

Ana Cristina Melo

# Exercitando modelagem em UML



## 51 exercícios resolvidos

- inclui questões de concursos públicos
- apresenta comentários
- modelos compatíveis com as versões 1.4 e 2.0 da UML



## **Apresentação**

Minha experiência lecionando disciplinas de modelagem de desenvolvimento de sistemas, com foco em UML, aliada à necessidade de ter um material de apoio técnico mais completo, me levaram a escrever *Desenvolvendo Aplicações com UML*<sup>1</sup> – do conceitual à implementação.

Entretanto, passei a perceber outras necessidades em sala de aula, nessa e em outras disciplinas: um material de apoio prático, que pudesse ser usado para fixação da matéria.

Notei que essa não era uma necessidade só da docência, mas também do aluno que entende a teoria, mas não consegue sair do ponto zero, não consegue começar. E por que isso? Porque sem prática não é possível aprender. Sem exercícios não há como termos certeza do aprendizado.

E olhando para minha experiência, há dezoito anos como Analista e Programadora, percebi que também temos a necessidade do exemplo. Principalmente quando aprendemos sozinhos. Nesse caso, aprendemos por meio de tentativas e erros, exemplos e exercícios.

Assim, espero que esse material atenda a todos: professores, alunos e profissionais da área.

O livro está dividido em capítulos, que tratam dos seguintes temas:

Capítulo 1 – Encontrando classes, atributos e relacionamentos.

Capítulo 2 – Desenhando diagrama de classes a partir de minicenários.

---

<sup>1</sup> e *Desenvolvendo Aplicações com UML 2.0* (2ª edição).

## **X • Exercitando Modelagem em UML**

**Capítulo 3 – Exercitando a identificação de casos de uso.**

**Capítulo 4 – Exercitando a identificação de classes a partir de casos de uso.**

**Capítulo 5 – Alguns exercícios sobre diagrama de máquina de estados.**

**Apêndice A – Questões de concursos públicos sobre orientação a objetos e UML.**

Alguns exercícios possuem complementos na forma de outros exercícios, oferecendo variações ou incrementos ao problema inicial.

Vocês vão reparar que a maioria dos cenários é descrita como sendo pessoas comuns que precisam de uma ferramenta para resolver seus problemas. Meu objetivo com esse tipo de abordagem foi o de aproximar o problema do conhecimento público do meu leitor. Assim, tenho certeza que vocês irão se familiarizar com os problemas de cada um de nossos personagens.

Quaisquer dúvidas, comentários, sugestões e/ou críticas podem ser enviadas para meu e-mail [livro@anacristinamelo.eti.br](mailto:livro@anacristinamelo.eti.br).

Obrigada e boa leitura!

*"A diferença não está no tamanho do seu sonho,  
mas no tamanho da sua vontade em querer realizá-lo!"*

*E sempre que você conseguir realizar um sonho, comece  
tudo de novo. Não podemos viver sem ter objetivos!"*

**Ana Cristina S. L. Melo**

# Sumário

<b>1. Encontrando Classes, Atributos e Relacionamentos .....</b>	<b>1</b>
1. Exercício: Conta de Luz .....	3
2. Exercício: Classe TextoSaida .....	4
3. Exercício: Classe BonecoEmMovimento .....	5
4. Exercício: Horário de Remédios .....	6
5. Exercício: Gastos Diários .....	8
6. Exercício: Comanda Eletrônica .....	9
7. Exercício: Lista de Compras .....	10
8. Exercício: Lista de Compras (Variação A) .....	12
9. Exercício: Lista de Compras (Variação B) .....	13
10. Exercício: Lista de Compras (Variação C) .....	15
11. Exercício: Lista de Compras (Variação D) .....	15
12. Exercício: Coleção de CD's .....	16
13. Exercício: Coleção de CD's (Variação A) .....	17
14. Exercício: Sala de Reunião .....	18
15. Exercício: Sala de Reunião (Variação A) .....	21
16. Exercício: Herança .....	22
<b>2. Desenhando Diagrama de Classes a partir de Minicenários .....</b>	<b>24</b>
17. Minicenário: Controle de Tarefas .....	25
18. Minicenário: Ligações telefônicas via modem .....	28
19. Minicenário: Testes de Fixação .....	30
20. Minicenário: Testes de Fixação (Variação A) .....	33
21. Minicenário: Festas Infantis .....	33
22. Minicenário: Festas Infantis (Variação A) .....	35
23. Minicenário: Cursos de Aperfeiçoamento .....	36
24. Minicenário: Cursos de Aperfeiçoamento (Variação A) .....	37
25. Minicenário: Classificados na Web .....	39
26. Minicenário: Jogo da Força .....	42

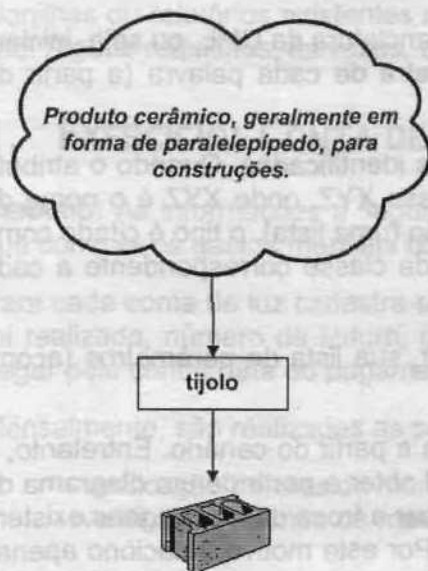


## **XII • Exercitando Modelagem em UML**

27. Minicenário: Ferramenta de UML.....	44
28. Minicenário: Clube da Leitura .....	46
29. Minicenário: Condomínio .....	48
30. Minicenário: Pediatra .....	51
31. Minicenário: Pediatra (Variação A) .....	54
32. Minicenário: Pediatra (Variação B) .....	56
33. Minicenário: Aposta Lotérica.....	58
<b>3. Exercitando a Identificação de Casos de Uso .....</b>	<b>60</b>
34. Cenário: Encomenda de placas.....	61
35. Cenário: Rádio Táxi Mar & Sol.....	67
36. Cenário: Treinador de vôlei.....	74
37. Cenário: Controle de Obra .....	79
38. Cenário: Controle de Obra (Variação A).....	86
39. Cenário: Senha de Atendimento .....	86
40. Cenário: Controle de Bolão.....	92
41. Cenário: Estacionamento.....	97
42. Cenário: Jogo da Velha.....	100
43. Cenário: Jogo da Memória.....	103
<b>4. Exercitando a Identificação de Classes a partir de Casos de Uso ..</b>	<b>107</b>
44. Cenário: Encomenda de placas.....	107
45. Cenário: Rádio Táxi Mar & Sol.....	108
46. Cenário: Treinador de Vôlei .....	110
47. Cenário: Controle de Bolão.....	111
48. Cenário: Jogo da Velha.....	113
<b>5. Alguns Exercícios sobre Diagrama de Máquina de Estados.....</b>	<b>114</b>
49. Cenário: Controle de Tarefas.....	114
50. Cenário: Encomenda de Placas.....	115
51. Cenário: Rádio Táxi Mar & Sol.....	115
<b>Apêndice. Questões de Concursos Públicos sobre Orientação a Objetos e UML.....</b>	<b>117</b>

## Encontrando Classes, Atributos e Relacionamentos

Um dos principais diagramas da UML é o Diagrama de Classes. Observando meus alunos, não vejo dificuldade no aprendizado do que é um relacionamento de associação, herança ou agregação. Vejo dificuldade em se abstrair o conceito desses relacionamentos do mundo real, ou melhor, de um problema real.



Provavelmente seja fácil entender que nome é atributo de pessoa, e que funcionário é uma herança de pessoa. Mas será que o nível de facilidade permanece o mesmo quando estamos diante de um problema real e este precisa ser abstraído para atributos, classes, relacionamentos e outros elementos? Posso apostar que não! E como é possível que essa abstração se torne uma tarefa natural? Respondendo: exercitando e/ou aprendendo por meio de análise de exercícios prontos.

## 2 • Exercitando Modelagem em UML

Vamos começar a exercitar essa abstração mais primária, que está diretamente ligada ao conceito de orientação a objetos, mas que sem ela não existe modelo em UML. Vamos perceber, na evolução dos exercícios, o quanto uma modelagem pode mudar, de acordo com o contexto em que estamos trabalhando. Nosso objetivo nesse capítulo é trabalhar a abstração, por isso só desenharemos a classe na notação da UML quando for necessário explicar os relacionamentos.

Os primeiros exercícios procuram extrair de um descritivo (que eu chamo de cenário) a classe existente, seus atributos e/ou métodos.

Os exercícios seguintes começam a explorar a existência de mais de uma classe. Nesses exercícios, teremos variações dos mesmos problemas, a fim de exercitar a evolução da modelagem.

Para entendimento geral desse capítulo, leia com atenção as **convenções** que adotei para resolução dos exercícios.

### CONVENÇÕES DO CAPÍTULO:

- Na maioria dos exercícios não são desenhadas classes no padrão da UML. As classes são apenas relacionadas numa tabela;
- Os atributos e métodos seguem a nomenclatura da UML, ou seja, iniciam com letra minúscula, tendo a primeira letra de cada palavra (a partir da segunda) em maiúscula;
- Todos os atributos possuem seus tipos identificados. Quando o atributo for um objeto, o tipo é citado como *Classe XYZ*, onde XYZ é o nome da classe. Quando o atributo for uma coleção (uma lista), o tipo é citado como *Coleção de XYZ*, onde XYZ é o nome da classe correspondente a cada item da lista;
- Todos os métodos possuem, se existir, sua lista de parâmetros (acompanhados do tipo) e o tipo de retorno;
- Os métodos descritos foram deduzidos a partir do cenário. Entretanto, a lista completa dos métodos só é possível obter a partir de um diagrama de seqüências, que tem por objetivo identificar a troca de mensagens existente entre objetos, em cada caso de uso. Por este motivo, relaciono apenas os métodos mais relevantes;

- Os relacionamentos são descritos na seção de “Comentários sobre a Resolução”, em vez de serem representados numa classe;
- Na fase de análise, cabe desenharmos uma classe representando seus relacionamentos. Na fase de projeto, cabe transformarmos esses relacionamentos em atributos, de forma a representar como eles serão implementados. Para melhor entendimento dos exercícios, especificamos esses atributos que correspondem aos relacionamentos implementados. Esses atributos são identificados como atributos derivados, pois seus valores não são informados e sim gerados a partir de um relacionamento. Exemplo: Um funcionário possui os atributos nome e cargo. Porém cargo é uma classe que possui os atributos descrição e salário. O atributo cargo dentro de funcionário é fruto do relacionamento das duas classes;
- Todo atributo cujo tipo seja uma classe *enumeration*, não é definido como um atributo derivado, visto a classe *enumeration* atuar como um tipo de dado e não como um relacionamento;
- Atributos e métodos de classe são representados sublinhados, conforme notação da UML;
- Em alguns exercícios, para elucidar o cenário, são exibidas cópias de planilhas ou relatórios existentes no Mundo Real de nossos clientes. Atenção! Alguns requisitos, às vezes, estão implícitos nesses exemplos.

## 1. EXERCÍCIO: CONTA DE LUZ

**CENÁRIO:** As informações a seguir se referem à planilha Excel de Gabriel, que controla os gastos mensais com sua conta de luz.

Para cada conta de luz cadastra-se: data em que a leitura do relógio de luz foi realizada, número da leitura, quantidade de Kw gasto no mês, valor a pagar pela conta, data do pagamento e média de consumo.

Mensalmente, são realizadas as seguintes pesquisas:

- verificação do mês de menor consumo;
- verificação do mês de maior consumo.



#### 4 • Exercitando Modelagem em UML

##### IMAGEM DA PLANILHA:

###### LISTA DE ACOMPANHAMENTO DE GASTO DE LUZ

data leitura	nº leitura	kw gasto	valor a pagar	data pagto	média consumo
04/07/2005	4166	460	206,43	15/07/2005	15,33
02/08/2005	4201	350	157,07	15/08/2005	12,06

Menor Consumo	350	ago/05
Maior Consumo	460	jul/05

##### EXERCÍCIO:

Identifique as classes, atributos e métodos desse cenário.

##### RESOLUÇÃO:

Classe	Atributos	Métodos
<b>ContaLuz</b>	dataLeitura : date numeroLeitura : integer qtdKwGasto : integer valorPagar : real dataPagamento : date mediaConsumo : real	cadastrarConta verificaMesMenorConsumo : string verificaMesMaiorConsumo : string

##### COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) O retorno do método é string pois traz o mês e o ano (ex: 082005).

## 2. EXERCÍCIO: CLASSE TEXTOSAIDA

**CENÁRIO:** Para fixação do conceito de classes em sala de aula, Profª Cristina criou com seus alunos a classe TextoSaida.

O objetivo do exercício é criar uma classe que permita configurar um texto por meio de atributos (tamanho da letra, cor da fonte e cor do fundo), escolhendo em que tipo de componente ele deve ser exibido (entre as opções: *label*, *edit* e *memo*).

Para não haver vínculo com linguagens de programação, essa classe não foi criada como herança de uma classe visual.

As cores podem ser escolhidas entre os tons: preto, branco, azul, amarelo ou cinza.

**EXERCÍCIO:**

Identifique as classes, atributos e métodos desse cenário.

**RESOLUÇÃO:**

Classe	Atributos	Métodos
TextoSaida	texto : string tipoComponente : EnumTipoComponente tamanhoLetra : integer corFonte: EnumCor corFundo : EnumCor	cadastrar exibirTexto
«enumeration» EnumTipoComponente	label edit memo	
«enumeration» EnumCor	preto branco azul amarelo cinza	

**COMENTÁRIOS SOBRE A RESOLUÇÃO:**

a) A classe *enumeration EnumTipoComponente* foi criada para suportar o domínio do atributo *tipoComponente*, que consiste de valores fixos, definidos como sendo os componentes onde o texto pode ser exibido. O mesmo aconteceu com a classe *EnumCor*.

**3. EXERCÍCIO: CLASSE BONECOEMMOVIMENTO**

**CENÁRIO:** Professora Cristina decidiu criar uma classe que permita mover um boneco na tela.

Esse boneco deve ter nome, posição da coordenada X, posição da coordenada Y e direção atual (cima, baixo, direita, esquerda).

**EXERCÍCIO:**

Identifique as classes, atributos e métodos desse cenário.

**RESOLUÇÃO:**

Classe	Atributos	Métodos
<b>Boneco</b>	nome : string coordenadaX : integer coordenadaY : integer direcaoAtual : EnumDirecao	cadastrar mover (novaCoordenadaX: integer, novaCoordenadaY: integer)
«enumeration» <b>EnumDirecao</b>	cima baixo direita esquerda	

**COMENTÁRIOS SOBRE A RESOLUÇÃO:**

a) Para que o boneco se mova, cabe ao ator informar a nova coordenada. Isso é feito por meio de parâmetros do método *Mover*. Após mover o boneco, os atributos *coordenadaX* e *coordenadaY* são atualizados com os valores dos parâmetros *novaCoordenadaX* e *novaCoordenadaY*, respectivamente.

Para alterar a direção, basta antes de chamar o método *mover*, mudar o valor do atributo *direcaoAtual*.

**4. EXERCÍCIO: HORÁRIO DE REMÉDIOS**

**CENÁRIO:** As informações a seguir se referem a uma aplicação de controle pessoal de horário de remédios, existente no Palm de Maurício.

Para cada remédio cadastra-se: o nome de quem vai tomar o remédio, a data de início, a quantidade de dias que foi prescrita pelo médico, a quantidade de vezes ao dia, a dosagem e o nome do remédio.

Ao cadastrar o remédio, a aplicação sugere todos os horários possíveis para tomar o remédio. O usuário escolhe o melhor horário e a aplicação avisa até quando o remédio deve ser tomado e prepara uma planilha de horários.

O usuário, no início do dia, seleciona a opção de planilha de horários de remédios do dia.

No caso de Maurício atrasar o horário de tomar o remédio num determinado dia, a planilha reorganiza os horários daquele dia.

**EXERCÍCIO:**

Identifique as classes, atributos e métodos desse cenário.  
Represente os relacionamentos como atributos derivados.

**RESOLUÇÃO:**

Classe	Atributos	Métodos
<b>PrescricaoRemedio</b>	usuario : string dataInicio : date qtdDias : integer qtdVezeDia : integer dosagem : real remédio : string /planilhaHorario : Coleção de HorarioRemedio	sugerirHorarios exibirPlanilhaHorarios (dataReferencia : date )
<b>HorarioRemedio</b>	data : date horário : time	cadastrarHorario

**COMENTÁRIOS SOBRE A RESOLUÇÃO:**

a) Numa primeira modelagem, os atributos *remédio* e *usuário* (do Remédio) foram definidos como string. Esses atributos são fortes candidatos a se tornarem classes de negócio. O que determina essa mudança é o contexto da aplicação. Nesse caso, estamos tratando de uma aplicação bem pequena, cujo objetivo principal é o controle de horários. Sendo assim, podemos nessa versão não transformar esses atributos em classes.

b) Toda prescrição tem associada a ela uma planilha de horários. Isso cria um relacionamento da classe *PrescricaoRemedio* com a classe *HorarioRemedio*, representada pelo atributo *planilhaHorario*, que corresponde a uma coleção de objetos da Classe *HorarioRemedio*.

c) O método *sugerirHorario* é responsável por montar um conjunto possível de horários, a partir do atributo *qtdVezeDia*. Por exemplo, se um remédio tiver que ser tomado de 8 em 8 horas, a aplicação pode sugerir os seguintes horários: [6h, 14h, 22h], [7h, 15h, 23h], [8h, 16h, 24h].

Uma vez escolhido o conjunto de horários, o método *HorarioRemedio.cadastrarHorario* é chamado quantas vezes forem necessárias para cadastrar todos os horários em todos os dias.



d) A classe *HorarioRemedio* foi criada incluindo o atributo *data*, para permitir o cadastramento de horários diferentes para cada dia. Se não fosse por essa abordagem, poderíamos ter criado um atributo *horário: time* na classe *PrescricaoRemedio*, com multiplicidade de [1..\*], prevendo de uma a várias doses (horários) por dia.

## 5. EXERCÍCIO: GASTOS DIÁRIOS

**CENÁRIO:** As informações a seguir se referem à Planilha Excel de Vera, que faz o controle de seus gastos diários.

Para cada gasto, Vera cadastra: o tipo do gasto (remédio, roupa, refeição etc.), a data do gasto, o valor gasto e a forma de pagamento (dinheiro, cheque, cartão ou cheque pré).

No final do mês, Vera lista o total dos gastos mensais, agrupados por tipo de gasto e exibindo o quanto foi gasto em cada tipo de forma de pagamento.

### EXERCÍCIO:

Identifique as classes, atributos e métodos desse cenário.  
Represente os relacionamentos como atributos derivados.

### RESOLUÇÃO:

Classe	Atributos	Métodos
<b>TipoGasto</b>	<i>descricaoTipo</i> : string	cadastrar
<b>Gasto</b>	/tipo : Classe TipoGasto data : date valor : real formaPagamento : Enum- FormaPagamento	cadastrar gerarRelatorioMen- sal(mesAno: string)
«enumeration» <b>EnumFormaPaga- mento</b>	dinheiro cheque cartao chequePre	

### COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) A classe *enumeration* foi criada para suportar o domínio do atributo *FormaPagamento*, que consistia de valores fixos e com características de

rara mutação. Numa aplicação de contexto maior, essa classe poderia deixar de ser *enumeration* para se transformar numa classe de negócios.

b) Todo gasto refere-se a somente um tipo. Isso cria um relacionamento da classe *Gasto* com a classe *TipoGasto*, representada pelo atributo *tipo*.

c) Toda a complexidade do relatório mensal está encapsulada no método *gerarRelatorioMensal*.

## 6. EXERCÍCIO: COMANDA ELETRÔNICA

**CENÁRIO:** As informações a seguir se referem a uma aplicação de controle de comanda eletrônica da padaria Doce Sabor do Seu Joaquim.

O cliente usa uma comanda eletrônica durante suas compras na Padaria. A cada produto consumido, o atendente registra em sua comanda (que possui uma numeração) o produto e a quantidade.

Ao passar no caixa na saída da Padaria, a Caixa lê os gastos da comanda, finalizando a compra. Na leitura da comanda, verifica-se o valor unitário de cada produto a fim de calcular o valor total da compra.

### EXERCÍCIO:

Identifique as classes, atributos e métodos desse cenário.  
Represente os relacionamentos como atributos derivados.

### RESOLUÇÃO:

Classe	Atributos	Métodos
<b>Produto</b>	codigo : string descricao : string precoUnitario : real	cadastrar
<b>Comanda</b>	numero : integer /valorTotal : real /itens : Coleção de ItemComanda	registrarItem( item : ItemComanda ) fecharComanda
<b>ItemComanda</b>	/produto : Produto quantidade : integer	cadastrar

**COMENTÁRIOS SOBRE A RESOLUÇÃO:**

a) O atributo *valorTotal* da classe *Comanda* é derivado, porque é o resultado do cálculo do método *fecharComanda*. Esse método obtém todos os itens da *Comanda* e para cada um verifica o preço Unitário do produto, calculando o sub-total, de acordo com a seguinte fórmula:

$$\text{ItemComanda.quantidade} * \text{ItemComanda.produto.precoUnitario}$$

O total da *Comanda* (atributo *valorTotal*) é o somatório dos subtotais.

b) Toda comanda possui uma lista de itens. Isso cria um relacionamento da classe *Comanda* com a classe *ItemComanda*, representada pelo atributo *itens*. Por sua vez, um item de comanda se refere a somente um produto. Esta representação é feita no atributo *produto*.

**7. EXERCÍCIO: LISTA DE COMPRAS**

**CENÁRIO:** Carolina controla em Excel uma planilha com a sua lista de compras mensal. Ela cadastra o nome do produto, a unidade de compra, a quantidade prevista para um mês, a quantidade que efetivamente será comprada e o preço estimado (atualizado todo mês).

**IMAGEM DA PLANILHA:**

Produto	Unidade de compra	Qtd Mês	Qtd Compra	Preço Estimado
Arroz	Kg	8	7	1,80
Feijão	Kg	6	6	2,10
Açúcar	Kg	3	2	1,05
Carne	Kg	6	7,5	8,00
...				
<b>Total Estimado</b>				<b>150,00</b>

**ESCLARECIMENTOS SOBRE O PROBLEMA:**

a) A quantidade de compra pode variar em virtude de sobra de um mês para o outro, ou da necessidade de um gasto maior no mês. Por exemplo: almoço em família.

b) As compras são feitas pela própria Carolina. Por esse motivo, ela não vê necessidade de relacionar as marcas dos produtos.

c) Mensalmente, Carolina analisa o quanto pagou por cada produto, e se achar necessário, atualiza o preço estimado de cada produto.

**EXERCÍCIO:**

Identifique as classes, atributos e métodos desse cenário.  
Represente os relacionamentos como atributos derivados.

**RESOLUÇÃO:**

Classe	Atributos	Métodos
<b>ListaCompra</b>	mes : integer /itens : Coleção de ItemCompra /totalEstimado : real	cadastrar obterTotalEstimado(mês : integer)
<b>ItemCompra</b>	/produto : Classe Produto qtdEfetivaCompra : real	cadastrar
<b>Produto</b>	nome : string unidadeCompra : string qtdPrevistoMes : real precoEstimado : real	cadastrar

**COMENTÁRIOS SOBRE A RESOLUÇÃO:**

a) O método *obterTotalEstimado* é um método que atua sobre a coleção de objetos *ItemCompra*, ou seja, só podemos saber qual é o total estimado de toda a Lista de Compras (conjunto dos itens de compra), se quantificarmos o subtotal de cada item. Sendo assim, podemos concluir que o atributo *totalEstimado* que pertence à Lista de Compras é derivado, pois é o resultado desse método. O algoritmo do método *obterTotalEstimado(mês: integer)* é:

```
define objItemCompra : ItemCompra

obter a coleção de objItemCompra,
    onde ListaCompra.mes = parâmetro mês

total ← 0
para cada objItemCompra
    subtotal ←      objItemCompra.qtdEfetivaCompra *
                   objItemCompra.produto.precoEstimado
    total ← total + subtotal
fim-para
obterTotalEstimado ← total
```



b) Toda lista de compras possui um conjunto de itens. Isso cria um relacionamento da classe *ListaCompra* com a classe *ItemCompra*, representada pelo atributo *itens*. Por sua vez, um item de compra se refere a somente um produto. Essa representação se dá por meio do atributo *produto*.

## 8. EXERCÍCIO: LISTA DE COMPRAS (VARIAÇÃO A)

**CENÁRIO:** Carolina não tem mais tempo de fazer as compras pessoalmente. Precizou detalhar o produto, de forma a lhe permitir delegar essa tarefa a outra pessoa. Além disso, não quer que paguem um valor absurdo por algum produto. Sendo assim, incluiu em sua planilha as colunas "preço máximo já comprado" e "preço máximo a pagar" no mês corrente, onde esta última coluna é calculada a partir da coluna anterior acrescida de 5%. O "preço máximo já comprado" é inserido na planilha, a partir das compras efetivamente realizadas.

### EXERCÍCIO:

Quais são os atributos e/ou métodos que precisam ser incluídos nas classes do exercício anterior, para refletir esse novo cenário.

### RESOLUÇÃO:

Classe	Atributos	Métodos
<b>ListaCompra</b>	mes : integer /itens : Coleção de ItemCompra /totalEstimado : real	cadastrar obterTotalEstimado(mês : integer)
<b>ItemCompra</b>	/produto : Classe Produto qtdEfetivaCompra : real <b>/precoMaximoMes : real</b>	cadastrar
<b>Produto</b>	nome : string <b>descricaoProduto : string</b> unidadeCompra : string qtdPrevistoMes : real <b>precoMaximoAnterior : real</b> <del>precoEstimado : real</del>	cadastrar

### COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) Os atributos e ou métodos criados estão em negrito. O que foi retirado consta riscado.

b) Com a criação dos atributos *precoMaximoAnterior* e *precoMaximoMes*, eliminou-se o atributo *precoEstimado*. Essa informação será obtida a partir do atributo *precoMaximoAnterior*.

c) O atributo *precoMaximoAnterior* é cadastrado pelo usuário.

d) O algoritmo do método *obterTotalEstimado* foi alterado:

```
define objItemCompra : ItemCompra

obter a coleção de objItemCompra,
    onde ListaCompra.mes = parâmetro mês

total ← 0
para cada objItemCompra
    subtotal ← objItemCompra.qtdEfetivaCompra *
               objItemCompra.produto.precoMaximoAnterior
    total ← total + subtotal
fim-para
obterTotalEstimado ← total
```

e) O atributo *precoMaximoMes* é derivado, pois ele não recebe um valor diretamente do usuário. Seu valor é derivado de um cálculo feito sobre o atributo *precoMaximoAnterior*, considerando a fórmula a seguir. Esse atributo também poderia ter sido definido como um método do tipo função, cujo retorno seria o preço máximo do mês.

```
ItemCompra.precoMaximoMes = ItemCompra.produto.precoMaximoAnterior * 1.05
```

## 9. EXERCÍCIO: LISTA DE COMPRAS (VARIAÇÃO B)

**CENÁRIO:** Carolina está assustada com a variação de preço de um Supermercado para outro. Tem feito compras (ou pedido para fazer) em até três Supermercados diferentes. Sendo assim, resolveu melhorar sua planilha. Criou uma segunda planilha que contém o preço mais baixo que ela pagou num determinado mês, indicando a que supermercado pertence. Veja a seguir:

### IMAGEM DA PLANILHA:

Produto	Mês de Compra	Valor Compra	Supermercado
Arroz	julho	1,80	XXX
Arroz	agosto	1,75	YYY
Arroz	setembro	2,05	XXX
Feijão	julho	2,10	XXX
Feijão	agosto	2,25	XXX

**EXERCÍCIO:**

Quais são os atributos e/ou métodos que precisam ser incluídos nas classes do exercício anterior, para refletir esse novo cenário. Verifique se há necessidade de criar novas classes.

**RESOLUÇÃO:**

Classe	Atributos	Métodos
<b>ListaCompra</b>	mes : integer /itens : Coleção de ItemCompra /totalEstimado : real	cadastrar obterTotalEstimado(mês : integer)
<b>ItemCompra</b>	/produto : Classe Produto qtdEfetivaCompra : real /precoMaximoMes : real	cadastrar
<b>Produto</b>	nome : string descricaoProduto : string unidadeCompra : string qtdPrevistoMes : real /precoMaximoAnterior : real	cadastrar
<b>HistoricoCompra</b>	mes : integer /produto : Classe Produto valorPago : real /supermercado : Classe Supermercado	cadastrar
<b>Supermercado</b>	nome : string	cadastrar

**COMENTÁRIOS SOBRE A RESOLUÇÃO:**

- a) O atributo *mes* da classe *HistoricoCompra* está definido como integer, para armazenar o número correspondente ao mês do ano.
- b) O atributo *precoMaximoAnterior* não é mais lançado manualmente pelo usuário. Agora, ele é obtido automaticamente por uma fórmula, que verifica o maior valor pago dentre os últimos meses, para o produto em questão. Exemplo: na planilha de exemplo, para o produto arroz, o resultado seria 2,05 e para o produto feijão seria 2,25.
- c) Cada histórico de compra está associado a um supermercado e a um produto. Isso cria os relacionamentos da Classe *HistoricoCompra* com a Classe *Produto* e a Classe *Supermercado*. Na classe *HistoricoCompra* é

que se armazena o valor pago a cada mês, indicando em que supermercado foi feita a compra.

## 10. EXERCÍCIO: LISTA DE COMPRAS (VARIAÇÃO C)

**CENÁRIO:** Carolina deseja saber qual o Supermercado que apresentou mais produtos baratos, num determinado mês.

### EXERCÍCIO:

A resolução dessa situação se dá com a inclusão de um método.  
Sugira um nome para esse método, indicando sua assinatura.  
A que classe pertence esse método e como seria seu algoritmo?

### RESOLUÇÃO:

#### MÉTODO DA CLASSE HISTORICOCOMPRA:

obterSupermercadoMaisVantajoso(mesReferencia: integer): Supermercado

#### ALGORITMO:

```
define supermercadoRef : Supermercado

Selecionar todos os objetos de HistoricoCompra,
    onde HistoricoCompra.mes = mesReferencia, agrupando por Supermercado
Para cada grupo de objHistoricoCompra.supermercado
    totalizar o atributo valorpago desse grupo
    se o total de valor pago for o maior obtido até o momento, guardar o
supermercado correspondente em supermercadoRef
fim-para
obterSupermercadoMaisVantajoso ← supermercadoRef
```

#### COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) Se a classe *HistoricoCompra* armazena o menor valor pago por um produto, num determinado mês, significa que o supermercado mais vantajoso é aquele onde se gastou mais, pois ofereceu mais produtos em oferta.

## 11. EXERCÍCIO: LISTA DE COMPRAS (VARIAÇÃO D)

**CENÁRIO:** Carolina deseja desconsiderar o preço de um determinado mês para cálculos de maior ou menor valor, ou ainda do supermercado mais vantajoso.



Exemplo (1): saber qual o Supermercado que apresentou mais produtos baratos, num determinado mês. Supondo que ela comprou um produto numa promoção relâmpago que ofereceu 50% de desconto, esse valor não pode ser parâmetro para suas compras futuras.

Exemplo (2): para calcular o *precoMaximoAnterior*, a aplicação pega o maior valor. Vamos supor que um determinado produto teve queda de preço. Isso significa que os meses de preço alto não podem ser considerados para as próximas compras.

**EXERCÍCIO:**

Em que classe deve ser incluído um atributo para resolver essa questão e como deve ser esse atributo?

**RESOLUÇÃO:**

**CLASSE:** HistoricoCompra  
**ATRIBUTO:** validoParaAnalise: boolean

**12. EXERCÍCIO: COLEÇÃO DE CD'S**

**CENÁRIO:** Adriano tem uma coleção grande de CD's e gostaria de cadastrar no seu Palm a lista desses CD's, pois às vezes nem sabe o que tem.

Ele pensou em cadastrar o nome do cantor(a) ou conjunto, o título do CD e o ano de lançamento.

**EXERCÍCIO:**

Identifique as classes, atributos e métodos desse cenário.  
 Represente os relacionamentos como atributos derivados.

**RESOLUÇÃO:**

Classe	Atributos	Métodos
Musico	nome : string ehSolo : boolean	cadastrar

Classe	Atributos	Métodos
CD	/musico : Classe Musico titulo : string ano : integer	cadastrar

**COMENTÁRIOS SOBRE A RESOLUÇÃO:**

- a) Todo CD está relacionado a um músico que pode ser um cantor solo ou um conjunto. Isso cria um relacionamento entre a classe *CD* e a classe *Musico*.
- b) O atributo *ehSolo* indica se o objeto se refere a um cantor solo ou a um conjunto.

