UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

Análise e Projeto de Software

Universidade Evangélica de Goiás

Curso de Engenharia de Software

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

Análise e projetos de Software: Análise Orientada a Objeto

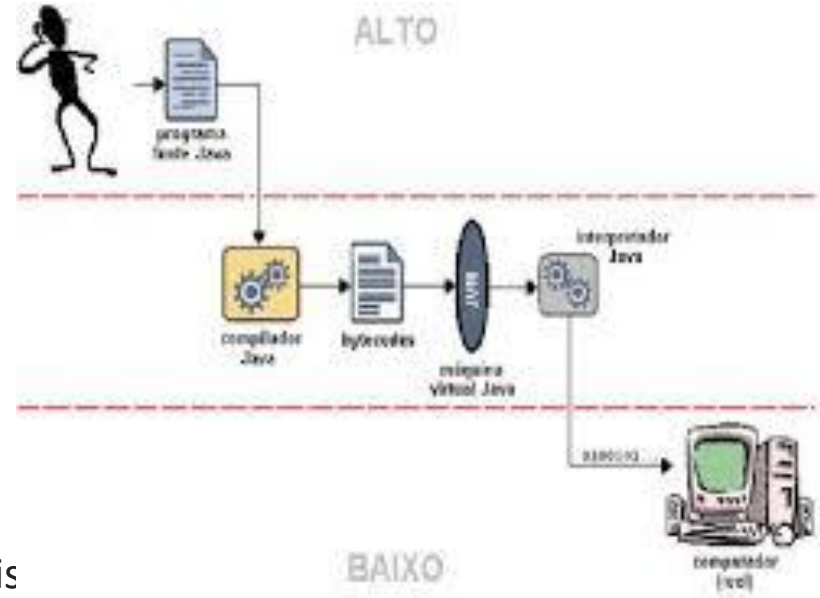
Introdução

A análise orientada a objetos tem a finalidade de definir as classes que são importantes ao problema a ser resolvido, descobrindo seus atributos e operações, as relações existentes entre elas e o comportamento exibido por elas. As tarefas necessárias para alcançar tais objetivos são:

1. levantar os requisitos básicos junto aos usuários;
2. identificar as classes (atributos e operações);
3. especificar uma hierarquia de classes;
4. representar as relações entre objetos;
5. modelar o comportamento dos objetos;
6. aplicar as tarefas anteriores repetidamente (interativamente) até que o modelo seja obtido.

