



Especificação do algoritmo de partição de um BPMN

Link Consulting – Tecnologias de Informação, S.A.

Janeiro 2024





Índice

1	Descrição detalhada do algoritmo	4
1.1	Restrições do algoritmo	4
1.2	Granularidade do algoritmo.....	4
1.3	Parte 1 do algoritmo: Descoberta e Partição.....	5
1.4	Parte 2 do Algoritmo: Agregação.....	9



Índice de Figuras

Figura 1: Processo com níveis de granularidade	5
Figura 2 Diagrama da partição	6
Figura 3 Diagrama de processamento de um elemento.....	7
Figura 4 Diagrama de processamento de uma gateway.....	8
Figura 5 Processo com fluxo adjacente	10
Figura 6 Processo com fluxo obrigatório.....	10
Figura 7 Resultado da agregação de ramos com fluxo obrigatório	10
Figura 8 Processo com fluxo alternativo.....	11
Figura 9 Resultado da agregação de ramos com fluxo alternativo	11
Figura 10 Processo com fluxo opcional	11
Figura 11 Resultado da agregação de ramos com fluxo opcional.....	12
Figura 12 Exemplo completo	13



1 Descrição detalhada do algoritmo

O Algoritmo de Particionamento de BPMN está dividido em três etapas principais: descoberta, partição e agregação. O objetivo é processar um modelo BPMN (Business Process Model and Notation) que contém atividades com a especificação do sistema alvo e dividi-lo em diferentes modelos BPMN funcionais, que representam os processos presentes em cada sistema distinto.

Dados de entrada:

- **Modelo BPMN:** Representado no formato XML.
- **Ficheiro de parâmetros:** Contém todas as informações necessárias para identificar e associar cada atividade ao seu sistema-alvo.

Resultados:

- **Conjunto de diagramas BPMN:** Representados por ficheiros XML, cada um correspondendo a um sistema específico. representados por ficheiros XML.

1.1 Restrições do algoritmo

O algoritmo requer um modelo BPMN funcional, sem feedback ou ciclos. Cada tarefa deve conter informações explícitas sobre o sistema no qual irá atuar. Cada pool deve conter apenas um evento inicial. Não pode existir uma ligação direta entre duas portas de decisão (*gateways*).

1.2 Granularidade do algoritmo

O algoritmo apresentado realizará a partição em várias pequenas partes. Em detalhe:

Processo:

- O processo é o nível mais alto de granularidade, deve ser funcional e pode ter múltiplos sistemas anotados. Um processo pode conter múltiplos subprocessos, um para cada sistema/motor considerado.

Subprocesso:

- Um subprocesso é uma combinação de um ou mais ramos, todos pertencentes ao mesmo sistema. Após a execução do algoritmo, haverá um subprocesso para cada sistema.



Ramo:

- Um ramo é composto, por uma ou mais elementos de fluxo, todas associadas ao mesmo sistema juntamente com os fluxos de sequência, que representam as transições entre elementos.

Elemento de fluxo:

- Um elemento de fluxo é uma atividade atômica dentro do fluxo de um processo.

Fluxo de sequência:

- O fluxo de sequência representa uma ligação entre dois elementos no processo, cada fluxo de sequência contém o identificador do elemento de origem e do elemento de destino.

Podemos observar um exemplo anotado da granularidade na Figura 1. O subprocesso não está presente, pois é criado apenas após a agregação.

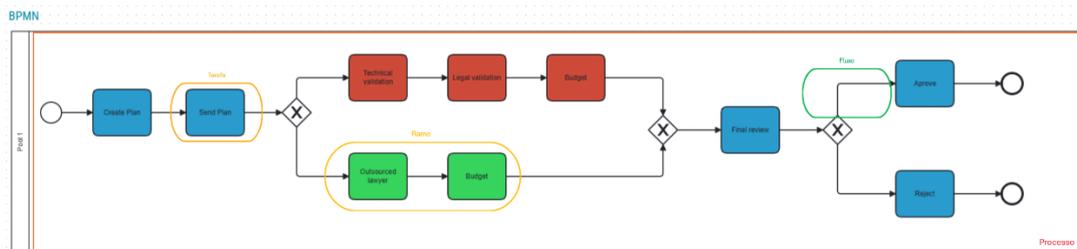


Figura 1: Processo com níveis de granularidade

1.3 Parte 1 do algoritmo: Descoberta e Partição

Dados de entrada:

- Modelo BPMN:** Representado no formato XML.
- Ficheiro de parâmetros:** Contém todas as informações necessárias para identificar e associar cada atividade ao seu sistema-alvo.