**13. EXERCÍCIO: COLEÇÃO DE CD'S (VARIAÇÃO A)**

**CENÁRIO:** Adriano notou que alguns CD's são de coletâneas. Sendo assim, não têm apenas um cantor (ou conjunto) e sim vários. Ele quer cadastrar essa lista de músicos, sem relacioná-los às músicas. Deseja controlar também se o CD é de coletânea e se é duplo.

Adriano gostaria de ter cadastrada a lista das músicas de cada CD, com o tempo de duração de cada faixa.

São relatórios desejados: os CD's de um determinado músico e em quais CD's está uma determinada música.

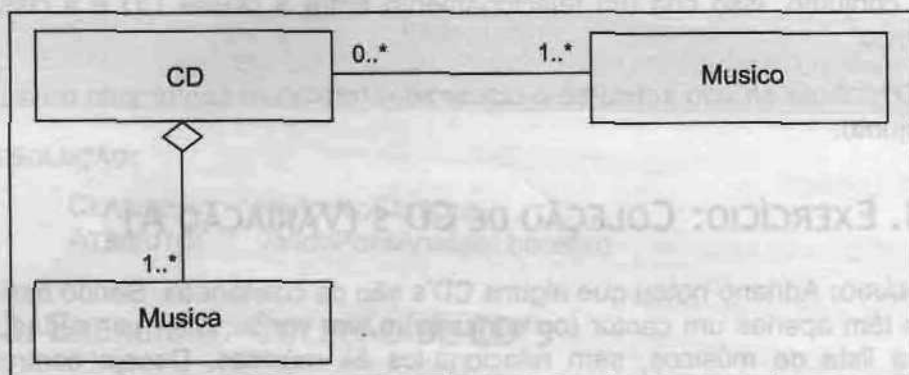
**EXERCÍCIO:**

Atualize a lista de classes, atributos e métodos para refletir esse cenário.  
 Represente os relacionamentos como atributos derivados.  
 Desenhe um diagrama de classes, somente com o nome das classes, especificando os relacionamentos e as multiplicidades.

**RESOLUÇÃO:**

Classe	Atributos	Métodos
Musico	nome : string ehSolo : boolean	cadastrar
CD	/musico : Classe Musico titulo : string ano : integer	cadastrar listarCdsPorMusico(musico: Classe Musico)

Classe	Atributos	Métodos
	ehDuplo : boolean ehColetânea : boolean /listaMusicas : Coleção de Musica	listarCdsPorMusica(musica: Classe Musica)
<b>Musica</b>	nome : string tempoFaixa : real	cadastrar



#### COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) Não há relacionamento da classe *Musica* com a classe *Musico*, pois este foi o requisito do usuário: não realizar o controle de que músico está interpretando que canção.

b) O atributo *listaMusicas* da classe *CD* representa o relacionamento de multiplicidade **1..\*** da classe *CD* com a classe *Musica*.

## 14. EXERCÍCIO: SALA DE REUNIÃO

**CENÁRIO:** Patrícia é secretária e dentre suas tarefas habituais existe a de controlar o uso das três salas de reunião, que são utilizadas por todos os setores da empresa.

Ela possui pastas de trabalho em Excel para cada mês do ano e, dentro de cada pasta, existem de vinte e oito a trinta e uma planilhas, uma para cada dia do mês.

Na planilha, ela criou a coluna de horários e três colunas, uma para cada sala de reunião. Num controle à parte, ela relaciona o nome do funcionário, seu cargo e ramal. Além da alocação, surge frequentemente a necessidade de realocação de uma reunião, mudando sala e/ou data e/ou horário.

Outra consulta constante que é feita à Patrícia, pelos Diretores, é sobre as salas que estarão livres numa determinada data, numa faixa de horário. Para cada sala, precisa-se saber o número de lugares. Veja o exemplo a seguir.

#### IMAGEM DA PLANILHA:

01/agosto

Horário	Sala 101	Sala 105	Sala 201
08h30m	Dr. Glauco – Assunto: processo empresa Mar e Lua		
09h		Dra. Maria – Assunto: palestra sobre a nova lei de falências	Mariana – assunto: análise de material
09h30			
10h			
10h30			
11h			Dra. Maria – assunto: reunião de projeto
11h30			

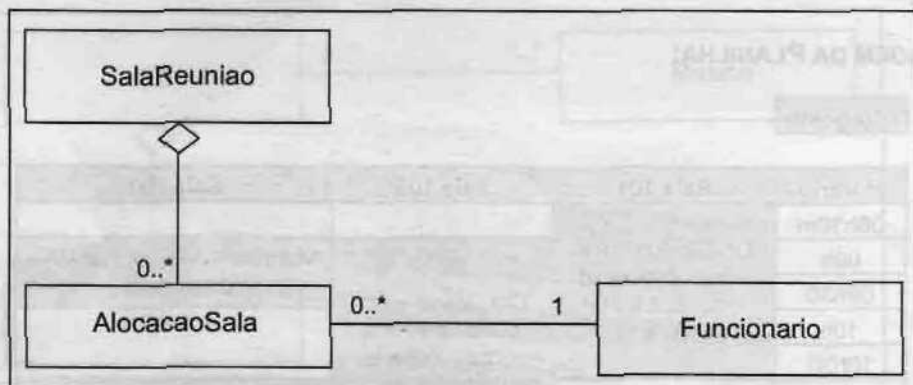
#### EXERCÍCIO:

Identifique as classes, atributos e métodos desse cenário.  
 Represente os relacionamentos como atributos derivados.  
 Desenhe um diagrama de classes, somente com o nome das classes, especificando os relacionamentos e as multiplicidades.

#### RESOLUÇÃO:

Classe	Atributos	Métodos
<b>AlocacaoSala</b>	/sala : Classe SalaReuniao dataAlocacao : date horaInicioAlocacao : time horaFimAlocacao : time /responsavelAlocacao : Classe Funcionario assunto : string	alocar realocar(novaData : date, novaHoraInicio: time, novaHoraFim: time, novaSala: Classe Sala) cancelar

Classe	Atributos	Métodos
<b>Funcionario</b>	nome : string cargo : string ramal : integer	cadastrar
<b>SalaReuniao</b>	numero : integer qtdLugares : integer	cadastrar <u>obterSalaLivre(data: date, horaInicio: time, horaFim: time) :</u> <u>coleção de SalaReuniao</u>

**COMENTÁRIOS SOBRE A RESOLUÇÃO:**

a) Toda alocação de sala registra qual sala foi alocada e qual foi o funcionário responsável pelo pedido de alocação. Isso cria os relacionamentos entre as classes *AlocaoSala*, *SalaReuniao* e *Funcionario*.

b) Repare que o cenário não citou a necessidade de se cadastrar o assunto da Reunião, mas essa informação é relevante, pois aparece em todas as alocações registradas na Planilha.

Isso mostra que, ao levantarmos os requisitos de uma aplicação, não basta levarmos em conta somente o que é dito. Precisamos analisar também o que é mostrado, na forma de fichas, documentos, formulários, planilhas, relatórios etc.

c) O método *obterSalaLivre* é de classe, pois sua execução não é sobre um objeto e sim sobre toda a coleção de objetos *SalaReuniao*. O resultado desse método é um conjunto (uma coleção) de objetos *SalaReuniao*, que pode ser um conjunto vazio (nenhuma sala disponível), um conjunto unitário ou retornar todas as salas de reunião.



## 15. EXERCÍCIO: SALA DE REUNIÃO (VARIAÇÃO A)

**CENÁRIO:** Os materiais de apoio são divididos entre as salas de reunião. Sendo assim, é preciso controlar os materiais que estão alocados e para quais reuniões. Por exemplo: retroprojeto – sala 101 na reunião de 9 às 10h30; flip – sala 102 na reunião de 10 às 12.

### EXERCÍCIO:

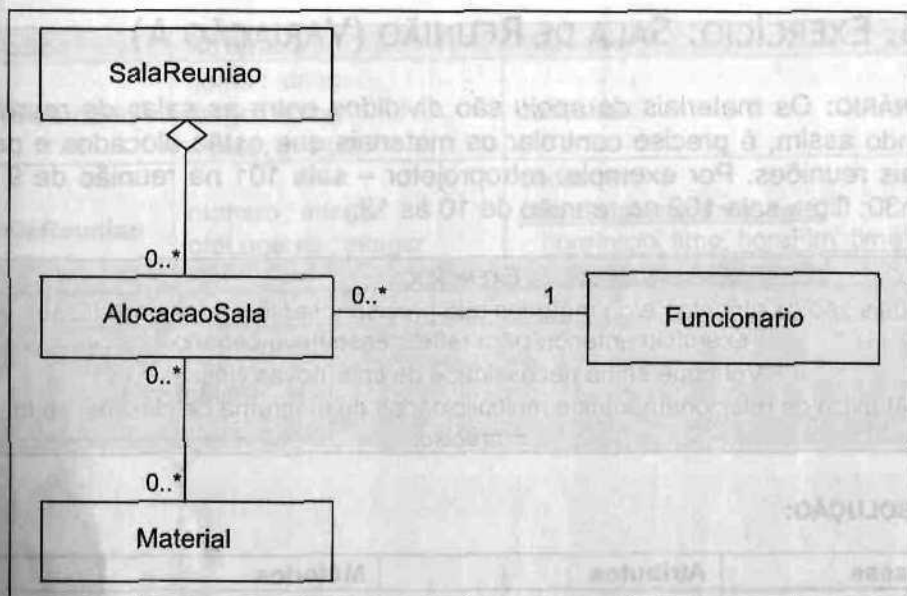
Quais são os atributos e/ou métodos que precisam ser incluídos nas classes do exercício anterior, para refletir esse novo cenário.

Verifique se há necessidade de criar novas classes.

Atualize os relacionamentos e multiplicidades do diagrama de classes, se for preciso.

### RESOLUÇÃO:

Classe	Atributos	Métodos
<b>AlocacaoSala</b>	/sala : Classe SalaReuniao dataAlocacao : date horaInicioAlocacao : time horaFimAlocacao : time /responsavelAlocacao : Classe Funcionario assunto : string /listaMateriaisAlocados : <b>Coleção de Material</b>	alocar realocar(novaData : date, novaHoraInicio: time, novaHoraFim: time, novaSala: Classe Sala) cancelar
<b>Funcionario</b>	nome : string cargo : string ramal : integer	cadastrar
<b>SalaReuniao</b>	numero : integer qtdLugares : integer	cadastrar <u>obterSalaLivre(data: date, horaInicio: time, horaFim: time) : coleção de SalaReuniao</u>
<b>Material</b>	nome : string	cadastrar



### COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) Foi incluído o atributo *listaMateriaisAlocados*, que está em negrito. Além disso, foi criada a classe *Material*.

b) Uma alocação de sala poderá ter, também, uma lista de materiais alocados. Isso cria o relacionamento entre a classe *AlocacaoSala* e *Material*, representada pelo atributo *listaMateriaisAlocados*.

## 16. EXERCÍCIO: HERANÇA

Observe as duas classes adiante:

Classe	Atributos	Métodos
<b>Funcionario</b>	matricula : integer nome : string dataNascimento : date /endereco : Classe Endereco /telsContato : Coleção de Telefone /cargo : Classe Cargo salario : real dataAdmissao : date	cadastrar obterIdade reajustarSalario (percentual : real) promover(novoCargo : Cargo)

Classe	Atributos	Métodos
<b>Cliente</b>	codigo : string nome : string dataNascimento : date /endereco : Classe Endereco /telsContato : Coleção de Telefone /profissao : Classe Profissao	cadastrar obterIdade

**EXERCÍCIO (A):**

Crie uma superclasse que contenha os atributos comuns, reformulando todas as classes.

**RESOLUÇÃO:**

Classe	Atributos	Métodos
<b>Pessoa</b>	nome : string dataNascimento : date /endereco : Classe Endereco /telsContato : Coleção de Telefone	cadastrar obterIdade
<b>Funcionário</b>	matricula : integer /cargo : Classe Cargo salario : real dataAdmissao : date	reajustarSalario (percentual : real) promover(novoCargo : Cargo)
<b>Cliente</b>	código : string /profissao : Classe Profissao	

**EXERCÍCIO (B):**

Que classes estão se relacionando com a classe Funcionário?

**RESOLUÇÃO:**

Por herança da classe Pessoa: Endereco e Telefone. Relacionando-se diretamente com Funcionário: Cargo.

**EXERCÍCIO (C):**

Que classes estão se relacionando com a classe Cliente?

**RESOLUÇÃO:**

Por herança da classe Pessoa: Endereco e Telefone. Relacionando-se diretamente com Cliente: Profissão.

## Capítulo 2

# Desenhando Diagrama de Classes a partir de Minicenários

Com a próxima lista de exercícios, você continuará a treinar seu poder de abstração dos conceitos de classe e atributos. Usaremos minicenários mais complexos que os apresentados no Capítulo 1. Desenharemos um diagrama de classes para cada um dos exercícios deste capítulo.

Repetiremos a fórmula de criar variações dos exercícios, que permitam a manutenção do modelo.

Nos exercícios deste capítulo incluiremos uma pequena lista de casos de uso, representados num diagrama de casos de uso. Entretanto, em virtude do número reduzido e da baixa complexidade dos mesmos, não escreveremos o descritivo de seus cenários. Tentaremos abstrair as classes e seus atributos diretamente da especificação do problema (minicenário).

### CONVENÇÕES DO CAPÍTULO:

- Todas as classes contêm atributos (com seus tipos), métodos (com seus parâmetros e tipo de retorno) e relacionamentos (com suas multiplicidades);
- Atributos derivados ou de classe também são representados. Atributos e métodos de classe são representados sublinhados;
- O diagrama de classes é da fase de análise. Sendo assim, não aparecerão atributos derivados que representem relacionamentos;
- Os métodos descritos foram deduzidos a partir do cenário. Entretanto, a lista completa dos métodos só é possível obter a partir de um diagrama de

seqüências, que tem por objetivo identificar a troca de mensagens existente entre objetos, em cada caso de uso;

– Os casos de uso Manter XXX indicam que o caso de uso tratará da consulta, inclusão, alteração e exclusão de XXX. Exemplo: Caso de Uso Manter Cliente é responsável pela consulta, inclusão, alteração e exclusão dos dados cadastrais de cliente;

– Os atributos e métodos seguem a nomenclatura da UML, ou seja, iniciam com letra minúscula, tendo a primeira letra de cada palavra (a partir da segunda) em maiúscula;

– Todo atributo cujo tipo seja uma classe *enumeration* não é definido como um atributo derivado, visto a classe *enumeration* atuar com um tipo de dado e não como um relacionamento.

Vamos exercitar nosso poder de abstração!

## 17. MINICENÁRIO: CONTROLE DE TAREFAS,

Arnaldo deseja escrever uma aplicação de controle de tarefas para colocar em seu Palm. As especificações da aplicação são as seguintes:

O cadastro de cada tarefa contém o número da prioridade, representado por um valor real. Isso permite entrar com intervalos intermediários. Além da prioridade, o cadastro deve conter: o nome da tarefa, a data limite de execução (se houver), o percentual já concluído e o detalhamento da tarefa.

Para cada tarefa há uma lista de itens que descrevem sua execução. Para cada item de execução, cadastram-se:

- o percentual correspondente
- a descrição da execução
- a data da execução (quando for concluída)

Quando uma tarefa receber 100% de execução, esta deve ser movida automaticamente para a lista de tarefas concluídas, podendo ser apagada, se for o caso.

Veja o exemplo desse controle em papel.



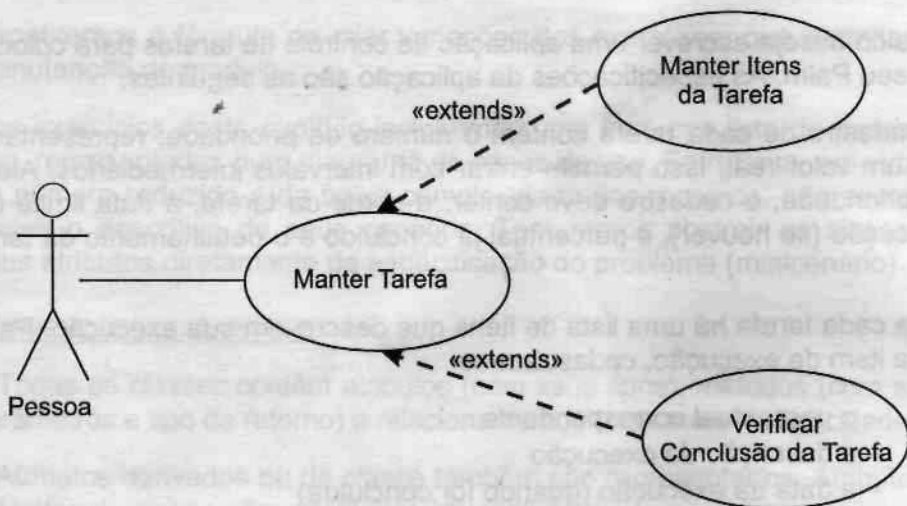
## 26 • Exercitando Modelagem em UML

TAREFA 1.1. – ANIVERSÁRIO DO FABIO		
Data limite	=	06/08/2005
Percentual já concluído	=	65%
Detalhamento	=	planejamento dos preparativos para a festa de aniversário do Fabio, no sábado, dia 6 de agosto.
Lista de Itens para serem executados:		
[ 20% ]	Aluguel do salão e da animação	- 01/03/2005
[ 20% ]	Encomenda do bolo, salgados e doces	- 15/07/2005
[ 05% ]	Compra das bebidas	-
[ 25% ]	Compra dos itens para a decoração	- 01/07/2005
[ 30% ]	Arrumação do salão	-

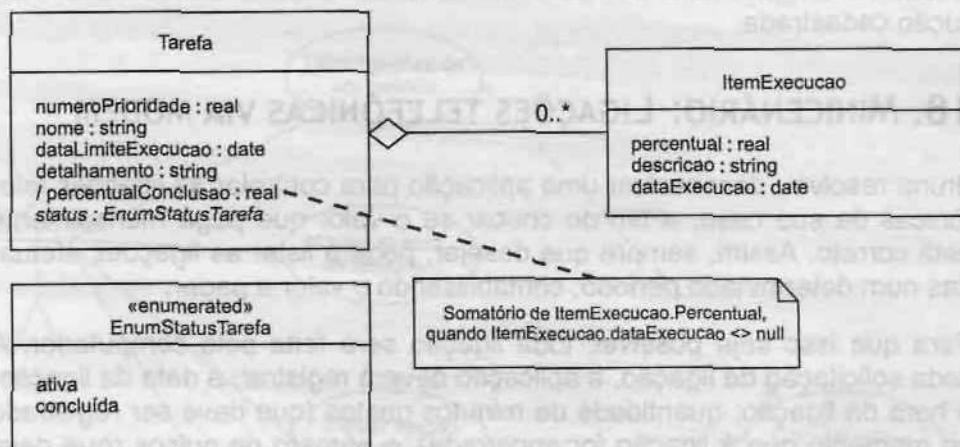
### EXERCÍCIO:

1. Desenhe o diagrama de casos de uso.
2. Desenhe o diagrama de classes completo.

### RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



## RESOLUÇÃO (DIAGRAMA DE CLASSES):



## COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) A lista de itens de execução faz parte da tarefa que representa o todo. Desta forma, o relacionamento foi representado como uma agregação.

b) O atributo *percentualConclusao* não é um atributo que será cadastrado. Seu valor é sempre calculado a partir do somatório dos percentuais dos itens de execução que já foram concluídos (ou seja, que tenham uma data de execução). Por este motivo, ele foi definido como atributo derivado. A regra de sua derivação foi explicada num item de nota, pertencente ao atributo. Existem diversas formas de expressar esse tipo de regra.

c) O atributo *status* poderia ter sido expresso como um atributo identificado como *"ativo: boolean"*. Entretanto, isso restringiria o atributo a dois valores. Ao se definir o tipo enumerado, já prevemos uma manutenção evolutiva do modelo, facilitando a inclusão de outros status.

d) O caso de uso *Verificar Conclusão da Tarefa* tem por objetivo verificar se todos os itens associados a uma tarefa já foram concluídos, alterando automaticamente o status da tarefa para concluída.

e) Como o único usuário dessa aplicação é o Arnaldo, o ator foi identificado como Pessoa.

f) No caso de uso *Manter Itens da Tarefa*, o ator cadastra, inicialmente, somente o percentual da tarefa e sua descrição. Depois, a cada tarefa

concluída, o ator volta nesse caso de uso para lançar a data de execução. O item de tarefa é considerado como concluído se tiver uma data de execução cadastrada.

## 18. MINICENÁRIO: LIGAÇÕES TELEFÔNICAS VIA MODEM

Bruna resolveu desenvolver uma aplicação para controlar as ligações telefônicas de sua casa, a fim de checar se o valor que paga mensalmente está correto. Assim, sempre que desejar, poderá listar as ligações efetuadas num determinado período, contabilizando o valor a pagar.

Para que isso seja possível, toda ligação será feita pelo computador. A cada solicitação de ligação, a aplicação deverá registrar: a data da ligação, a hora da ligação, quantidade de minutos gastos (que deve ser registrado no momento que a ligação for encerrada), o número de pulsos (que deve ser calculado pela aplicação) e o telefone para onde se discou.

A aplicação permitirá o controle de uma agenda de telefones, com número do telefone e nome da pessoa de contato. O usuário poderá escolher, no momento da ligação, se deseja um dos registros da agenda ou se digitará diretamente o número do telefone.

A forma de cálculo dos pulsos considera os seguintes critérios:

- A ligação ao ser completada já conta um pulso. A partir daí, a cada quatro minutos de conversação concluída, cobra-se mais um pulso.
- Cada pulso custa R\$ 0,08 para ligações locais.

Exemplo:

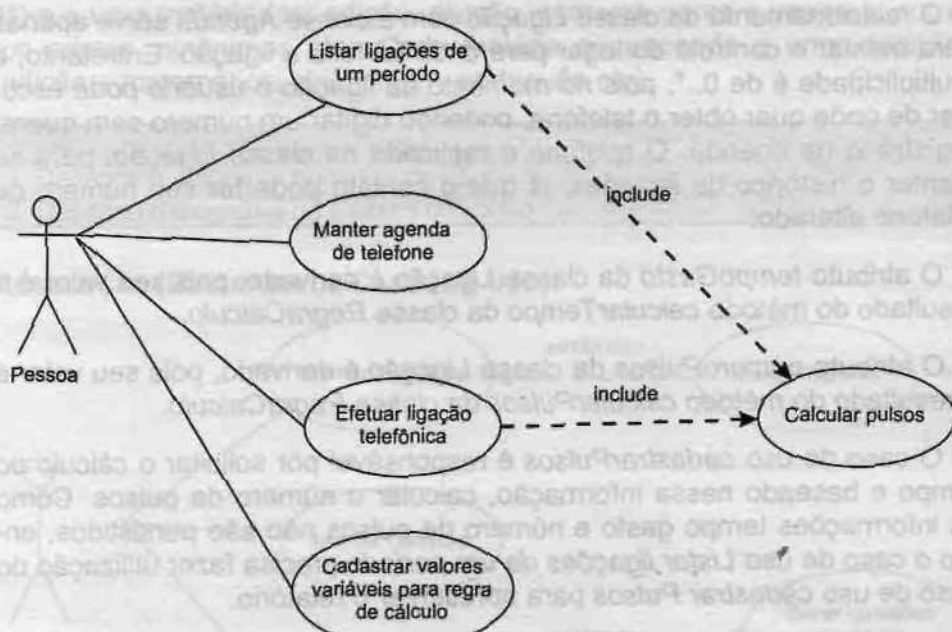
Ligação de 2m	- 1 pulso
Ligação de 4m30s	- 2 pulsos
Ligação de 8m	- 3 pulsos

- Os finais de semana possuem uma promoção. Cada ligação contabiliza somente um pulso, independente do número de minutos de conversação.

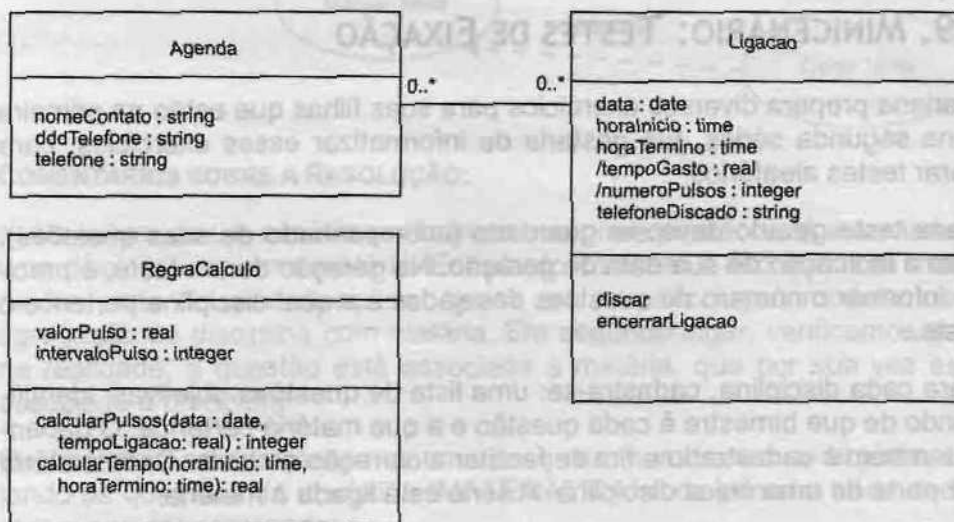
### EXERCÍCIO:

1. Desenhe o diagrama de casos de uso.
2. Desenhe o diagrama de classes completo.

## RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



## RESOLUÇÃO (DIAGRAMA DE CLASSES):



**COMENTÁRIOS SOBRE A RESOLUÇÃO:**

- a) O relacionamento da classe *Ligação* com a classe *Agenda* serve apenas para manter o controle do lugar para onde foi feita a ligação. Entretanto, a multiplicidade é de 0..\*, pois no momento da ligação o usuário pode escolher de onde quer obter o telefone, podendo digitar um número sem querer registrá-lo na agenda. O *telefone* é replicado na classe *Ligação*, para se manter o histórico de ligações, já que o contato pode ter seu número de telefone alterado.
- b) O atributo *tempoGasto* da classe *Ligação* é derivado, pois seu valor é o resultado do método *calcularTempo* da classe *RegraCalculo*.
- c) O atributo *numeroPulsos* da classe *Ligação* é derivado, pois seu valor é o resultado do método *calcularPulsos* da classe *RegraCalculo*.
- d) O caso de uso *cadastrarPulsos* é responsável por solicitar o cálculo do tempo e baseado nessa informação, calcular o número de pulsos. Como as informações tempo gasto e número de pulsos não são persistidos, então o caso de uso *Listar ligações de um período* precisa fazer utilização do caso de uso *cadastrar Pulsos* para apresentar o relatório.
- e) Como o único usuário dessa aplicação é a Bruna, o ator foi identificado como Pessoa.

## 19. MINICENÁRIO: TESTES DE FIXAÇÃO

Mariana prepara diversos exercícios para suas filhas que estão na primeira e na segunda séries. Ela gostaria de informatizar esses exercícios, para gerar testes aleatórios.

Cada teste gerado deve ser guardado (acompanhado de suas questões), com a indicação de sua data de geração. Na geração de um teste, é preciso informar o número de questões desejadas e a qual disciplina pertence o teste.

Para cada disciplina, cadastra-se: uma lista de questões objetivas, identificando de que bimestre é cada questão e a que matéria pertence. O gabarito também é cadastrado a fim de facilitar a correção do teste. Cada matéria faz parte de uma única disciplina. A série está ligada à matéria.

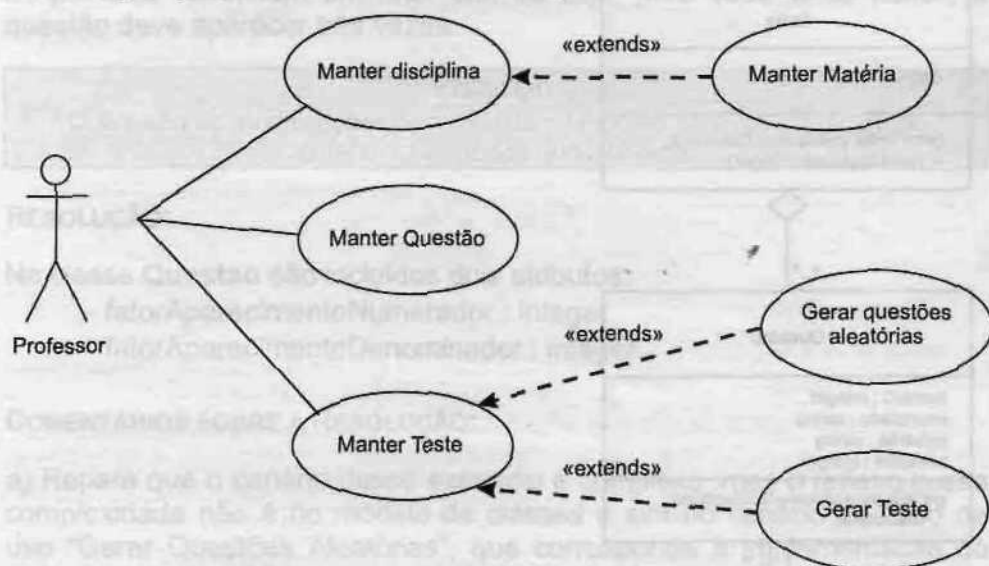


Por exemplo: para a disciplina de matemática, Mariana prepara um teste com 20 questões. Cada questão corresponde a um bimestre (1º, 2º, 3º ou 4º) e a uma matéria (ex: adição, divisão, números pares e ímpares, números primos, sinônimos etc). Cada matéria corresponde a uma disciplina (adição – matemática; sinônimos – português etc).

#### EXERCÍCIO:

1. Desenhe o diagrama de casos de uso.
2. Desenhe o diagrama de classes completo.

#### RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



#### COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) Na leitura de um cenário verificamos que uma questão está associada a uma disciplina e a uma matéria. Entretanto, também é dito que uma matéria faz parte de uma disciplina. Assim, em primeiro lugar, modelamos a agregação de disciplina com matéria. Em segundo lugar, verificamos que, na realidade, a questão está associada à matéria, que por sua vez está associada à disciplina.

Desta forma, se desejássemos uma pesquisa, num banco de objetos, obtendo as questões da disciplina "MATEMÁTICA" e da primeira série, teríamos o seguinte comando:

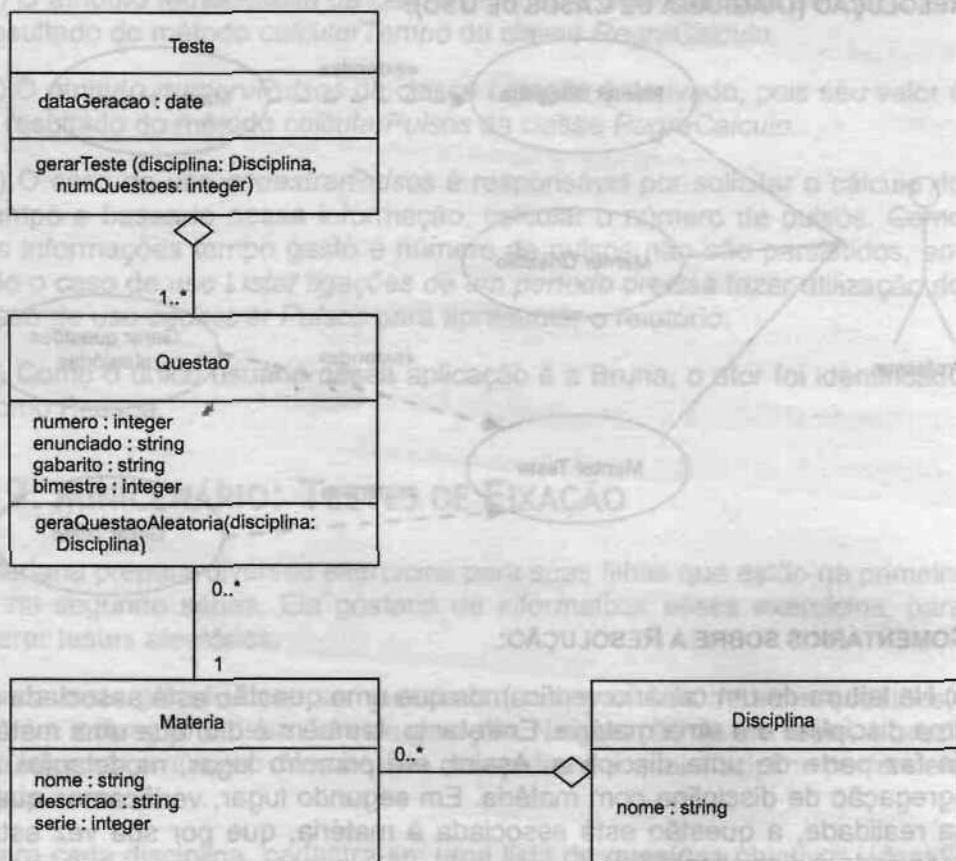
```

SELECT Questao.numero, Questao.enunciado
FROM Questao
WHERE Questao.materia.disciplina.nome = "MATEMÁTICA"
AND Questao.materia.serie = 1

```

b) Como o único usuário dessa aplicação é a Mariana, e ela assume o papel de professora particular ao preparar os testes para as filhas, o ator foi identificado como Professor.

