



GABARITOS



Lista Extra - Engenharia de Software

Olá aluno(a)! Espero que você tenha aprendido muito com os exercícios extras. Agora você já pode conferir os gabaritos e caso tenha dúvidas nos envie.

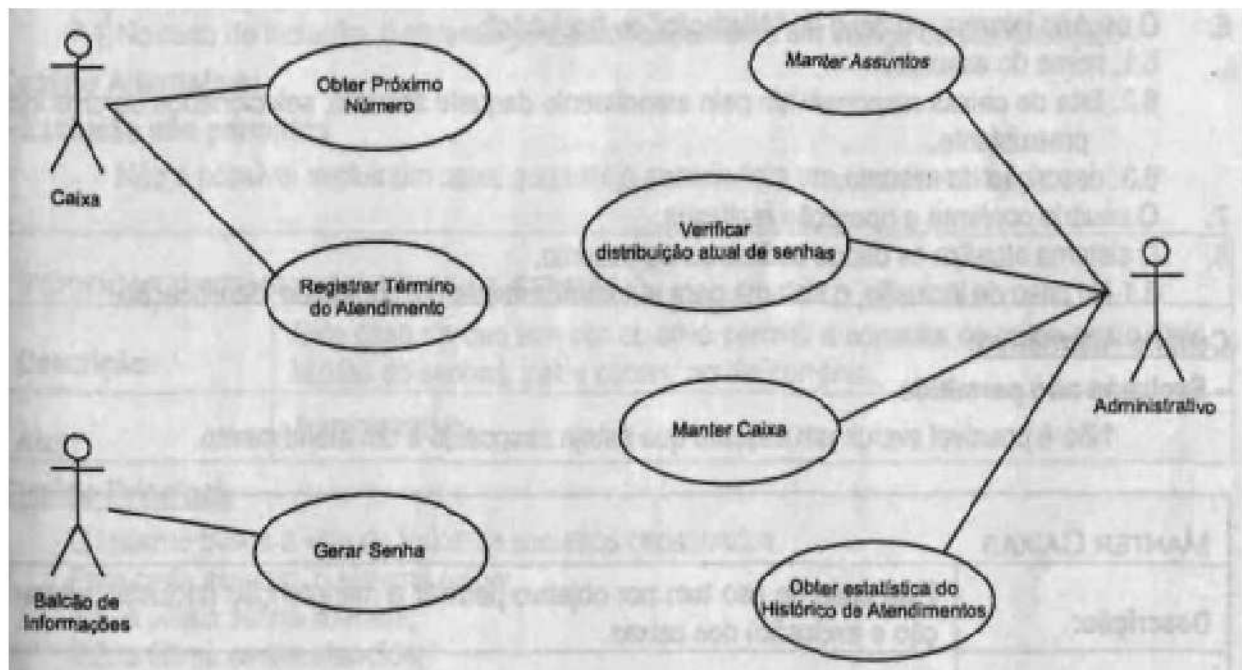
Lembro a você que no final deste documento está disponível uma lista de tutoriais de como instalar e utilizar as ferramentas Astah e StarUML. Estas ferramentas são para a modelagem dos diagramas.

Bons estudos =)

Diagrama de Casos de Uso

Cenário 01: Senha de Atendimento

Orientação de Resposta:

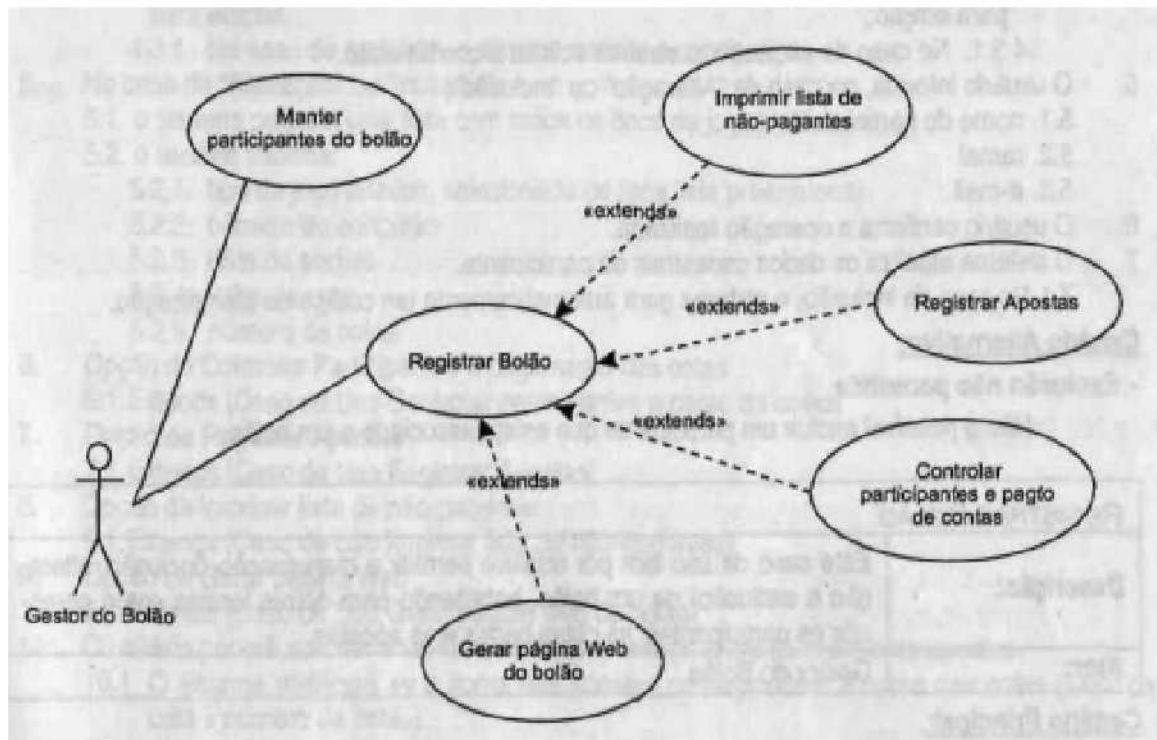


Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Diagrama de Casos de Uso

Cenário 02: Controle de Bolão

Orientação de Resposta:

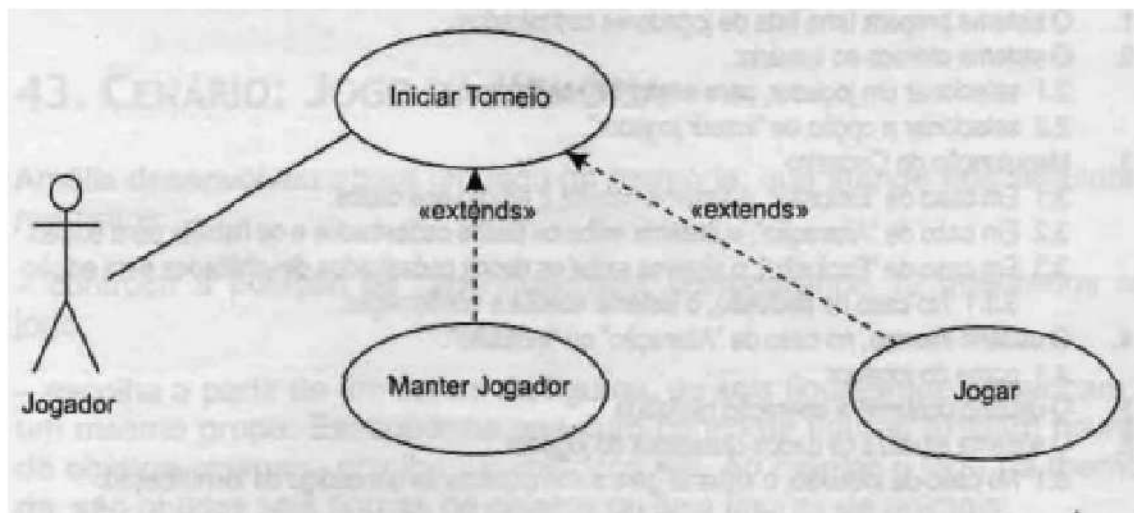


Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Diagrama de Casos de Uso

Cenário 03: Jogo da Velha

Orientação de Resposta:

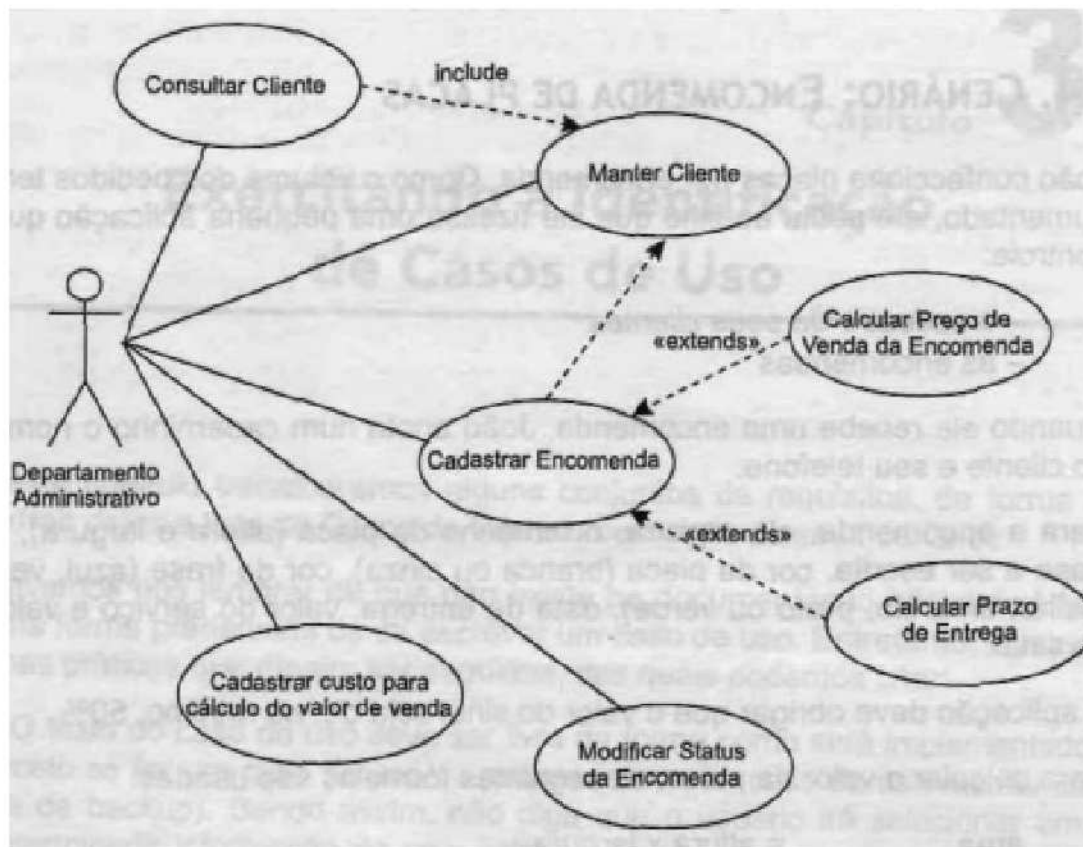


Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Diagrama de Casos de Uso

Cenário 04: Encomenda de Placas

Orientação de Resposta:



Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Diagrama de Classes

Cenário 01: Conta de Luz

Orientação de Resposta:

Classe	Atributos	Métodos
ContaLuz	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).

Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Diagrama de Classes

Cenário 02: Horário de Remédios

Orientação de Resposta:

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

Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

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 tomarem 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.

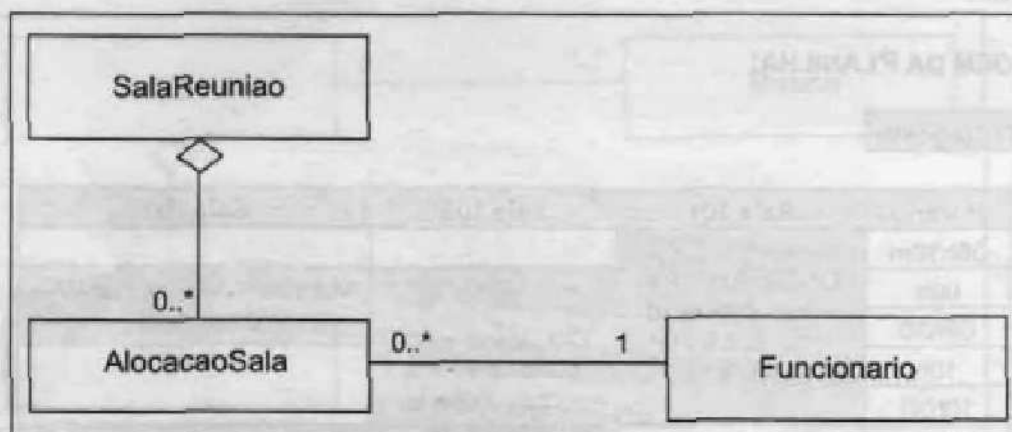
d) A classe HorarioRemedio foi criada incluindo o atributo data, para permitir o cadastramento de horários diferentes para cada dia. Se não fosse por essa abordagem, poderíamos ter criado um atributo horário: time na classe PrescricaoRemedio, com multiplicidade de [1..*], prevendo de uma a várias doses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

Diagrama de Classes
Cenário 03:Sala de Reunião

Orientação de Resposta:

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



Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Diagrama de Classes

Cenário 03:Sala de Reunião

Orientação de Resposta:

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 *AlocacaoSala*, *SalaReuniao* e *Funcionário*.

