



iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA



emprego
digital

Módulo 3: Princípios de Desenvolvimento de Software

Aula 6

Diagramas de Estado



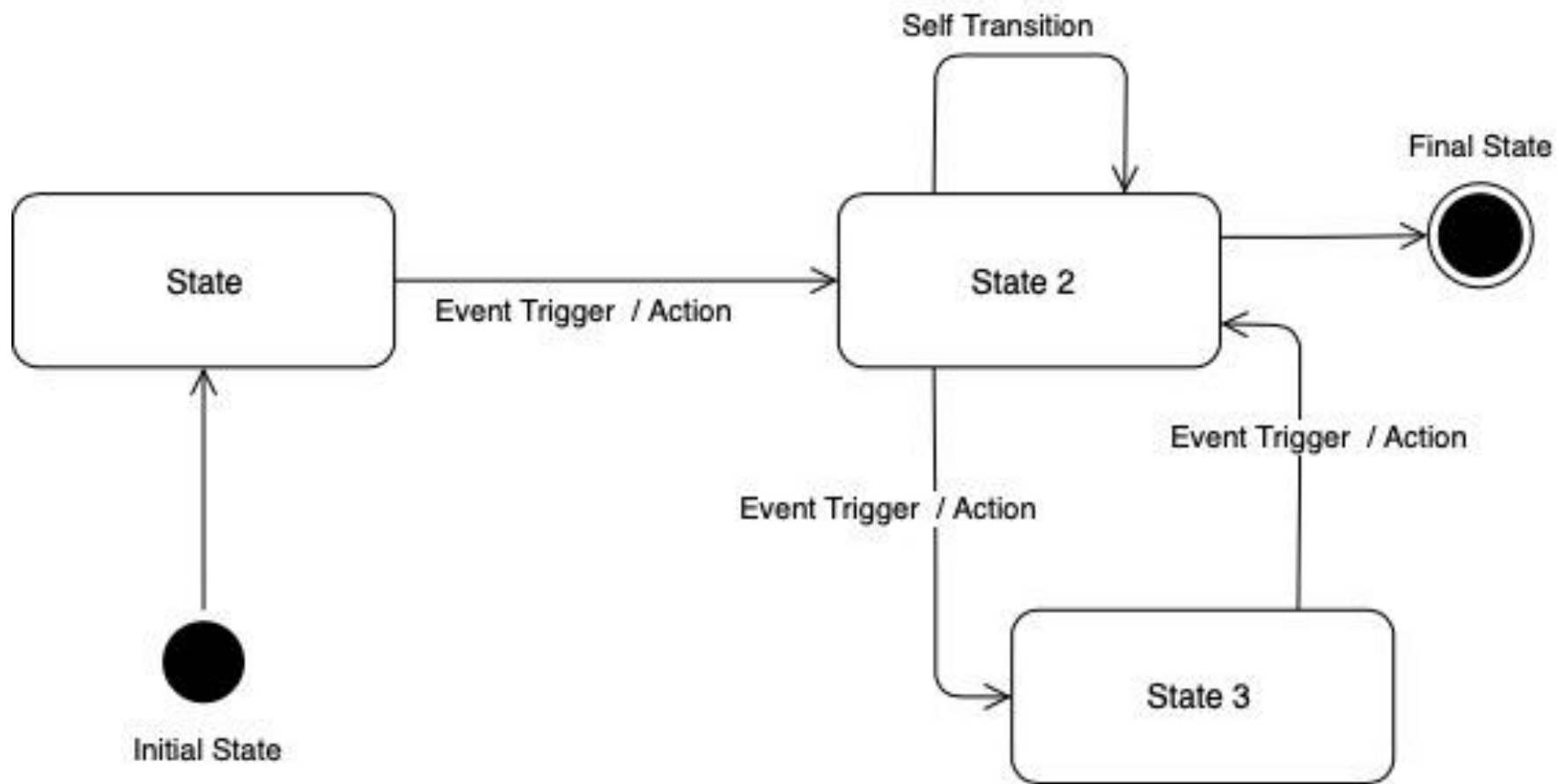
Recordando...

- Os **diagramas de sequência** são utilizados para representar a **sequência de processos** num programa de computador.
- Representa interações entre objetos, realizadas através de operações ou métodos.
- Este diagrama é construído a partir do Diagrama de Use Case. Primeiro, define-se qual o papel do sistema (Use Cases), depois, é definido como o software realizará seu papel (Sequência de operações).

Diagramas de estados

- Os **diagramas de estados** são uma representação dos estados em que um objeto pode se encontrar no decorrer da execução de um sistema, bem como as transições entre os estados.
- Um objeto pode passar de um estado inicial para um estado final através de uma transição.

