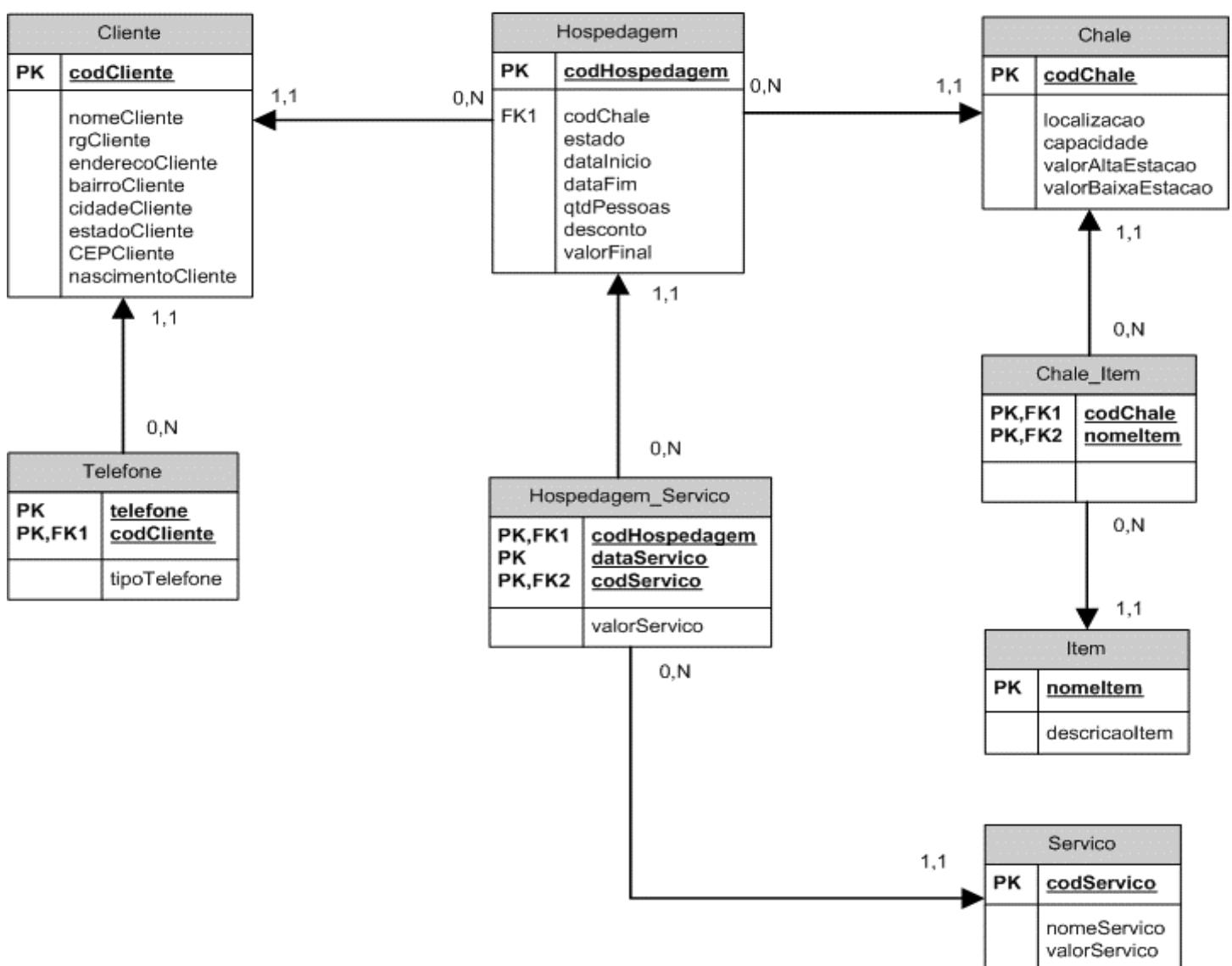


1. DIAGRAMA DE ENTIDADE RELACIONAMENTO (DER)

Um diagrama entidade relacionamento (ER) é um tipo de fluxograma que ilustra como “entidades”, p. ex., pessoas, objetos ou conceitos, se relacionam entre si dentro de um sistema. Diagramas ER são mais utilizados para projetar ou depurar bancos de dados relacionais nas áreas de engenharia de software, sistemas de informações empresariais, educação e pesquisa. Também conhecidos como DERs, ou modelos ER, usam um conjunto definido de símbolos, tais como retângulos, diamantes, ovais e linhas de conexão para representar a interconectividade de entidades, relacionamentos e seus atributos. Eles espelham estruturas gramaticais, onde entidades são substantivos e relacionamentos são verbos.

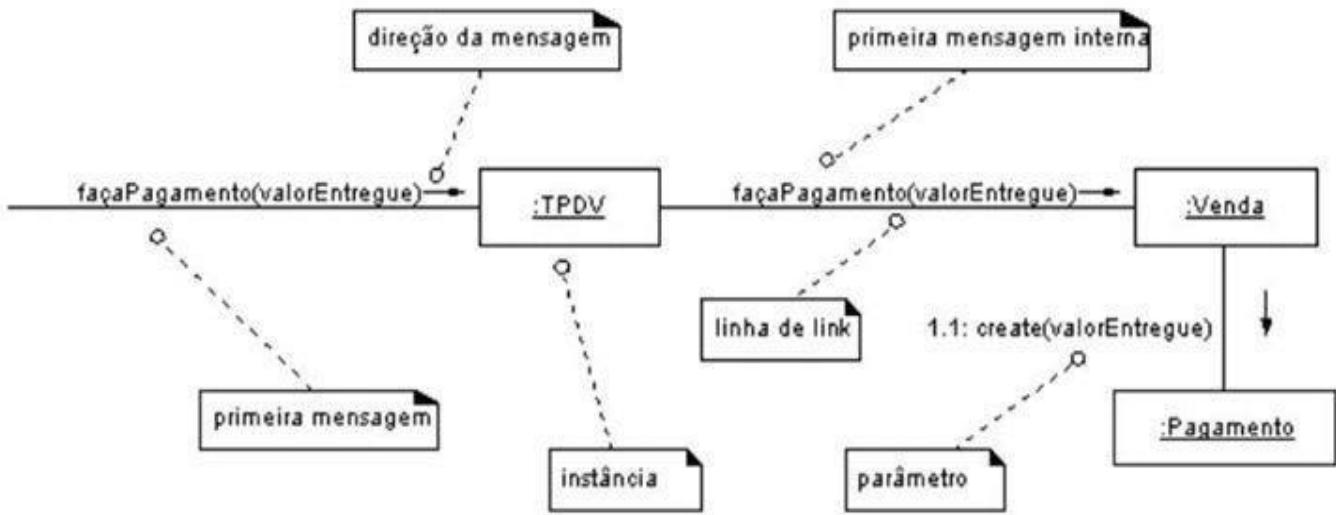
Exemplo de DER:



