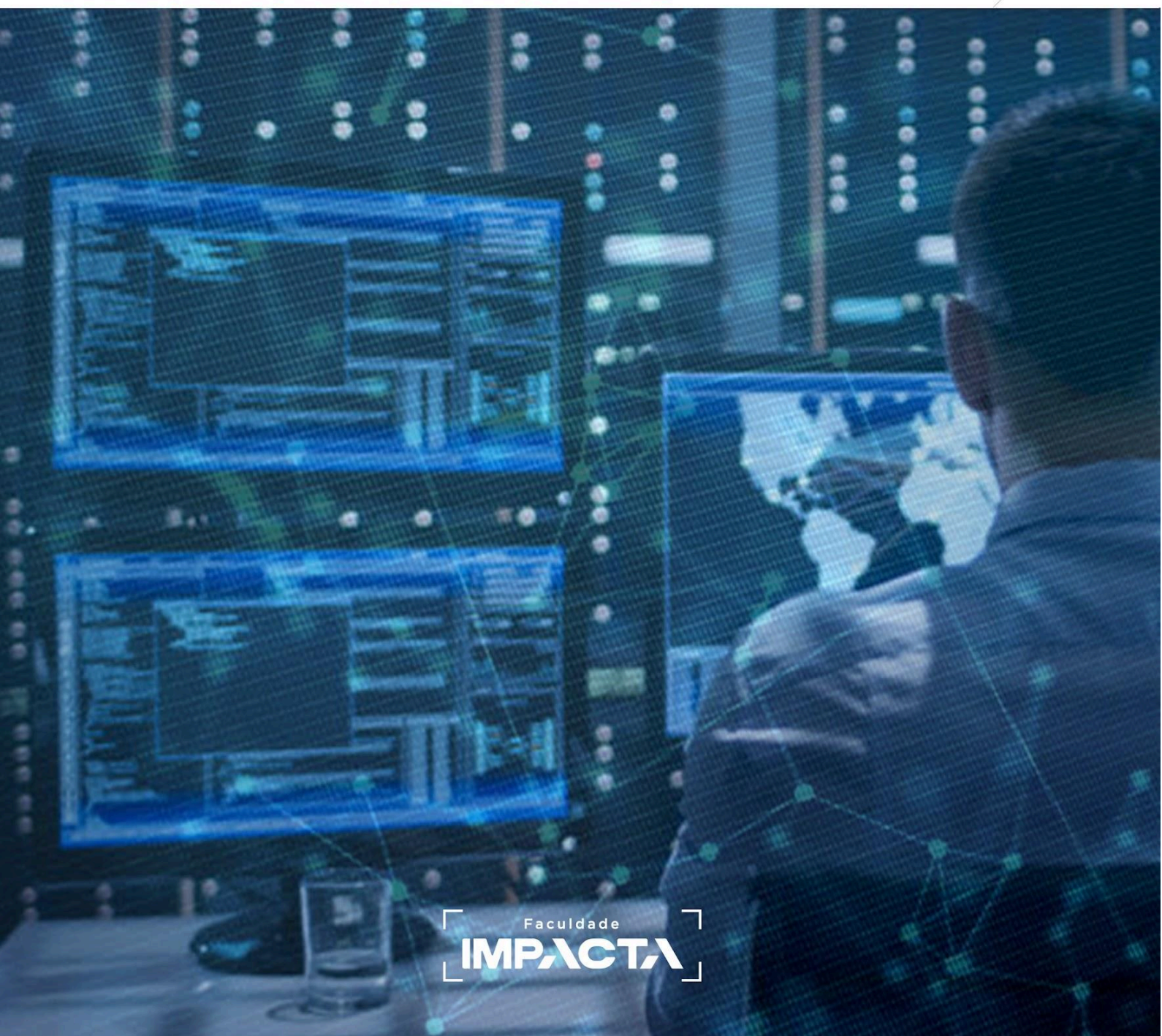




10

TEXTO BASE

ANÁLISE E MODELAGEM DE SISTEMAS



Texto base

10

Diagrama de Sequência

Definição, Elementos e Modelagem

Prof. Renato de Tarso Silva

Resumo

Neste texto será abordado o diagrama de sequência. Será explanada a sua definição e sua utilização. Conheceremos seus elementos e como aplicá-los a fim de demonstrar a sequência de operações e estruturas necessárias ao realizar um processo.