### RESOLUÇÃO (DIAGRAMA DE CLASSES):



## 20. MINICENÁRIO: TESTES DE FIXAÇÃO (VARIAÇÃO A)

Conforme Mariana foi aplicando os testes às suas filhas, ela percebeu que algumas questões são mais problemáticas que outras. Assim, desenvolveu uma fórmula para a geração dos testes aleatórios, que possa levar em conta que algumas questões devem ser fixas, e outras devem aparecer um número de vezes dentro de um determinado intervalo de testes. Por exemplo: as questões sobre divisão (matéria), na primeira série, têm fator 1/1, que indicam que para cada um teste (denominador), deve aparecer uma vez (numerador), ou seja, 100% das vezes. Já as questões de sinônimos, na primeira série, têm um fator 3/5, ou seja, para cada cinco testes, a questão deve aparecer três vezes.

### EXERCÍCIO:

Quais são as modificações necessárias no modelo para refletir o cenário anterior? Responda textualmente.

### RESOLUÇÃO:

Na classe **Questao** são incluídos dois atributos:

- fatorAparecimentoNumerador : integer
- fatorAparecimentoDenominador : integer

### COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) Repare que o cenário desse exercício é complexo, mas o reflexo dessa complexidade não é no modelo de classes e sim no cenário do caso de uso "Gerar Questões Aleatórias", que corresponde à implementação do método *geraQuestaoAleatoria*. No modelo de classes, bastou acrescentar dois atributos.

## 21. MINICENÁRIO: FESTAS INFANTIS

Rafaela possui vários temas de festas infantis para aluguel.

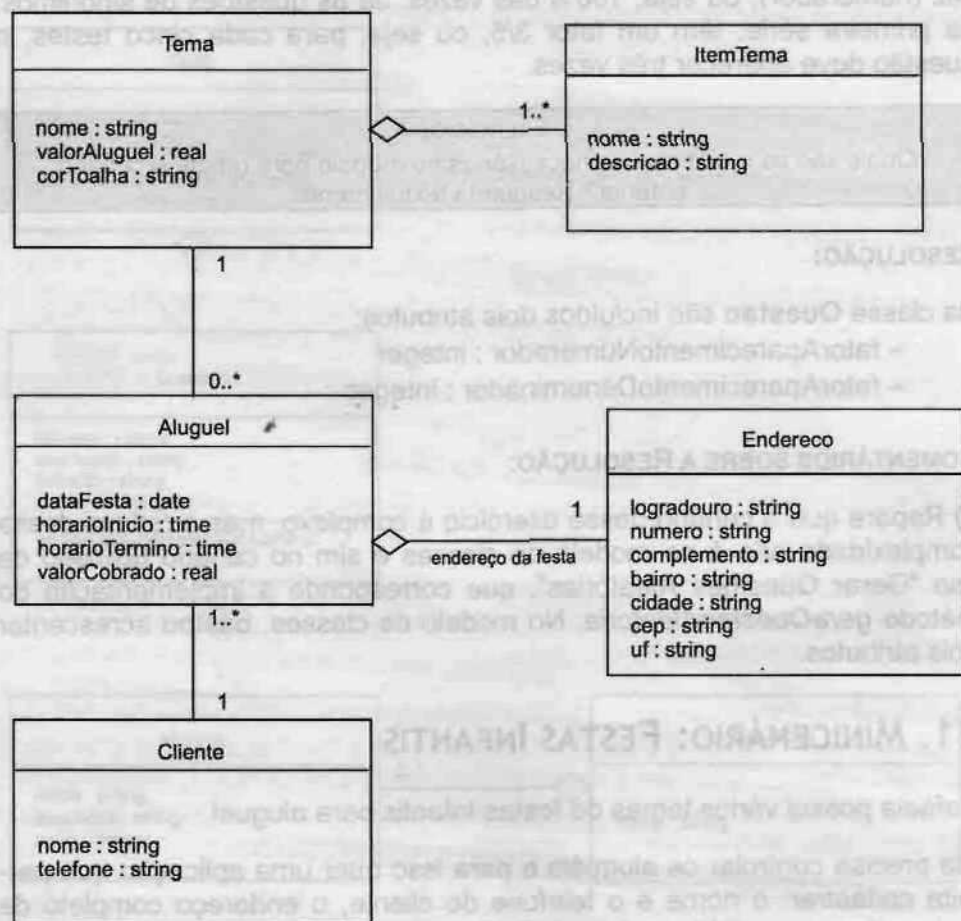
Ela precisa controlar os aluguéis e para isso quer uma aplicação que permita cadastrar: o nome e o telefone do cliente, o endereço completo da festa, o tema escolhido, a data da festa, a hora de início e término da festa.

Além disso, para alguns clientes antigos, Rafaela oferece descontos. Sendo assim, é preciso saber o valor realmente cobrado num determinado aluguel.

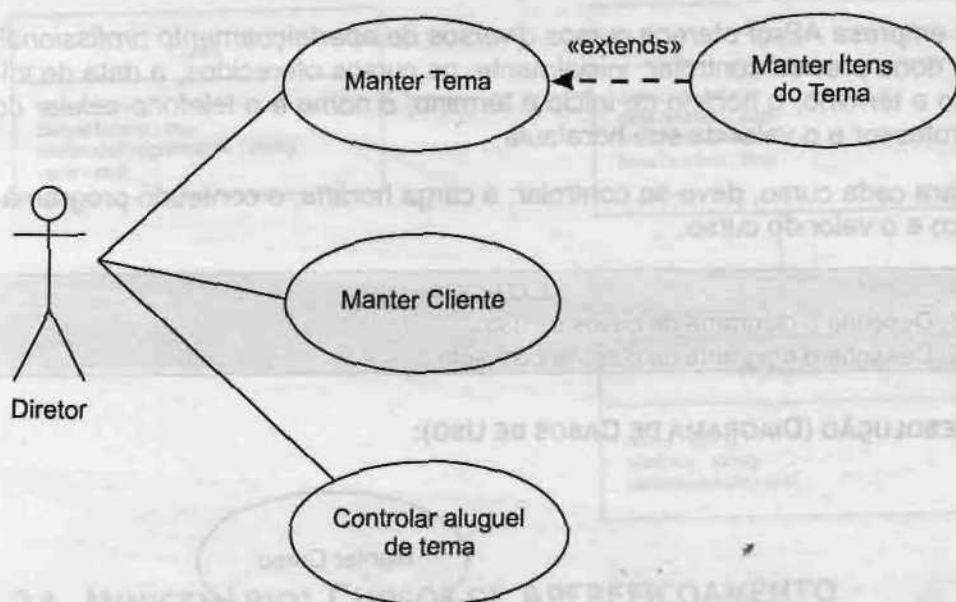
Para cada tema, é preciso controlar: a lista de itens que compõem o tema (ex: castelo, boneca da Cinderela, bruxa etc.), o valor do aluguel e a cor da toalha da mesa que deve ser usada com o tema.

**EXERCÍCIO:**

1. Desenhe o diagrama de casos de uso.
2. Desenhe o diagrama de classes completo.

**RESOLUÇÃO (DIAGRAMA DE CLASSES):**

## RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



## COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) Como o único usuário dessa aplicação é a Rafaela, e ela assume o papel de dona de uma microempresa informal, o ator foi identificado como Diretor.

## 22. MINICENÁRIO: FESTAS INFANTIS (VARIAÇÃO A)

Rafaela percebeu que também precisa controlar o pagamento de sinal e a data da quitação do serviço.

## EXERCÍCIO:

Quais são as modificações necessárias no modelo para refletir esse cenário?  
Responda textualmente.

## RESOLUÇÃO:

Na classe **Aluguel** são incluídos dois atributos:

- valorPagoSinal : real
- dataQuitacao : date



## 23. MINICENÁRIO: CURSOS DE APERFEIÇOAMENTO

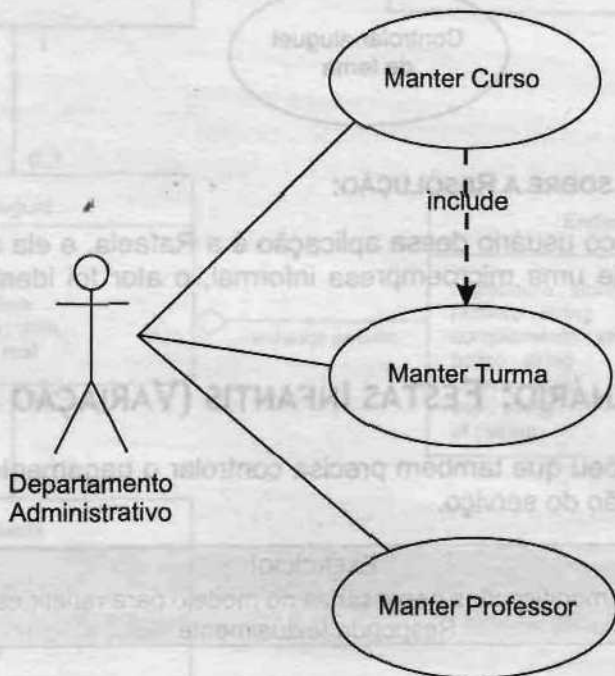
A empresa AProf oferece cursos diversos de aperfeiçoamento profissional. O dono precisa controlar, inicialmente: os cursos oferecidos, a data de início e término, o horário de início e término, o nome e o telefone celular do professor e o valor da sua hora/aula.

Para cada curso, deve-se controlar: a carga horária, o conteúdo programático e o valor do curso.

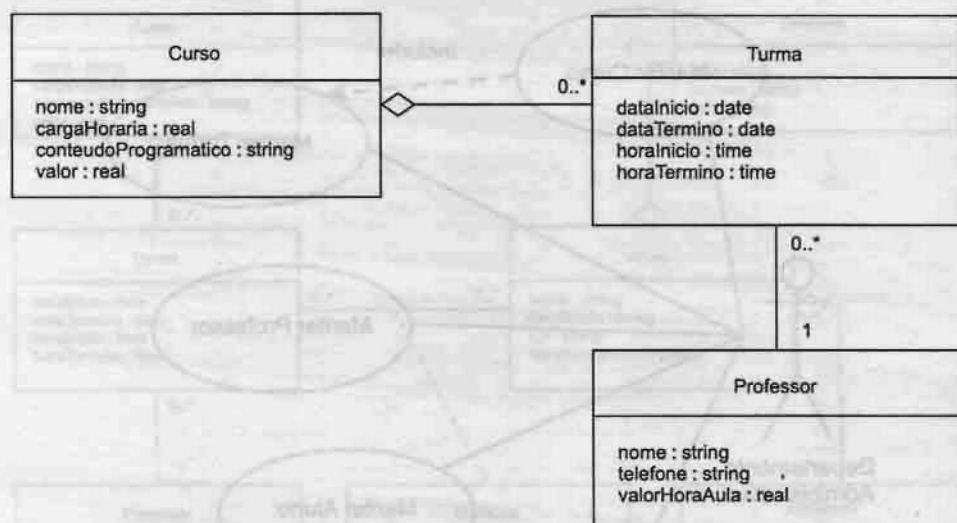
### EXERCÍCIO:

1. Desenhe o diagrama de casos de uso.
2. Desenhe o diagrama de classes completo.

### RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



## RESOLUÇÃO (DIAGRAMA DE CLASSES):



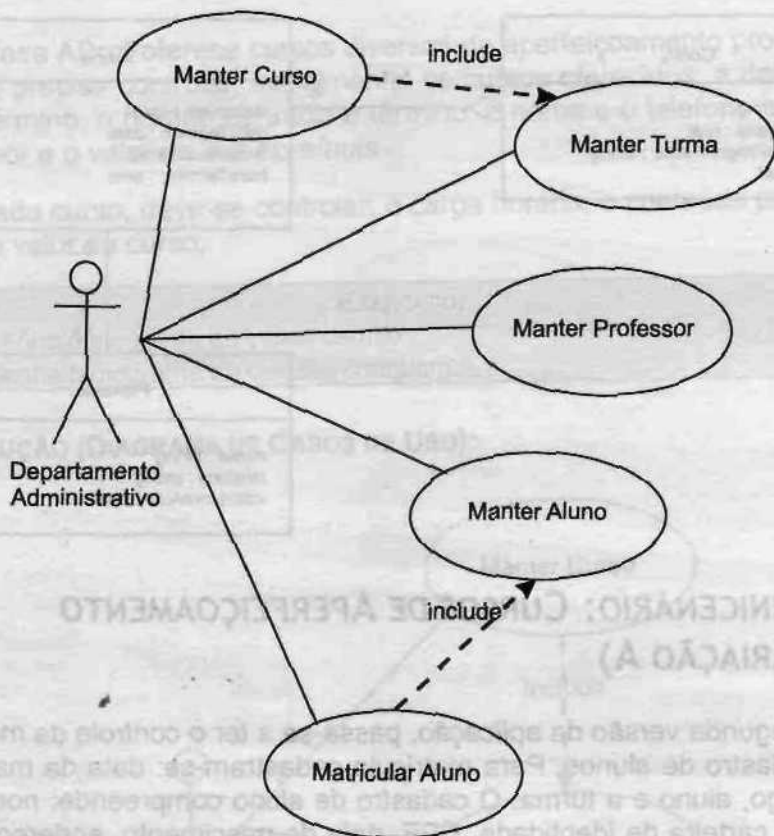
## 24. MINICENÁRIO: CURSOS DE APERFEIÇOAMENTO (VARIAÇÃO A)

Numa segunda versão da aplicação, passa-se a ter o controle da matrícula e do cadastro de alunos. Para matrícula cadastram-se: data da matrícula, valor pago, aluno e a turma. O cadastro de aluno compreende: nome, número da carteira de identidade, CPF, data de nascimento, endereço completo e telefones de contato.

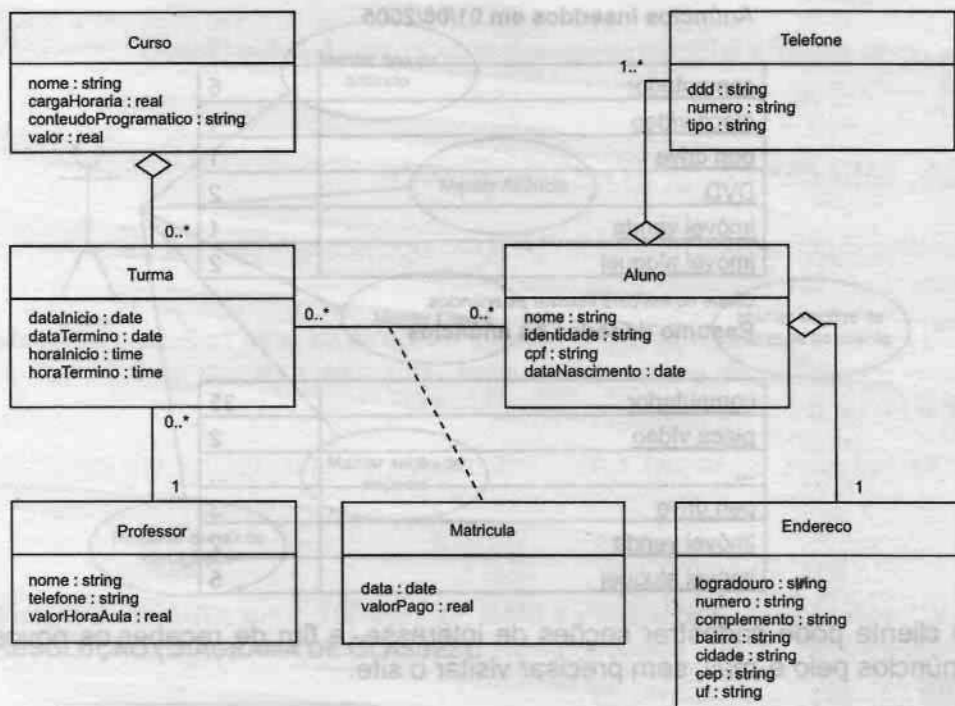
### EXERCÍCIO:

Atualize o diagrama de casos de uso e o modelo de classes.

RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



## RESOLUÇÃO (DIAGRAMA DE CLASSES):



## 25. MINICENÁRIO: CLASSIFICADOS NA WEB

Dalila está oferecendo aos amigos da escola, do bairro e do clube, o serviço de classificados pela Web.

Quem anuncia paga R\$ 2,00 pelo anúncio simples de 20 palavras ou R\$ 5,00 pelo anúncio destaque de página, que pode conter até 50 palavras e uma imagem. O anúncio fica disponível por quinze dias. Não entram na contagem: o valor do produto, o texto título, o nome do contato, até dois telefones de contato e uma observação sobre os telefones de até cinco palavras (por exemplo: de 18h às 20h).

Quem se torna assinante do classificado virtual recebe diariamente em seu e-mail o resumo de ofertas do site.

Exemplo da divulgação feita por e-mail:

**Anúncios inseridos em 01/08/2005**

<u>computador</u>	5
<u>placa vídeo</u>	1
<u>pen drive</u>	1
<u>DVD</u>	2
<u>imóvel venda</u>	1
<u>imóvel aluguel</u>	2

Clique no link para acessar os anúncios.

**Resumo de todos os anúncios**

<u>computador</u>	35
<u>placa vídeo</u>	2
...	...
<u>pen drive</u>	3
<u>imóvel venda</u>	2
<u>imóvel aluguel</u>	5

O cliente pode cadastrar seções de interesse, a fim de receber os novos anúncios pelo e-mail, sem precisar visitar o site.

O anúncio segue o formato desse exemplo:

**Seção Computador**

<b>Pentium 4 3.0 Ghz</b>	<b>R\$ 1.300,00</b>
Computador Pentium 4.0 3.0 Ghz, 256 Mb Memória, HD 40 Gb, gravador de CD, monitor Samsung 17".	
Tratar Marta	2222-1111 / 2111-2222 à noite

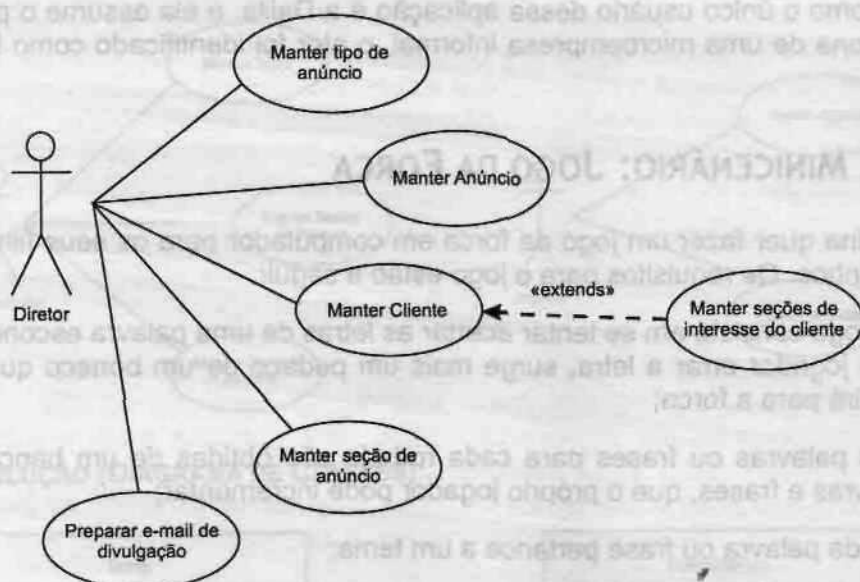
inserido em: 01/08/2005

**EXERCÍCIO:**

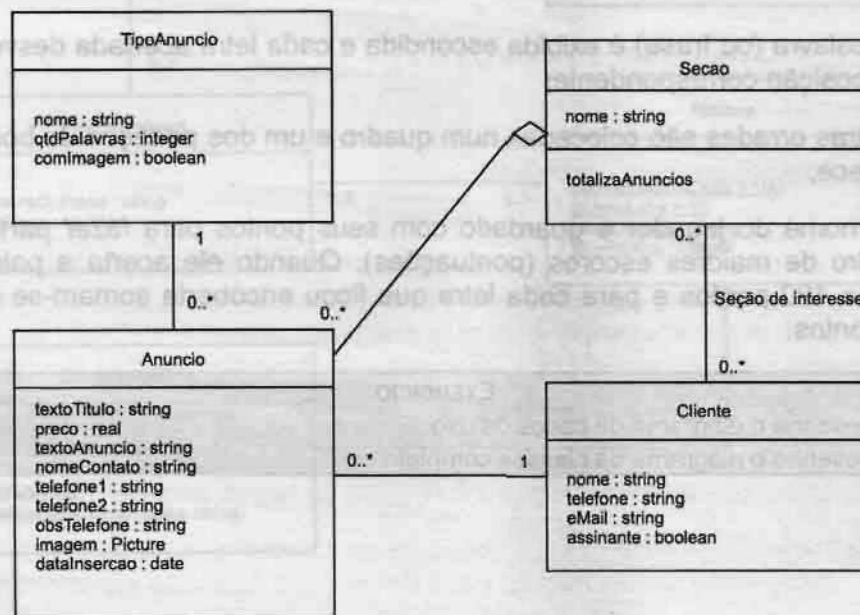
1. Desenhe o diagrama de casos de uso.
2. Desenhe o diagrama de classes completo.



# RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



# RESOLUÇÃO (DIAGRAMA DE CLASSES):



**COMENTÁRIOS SOBRE A RESOLUÇÃO:**

a) Como o único usuário dessa aplicação é a Dalila, e ela assume o papel de dona de uma microempresa informal, o ator foi identificado como Diretor.

**26. MINICENÁRIO: JOGO DA FORÇA**

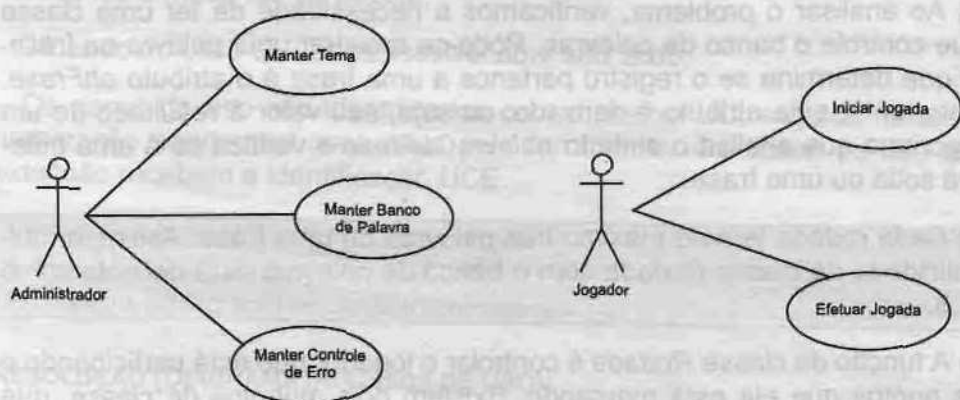
Cristina quer fazer um jogo da força em computador para os seus filhos e sobrinhos. Os requisitos para o jogo estão a seguir:

- O jogo consiste em se tentar acertar as letras de uma palavra escondida. Se o jogador errar a letra, surge mais um pedaço de um boneco que ao final irá para a forca;
- As palavras ou frases para cada rodada são obtidas de um banco de palavras e frases, que o próprio jogador pode incrementar;
- Cada palavra ou frase pertence a um tema;
- A cada rodada, a aplicação sorteia se mostrará uma, duas ou três palavras, ou uma frase e escolhe aleatoriamente o tema e as palavras (ou frase);
- A palavra (ou frase) é exibida escondida e cada letra acertada desvenda sua posição correspondente;
- Letras erradas são colocadas num quadro e um dos pedaços do boneco aparece;
- O nome do jogador é guardado com seus pontos para fazer parte do quadro de maiores escores (pontuações). Quando ele acerta a palavra, ganha 100 pontos e para cada letra que ficou encoberta somam-se mais 15 pontos.

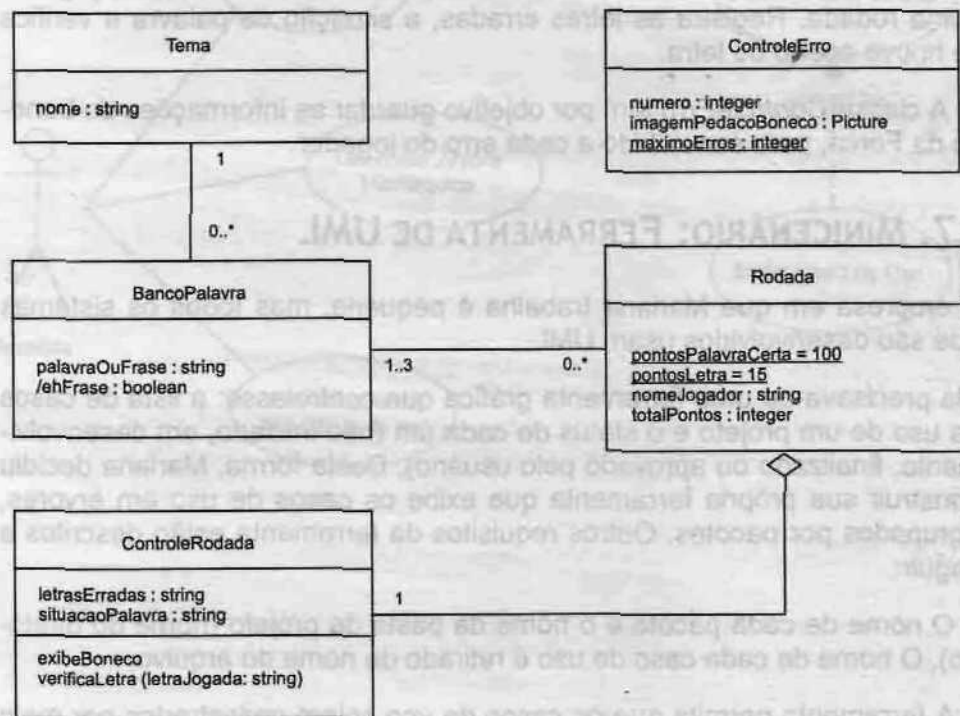
**EXERCÍCIO:**

1. Desenhe o diagrama de casos de uso.
2. Desenhe o diagrama de classes completo.

## RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



## RESOLUÇÃO (DIAGRAMA DE CLASSES):



**COMENTÁRIOS SOBRE A RESOLUÇÃO:**

- a) Ao analisar o problema, verificamos a necessidade de ter uma classe que controle o banco de palavras. Pode-se registrar uma palavra ou frase. O que determina se o registro pertence a uma frase é o atributo *ehFrase*. Entretanto, este atributo é derivado, ou seja, seu valor é resultado de um algoritmo que analisa o atributo *palavraOuFrase* e verifica se é uma palavra solta ou uma frase.
- b) Cada rodada tem no máximo três palavras ou uma frase. Assim, a multiplicidade da classe *Rodada* com o banco de palavras está definida como 1..3.
- c) A função da classe *Rodada* é controlar o jogador que está participando e os pontos que ele está marcando. Existem dois atributos de classe, que são responsáveis por definir a pontuação para acerto de palavra e acerto de letra. Além disso, a classe *Rodada* se relaciona com a classe *ControleRodada*, que é responsável por controlar todos os movimentos do jogador numa rodada. Registra as letras erradas, a situação da palavra e verifica se houve acerto de letra.
- d) A classe *ControleErro* tem por objetivo guardar as informações do boneco da Forca, para ser exibido a cada erro do jogador.

**27. MINICENÁRIO: FERRAMENTA DE UML**

A empresa em que Mariana trabalha é pequena, mas todos os sistemas que são desenvolvidos usam UML.

Ela precisava de uma ferramenta gráfica que controlasse: a lista de casos de uso de um projeto e o status de cada um (não iniciado, em desenvolvimento, finalizado ou aprovado pelo usuário). Desta forma, Mariana decidiu construir sua própria ferramenta que exibe os casos de uso em árvores, agrupados por pacotes. Outros requisitos da ferramenta estão descritos a seguir:

- O nome de cada pacote é o nome da pasta de projeto (nome do diretório). O nome de cada caso de uso é retirado do nome do arquivo;
- A ferramenta permite que os casos de uso sejam cadastrados por meio dela. Nesse caso, a ferramenta gera um arquivo baseado no template. É

possível realizar pesquisa de textos dentro de todos os casos de uso, renomeá-los e mudá-los de pacote;

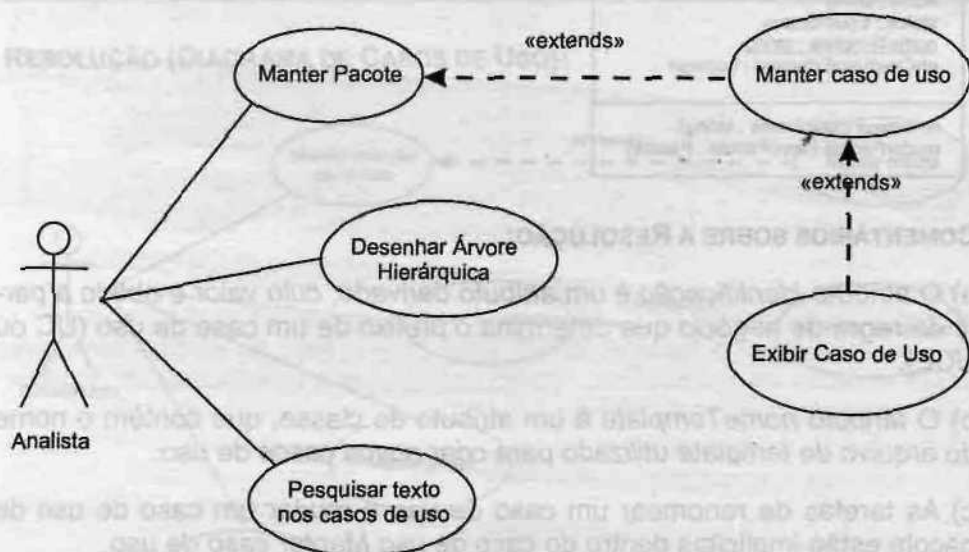
– Clicando no caso de uso, é possível abrir seu texto;

– Os casos de uso são identificados com o prefixo UC, seguido de uma numeração seqüencial controlada pela ferramenta. Os casos de uso de extensão recebem a identificação UCE.

#### EXERCÍCIO:

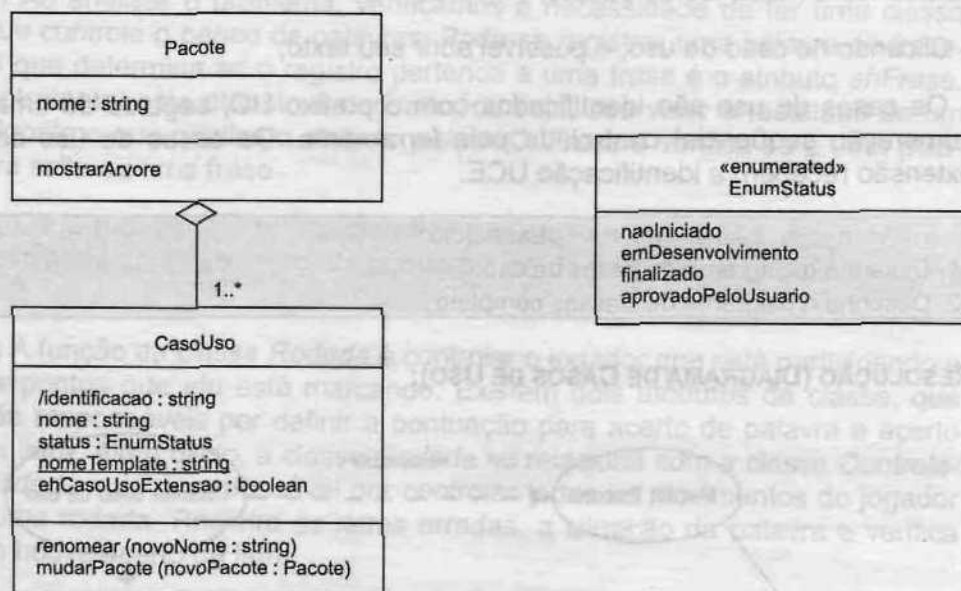
1. Desenhe o diagrama de casos de uso.
2. Desenhe o diagrama de classes completo.

#### RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):





## RESOLUÇÃO (DIAGRAMA DE CLASSES):



## COMENTÁRIOS SOBRE A RESOLUÇÃO:

- O atributo *identificação* é um atributo derivado, cujo valor é obtido a partir da regra de negócio que determina o prefixo de um caso de uso (UC ou UCE).
- O atributo *nomeTemplate* é um atributo de classe, que contém o nome do arquivo de template utilizado para criar novos casos de uso.
- As tarefas de renomear um caso de uso e mudar um caso de uso de pacote estão implícitas dentro do caso de uso *Manter caso de uso*.