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

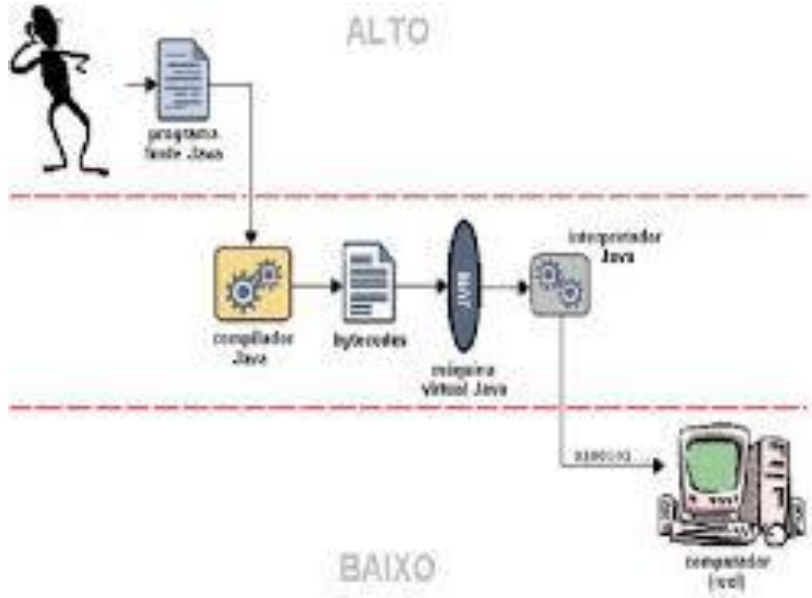
Abordagem



No final dos anos 80 e durante os anos 90 vários métodos de análise orientada a objetos foram sugeridos. Estes métodos eram compostos por um processo de anális um conjunto de diagramas e uma notação utilizada para criar o modelo de análise. Entre todos, o que mais se destacou, na verdade, foi uma combinação de 3 (três), que propunha a combinação dos métodos idealizados por Grady Booch (Método Booch), James Rumbaugh (Técnica de Modelagem de Objetos (OMT – *Object Modeling Technique*) e Ivar Jacobson (Engenharia de Software Orientada a Objetos (OOSE – *Object-Oriented Software Engineering*). Assim, surgiu a UML (*Unified Modeling Language* – Linguagem de Modelagem Unificada), cujo propósito foi o de agrupar os pontos fortes dos 3 (três) métodos.

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

UML

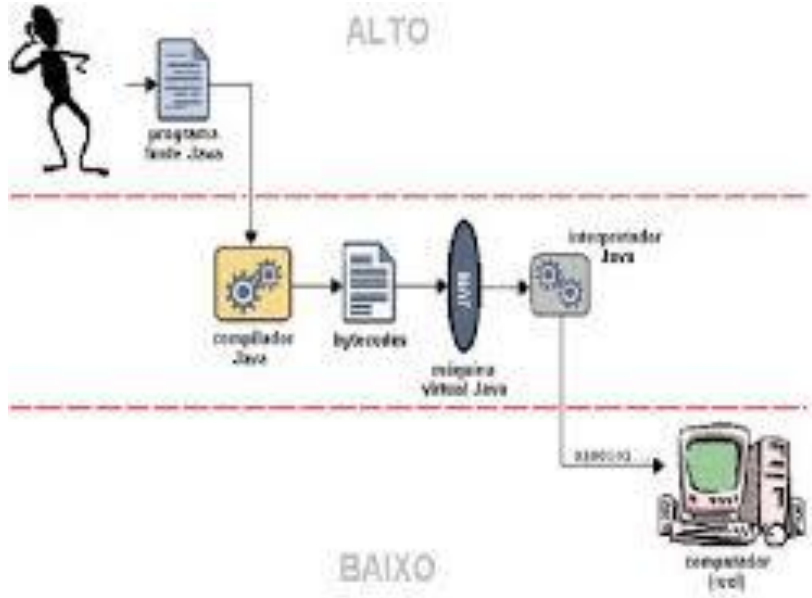


Na UML a modelagem do software é expressa mediante uma notação de modelagem, que é regida pelas seguintes regras:

- Sintáticas: define quais símbolos devem existir e como são combinados para formas sentenças;
- Semânticas: diz respeito ao significado de cada símbolo e como ele é interpretado por si só e no contexto de outros símbolos;
- Pragmáticas: corresponde às regras para a criação de sentenças através da definição das intenções dos símbolos.

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

UML



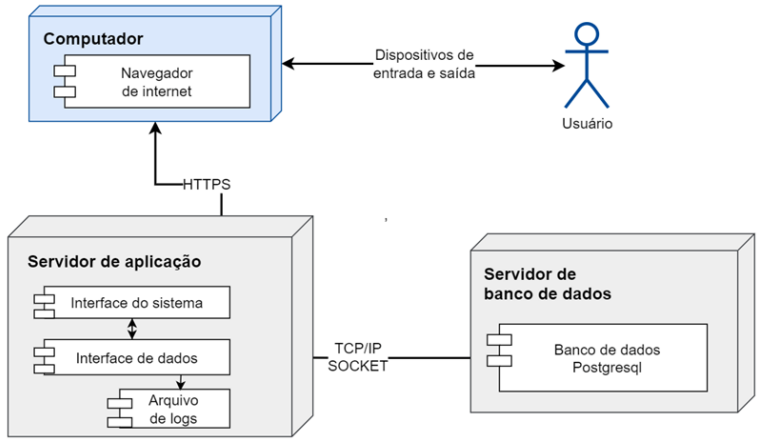
Um sistema é representado em UML por 5 (cinco) diferentes visões, cada uma definida por um conjunto de diagramas. A seguir serão apresentadas 2 (duas) das 5 (cinco) visões com as quais a análise se preocupa.