2. DIAGRAMAS DE COLABORAÇÃO

O objetivo é de mostrar como as pós-condições dos contratos serão realizadas. O diagrama de colaboração é mais adequado quando se deseja expressar mais detalhes da colaboração entre objetos.

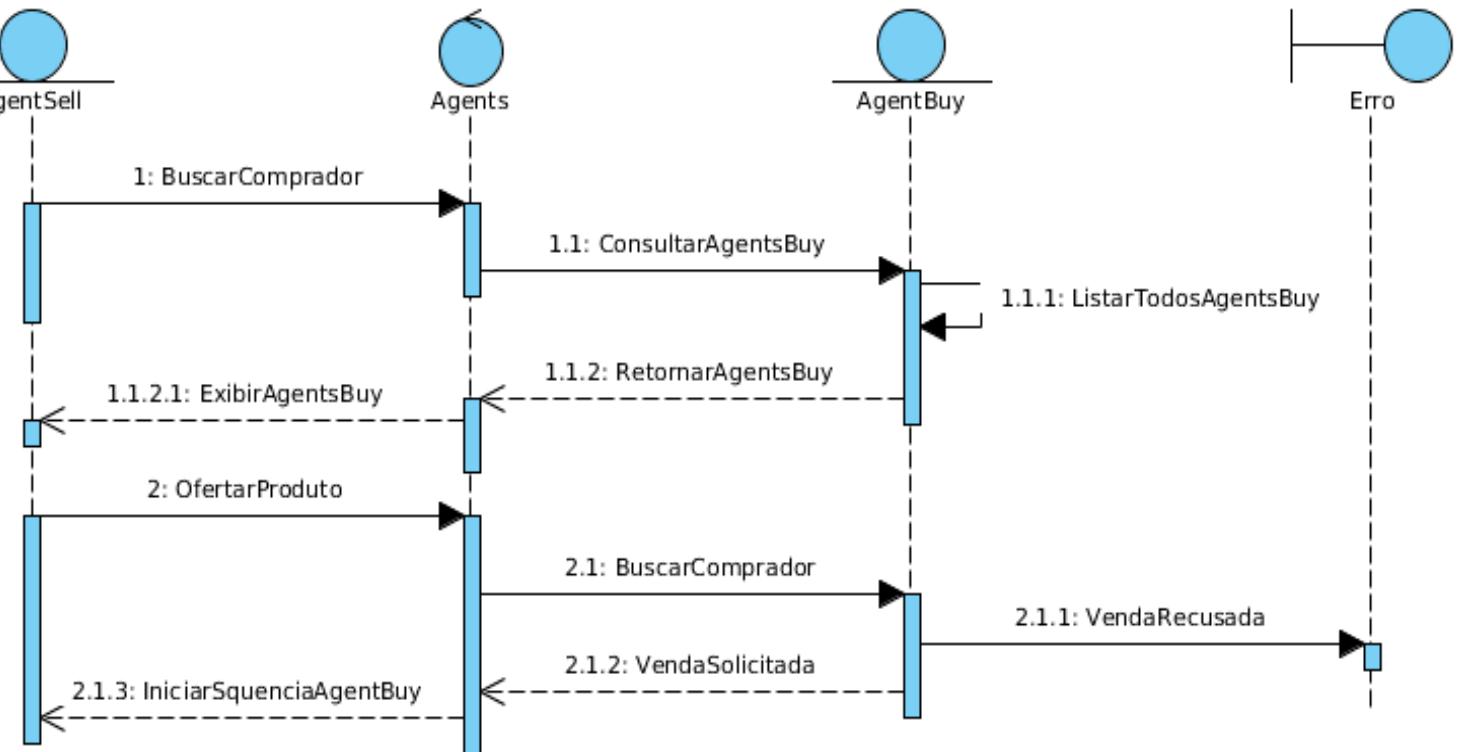
Exemplo de um diagrama de colaboração:



3. DIAGRAMA DE SEQUÊNCIA

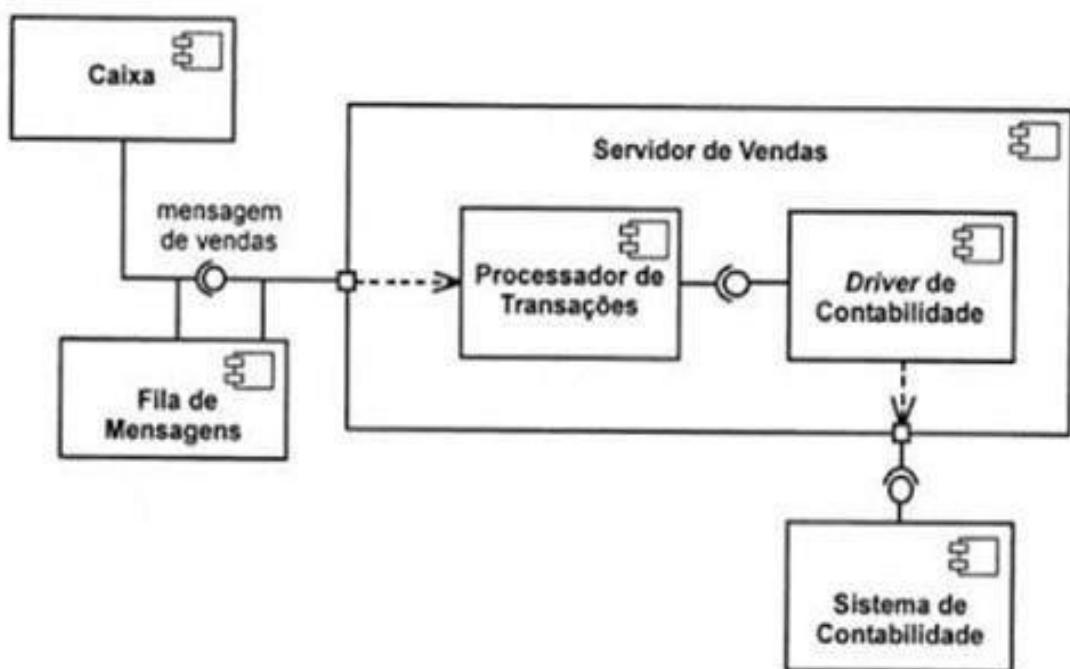
O Diagrama de Sequência da UML (Unified Modeling Language) é a representação da sequências de processos num programa. Ou seja: um diagrama de sequência descreve o modo como os grupos de objectos colaboram em algum comportamento ao longo do tempo.

Exemplo:



4. DIAGRAMA DE COMPONENTES

O Diagrama de Componentes ilustra a distribuição de classes para cada componente do projeto. Cada componente pode entender qual das classes ele representa.



5. DIAGRAMA DE CASO DE USO

Esse diagrama documenta o que o sistema faz do ponto de vista do usuário. Em outras palavras, ele descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema. Nesse diagrama não nos aprofundamos em detalhes técnicos que dizem como o sistema faz.

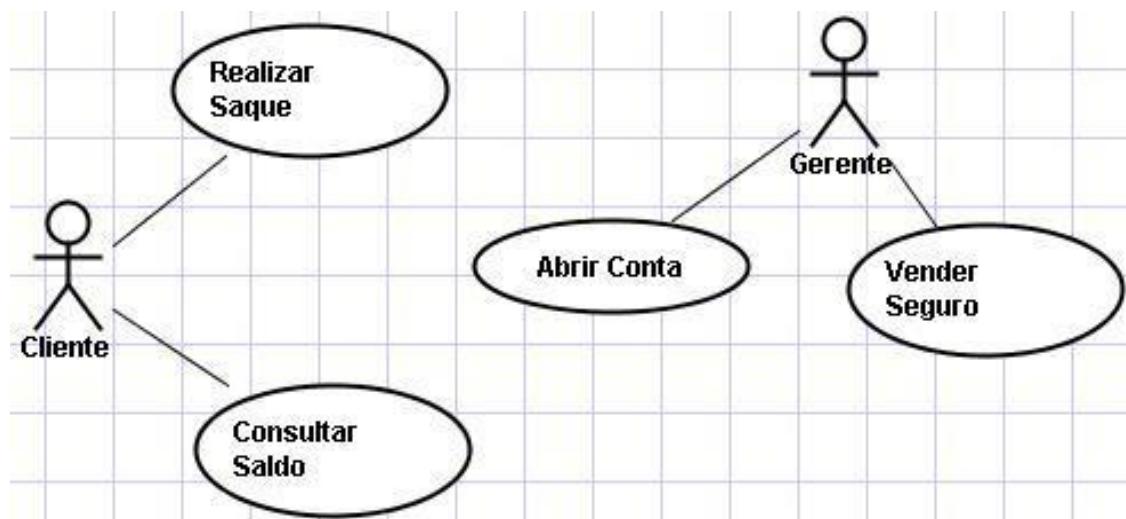
Este artefato é comumente derivado da especificação de requisitos, que por sua vez não faz parte da UML. Pode ser utilizado também para criar o documento de requisitos. Diagramas de Casos de Uso são compostos basicamente por quatro partes:

Cenário: Sequência de eventos que acontecem quando um usuário interage com o sistema.

Autor: Usuário do sistema, ou melhor, um tipo de usuário.

Use Case: É uma tarefa ou uma funcionalidade realizada pelo autor (usuário).

Comunicação: é o que liga um autor com um caso de uso.