1.1. Conceito

Este diagrama da UML, por ser um dos tipos de diagramas comportamentais e por também utilizar estruturas sistêmicas, é motivado pelas descobertas e análises tanto do diagrama de casos de uso e quanto do diagrama de classes do sistema. Este diagrama representa de forma objetiva como será implementado um processo no sistema.

É essencial a utilização, como referência, dos fluxos de casos de uso para concepção do diagrama de sequências, pois lá estão passos sequenciais dos comportamentos

1.2. Definição

O diagrama de sequência denota, em ordem temporal, as mensagens trocadas entre os objetos de um sistema, os serviços solicitados entre os objetos (instâncias de classes), e as respostas implementadas em seus métodos.

A sequência de mensagens são respostas a eventos em sequência que fazem parte de um ou mais processos executados em fluxos de cenário do sistema, fluxos estes presentes em cada caso de uso, utilizando propriedades de objetos. Note um exemplo deste diagrama, com troca de mensagens e entre objetos, na figura 1.2.1.

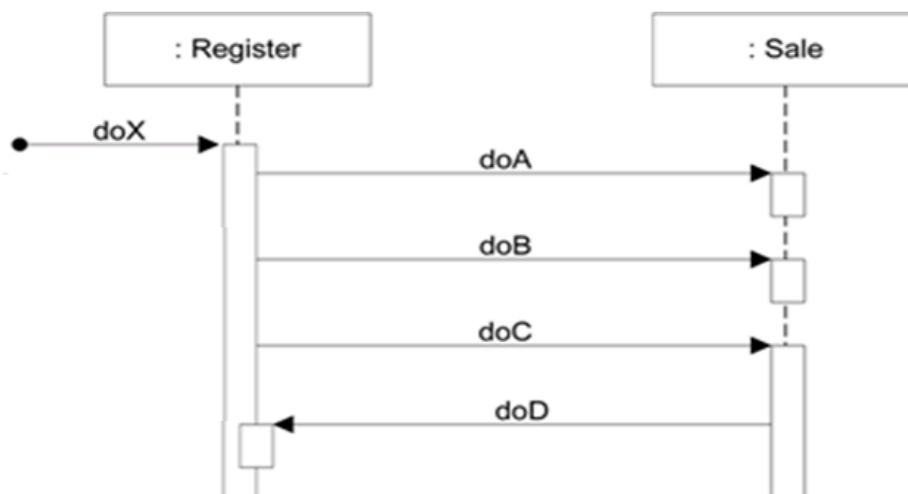


Figura 1.2.1. Exemplo de Diagrama de Sequência

No diagrama acima, o objeto “**:Register**” recebe uma solicitação inicial, através da mensagem “**doX**”, que aciona o método “doX” que o objeto “:Register” contém. Em seguida a mensagem “**doA**” é enviada para o objeto “**:Sale**”, assim ativando o método “doA” que “:Sale” disponibiliza como serviço a operação “doA”. Assim por diante as solicitações (mensagens) são trocadas entre objetos na sequência.

O diagrama de sequência dá a visão de quando e como os objetos interagem para finalizar determinados processos. Booch (2005) afirma o seguinte: *Um diagrama de interação é basicamente uma projeção dos elementos encontrados em uma interação. A semântica de um contexto da interação, objetos e papéis, vínculos, mensagens e sequências se aplica aos diagramas de interação.*

1.2. Elementos

O diagrama de sequência referencia elementos encontrados no diagrama de casos de uso, como o ator, e utiliza propriedades (atributos e métodos) dos objetos especificados no diagrama de classes. Além de sinais de interação e sequência do diagrama próprio, como a “linha de vida” (lifeline), bloco de execução e conta com elementos de controle que permitem representar ou reusar comportamentos comuns.

- **Elemento Linha de Vida:** A linha tracejada e vertical traça um período de tempo para a existência de um objeto. Comumente objetos contém existência igual à da interação da sequência toda e ficando alinhados na parte superior do diagrama, mas objetos podem ser criados (*create*), ou destruídos, no decorrer da sequência (*destroy*). Veja na figura 1.2.1.1 uma representação da *lifeline*.

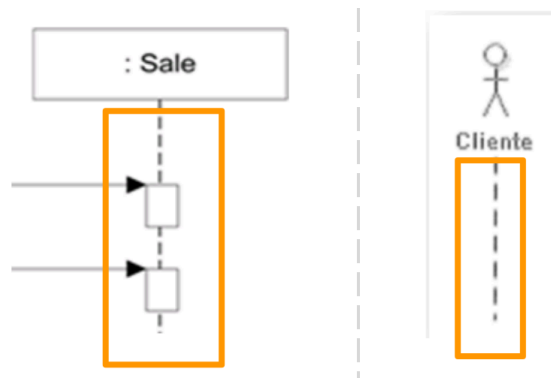


Figura 1.2.1.1. Linha de Vida - Lifeline

- **Elemento Ator:** Concebido no diagrama casos de uso, ele representa um usuário ou um sistema que participa da interação. Veja na figura 1.2.1.2 um exemplo de ator referenciado na sequência.