## 28. MINICENÁRIO: CLUBE DA LEITURA

Gustavo tem uma coleção grande de revistas em quadrinhos. Por isso, resolveu emprestar para os amigos. Assim foi criado o Clube da Leitura. Mas para não perder nenhuma revista, seu pai lhe fez uma aplicação que cadastra as revistas e controla o empréstimo.

Para cada revista cadastram-se: o tipo da coleção (por exemplo: Cebolinha, Pato Donald, Batman etc.), o número da edição, o ano da revista e a caixa onde está guardada. Cada caixa tem uma cor, uma etiqueta e um número.

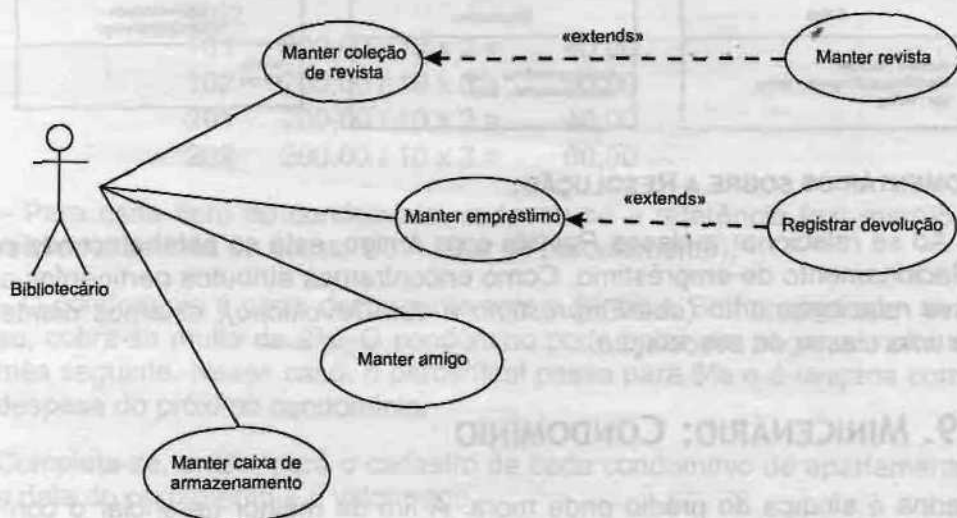
Para cada empréstimo cadastram-se: o amiguinho que pegou a revista, qual foi a revista, a data de empréstimo e a data de devolução. Cada criança só pode pegar uma revista por empréstimo.

O cadastro do amiguinho consiste de: o nome do amiguinho, o nome da mãe, o telefone e de onde é o amigo (do prédio ou da escola).

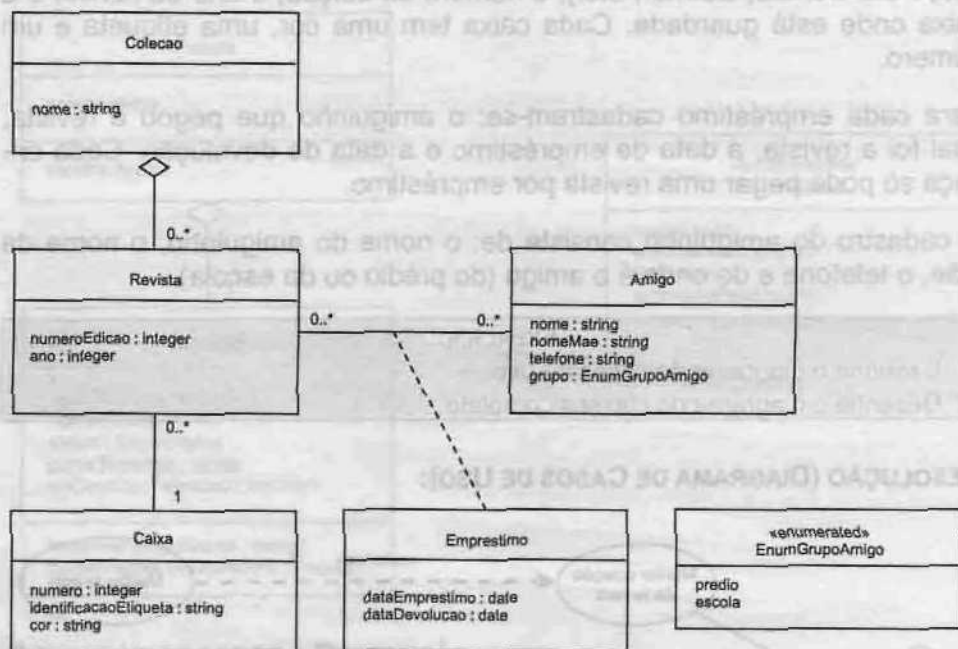
#### EXERCÍCIO:

1. Desenhe o diagrama de casos de uso.
2. Desenhe o diagrama de classes completo.

#### RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



## RESOLUÇÃO (DIAGRAMA DE CLASSES):



## COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) Ao se relacionar a classe *Revista* com *Amigo*, está se estabelecendo o relacionamento de empréstimo. Como encontramos atributos pertinentes a esse relacionamento (*dataEmprestimo* e *dataDevolucao*), estamos diante de uma classe de associação.

## 29. MINICENÁRIO: CONDOMÍNIO

Marina é síndica do prédio onde mora. A fim de melhor gerenciar o condomínio, ela encomendou uma aplicação a um amigo.

A aplicação deve atender aos seguintes requisitos:

- Para um apartamento, devem-se cadastrar: seu número de porta, a quantidade de quartos, o tipo de ocupação (proprietário, inquilino ou vazio), o nome do proprietário e o telefone do proprietário. Um proprietário pode ter mais de um apartamento no prédio;

– Mensalmente, devem ser lançadas as despesas para o condomínio do mês, além das despesas específicas de um apartamento ou um grupo de apartamentos, como por exemplo: a taxa de salão de festas;

– O condomínio é calculado por tipo de apartamento (de acordo com o número de quartos). Para chegar ao valor pago pelas despesas, têm-se os seguintes passos:

a) verificar o somatório de quartos de todo o prédio

Ex: apto 101 (2 quartos); 102 (3 quartos); 201 (2 quartos); 202 (3 quartos) = total de quartos = 10

b) dividir o valor da despesa (que pode ser fixo, como fundo de reserva; ou variável, como água) pelo somatório de quartos e multiplicar o resultado pelo número de quartos do apartamento.

Ex: água – 200,00

apto

101  $200,00 / 10 \times 2 =$  40,00

102  $200,00 / 10 \times 3 =$  60,00

201  $200,00 / 10 \times 2 =$  40,00

202  $200,00 / 10 \times 3 =$  60,00

– Para cada item do condomínio, cadastra-se a referência (ex: mar/05 – referente ao mês da conta; 1/3 – fator de parcelamento);

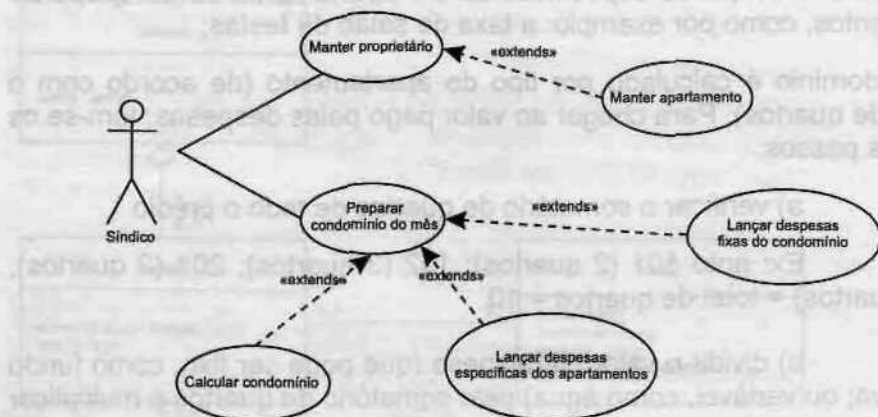
– O condomínio é pago diretamente com a Síndica. Se for pago com atraso, cobra-se multa de 2%. O condômino pode optar em pagar a multa no mês seguinte. Nesse caso, o percentual passa para 5% e é lançada como despesa do próximo condomínio.

Completa-se, então, para o cadastro de cada condomínio de apartamento, a data do pagamento e o valor pago.

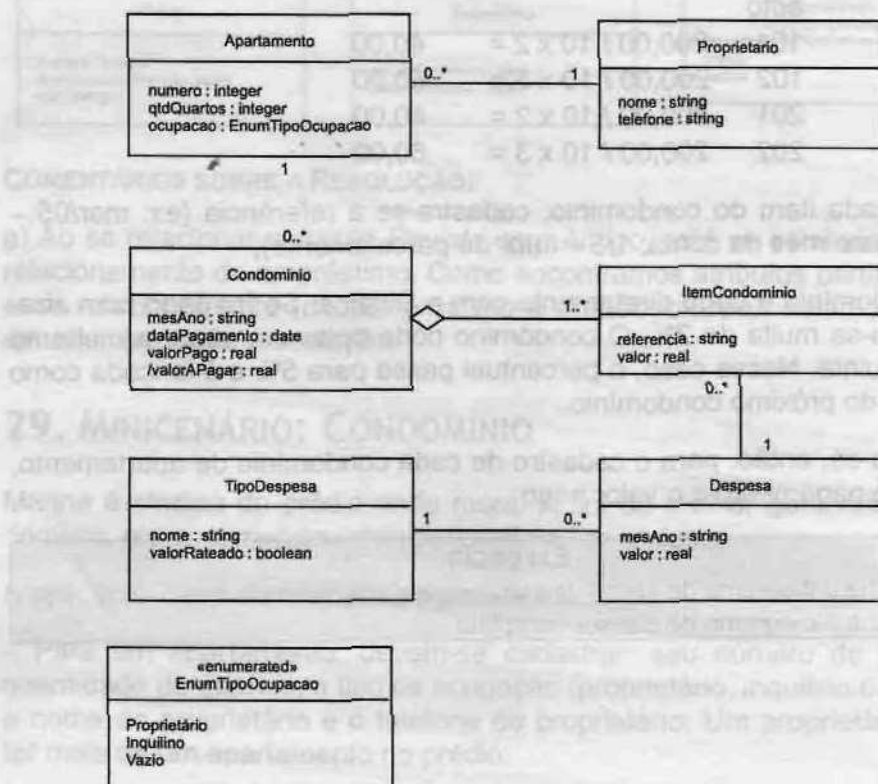
#### EXERCÍCIO:

1. Desenhe o diagrama de casos de uso.
2. Desenhe o diagrama de classes completo.

## RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



## RESOLUÇÃO (DIAGRAMA DE CLASSES):





**COMENTÁRIOS SOBRE A RESOLUÇÃO:**

a) O caso de uso *Preparar Condomínio do Mês* centraliza todas as tarefas necessárias para gerar os boletos dos condôminos. Primeiramente, o usuário lança as despesas fixas do condomínio de um determinado mês (por exemplo: água, luz etc). Depois, o usuário lança as despesas específicas dos apartamentos (por exemplo: multa do mês anterior, taxa do salão de festas etc). O caso de uso *Calcular Condomínio* é chamado para ratear todas as despesas, lançando para cada apartamento as contas a serem pagas.

Por exemplo: Suponha que:

- existam os seguintes apartamentos - 101 (2 quartos), 102 (3 quartos), 201 (2 quartos), 202 (3 quartos);
- totalizando todos os quartos do prédio, há 10 quartos;
- foram lançadas as despesas de água (100,00) e luz (200,00);
- o apartamento 101 tem uma taxa de salão de festas de R\$ 25,00.

O condomínio do apartamento 101 será:

Água (100,00 / 10 x 2 quartos) .....	R\$ 20,00
Luz (200,00 / 10 x 2 quartos) .....	R\$ 40,00
Taxa do Salão de Festas .....	R\$ 25,00
Total a pagar .....	R\$ 85,00

O condomínio do apartamento 102 será:

Água (100,00 / 10 x 3 quartos) .....	R\$ 30,00
Luz (200,00 / 10 x 3 quartos) .....	R\$ 60,00
Total a pagar .....	R\$ 90,00

b) A classe *Despesa* armazena as despesas do condomínio. O atributo *valorRateado* indica se esse tipo de despesa é de lançamento fixo (o mesmo valor para todos os condôminos, como acontece com o salão de festas; ou de valor variável, como acontece com a água).

**30. MINICENÁRIO: PEDIATRA**

Dra. Janete é Pediatra e tem três consultórios em bairros distintos, onde atende em horários diferentes. Ana, sua secretária, trabalha nos três consultórios. Para que a marcação de consultas seja centralizada, Ana tem que carregar as três agendas de um lado para outro.

Dra. Janete contratou um analista free-lancer para lhe desenvolver um sistema que controle a marcação de consultas e a ficha dos pacientes.

Para os pacientes, é preciso controlar: nome, endereço, telefones de contato, data de nascimento, data da primeira consulta, e-mail, se é particular ou conveniado a plano de saúde. No caso de ser conveniado, registrar qual é o plano de saúde. Para cada plano de saúde credenciado é preciso controlar o limite de consultas no mês, por paciente.

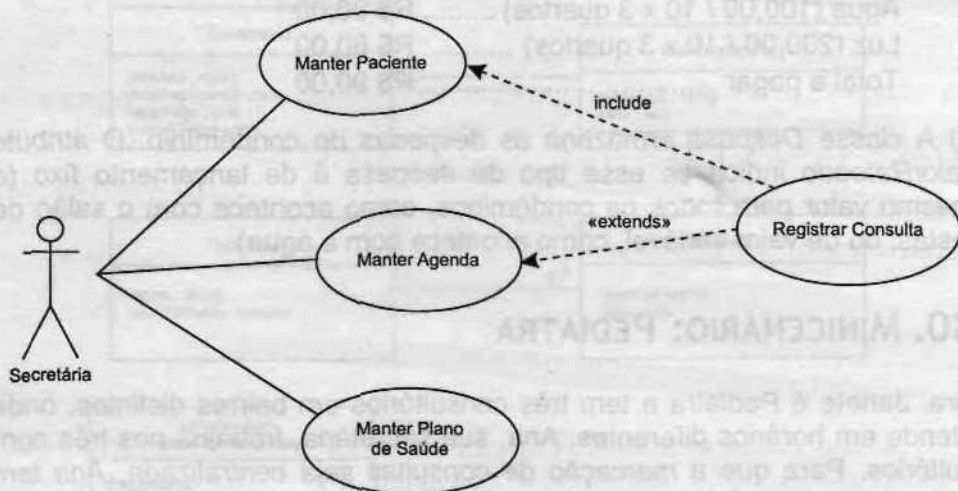
Dra. Janete atende no consultório da Ilha às segundas e sextas, de 9h às 18h. Na Tijuca, ela atende terça e quarta de 10h às 18h. Na quinta, ela atende em Bonsucesso, de 10h às 18h. O intervalo das consultas é de trinta minutos. O horário de almoço é sempre de 12h30 às 13h30. Só são permitidos três encaixes por dia. Para clientes novos, Ana anota o nome do cliente e o seu telefone. É preciso identificar se uma consulta é de revisão, como por exemplo para entrega de exames. Nesse caso, ela não é cobrada.

A marcação de consultas deve considerar uma data futura limite.

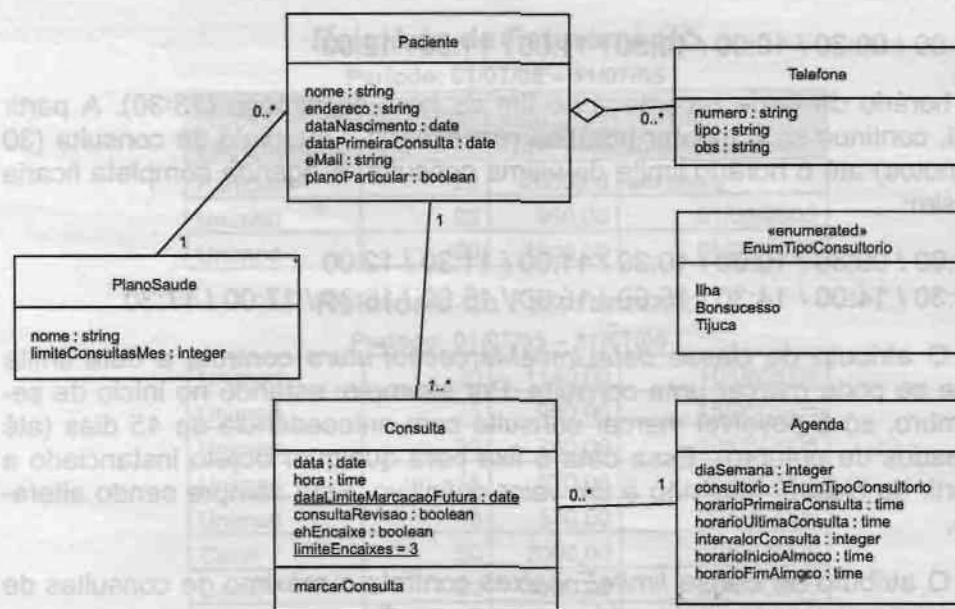
#### EXERCÍCIO:

1. Desenhe o diagrama de casos de uso.
2. Desenhe o diagrama de classes completo.

#### RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



## RESOLUÇÃO (DIAGRAMA DE CLASSES):



## COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) A classe *Agenda* contempla o padrão de agenda, de cada consultório, em cada dia da semana. Exemplo de um objeto instanciado a partir dessa classe. Objeto *AgendaIlhaSeg* (diaSemana = 2; consultório = Ilha; horarioPrimeiraConsulta = 09:00; horarioUltimaConsulta = 18:00; intervaloConsulta = 30; horarioInicioAlmoco = 12:30; horarioFimAlmoco = 13:30);

b) O caso de uso *Manter Agenda* é responsável por verificar e exibir a agenda de um consultório numa determinada data. A partir daí, para cada horário vago, é possível marcar uma consulta (*Extends Caso de Uso Registrar Consulta*).

Por exemplo, suponha que o usuário queira ver a agenda do dia 30/09/2005.

O sistema verificará que dia 30/09 é uma sexta (dia da semana = 6). Para o dia da semana 6, a agenda é do consultório da Ilha. Considerando os atributos desse objeto, o sistema monta a agenda do dia. Inicia com o horário da primeira consulta (09:00). Depois monta um horário a cada interva-

lo de consulta (30 minutos) até o limite do horário de almoço (12:30). Algo assim:

09:00 / 09:30 / 10:00 / 10:30 / 11:00 / 11:30 / 12:00

O horário da tarde recomeça no fim da hora de almoço (13:30). A partir daí, continua-se a montar horários respeitando o intervalo de consulta (30 minutos) até o horário limite da última consulta. A agenda completa ficaria assim:

09:00 / 09:30 / 10:00 / 10:30 / 11:00 / 11:30 / 12:00

13:30 / 14:00 / 14:30 / 15:00 / 15:30 / 16:00 / 16:30 / 17:00 / 17:30

c) O atributo de classe *dataLimiteMarcacaoFutura* controla a data limite que se pode marcar uma consulta. Por exemplo: estando no início de setembro, só é possível marcar consulta com antecedência de 45 dias (até meados de outubro). Essa data é fixa para qualquer objeto instanciado a partir da classe. Mas não é um valor definitivo. Está sempre sendo alterado.

d) O atributo de classe *limiteEncaixes* controla o máximo de consultas de encaixes que pode ser inserido num mesmo dia.

## 31. MINICENÁRIO: PEDIATRA (VARIAÇÃO A)

Na versão 1.1 do sistema, Dra. Janete deseja controlar os valores das consultas.

Ela deseja obter, no fim do mês, um relatório com o valor recebido pelas consultas particulares e o faturado pelos convênios.

Cada convênio paga um valor diferente por consulta e credita em prazos diferentes (alguns creditam em 45 dias, outros em 60 etc.) a partir da data de apresentação da consulta ao plano. A previsão de pagamento deve aparecer no relatório.

O valor da consulta particular é fixo, mas um ou outro paciente pode receber um desconto. Sendo assim, o valor efetivamente pago deve ser registrado.

Exemplo do Relatório de Faturamento:

**Relatório de Faturamento**

Período: 01/07/05 – 31/07/05

Tipo Pagamento	Qtd	Valor Total	Previsão Recebimento
Particular	20	2000,00	em caixa
Unimed	25	950,00	01/08/2005
Unimed	50	1900,00	05/08/2005

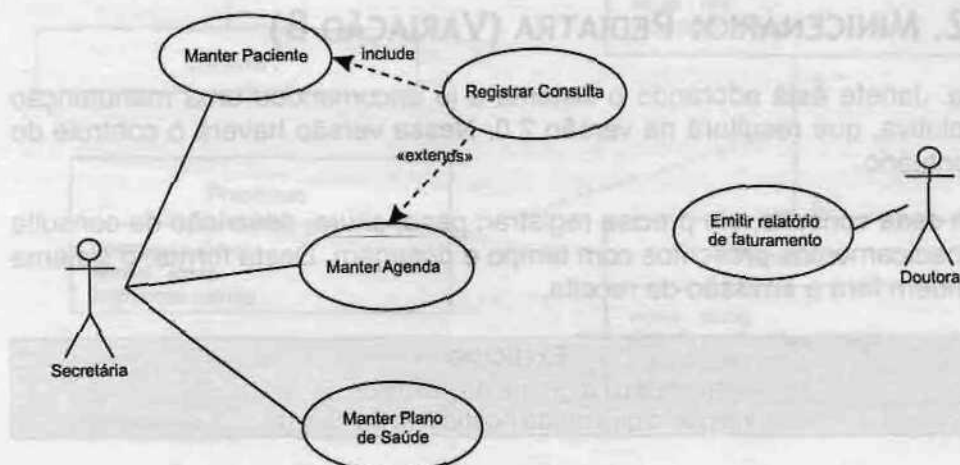
**Relatório de Faturamento**

Período: 01/07/05 – 31/07/05

Unimed	30	1140,00	10/08/2005
Unimed	25	950,00	15/08/2005
Unimed	20	760,00	20/08/2005
Unimed	15	570,00	25/08/2005
Unimed	15	570,00	30/08/2005
Cassi	50	2000,00	20/08/2005
Cassi	30	1200,00	30/08/2005
<b>Total</b>	<b>280</b>	<b>12040,00</b>	

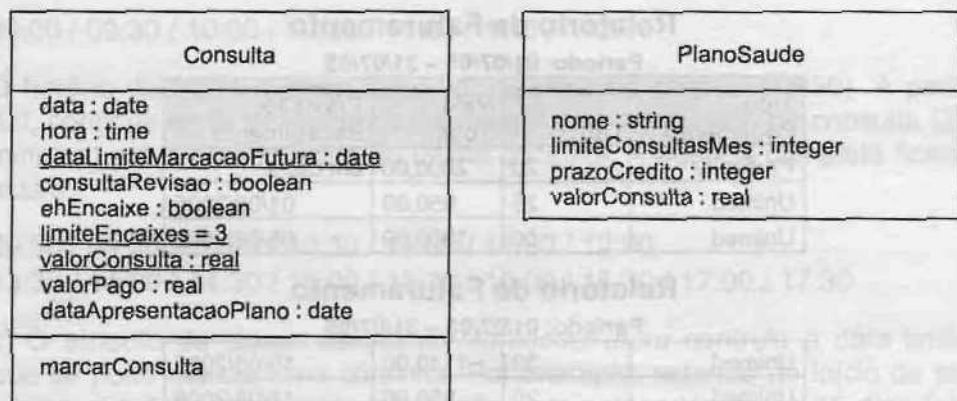
**EXERCÍCIO:**

Atualize o diagrama de casos de uso e indique o que muda no modelo de classes.

**RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):**



## RESOLUÇÃO (DIAGRAMA DE CLASSES – SOMENTE AS CLASSES ALTERADAS):



## COMENTÁRIOS SOBRE A RESOLUÇÃO:

- a) O atributo de classe *valorConsulta* indica o valor da consulta para qualquer paciente que seja particular (não esteja vinculado a um plano de saúde).
- b) O atributo *valorPago* registrará efetivamente o valor pago por uma consulta, considerando um eventual desconto concedido (se o cliente for particular).
- c) O atributo *dataApresentacaoPlano* registrará a data em que a consulta foi apresentada ao plano, a fim de que se possa calcular o prazo estimado de recebimento do reembolso, de acordo com o prazo de cada plano de saúde.

## 32. MINICENÁRIO: PEDIATRA (VARIAÇÃO B)

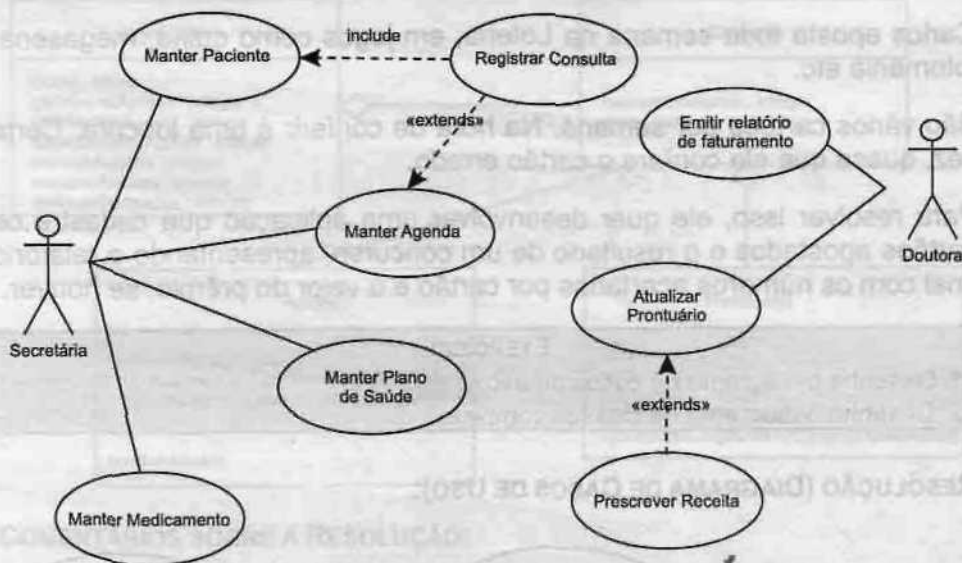
Dra. Janete está adorando o sistema e já encomendou uma manutenção evolutiva, que resultará na versão 2.0. Nessa versão haverá o controle do prontuário.

Em cada consulta, ela precisa registrar: peso, altura, descrição da consulta e medicamentos prescritos com tempo e dosagem. Desta forma, o sistema também fará a emissão da receita.

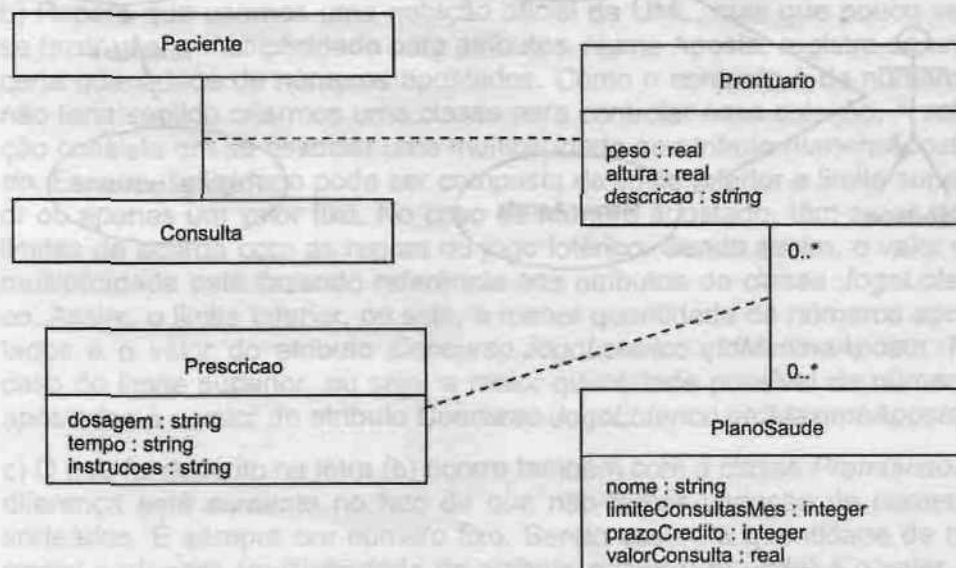
## EXERCÍCIO:

Atualize o diagrama de casos de uso e indique o que muda no modelo de classes.

## RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



## RESOLUÇÃO (DIAGRAMA DE CLASSES – SOMENTE AS CLASSES ALTERADAS):



### 33. MINICENÁRIO: APOSTA LOTÉRICA

Carlos aposta toda semana na Loteria, em jogos como quina, megasena, lotomania etc.

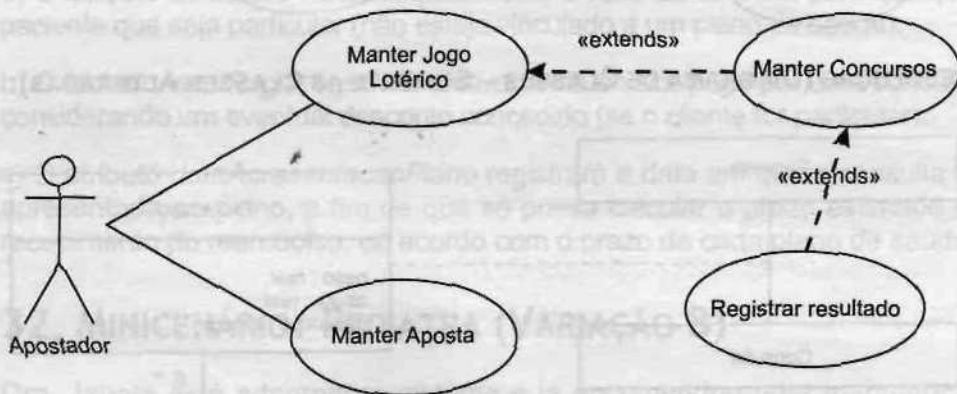
São vários cartões por semana. Na hora de conferir é uma loucura. Certa vez, quase que ele confere o cartão errado.

Para resolver isso, ele quer desenvolver uma aplicação que cadastre os cartões apostados e o resultado de um concurso, apresentando o relatório final com os números acertados por cartão e o valor do prêmio, se houver.

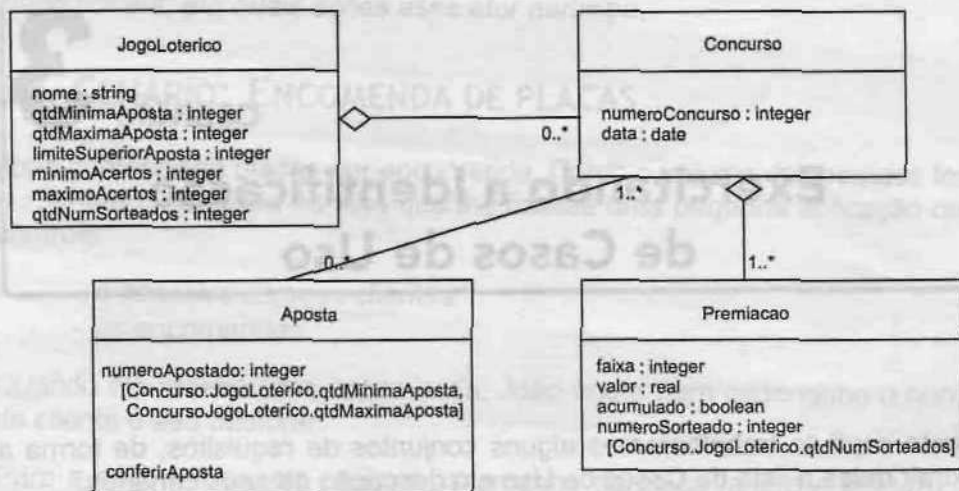
#### EXERCÍCIO:

1. Desenhe o diagrama de casos de uso.
2. Desenhe o diagrama de classes completo.

#### RESOLUÇÃO (DIAGRAMA DE CASOS DE USO):



## RESOLUÇÃO (DIAGRAMA DE CLASSES):



## COMENTÁRIOS SOBRE A RESOLUÇÃO:

a) Como o único usuário dessa aplicação é o Carlos, o ator foi identificado como Apostador, que é o papel assumido por ele na aplicação.

b) Repare que usamos uma notação oficial da UML, mas que pouco vejo se fazer uso: a multiplicidade para atributos. Numa Aposta, registra-se uma certa quantidade de números apostados. Como o conjunto é de números, não teria sentido criarmos uma classe para controlar essa coleção. A solução consiste em se associar uma multiplicidade ao atributo *numeroApostado*. Essa multiplicidade pode ser composta de limite inferior e limite superior ou apenas um valor fixo. No caso do número apostado, têm-se os dois limites de acordo com as regras do jogo lotérico. Sendo assim, o valor da multiplicidade está fazendo referência aos atributos da classe *JogoLoterico*. Assim, o limite inferior, ou seja, a menor quantidade de números apostados é o valor do atributo *Concurso.JogoLoterico.qtdMinimaAposta*. No caso do limite superior, ou seja, a maior quantidade possível de números apostados é o valor do atributo *Concurso.JogoLoterico.qtdMaximaAposta*.

c) O que foi descrito na letra (b) ocorre também com a classe *Premiacao*. A diferença está somente no fato de que não temos variação de números sorteados. É sempre um número fixo. Sendo assim, a quantidade de números sorteados (multiplicidade do atributo *numeroSorteado*) é o valor do atributo *Concurso.JogoLoterico.qtdNumSorteados*.