visão do modelo do usuário

visão do modelo estrutural

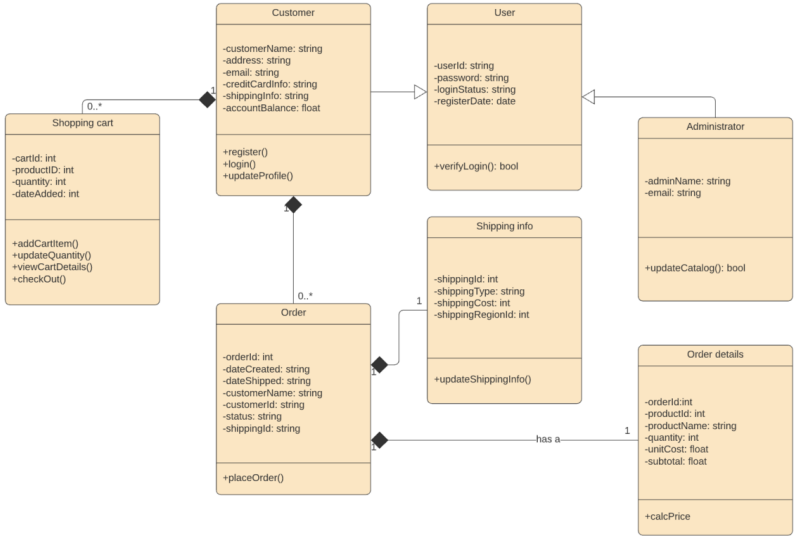
UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

Visão do modelo do usuário



Na **visão do modelo do usuário** o sistema é representado sob o ponto de vista do usuário, denominado ator, e utiliza a abordagem de casos de uso, que descrevem os cenários de uso do sistema a partir da perspectiva do usuário. Em outras palavras, os casos de uso descrevem a interação dos atores com o sistema e como este é usado.

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO



Visão do modelo Estrutural

A **visão do modelo estrutural** se preocupa com a modelagem da estrutura estática do sistema (classes, objetos e relacionamentos) visando os dados e funcionalidades do sistema.

