que regras de negócio devem ser feitas de forma declarativa, **não** indicando como vão ser implementadas, onde vão funcionar, quem será responsável ou quando devem ser testadas. Desse jeito as regras serão flexíveis o suficiente para ser utilizadas na modelagem do negócio. Como devem ser declarativas, elas também não devem ser escritas como um procedimento.

I.8 Outras Ferramentas de Modelagem

O uso de tabelas permite relacionar informações levantadas separadamente. Elas são bastante importantes nas conferências dos modelos.

I.8.1 Responsáveis por decisão (Processo x Organização)

Essa tabela associa as pessoas da organização, que foram representadas no organograma, com as funções da empresa, que podem ter sido representadas anteriormente de várias formas, como o diagrama funcional.

O objetivo da tabela é determinar o envolvimento dessas pessoas com as decisões relativas às funções da empresa. Esse envolvimento é determinado em três níveis: principal responsável pela decisão, grande envolvimento com a decisão e, finalmente, algum envolvimento com a decisão.

	Principal responsável pela decisão
	Grande envolvimento com a decisão
	Algum envolvimento com a decisão

Tabela 6. Tipos de envolvimento com a decisão

		Processos							
		Vendas				Produção			
		Gerência do Território	Venda	Administração	Pedidos de Serviço	Agendamento	Planejamento	Fornecimento	Operação
Organização	VP de Vendas								
	Gerente de pedidos								
	Gerente de vendas								
	VP Engenharia								
	VP Produção								
	Diretor da Fábrica								

Tabela 7 Exemplo de uma tabela Processo x Organização

I.8.2 Dados x Processos (CRUD)

Esta tabela é uma das mais interessantes e relaciona as entidades (inicialmente do modelo de conceitual de dados) com os processos que as utilizam, indicando pelas letras CRUD se o processo **C**ria, lê (**R**ead), altera (**U**psert) ou apaga (**D**elete) a entidade.

Uma de suas características principais é que pode ser construída com vários pares linha x coluna, dependendo do tipo de abstração que está sendo usado. No nosso caso usaremos principalmente no mapeamento de ações (CRUD) entre eventos essenciais e entidades (os dois temas serão tratados mais tardes), mas podem ser usados em modelos de negócio (processo x dados utilizados) ou orientados a objetos (casos de uso x objetos). Na prática, é uma tabela de trabalho muito interessante, que permite a visualização rápida da relação entre funcionalidade e dados.

Como usaremos esta tabela para relacionar eventos e entidades, deixaremos esse tema para ser tratado mais tarde, na unificação dos modelos funcional e de dados.

	Dados					
	Dado 1	Dado 2	Dado 3	Dado 4	Dado 5	Dado 6
Processos						
Processo A	CRUD	R	R			
Processo B		CRUD	R		R	
Processo C		RUD	C			
Processo D				C	R	
Processo E				R	C	
Processo F		R			RU	C

Figura 51. Exemplo simplificado de uma Matriz CRUD

I.8.3 Corrente/Planejado

Esta tabela indica que processos são correntes e que processos estão apenas planejados. Pode ser feita tanto a nível de negócios quanto a nível de sistemas. No primeiro caso estamos interessados nos processos correntes ou implementados no dia a dia da empresa. No segundo caso estamos interessados em processos automatizados.

Corrente x Planejado	
Função	Situação
Cadastro de Cliente	Corrente
Cadastro de Fornecedor	Corrente
Cadastro de Pedidos	Planejado
Criação de Pedidos	Planejado

Tabela 8. Exemplo de tabela Corrente x Planejado

I.9 Resumo da Modelagem de Negócio

Não há uma forma correta única de se fazer a modelagem de negócio. Recomendamos que sempre seja levantado o organograma da empresa. A seguir, os modelos podem ser escolhidos e levantados de forma adequada a um processo ou projeto específico.

No capítulo vimos, em ordem decrescente de abstração, métodos que podem ser utilizados tanto de forma isolada como unificada. Deve ser levado em consideração que o método IDEF0 faz parte de uma família de métodos que podem ser utilizados juntos (no caso de modelagem de processo existe o método IDEF3, que não usaremos nesse texto por considerarmos o EPC um método superior).

I.10 Exercícios

I.10.1 Projeto 1: Livraria ABC

Faça todos os modelos de negócios descritos nesse capítulo para a Livraria ABC, da forma mais completa possível.