Os Diagramas de Estado representam os estados dos objetos da nossa aplicação.



Componentes

- Os diagramas de *Estados* são representados pelos seguintes componentes:
 - **Estado:** Condição ou situação durante a vida de um objeto na qual ele satisfaz algumas condições, executa algumas atividades ou espera por eventos.
 - **Transição:** O relacionamento entre dois estados, indicando que o objeto que está no primeiro estado irá passar para o segundo estado mediante a ocorrência de um determinado evento e em certos casos uma condição.

Componentes

- Os diagramas de *Estados* são representados pelos seguintes componentes:
 - **Condição:** Causa necessária para que haja a transição de estado. Decorre da ocorrência de um evento ou circunstância que propicia a transição de estado.
 - **Estado inicial:** Estado por onde se começa a leitura de um diagrama de estado.
 - **Estado final:** Estado que representa o fim de uma máquina

Transições

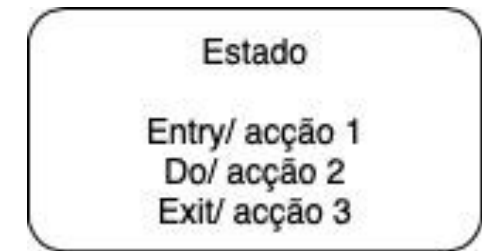
- As transições são uma relação entre dois estados que indica que a execução de certas acções de um objecto o fará passar de um estado a outro quando um evento específico ocorrer e certas condições específicas forem satisfeitas. Cinco componentes:
 - **Estado fonte:** o estado afectado pela transição
 - **Event Trigger:** evento cuja recepção pelo estado fonte torna a transição elegível caso a condição de guarda seja satisfeita (o evento pode ser a passagem de tempo, uma invocação explícita de um método, um sinal)

Transições

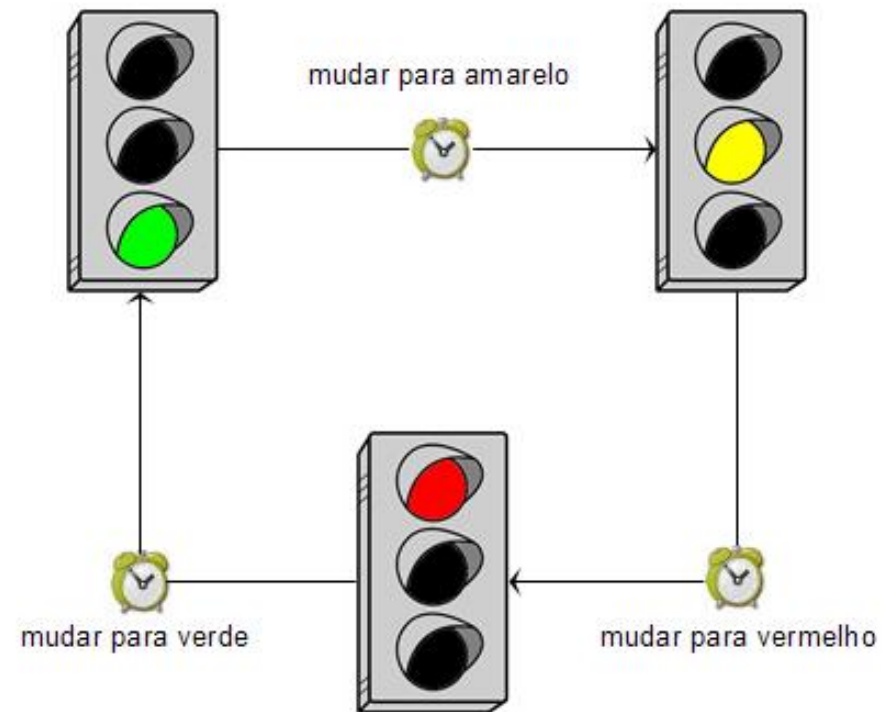
- **Guarda:** expressão booleana que é avaliada quando ocorre um event trigger (caso seja verdadeira a transição ocorre, caso contrário a transição não ocorre e o evento morre)
- **Acção:** operação atómica (não pode ser interrompida) que pode ocorrer no momento da transição (usualmente é a responsável pela transição embora possam ocorrer transições sem acções, e.g., time out)
- **Estado Destino:** estado activo depois da transição

Ações e Atividades

- É possível representar as acções que ocorrem dentro de cada estado, no diagrama de estados:
 - **Entry:** acção que acontece ao entrar no estado
 - **Do:** acção que acontece durante a permanência no estado
 - **Exit:** acção que acontece ao sair do estado