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

Analise orientada a objeto

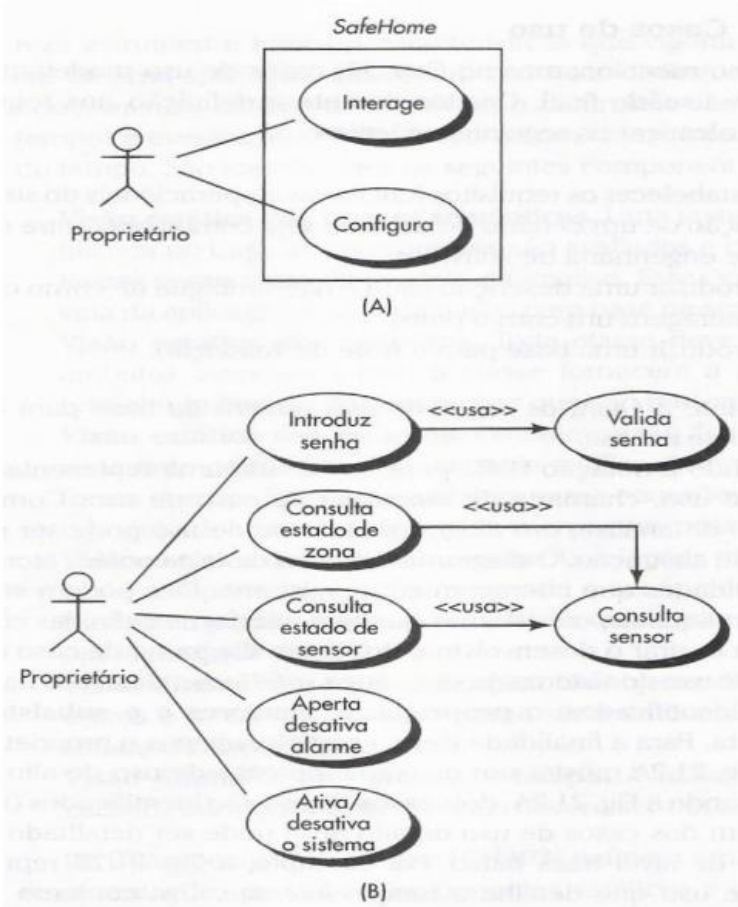
- Como dito, o processo de análise orientada a objetos se preocupa em compreender como o sistema será usado. Uma vez que o cenário de uso tenha sido definido, inicia-se a modelagem do software.
- Os casos de uso modelam o sistema do ponto de vista do usuário e servem como base para o elemento inicial do modelo de análise. Com o auxílio da UML uma representação gráfica do caso de uso é criada. Conhecida como Diagrama de Caso de Uso, ele apresenta atores e casos de uso.

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

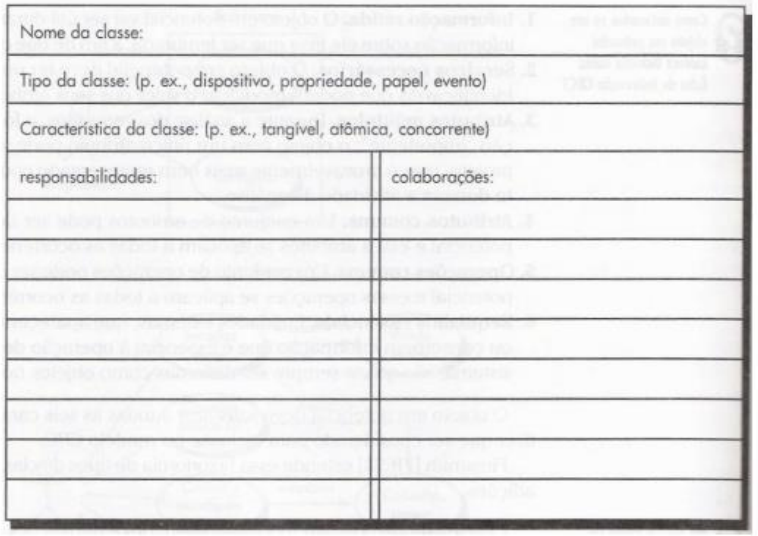
Dois casos de Uso

Como exemplo, tem-se um sistema hipotético de segurança domiciliar. Entre outros, identificou-se o ator proprietário e 2 (dois) casos de uso, representados pela figura oval, apresentados na Figura A. Cada caso de uso pode ser detalhado a níveis mais baixos, semelhante ao DFD, visto anteriormente.

A Figura B mostra o caso de uso para a função interage, apresentada na Figura A.