## Exercitando a Identificação de Casos de Uso

Neste capítulo trabalharemos alguns conjuntos de requisitos, de forma a extrair deles a lista de Casos de Uso e a descrição de seus cenários.

Devemos nos lembrar de que não existe na documentação oficial da UML uma forma predefinida de se escrever um caso de uso. Entretanto, existem boas práticas que devem ser seguidas, das quais podemos citar:

- O texto do caso de uso deve ser livre da forma como será implementado, exceto se for um caso de uso de projeto (exemplo: caso de uso Manter cópia de backup). Sendo assim, não diga que o usuário irá selecionar uma determinada informação de uma listbox ou combobox; ou que clicará num botão. Diga, apenas, que será selecionado um valor de uma lista; ou que o usuário escolherá uma determinada opção, que pode ser implementada por meio de um botão, de um menu, ou de dezenas de outras formas diferentes;
- Após escrever sua primeira versão de caso de uso, volte e analise-o. Verifique se alguma parte de seu caso de uso é reaproveitável em outro caso de uso. Se existir, devemos separá-la em outro caso de uso e estabelecer um relacionamento de *Include*. Verifique se algum cenário alternativo do seu caso de uso está extenso demais. Se existir, devemos separá-lo em outro caso de uso e estabelecer um relacionamento de *Extends*;
- Um caso de uso pode se transformar numa rotina de um sistema ou apenas em um método de uma classe, dependendo de sua complexidade;
- Para começar a escrever a lista de casos de uso, você pode iniciar pela lista de atores que interagem com a aplicação. Para cada ator, você pode



questionar: qual a influência do ator na aplicação ou o quanto ele é influenciado por ela; em quais ações esse ator participa.

## 34. CENÁRIO: ENCOMENDA DE PLACAS

João confecciona placas por encomenda. Como o volume dos pedidos tem aumentado, ele pediu ao filho que lhe fizesse uma pequena aplicação que controle:

- o cadastro de seus clientes
- as encomendas

Quando ele recebe uma encomenda, João anota num caderninho o nome do cliente e seu telefone.

Para a encomenda, ele registra: o tamanho da placa (altura e largura), a frase a ser escrita, cor da placa (branca ou cinza), cor da frase (azul, vermelho, amarelo, preto ou verde), data de entrega, valor do serviço e valor do sinal.

A aplicação deve obrigar que o valor do sinal seja de, no mínimo, 50%.

Para calcular o valor da placa, as seguintes fórmulas são usadas:

$$\begin{aligned} \text{área} &= \text{altura} \times \text{largura} \\ \text{custo\_material} &= \text{área} \times \text{R\$ } 147,30 \\ \text{custo\_desenho} &= \text{número\_letras} \times \text{R\$ } 0,32 \\ \text{valor\_placa} &= \text{custo\_material} + \text{custo\_desenho} \end{aligned}$$

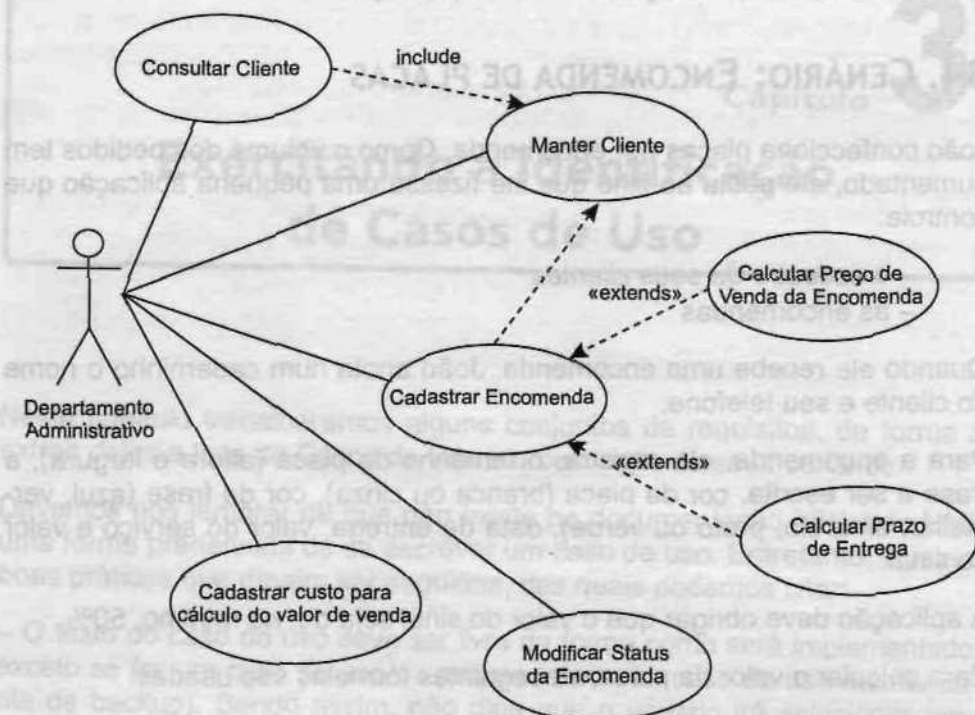
Para calcular o prazo de entrega, considera-se que ele só consegue produzir seis placas por dia.

João deseja que o sistema controle os pedidos, calcule o preço final das peças e o prazo de entrega. Para cada encomenda cadastrada, deve ser emitido um recibo em duas vias (cliente e empresa), contendo todos os dados da encomenda e do pagamento.

### EXERCÍCIO:

A partir do cenário descrito, desenhe o diagrama de casos de uso e escreva seus cenários. Considere que só existe um ator, que é o Sr. João, dono de sua micro-empresa, que pode ser identificado como "Diretor da empresa".

## RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):



CONSULTAR CLIENTE	
Descrição:	Este caso de uso tem por objetivo apresentar os clientes cadastrados e habilitar a inclusão, alteração ou exclusão de clientes.
Ator:	Diretor da empresa

**Cenário Principal:**

- O sistema prepara uma lista de todos os clientes cadastrados.
- O sistema oferece ao usuário:
  - selecionar um cliente, para alterar seu cadastro;
  - localizar um cliente ou conjunto de clientes por meio de pesquisa;
  - selecionar a opção de "inserir cliente".
- Pesquisa de Cliente
  - Para localizar um cliente, o usuário deve inserir um trecho de nome e/ou um trecho de telefone. O sistema fará a busca parcial.
  - O sistema exibe a lista de clientes que satisfaça o critério, exibindo para cada um:

3.2.1. código de identificação

3.2.2. nome do cliente

3.2.3. telefone

4. Inserção de Cliente

4.1. [Inclui Caso de Uso Manter Cliente]

5. Seleção de Cliente

5.1. Após selecionar um cliente, o sistema habilita as opções de "alterar cliente" e "excluir cliente".

5.2. Se o usuário selecionar uma dessas opções, o sistema aciona o cadastro de cliente. [Inclui Caso de Uso Manter Cliente]

MANTER CLIENTE	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a inclusão, alteração ou exclusão de dados ligados ao cadastro de clientes.
<b>Ator:</b>	Diretor da empresa

**Pré-condição:**

Receber a identificação do tipo de operação e os dados cadastrais do cliente, no caso de alteração ou exclusão.

**Cenário Principal:**

1. Manutenção do Cadastro
  - 1.1. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 1.2. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 1.3. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 1.3.1. No caso de exclusão, o sistema solicita a confirmação.
2. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 2.1. nome do cliente
  - 2.2. telefone de contato
3. O usuário confirma a operação realizada.
4. O sistema atualiza os dados cadastrais do cliente.
  - 4.1. No caso de inclusão, o sistema gera automaticamente um código de identificação.

**Cenário Alternativo:**

**– Exclusão não permitida**

Não é possível excluir um cliente que esteja associado a uma encomenda.

CADASTRAR CUSTO PARA CÁLCULO DO VALOR DE VENDA	
<b>Descrição:</b>	Este caso de uso tem por objetivo cadastrar os valores fixos de custo, utilizados no cálculo do valor de venda das placas.
<b>Ator:</b>	Diretor da empresa

**Cenário Principal:**

1. O sistema busca os valores cadastrados para:
  - 1.1. valor fixo do material
  - 1.2. valor fixo da letra
2. O usuário altera:
  - 2.1. valor fixo do material
  - 2.2. valor fixo da letra
3. O usuário confirma o cadastramento.
4. O sistema atualiza os valores no cadastro.

**Cenário Alternativo:****- Valores Inexistentes no cadastro**

Se não existir valor cadastrado para "valor fixo do material" e/ou "valor fixo da letra", o sistema apresenta os campos em branco.

**- Valores inconsistentes**

Não pode ser cadastrado valor negativo para "valor fixo do material" e "valor fixo da letra".

CADASTRAR ENCOMENDA	
<b>Descrição:</b>	Este caso de uso tem por objetivo cadastrar encomendas de placas.
<b>Ator:</b>	Diretor da empresa

**Cenário Principal:**

1. O sistema busca e exibe a lista dos clientes cadastrados, em ordem alfabética de nome.
2. O usuário seleciona um nome de cliente da lista preexistente.
3. O sistema exibe o telefone do cliente.
4. O usuário informa os dados da encomenda:
  - 4.1. altura da placa
  - 4.2. largura da placa
  - 4.3. frase para impressão
  - 4.4. cor da placa, selecionada dentre as opções: cinza ou branca.
  - 4.5. cor da frase, selecionada dentre as opções: azul, vermelho, amarelo, preto ou verde.
  - 4.6. o sistema associa a data da encomenda como sendo a data atual.
5. O sistema calcula e exibe a data prevista de entrega do pedido.
  - 5.1. [Extends Caso de Uso Calcular Prazo de Entrega]
6. O sistema calcula e exibe o valor a pagar pela encomenda.

- 6.1. [Extends Caso de Uso Calcular Preço de Venda da Encomenda]
7. O usuário informa o valor do sinal.
8. O usuário confirma a encomenda.
9. O sistema gera automaticamente um número de encomenda.
10. O sistema emite um recibo, em duas vias, com os seguintes dados:
  - 10.1. nome do cliente, telefone de contato, data da encomenda, frase a ser impressa na placa, tamanho da placa (altura e largura), cor da placa, cor da frase, valor da encomenda, data prevista de entrega e valor do sinal.
11. O sistema atualiza os valores no cadastro, lançando o status da encomenda como "aberto".

### **Cenários Alternativos:**

#### **– Cliente não cadastrado**

Se for um cliente novo, o usuário seleciona a opção de "cadastrar novo cliente". [Include Manter Cliente].

#### **– Valor do sinal insuficiente**

O sistema não deve aceitar um valor de sinal inferior a 50% do valor de venda da peça. No caso do sinal ser inferior, o sistema deve exibir uma mensagem de erro, incluindo na mensagem o valor mínimo permitido.

CALCULAR PREÇO DE VENDA DA ENCOMENDA	
<b>Descrição:</b>	Este caso de uso tem por objetivo calcular o preço de venda de uma placa, baseado nas informações recebidas para o cálculo.
<b>Ator:</b>	Diretor da empresa

### **Pré-condição:**

Receber as seguintes informações: altura da placa, largura da placa, frase para impressão.

### **Cenário Principal:**

1. O sistema busca os valores cadastrados para "valor fixo do material" e "valor fixo da letra".
2. O sistema calcula o preço de venda da encomenda, considerando as seguintes fórmulas:
 

área	= "altura da placa" x "largura da placa"
custo_material	= área x "valor fixo do material"
número_letras	= quantidade de letras da "frase para impressão".
custo_desenho	= número_letras x "valor fixo da letra".
valor_placa	= custo_material + custo_desenho
3. O sistema retorna o "valor\_placa".

### **Cenário Alternativo:**

#### **– Valores nulos**

Se qualquer um dos valores de pré-condição estiver nulo, o sistema não efetuará o cálculo. Será exibida uma mensagem de erro e o valor de retorno será zero.



**- Valores fixos inexistentes**

Se não houver valor válido para "valor fixo do material" e/ou para "valor fixo da letra", o sistema deve exibir uma mensagem de erro, informando que faltam dados de referência para cálculo da encomenda.

CALCULAR PRAZO DE ENTREGA	
<b>Descrição:</b>	Este caso de uso tem por objetivo calcular o prazo de entrega de uma determinada placa, de acordo com as encomendas que estão com o status = "aberto".
<b>Ator:</b>	Diretor da empresa

**Cenário Principal:**

1. O sistema busca o total de encomendas com status = "aberto", agrupados por data, excluindo-se o dia atual.
2. O sistema verifica a primeira data disponível da lista, onde o número de encomendas seja inferior a seis.
3. O sistema retorna a data disponível no item 2, como a data prevista de entrega.

**Cenário Alternativo:****- Nenhuma data disponível**

Se não houver nenhuma data disponível dentro da lista recebida, o sistema deve calcular a data prevista de entrega como sendo a maior data da lista acrescida de um dia. Se a data prevista cair num sábado ou domingo, deve ser incrementada até a segunda-feira.

**- Nenhuma encomenda cadastrada**

Se não houver nenhuma encomenda cadastrada, o sistema deve calcular a data prevista de entrega como sendo a data da encomenda acrescida de um dia. Se a data prevista cair num sábado ou domingo, deve ser incrementada até a segunda-feira.

MODIFICAR STATUS DA ENCOMENDA	
<b>Descrição:</b>	Este caso de uso tem por objetivo modificar o status de uma encomenda durante a sua execução.
<b>Ator:</b>	Diretor da empresa

**Cenário Principal:**

1. O usuário informa o número da encomenda.
2. O sistema busca a encomenda e exibe:
  - 2.1. o nome do cliente;
  - 2.2. o telefone;
  - 2.3. a data da encomenda;
  - 2.4. a data de entrega;

- 2.5. o valor do pedido;
- 2.6. o valor do sinal;
- 2.7. o status atual da encomenda.
- 3. O usuário modifica o status da encomenda para um dos seguintes valores: "Pronto", "Cancelado" ou "Fechado".
- 4. O usuário confirma a alteração do status.
- 5. O sistema atualiza o cadastro com o novo status.

#### **Cenários Alternativos:**

##### **– Encomenda Inexistente**

Se o número da encomenda não existir, exibir ao usuário uma mensagem de erro, e abrir uma lista de encomendas com status diferente de "Fechado" e "Cancelado" para seleção.

##### **– Alteração não permitida**

Não é possível alterar o status de encomendas que estejam com o status "Cancelado" ou "Fechado".

##### **– Validação do Status**

O status = "Aberto" só pode ser alterado para "Pronto" ou "Cancelado".

O status = "Pronto" só pode ser alterado para "Cancelado" ou "Fechado".

## **35. CENÁRIO: RÁDIO TÁXI MAR & SOL**

A empresa de Rádio Táxi Mar & Sol precisa de uma aplicação que controle:

- o cadastro de seus clientes
- o cadastro dos cooperados
- o cadastro das corridas programadas

Para cada cliente são cadastrados os seguintes dados: código (que deve ser gerado pelo sistema), nome, endereço completo (logradouro, número, complemento, bairro, município, estado) e dois telefones de contato.

O cliente pode se cadastrar apenas com o nome para agilizar o processo. Quando fizer sua primeira chamada por telefone, seus dados serão atualizados.

Para o cooperado (taxista) cadastram-se: nome, CPF, número da carteira de motorista, categoria, data de validade da carteira, número do táxi na cooperativa (conhecido como número de VR), número da placa, modelo do veículo, fabricante, cor do veículo, endereço residencial completo, telefone residencial e celular e data de entrada na Cooperativa. Quando o cooperado se desliga, deve ser cadastrada a data de desligamento.

Quando o cliente solicitar uma corrida programada (pedidos com antecedência maior do que meia hora), cadastra-se no controle de corridas: o endereço de saída do carro, o bairro de destino, a data e hora de saída, telefone de contato (se local de saída diferente do cadastro). Se o cliente não for cadastrado, seu cadastro deve ser feito no momento da solicitação do carro. O status dessa corrida deve ser definido como: "aguardando VR".

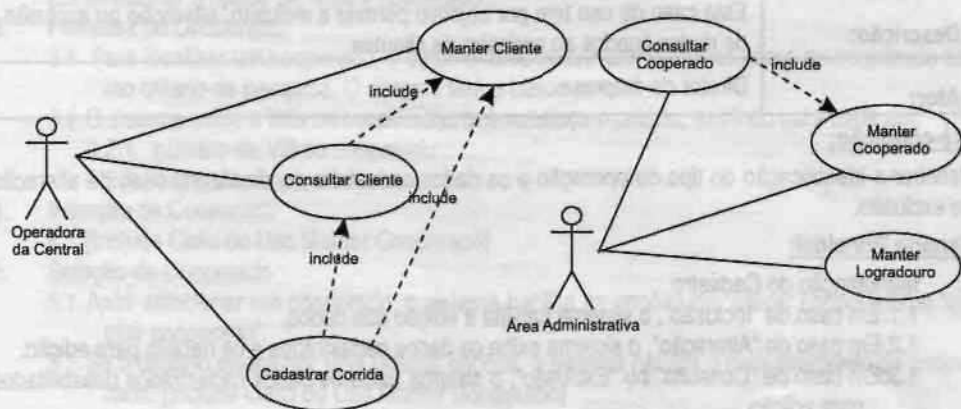
Uma hora antes da corrida programada, a operadora questiona, pelo rádio, aos cooperados que estejam em trânsito, qual deseja pegar a corrida programada. Deve ser cadastrado na aplicação o número da VR do taxista que se candidatou à corrida. Meia hora antes do horário, o cliente deve ser avisado a respeito do número da VR. Antes de avisar ao cliente, o status deve ser assinalado como: "aguardando aviso". Após o aviso, o status muda para "aviso efetuado". Após ser atendido, o status deve ser alterado para: "tripulado". Em qualquer momento a corrida pode ser cancelada pelo passageiro.

Se for uma solicitação de carro imediato, a operadora deve retornar à tela, informando o status dentre as opções: "aguardando aviso", "aviso efetuado", "cancelado pelo passageiro" ou "cancelado pela cooperativa por falta de carro". Se um logradouro não estiver na lista, a solicitação não será atendida.

Quando o cliente for atendido, o status deve ser alterado para: "tripulado".

#### EXERCÍCIO:

A partir do cenário descrito, desenhe o diagrama de casos de uso e escreva os seus cenários. Considere que: o cadastramento das corridas e dos clientes é feito pela Operadora da Central; o cadastramento dos cooperados é feito por qualquer funcionário da Área Administrativa; e o controle mensal de pagamentos de diárias é feito pela Área Financeira.

**RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):**

CONSULTAR CLIENTE	
<b>Descrição:</b>	Este caso de uso tem por objetivo apresentar os clientes cadastrados e habilitar a inclusão, alteração ou exclusão de clientes.
<b>Ator:</b>	Operadora

**Cenário Principal:**

1. O sistema prepara uma lista de todos os clientes cadastrados.
2. O sistema oferece ao usuário:
  - 2.1. selecionar um cliente, para alterar seu cadastro;
  - 2.2. localizar um cliente ou conjunto de clientes por meio de pesquisa;
  - 2.3. selecionar a opção de "inserir cliente".
3. Pesquisa de Cliente
  - 3.1. Para localizar um cliente, o usuário deve inserir um trecho do nome do cliente como critério de pesquisa. O sistema fará a busca parcial.
  - 3.2. O sistema exibe a lista de clientes que satisfaça o critério, exibindo para cada um:
    - 3.2.1. código de identificação
    - 3.2.2. nome do cliente
    - 3.2.3. telefone
4. Inserção de Cliente
  - 4.1. [Include Caso de Uso Manter Cliente]
5. Seleção de Cliente
  - 5.1. Após selecionar um cliente, o sistema habilita as opções de "alterar cliente" e "excluir cliente".
  - 5.2. Se o usuário selecionar uma dessas opções, o sistema aciona o cadastro de cliente. [Include Caso de Uso Manter Cliente]

MANTER CLIENTE	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a inclusão, alteração ou exclusão de dados ligados ao cadastro de clientes.
<b>Ator:</b>	Diretor da empresa

**Pré-condição:**

Receber a identificação do tipo de operação e os dados cadastrais do cliente, no caso de alteração ou exclusão.

**Cenário Principal:**

1. Manutenção do Cadastro
  - 1.1. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 1.2. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 1.3. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 1.3.1. No caso de exclusão, o sistema solicita a confirmação.
2. O sistema prepara uma lista de todos os logradouros atendidos pela Cooperativa.
3. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 3.1. nome do cliente
  - 3.2. logradouro, selecionado de uma lista preexistente. O sistema exibe o bairro, a cidade e o estado.
  - 3.3. número e complemento do endereço.
  - 3.4. dois telefones de contato, informando para cada um:
    - 3.4.1. prefixo
    - 3.4.2. número
    - 3.4.3. tipo, selecionado entre as opções: residencial, comercial, celular ou recado.
4. O usuário confirma a operação realizada.
5. O sistema atualiza os dados cadastrais do cliente.
  - 5.1. No caso de inclusão, o sistema gera automaticamente um código de identificação.

**Cenário Alternativo:****- Exclusão não permitida**

Não é possível excluir um cliente que esteja associado a uma corrida cadastrada.

CONSULTAR COOPERADO	
<b>Descrição:</b>	Este caso de uso tem por objetivo apresentar os cooperados (taxistas) cadastrados e habilitar a inclusão, alteração ou exclusão de cooperados.
<b>Ator:</b>	Área Administrativa

**Cenário Principal:**

1. O sistema prepara uma lista de todos os cooperados cadastrados.
2. O sistema oferece ao usuário:



- 2.1. selecionar um cooperado, para alterar seu cadastro;
- 2.2. localizar um cooperado ou conjunto de cooperados por meio de pesquisa;
- 2.3. selecionar a opção de "inserir cooperado".
- 3. Pesquisa de Cooperado
  - 3.1. Para localizar um cooperado, o usuário deve inserir um trecho do nome do cooperado como critério de pesquisa. O sistema fará a busca parcial.
  - 3.2. O sistema exibe a lista de cooperados que satisfaça o critério, exibindo para cada um:
    - 3.2.1. número da VR do cooperado
    - 3.2.2. nome do cooperado
- 4. Inserção de Cooperado
  - 4.1. [Include Caso de Uso Manter Cooperado]
- 5. Seleção de Cooperado
  - 5.1. Após selecionar um cooperado, o sistema habilita as opções de "alterar cooperado" e "excluir cooperado".
  - 5.2. Se o usuário selecionar uma dessas opções, o sistema habilita o cadastramento de cooperado. [Include Caso de Uso Manter Cooperado]

MANTER COOPERADO	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a inclusão ou alteração de dados ligados ao cadastro de cooperados.
<b>Ator:</b>	Área Administrativa

#### Pré-condição:

Receber a identificação do tipo de operação e os dados cadastrais do cooperado, no caso de alteração ou exclusão.

#### Cenário Principal:

- 1. Manutenção do Cadastro
  - 1.1. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 1.2. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 1.3. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 1.3.1. No caso de exclusão, o sistema solicita a confirmação.
- 2. O sistema prepara uma lista de logradouros cadastrados.
- 3. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 3.1. CPF
  - 3.2. nome do cooperado
  - 3.3. dados da carteira de motorista: número, categoria e data de validade
  - 3.4. dados do veículo: número da VR (identificação do veículo na cooperativa), número da placa, modelo, fabricante
  - 3.5. endereço completo, considerando que o logradouro é selecionado de uma lista preexistente. Ao selecionar o logradouro, o sistema exibe o bairro, a cidade e o estado. O usuário completa o cadastro do endereço com o número e complemento.

- 3.6. telefones residencial e celular
- 3.7. data de entrada na Cooperativa
- 3.8. data de saída da Cooperativa (somente para alteração)
- 4. O usuário confirma a operação realizada.
- 5. O sistema atualiza os dados cadastrais do cooperado.

MANTER LOGRADOURO	
<b>Descrição:</b>	Este caso de uso tem por objetivo apresentar os logradouros atendidos pela cooperativa e habilitar a inclusão, alteração ou exclusão de logradouros.
<b>Atores:</b>	Área Administrativa

**Cenário Principal:**

1. O sistema prepara uma lista de todos os logradouros cadastrados.
2. O sistema oferece ao usuário:
  - 2.1. selecionar um logradouro, para alterar seu cadastro;
  - 2.2. localizar um logradouro ou conjunto de logradouros por meio de pesquisa;
  - 2.3. selecionar a opção de "inserir cliente".
3. Pesquisa de Logradouro
  - 3.1. Para localizar um logradouro, o usuário deve inserir um trecho do nome e/ou do bairro como critério de pesquisa. O sistema fará a busca parcial.
  - 3.2. O sistema exibe a lista de logradouros que satisfaça o critério, exibindo para cada um:
    - 3.2.1. nome do logradouro
    - 3.2.2. bairro
4. Manutenção do Cadastro
  - 4.1. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 4.2. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 4.3. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 4.3.1. No caso de exclusão, o sistema solicita a confirmação.
5. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 5.1. nome do logradouro
  - 5.2. bairro
  - 5.3. cidade
  - 5.4. UF
6. O usuário confirma a operação realizada.
7. O sistema atualiza os dados cadastrais do logradouro.

**Cenário Alternativo:****- Exclusão não permitida**

Não é possível excluir um logradouro que esteja associado a uma corrida, cooperado ou cliente.

CADASTRAR CORRIDA	
<b>Descrição:</b>	Este caso de uso tem por objetivo cadastrar a solicitação dos clientes de corridas programadas (que são pedidas com antecedência maior do que meia hora) ou imediatas.
<b>Ator:</b>	Operadora da central

**Cenário Principal:**

1. O usuário informa o código de identificação do cliente.
  - 1.1. O sistema pesquisa o código e exibe: o nome do cliente, seu endereço e telefones.
  - 1.2. O sistema exibe a lista de corridas programadas.
  - 1.3. O sistema oferece ao usuário:
    - 1.3.1. selecionar uma corrida, para alterar seu cadastro;
    - 1.3.2. alterar o cadastro do cliente;
    - 1.3.3. selecionar a opção de "inserir corrida".
2. Manutenção do Cadastro
  - 2.1. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 2.2. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 2.3. Em caso de "Consulta", o sistema exibe os dados cadastrados desabilitados para edição.
3. Alteração do Cadastro de Cliente:
  - 3.1. [Inclui Caso de Uso Manter Cliente]
4. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 4.1. Se o endereço de origem da corrida é o mesmo endereço do cliente.
  - 4.2. Se não for o mesmo endereço:
    - 4.2.1. o sistema prepara uma lista dos logradouros atendidos pela Cooperativa.
    - 4.2.2. o usuário informa o logradouro de origem, selecionando de uma lista preexistente.
    - 4.2.3. o usuário informa o número e o complemento do logradouro, além de um telefone de contato.
  - 4.3. O usuário informa o bairro de destino da corrida.
  - 4.4. O usuário informa a data e a hora para a qual a corrida deve ser programada; ou se é uma corrida imediata.
5. Somente para "Alteração" de corrida:
  - 5.1. o sistema prepara uma lista de veículos cadastrados.
  - 5.2. o usuário informa o número da VR escolhida para a corrida.
  - 5.3. se a VR tiver sido informada, o usuário poderá alterar o status da corrida para uma das seguintes opções:
    - 5.3.1. "aguardando aviso";
    - 5.3.2. "aviso efetuado";
    - 5.3.3. "tripulado".
  - 5.4. Em qualquer situação, o usuário poderá alterar o status da corrida para:
    - 5.4.1. "cancelado pelo passageiro";
  - 5.5. No caso de corrida imediata e não tendo sido informada a VR, o usuário poderá alterar o status da corrida para:
    - 5.5.1. "cancelado pela cooperativa por falta de carro".



6. O usuário confirma a operação realizada.
7. O sistema atualiza os dados cadastrais da corrida.
  - 7.1. Se for inclusão, o sistema atualiza o status automaticamente com o valor "aguardando VR".

#### **Cenário Alternativo:**

##### **– Código desconhecido**

Se o usuário não possuir o código do cliente, ele poderá pesquisá-lo a partir do nome do cliente [Include Caso de Uso Consultar Cliente].

##### **– Cliente não cadastrado**

Se o cliente não for cadastrado, o usuário poderá efetuar o cadastramento a partir deste caso de uso. [Include Caso de Uso Manter Cliente]

##### **– Validação do Status**

O status "aguardando VR" só pode ser alterado para "aguardando aviso", "cancelado pelo passageiro" ou "cancelado pela cooperativa por falta de carro" (este último, no caso específico de corrida imediata).

O status "aguardando aviso" só pode ser alterado para "aviso efetuado" ou "cancelado pelo passageiro".

O status "aviso efetuado" só pode ser alterado para "tripulado" ou "cancelado pelo passageiro".

Os status "tripulado" e "cancelado" não podem ser alterados.

## **36. CENÁRIO: TREINADOR DE VÔLEI**

Marcão é um treinador de vôlei e deseja levar um Notebook para os jogos, a fim de obter:

– controle do placar

– controle dos pontos de cada partida, identificando-os como: ponto de saque, ponto de ataque (quando a vantagem estiver com o time adversário), ponto de contra-ataque (quando a vantagem estiver com o próprio time), ponto de bloqueio, erro do adversário. No caso de bloqueio é necessário cadastrar se foi individual, duplo ou triplo.

São requisitos para a implantação dessa aplicação:

– cadastrar o nome de todos os jogadores do time e o número de suas camisas;

– para cada jogo agendado, cadastrar: a data e hora do jogo, o local, o nome do time adversário, os nomes do juiz e do juiz auxiliar.

– a aplicação deve exibir para controle em cada set o placar que pode ser alterado pelo auxiliar técnico, informando quem fez o último ponto e o tipo

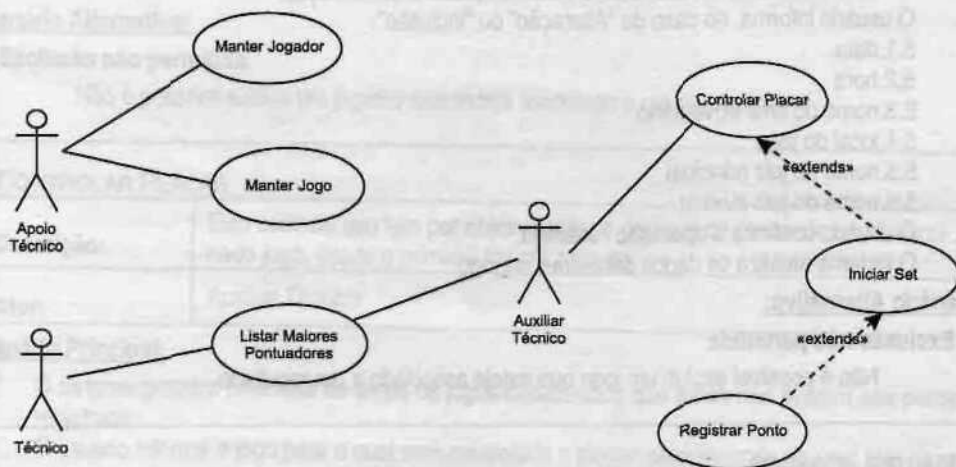
do ponto. No caso do ponto ser do time adversário, basta identificar o tipo do ponto.

– ao final de um jogo, o sistema deve exibir a lista dos maiores pontuadores e o somatório de pontos, por tipo, do jogo.

### EXERCÍCIO:

A partir do cenário descrito, desenhe o diagrama de casos de uso e escreva os cenários. Considere que: o cadastramento dos jogadores e das partidas é feito pelo Apoio. O cadastramento das informações de um jogo é feito pelo Auxiliar Técnico. O resumo de final de partida é acessado tanto pelo Auxiliar Técnico quanto pelo Técnico.

**RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):**



MANTER JOGO	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a inclusão ou alteração de dados ligados ao cadastro de jogos (competições).
<b>Ator:</b>	Apoio Técnico

### Cenário Principal:

1. O sistema prepara uma lista de todos os jogos cadastrados.
2. O sistema oferece ao usuário:
  - 2.1. selecionar um jogo, para alterar seu cadastro;
  - 2.2. localizar um jogo ou grupo de jogos por meio de pesquisa;



- 2.3. selecionar a opção de "inserir jogo".
3. Pesquisa de Jogo
  - 3.1. Para localizar um jogo, o usuário deve inserir a data e/ou um trecho do nome do time adversário como critério de pesquisa. O sistema fará a busca parcial.
  - 3.2. O sistema exibe a lista de jogos que satisfaça o critério, exibindo para cada um:
    - 3.2.1. data do jogo
    - 3.2.2. hora do jogo
    - 3.2.3. nome do adversário
4. Manutenção do Cadastro
  - 4.1. Após selecionar um jogo, o sistema habilita as opções de "Alteração", "Consulta" ou "Exclusão" de um jogo.
  - 4.2. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 4.3. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 4.4. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 4.4.1. No caso de exclusão, o sistema solicita a confirmação.
5. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 5.1. data
  - 5.2. hora
  - 5.3. nome do time adversário
  - 5.4. local do jogo
  - 5.5. nome do juiz principal
  - 5.6. nome do juiz auxiliar
6. O usuário confirma a operação realizada.
7. O sistema atualiza os dados cadastrais do jogo.