Resultados:

- Mapa de ramos:** Um mapa onde a chave é o identificador único de cada ramo e o valor é o próprio ramo.

A fase de descoberta tem como objetivo identificar todos os elementos e fluxos de um BPMN. Esta informação será organizada e armazenada de acordo com o



processo na qual o elemento está presente. Para alcançar este resultado, o ficheiro XML recebido é processado, resultando numa estrutura composta por uma lista de processos. Cada processo contém uma lista de todos os elementos e fluxos de sequência associadas, esta fase é também encarregue de associar a cada elemento possíveis anotações ou objetos de dados presentes.

A fase de Descoberta, para cada processo, produz os seguintes resultados:

- **Mapa de elementos:** Um mapa que associa cada elemento ao seu identificador único, onde a chave é o identificador do elemento e o valor é o próprio elemento.
- **Mapa de fluxos de sequência:** Um mapa que organiza os fluxos de sequência, onde a chave é o identificador do elemento de origem e o valor é uma lista que contém todos os fluxos para os quais este elemento pode direcionar.

A fase de partição é replicada para todos os processos identificados na fase de descoberta. Este passo começa pela identificação do elemento de início do processo para realizar a inicialização onde é criado o primeiro ramo. Depois da inicialização, o algoritmo percorre todos os fluxos possíveis, através de um mecanismo de controle garantimos que cada elemento é percorrido apenas uma vez (figura 2).

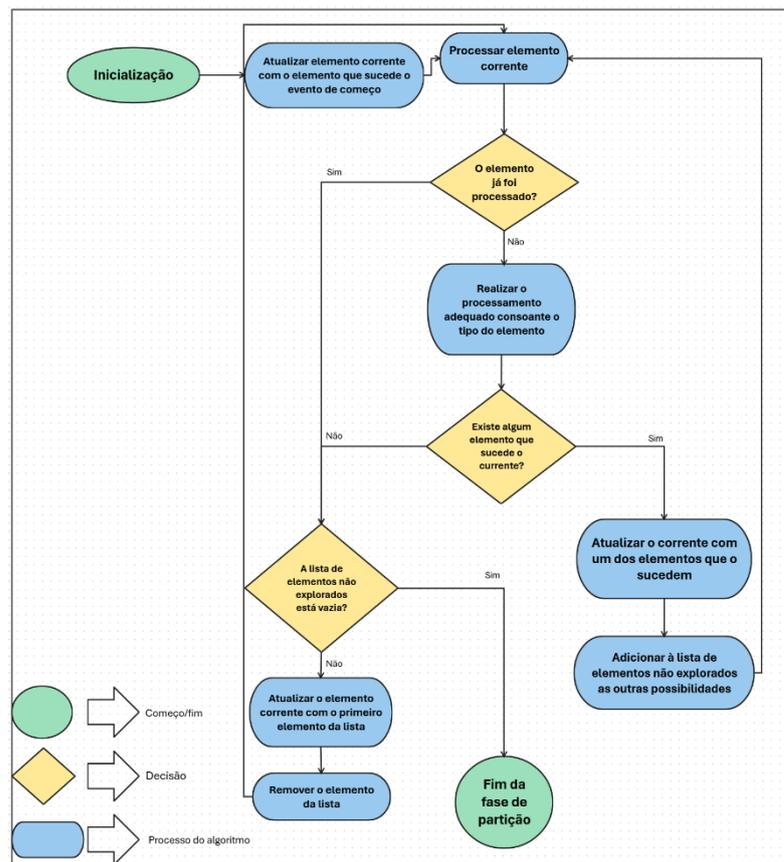


Figura 2 Diagrama da partição

Durante a fase de partição um novo ramo é criado quando encontramos:

- Um sistema diferente do anterior.