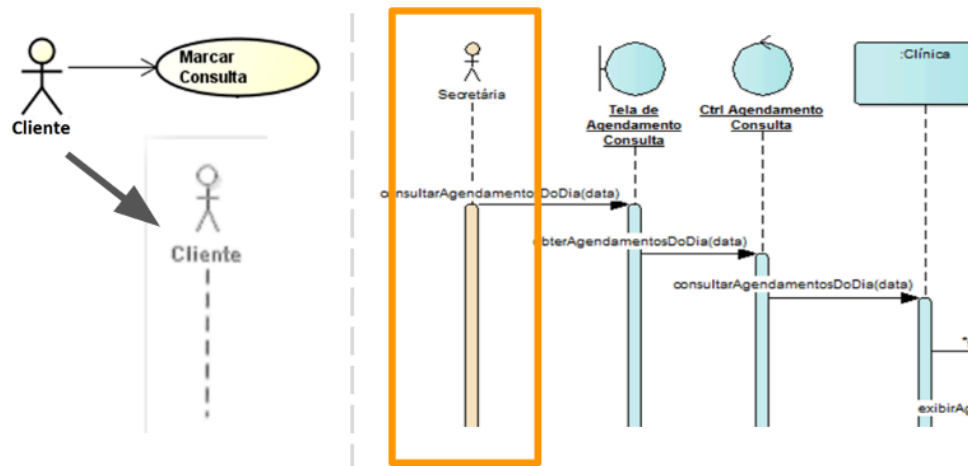


Figura 1.2.1.2. Ator

- **Elemento Objeto:** Representa uma instância de uma classe, portanto é um objeto que atende ou solicita serviços durante as trocas de mensagens da sequência especificada. Note na figura 1.2.1.3 os objetos :Sale e :Register.

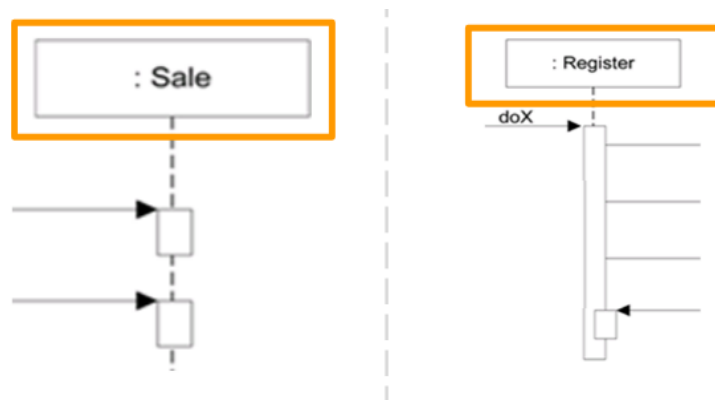


Figura 1.2.1.3. Objeto

- **Elemento Bloco/Ativação de Execução:** Representa o trecho de período em que um objeto participando ativamente, executando métodos, no processo sequenciado no diagrama. Note na figura 1.2.1.4 os objetos *:Sale* e *:Register*.

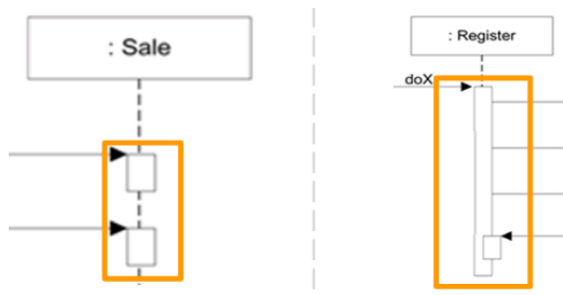


Figura 1.2.1.4. Bloco/Ativação de Execução

- **Elemento Mensagem:** Representa um evento de solicitação ou resposta. É uma chamada de um método do objeto de destino, denotada com uma seta em direção ao objeto que consegue atender a mensagem passada, um método.