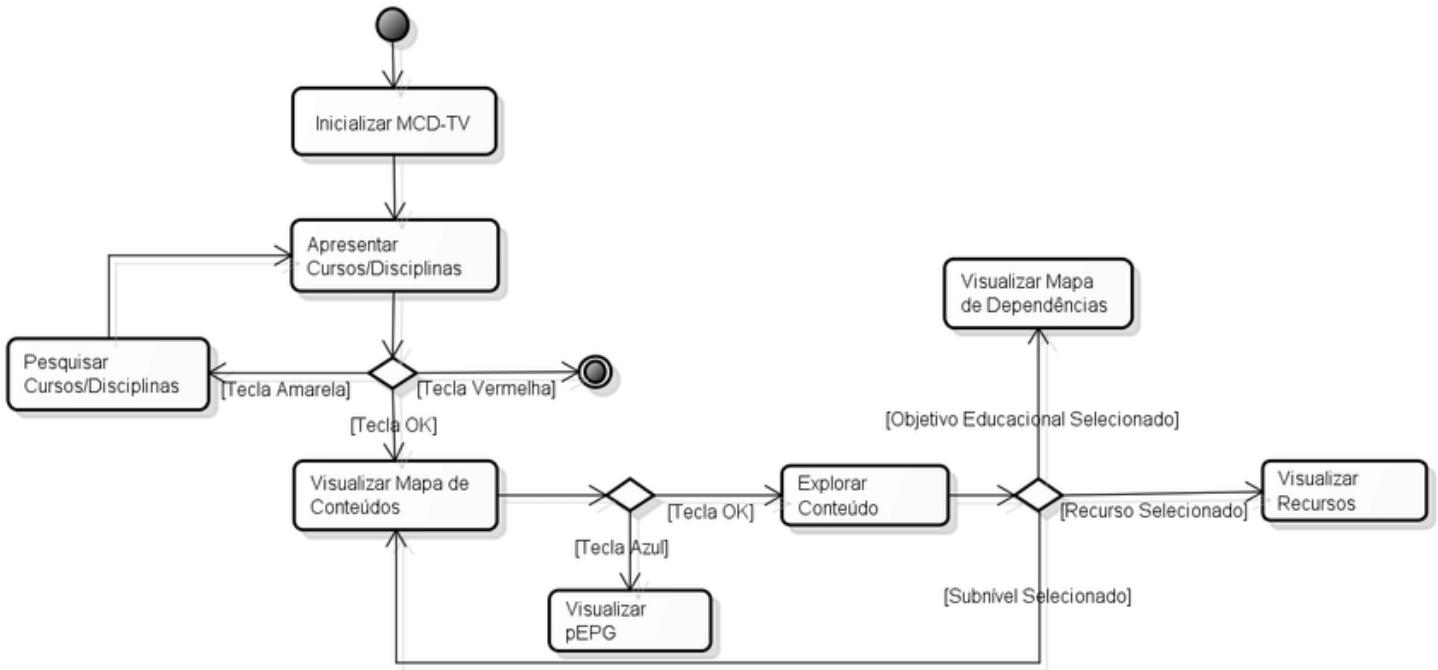


6. DIAGRAMA DE ATIVIDADES

O diagrama de atividades tem como objetivo principal a especificação do comportamento do software, do ponto de vista funcional, ou seja, das suas funcionalidades.

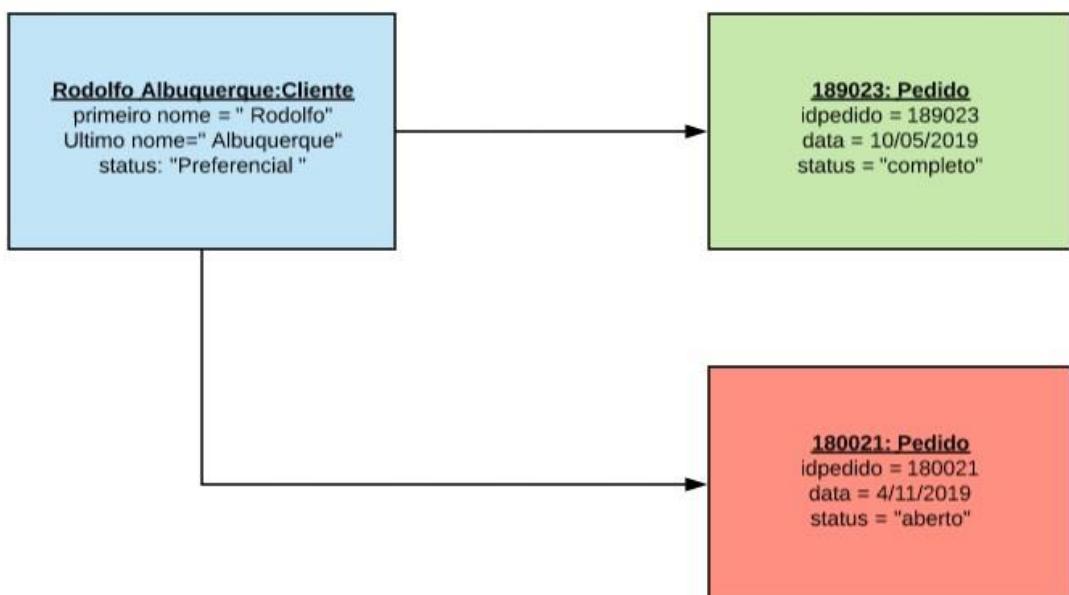
É pertinente utilizá-lo quando o propósito é:

- Documentar o aspecto funcional (não estrutural) do software, quando é necessário representar o fluxo da informação que o software trabalhará, e quando existem condições/decisões que precisam detalhadas/descritas.
- Mostrar aspectos específicos de alguma rotina do negócio que será automatizada pelo software, como um “zoom” em parte de alguma funcionalidade, por exemplo. Obs.: muito cuidado ao especificar toda uma funcionalidade (uma tela ou rotina batch por exemplo) num diagrama de atividades. Geralmente isso gera diagramas de atividades complexos, o que pode gerar efeito inverso (ou seja, “subtrair mais que somar”).
- Mostrar como Funcionalidades vão realizar Requisitos Funcionais (funções executadas pelas funcionalidades), e a relação dos Requisitos Funcionais com as Regras de Negócio.
- Documentar de forma macro como o sistema irá funcionar, mas orientado ao software, não ao processo de negócio.
- Mostrar como os módulos do sistema interagem entre si, as principais informações trafegadas durante a execução do software, entradas e saídas principalmente.