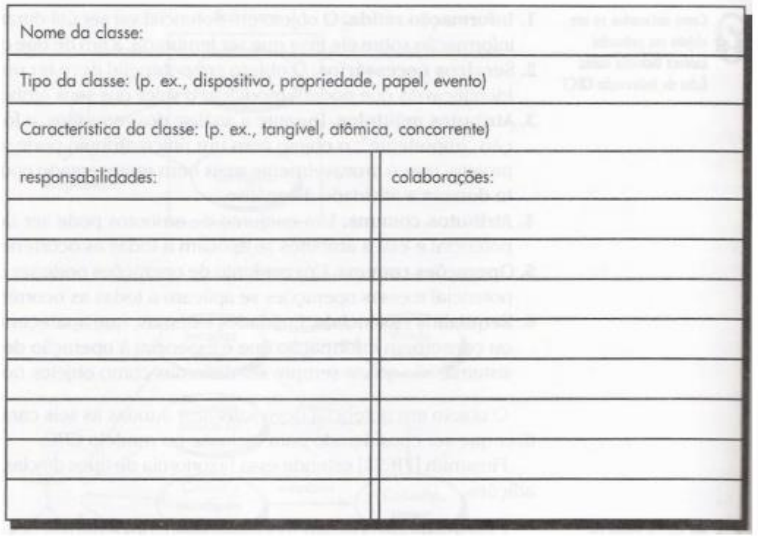
UniEVANGÉLICA
CENTRO UNIVERSITÁRIO



Cenário de uso

Uma vez levantados os cenários de uso, passa-se a identificar as possíveis classes e quais são suas responsabilidade e colaborações. Este processo recebe o nome de modelagem classe-responsabilidade-colaboração (*class-responsbility-collaborator* – CRC) e pode ser realizado com fichas de indexação reais, com exemplo apresentado na Figura, ou virtuais. O objetivo da modelagem CRC é criar uma representação organizada de classes, que são levantadas a partir de uma análise gramatical na descrição do sistema. Todos os substantivos são classes em potencial.

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

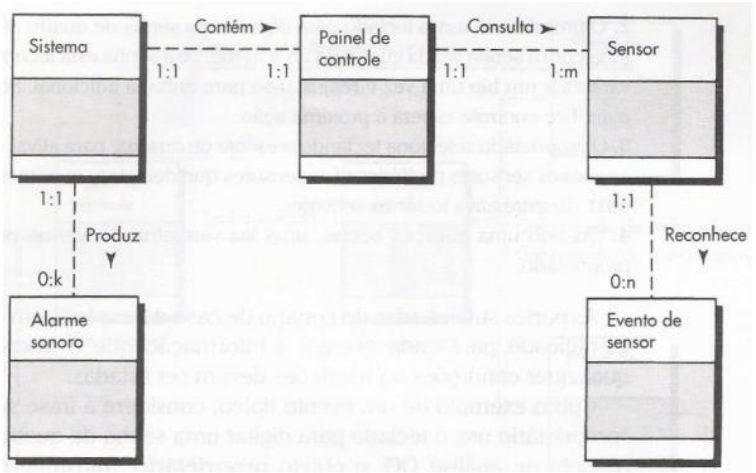


Cenário de uso

Os atributos e operações identificam as responsabilidades, que é qualquer coisa que a classe sabe ou faz. A classe se vale das colaborações com outras classes, que fornecem a ela informações necessárias para finalizar uma responsabilidade.

