

PROJETO PROCEDIMENTAL

Projeto de Programas – PPR0001

Introdução

- A trípole de modelagem é composta por:
 - Modelo de Objetos: especifica a estrutura dos objetos. É importante quando muitas classes não triviais são identificadas no problema
Diagrama Entidade Relacionamento e Diagrama de Classes
 - Modelo Funcional: especifica os resultados de um processamento sem especificar como ou quando eles serão processados. Evidencia quais dados são entradas de um processo e quais devem ser as saídas.
Diagrama de Fluxo de Dados (DFD)
 - Modelo Dinâmico: representa a parte dinâmica do sistema, especificando os principais estados e eventos do sistema.
Diagrama de Eventos e Diagrama de Estados

Modelagem Dinâmica

- Vamos estudar e escrever o relacionamento entre as entidades em relação ao tempo (modelo dinâmico)
 - Diagrama de Eventos
 - Diagrama de Estados
 - Descreve sequência de operações que ocorrem em resposta a um evento
 - Descreve os estados em que um objeto pode estar
 - **Não** há preocupação em **como** as operações são implementadas
- Devemos escrever apenas os diagrama de acordo com as necessidades e a complexidade do sistema

Evento e Estados

- Estado: representa um estado do sistema; um conjunto característico de valores dos atributos
 - ❖ Período de matrícula, aulas, exames
 - ❖ Condição normal, cansado, envenenado, dormindo.
- Evento: estímulo individual de um objeto para outro
 - ❖ Data fim do período de aulas → início do período de exames
 - ❖ Dormindo → recebe ataque → acordado / condição normal

Diagrama de Eventos

- Representam cenários: uma sequência de eventos que ocorrem durante uma determinada execução do sistema
 - ❖ A abrangência do cenário pode variar
- Eventos podem transmitir informações de um objeto para outro
- Representamos o tempo de cima para baixo
- Distâncias são irrelevantes
- Foco na sequência de eventos
- Podemos usar a ferramenta ASTAH para gerar tal diagrama (usar diagrama de sequencia com mensagens assíncronas)
 - *ASTAH Community* (versão gratuita) permite criar este diagrama



Diagrama de Eventos

Exemplo – Chamada Telefônica

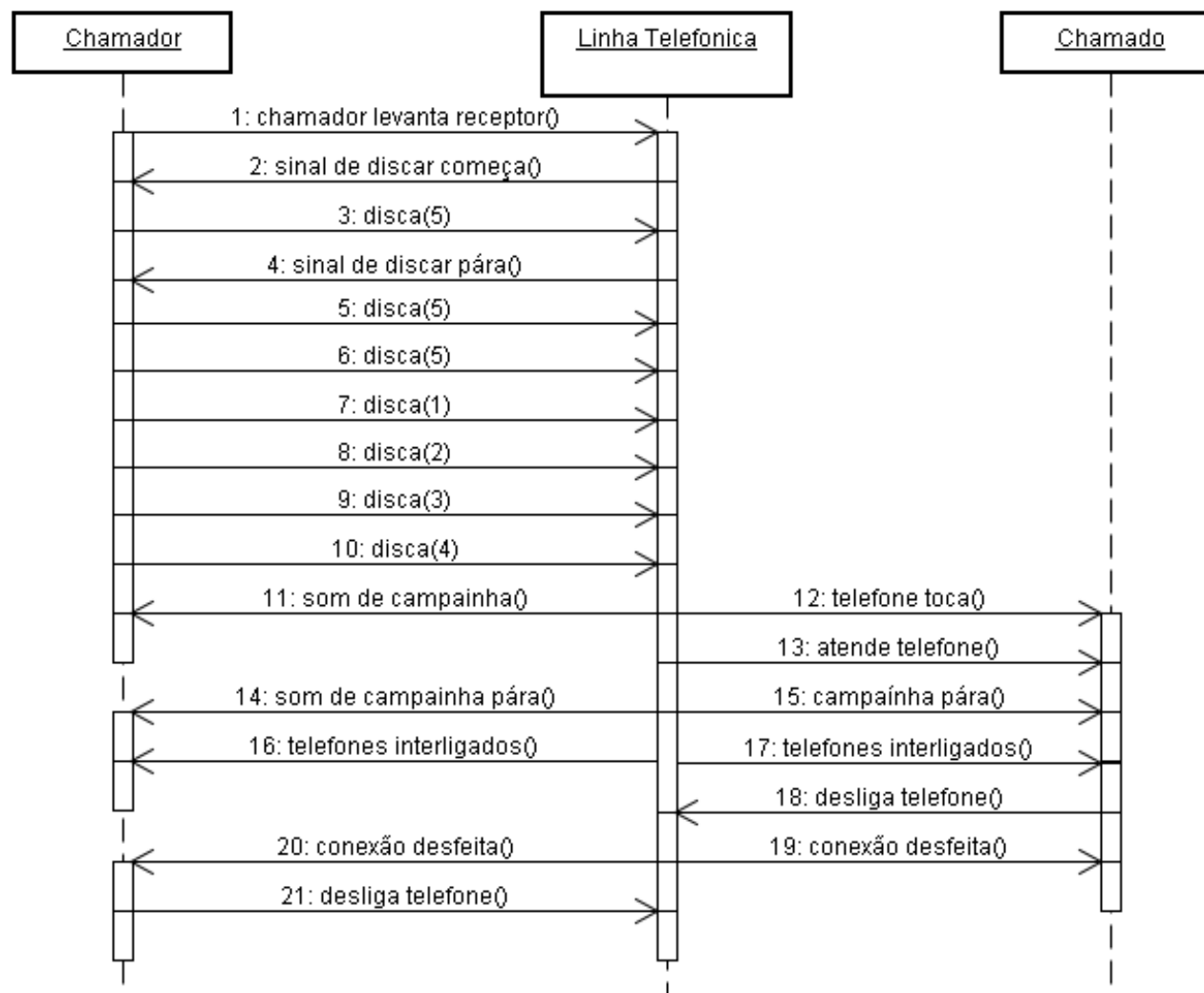


Diagrama de Estados

- É um diagrama que representa os possíveis estados dos objetos e os eventos que levam de um estado a outro
- Um estado pode ser duradouro (intervalo de tempo)
 - e.g. viajar de uma cidade a outra, alarme soando, ou inatividade (telefone).
- Um estado está relacionado aos valores de objetos
 - e.g água líquida → temperatura maior que 0°C e menor que 100°C;
transmissão de um automóvel → ré || ponto morto || primeira || ...;
- Deve-se concentrar nas qualidades (das entidades) que são relevantes ao sistema e abstrair as irrelevantes

Diagrama de Estados

- Diagramas de Estados podem representar:
 - Ciclos de vida (e.g. jogo de xadrez)
Marcados por um início e um fim
 - Laços contínuos (e.g. linha telefônica)
Estado inicial e final não existe, ou não se conhece, ou não é de interesse para a representação do sistema.
- Podemos utilizar o *ASTAH Community* para desenhar estes diagramas também:
 - Diagrama de Máquina de Estados

Diagrama de Estados - Representação

- Estado:



- Representado por um retângulo arredondado com nome que o identifica;
- Cada estado pode ter um ou mais eventos associados a ele;

- Transição:



- Passagem de um estado para outro por meio de um evento;
- Representada por uma seta direcionada e rotulada.