- Uma gateway.
- Um evento de fim.

O processamento de cada elemento é, na maioria dos casos, semelhante. Primeiro, verifica-se o sistema associado ao elemento e, caso este seja diferente, é criado um ramo (figura 3). No entanto, elementos do tipo evento de fim exigem o reinício do sistema após o seu processamento.

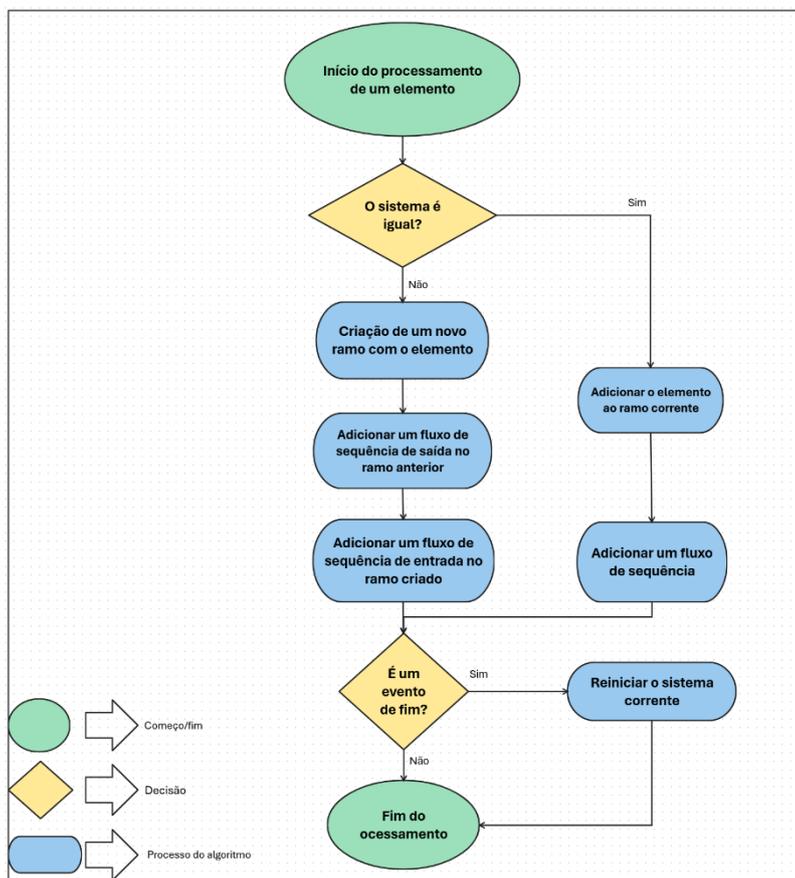


Figura 3 Diagrama de processamento de um elemento

Elementos do tipo *gateway* representam um caso particular, pois são pontos de decisão que podem ocorrer em sistemas diferentes, dependendo do percurso do fluxo durante a execução. Como estes elementos não possuem um sistema especificado pelo utilizador, não é realizada a verificação de sistema, o elemento é adicionado ao ramo corrente e em seguida o sistema é reiniciado (figura 4).