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

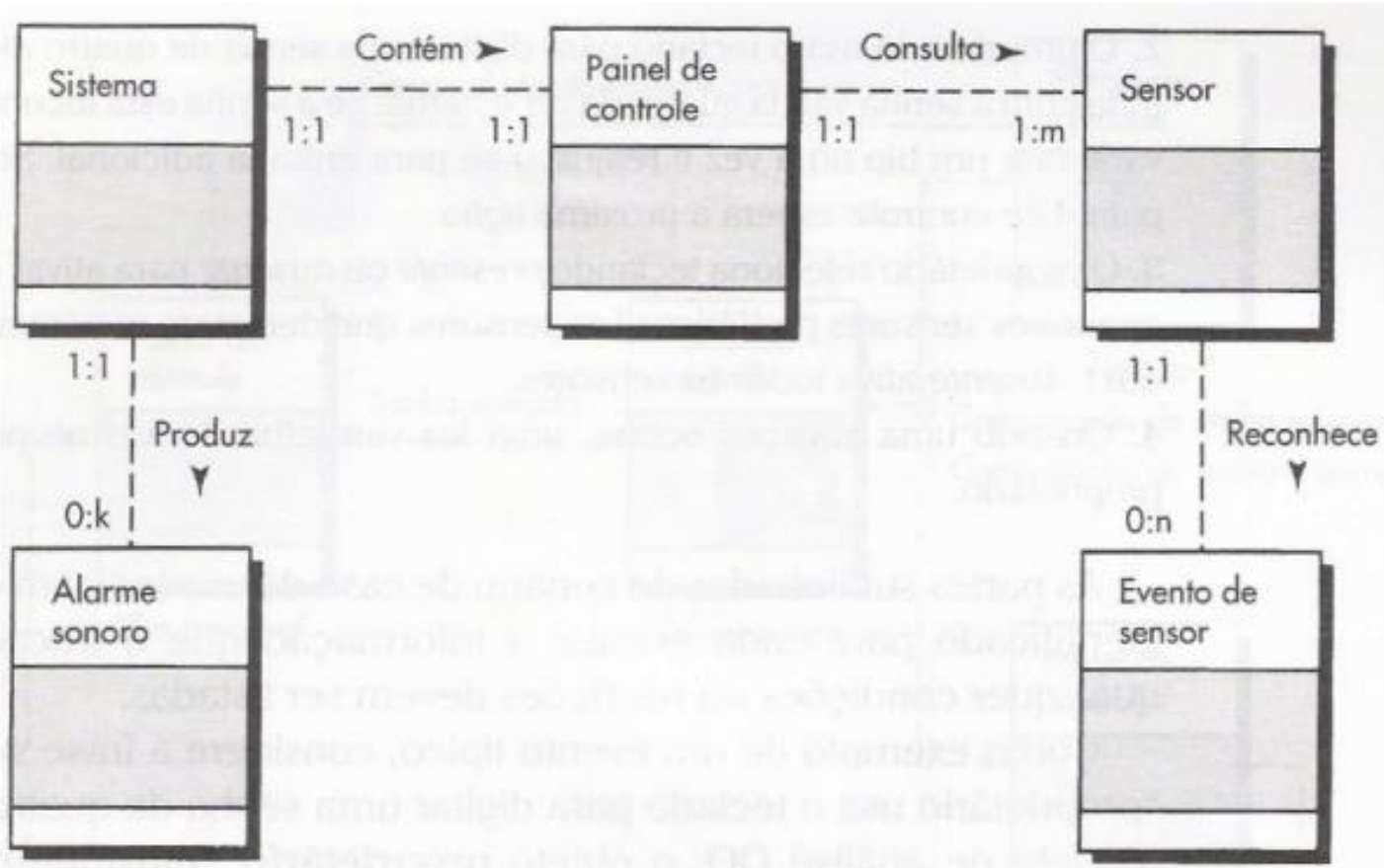
Objeto-Relacionamento



Tendo sido estabelecidas as responsabilidades, a tarefa de modelagem objeto-relacionamento define as classes que colaboram entre si para satisfazer cada responsabilidade. Isto estabelece a associação (ou conexão) entre classes. Pressman (2006) sugere examinar os verbos na declaração dos casos de uso do sistema para que se possam encontrar os possíveis relacionamentos. A UML vale-se da notação para modelagem objeto-relacionamento a partir de uma simbologia adaptada da modelagem entidade-relacionamento. As associações entre objetos são realizada usando-se relacionamentos rotulados juntamente com a cardinalidade da associação.