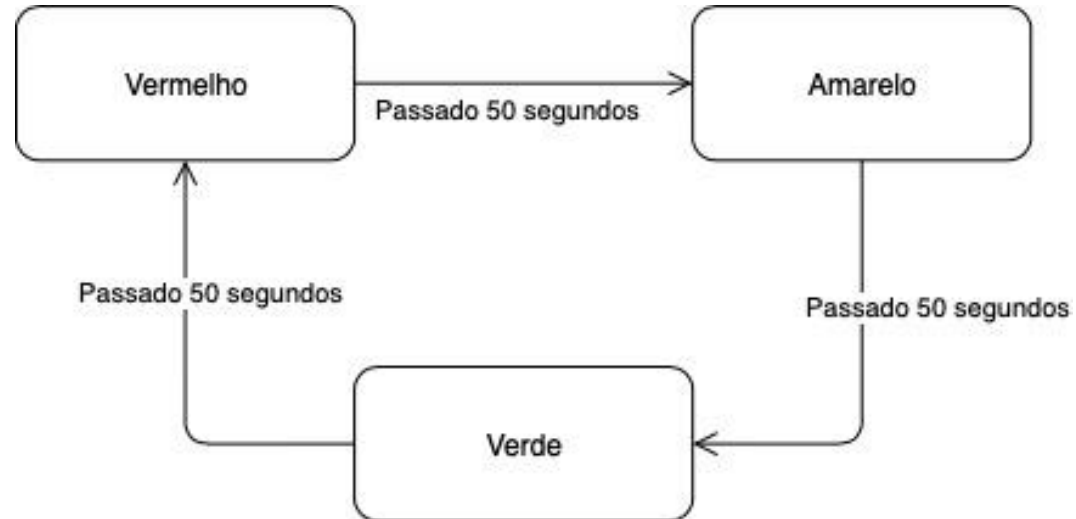


Exemplo 1 - Semáforo



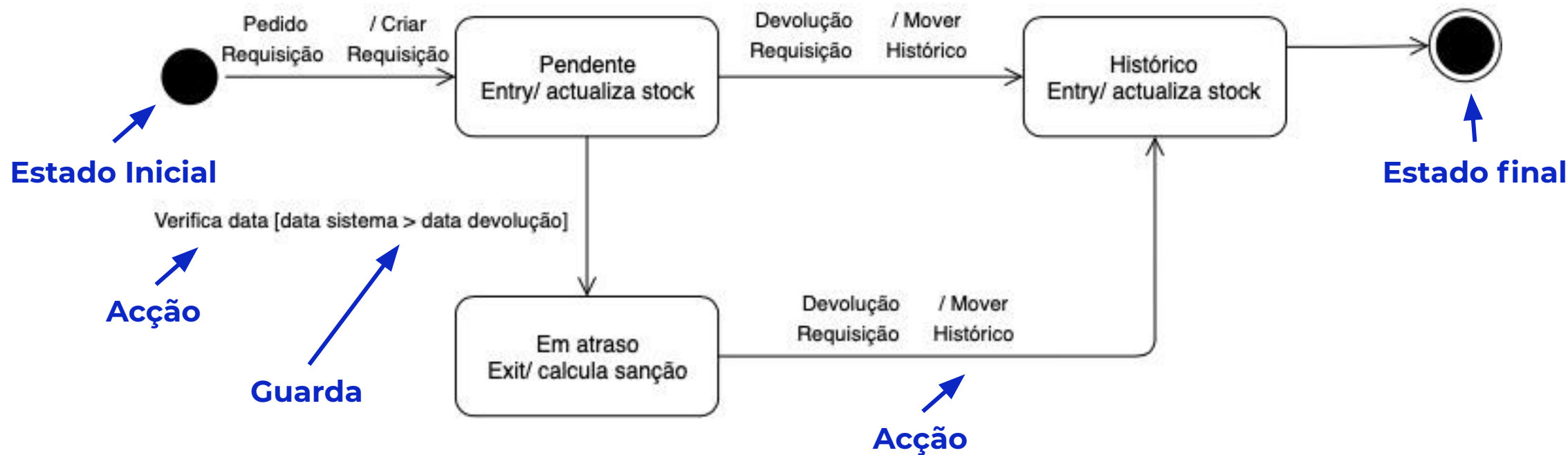
Exemplo 1 - Semáforo

Estado

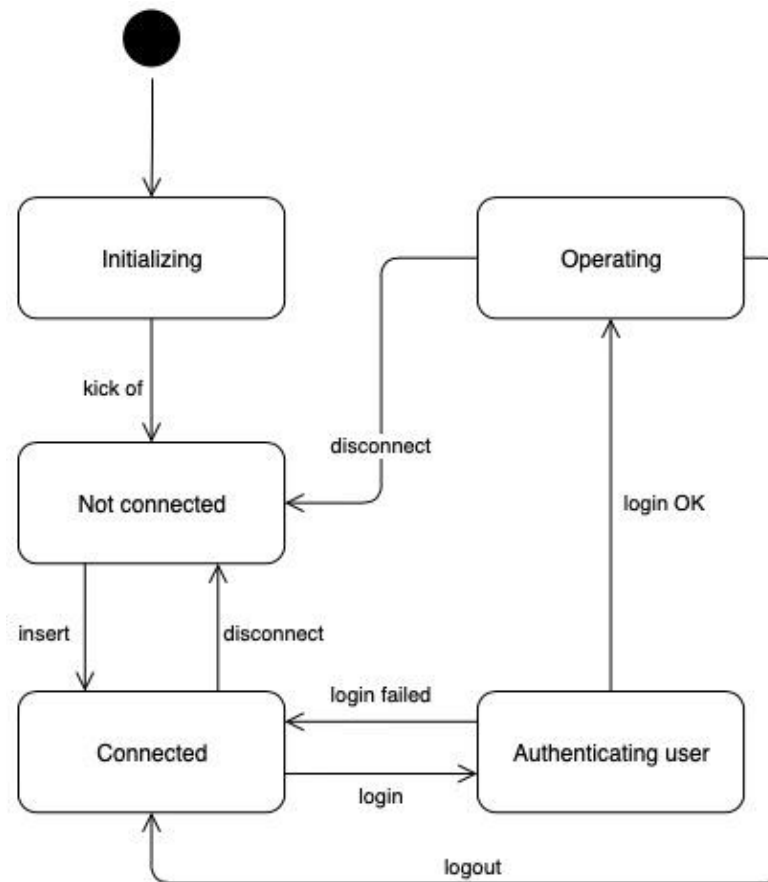


Acção

Exemplo 2 - Pedido de Requisição



Exemplo 3 - Login



