



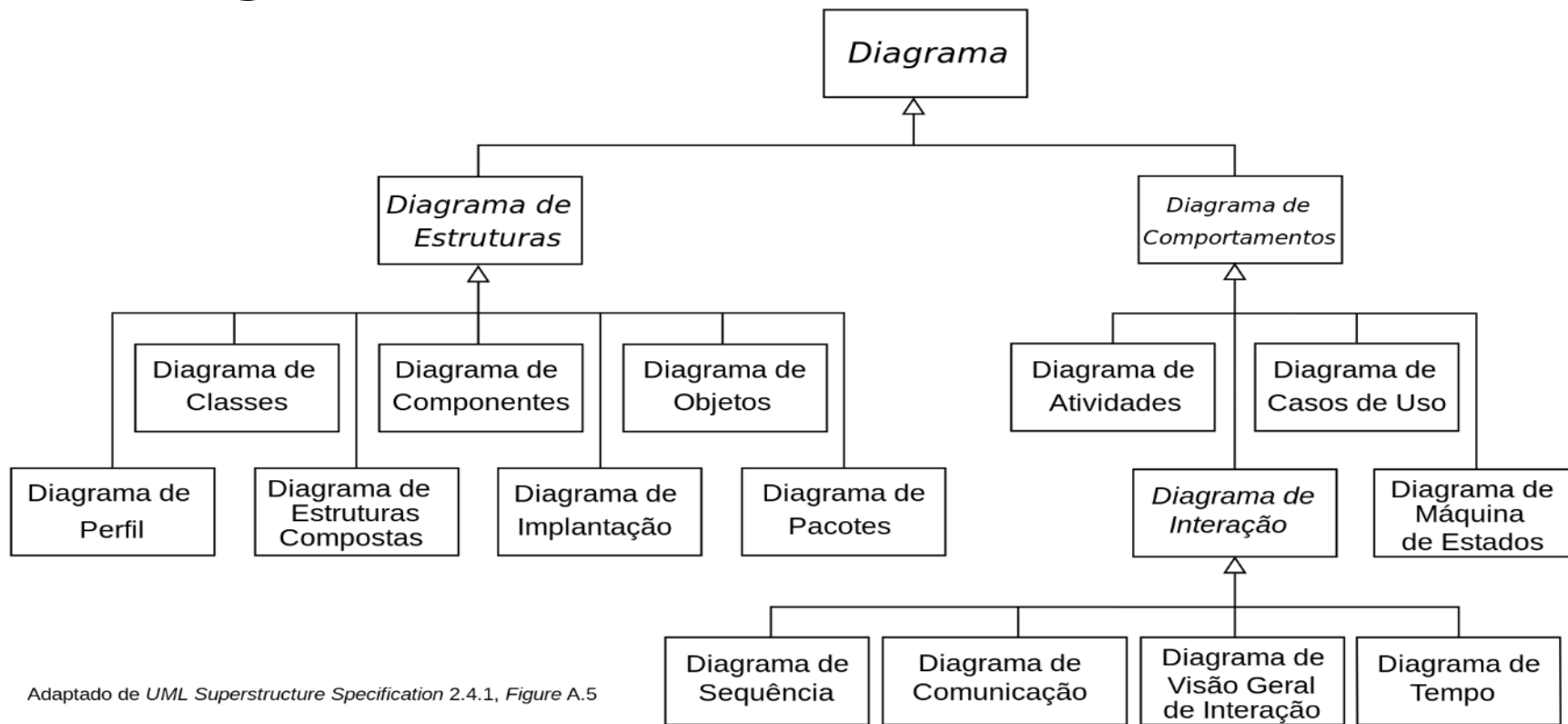
Quem se prepara, não para.

Análise de Sistemas

3º período

Professora: Michelle Hanne

Modelagem de Sistemas em UML



Adaptado de UML Superstructure Specification 2.4.1, Figure A.5

Diagrama de Sequência

O objetivo é demonstrar em linhas de tempo quais são as interações entre os objetos de um determinado cenário representado pelo diagrama. Geralmente um **diagrama de sequência é criado a partir de um diagrama de casos de uso**, com a finalidade de descrever como serão as interações/mensagens entre cada objeto/elemento do diagrama.

Este diagrama possui dois eixos: o eixo vertical que indica a sequência das mensagens e o tempo de vida dos objetos e o eixo horizontal, este último indica quais são os objetos que participam do diagrama.

Diagrama de Sequência - Elementos

Objetos

Os objetos podem **representar atores ou mesmo instâncias de classes do sistema**, desde que tenham alguma interação no cenário do diagrama. Ou seja, geralmente, um objeto vem predefinido de um diagrama de classes.