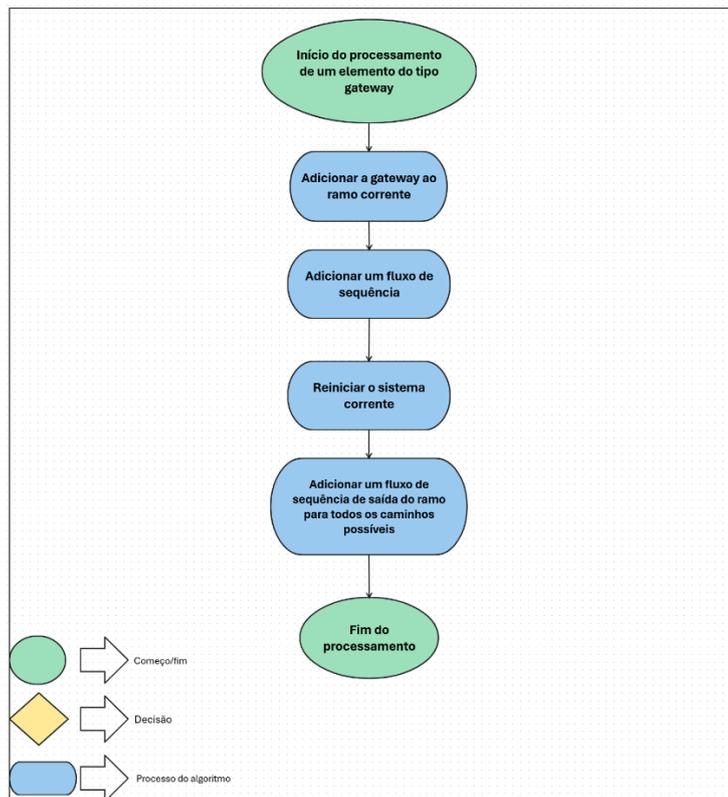


Figura 4 Diagrama de processamento de uma gateway

Para que o processamento dos diferentes elementos funcione corretamente, é necessário manter registo de várias informações, incluindo o ramo corrente, o sistema corrente, o último elemento processado, os elementos já processados e os caminhos ainda não explorados. Além disso, é fundamental manter registo de todos os ramos construídos.

Para alcançar este objetivo, utilizam-se as seguintes estruturas de dados:

- **Mapa de ramos:** Um mapa onde a chave é o identificador único de cada ramo e o valor é o próprio ramo.
- **Lista de elementos processados:** Uma lista que contém os identificadores de todos os elementos já processados.
- **Lista de elementos não explorados:** Uma lista com os identificadores de todos os elementos que pertencem a caminhos não explorados.
- **Identificador do último elemento explorado:** Uma variável para armazenar o identificador do elemento mais recentemente processado.
- **Identificador do ramo atual:** Uma variável que indica o identificador do ramo ativo.
- **Sistema atual:** Uma variável que regista o sistema ativo.

O resultado do algoritmo após o processo de descoberta deve ser composto por vários ramos, onde cada ramo é uma coleção de um ou mais elementos que partilham o mesmo sistema. Sempre que um novo ramo é criado, é necessário criar



um fluxo de sequência. Este fluxo é estabelecido com base na última tarefa do ramo inicial e na primeira tarefa do novo ramo.

Os fluxos de sequência entre ramos representam as mensagens enviadas entre sistemas para assegurar a sequência do processo. Por conseguinte, os objetos de dados necessários para o novo sistema precisam de ser transmitidos através destas tarefas de ligação.

Para atingir este objetivo, quando um novo sistema é identificado durante o ciclo de ramos, a ligação é criada e todos os objetos de dados presentes na lista de dados do ramo atual serão considerados como "entradas" para o novo sistema identificado. No sistema novo é identificado a que objetos de dados acede e armazenado, para no final ser aplicada uma filtragem nos objetos de dados presentes nas ligações com apenas os que vão ser efetivamente acedidos.

1.4 Parte 2 do Algoritmo: Agregação

Após dividir o processo em ramos, é necessário agregá-los num diagrama BPMN funcional e coeso. O objetivo desta etapa é reconstruir um BPMN estruturado, para cada sistema diferente

Entrada:

- Mapa de ramos: Um mapa onde a chave é o identificador único de cada ramo e o valor é o próprio ramo.

Saída:

- Um diagrama BPMN funcional e coeso, por cada sistema diferente.