7. DIAGRAMA DE OBJETOS

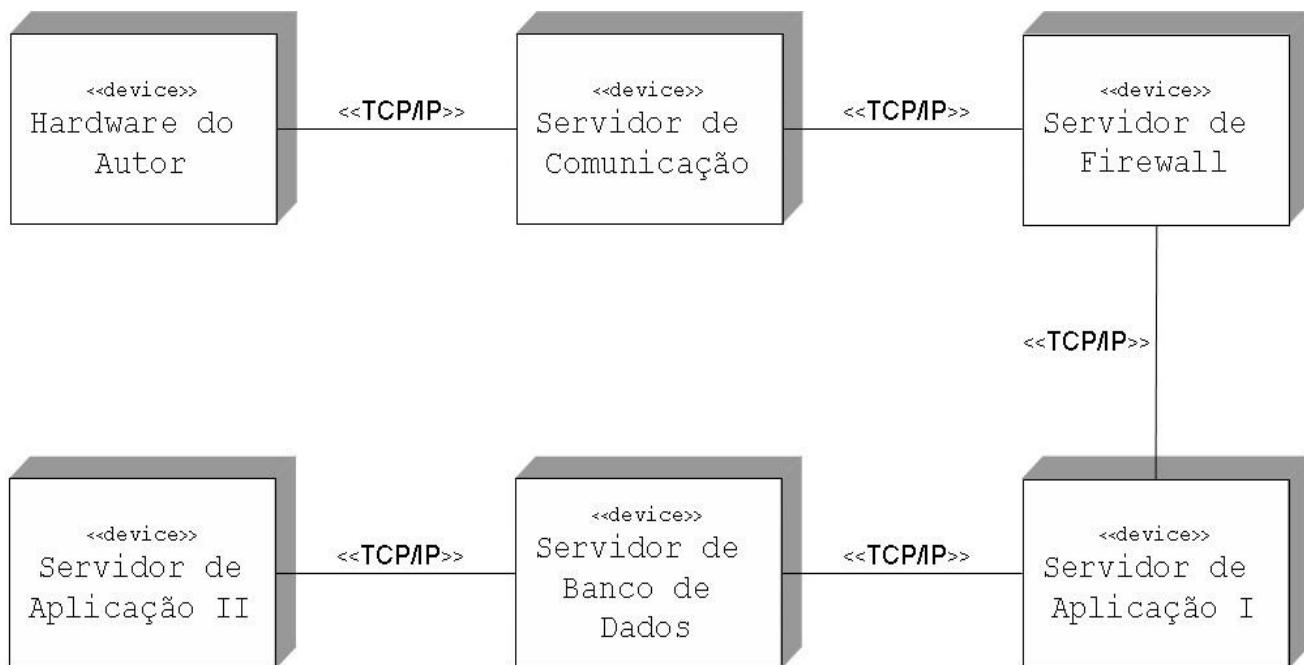
O diagrama de objetos modela as instâncias das classes contidas no diagrama de classes, isto é, o diagrama de objetos mostra um conjunto de objetos e seus relacionamentos no tempo. Estes diagramas são importantes para construir os aspectos estáticos do sistema. Normalmente, são compostos por: objetos e vínculos.



O exemplo acima mostra um diagrama de objetos para o cliente Rodolfo Albuquerque e seus dois pedidos. O diagrama pode ser lido da seguinte maneira: O objeto Rodolfo Albuquerque da classe Cliente está associado a ambos os objetos 189023 e 180021 da classe Pedido.