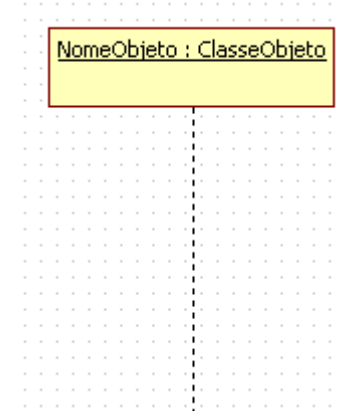


Diagrama de Sequência - Elementos

Call: este é o tipo mais comum, pois representa a chamada de um método de algum objeto (ver figura 3 e 4).

Send: este tipo de mensagem também chama um método, a diferença é que esta é uma mensagem assíncrona, ou seja, a sua execução decorre em paralelo aos demais processos (ver figura 5 e 6).

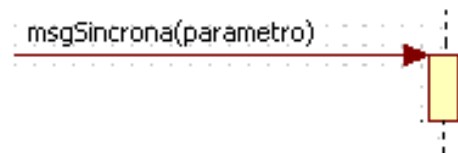


Figura 3 – Representação da mensagem do tipo Call



Figura 4 – Representação da mensagem do tipo Call de auto delegação

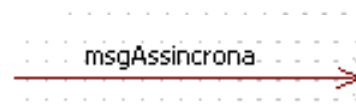


Figura 5 – Representação da mensagem do tipo Send

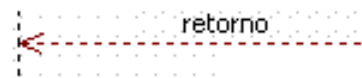


Figura 6 – Representação da mensagem do tipo Return

Diagrama de Sequência - Elementos

Create: indica a criação de um objeto (ver figura 7);

Destroy: indica a eliminação de um objeto (ver figura 8).

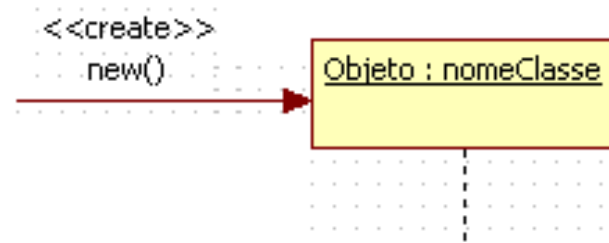


Figura 7 – Representação da mensagem do tipo Create

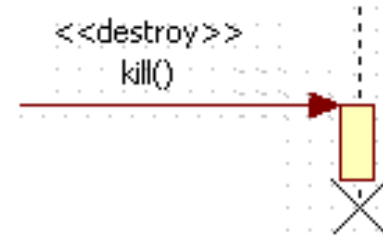
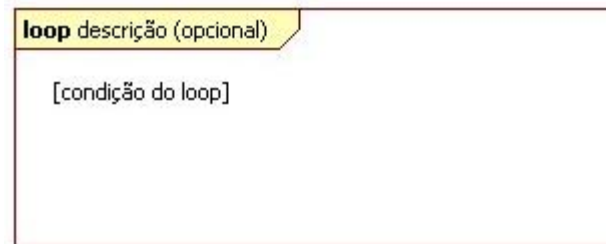


Figura 8 – Representação da mensagem do tipo Destroy

Diagrama de Sequência - Elementos

Frames

Além de Objetos e Mensagens podemos ter ainda no diagrama de sequência o elemento Frame. Este elemento serve para tratar alguns conceitos mais avançados, como estruturas de repetição e de decisão.



Loop

Como o próprio nome já indica este tipo de frame indica que as mensagens que estiverem dentro do escopo deste frame farão parte de uma estrutura de repetição considerando alguma condição, que também deve estar descrita no frame

Diagrama de Sequência - Elementos

Opt

Este tipo de frame funciona como um **If na programação, portanto logicamente, deve-se especificar uma condição para executar as mensagens dentro do frame,,**



Alt

Este tipo de frame funciona de forma **semelhante ao Opt**, a diferença é que este tipo dá a opção de utilizar o um **Else, ou seja, caso a condição não seja atendida, pode-se executar outras mensagens.**