- Estado Inicial:

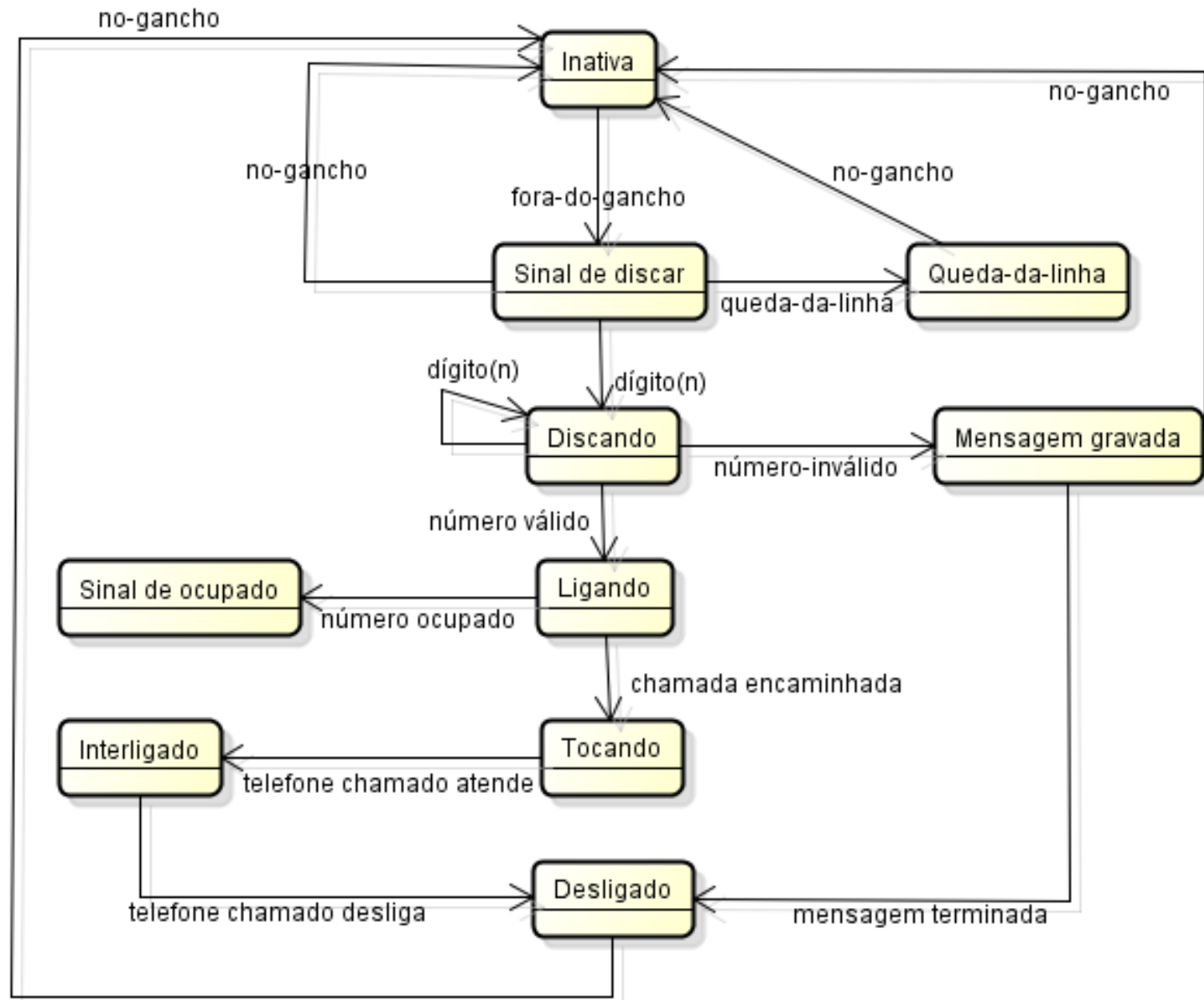
- Indica um inicio e pode representar a criação de um objeto
- Representado por um círculo cheio, pode ser rotulado.