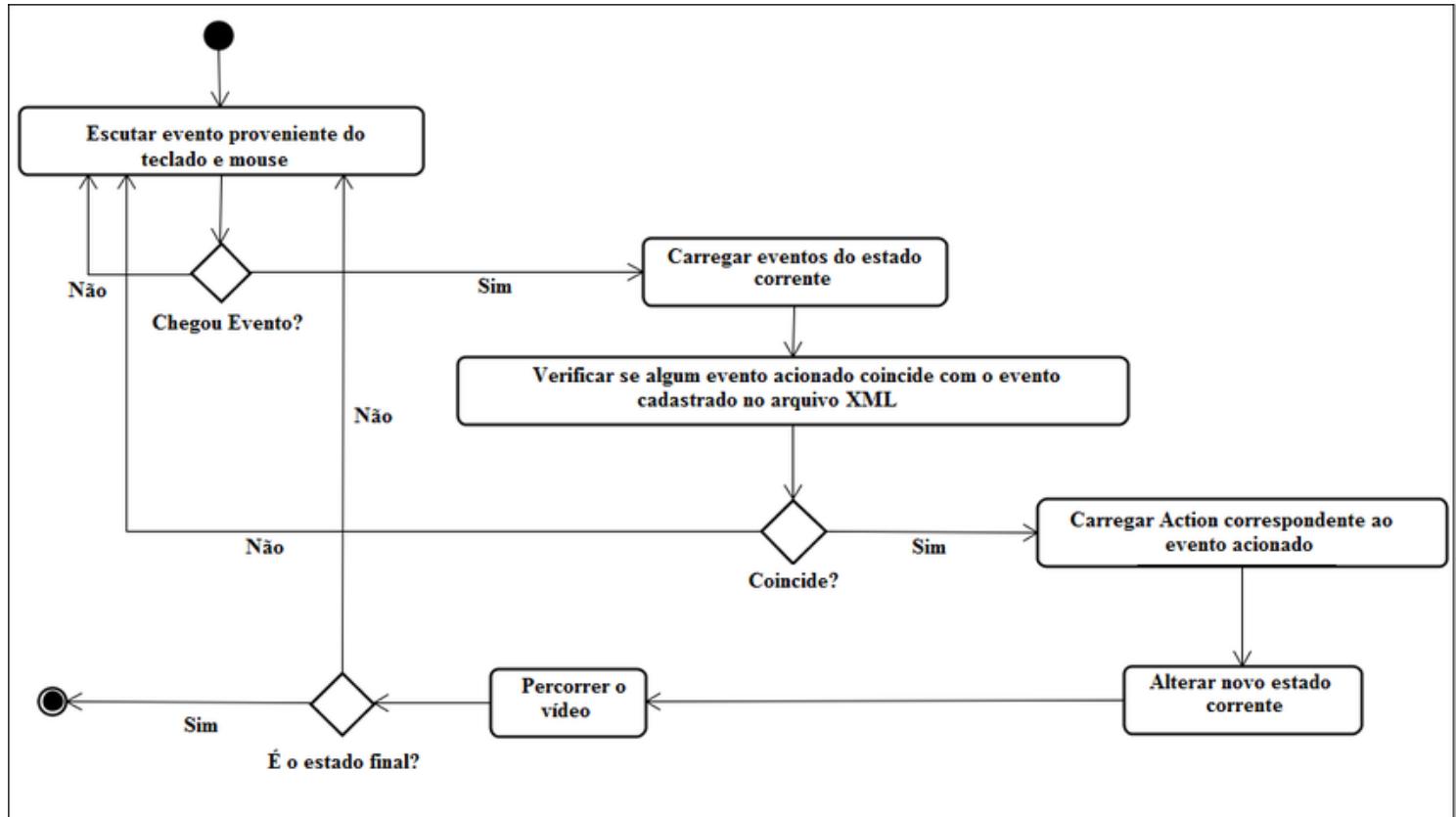
8. DIAGRAMA DE DEPURAÇÃO

Os diagramas de implementação, que normalmente são preparados durante a fase de desenvolvimento da implementação, mostram a organização física dos nós em um sistema distribuído, os artefatos que estão armazenados em cada nó e os componentes e outros elementos que os artefatos implementam. Os nós representam dispositivos de hardware como por exemplo servidores, sensores e impressoras, bem como outros dispositivos que suportam o ambiente de tempo de execução de um sistema. Caminhos de comunicação e relacionamentos de implementação modelam as conexões do sistema. Ou seja representa uma coleção de componentes e mostra como esses são distribuídos em um ou vários nós de hardware.



9. DIAGRAMA DE ESTADOS

Esse diagrama representa um conjunto de estados que um objeto pode estar e as mudanças sofridas por esse objeto dentro de um determinado processo.



Descrição do exemplo: Modelagem do sistema de login. Para que o usuário seja autenticado, ele deve fornecer dois valores: SSN (Social Security Number) e o PIN (Personal ID Number). Após a submissão é feita uma validação.

Diagrama de estado para o objeto *Login*.

10. DIAGRAMA MER

Modelo de Entidade Relacionamento é utilizado para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).

Entidades

Entidade é um objeto que existe no mundo real com uma identificação e significado próprio. As entidades são classificadas a partir do motivo de sua existência.

- **Entidades fortes:** a existência independe outras entidades para existir, por si só elas já possuem total sentido de existir.
- **Entidades fracas:** precisam de outra entidade para garantir a sua existência, são aquelas cuja existência independe de outras entidades, ou seja, por si só elas já possuem total sentido de existir.
- **Entidades associativas:** As entidades associativas existem a partir da necessidade de associar uma entidade a um relacionamento existente.

Atributos

Atributos são as características que descrevem cada entidade.

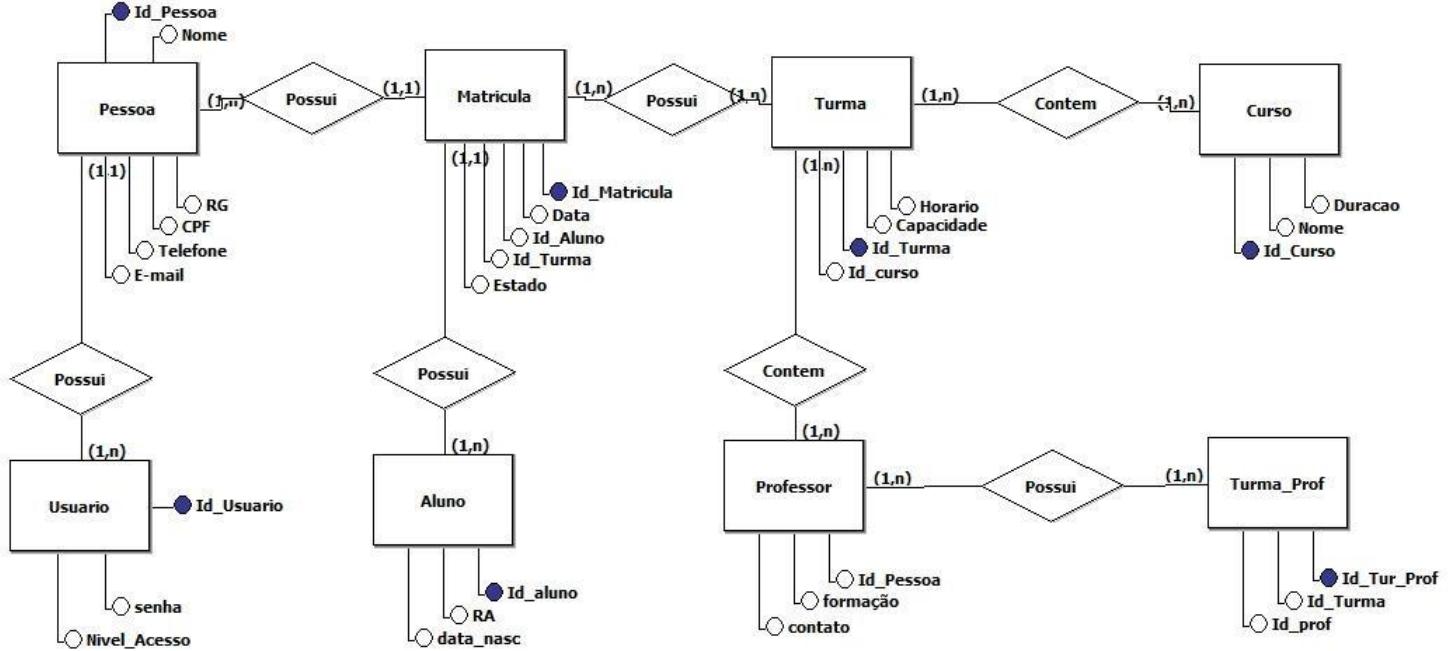
- **Atributo Monovalorado:** assume um único valor para cada elemento do conjunto x entidade
- **Atributo Composto:** formado por um ou mais sub-atributos
- **Atributo Multivalorado:** uma única entidade tem diversos valores para este atributo (seu nome é representado no plural).
- **Atributo Determinante:** identifica cada entidade de um conjuntoentidade (também conhecido como atributo chave).