Estados vs. Acções

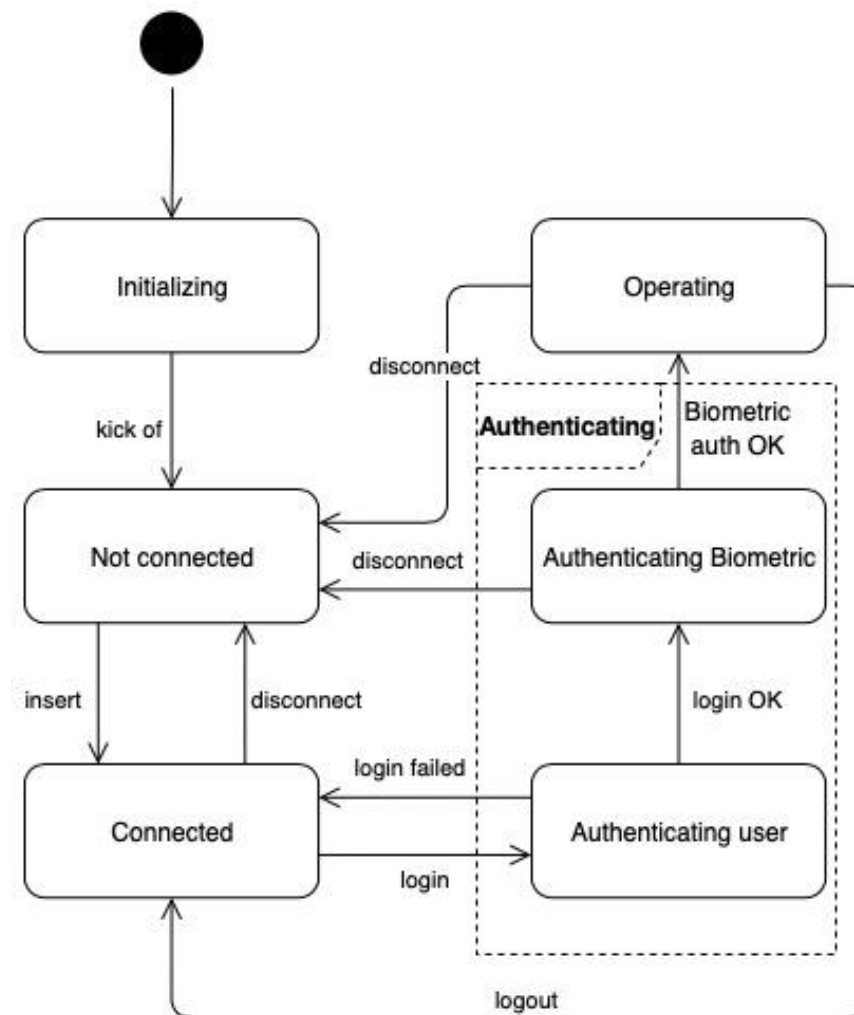
- É muito importante diferenciar o que é um estado, do que é uma acção.
- Passar o semáforo a vermelho -> **Acção**
- Vermelho -> **Estado**
- Emitir fatura -> **Acção**
- Fatura emitida -> **Estado**