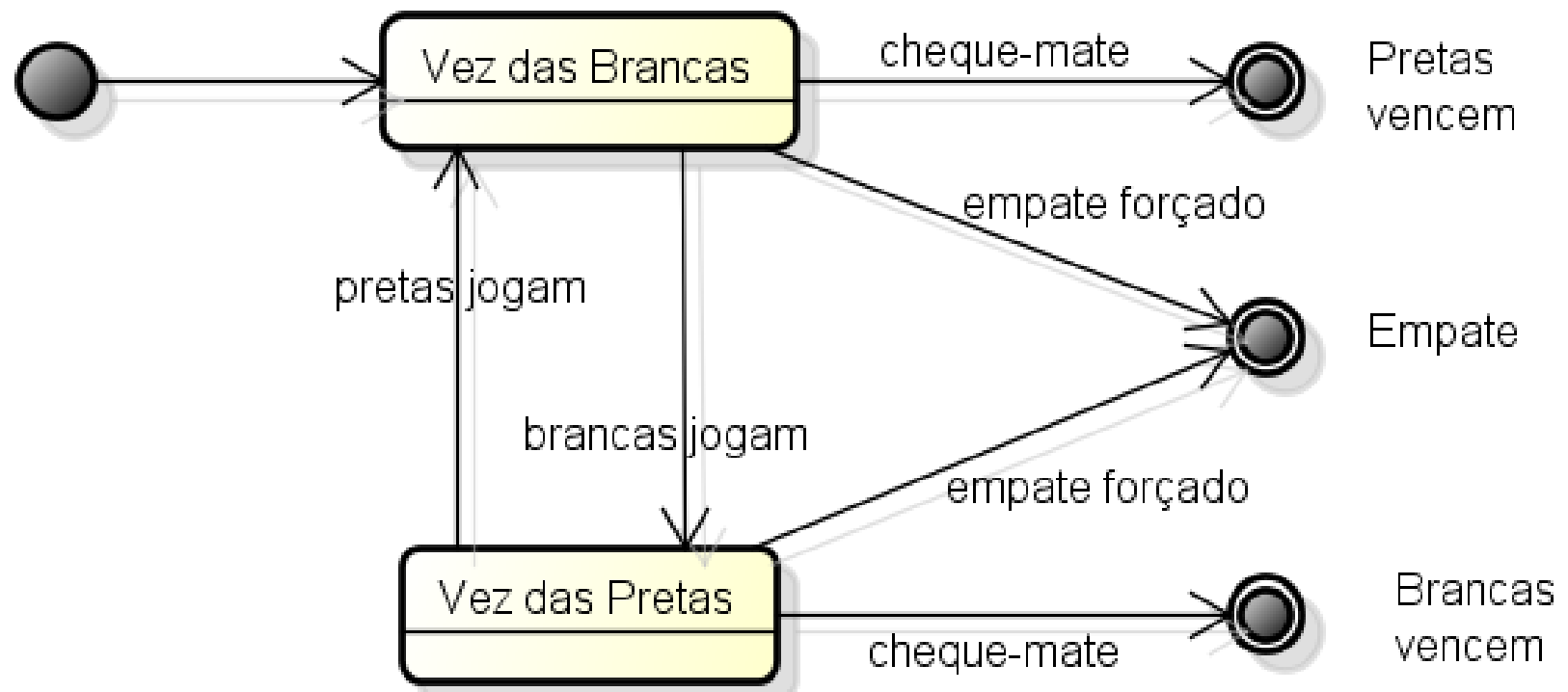


- Estado Final:

- Indica uma possível terminação e pode representar a destruição de um objeto
- Representado por um “olho-de-boi”, pode ser rotulado.







Podemos representar o estado de “estar” em uma tela como um estado:

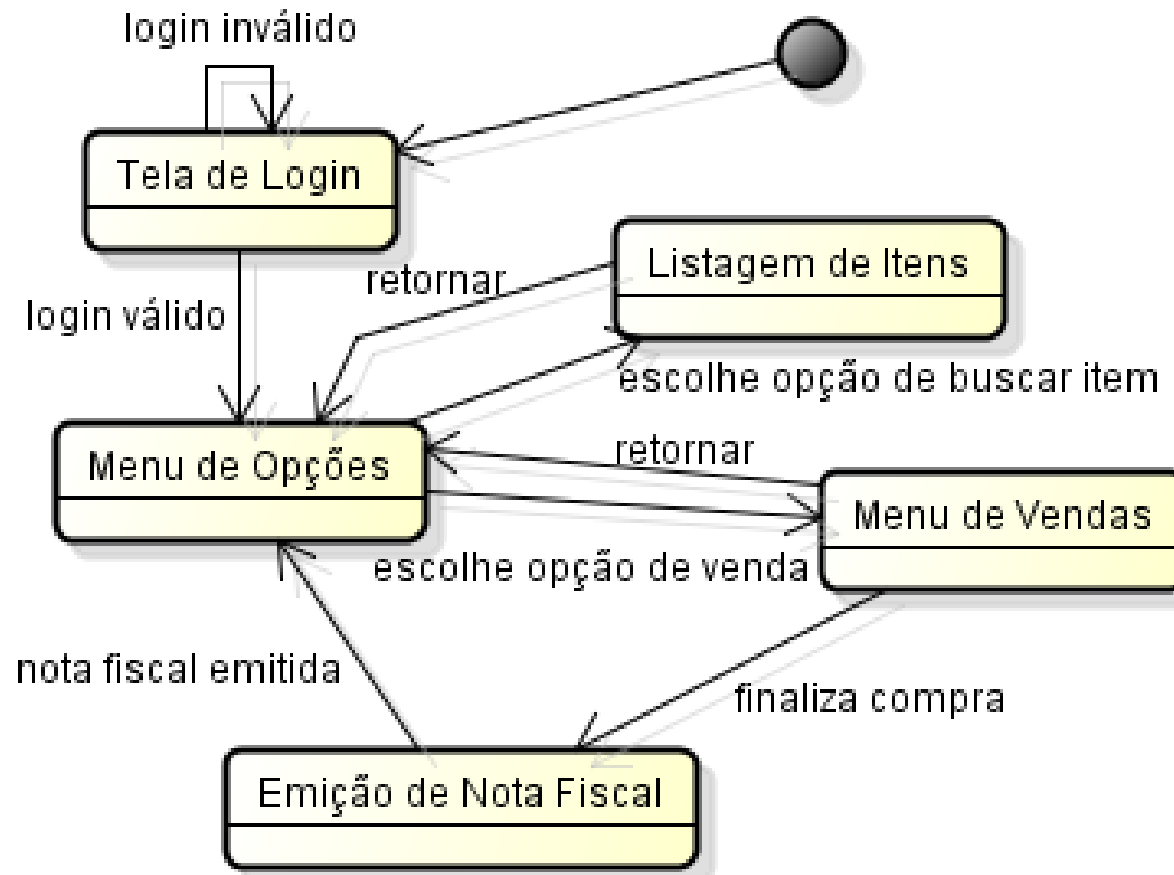


Diagrama de Estados

- Condições:

- Podem ser utilizadas como guardas nas transições, garantindo que uma transição só ocorre sob certa condição;
- Representadas de forma textual entre colchetes;
- No ASTAH utilize o campo “guard” das setas de transição;

- Operações:

- Descrição comportamental que especifica o que um objeto faz em determinado estado ou em resposta a eventos;
- Representação textual após uma barra (/) em transições;
- Representação textual após a expressão “faça /” (ou “do /”) em estados;