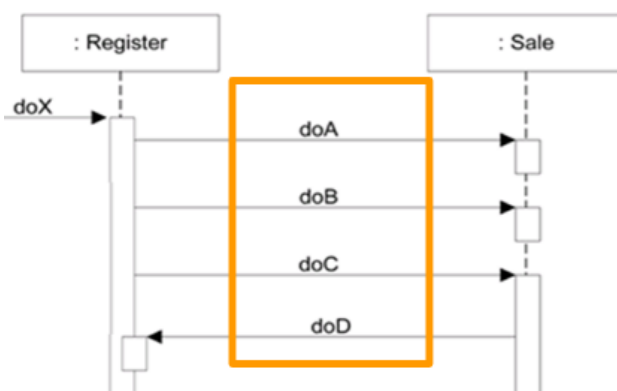


Figura 1.2.1.5. Mensagem

- **Elemento Controle Opção (OPT):** Representa uma condição ou decisão lógica que permite variar o comportamento do processo em função de ser verdade o resultado de uma expressão, como é demonstrado na figura 1.2.1.6. O bloco de opção somente solicitará *calculate* se o valor *color=red*.

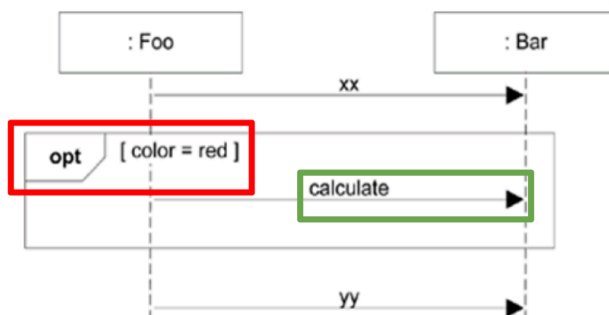


Figura 1.2.1.6. Opção - OPT

- **Elemento Alternativo (ALT):** Permite a alternância entre eventos da sequência, escolhendo mensagens diferentes através de decisão lógica. Note o exemplo da figura 1.2.1.7, onde caso seja verdadeira a expressão “ $x < 10$ ” a mensagem *calculate* é solicitada ao objeto :B, senão, é solicitada *calculate* do objeto :C.

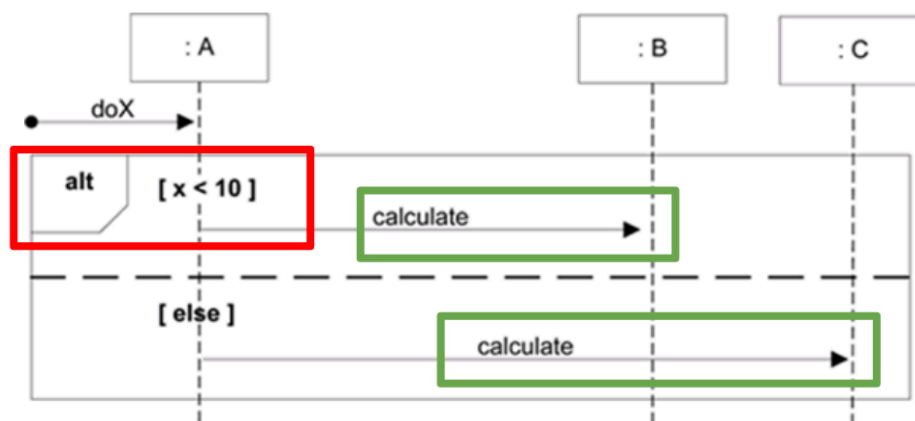


Figura 1.2.1.7. Alternativo - ALT

- **Elemento Laço (LOOP):** Permite a criação de iteração de repetição através de decisão lógica, onde seja necessário representar um laço para determinadas mensagens. O exemplo da figura 1.2.1.8 representa um laço onde enquanto houver “more items” a mensagem “enterItem()” é executada no objeto :B.



Figura 1.2.1.8. Laço - LOOP

Serão mostrados exemplos em um formato similar ao MVC (Model, View Control). O MVC serve para separar em camadas de responsabilidade o funcionamento do software. O *Model* seria o modelo (um estrutura, um objeto); a *View* seria a visualização (a interface de interação, como fronteira entre o interno e o externo do sistema); e o *Control* quem controla o processo na sequência, atendendo ou utilizando os objetos(*model*) e as interfaces de interação (*view*). Conheça melhor os elementos.

- **Elemento Fronteira (Boundary):** Serve para denotar uma interface que permitirá a interação com as entidades (usuários ou sistemas) externas ao sistema especificado. Ex: telas de sistema, formulários com campos, janelas de mensagens, botões, teclas. O *boundary* é representado no diagrama de sequências pelo elemento ilustrado na figura 1.2.1.9.