A fase de agregação funciona com base nos fluxos de entrada de cada ramo, que representam as conexões provenientes de outros ramos. Cada fluxo de entrada está associado a um objeto de saída de um ramo anterior. O processo consiste em:

- **Percorrer os ramos:** A agregação percorre todos os ramos presentes no mapa, onde cada um é avaliado individualmente.
- **Identificar fluxos de entrada:** Para cada ramo é analisado os fluxos de sequência de entrada.
- **Ramo de origem:** A partir do fluxo de entrada, identifica-se o ramo anterior.
- **Verificação:** É verificado que o ramo anterior é do mesmo sistema que o atual e que o anterior ainda não foi agregado e, portanto, vai ser removido.
- **Fusão:** É identificado o tipo de fusão que vai ser aplicado, e após efetuada o ramo que foi não contém o resultado da fusão é assinalado para ser removido após o algoritmo
- **Avançar para o próximo:** Após o processamento de todos os fluxos de entrada de um ramo, o algoritmo avança para o próximo ramo no mapa.



A fusão de ramos representa a convergência de fluxos de sequência. Existem diversos cenários possíveis. Os principais cenários são:

Ramos Adjacentes (Fluxo contínuo/Sequencial): Neste cenário, a saída de uma atividade ou evento conecta-se diretamente à entrada da próxima atividade ou evento (figura 5). É uma transição linear e direta.

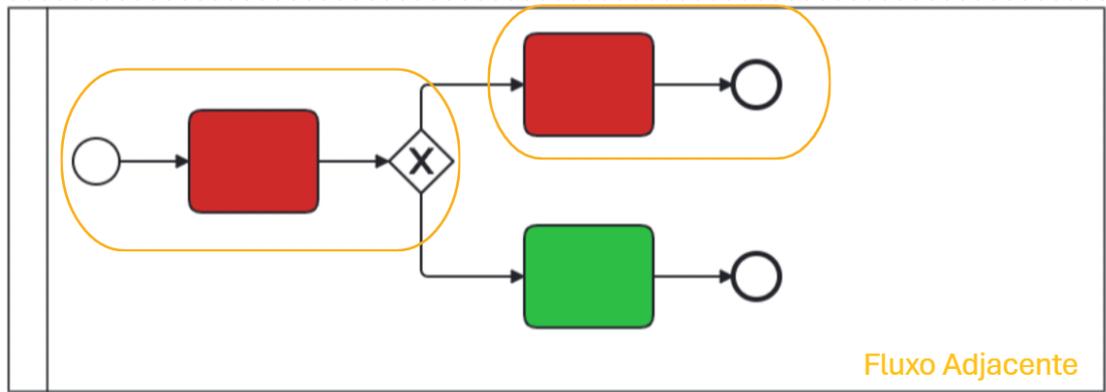


Figura 5 Processo com fluxo adjacente

Ramos com fluxo obrigatório e não adjacente: Neste cenário, um ramo deve obrigatoriamente terminar antes que o outro ramo inicie, mas não há conexão direta entre os dois (figura 6).

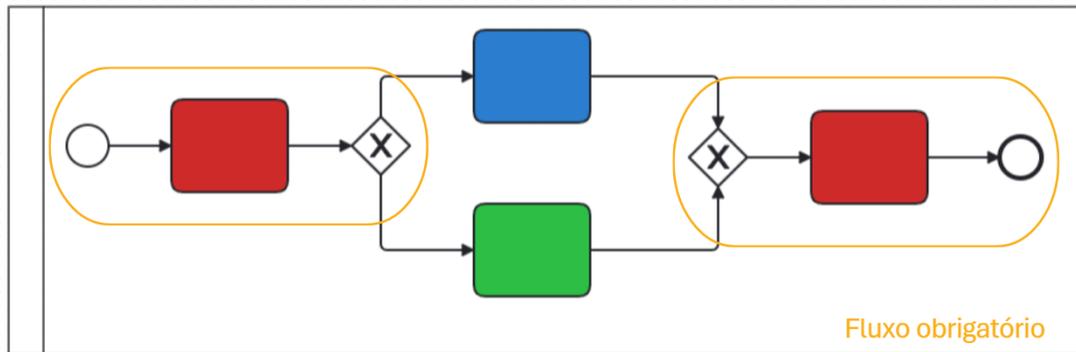


Figura 6 Processo com fluxo obrigatório

O resultado implica a adição de tarefas de ligação (figura 7).