Os Diagramas de Estado modelam apenas os estados dos nossos objetos, e não as acções.


Embora possamos incluir as acções como especificação adicional através do entry, do e exit

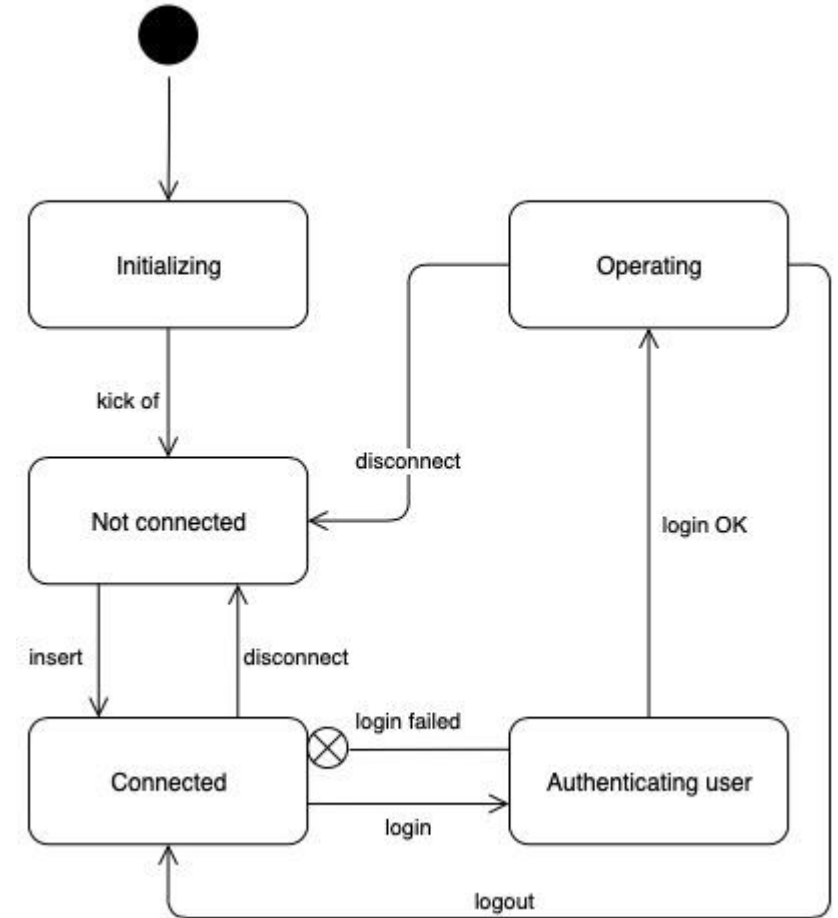
Estados compostos

- Um diagrama pode conter sub-diagramas.
- Tomemos como base o anterior exemplo 3, mas agora com um sistema de dupla autenticação.
 - **Podemos criar um sub diagrama para representar a autenticação**



Exit point

- O Exit Point permite explicitar situações abruptas de término.
- É representado pelo símbolo: 
- **Exemplo:** login falhado gera terminação abrupta.