**Cenário Alternativo:****- Exclusão não permitida**

Não é possível excluir um jogo que esteja associado a um resultado.

MANTER JOGADOR	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a inclusão ou alteração de dados referentes ao cadastro de jogadores.
<b>Ator:</b>	Apoio Técnico

**Cenário Principal:**

1. O sistema prepara uma lista de todos os jogadores cadastrados.
2. O sistema oferece ao usuário:
  - 2.1. selecionar um jogador, para alterar seu cadastro;
  - 2.2. localizar um jogador ou grupo de jogadores por meio de pesquisa;
  - 2.3. selecionar a opção de "inserir jogador".
3. Pesquisa de Jogador
  - 3.1. Para localizar um jogador, o usuário deve inserir um trecho do nome como critério de pesquisa. O sistema fará a busca parcial.

- 3.2. O sistema exibe a lista de jogadores que satisfaça o critério, exibindo para cada um:
  - 3.2.1. nome do jogador
  - 3.2.2. número da camisa
4. Manutenção do Cadastro
  - 4.1. Após selecionar um jogador, o sistema habilita as opções de "Alteração", "Consulta" ou "Exclusão" de um jogador.
  - 4.2. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 4.3. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 4.4. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 4.4.1. No caso de exclusão, o sistema solicita a confirmação.
5. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 5.1. nome do jogador
  - 5.2. número da camisa
6. O usuário confirma a operação realizada.
7. O sistema atualiza os dados cadastrais do jogador.

**Cenário Alternativo:****- Exclusão não permitida**

Não é possível excluir um jogador que esteja associado a um resultado.

CONTROLAR PLACAR	
<b>Descrição:</b>	Este caso de uso tem por objetivo fazer o controle do placar de um determinado jogo, desde o primeiro set até o último.
<b>Ator:</b>	Auxiliar Técnico

**Cenário Principal:**

1. O sistema prepara uma lista de todos os jogos cadastrados que ainda não tiveram seu placar registrado.
2. O usuário informa o jogo para o qual será controlado o placar, selecionando de uma lista preexistente.
  - 2.1. O sistema exibe a data e hora do jogo, o nome do adversário, o nome do juiz principal e o nome do juiz auxiliar.
3. O sistema exibe o placar e habilita a opção de "Iniciar Set".
  - 3.1. [Extends Iniciar Set]
  - 3.2. O sistema obtém o placar e o associa ao set jogado.
  - 3.3. O sistema verifica quem foi o ganhador do set, incrementando no placar o número de sets ganhos do time que venceu o set.
  - 3.4. Se nenhum dos times tiver alcançado três vitórias, num limite de cinco sets, o sistema se prepara para um novo set. (Passo 3).
4. O sistema atualiza os dados cadastrais do jogo, incluindo os resultados dos sets.

INICIAR SET	
<b>Descrição:</b>	Este caso de uso tem por objetivo dar início ao controle de um set de vôlei.
<b>Ator:</b>	Auxiliar Técnico

**Pré-condição:**

Receber a identificação se o set é de *tie-break*.

**Cenário Principal:**

1. O sistema prepara uma lista de jogadores do time.
2. O usuário seleciona os jogadores que iniciarão a partida, cadastrando-os na seguinte ordem: rede – ataque – esquerda, rede – ataque – meio, rede – ataque – direita, defesa – esquerda, defesa – meio, defesa – direita).
3. O usuário informa qual time iniciará o jogo.
4. O sistema controla cada ponto do set.
  - 4.1.[Extends Caso de Uso Registrar Ponto]
  - 4.2. Se o set não for de *tie-break*, os pontos são controlados até que um dos times atinja 25 pontos, tendo dois pontos de diferença para o time adversário. Se não houver essa diferença, a pontuação prossegue até que essa diferença seja atingida.
  - 4.3. Se o set for de *tie-break*, os pontos são controlados até que um dos times atinja 15 pontos, tendo dois pontos de diferença para o time adversário. Se não houver essa diferença, a pontuação prossegue até que essa diferença seja atingida.
5. O sistema retorna o placar do set.

**Pós-condição:**

Retornar o placar do set.

REGISTRAR PONTO	
<b>Descrição:</b>	Este caso de uso tem por objetivo identificar quem foi responsável por um ponto num determinado set, e registrar o tipo de ponto.
<b>Ator:</b>	Auxiliar Técnico

**Pré-condição:**

Receber a informação de que time encontra-se com a vantagem no jogo.

**Cenário Principal:**

1. O sistema prepara uma lista dos jogadores do time.
2. O usuário informa quem fez o ponto, selecionando entre as opções: nosso time e time adversário.
3. O usuário informa o tipo de ponto, selecionando dentre as seguintes opções:
  - 3.1. Se o time pontuador estava com a vantagem:
    - 3.1.1. ponto de saque
    - 3.1.2. ponto de ataque
    - 3.1.3. ponto de bloqueio

- 3.1.4. erro do adversário
- 3.2. Se o time pontuador não estava com a vantagem:
  - 3.2.1. ponto de contra-ataque
  - 3.2.2. ponto de bloqueio
  - 3.2.3. erro do adversário
- 4. Se for escolhido ponto de bloqueio, o usuário deve informar o tipo do bloqueio, selecionando entre as seguintes opções: individual, duplo ou triplo.
- 5. Se o time pontuador for o próprio time e o ponto não tiver sido de "erro de adversário", o usuário deve selecionar o(s) jogador(es) que foi(ram) responsável(is) pelo ponto.

**Pós-condição:**

Retornar os dados relacionados ao ponto.

LISTAR MAIORES PONTUADORES	
<b>Descrição:</b>	Este caso de uso tem por objetivo oferecer uma consulta dos maiores pontuadores de um jogo e o somatório de pontos, por tipo, do jogo.
<b>Ator:</b>	Auxiliar Técnico, Técnico

**Pré-condição:**

Receber a identificação do jogo.

**Cenário Principal:**

1. O sistema contabiliza e exibe o total de pontos, agrupados por set e por tipo de ponto (saque, ataque, contra-ataque, bloqueio e erro do adversário).
2. O sistema contabiliza e exibe, em ordem crescente, o total de pontos conseguido, agrupados por jogador e depois por tipo de ponto (saque, ataque, contra-ataque, bloqueio e erro do adversário).

## 37. CENÁRIO: CONTROLE DE OBRA

Álvaro está fazendo uma ampliação de sua residência. Todo dia existe demanda de compra de material. Sendo assim, ele desenvolveu uma pequena aplicação que controla essa demanda de solicitações e as compras efetuadas, de forma a montar uma base de cotações para as compras futuras.

A aplicação possui um cadastro de produtos, contendo: nome, descrição, medida de venda do produto (kg, ml ou m; indicando peso, volume ou comprimento) e valor da medida de venda (ex: 1,5).

A cada solicitação de compra cadastram-se os itens dessa solicitação. Cada item possui: o produto e a quantidade. Quando cada item é adquirido, atualiza-se a solicitação com o preço unitário de compra, a forma de pagamento (dinheiro, cheque, cheque pré ou cartão), a data de compra e o local da compra.

São controles oferecidos pela aplicação:

Quando há uma nova solicitação, é possível obter de cada item a lista dos três menores preços que já foram pagos para o referido produto, incluindo na listagem o local onde foi comprado.

A lista de compras é impressa a partir dos itens que não foram fechados, de todas as solicitações de compra que estejam com status em aberto.

Uma solicitação pode ser cancelada (status = "cancelado").

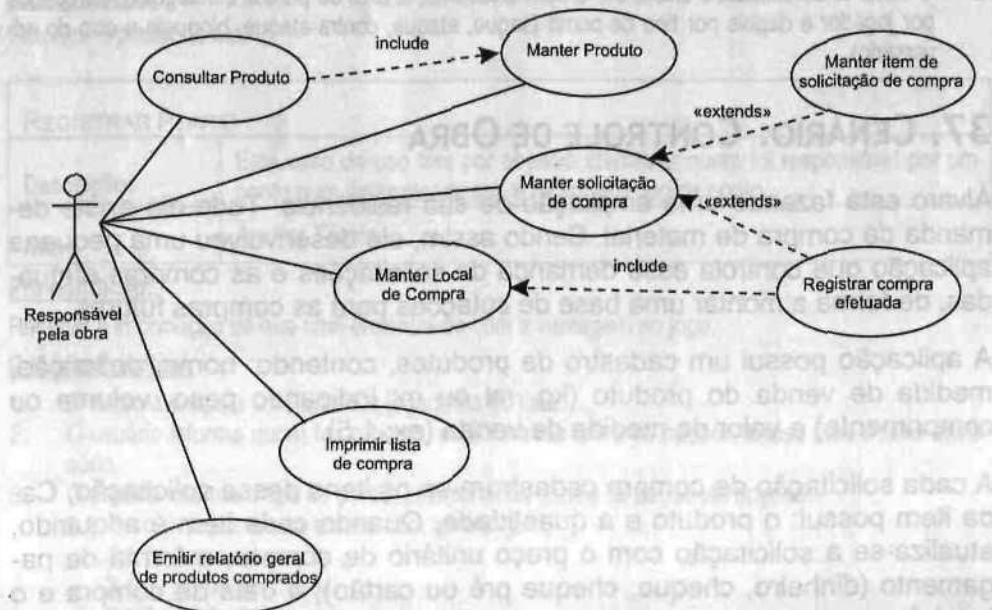
Quando todos os itens de uma solicitação tiverem sido comprados, o sistema atualiza automaticamente o status dessa solicitação para "fechado".

Deve ser emitida uma listagem de todos os produtos já comprados, com seu somatório de quantidade e de valor.

#### EXERCÍCIO:

A partir do cenário descrito, desenhe o diagrama de casos de uso desse sistema. Escreva, também, os cenários. Considere que todas as operações são feitas pelo Álvaro, que pode ser identificado como Responsável pela Obra.

#### RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):





CONSULTAR PRODUTO	
<b>Descrição:</b>	Este caso de uso tem por objetivo apresentar os produtos cadastrados e habilitar a inclusão, alteração ou exclusão de produtos.
<b>Ator:</b>	Responsável pela obra

**Cenário Principal:**

1. O sistema prepara uma lista de todos os produtos cadastrados.
2. O sistema oferece ao usuário:
  - 2.1. selecionar um produto, para alterar seu cadastro;
  - 2.2. localizar um produto ou conjunto de produtos por meio de pesquisa;
  - 2.3. selecionar a opção de "inserir produto".
3. Pesquisa de Produto
  - 3.1. Para localizar um produto, o usuário deve inserir um trecho do nome do produto como critério de pesquisa. O sistema fará a busca parcial.
  - 3.2. O sistema exibe a lista de produtos que satisfaça o critério, exibindo para cada um:
    - 3.2.1. nome do produto
    - 3.2.2. descrição
4. Inserção de Produto
  - 4.1. [Inclui Caso de Uso Manter Produto]
5. Seleção de Produto
  - 5.1. Após selecionar um produto, o sistema habilita as opções de "alterar produto" e "excluir produto".
  - 5.2. Se o usuário selecionar uma dessas opções, o sistema aciona o cadastro de produto. [Inclui Caso de Uso Manter Produto]

MANTER PRODUTO	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a manutenção dos dados cadastrais de produto (inclusão, alteração e exclusão).
<b>Ator:</b>	Responsável pela obra

**Pré-condição:**

Receber a identificação do tipo de operação e os dados cadastrais do produto, no caso de alteração ou exclusão.

**Cenário Principal:**

1. Manutenção do Cadastro
  - 1.1. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 1.2. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 1.3. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 1.3.1. No caso de exclusão, o sistema solicita a confirmação.
2. O usuário informa, no caso de "Alteração" ou "Inclusão":

- 2.1. nome do produto
- 2.2. descrição do produto
- 2.3. medida de venda do produto, escolhido entre as opções: peso (kg), volume (ml) ou comprimento (m)
- 2.4. valor da medida de venda, como por exemplo 1,5
3. O usuário confirma a operação realizada.
4. O sistema atualiza os dados cadastrais do produto.
  - 4.1. No caso de inclusão, o sistema gera automaticamente um código de identificação.

**Cenário Alternativo:****- Exclusão não permitida**

Não é possível excluir um produto que esteja associado a uma solicitação de compra.

MANTER SOLICITAÇÃO DE COMPRA	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a manutenção das solicitações de compras (inclusão, alteração e exclusão)
<b>Ator:</b>	Responsável pela obra

**Cenário Principal:**

1. O sistema prepara uma lista de todas as solicitações cadastradas, que ainda não tenham sido fechadas (status = "aberto").
2. O sistema oferece ao usuário:
  - 2.1. selecionar uma solicitação, para alterar seu cadastro;
  - 2.2. localizar uma solicitação ou grupo de solicitações por meio de pesquisa;
  - 2.3. selecionar a opção de "inserir solicitação de compra".
3. Pesquisa de Solicitação de Compra
  - 3.1. Para localizar um produto, o usuário deve inserir a data da solicitação e/ou um trecho do nome de algum produto que conste da solicitação como critério de pesquisa. O sistema fará a busca parcial.
  - 3.2. O sistema exibe a lista de solicitações de compra que satisfaçam o critério, exibindo para cada um:
    - 3.2.1. data da solicitação
    - 3.2.2. lista de produtos que fazem parte da solicitação
4. Manutenção do Cadastro
  - 4.1. Após selecionar uma solicitação de compra, o sistema habilita as opções de "Alteração", "Consulta" ou "Cancelamento".
  - 4.2. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 4.3. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 4.4. Em caso de "Consulta" ou "Cancelamento", o sistema exibe os dados cadastrados desabilitados para edição.
    - 4.4.1. No caso de cancelamento, o sistema solicita a confirmação.
5. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 5.1. data da solicitação
  - 5.2. itens da solicitação

- 5.2.1. Extends [Caso de Uso Manter Item de Solicitação de Compra]
6. O usuário pode solicitar o cancelamento da solicitação de compra.
    - 6.1. O sistema altera o status da solicitação de compra para "cancelada".
  7. O usuário confirma a operação realizada.
  8. O sistema atualiza os dados cadastrais da solicitação de compra.
    - 8.1. No caso de inclusão, o sistema atualiza o status com o valor "aberto".

#### Cenário Alternativo:

##### **- Cancelamento não permitido**

Não é possível cancelar uma solicitação de compra que já tenha tido algum item comprado. Nesse caso, só é possível cancelar os itens da solicitação.

MANTER ITEM DE SOLICITAÇÃO DE COMPRA	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a manutenção dos itens de uma solicitação de compras (inclusão, alteração e exclusão).
<b>Ator:</b>	Responsável pela obra

#### Cenário Principal:

1. O sistema prepara uma lista de todos os itens cadastrados.
2. O sistema oferece ao usuário:
  - 2.1. selecionar um item, para alterar seu cadastro;
  - 2.2. localizar um item ou grupo de itens por meio de pesquisa;
  - 2.3. selecionar a opção de "inserir item".
3. Pesquisa de Item
  - 3.1. Para localizar um item, o usuário deve inserir trecho do nome de algum produto como critério de pesquisa. O sistema fará a busca parcial.
  - 3.2. O sistema exibe a lista de itens que satisfaça o critério, exibindo para cada um:
    - 3.2.1. produto
    - 3.2.2. quantidade solicitada
4. Manutenção do Cadastro
  - 4.1. Após selecionar um item, o sistema habilita as opções de "Alteração", "Consulta" ou "Exclusão" de um item.
  - 4.2. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 4.3. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 4.4. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 4.4.1. No caso de exclusão, o sistema solicita a confirmação.
5. O sistema prepara uma lista de todos os produtos cadastrados.
6. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 6.1. produto, selecionado de uma lista preexistente;
  - 6.2. quantidade solicitada.
7. O usuário confirma a operação realizada.
8. O sistema atualiza os dados cadastrais do item.

**Cenário Alternativo:****- Exclusão não permitida**

Não é possível excluir um item que esteja associado a uma compra efetuada.

REGISTRAR COMPRA EFETUADA	
<b>Descrição:</b>	Este caso de uso tem por objetivo registrar as compras que foram efetuadas para atender a uma determinada solicitação de compra.
<b>Ator:</b>	Responsável pela obra

**Cenário Principal:**

1. O sistema apresenta todos os itens de compra que ainda não foram atendidos, agrupando-os e totalizando a quantidade daqueles que sejam do mesmo produto.
2. O sistema deve preparar uma lista dos locais cadastrados.
3. Para cada item de compra do qual o usuário queira registrar a aquisição:
  - 3.1. O usuário seleciona um dos itens de compra para registrar sua aquisição.
  - 3.2. O usuário informa:
    - 3.2.1. a quantidade adquirida.
    - 3.2.2. a forma de pagamento, selecionada entre as opções: dinheiro, cheque, cheque pré ou cartão.
    - 3.2.3. a data da compra.
    - 3.2.4. local da compra, selecionado de uma lista preexistente.
4. O usuário confirma a operação realizada.
5. O sistema atualiza os dados cadastrais do item de compra.

**Cenário Alternativo:****- Local Inexistente**

Se não existir um local cadastrado, permitir a inclusão de um local. Inclui [Caso de Uso Manter Local de Compra]

**- Quantidade adquirida diferente da quantidade solicitada**

Se a quantidade adquirida for inferior à quantidade solicitada, o sistema deve aceitar e não fechar o item de compra como atendido. No próximo Registro de Compra, este item deve aparecer somente com a quantidade faltante.

MANTER LOCAL DE COMPRA	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a manutenção dos locais de compra onde são adquiridos os itens da obra.
<b>Ator:</b>	Responsável pela obra

**Cenário Principal:**

1. O sistema prepara uma lista de locais de compra cadastrados.
2. O sistema oferece ao usuário:

- 2.1. selecionar um local, para alterar seu cadastro;
- 2.2. localizar um local por meio de pesquisa;
- 2.3. selecionar a opção de "inserir local".
3. Manutenção do Cadastro
  - 3.1. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 3.2. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 3.3. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 3.3.1. No caso de exclusão, o sistema solicita a confirmação.
4. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 4.1. nome do estabelecimento.
  - 4.2. endereço completo, contendo: nome do logradouro, número, complemento e bairro.
  - 4.3. telefones de contato.
5. O usuário confirma a operação realizada.
6. O sistema atualiza os dados cadastrais dos locais de compra.
  - 6.1. No caso de inclusão, o sistema gera automaticamente um código de identificação.

**Cenário Alternativo:****- Exclusão não permitida**

Não é possível excluir um local de compra que esteja associado a um item de compra.

IMPRIMIR LISTA DE COMPRA	
<b>Descrição:</b>	Este caso de uso tem por objetivo gerar uma lista de compra, contendo todos os itens ainda não adquiridos de todas as solicitações de compra que ainda encontram-se abertas, agrupando por produto, a fim de facilitar e agilizar a compra.
<b>Ator:</b>	Responsável pela obra

**Cenário Principal:**

1. O sistema obtém a lista de todos os itens de compra que ainda não foram atendidos, agrupando e totalizando a quantidade daqueles que sejam do mesmo produto.
2. Para cada item de compra obtido:
  - 2.1. O sistema busca todas as compras já realizadas para o referido item.
  - 2.2. Desse conjunto, o sistema obtém os três menores valores já pagos para aquele item, trazendo para cada valor, a data de aquisição e o local de compra.
3. O sistema exibe um relatório contendo:
  - 3.1. item de compra
  - 3.2. quantidade total solicitada
  - 3.3. valor, data e local dos três menores preços



EMITIR RELATÓRIO GERAL DE PRODUTOS COMPRADOS	
<b>Descrição:</b>	Este caso de uso tem por objetivo emitir um relatório geral contendo todos os produtos comprados num determinado período.
<b>Ator:</b>	Responsável pela obra

**Cenário Principal:**

1. O usuário informa o período desejado para o relatório (data de início e data de término).
2. O sistema busca todos os itens de compra que já foram adquiridos, que estejam dentro do período informado pelo usuário.
  - 2.1. O sistema agrupa todos os produtos comprados, calculando o que foi gasto para cada produto.
3. O sistema emite um relatório contendo os seguintes dados:
  - 3.1. nome do produto;
  - 3.2. total adquirido do produto;
  - 3.3. total gasto na aquisição;
  - 3.4. somatório geral do total gasto com os produtos.

## 38. CENÁRIO: CONTROLE DE OBRA (VARIAÇÃO A)

Álvaro sentiu a necessidade de saber quanto estava gastando com acabamento, alvenaria etc. Sendo assim, preparou uma nova versão de sua aplicação que controlasse o tipo do produto.

**EXERCÍCIO:**

Que novo caso de uso precisaria ser incluído nesse modelo?

**RESOLUÇÃO:**

- O CASO DE USO MANTER TIPO DE PRODUTO.

## 39. CENÁRIO: SENHA DE ATENDIMENTO

A empresa Compre Bem implantou uma senha de atendimento para o SAC de suas lojas. O objetivo é reduzir o tempo de espera na fila.

O atendimento é dividido por assuntos e cada caixa pode cuidar de um ou mais assuntos, ou um assunto pode ser tratado por um ou mais caixas.

Para cada caixa deve-se saber o número e a posição (direita ou esquerda da máquina de senhas).

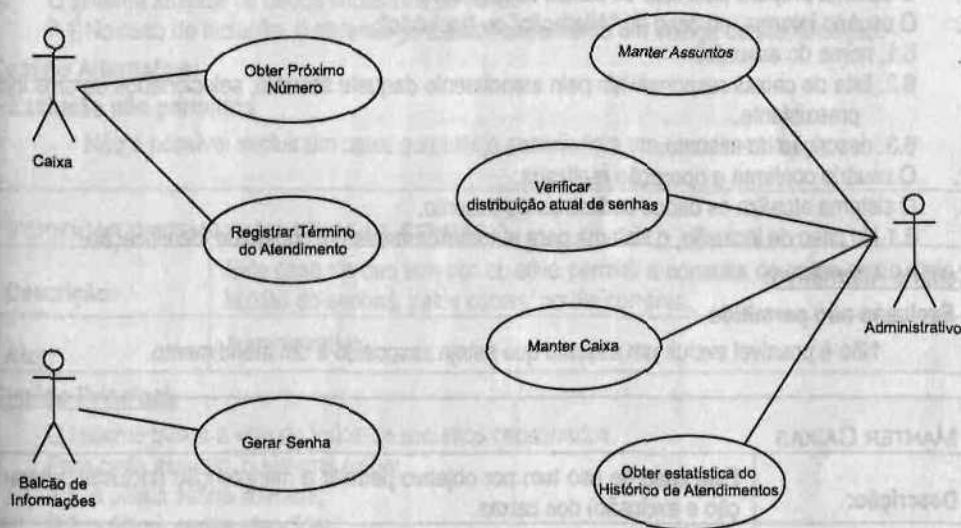
Para cada caixa, deve-se ter um histórico de atendimentos, para se obter estatística. A estatística deve ser detalhada quanto ao tempo mínimo, médio e máximo de atendimento por caixa e por dia, além do número de atendimentos por assunto.

A qualquer momento é preciso saber que caixa está com um determinado número de atendimento.

#### EXERCÍCIO:

A partir do cenário descrito, desenhe o diagrama de casos de uso desse sistema. Escreva, também, os cenários. Considere que as tarefas de controlar os assuntos e os caixas, obter estatística e relatórios são do Setor Administrativo. O Caixa se responsabiliza por controlar a próxima senha e o Balcão de Informações é que gera novas senhas.

#### RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):



#### MANTER ASSUNTOS

##### Descrição:

Este caso de uso tem por objetivo permitir a manutenção (inclusão, alteração e exclusão) dos assuntos controlados no atendimento.

##### Ator:

Administrativo

#### Cenário Principal:

1. O sistema prepara uma lista de assuntos cadastrados.
2. O sistema oferece ao usuário:

- 2.1. selecionar um assunto, para alterar seu cadastro;
- 2.2. localizar um assunto por meio de pesquisa;
- 2.3. selecionar a opção de "inserir assunto".
3. Pesquisa de Assunto
  - 3.1. Para localizar um assunto, o usuário deve inserir um trecho do nome como critério de pesquisa. O sistema fará a busca parcial.
  - 3.2. O sistema exibe a lista de assuntos que satisfaça o critério, exibindo para cada um:
    - 3.2.1. nome do assunto
    - 3.2.2. os caixas responsáveis
4. Manutenção do Cadastro
  - 4.1. Após selecionar um assunto, o sistema habilita as opções de "Alteração", "Consulta" ou "Exclusão" de um assunto.
  - 4.2. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 4.3. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 4.4. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 4.4.1. No caso de exclusão, o sistema solicita a confirmação.
5. O sistema prepara uma lista de caixas cadastrados.
6. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 6.1. nome do assunto.
  - 6.2. lista de caixas responsáveis pelo atendimento daquele assunto, selecionados de uma lista preexistente.
  - 6.3. descrição do assunto.
7. O usuário confirma a operação realizada.
8. O sistema atualiza os dados cadastrais do assunto.
  - 8.1. No caso de inclusão, o sistema gera automaticamente um código de identificação.

**Cenário Alternativo:****– Exclusão não permitida**

Não é possível excluir um assunto que esteja associado a um atendimento.

MANTER CAIXAS	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a manutenção (inclusão, alteração e exclusão) dos caixas.
<b>Ator:</b>	Administrativo

**Cenário Principal:**

1. O sistema prepara uma lista de todos os caixas cadastrados.
2. O sistema oferece ao usuário:
  - 2.1. selecionar um caixa, para alterar seu cadastro;
  - 2.2. localizar um caixa por meio de pesquisa;
  - 2.3. selecionar a opção de "inserir caixa".

3. Pesquisa de Caixa
  - 3.1. Para localizar um caixa, o usuário deve inserir um trecho do nome do atendente como critério de pesquisa. O sistema fará a busca parcial.
  - 3.2. O sistema exibe a lista de caixas que satisfaça o critério, exibindo para cada um:
    - 3.2.1. número do caixa
    - 3.2.2. atendente responsável
4. Manutenção do Cadastro
  - 4.1. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 4.2. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 4.3. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 4.3.1. No caso de exclusão, o sistema solicita a confirmação.
5. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 5.1. número do caixa
  - 5.2. atendente responsável
  - 5.3. posição do caixa, selecionando entre as opções: lado esquerdo ou lado direito.
6. O usuário confirma a operação realizada.
7. O sistema atualiza os dados cadastrais do caixa.
  - 7.1. No caso de inclusão, o sistema gera automaticamente um código de identificação.

#### **Cenário Alternativo:**

##### **– Exclusão não permitida**

Não é possível excluir um caixa que esteja associado a um atendimento.

VERIFICAR DISTRIBUIÇÃO ATUAL DE SENHAS	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a consulta de como estão distribuídas as senhas, pelos caixas, no dia corrente.
<b>Ator:</b>	Administrativo

#### **Cenário Principal:**

1. O sistema busca a lista de todos os assuntos cadastrados.
2. Para cada assunto, o sistema exibe:
  - 2.1. a última senha liberada;
  - 2.2. a última senha atendida;
  - 2.3. o caixa que está atendendo ou que atendeu a última senha;
  - 2.4. horário de início do último atendimento;
  - 2.5. horário de término (se houver) do último atendimento.

## OBTHER ESTATÍSTICA DO HISTÓRICO DE ATENDIMENTOS

<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a consulta ao histórico de atendimentos, gerando estatística de tempo gasto para cada atendimento.
<b>Ator:</b>	Administrativo

**Cenário Principal:**

1. O usuário informa um período (data de início e data de fim) de pesquisa.
2. O sistema busca todos os atendimentos efetuados dentro do período informado.
3. Para cada dia do referido período, são agrupados os atendimentos por caixa. Para cada caixa é exibido:
  - 3.1. número do caixa
  - 3.2. nome do atendente responsável pelo caixa
  - 3.3. lista de atendimentos, exibindo para cada um:
    - 3.3.1. horário de início
    - 3.3.2. horário de término
    - 3.3.3. tempo gasto no atendimento
    - 3.3.4. assunto de referência do atendimento
4. Para cada caixa, o sistema calcula e exibe o tempo mínimo, médio e máximo de atendimento.
5. Para cada dia do período de busca, o sistema calcula e exibe:
  - 5.1. o tempo médio de atendimento
  - 5.2. o caixa que teve o menor índice médio de atendimento
  - 5.3. o caixa que teve o maior índice médio de atendimento
6. Ao final, o sistema calcula e exibe o total de atendimentos, por assunto, exibindo em ordem decrescente:
  - 6.1. o nome do assunto
  - 6.2. a quantidade de atendimentos

## OBTHER PRÓXIMO NÚMERO

<b>Descrição:</b>	Este caso de uso tem por objetivo verificar para um determinado caixa e assunto que ele trata, qual é a próxima senha de atendimento, exibindo-a para os clientes.
<b>Ator:</b>	Caixa

**Pré-condição:**

Receber a identificação do caixa que está operando o sistema.

**Cenário Principal:**

1. O sistema verifica todos os assuntos que são atendidos pelo caixa.
2. O sistema verifica o último assunto atendido, com sua respectiva senha.
  - 2.1. Se o caixa atender mais de um assunto, o sistema verifica a última senha atendida do próximo assunto do seu conjunto de atendimentos.
3. O sistema verifica se a última senha atendida é menor que a última senha gerada.
  - 3.1. Se for, o sistema incrementa a última senha atendida em uma unidade.
4. O sistema atualiza o cadastro de senhas.



5. O sistema exibe no painel de senhas:
  - 5.1. o número da senha
  - 5.2. o código do assunto
  - 5.3. a posição do Caixa (esquerda ou direita)

REGISTRAR TÉRMINO DE ATENDIMENTO	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir que o usuário registre o término do seu atendimento, preparando para a obtenção da próxima senha.
<b>Ator:</b>	Caixa

**Pré-condição:**

Receber a identificação do caixa que está operando o sistema.

**Cenário Principal:**

1. O sistema verifica o último atendimento feito para a referida caixa, habilitando a opção de encerrar atendimento.
2. O usuário seleciona a opção de encerrar atendimento.
3. O sistema atualiza o histórico de atendimentos, registrando a data de término do atendimento.

GERAR SENHA	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a geração de uma senha baseada numa lista de assuntos.
<b>Ator:</b>	Balcão de Informações

**Cenário Principal:**

1. O sistema prepara uma lista de todos os assuntos cadastrados.
2. O usuário informa um assunto, selecionado de uma lista preexistente.
3. O sistema obtém a última senha gerada para aquele assunto e incrementa esse valor em uma unidade.
4. O sistema atualiza o cadastro de senhas geradas, colocando o novo valor associado ao assunto.
5. O sistema emite um ticket impresso informando:
  - 5.1. data do atendimento
  - 5.2. assunto
  - 5.3. senha gerada

**Cenário Alternativo:****– Primeiro acesso do dia**

O sistema verifica se é o primeiro acesso do dia para geração de senhas. Se for, todas as senhas associadas a todos os assuntos são zeradas.

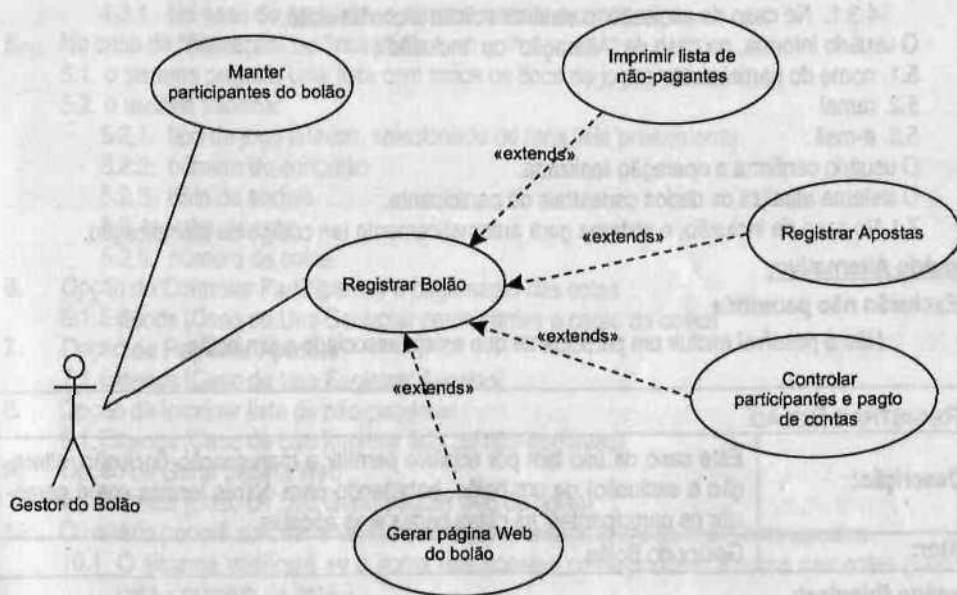
## 40. CENÁRIO: CONTROLE DE BOLÃO

Jairo trabalha no Departamento de Informática de uma grande empresa. Ele e seus amigos estão sempre fazendo bolão da MegaSena, Quina e outros tipos de jogos. Jairo sempre controla numa planilha Excel os números apostados, além das pessoas que entraram no bolão, seus e-mails (para receberem os números apostados) e se pagaram suas cotas. Entretanto, isso tem lhe tomado um tempo considerável. Sendo assim, ele pensou em desenvolver uma aplicação que atenda às seguintes funcionalidades:

- permita cadastrar os participantes de cada bolão, com seus ramais e e-mails;
- para cada bolão feito, cadastrar o valor da cota, número de cotas, os cartões apostados (com sua relação de números), o tipo de jogo (MegaSena, Quina etc.), o número do concurso e a data em que será realizado o sorteio;
- controlar quem pagou cada cota;
- gerar automaticamente uma página Web com os dados do sorteio, participantes do bolão com suas cotas e os números apostados. O arquivo HTML dessa página será enviada por e-mail;
- cada participante poderá adquirir mais de uma cota;
- gerar a lista de participantes que ainda não pagaram;
- a aplicação deve verificar se o total das cotas é igual ao total apostado;
- uma determinada aposta pode ser aproveitada em outros bolões.