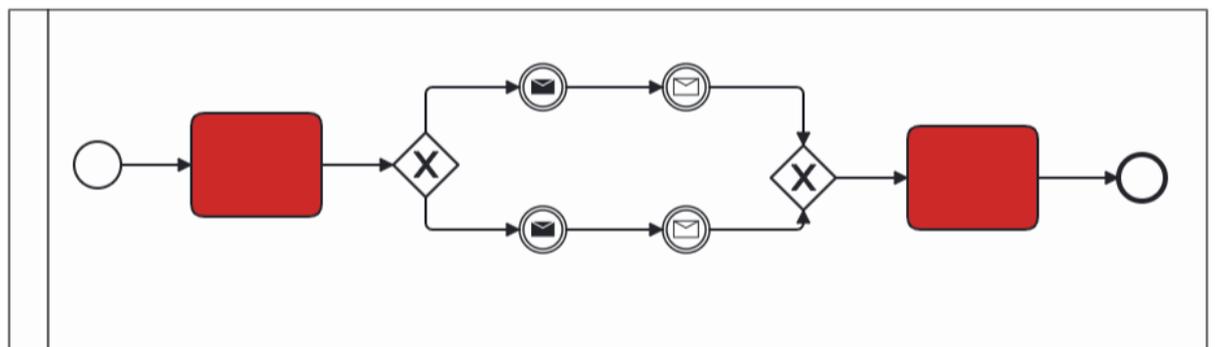


Figura 7 Resultado da agregação de ramos com fluxo obrigatório



Ramos com Fluxo Alternativo (Exclusivo): Este cenário ocorre quando o fluxo deve seguir para um ramo ou outro, mas não pode ocorrer em simultâneo (figura 8).

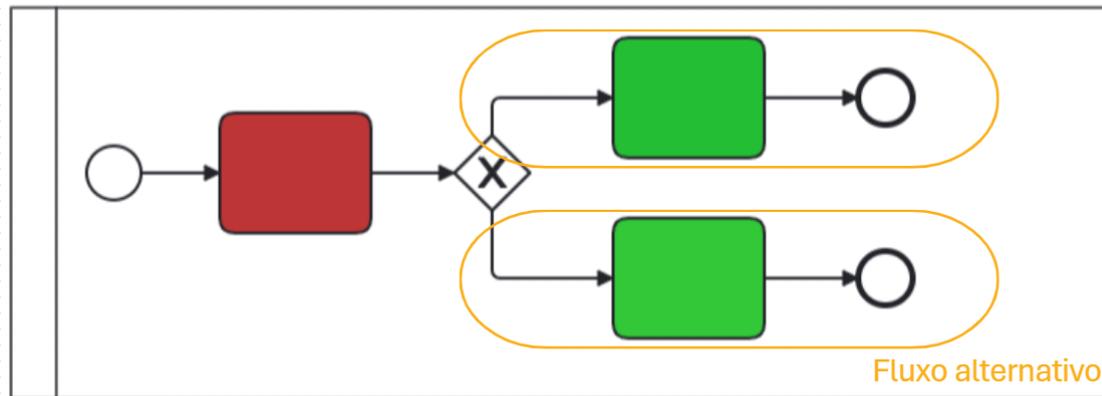


Figura 8 Processo com fluxo alternativo

O resultado é obtido através da adição de uma *gateway* baseada em eventos (figura 9)