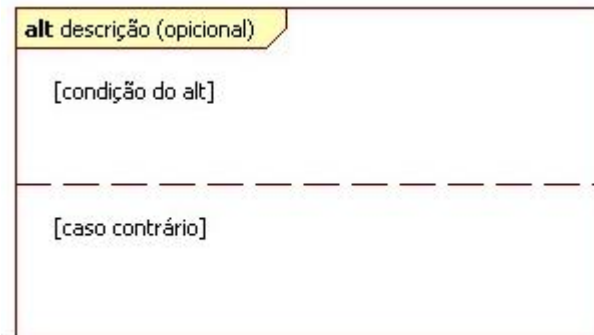


Diagrama de Sequência - Exemplo

O usuário que chama o método **informaDados** do objeto **telaLogin**, fornecendo os dados de **login e senha**. Em seguida, **telaLogin** envia uma mensagem com o objetivo de consultar os dados do **usuário, logicamente passando as informações de login e senha**, onde o retorno desta mensagem será armazenado em **usuarioOk**. Na sequência, se **usuarioOk** for igual a **True**, a **telaLogin** envia uma mensagem do tipo **Create** para o objeto **telaPrincipal**. Por fim o objeto **telaLogin** envia uma auto mensagem do tipo **Destroy**.

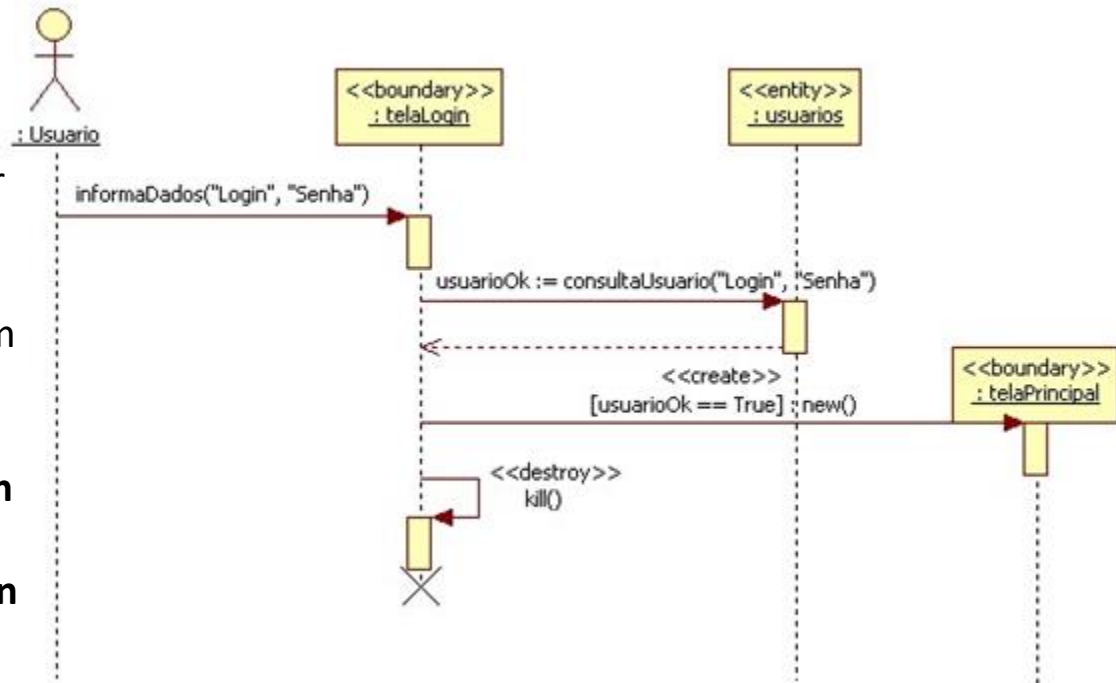


Diagrama de Sequência - Exemplo

Adaptação feita no diagrama de exemplo Fazer Login utilizando alguns frames. Utilizando um Loop para validar o usuário e um Alt para exibir se o login deu certo ou não.