Concorrência e Convergência

- **Concorrência**

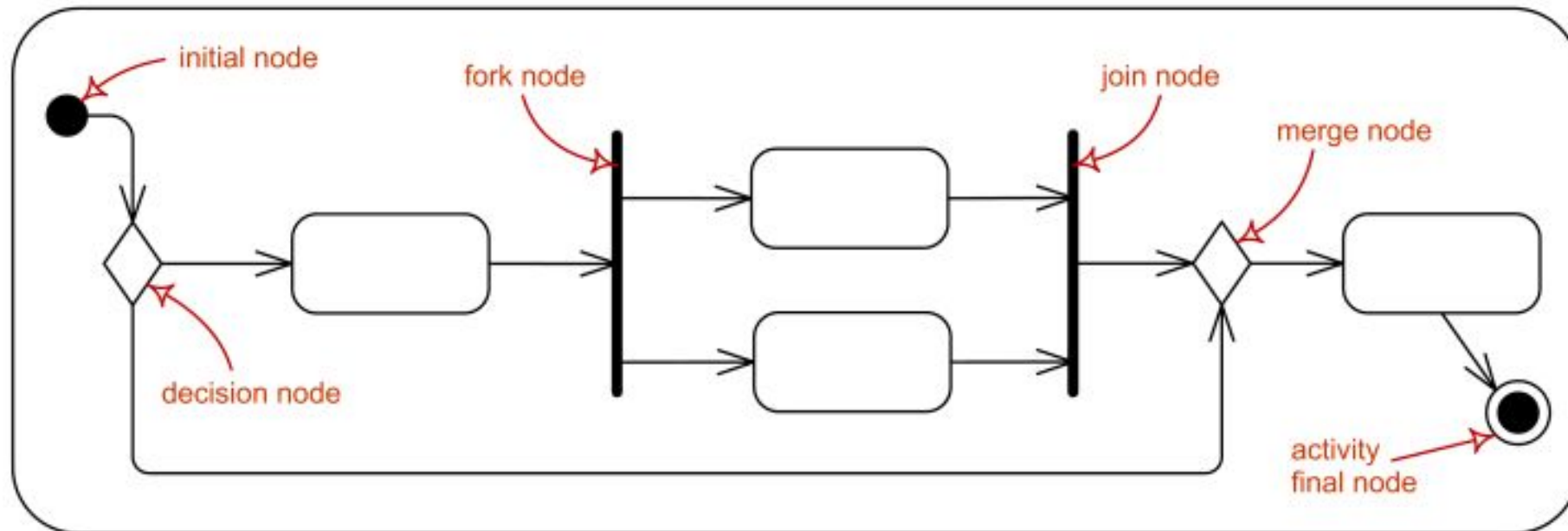
- É possível representar **fork** e **join**

- **Convergência**

- Podemos utilizar nós de decisão para representar decisões e convergência.

- Não é obrigatório ter um fork antes de um join.

Concorrência e Convergência



**Boa prática: um diagrama de estados
deve modelar apenas os estados de
um único objeto.**

Vamos praticar?

