

Diagrama de Atividades

Um diagrama de atividades é uma ótima ferramenta para criar um modelo de como um **sistema deve se comportar** ao realizar uma série de ações.

O diagrama de atividades, tem como objetivo principal a especificação do **comportamento** do software, **do ponto de vista funcional**, ou seja, das suas funcionalidades. É muito semelhante a um fluxograma, uma ferramenta utilizada há muitas décadas, principalmente na administração.

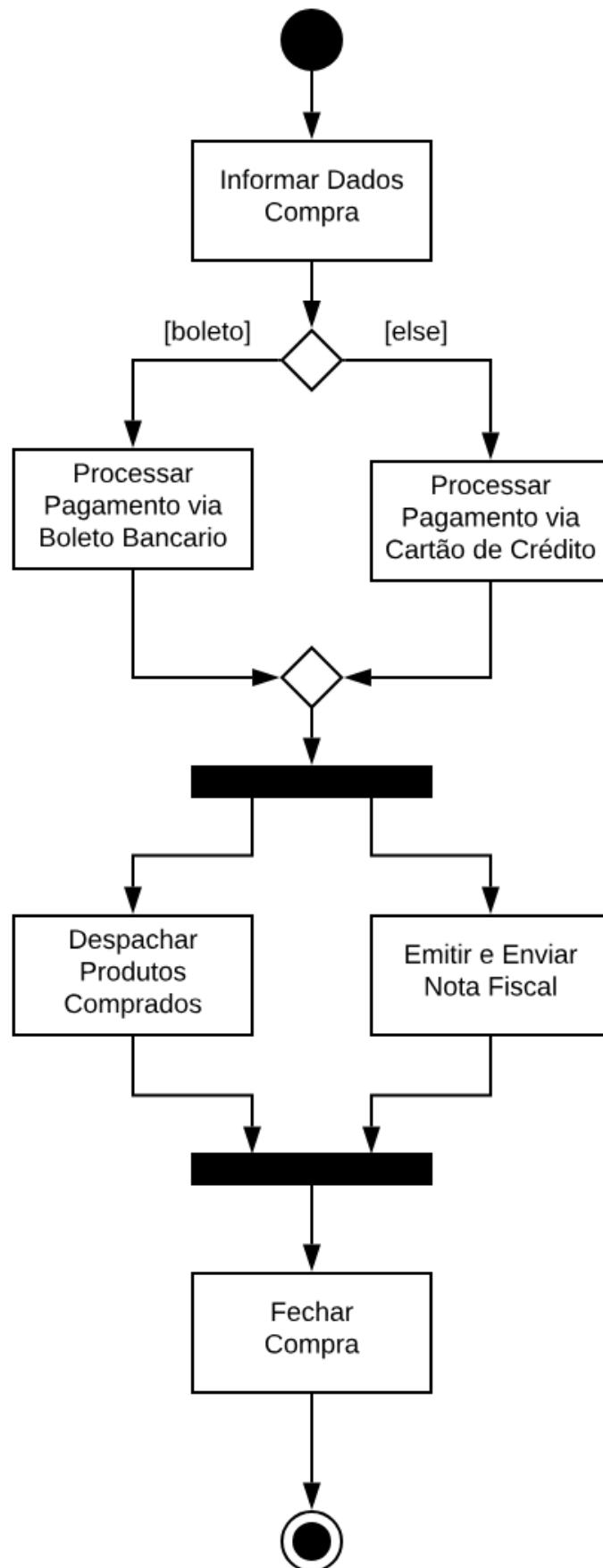
Os diagramas de atividades são especialmente usados no desenvolvimento de software e podem ser empregados nas diferentes fases de um projeto.

É utilizado para:

- **Documentar o aspecto funcional** (não estrutural) do software, quando é necessário representar o fluxo da informação que o software trabalhará, e quando existem condições/decisões que precisam detalhadas/descritas.
- **Mostrar aspectos específicos de alguma rotina do negócio que será automatizada pelo** software, como um “zoom” em parte de alguma funcionalidade, por exemplo. Obs.: muito cuidado ao especificar toda uma funcionalidade (uma tela ou rotina batch por exemplo) num diagrama de atividades. Geralmente isso gera diagramas de atividades super complexos, o que pode gerar efeito inverso (ou seja, “subtrair mais que somar”).
- **Mostrar como Funcionalidades vão realizar Requisitos Funcionais** (funções executadas pelas funcionalidades), e a **relação dos Requisitos Funcionais com as Regras de Negócio**.
- **Documentar de forma macro como o sistema irá funcionar**, mas orientado ao software, não ao processo de negócio. Mostrar como os módulos do sistema interagem entre si, as principais as informações trafegadas durante a execução do software, entradas e saídas principalmente.