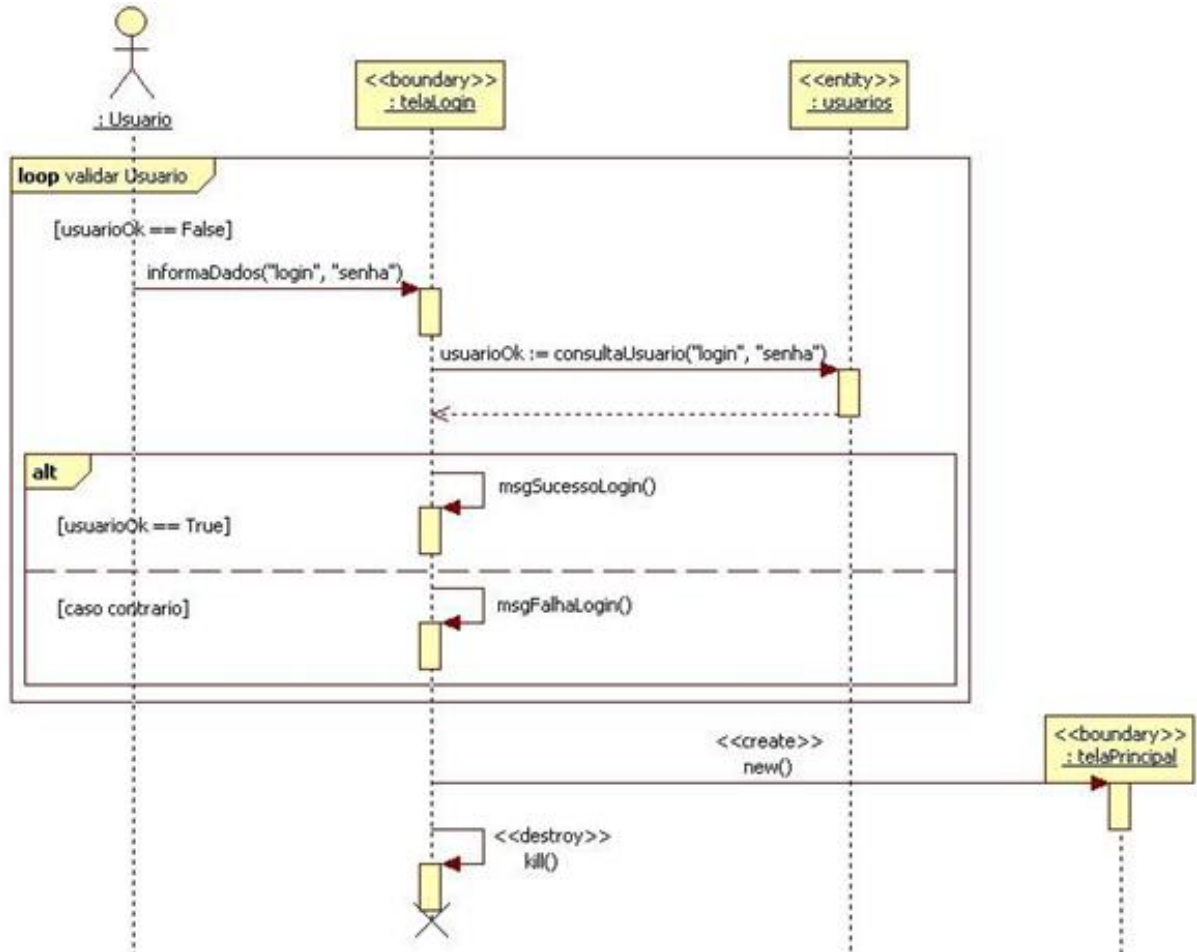


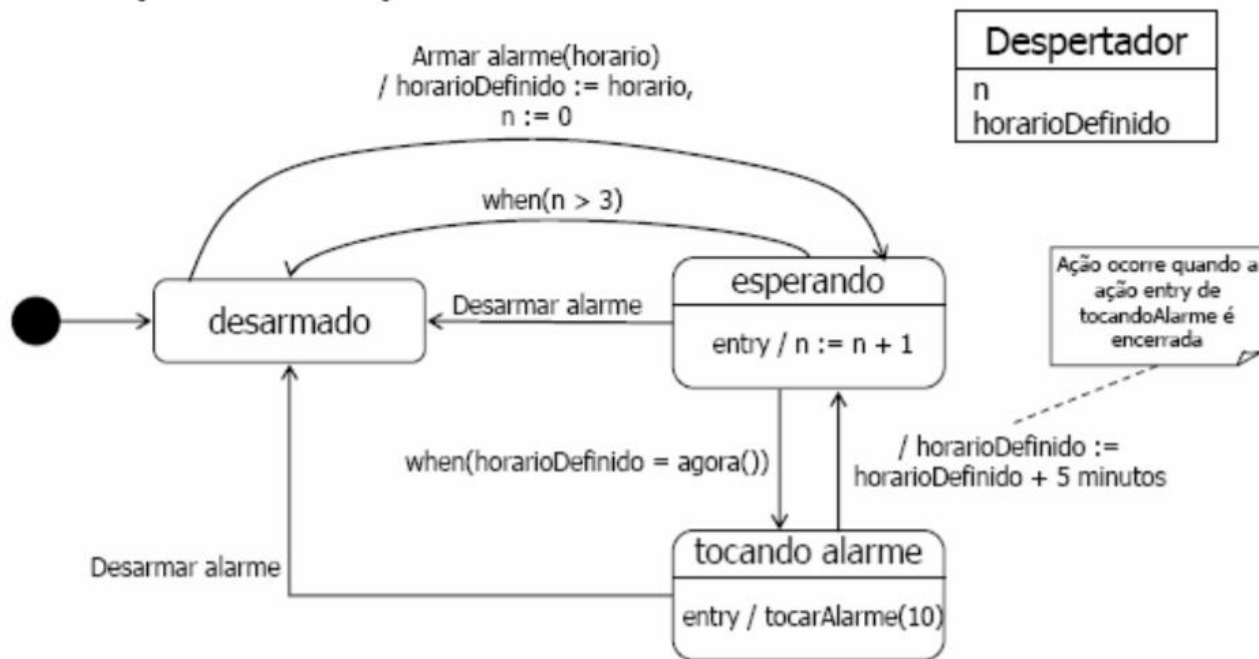
Diagrama de Máquina de Estado

■ Exemplo: Despertador



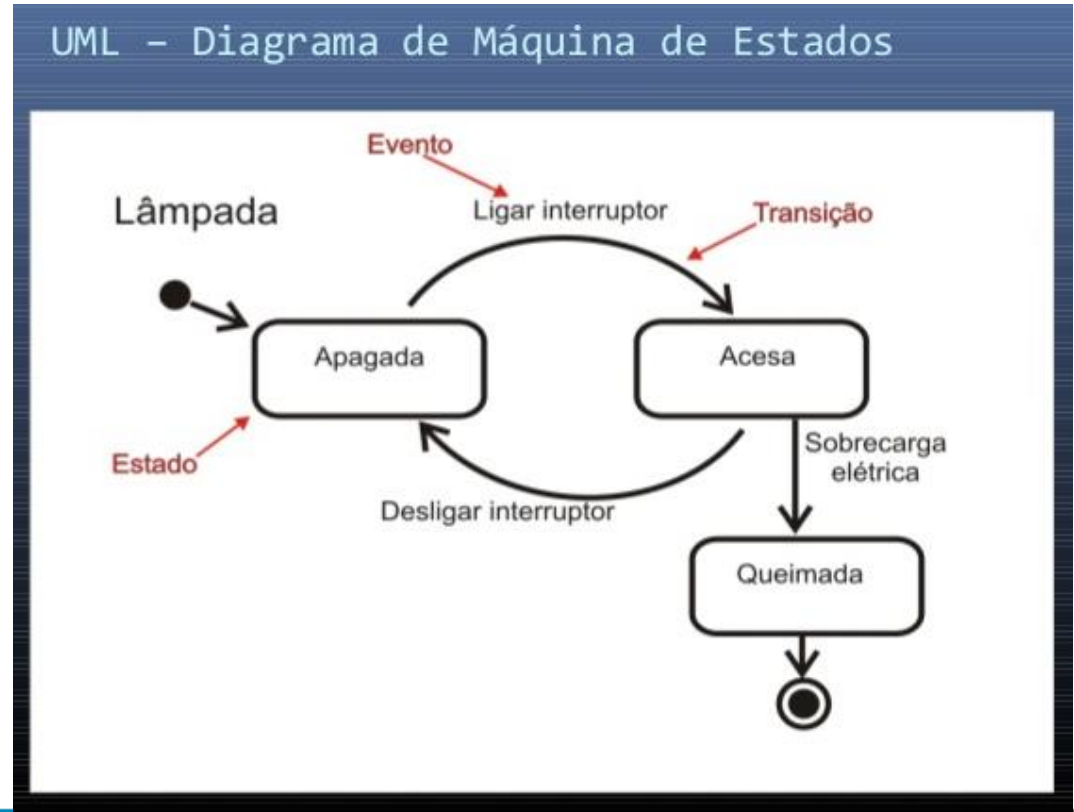
Diagrama de Máquina de Estado

■ Exemplo: Despertador



Exemplo Diagrama de máquina de estado

O estado de um objeto é a condição ou situação que o objeto assume durante a sua vida, a qual ele satisfaz a alguma condição, realiza alguma atividade ou aguarda algum evento (BOOCH; JACOBSON; RUMBAUGH, 2016).



Diagramas de estados para classes de análise

O componente de um modelo comportamental é um diagrama de estados em UML, que representa **estados ativos para cada uma das classes** e para os eventos (gatilhos) que provocam mudanças entre esses estados ativos.