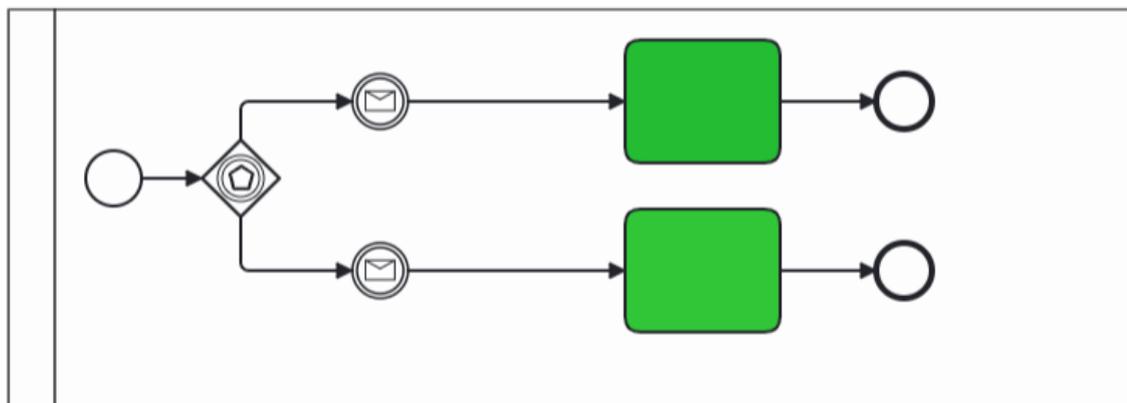


Figura 9 Resultado da agregação de ramos com fluxo alternativo

Ramos com Fluxo opcional: Este cenário ocorre quando o fluxo pode seguir para um ramo, para outro ou ambos (figura 10).

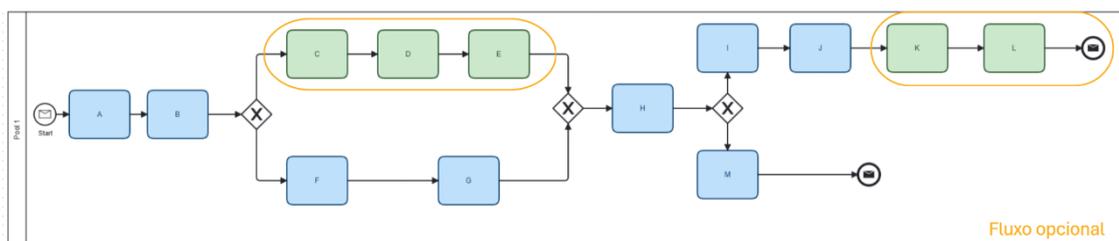


Figura 10 Processo com fluxo opcional

O resultado da fusão de ramos opcionais vai modificar a estrutura do processo (figura 11)

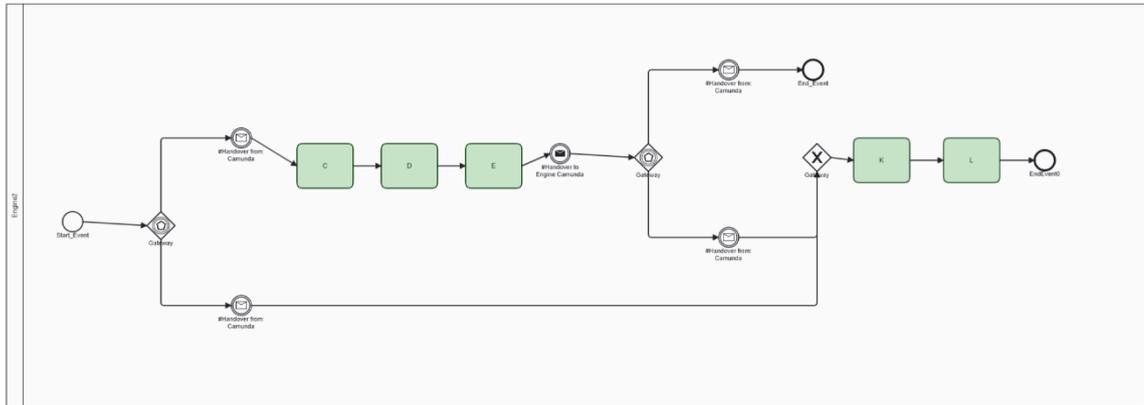


Figura 11 Resultado da agregação de ramos com fluxo opcional

Como demonstrado nos exemplos, a fusão de ramos impacta a estrutura do processo modelado em BPMN, direcionando a convergência dos fluxos de sequência. Para garantir o resultado desejado, podem ser necessários elementos de fluxo adicionais, como *gateways* e eventos intermediários de mensagem para a comunicação entre diferentes sistemas.