Diagramas de atividades são usados para representar, em alto nível, um **processo** ou **fluxo** de **execução**. Os principais elementos desses diagramas são **ações** representadas por retângulos. Existem ainda elementos de **controle**, que definem a ordem de execução das ações.

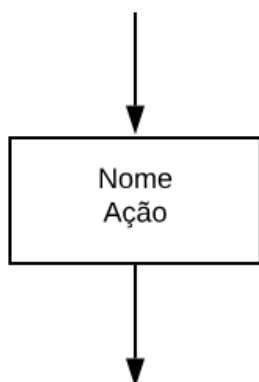
A figura abaixo mostra um diagrama de atividades que modela o processo seguido após um usuário fechar uma compra em uma loja virtual. Para isso, assume-se que os produtos comprados já estão no carrinho de compra.



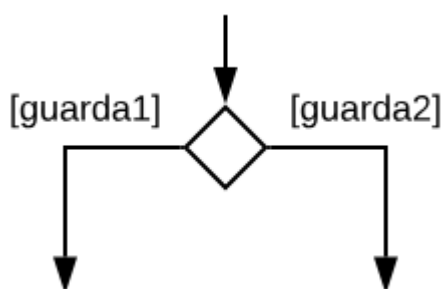
Nodo Inicial: Cria um novo fluxo para dar início à execução do processo. Feito isso, repassa a fluxo para seu único fluxo de saída. Por definição, o nodo inicial não possui fluxo de entrada.



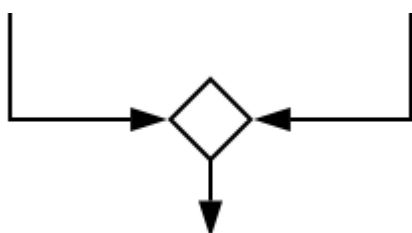
Ações: Possuem um único fluxo de entrada e um único fluxo de saída. Para uma ação ser executada uma fluxo precisa chegar de entrada. Após a execução, repassa-se o fluxo de saída.



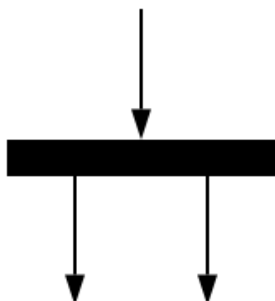
Decisões: Possuem um único fluxo de entrada e dois ou mais fluxos de saída. Cada fluxo de saída possui uma variável booleana associada, chamada de guarda. Para se tomar uma decisão, precisa-se receber uma fluxo de entrada. Quando isso acontece, o fluxo segue apenas para a saída cuja condição é verdadeira.



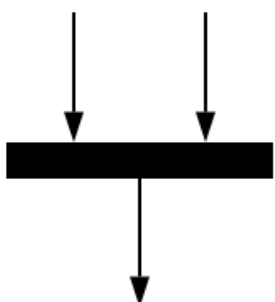
Merges: Podem possuir vários fluxos de entrada, mas um único fluxo de saída. Quando uma sinal chega em um dos fluxos de entrada, fazem seu repasse para o fluxo de saída. São usados para unir os fluxos de nodos de decisão.



Forks (Divergência): Possuem **um único fluxo de entrada** e **um ou mais fluxos de saída**. Atuam como multiplicadores de fluxos: quando recebem uma fluxo de entrada, criam e repassam fluxos idênticos de saída. Como resultado, passam a existir múltiplos processos em execução de forma paralela.