Relacionamento

Relacionamento é o tipo de associação existente entre entidades

- **Relacionamentos um para muito (1:n):** uma das entidades envolvidas pode se relacionar com várias unidades da outra, porém, do outro lado cada uma das várias unidades só pode estar ligadas a uma unidade da outra entidade
- **Relacionamentos muitos para muitos (n:n):** cada entidade, de ambos os lados, podem referenciar múltiplas unidades da outra.

- **Relacionamento um para um (1:1):** cada uma das duas entidades envolvidas relacionam-se obrigatoriamente apenas com uma unidade da outra.



11. DIAGRAMA DE CLASSE

Diagrama de classes é uma representação estática no qual descreve o que deve estar presente no sistema a ser modelado, apresentando suas classes, atributos, operações e as relações entre os objetos, ou seja, como o sistema deve ser dividido em termos de classes. Na maioria desses modelos incluem-se em seu diagrama:

- uma classe
- uma interface
- um tipo de dado
- um componente.

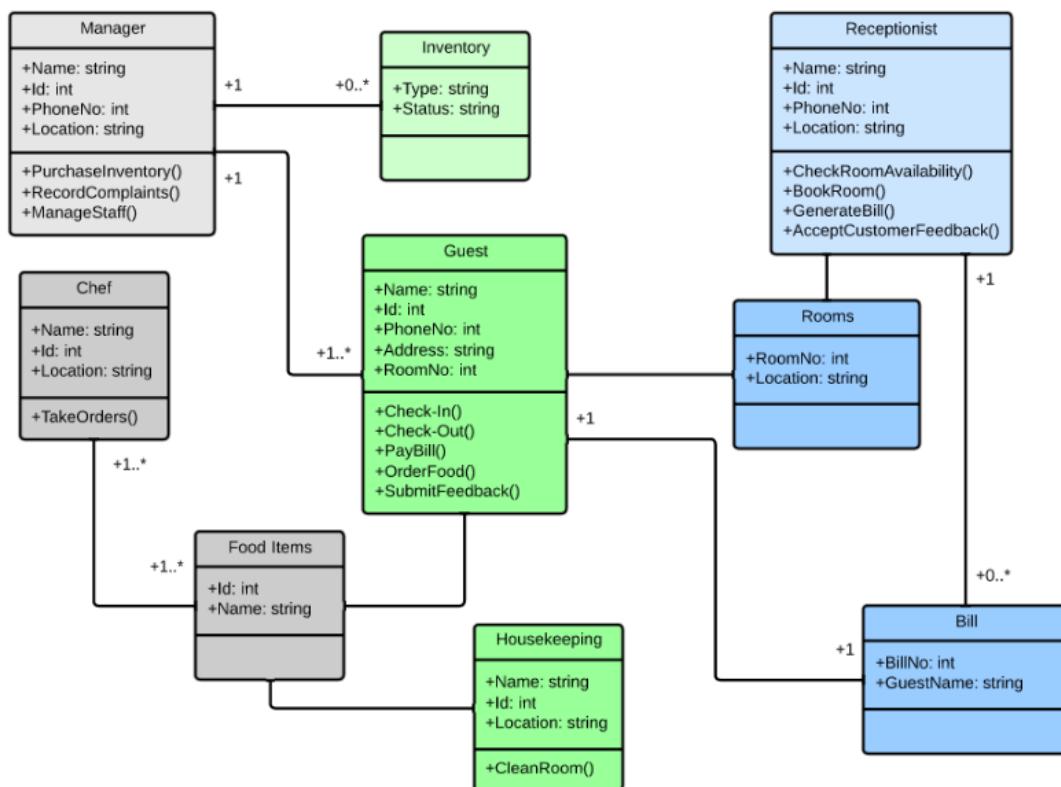
Os Diagramas de classes oferecem diversas vantagens para qualquer organização como, uma melhor visão dos esquemas de uma aplicação, um nível

mais favorável para a visualização das necessidades específicas de um sistema e divulgação dessas informações.

O diagrama de classes apresenta diversos tipos de interações que se referem as diversas relações e ligações que podem existir em diagramas de classes e objetos. Algumas das interações mais comuns incluem:

- **Hereditariedade:** também conhecida como generalização, é o processo de um secundário, ou subclasse, assumindo a funcionalidade de um primário, ou superclasse. É representada por uma linha conectada reta com uma ponta de seta fechada apontando para a superclasse.
- **Associação bidirecional:** a relação padrão entre duas classes. Ambas as classes estão cientes uma da outra e de sua relação entre si. Essa associação é representada por uma linha reta entre duas classes.

O exemplo abaixo oferece uma visão geral útil do sistema de gestão hoteleira.