Diagrama de Estados – Condições

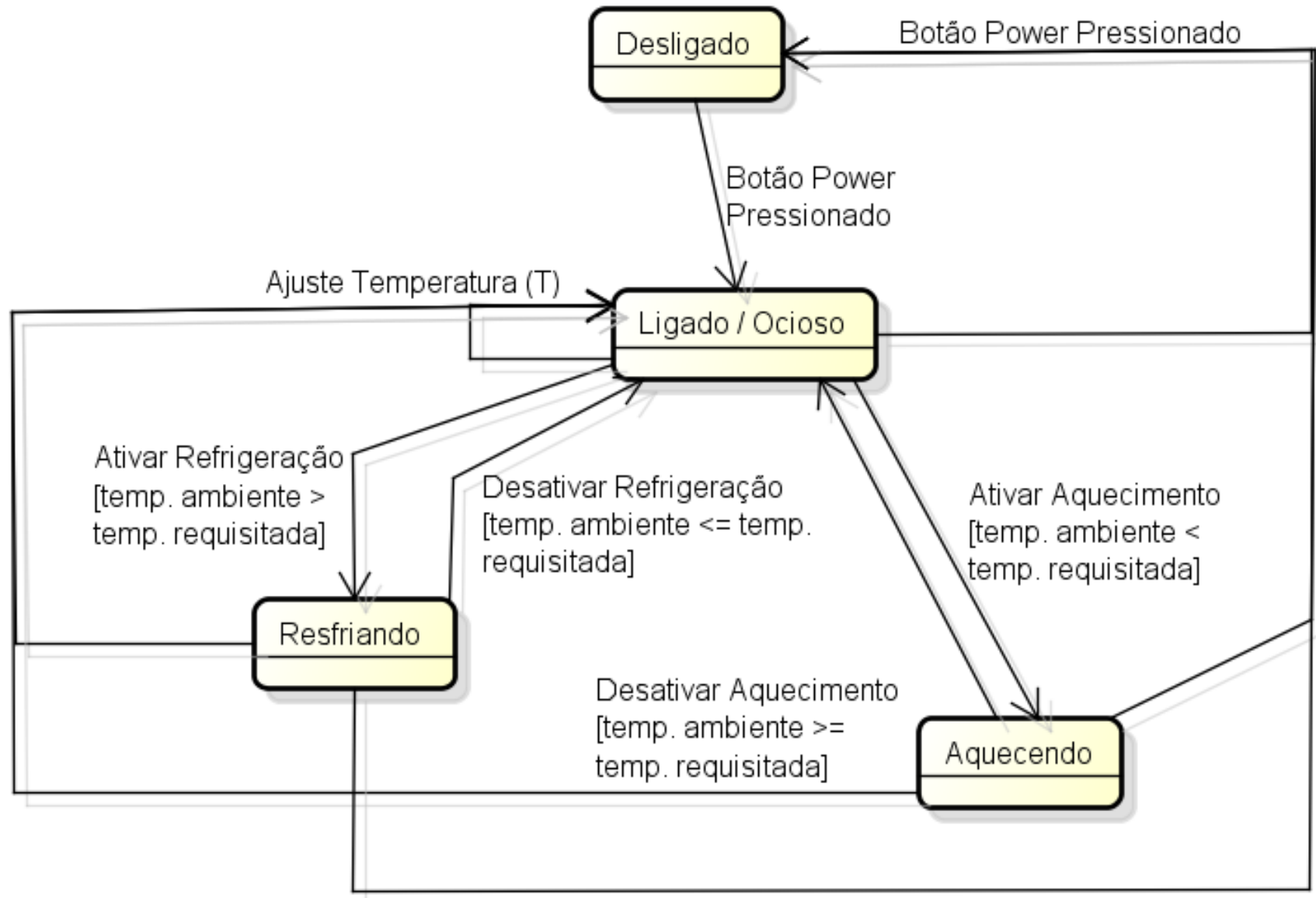


Diagrama de Estados – Operações

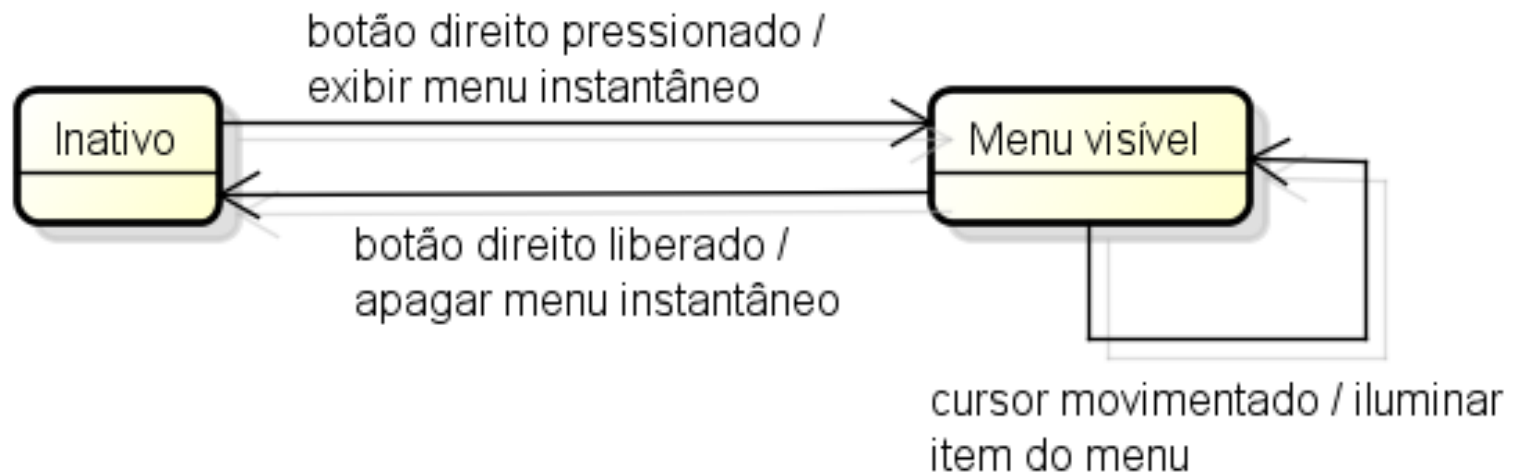


Diagrama de Estados – Operações