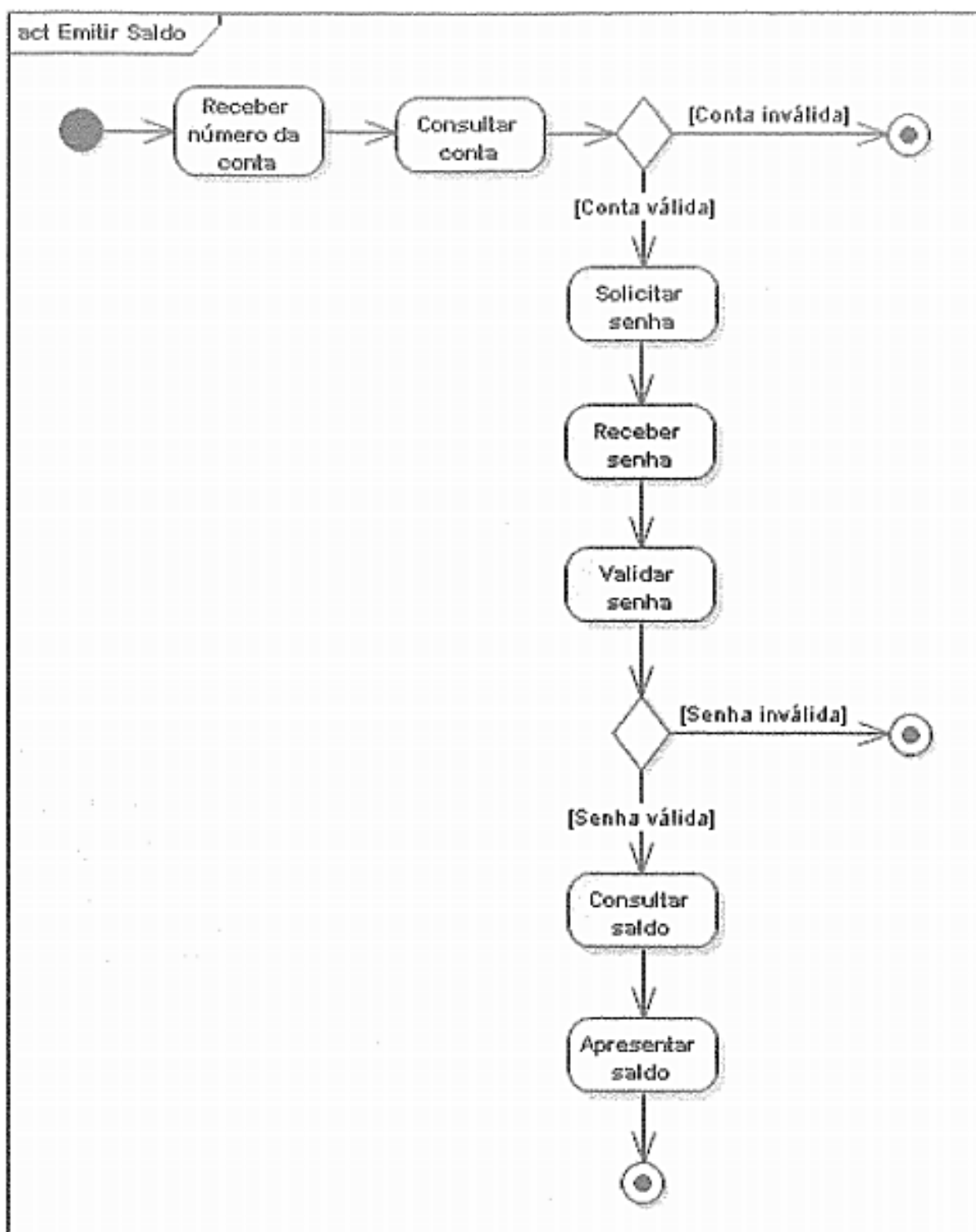
Joins (Convergência) : Possuem **vários fluxos de entrada**, mas um **único fluxo de saída**. Atuam como sincronizador de fluxos: espera a chegada de todos os fluxos de entrada. Quando isso acontece, repassam uma único fluxo de saída. Logo, são usados para sincronizar processos. Em outras palavras, transformar vários fluxos de execução em um único fluxo.



Nodo Final: Pode possuir mais de um fluxo de entrada; mas não possui fluxos de saída. Quando uma sinal chega em um dos fluxos de entrada, encerra-se a execução do diagrama de atividades.



Outro exemplo



Aprofundamento: Existem pelo menos três outras alternativas para modelagem de fluxos e processos:

- **Fluxogramas**, os quais foram propostos tão logo se começou a desenvolver os primeiros programas para computadores modernos. Diagramas de atividades são parecidos com fluxogramas; porém, eles incluem suporte a concorrência, por meio de *forks* e *joins*. Por outro lado, fluxogramas modelam processos sequenciais.
- **Redes de Petri** é uma notação gráfica, proposta pelo matemático alemão Carl Adam Petri, em 1962, para modelagem de sistemas concorrentes. Redes de Petri possuem uma representação gráfica e também usam fichas (*tokens*) para marcar o estado corrente do sistema. Elas têm ainda a vantagem de possuir uma definição mais formal, principalmente quando comparada com a definição de diagramas de atividades. Por outro lado, esses últimos tendem a oferecer uma notação mais simples e fácil de entender.
- **BPMN** (*Business Process Model and Notation*) é um esforço mais recente, que teve início nos anos 2000, visando a proposição de uma notação gráfica mais amigável para modelagem de processos de negócio do que aquela oferecida por diagramas de atividades. Um dos objetivos é propiciar que analistas de negócio possam ler, interpretar e validar diagramas BPMN.