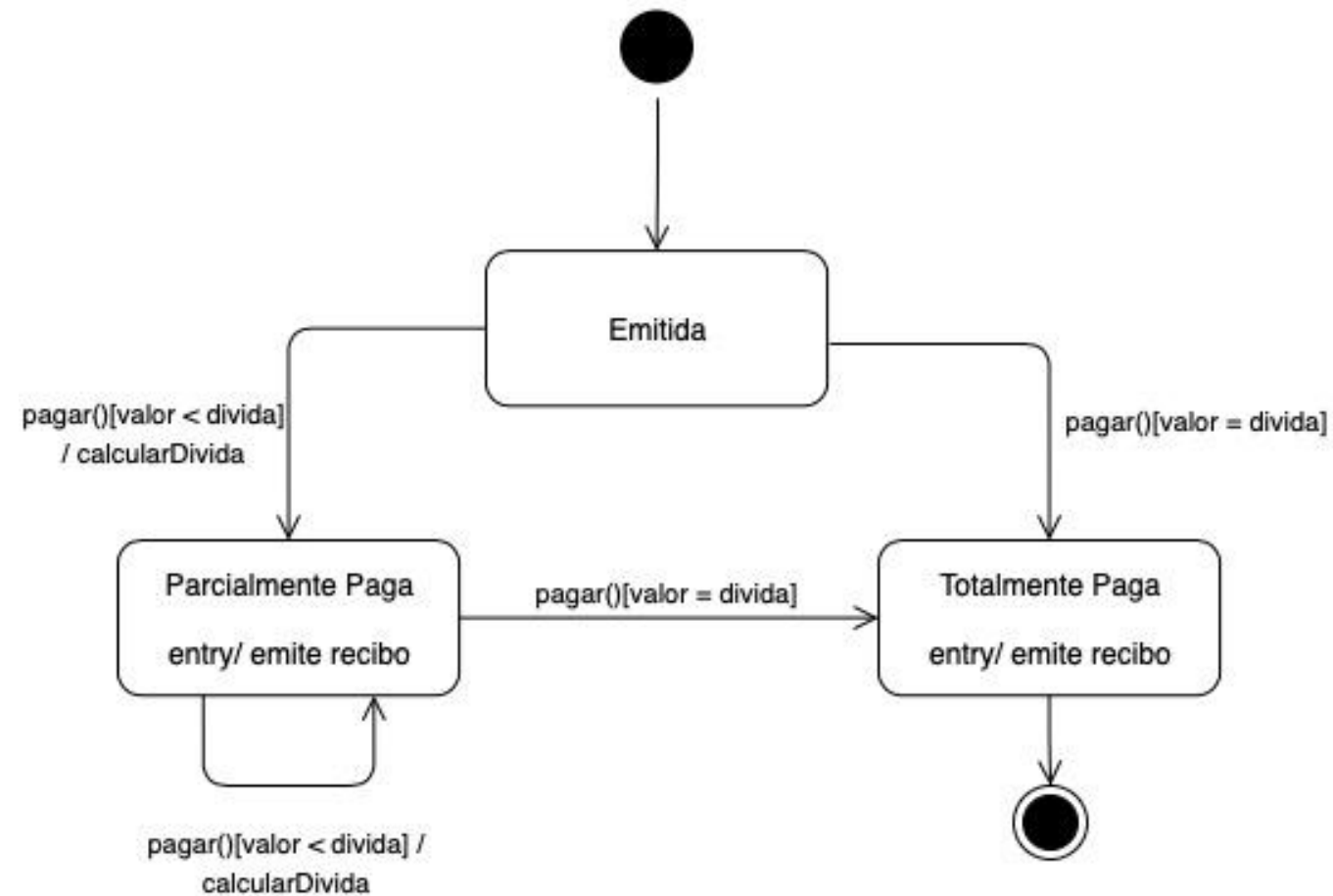
Exercício 1

A empresa Bem Gerida mantém um rigoroso controlo sobre cada factura, a partir do momento em que é emitida até se encontrar totalmente paga. Para tal, é controlado o estado da fatura, que pode assumir os seguintes valores: *Emitida*, *Parcialmente Paga*, *Totalmente Paga*.

Se for feito um pagamento inferior ao valor em dívida, a fatura considera-se *Parcialmente Paga*; se o valor do pagamento for igual ao valor em dívida, a fatura ficará *Totalmente Paga*. Cada vez que é feito um pagamento, tem de ser apurado o valor que permanece em dívida e emitido o respetivo recibo.

Elabore o diagrama de estados para a fatura.

Exercício 1 - Resolução



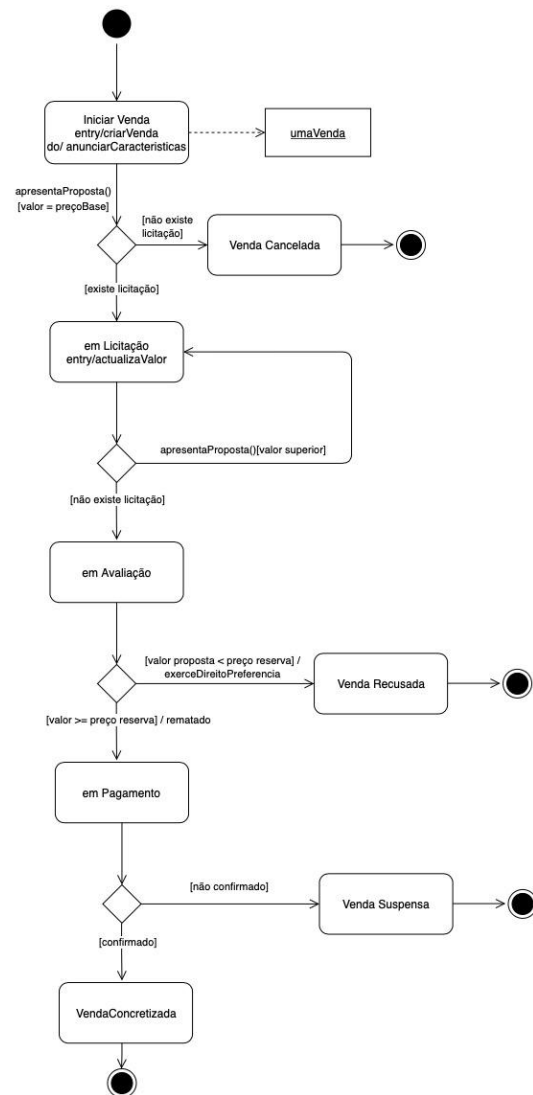
Exercício 2

Pretende utilizar-se um diagrama para descrever os estados da venda de uma peça pelo método de leilão Inglês.