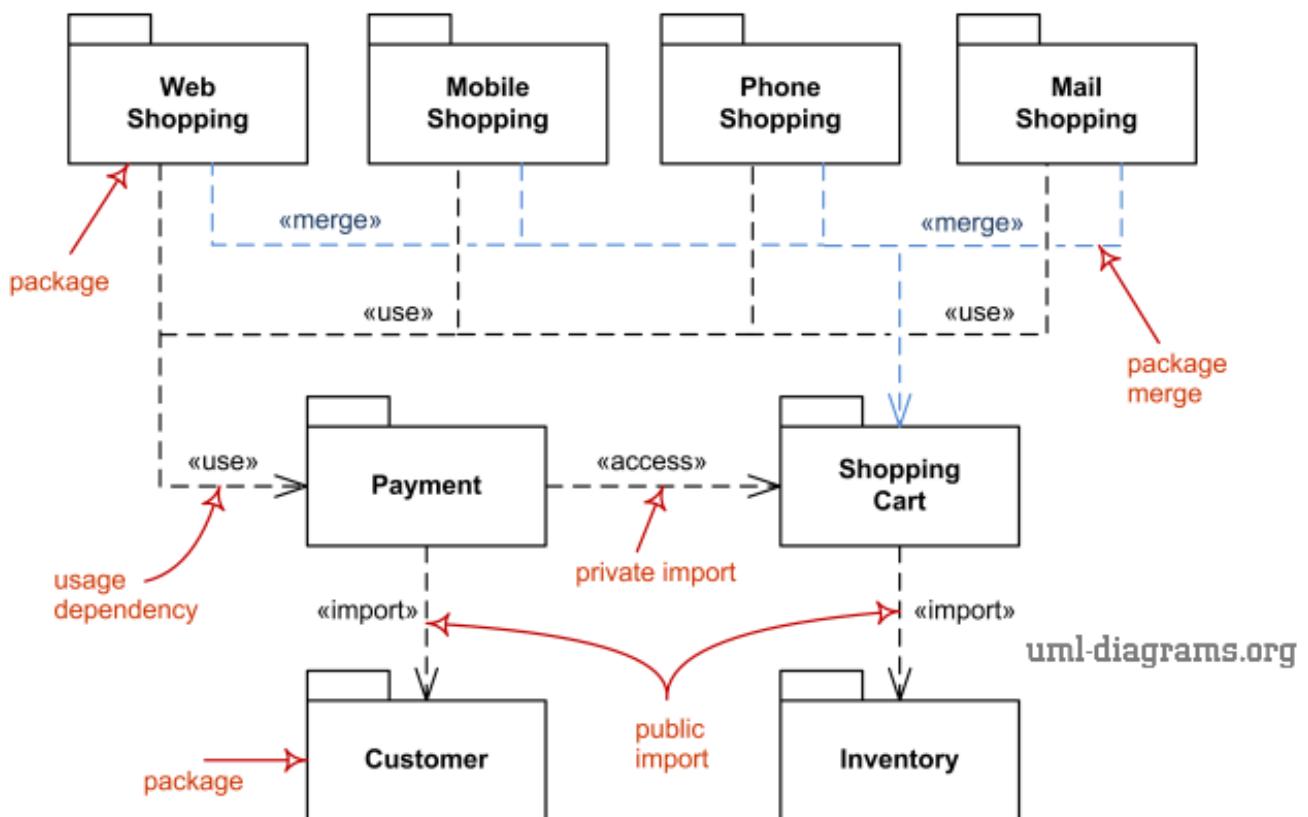
12. DIAGRAMA DE PACOTES

O diagrama de pacotes, um tipo de diagrama estrutural, mostra o arranjo e a organização dos elementos do modelo em projetos de média e grande escala. O diagrama de pacotes pode mostrar a estrutura e as dependências entre subsistemas ou módulos, mostrando diferentes visões de um sistema, por exemplo, como um aplicativo de várias camadas.

Os diagramas de pacote são usados para estruturar elementos de sistema de alto nível. Os pacotes são usados para organizar grandes sistemas que contêm diagramas, documentos e outras entregas importantes.

O Diagrama de pacotes pode ser usado para simplificar diagramas de classes complexos, podendo agrupar classes em pacotes sendo que um pacote é uma coleção de elementos UML logicamente relacionados. Os pacotes são representados como pastas de arquivos e podem ser usados em qualquer um dos diagramas UML.

Temos como exemplo abaixo um sistema de comprar no shopping. Alguns elementos principais do diagrama de pacotes são mostrados no desenho abaixo. Os pacotes Web Shopping, Mobile Shopping, Phone Shopping e Mail Shopping mesclam o pacote do carrinho de compras. Os mesmos 4 pacotes usam o pacote de pagamento. Os pacotes Payment e Shopping Cart importam outros pacotes.



REFERÊNCIAS

- <http://www.dsc.ufcg.edu.br/~jacques/cursos/apoo/html/proj1/proj4.htm>
- <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-entidaderelacionamento>
- <https://www.ateomomento.com.br/diagrama-de-sequencia-uml/>
- <https://www.ateomomento.com.br/uml-diagrama-de-atividades/>
- <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-deuso-introducao-pratica-a-uml/23408>
- https://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/cont_eudo/objetos/diagrama_de_objetos.htm
- https://www.ibm.com/support/knowledgecenter/ptbr/SS5JSH_9.5.0/com.ibm.xtools.modeler.doc/topics/cobjdiags.html
- <https://www.devmedia.com.br/artigo-sql-magazine-68-utilizando-umldiagramas-de-implantacao-comunicacao-e-tempoartigo-sql-magazine68-utilizando-uml-diagramas-de-implantacao-comunicacao-etempo/16353>
- https://www.ibm.com/support/knowledgecenter/ptbr/SS4JE2_7.5.5/com.ibm.xtools.modeler.doc/topics/cdepd.html
- <http://docente.ifrn.edu.br/diegooliveira/disciplinas/pds/aula-05-analise>
- https://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/umldiagramas_v01-1.pdf
- http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/estado/diag_estados.htm
- <https://www.profissionaisti.com.br/2011/07/os-principais-diagramas-dauml-resumo-rapido/>
- <https://www.inf.ufsc.br/~r.fileto/Disciplinas/INE5423-2010-1/Aulas/02MER.pdf>
- <https://www.devmedia.com.br/modelo-entidade-relacionamento-merediagrama-entidade-relacionamento-der/14332>
- <https://sites.google.com/site/unilibancodedados1/aulas/aula-7---tiposde-relacionamento>