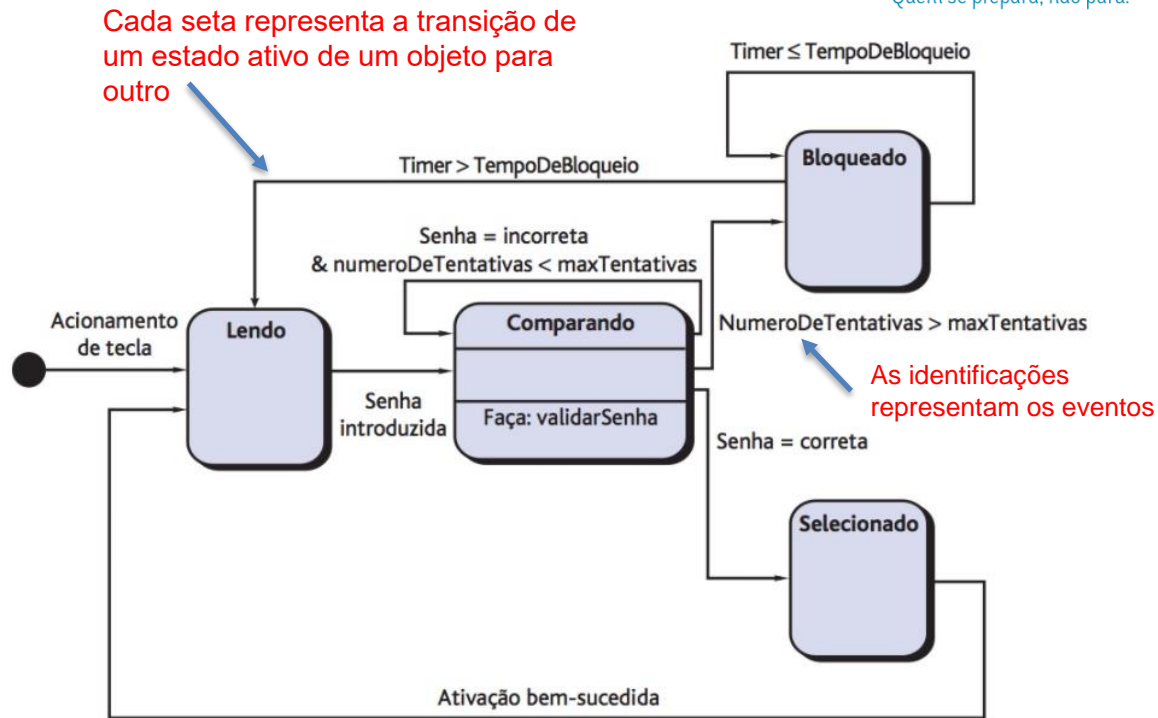


FIGURA 11.1 Diagrama de estados para a classe PainelDeControle.

Casa Segura

Continuando com o exemplo da função de segurança domiciliar do Casa- Segura, poderíamos definir o conjunto de componentes de alto nível que trata da seguinte funcionalidade:

- **Gerenciamento da comunicação externa** – coordena a comunicação da função de segurança com entidades externas, como sistemas baseados na Internet e notificação externa de alarme.
- **Processamento de painel de controle** – gerencia toda a funcionalidade do painel de controle.
- **Gerenciamento de detectores** – coordena o acesso a todos os detectores conectados ao sistema.
- **Processamento de alarme** – verifica e atua sobre todas as condições de alarme.

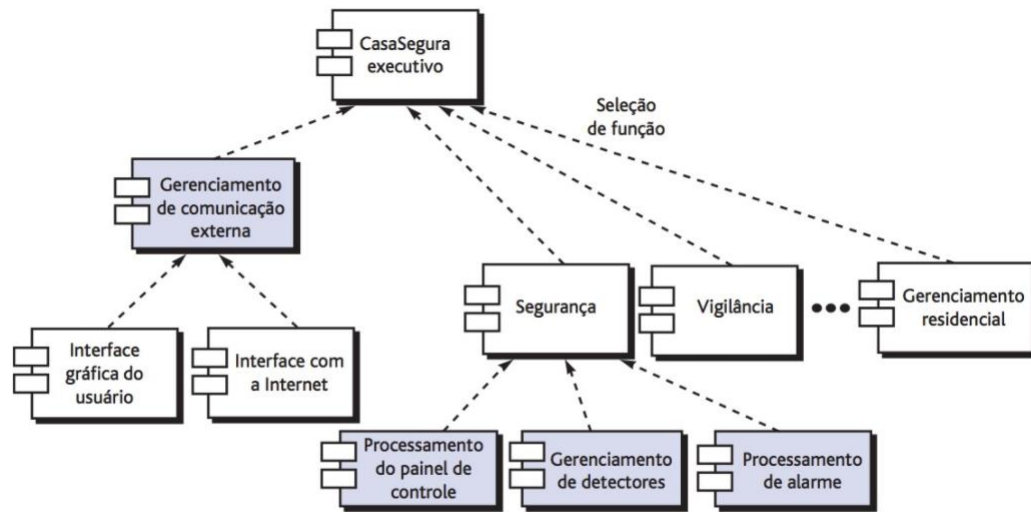


FIGURA 13.8 Estrutura arquitetural global para o sistema CasaSegura com os componentes de alto nível.