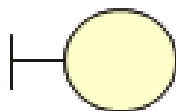


Figura 1.2.1.9. Fronteira - Boundary

- **Elemento Controle (Control):** Representa uma entidade de controle, que no sistema terá a responsabilidade de intermediar as trocas de mensagens entre objetos, e de executar operações necessárias para o cumprimento do processo da sequência. O *control* é representado no diagrama de sequências pelo elemento ilustrado na figura 1.2.1.10.



Figura 1.2.1.10. Controle - Control

- **Elemento Entidade (Entity):** São as instâncias das classes, ou seja, são os objetos que participam da execução de operações necessárias para o cumprimento da sequência e seus eventos. O *entity* é representado no diagrama de sequências pelo elemento ilustrado na figura 1.2.1.11.



Figura 1.2.1.11. Entidade - Entity

A utilização destes elementos têm a incumbência de isolar as camadas sistêmicas, e permitam melhor esclarecer a segregação existente e necessária nas tratativas de solicitações que variam entre a exibição, o processamento e a persistência de processos. Este esquema propõe que entidades (objetos) não interajam diretamente com fronteiras, e menos ainda com atores.

Veja o exemplo destes elementos na figura 1.2.1.12. O ator *student* solicita a mensagem *new* na *boundary* **:StudentPage**, que por sua vez solicita e encaminha a solicitação ao *control* **:TranscriptBuilder**, que é responsável por finalmente solicitar *getSeminars()* à entidade **:Student**, e por solicitar, dentro de um loop, a mensagem *getMark()* ao objeto **:Seminar**.

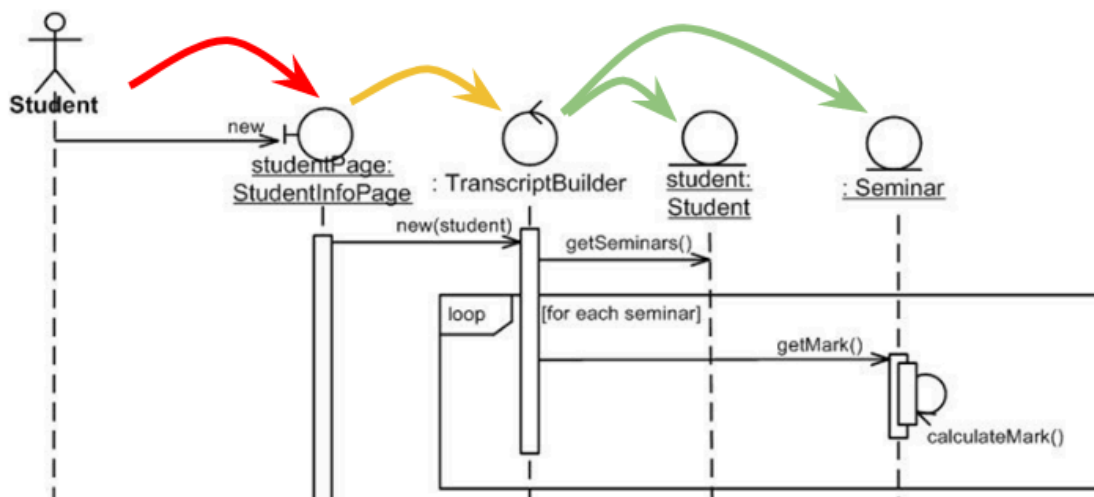


Figura 1.2.1.12. Interface, Controle e Entidade

Para a notação de entidades (objetos), podem ser utilizadas as duas notações, tanto o retângulo com o **:NomeObjeto**, quanto o círculo com a linha e o **:NomEntidade** na base. Veja o exemplo disso na figura 1.2.1.13.

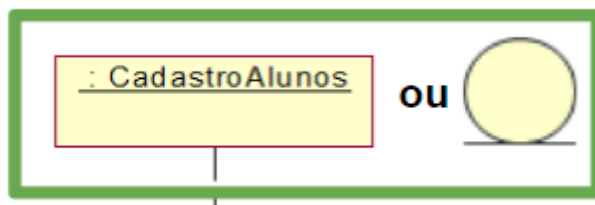


Figura 1.2.1.13. Notações válidas para Objeto / Entidade

Referências

Booch G., Rumbaugh J., Jacobson I (2005). “The Unified Modeling Language user Guide” 2ª Edição.