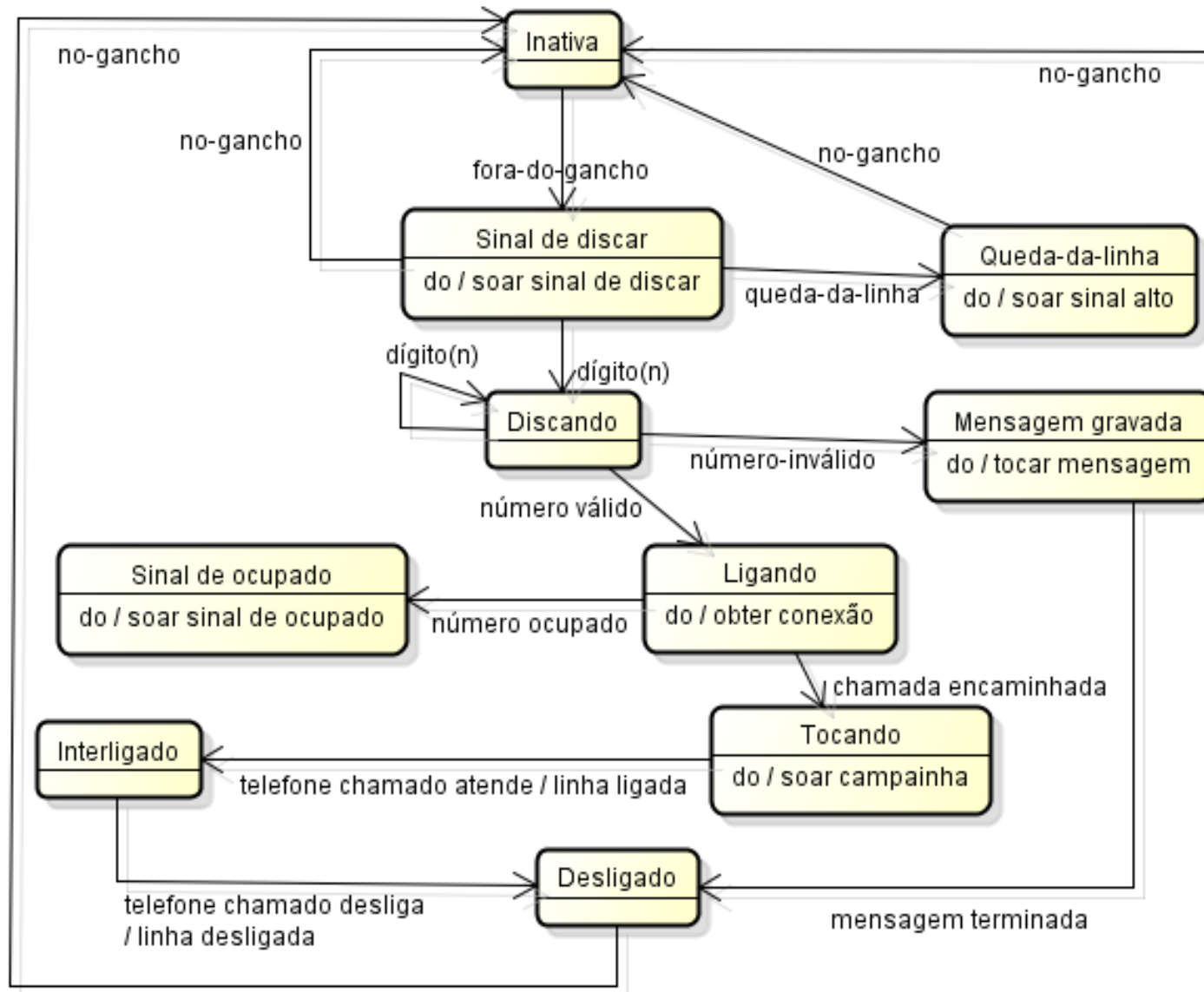


Diagrama de Estados Multinivelados

- Sistemas complexos podem requerer diagramas com muitos estados e eventos → falta de legibilidade