Casa Segura

A Figura 13.8 mostra a estrutura global da arquitetura (representada na forma de um diagrama de componentes UML). As transações são capturadas pelo gerenciamento de comunicação externa, à medida que se deslocam de componentes que processam a interface gráfica do usuário do CasaSegura e a interface com a Internet. Tais informações são gerenciadas por um componente executivo do **CasaSegura** que seleciona a função do produto apropriada (neste caso, segurança).

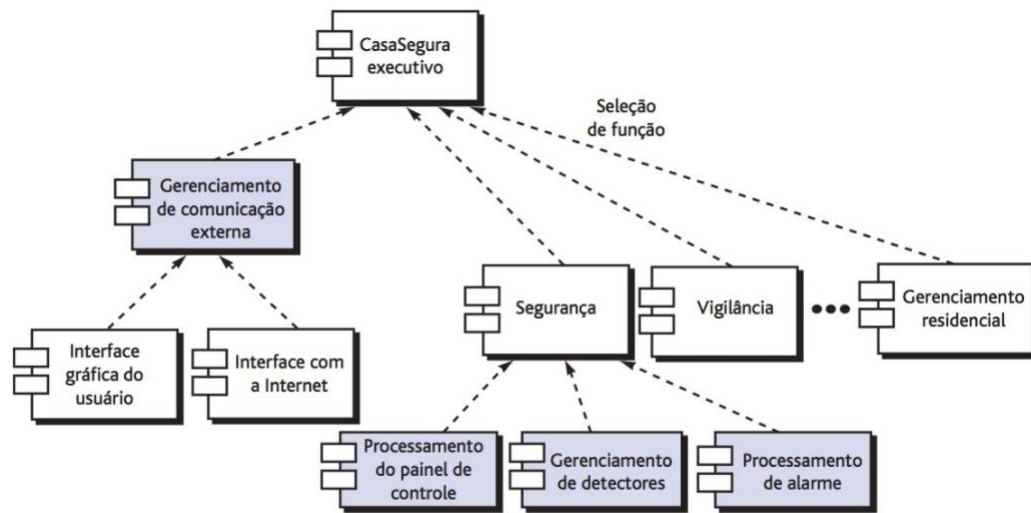


FIGURA 13.8 Estrutura arquitetural global para o sistema CasaSegura com os componentes de alto nível.

Diagrama de Sequência - Exemplo

A Figura 11.2 ilustra um diagrama de sequência parcial para a função de segurança do CasaSegura.