1. O leiloeiro apresenta as características da peça, incluindo o preço base, e convida os clientes a apresentarem uma licitação inicial.
2. Se não existirem interessados que licitem, a venda é cancelada.
3. Se receber uma licitação, o leiloeiro propõe um novo valor e convida os clientes a apresentarem uma nova licitação. Esta situação mantém-se até que não surja mais nenhuma licitação.
4. Então, o leiloeiro faz uma avaliação do leilão: se o preço for superior ao valor de reserva, a licitação é aceite, caso contrário a venda é recusada.
5. Após o pagamento ter sido efetuado a venda é dada como concretizada. Se o pagamento não for efetuado, a venda é considerada suspensa.

Elabore o diagrama de estados que descreve o cenário apresentado.

Exercício 2 - Resolução

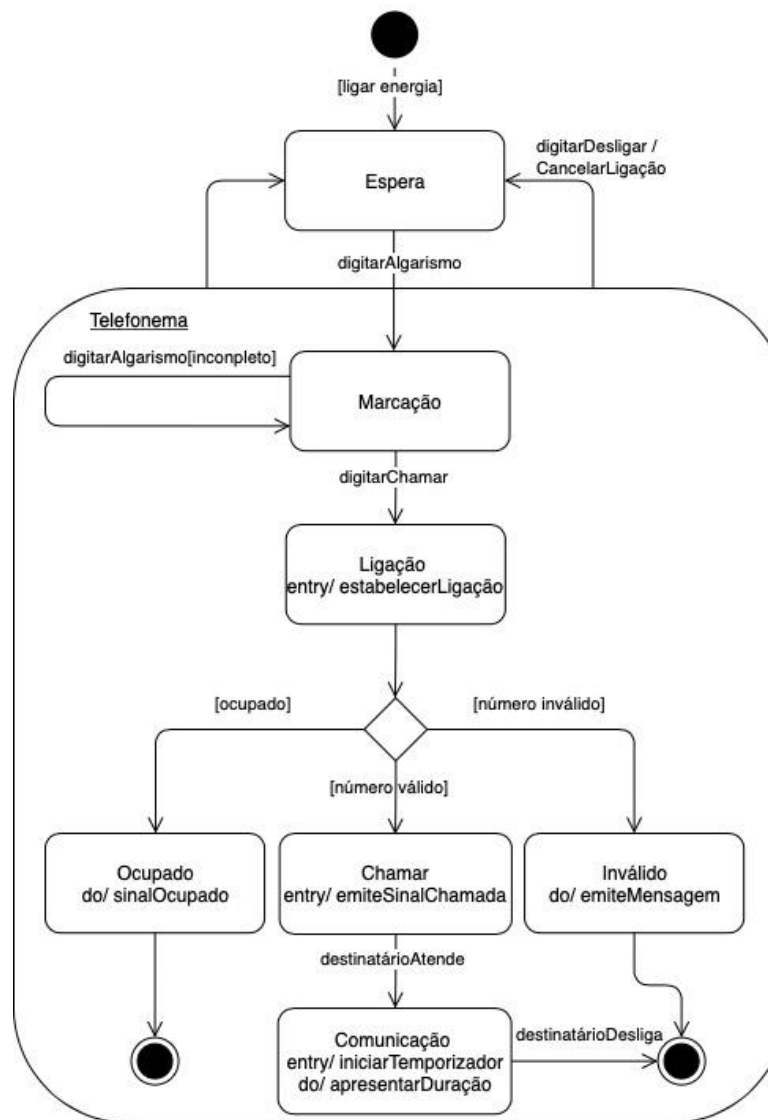


Exercício 3

Pretende utilizar-se um diagrama para descrever os estados de um telemóvel quando está a fazer uma chamada.

Inicialmente, o telemóvel encontra-se em *Espera*. Quando se começam a digitar os números, passa para um estado *Marcação*, onde se mantém até que seja clicada a tecla de ligar. Nesse momento, é efetuada a ligação. Se o telefone do destinatário estiver ocupado, surge um sinal. Se o número não for correto, é emitida uma mensagem de número inválido. Se o número for válido, o telefone fica a receber um sinal de chamar. Se o destinatário atender a chamada o telefone fica num estado que permite a comunicação e é iniciada a contagem do tempo de chamada. A chamada pode ser concluída pelo destinatário ou pelo emissor. Nesse momento, é apresentada a duração da chamada. Após concluída a chamada, o telemóvel volta ao estado de *Espera*. A qualquer momento, é possível clicar na tecla desligar, regressando ao estado de *Espera*.

Exercício 3 - Resolução



O futuro profissional começa aqui

iscte
INSTITUTO
UNIVERSITÁRIO
DE LISBOA

**emprego
digital**

UPskill