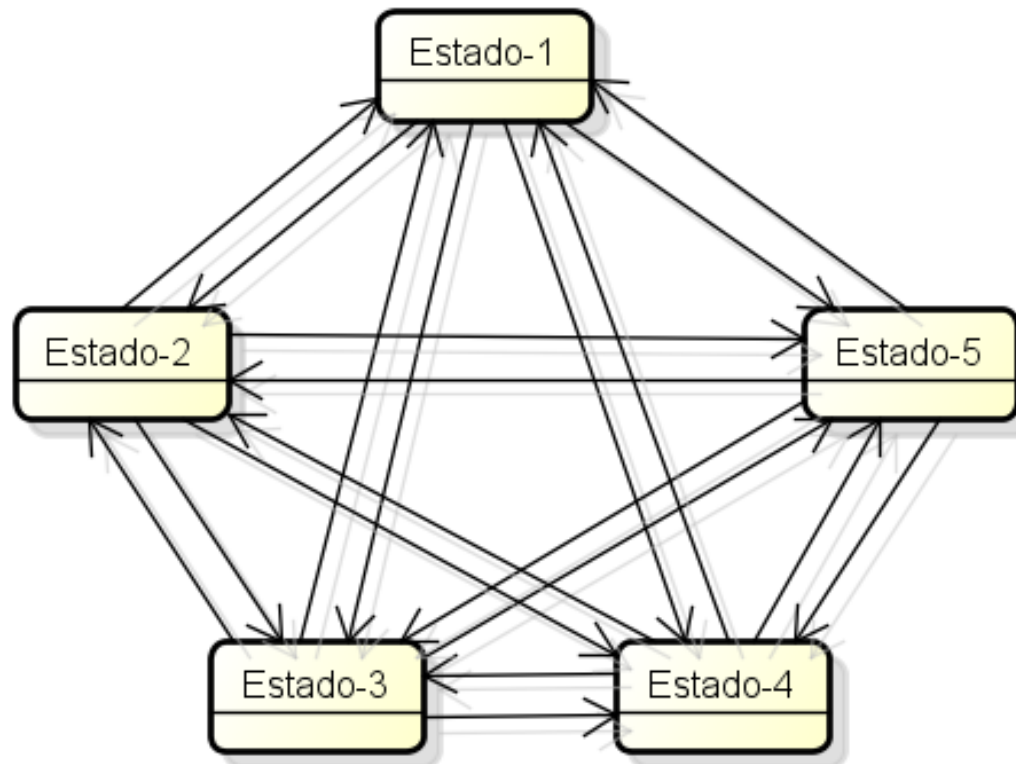
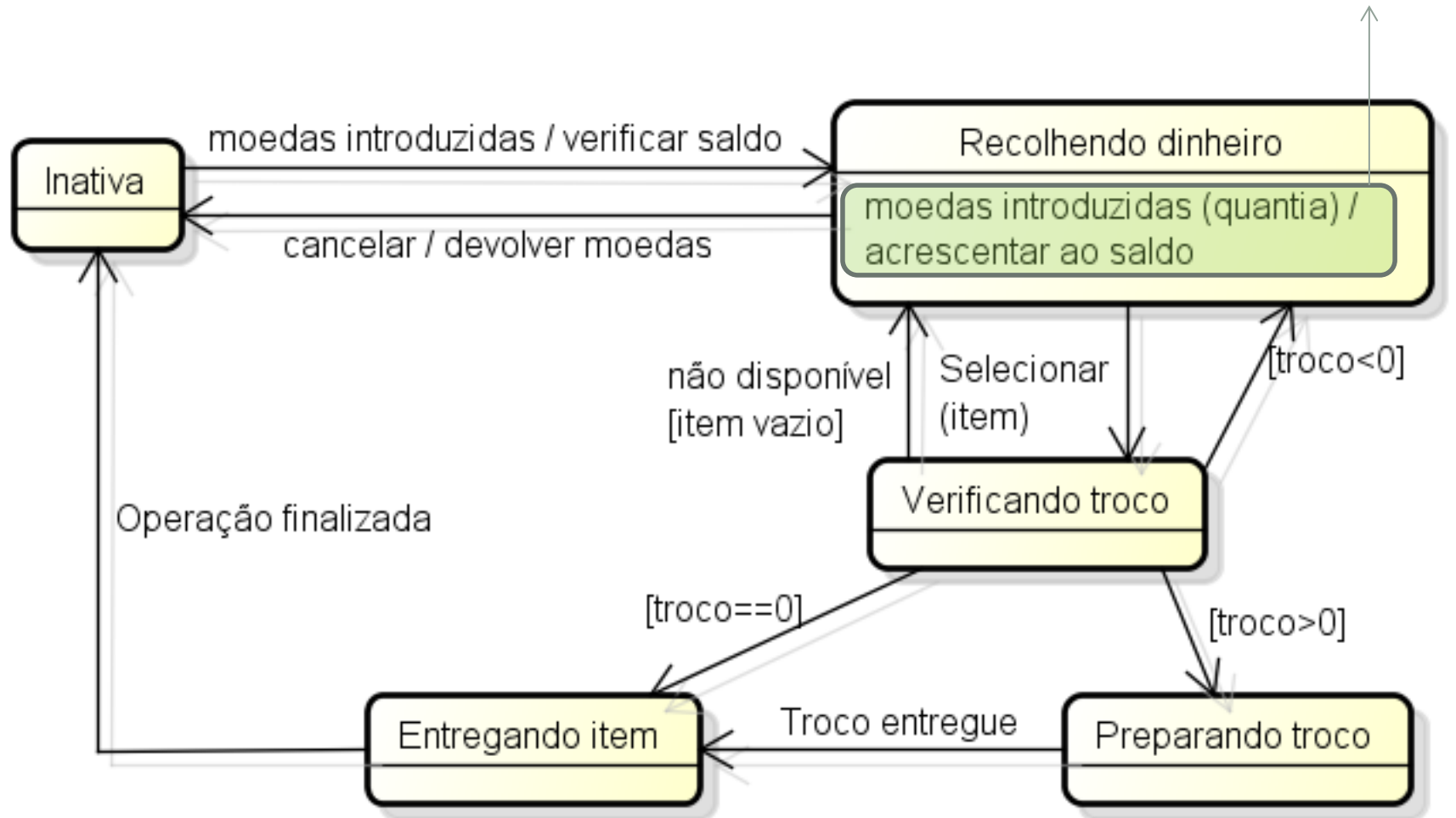
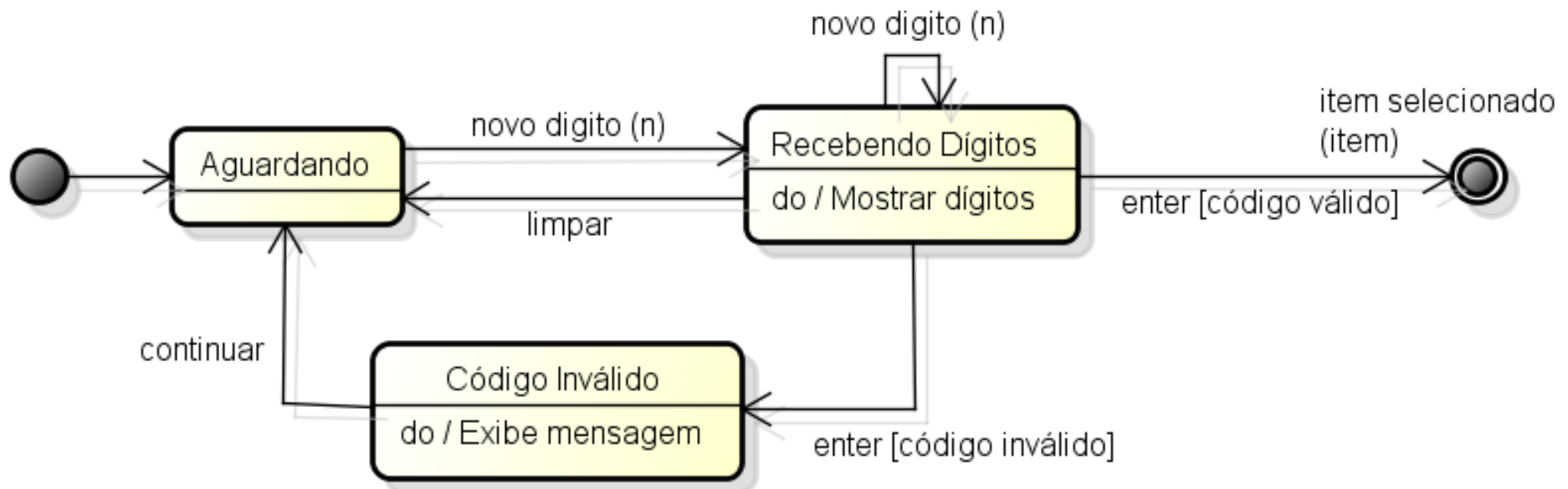


Diagrama de Estados Multinivelados

- Para manter a qualidade de expressão e melhorar a legibilidade os diagramas são desenhados em vários níveis de detalhamento
 - A quantidade de níveis pode variar de sistema para sistema;
 - A relação entre os diagramas deve ficar clara;
 - Vários diagramas podem fazer parte de um mesmo nível de detalhamento;

Exemplo: Máquina de vender (Nível 1)