### EXERCÍCIO:

A partir do cenário descrito, desenhe o diagrama de casos de uso desse sistema. Escreva, também, os cenários. Considere que todas as operações são feitas pelo Jairo, que pode ser identificado como Gestor do Bolão.

**RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):****MANTER PARTICIPANTES DO BOLÃO****Descrição:**

Este caso de uso tem por objetivo permitir a manutenção (inclusão, alteração e exclusão) dos participantes que possam vir a participar de um bolão.

**Ator:**

Gestor do Bolão

**Cenário Principal:**

1. O sistema prepara uma lista de participantes cadastrados.
2. O sistema oferece ao usuário:
  - 2.1. selecionar um participante, para alterar seu cadastro;
  - 2.2. localizar um participante por meio de pesquisa;
  - 2.3. selecionar a opção de "inserir participante".
3. Pesquisa de Participante
  - 3.1. Para localizar um participante, o usuário deve inserir um trecho do nome como critério de pesquisa. O sistema fará a busca parcial.
  - 3.2. O sistema exibe a lista de participantes que satisfaça o critério, exibindo para cada um:
    - 3.2.1. nome do participante
    - 3.2.2. ramal
4. Manutenção do Cadastro
  - 4.1. Em caso de "inclusão", o sistema habilita a edição dos dados.
  - 4.2. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.

4.3. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.

4.3.1. No caso de exclusão, o sistema solicita a confirmação.

5. O usuário informa, no caso de "Alteração" ou "Inclusão":

5.1. nome do participante

5.2. ramal

5.3. e-mail

6. O usuário confirma a operação realizada.

7. O sistema atualiza os dados cadastrais do participante.

7.1. No caso de inclusão, o sistema gera automaticamente um código de identificação.

#### **Cenário Alternativo:**

##### **– Exclusão não permitida**

Não é possível excluir um participante que esteja associado a um bolão.

REGISTRAR BOLÃO	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a manutenção (inclusão, alteração e exclusão) de um bolão, habilitando para outras tarefas como gerenciar os participantes, as cotas pagas e as apostas.
<b>Ator:</b>	Gestor do Bolão.

#### **Cenário Principal:**

- O sistema prepara uma lista de todos os bolões cadastrados.
- O sistema oferece ao usuário:
  - localizar um bolão por meio de pesquisa.
  - selecionar a opção de "inserir bolão".
  - selecionar um bolão, para realizar as seguintes operações, caso o sorteio a que se refere o bolão ainda não tenha sido realizado:
    - alterar o cadastro do bolão, com as opções de "Alteração", "Consulta" e "Exclusão".
    - controlar participantes e pagamento das cotas
    - registrar apostas
  - selecionar um bolão, para realizar as seguintes operações, em qualquer situação do bolão:
    - imprimir lista de não-pagantes
    - gerar página web do bolão
- Pesquisa de Bolão
  - Para localizar um bolão, o usuário deve inserir a data do sorteio e/ou o tipo de jogo como critério de pesquisa
  - O sistema exibe a lista de bolões que satisfaçam o critério, exibindo para cada um:
    - tipo do jogo lotérico
    - número do concurso
    - data do sorteio
    - valor da cota
- Manutenção do Cadastro
  - Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.

- 4.3. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
  - 4.3.1. No caso de exclusão, o sistema solicita a confirmação.
5. No caso de "Alteração" ou "Inclusão":
  - 5.1. o sistema prepara uma lista com todos os tipos de jogos lotéricos.
  - 5.2. o usuário informa:
    - 5.2.1. tipo de jogo lotérico, selecionado de uma lista preexistente.
    - 5.2.2. número do concurso
    - 5.2.3. data do sorteio
    - 5.2.4. valor da cota
    - 5.2.5. número de cotas
6. Opção de Controlar Participantes e pagamento das cotas
  - 6.1. Extends [Caso de Uso Controlar participantes e pagto de cotas]
7. Opção de Registrar Apostas
  - 7.1. Extends [Caso de Uso Registrar Apostas]
8. Opção de Imprimir lista de não-pagantes
  - 8.1. Extends [Caso de Uso Imprimir lista de não-pagantes]
9. Opção de Gerar página Web
  - 9.1. Extends [Caso de Uso Gerar página web do bolão]
10. O usuário poderá solicitar a verificação de sincronismo entre as cotas e as apostas.
  - 10.1. O sistema verificará se a soma das apostas correspondem à soma das cotas (valor da cota x número de cotas).
11. O usuário confirma as alterações.
12. O sistema atualiza os dados cadastrais do bolão.

GERAR PÁGINA WEB DO BOLÃO	
<b>Descrição:</b>	Este caso de uso tem por objetivo gerar uma página Web (em HTML) contendo todas as informações sobre o bolão, para ser distribuída para os participantes.
<b>Ator:</b>	Gestor do Bolão

**Pré-condição:**

Receber a identificação do bolão.

**Cenário Principal:**

1. O sistema busca a lista de participantes do bolão, com as seguintes informações:
  - 1.1. nome do participante
  - 1.2. ramal do participante
  - 1.3. número de cotas
  - 1.4. situação de pagamento das cotas
2. O sistema busca os números apostados para o bolão.
3. O sistema busca os dados referentes ao bolão.
4. O sistema gera uma página web com as seguintes informações:
  - 4.1. tipo de jogo lotérico
  - 4.2. número do concurso



- 4.3. data do concurso
- 4.4. valor da cota
- 4.5. lista de participantes, com nome, ramal, número de cotas e situação de pagamento das cotas
- 4.6. lista de apostas, mostrando os números para cada aposta e o valor apostado.

IMPRIMIR LISTA DE NÃO-PAGANTES	
<b>Descrição:</b>	Este caso de uso tem por objetivo imprimir um relatório com os participantes de um bolão que não efetuaram o pagamento de suas cotas.
<b>Ator:</b>	Gestor do Bolão

**Pré-condição:**

Receber a identificação do bolão.

**Cenário Principal:**

1. O sistema busca a lista dos participantes do bolão que não estejam com as cotas pagas.
2. O sistema emite um relatório com os seguintes dados:
  - 2.1. nome do participante
  - 2.2. ramal do participante
  - 2.3. e-mail do participante
  - 2.4. quantidade de cotas em aberto para pagamento

REGISTRAR APOSTAS	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir o controle das apostas realizadas para um determinado bolão.
<b>Ator:</b>	Gestor do Bolão

**Pré-condição:**

Receber a identificação do bolão.

**Cenário Principal:**

1. O sistema busca e exibe todas as apostas cadastradas para o referido bolão.
2. O sistema oferece ao usuário:
  - 2.1. selecionar a opção de "inserir aposta".
  - 2.2. selecionar uma aposta, para realizar as seguintes operações:
    - 2.2.1. alterar a aposta, com as opções de "Alteração", "Consulta" e "Exclusão".
3. No caso de "Alteração" ou "Inclusão":
  - 3.1. o usuário informa:
    - 3.1.1. valor da aposta
    - 3.1.2. números apostados

**Cenário Alternativo:****– Permissão de inclusão de nova aposta**

O sistema só habilita a opção de "inserir aposta" se a soma dos valores das apostas cadastradas for menor que a soma das cotas (número de cotas x valor da cota).

**- Limite de números apostados**

O intervalo permitido para a quantidade de números apostados é de: valor de quantidade mínima de apostas do jogo lotérico como limite inferior, e valor de quantidade máxima de apostas do jogo lotérico como limite superior.

CONTROLAR PARTICIPANTES E PAGAMENTO DE COTA	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a manutenção (inclusão, alteração e exclusão) dos participantes efetivos de um bolão e do pagamento de suas cotas.
<b>Ator:</b>	Gestor do Bolão

**Pré-condição:**

Receber a identificação do bolão.

**Cenário Principal:**

1. O sistema busca e exibe a lista de participantes já cadastrados para o bolão, exibindo para cada um:
  - 1.1. o nome do participante
  - 1.2. o ramal do participante
  - 1.3. o número de cotas
  - 1.4. se as cotas foram pagas
2. O usuário pode alterar a quantidade de cotas ou o status do pagamento de um dos participantes ou cadastrar um novo participante.
  - 2.1. O sistema prepara uma lista de todos os participantes credenciados para o bolão que está cadastrado.
  - 2.2. O usuário informa um novo participante, selecionado de uma lista preexistente.
  - 2.3. O usuário informa a quantidade de cotas e se as cotas já foram pagas.

## 41. CENÁRIO: ESTACIONAMENTO

Bruno e seu pai compraram um terreno e inaugurarão um estacionamento.

Para ajudar, a irmã de Bruno está desenvolvendo uma aplicação de controle de estacionamento.

Quando o veículo entra no estacionamento, o atendente observa sua placa e a mesma é cadastrada, juntamente com o modelo do veículo e sua cor. A hora de entrada é gerada automaticamente, correspondendo ao momento do cadastramento da placa. Após estacionar o veículo, o cliente pega o ticket onde está impresso: o número da placa, o modelo do veículo, a cor, a data e a hora da entrada.

Ao retornar ao estacionamento, o cliente entrega o ticket. O tempo de permanência é calculado. Considerando esse tempo de permanência, é aplicada a tabela de preços, sabendo-se que a tabela de sábado não é a mesma dos dias úteis e, às vezes, dependendo da época do ano, os donos lançam promoções durante os dias úteis. Veja exemplo das tabelas de preço:

Segunda à sexta

1ª hora = R\$ 2,00

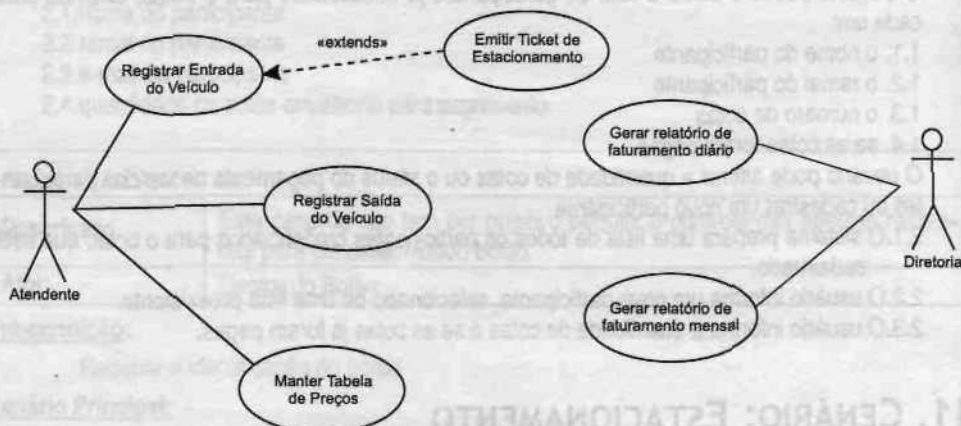
a partir da 2ª hora (inteiro ou fração) = + R\$ 1,00

Sábado

Preço único = R\$ 3,00

Os donos precisam de relatórios de faturamento diário e semanal.

**RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):**



REGISTRAR ENTRADA DO VEÍCULO	
<b>Descrição:</b>	Este caso de uso tem por objetivo registrar os dados do veículo que esteja entrando no estacionamento.
<b>Ator:</b>	Atendente

**Cenário Principal:**

1. O sistema prepara uma lista de modelos de carro.
2. O usuário informa:
  - 2.1. a placa do carro
  - 2.2. o modelo, selecionado de uma lista preexistente.
  - 2.3. a cor
3. O sistema verifica e registra automaticamente a data e a hora de início do estacionamento.

4. O usuário confirma as alterações.
5. O sistema atualiza os dados cadastrais do veículo.
  - 5.1. O sistema imprime o ticket de estacionamento, como comprovante do motorista. Extends [Caso de Uso Emitir Ticket de Estacionamento]

EMITIR TICKET DE ESTACIONAMENTO	
<b>Descrição:</b>	Este caso de uso tem por objetivo emitir o ticket de estacionamento que o cliente irá levar após estacionar o veículo.
<b>Ator:</b>	Atendente

**Cenário Principal:**

1. O sistema imprime:
  - 1.1. data de ocupação da vaga
  - 1.2. hora de início de ocupação da vaga
  - 1.3. placa do veículo
  - 1.4. modelo do veículo
  - 1.5. cor do veículo

REGISTRAR SAÍDA DO VEÍCULO	
<b>Descrição:</b>	Este caso de uso tem por objetivo registrar a saída do veículo, calculando o tempo de permanência e o valor a pagar pelo estacionamento.
<b>Ator:</b>	Atendente

**Cenário Principal:**

1. O sistema prepara uma lista dos veículos que ainda não tiveram sua saída registrada.
  - 1.1. Para cada veículo, é exibido:
    - 1.1.1. a placa do veículo
    - 1.1.2. a hora de início
2. O usuário informa a placa da qual será dada a saída, selecionando de uma lista preexistente.
  - 2.1. O sistema calcula o tempo de permanência.
  - 2.2. O sistema calcula o preço do estacionamento, baseado no tempo de permanência.
3. O sistema atualiza os dados cadastrais do veículo.

MANTER TABELA DE PREÇOS	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a manutenção da tabela de preços utilizada para calcular a permanência no estacionamento.
<b>Ator:</b>	Atendente

**Cenário Principal:**

1. O sistema busca e exibe os valores para as seguintes informações:
  - 1.1. dia da semana
  - 1.2. valor da primeira hora

1.3. valor da hora subsequente

1.4. se no dia é preço único

GERAR RELATÓRIO DE FATURAMENTO DIÁRIO	
<b>Descrição:</b>	Este caso de uso tem por objetivo emitir um relatório com o faturamento diário do estacionamento.
<b>Ator:</b>	Diretoria

**Cenário Principal:**

1. O sistema prepara uma lista de todas as vagas ocupadas no dia.
2. O sistema exibe:
  - 2.1. placa do carro
  - 2.2. tempo de permanência
  - 2.3. valor pago
3. No final, o sistema exibe o total de valor recebido no dia.

GERAR RELATÓRIO DE FATURAMENTO MENSAL	
<b>Descrição:</b>	Este caso de uso tem por objetivo emitir um relatório com o faturamento mensal do estacionamento.
<b>Ator:</b>	Diretoria

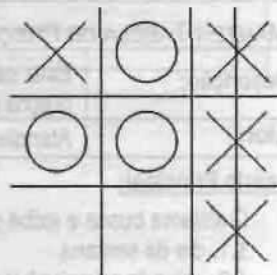
**Cenário Principal:**

1. O sistema busca todas as vagas ocupadas durante o mês corrente.
2. O sistema exibe, para cada dia, que aparecerá em ordem crescente:
  - 2.1. número de veículos atendidos
  - 2.2. valor faturado no dia

## 42. CENÁRIO: JOGO DA VELHA

Amélia gosta de desenvolver pequenos jogos para os seus filhos. O último que ela desenvolveu foi o jogo da velha.

Para isso, ela relacionou as regras do jogo a fim de facilitar sua implementação. Haverá sempre um jogador virtual que será o próprio sistema, na figura do computador. O jogador real poderá escolher o nome do seu adversário. Além disso, o jogador real escolhe o símbolo que deseja usar (círculo ou cruz).

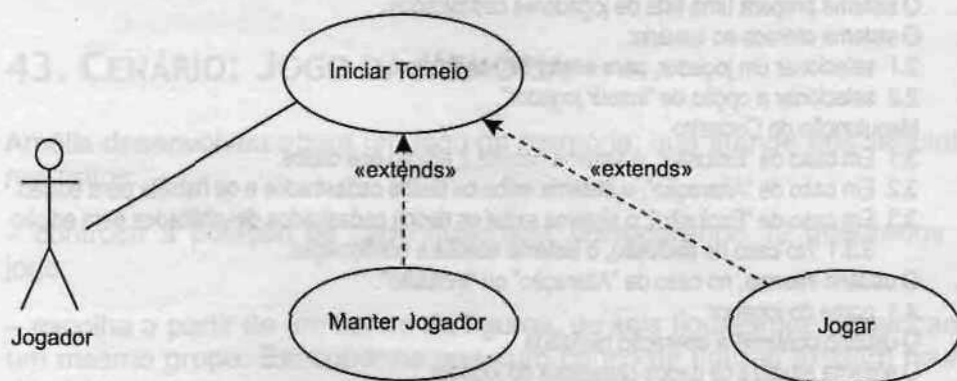




As regras são:

- Traçam-se duas linhas verticais e duas linhas horizontais, de modo a criar nove quadrados de tamanhos iguais.
- Um jogador joga desenhando um círculo e o seu adversário uma cruz.
- O objetivo é cada jogador desenhando seu símbolo num quadrado vazio, tentando marcar uma linha de três símbolos iguais na horizontal, vertical ou diagonal; ou evitar que o adversário consiga essa linha de símbolos.

### RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):



INICIAR TORNEIO	
<b>Descrição:</b>	Este caso de uso tem por objetivo dar início a um torneio de jogos da velha.
<b>Ator:</b>	Jogador

#### Cenário Principal:

1. O sistema prepara a lista de todos os jogadores cadastrados.
2. O usuário informa:
  - 2.1. o nome do jogador físico, selecionado de uma lista preexistente.
  - 2.2. o símbolo usado pelo jogador físico (círculo ou cruz).
  - 2.3. o nome do jogador virtual, selecionado de uma lista preexistente.
  - 2.4. o sistema associa automaticamente o símbolo usado pelo jogador virtual (círculo ou cruz).
3. O sistema define o jogador 1 como sendo o jogador físico e zera seu score.
4. O sistema define o jogador 2 como sendo o jogador virtual e zera seu score.
5. Para cada partida, alternadamente, jogador físico e virtual fazem a sua jogada.
  - 5.1. Extends [Caso de Uso Jogar]

**Cenário Alternativo:****- Jogador não cadastrado**

Se o nome do jogador não tiver cadastrado, o usuário poderá fazer o cadastramento na hora. Extends [Caso de Uso Manter Jogador]

MANTER JOGADOR	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a manutenção do jogador que participa de um torneio.
<b>Ator:</b>	Jogador

**Cenário Principal:**

1. O sistema prepara uma lista de jogadores cadastrados.
2. O sistema oferece ao usuário:
  - 2.1 selecionar um jogador, para alterar seu cadastro;
  - 2.2 selecionar a opção de "inserir jogador".
3. Manutenção do Cadastro
  - 3.1 Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 3.2 Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 3.3 Em caso de "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 3.3.1 No caso de exclusão, o sistema solicita a confirmação.
4. O usuário informa, no caso de "Alteração" ou "Inclusão":
  - 4.1 nome do jogador
5. O usuário confirma a operação realizada.
6. O sistema atualiza os dados cadastrais do jogador.
  - 6.1 No caso de inclusão, o sistema gera automaticamente um código de identificação.

JOGAR	
<b>Descrição:</b>	Este caso de uso tem por objetivo receber e controlar a jogada de um jogador virtual ou físico.
<b>Ator:</b>	Jogador

**Pré-condição:**

Receber a identificação do último vencedor.

**Cenário Principal:**

1. O sistema define de quem é a vez da jogada.
  - 1.1. Se o último vencedor tiver sido o jogador virtual ou for a primeira partida, a vez é do jogador físico. Caso contrário, a vez é do jogador virtual.
2. Enquanto existir quadrado livre e nenhum dos jogadores ganhou o jogo:
  - 2.1. Se a vez for do jogador físico:
    - 2.1.1. O usuário informa a posição em que colocará o seu símbolo.
  - 2.2. Se a vez for do jogador virtual, somente uma dessas operações será feita, na ordem:

- 2.2.1. Se o centro estiver desocupado, o sistema coloca o seu símbolo no centro.
  - 2.2.2. Se o jogador físico possuir duas figuras numa mesma linha horizontal, vertical ou diagonal, onde exista um espaço vago, o sistema colocará o seu símbolo no espaço desocupado.
  - 2.2.3. Se houver canto livre, o sistema colocará o seu símbolo no primeiro canto livre desocupado.
  - 2.2.4. Em caso contrário, o sistema colocará o seu símbolo na primeira posição vaga que encontrar.
- 2.3. O sistema verifica se houve a marcação de três figuras iguais, horizontalmente, verticalmente ou diagonalmente. Se houver, o jogador que possua a referida figura é determinado como vencedor.

**Pós-condição:**

Notificar o vencedor da partida.

## 43. CENÁRIO: JOGO DA MEMÓRIA

Amélia desenvolveu agora um jogo da memória, que atende aos seguintes requisitos:

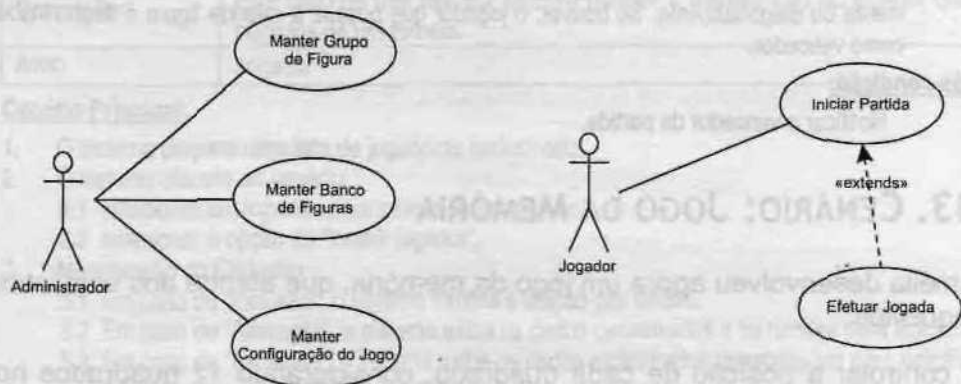
- controlar a posição de cada quadrado, considerando 12 quadrados no jogo;
- escolha a partir de um banco de figuras, de seis figuras que pertençam a um mesmo grupo. Ex: suponha que num banco de figuras existam figuras de objetos, animais, objetos geométricos etc. Ao montar o jogo da memória, são obtidas seis figuras de objetos ou seis figuras de animais;
- sorteio aleatório. A cada dois números associar a um par. Guardar essas coordenadas, associadas à figura.

O jogo consiste no jogador tentar adivinhar a posição de todos os pares. A cada jogada, ele seleciona uma posição. A aplicação revela a figura que está nessa posição e fica esperando a próxima jogada. Ao selecionar a próxima posição, a aplicação exibe a figura. Se for a figura-par, conta-se uma pontuação (conforme a regra de pontuação) e as figuras ficam à mostra. Caso contrário, as figuras são escondidas e espera-se uma nova rodada.

Quando o jogador conseguir desvendar todos os pares, o programa guarda a pontuação como sendo o último score e verifica se esse foi o maior score do jogo, substituindo este último se for o caso. Questiona-se, nesse caso, o nome do jogador para registrar.

A regra da pontuação é a seguinte: cada acerto vale 100 pontos. A cada rodada errada, a pontuação a ser creditada diminui em 5 pontos, limitado a 10 pontos de crédito. Exemplo: se o jogador acertar um par na primeira jogada, ele recebe 100 pontos. Na segunda jogada, ele erra o par. Na terceira jogada, ele acerta o par; então receberá 95 pontos.

### RESOLUÇÃO (DIAGRAMA DE CASOS DE USO E DESCRIÇÃO DOS CENÁRIOS):



MONTAR GRUPO DE FIGURA	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a manutenção dos grupos de figura do jogo da memória.
<b>Ator:</b>	Administrador

#### Cenário Principal:

- O sistema prepara uma lista de grupos de figuras cadastrados.
- O sistema oferece ao usuário:
  - selecionar um grupo, para alterar seu cadastro;
  - selecionar a opção de "inserir grupo".
- Manutenção do Cadastro
  - Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - No caso de exclusão, o sistema solicita a confirmação.
- O usuário informa, no caso de "Alteração" ou "Inclusão":
  - nome do grupo
- O usuário confirma a operação realizada.
- O sistema atualiza os dados cadastrais do grupo.
  - No caso de inclusão, o sistema gera automaticamente um código de identificação.

**Cenário Alternativo:****– Exclusão não permitida**

Não é possível excluir um grupo que esteja associado a uma figura.

MANTER BANCO DE FIGURAS	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a manutenção das figuras que participarão do jogo da velha.
<b>Ator:</b>	Jogador

**Cenário Principal:**

1. O sistema prepara uma lista de figuras cadastradas.
2. O sistema oferece ao usuário:
  - 2.1. selecionar uma figura, para alterar seu cadastro;
  - 2.2. selecionar a opção de "inserir figura".
1. Manutenção do Cadastro
  - 1.1. Em caso de "Inclusão", o sistema habilita a edição dos dados.
  - 1.2. Em caso de "Alteração", o sistema exibe os dados cadastrados e os habilita para edição.
  - 1.3. Em caso de "Consulta" ou "Exclusão", o sistema exibe os dados cadastrados desabilitados para edição.
    - 1.3.1. No caso de exclusão, o sistema solicita a confirmação.
2. No caso de "Alteração" ou "Inclusão":
  - 2.1. O sistema prepara uma lista dos grupos de figura.
  - 2.2. O usuário informa:
    - 2.2.1. grupo ao qual pertence a figura, selecionado de uma lista preexistente
    - 2.2.2. nome da figura
    - 2.2.3. nome físico da figura
      - 2.2.3.1. o sistema carrega e exibe a imagem da figura
3. O usuário confirma a operação realizada.
4. O sistema atualiza os dados cadastrais da figura.
  - 4.1. No caso de inclusão, o sistema gera automaticamente um código de identificação.

MANTER CONFIGURAÇÃO DO JOGO	
<b>Descrição:</b>	Este caso de uso tem por objetivo permitir a manutenção das configurações do jogo da memória.
<b>Ator:</b>	Jogador

**Cenário Principal:**

1. O sistema busca e exibe os valores para alteração do usuário:
  - 1.1. quantidade de pontos para cada acerto.
  - 1.2. quantidade de pontos perdidos para cada erro.
  - 1.3. quantidade de pontos do jogador atual.
2. O usuário pode solicitar a limpeza dos dados referentes ao score máximo do jogo.



INICIAR PARTIDA	
<b>Descrição:</b>	Este caso de uso tem por objetivo dar início a uma partida.
<b>Ator:</b>	Jogador

**Cenário Principal:**

1. O sistema escolhe aleatoriamente um grupo de figuras.
2. Para o grupo de figuras escolhido, o sistema obtém a lista de figuras pertencentes àquele grupo.
3. O usuário escolhe seis figuras da lista preexistente.
4. O sistema duplica as figuras e sorteia a coordenada das figuras dentro do jogo.
  - 4.1. O sistema guarda a posição de todas as figuras no jogo.
  - 4.2. O sistema aguarda as jogadas. Extends [Caso de Uso Efetuar Jogada]
  - 4.3. O sistema obtém o *score* máximo já alcançado.
  - 4.4. O sistema guarda e exibe o *score* do jogador.
    - 4.4.1. O sistema verifica se o *score* do jogador é maior que o *score* máximo já alcançado. Se for, o usuário informa o seu nome e o *score* do jogador substitui o *score* máximo já alcançado.

EFETUAR JOGADA	
<b>Descrição:</b>	Este caso de uso tem por objetivo receber e controlar uma jogada para tentar descobrir um par no jogo da memória.
<b>Ator:</b>	Jogador

**Cenário Principal:**

1. O sistema verifica a pontuação para cada acerto e o decréscimo para cada erro.
2. Enquanto houver par coberto:
  - 2.1. O usuário escolhe um local que esteja coberto.
  - 2.2. O sistema exibe a figura do local escolhido.
  - 2.3. O usuário escolhe um outro local que esteja coberto.
  - 2.4. O sistema exibe a figura do local escolhido.
  - 2.5. Se as figuras forem iguais:
    - 2.5.1. O sistema acrescenta no *score* do jogador a pontuação de acerto.
  - 2.6. Se as figuras não forem iguais:
    - 2.6.1. O sistema cobre as figuras novamente.
    - 2.6.2. O sistema decremente da pontuação para cada acerto o valor do decréscimo por erro, limitado ao valor de 10 pontos.

**Pós-condição:**

Retornar o *score* do jogador, após término da partida.

## Exercitando a Identificação de Classes a partir de Casos de Uso

Neste capítulo reaproveitaremos alguns cenários do Capítulo 3. O objetivo é, a partir da análise de um conjunto de casos de uso, desenhar um modelo de classes.

Exercitamos nos primeiros capítulos o desenho de um modelo de classes, a partir de cenários de baixa complexidade. Entretanto, ao aumentar a complexidade de um cenário, faz-se necessária a análise do caso de uso, a fim de obtermos informações mais precisas para modelar as classes da aplicação.

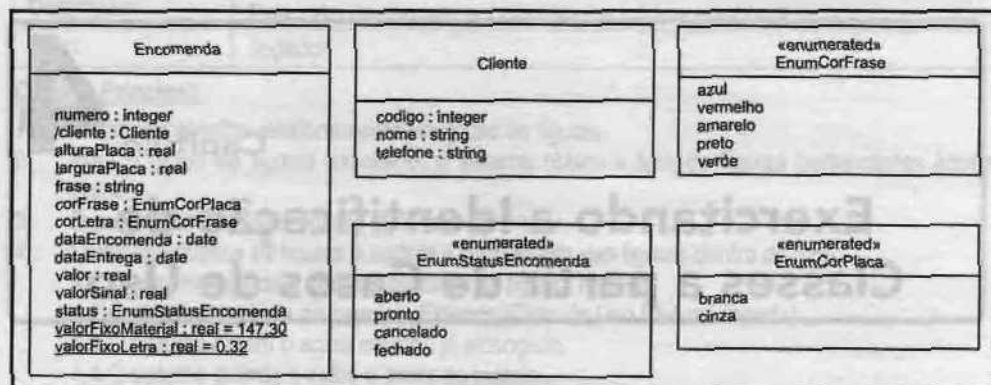
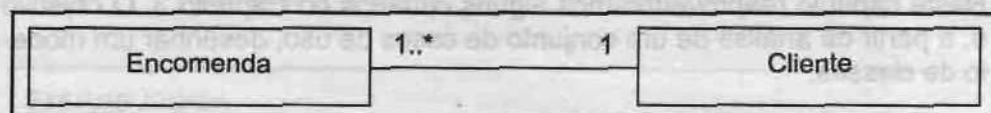
### 44. CENÁRIO: ENCOMENDA DE PLACAS

#### EXERCÍCIO (A):

A partir do cenário descrito no **Exercício 34 do Capítulo 3 (Encomenda de Placas)** e dos casos de uso descritos, identifique as classes, com seus atributos e métodos. Represente os atributos que sejam do tipo objeto.

#### EXERCÍCIO (B):

Utilizando apenas o nome das classes, desenhe o relacionamento entre as classes.