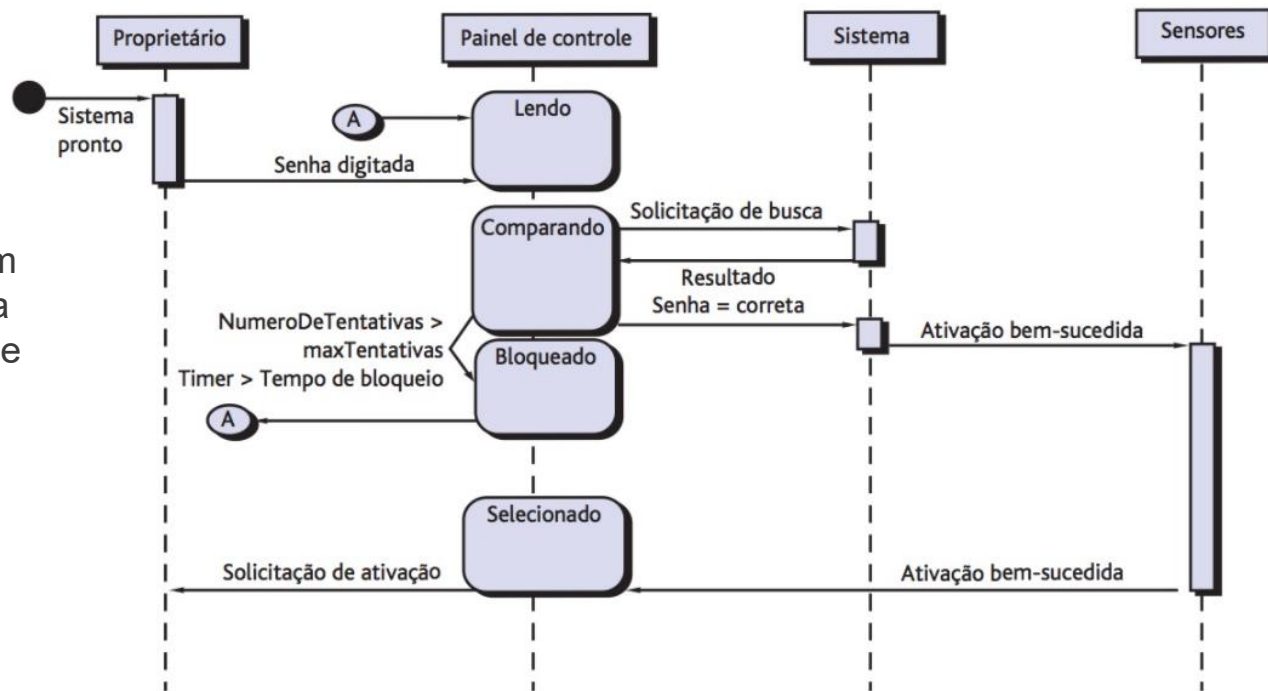


FIGURA 11.2 Diagrama de sequência (parcial) para a função de segurança do CasaSegura.

Diagrama de Sequência - Exemplo

O primeiro evento, sistema pronto, é derivado do ambiente externo e canaliza comportamento para o **objeto Proprietário**. O proprietário do imóvel digita uma senha. Um evento **solicitação de busca** é passado para **Sistema**, o qual busca uma senha em um banco de dados simples e **retorna um resultado** (encontrada ou não encontrada) para **PainelDeControle** (agora no estado comparando). Uma senha válida resulta em um evento **senha=correta** para **Sistema**, o qual ativa **Sensores** com um evento solicitação de ativação. Por fim, o controle retorna ao proprietário com o evento ativação bem-sucedida.

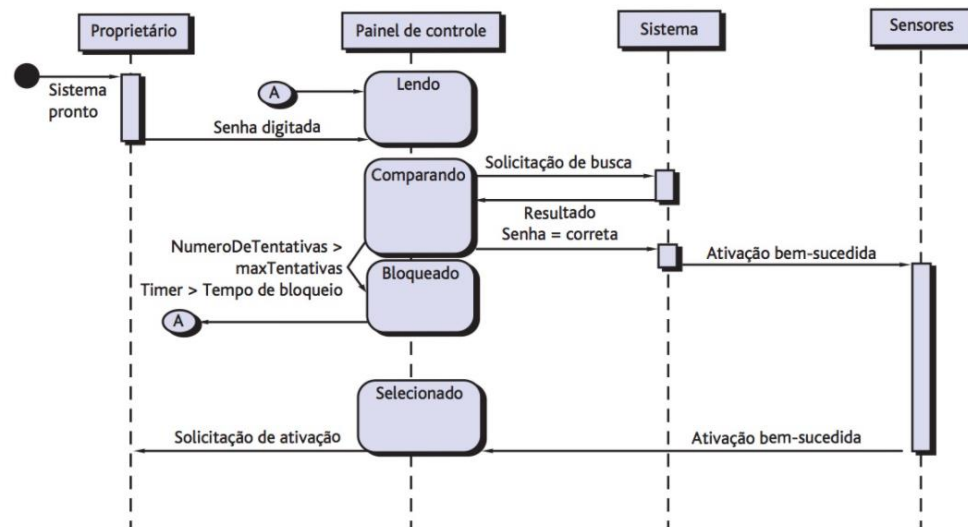


FIGURA 11.2 Diagrama de sequência (parcial) para a função de segurança do CasaSegura.

Referências

OMG, UnifiedModelingLanguage2.5.1, 2017. Disponível em
<<https://www.omg.org/spec/UML/2.5.1/PDF>>. Acesso em 04 de fevereiro de
2020

PRESSMAN, Roger S. MAXIM, Bruce R. **Engenharia de Software - Uma Abordagem
Profissional**. 8.ed. Porto Alegre: Amgh Editora, 2016. 968p. ISBN 9788580555332.

SOMMERVILLE, Ian. **Engenharia de software**. 8.ed. São Paulo: A. Wesley publishing
company, 2010.

.