Exemplo: Máquina de vender (Nível 2)
Selecionando item



Exemplo: Máquina de vender (Nível 2)
Entregando item

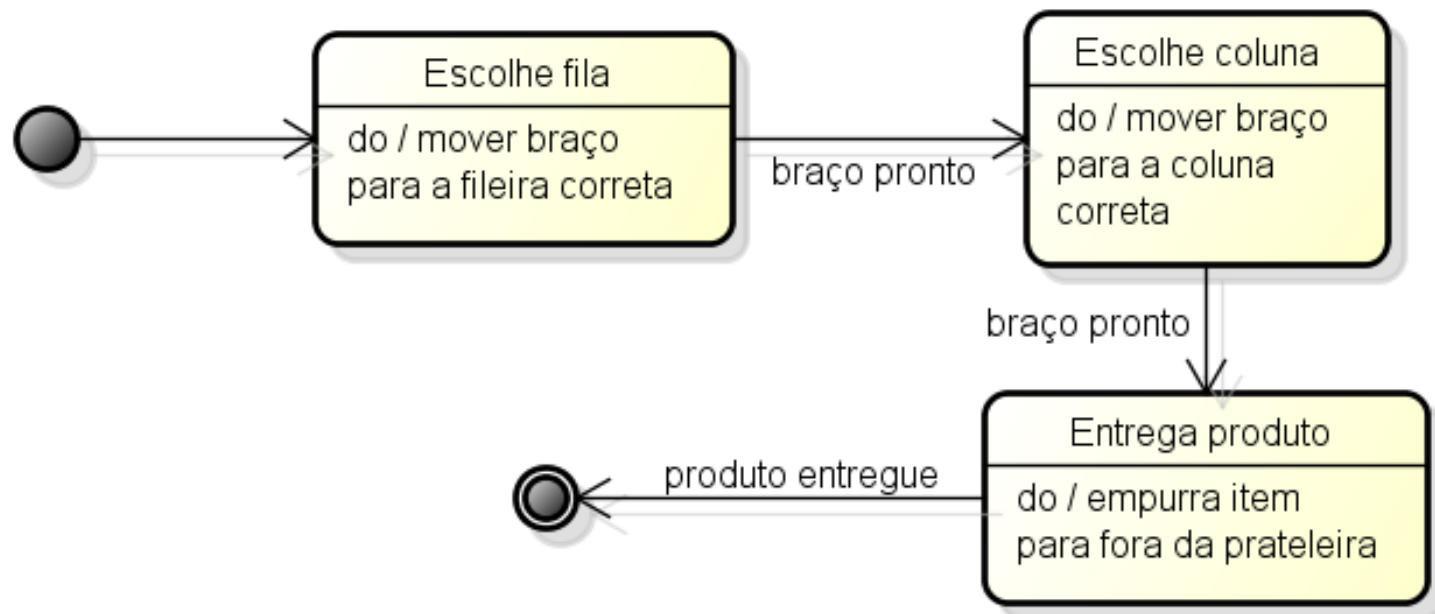


Diagrama de Estados

- Ações de Entrada:

- Descrição de operações que devem ser realizadas assim que o fluxo alcança um dado estado;
- Representação textual após a expressão “entrada /” (ou “*entry* /”)

- Ações de Saída:

- Descrição de operações que devem ser realizadas assim que o fluxo for sair de um dado estado para outro;
- Representação textual após a expressão “saída /” (ou “*exit* /”);

- Ordem execução:

- Ações da transição que chega // Ação de entrada (*entry*) // Operações (*do*) // Ação de saída (*exit*) // ações da transição de saída

Diagrama de Estados

- Ações Internas:

- Descrição de transição que ocorre sem causar uma mudança de estado;
- Representação textual que aparece na parte inferior de um estado;
- Parecido com uma auto transição, mas não igual!

Qual a diferença?

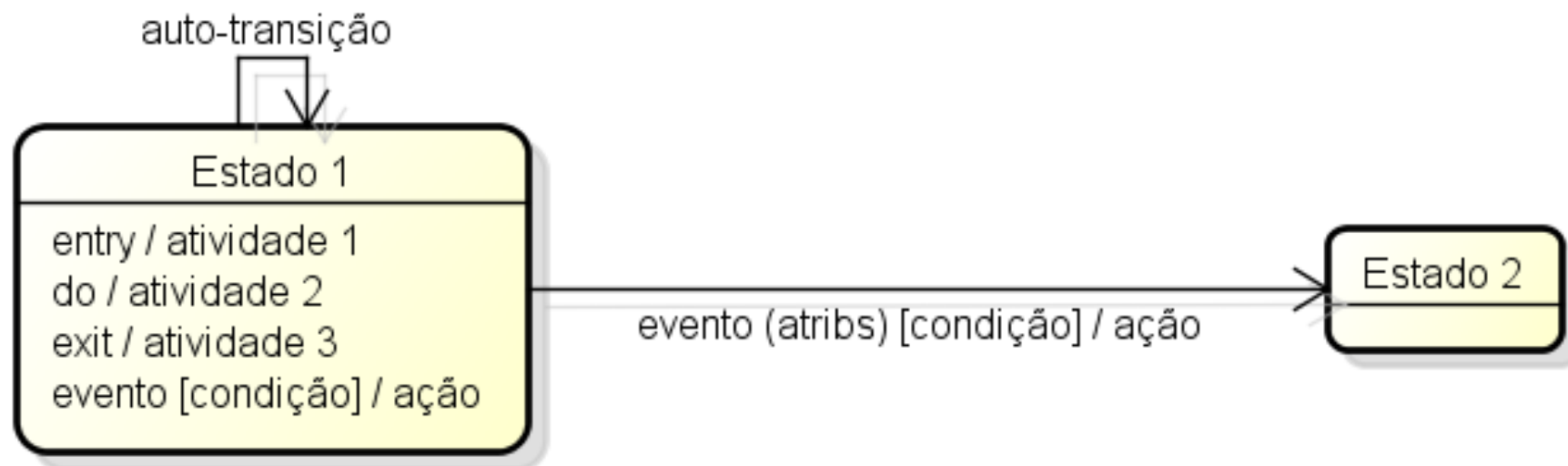
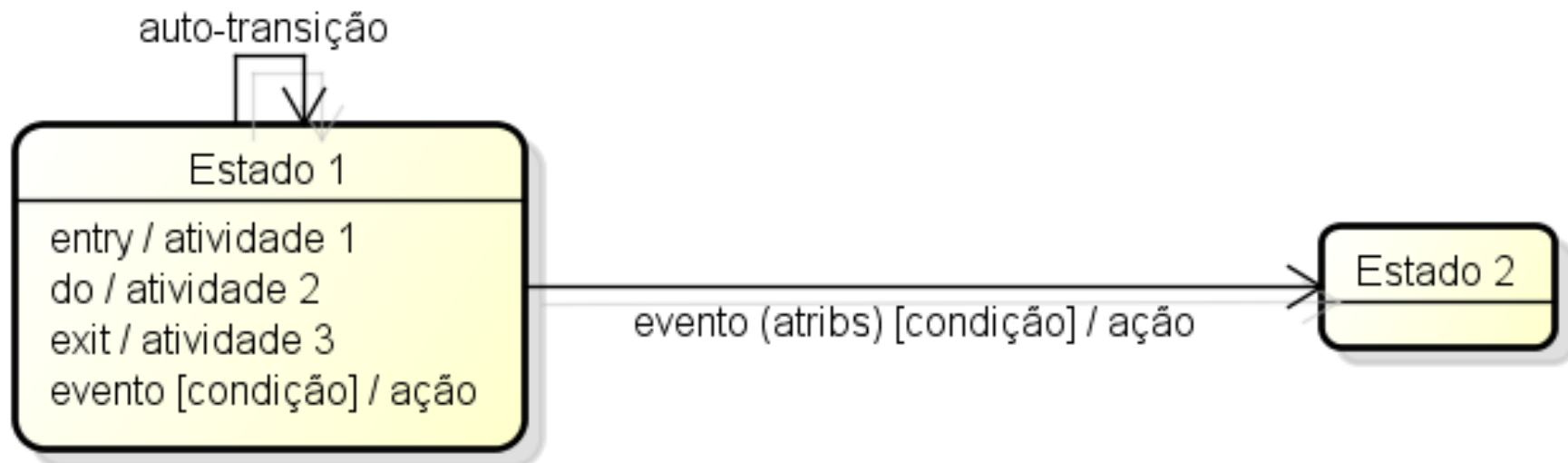