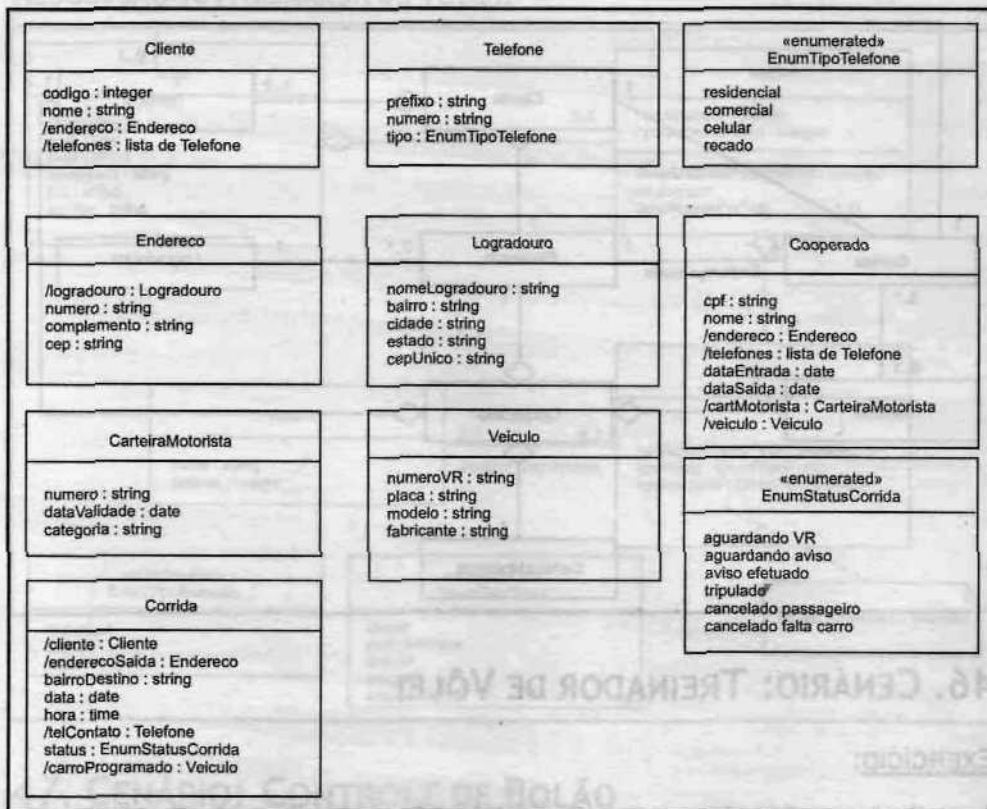
**RESOLUÇÃO 44.A (ENCOMENDA DE PLACAS):****RESOLUÇÃO 44.B (ENCOMENDA DE PLACAS):****45. CENÁRIO: RÁDIO TÁXI MAR & SOL****EXERCÍCIO (A):**

A partir do cenário descrito no **Exercício 35 do Capítulo 3 (Rádio Táxi Mar & Sol)** e dos casos de uso descritos, identifique as classes, com seus atributos e métodos. Represente os atributos que sejam do tipo objeto.

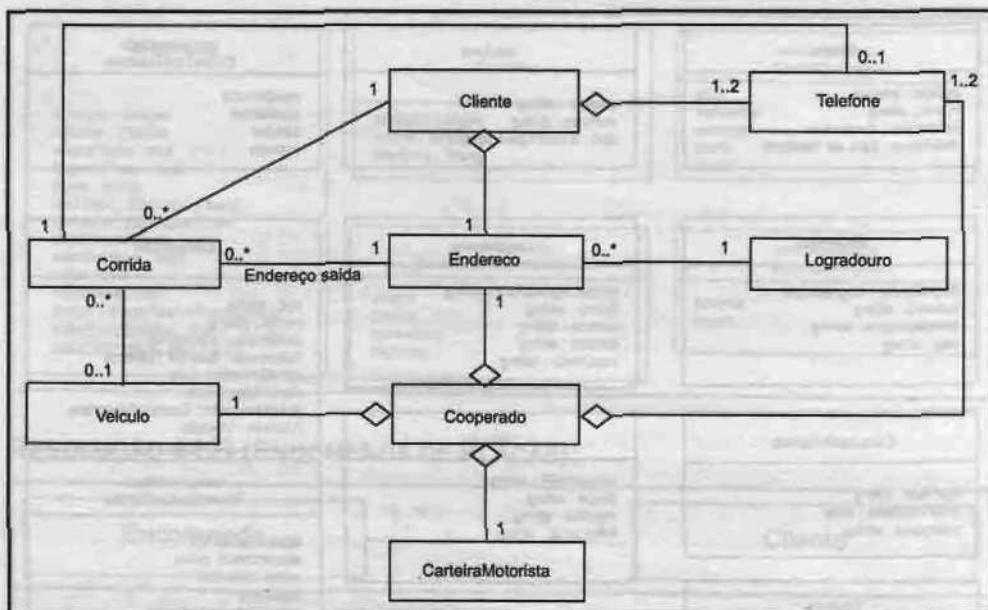
**EXERCÍCIO (B):**

Utilizando apenas o nome das classes, desenhe o relacionamento entre as classes.

## RESOLUÇÃO 45.A (RÁDIO TÁXI):



## RESOLUÇÃO 45.B (RÁDIO TÁXI):



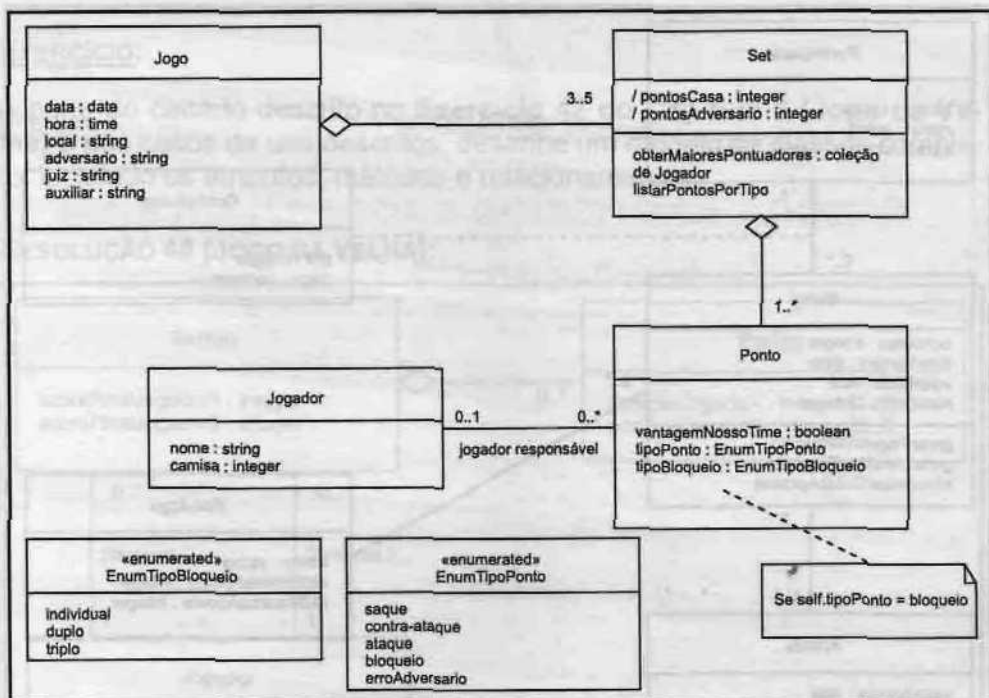
## 46. CENÁRIO: TREINADOR DE VÔLEI

EXERCÍCIO:

A partir do cenário descrito no **Exercício 36 do Capítulo 3 (Treinador de Vôlei)** e dos casos de uso descritos, desenhe um modelo de classes completo, incluindo os atributos, métodos e relacionamentos.



## RESOLUÇÃO 46 (TREINADOR DE VÔLEI):

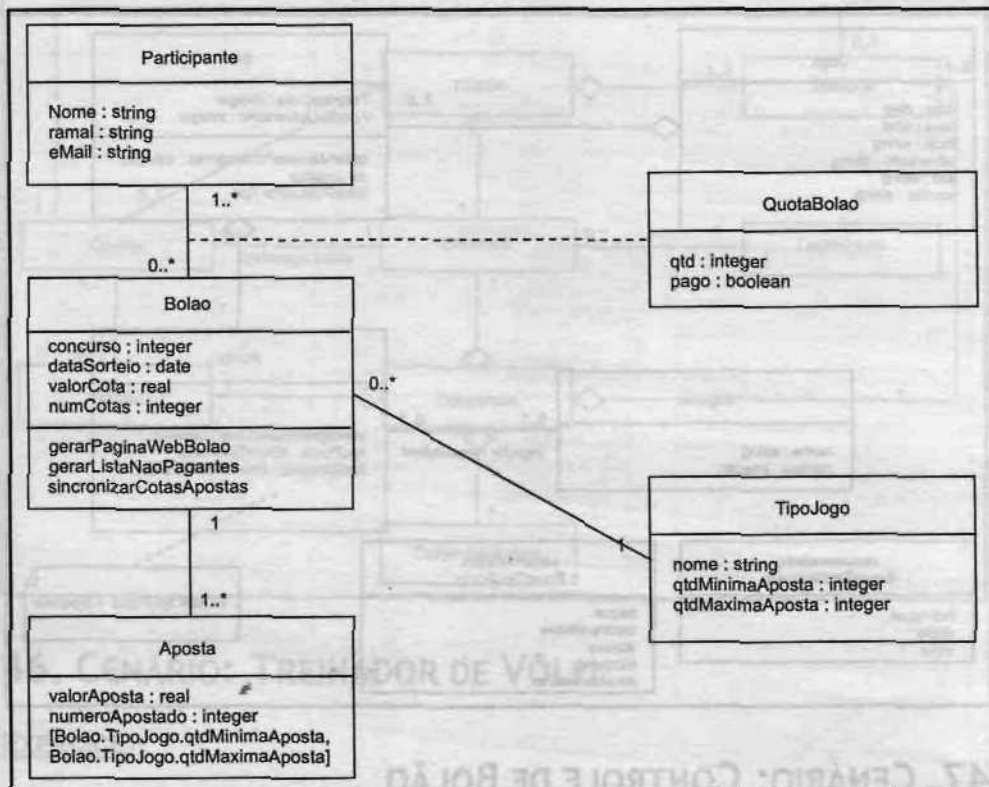


## 47. CENÁRIO: CONTROLE DE BOLÃO

EXERCÍCIO:

A partir do cenário descrito no **Exercício 40 do Capítulo 3 (Controle de Bolão)** e dos casos de uso descritos, desenhe um modelo de classes completo, incluindo os atributos, métodos e relacionamentos.

## RESOLUÇÃO 47 (CONTROLE DE BOLÃO):

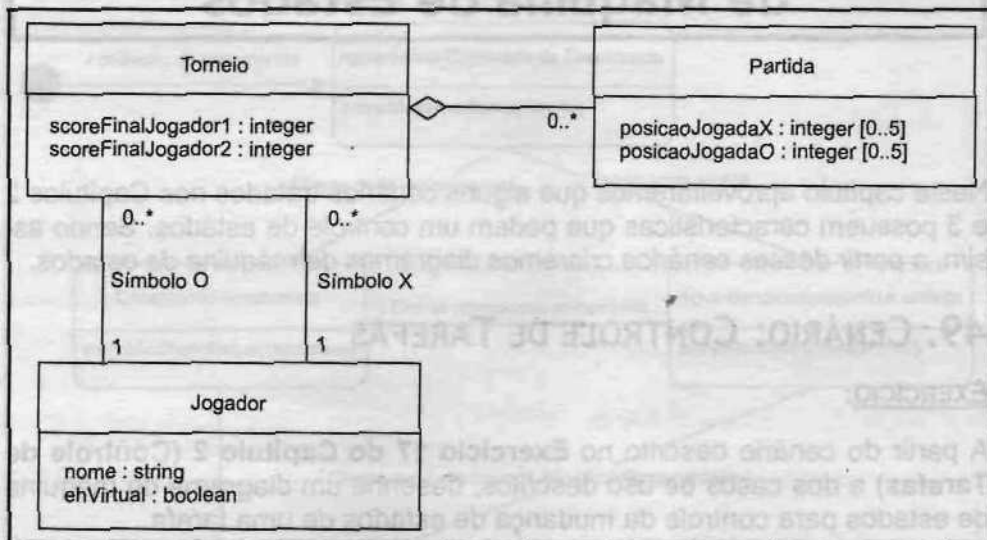


## 48. CENÁRIO: JOGO DA VELHA

### EXERCÍCIO:

A partir do cenário descrito no **Exercício 42 do Capítulo 3 (Jogo da Velha)** e dos casos de uso descritos, desenhe um modelo de classes completo, incluindo os atributos, métodos e relacionamentos.

### RESOLUÇÃO 48 (JOGO DA VELHA):



## Alguns Exercícios sobre Diagrama de Máquina de Estados

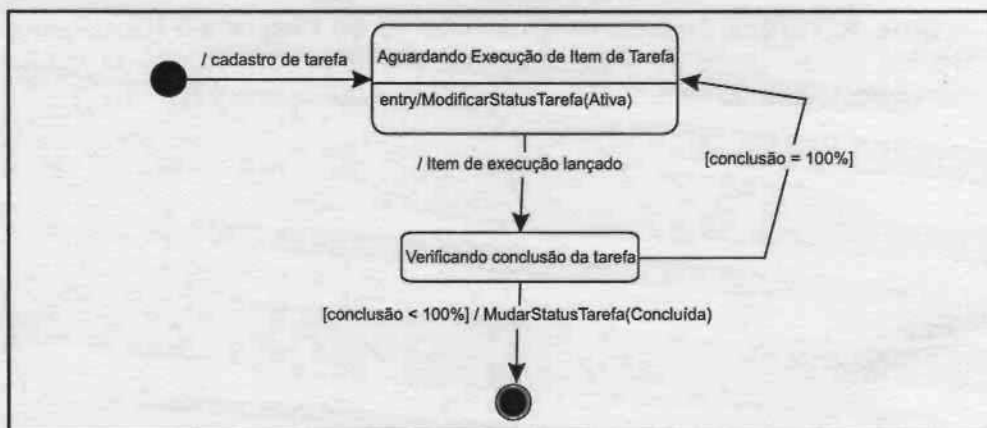
Neste capítulo aproveitaremos que alguns cenários tratados nos Capítulos 2 e 3 possuem características que pedem um controle de estados. Sendo assim, a partir desses cenários criaremos diagramas de máquina de estados.

### 49. CENÁRIO: CONTROLE DE TAREFAS

#### EXERCÍCIO:

A partir do cenário descrito no **Exercício 17 do Capítulo 2 (Controle de Tarefas)** e dos casos de uso descritos, desenhe um diagrama de máquina de estados para controle da mudança de estados de uma tarefa.

#### RESOLUÇÃO 49 (CONTROLE DE TAREFAS):

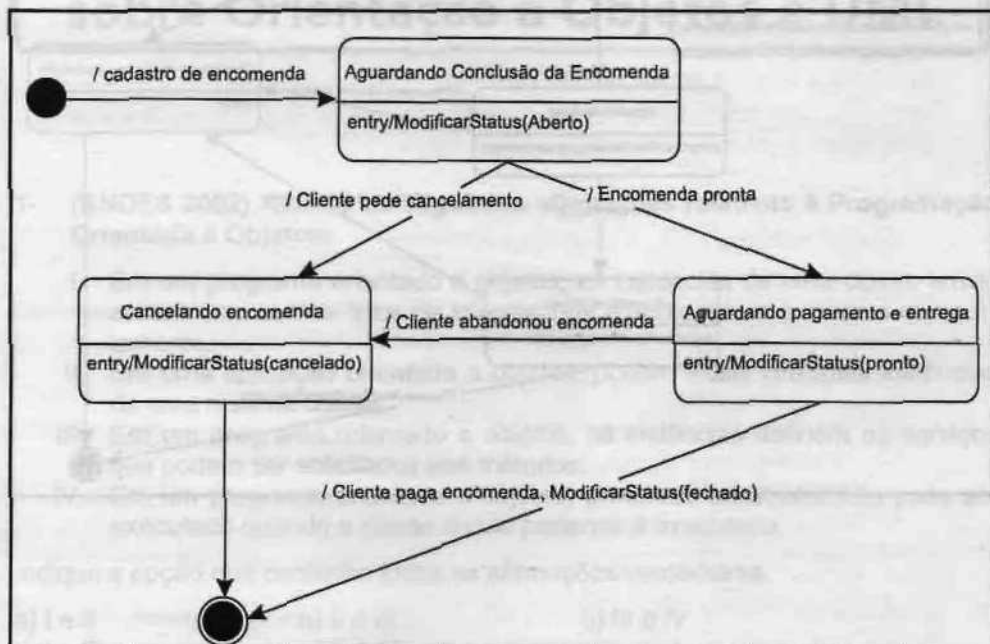


## 50. CENÁRIO: ENCOMENDA DE PLACAS

### EXERCÍCIO:

A partir do cenário descrito no **Exercício 34 do Capítulo 3 (Encomenda de Placas)** e dos casos de uso descritos, desenhe um diagrama de máquina de estados para controle da mudança de estados de uma tarefa.

### RESOLUÇÃO 50 (ENCOMENDA DE PLACAS):



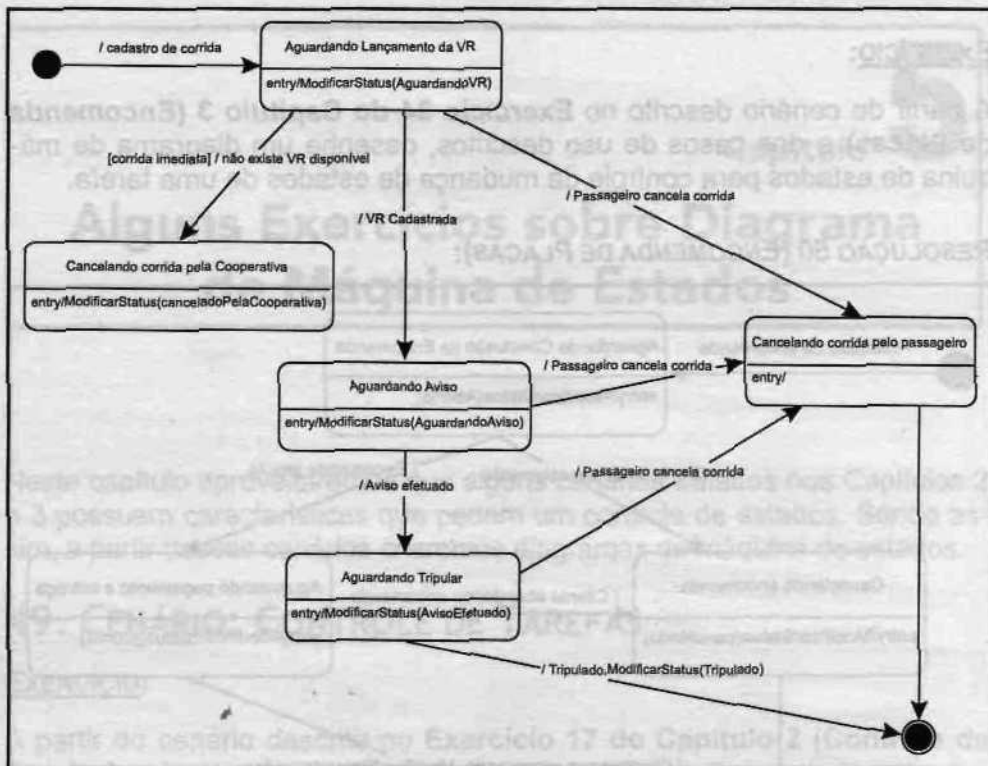
## 51. CENÁRIO: RÁDIO TÁXI MAR & SOL

### EXERCÍCIO:

A partir do cenário descrito no **Exercício 35 do Capítulo 3 (Rádio Táxi Mar & Sol)** e dos casos de uso descritos, desenhe um diagrama de máquina de estados para controle da mudança de estados de uma tarefa.



## RESOLUÇÃO 51 (RÁDIO TÁXI MAR &amp; SOL):



## Questões de Concursos Públicos sobre Orientação a Objetos e UML

**1- (BNDES 2002) Analise as seguintes afirmações relativas à Programação Orientada a Objetos:**

- I. Em um programa orientado a objetos, as instâncias de uma classe armazenam os mesmos tipos de informações e apresentam o mesmo comportamento.
- II. Em uma aplicação orientada a objetos, podem existir múltiplas instâncias de uma mesma classe.
- III. Em um programa orientado a objetos, as instâncias definem os serviços que podem ser solicitados aos métodos.
- IV. Em um programa orientado a objetos, o método construtor não pode ser executado quando a classe à qual pertence é executada.

Indique a opção que contenha todas as afirmações verdadeiras.

- |            |             |             |
|------------|-------------|-------------|
| a) I e II  | b) II e III | c) III e IV |
| d) I e III | e) II e IV  |             |

**2- (TCM) No contexto da Orientação a Objetos, o conjunto de objetos que têm a mesma estrutura e o mesmo comportamento é denominado:**

- |                 |                   |
|-----------------|-------------------|
| a) classe       | b) herança        |
| c) polimorfismo | d) encapsulamento |

**3- (Eletrobrás) Considere as seguintes assertivas sobre o diagrama de classes na notação UML:**

- V. uma classe representa um conjunto de objetos cuja cardinalidade é sempre maior do que um.

- VI. o polimorfismo das operações é uma consequência da relação de herança entre classes.
- VII. se um objeto A está associado a um objeto B e B está associado a mais de um objeto C então, obrigatoriamente, um objeto C estará associado a mais de um objeto A.

As assertivas corretas são:

- a) somente I                      b) somente II                      c) somente III  
d) somente I e II                e) somente II e III.

- 4- (Eletrobrás) Em relação à tecnologia de orientação a objetos, a afirmativa de que o estado de um objeto não deve ser acessado diretamente, mas sim por intermédio de métodos de acesso (ou propriedades) está diretamente relacionada ao conceito de:

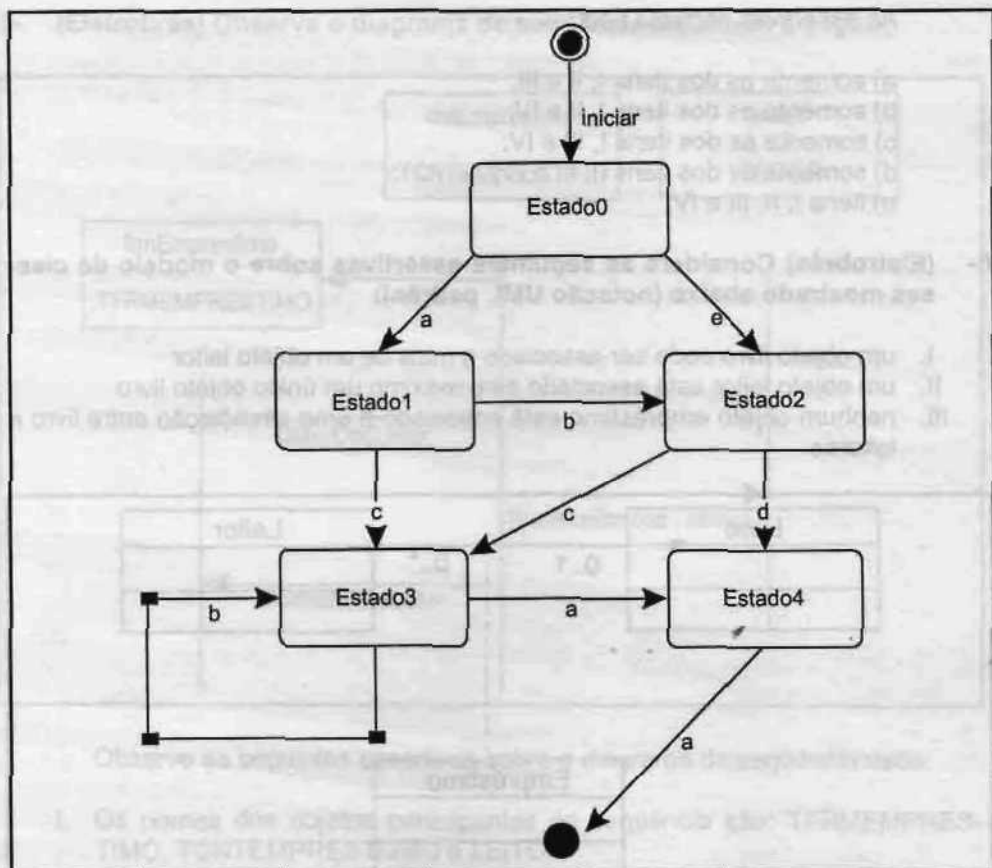
- a) herança                      b) interface                      c) classe  
d) polimorfismo              e) encapsulamento

- 5- (TRE-2001) Generalização / Especialização é um tipo de relacionamento possível de ser aplicado ao(s) seguinte(s) elemento(s) de modelo na UML:

- a) Casos de uso, classes e atores;  
b) Casos de uso e classes;  
c) Classes e atores;  
d) Casos de uso e atores;  
e) Classes.

- 6- (Eletrobrás) Observe o seguinte diagrama de transição de estados. Suponha que o sistema se encontra no estado Estado0, e que ocorra a seguinte sequência de eventos:

a b c b b b c a



O estado do sistema após a ocorrência destes eventos é:

- |            |            |            |
|------------|------------|------------|
| a) Estado0 | b) Estado1 | c) Estado2 |
| d) Estado3 | e) Estado4 |            |

7- (Eletrobrás) Modelo de Casos de Uso: Considere as seguintes assertivas sobre casos de uso:

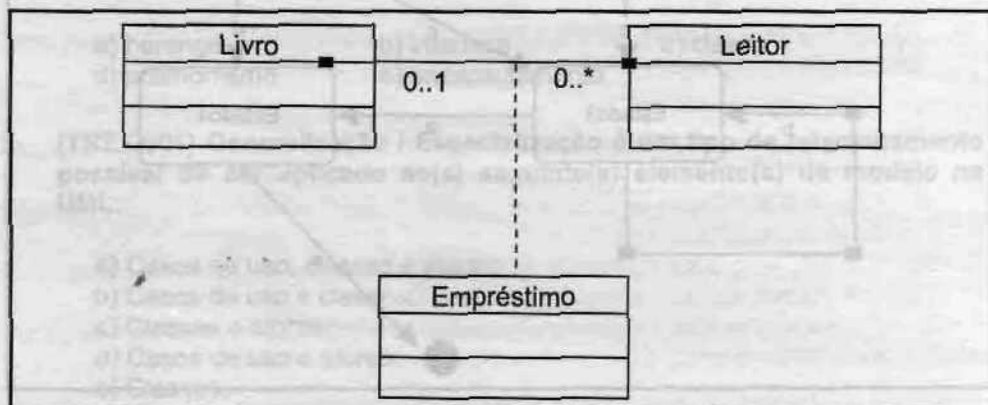
- I. descrevem uma sequência de passos iniciada por um ator.
- II. o estado em que o mundo deve se encontrar para que o caso possa ser executado é chamado de pré-condição
- III. o caso de uso não descreve as respostas do sistema aos estímulos enviados pelo ator
- IV. a recuperação de falhas na execução de um passo é descrita nas exceções

As assertivas corretas são:

- a) somente as dos itens I, II e III;
- b) somente as dos itens I, II e IV;
- c) somente as dos itens I, III e IV;
- d) somente as dos itens II, III e IV;
- e) itens I, II, III e IV;

**8- (Eletrobrás) Considere as seguintes assertivas sobre o modelo de classes mostrado abaixo (notação UML padrão):**

- I. um objeto livro pode ser associado a mais de um objeto leitor
- II. um objeto leitor está associado a no máximo um único objeto livro
- III. nenhum objeto empréstimo está associado a uma associação entre livro e leitores

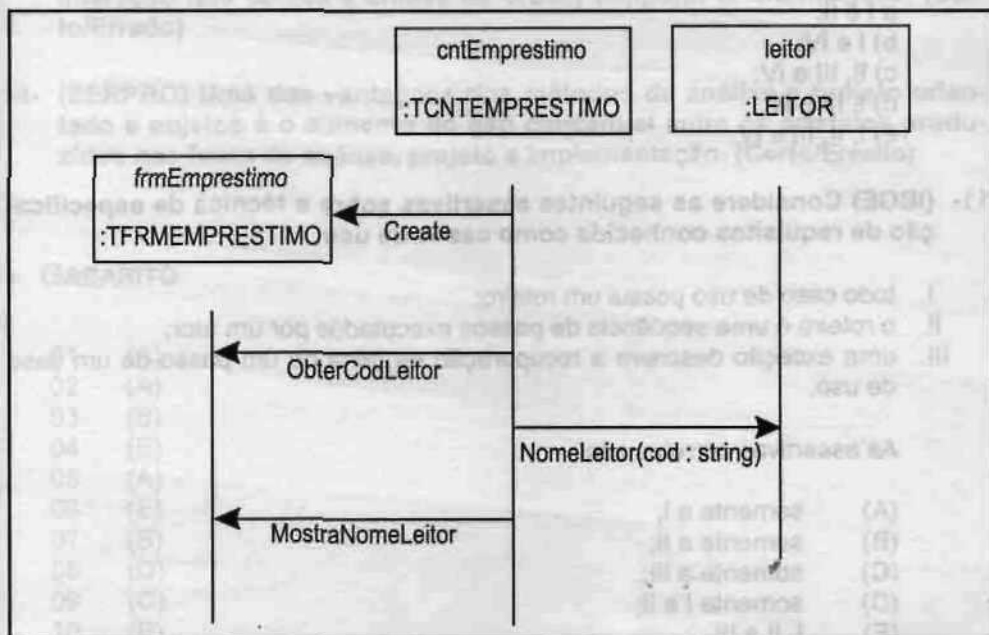


As assertivas corretas são:

- a) somente o item I;
- b) somente o item II;
- c) somente o item III
- d) somente os itens I e II;
- e) itens I, II e III.



9- (Eletrobrás) Observe o diagrama de seqüência mostrado a seguir.



Observe as seguintes assertivas sobre o diagrama de seqüência dado:

- I. Os nomes dos objetos participantes da seqüência são: TFRMEMPRESTIMO, TCNTEMPRESTIJO e LEITOR
- II. ObterCodLeitor é um serviço de cntEmprestimo
- III. ObterCodLeitor é executado antes de MostraNomeLeitor

As assertivas corretas são:

- a) somente a I
- b) somente a II
- c) somente a III
- d) somente I e II
- e) somente II e III

10- (Infraero) Observe as seguintes assertivas sobre a forma de especificação de requisitos conhecida como Casos de Uso:

- I. Todo caso de uso tem a participação de pelo menos um ator;
- II. Um ator pode participar somente de um caso de uso;
- III. O roteiro de um caso de uso não pode utilizar roteiros de outros casos de uso;
- IV. O roteiro de um caso de uso pode ser alterado por outro caso de uso.

Os atributos corretos são somente:

- a) I e II;
- b) I e IV;
- c) II, III e IV;
- d) I, III e IV;
- e) I, II, III e IV.

11- (IBGE) Considere as seguintes assertivas sobre a técnica de especificação de requisitos conhecida como casos de uso:

- I. todo caso de uso possui um roteiro;
- II. o roteiro é uma seqüência de passos executados por um ator;
- III. uma exceção descreve a recuperação da falha de um passo de um caso de uso.

As assertivas corretas são:

- (A) somente a I;
- (B) somente a II;
- (C) somente a III;
- (D) somente I e II;
- (E) I, II e III.

12- (IBGE) Considere os seguintes elementos da UML:

- I. classes
- II. associações
- III. mensagem
- IV. estado
- V. transição

Os elementos que fazem parte de um diagrama de classes são:

- (A) somente I e II;
- (B) somente I, II e III;
- (C) somente I, III e IV;
- (D) somente II, III e IV;
- (E) somente III, IV e V.

- 13- (TCU) Na UML, um diagrama de colaboração é um tipo de diagrama de interação que coloca a ênfase na ordem temporal de mensagens. (Certo/Errado)
- 14- (SERPRO) Uma das vantagens dos métodos de análise e projeto orientado a objetos é o aumento do gap conceitual entre os artefatos produzidos nas fases de análise, projeto e implementação. (Certo/Errado)

♦ GABARITO

- 01 (A)  
02 (A)  
03 (B)  
04 (E)  
05 (A)  
06 (E)  
07 (B)  
08 (D)  
09 (C)  
10 (B)  
11 (E)  
12 (A)  
13 Errado  
14 Errado

# Exercitando modelagem em UML



## 51 exercícios resolvidos

Após escrever seu primeiro livro, *Desenvolvendo Aplicações com UML* – do conceitual à implementação, atualizá-lo para *Desenvolvendo Aplicações com UML 2.0*, a autora, ao basear-se na experiência em lecionar disciplinas de modelagem de desenvolvimento de sistemas, percebeu que surgia uma nova demanda em sala de aula: um material de apoio prático, que pudesse ser usado para fixação da matéria.

**A**ssim, percebendo essa necessidade como não exclusiva da docência, mas também do aluno que entende a teoria, contudo não consegue sair do ponto zero; e do profissional, que tem a necessidade do exemplo, principalmente quando caminha sozinho numa nova estrada de aprendizagem, surgiu a idéia de "Exercitando modelagem em UML".

Este livro oferece ao leitor um caminho para a aprendizagem, por meio de tentativas e erros, exemplos e exercícios. Para atingir este objetivo, este material está dividido em capítulos, onde cada um avança gradualmente pelos conceitos de orientação a objetos e principais diagramas da UML. No apêndice, como bônus, são oferecidas algumas questões de concursos públicos relacionadas ao tema.



**Ana Cristina de Souza Luiz de Melo**, especialista em Análise e Gerência de Sistemas, é professora do Instituto Politécnico e Pós-graduação da Universidade Estácio de Sá. Trabalha com análise, programação e atividades desde 1987. Presta serviços de consultoria e profere palestras em eventos da área, entre eles, o Congresso Fenasoft, OD e Sepai. Autora do livro *Desenvolvendo Aplicações com UML*, em sua segunda edição, também publicado pela Brasport.

visite nosso site!

[www.brasport.com.br](http://www.brasport.com.br)



ISBN 85-7452-254-6



9 788574 522548