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

Relacionamento entre Objetos

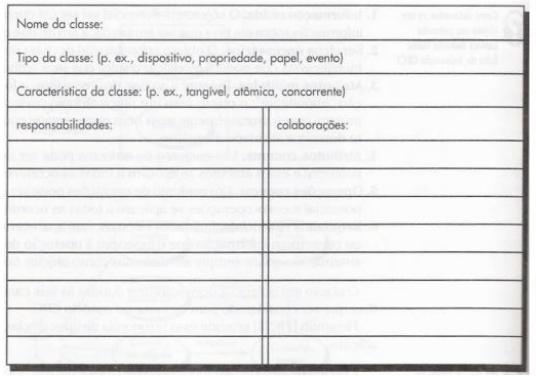


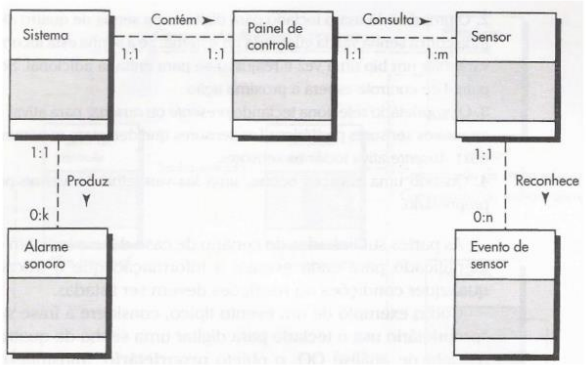
UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

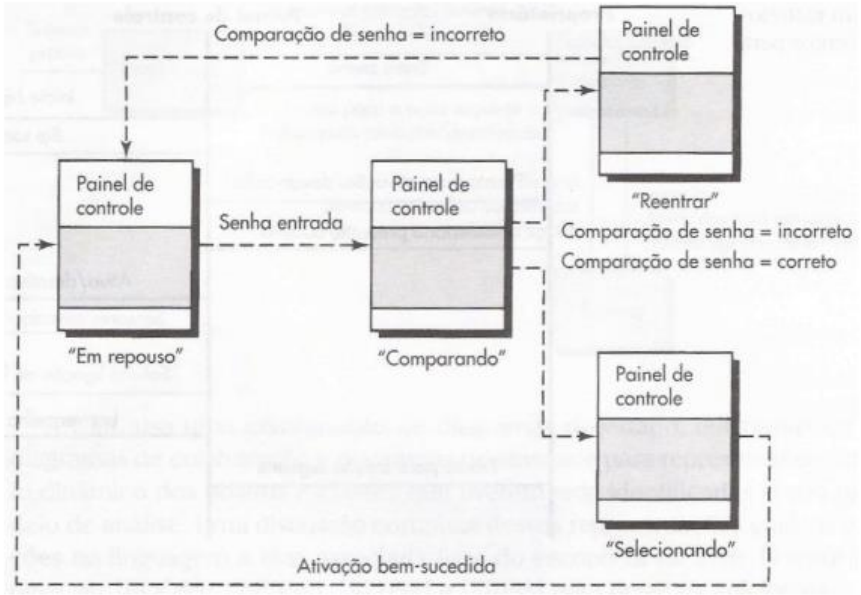
Modelo CRC / Objeto-relacionamento

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO

Tanto a modelagem CRC quanto a modelagem objeto-relacionamento correspondem a elementos estáticos no modelo de análise orientada a objetos. Para demonstrar o comportamento dinâmico do sistema é necessário representar o comportamento do sistema como uma função de eventos e tempo, ambos específicos.







Objeto-Comportamento

Neste sentido, a modelagem objeto-comportamento descreve como o sistema orientado a objetos vai responder a eventos e estí- mulos externos e é criado de acordo com os seguintes passos:

1. A interação interna do sistema é avaliada utilizando-se todos os casos de uso;
2. Identificar e entender os eventos, como eles direcionam as interações e como se relacionam a objetos específicos;
3. Para cada caso de uso deve-se criar uma marcação de eventos;
4. Construir um Diagrama de Transição de Estados para todos os casos de uso;
5. Verificar a precisão e consistência do modelo objeto-comportamento.

UniEVANGÉLICA
CENTRO UNIVERSITÁRIO