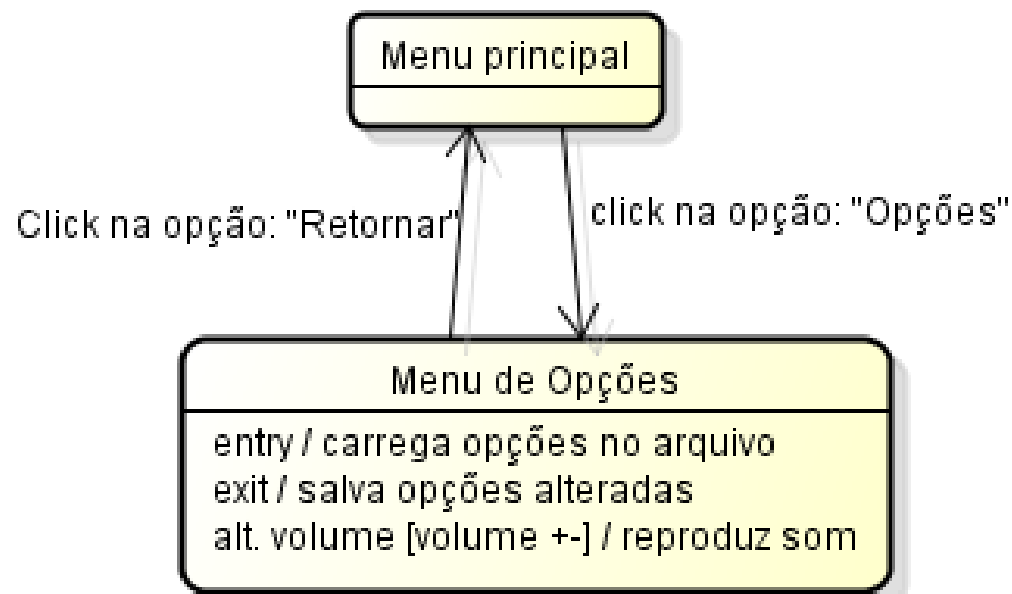
Diagrama de Estados

- Ações Internas:

- Descrição de transição que ocorre sem causar uma mudança de estado;
- Representação textual que aparece na parte inferior de um estado;
- Parecido com uma auto transição, mas não igual!
 - ❖ Em ações internas as operações de entrada e saída não são realizadas
 - ❖ Em auto transições as operações de entrada e saída são realizadas



Exemplo: Salvar alterações realizadas no menu de opções



Exemplo

Gerar um diagrama de estados para um sistema acadêmico que permite o controle e gerenciamento de matrícula, frequência e desempenho dos discentes e a organização das disciplinas ofertadas. O sistema acadêmico deverá permitir que os acadêmicos realizem suas matrículas nas turmas de disciplinas disponíveis, considerando restrições de pré-requisitos, número máximo de créditos (9) e limite de alunos por turma. Deverá permitir que chefes de departamento incluam novas disciplinas e novos professores, abram novas turmas para as disciplinas existentes com sala, horário, lotação máxima e professor definidos. As disciplinas só poderão ser ofertadas entre 7:30 e 12:00, e, 13:30 e 21:40, em blocos de 50 minutos por aula (hora-aula). Também deverá ser possível que professores acessem suas turmas e registrem frequência e notas para seus alunos.

Exemplo

O sistema deverá ter uma opção para finalizar o semestre, possibilitando a inclusão das notas de exame. Um aluno deverá ter frequência superior a 75% e deverá ter uma média superior a 3 para realizar exame. Caso sua nota seja maior ou igual a 7 está aprovado (desde que tenha a frequência necessária). Após a digitação das notas de exame o professor deverá finalizar a turma e o sistema mostrará o resultado final. O sistema deverá funcionar nos sistemas operacionais Windows e Linux e deverá ter seu acesso controlado por login e senha.

(adicionar atributos que considerar relevantes ao problema)

Atividade

Agora é a sua vez!

Construa um diagrama de estados para o sistema descrito no documento que está disponível na página da disciplina.

Bibliografia

- **Básica:**

BEZERRA, E. Princípios de Análise e Projetos de Sistemas com UML. Rio de Janeiro: Campus, 2003.

PRESSMAN, R.S. Engenharia de Software. São Paulo: Makron Books, 2002.

SOMMERVILLE, I. Engenharia de Software. São Paulo: Addison Wesley, 2003.

- **Complementar:**

WARNIER, J. Lógica de Construção de Programas. Rio de Janeiro: Campus, 1984.

JACKSON, M. Princípios de Projeto de Programas. Rio de Janeiro: Campus, 1988.

PAGE-JONES, M. Projeto Estruturado de Sistemas. São Paulo: McGraw-Hill, 1988.