Objetivo da Agregação

O processo de agregação garante que todos os ramos identificados durante a fase de partição sejam corretamente integrados resultando num único BPMN por sistema, funcional e bem estruturado, com a comunicação e troca de dados entre os sistemas devidamente representadas, um exemplo do funcionamento é observado na figura 12.

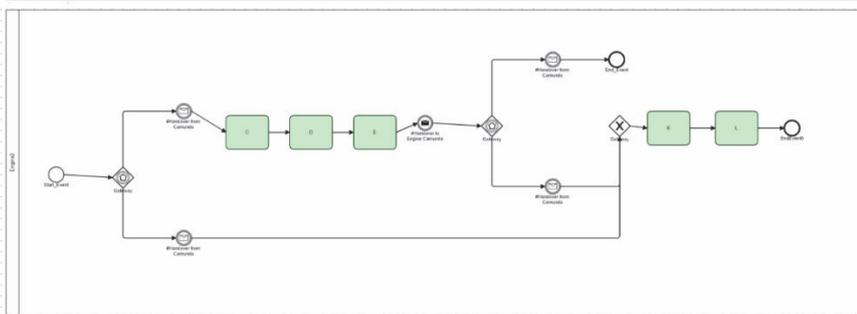
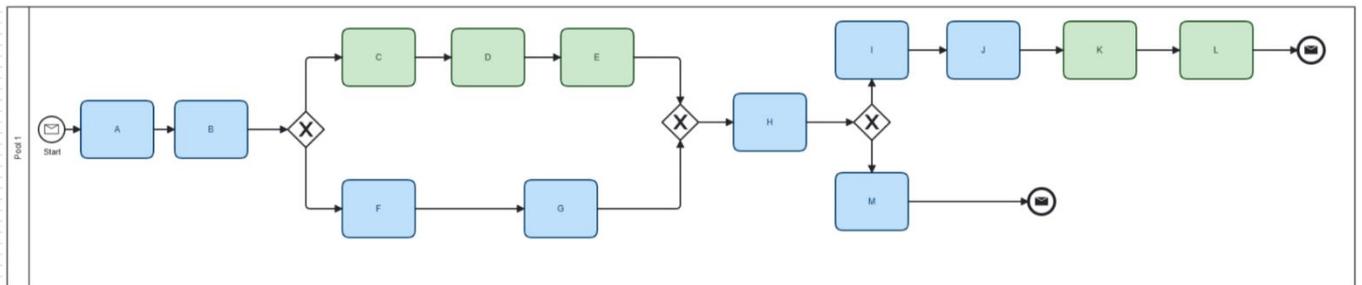
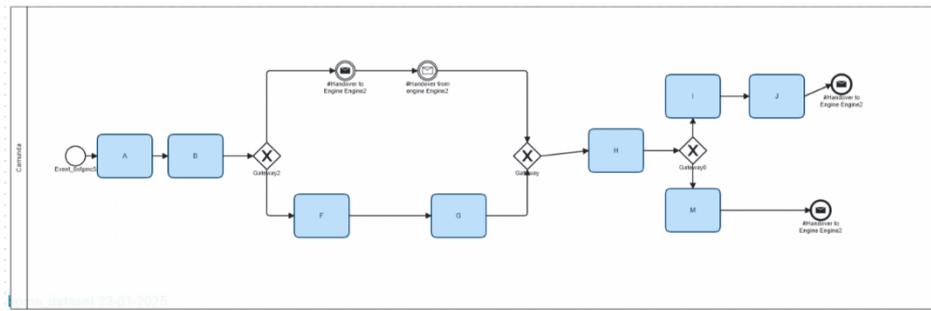


Figura 12 Exemplo completo