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 retomar todas as salas de reunião.

Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Diagrama de Classes

Cenário 04: Herança

Orientação de Resposta:

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

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

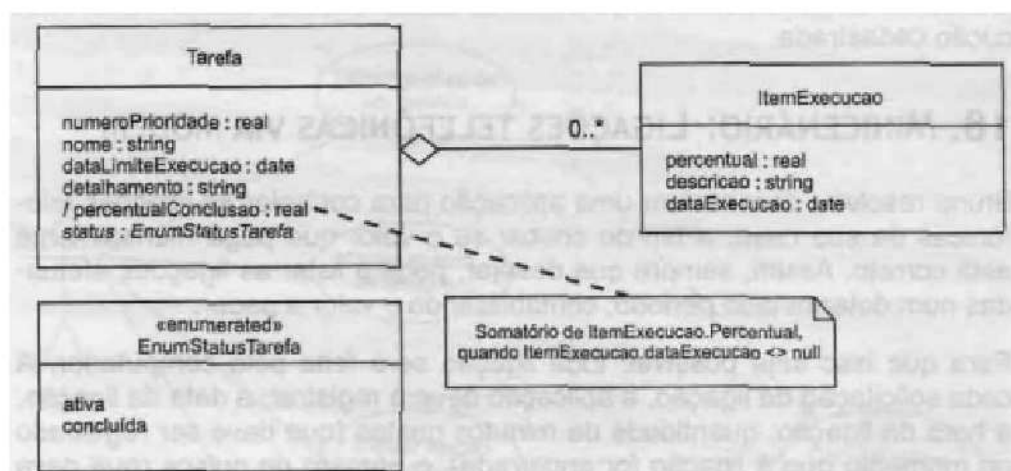
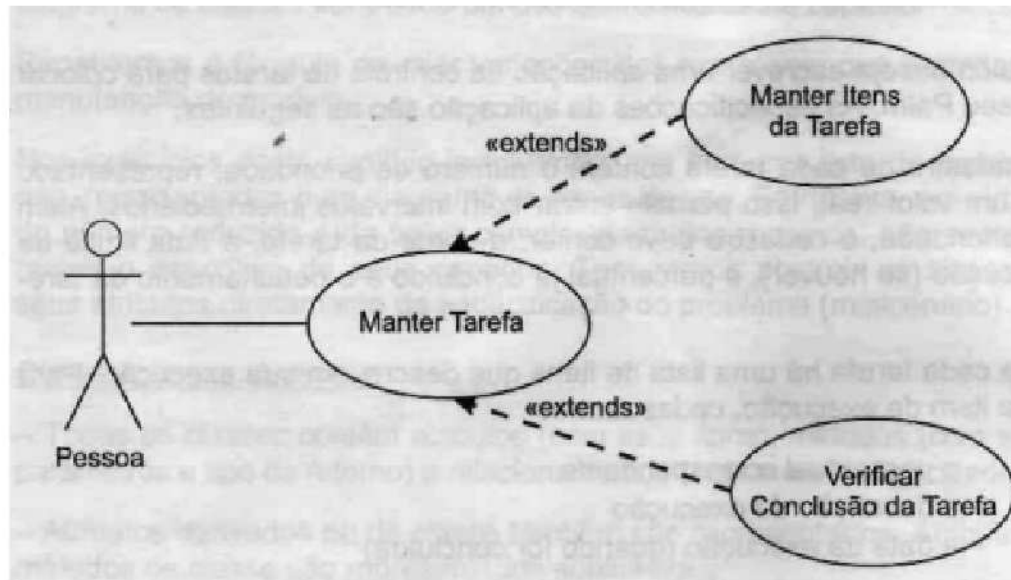
Classe	Atributos	Métodos
Pessoa	nome : string dataNascimento : date /endereco : Classe Endereco /telsContato : Coleção de Telefone	cadastrar obterIdade
Funcionário	matricula : integer /cargo : Classe Cargo salario : real dataAdmissao : date	reajustarSalario (percentual : real) promover(novoCargo : Cargo)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

Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Pratique Mais

Cenário 01: Controle de Tarefas

Orientação de Resposta:



Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Orientação de Resposta:

COMENTÁRIOS SOBRE A RESOLUÇÃO:

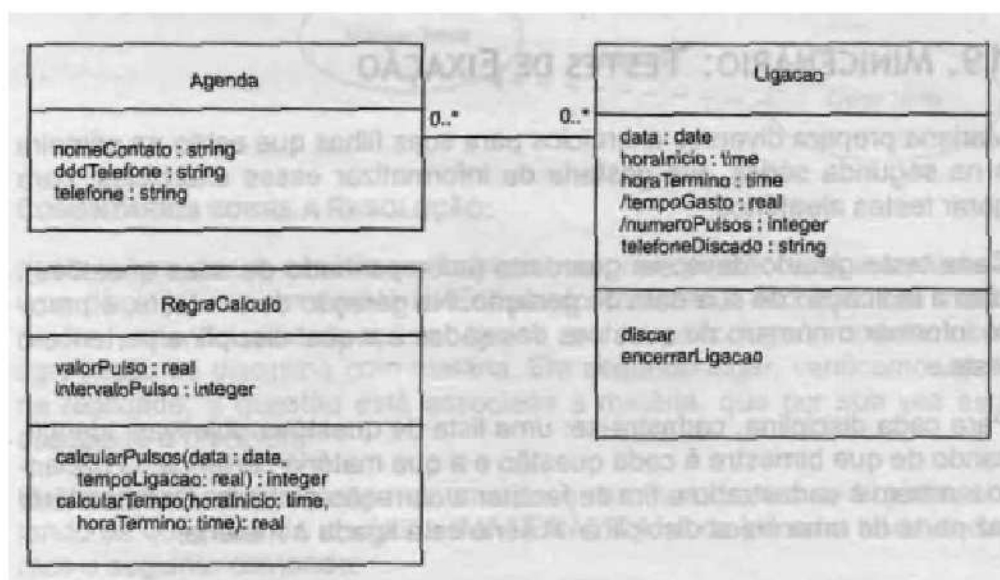
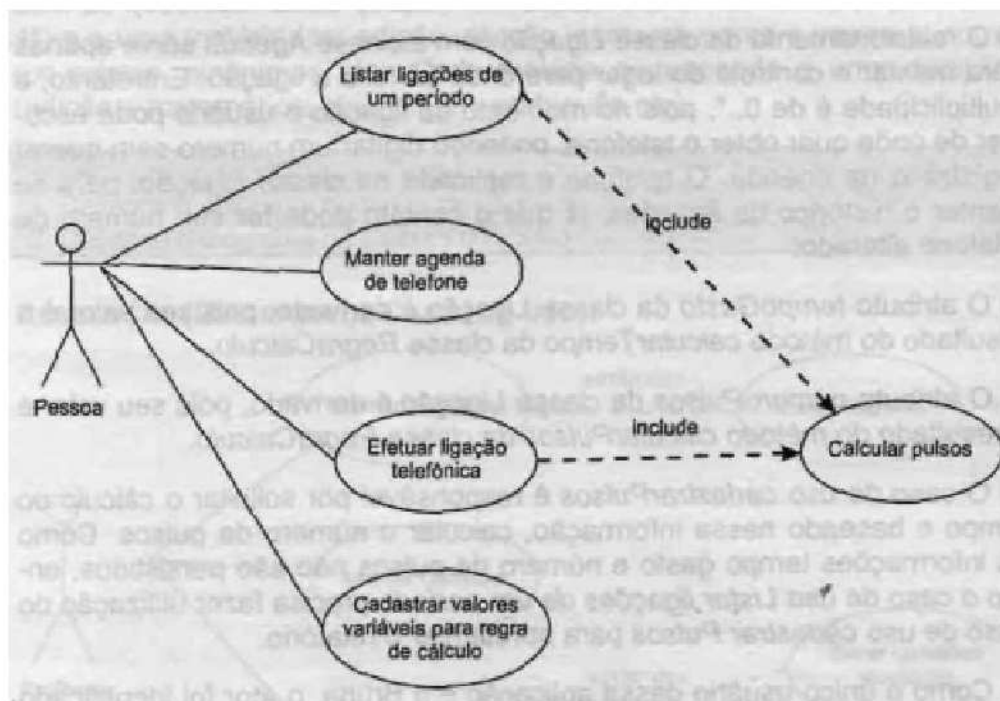
- a) A lista de itens de execução faz parte da tarefa que representa o todo. Desta forma, o relacionamento foi representado como uma agregação.
- b) O atributo `percentualConclusao` não é um atributo que será cadastrado. Seu valor é sempre calculado a partir do somatório dos percentuais dos itens de 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.

Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Pratique Mais

Cenário 02:Ligações Telefônicas via Modem

Orientação de Resposta:



Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Orientação de Resposta:

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 *cadastrarPulsos* para apresentar o relatório.

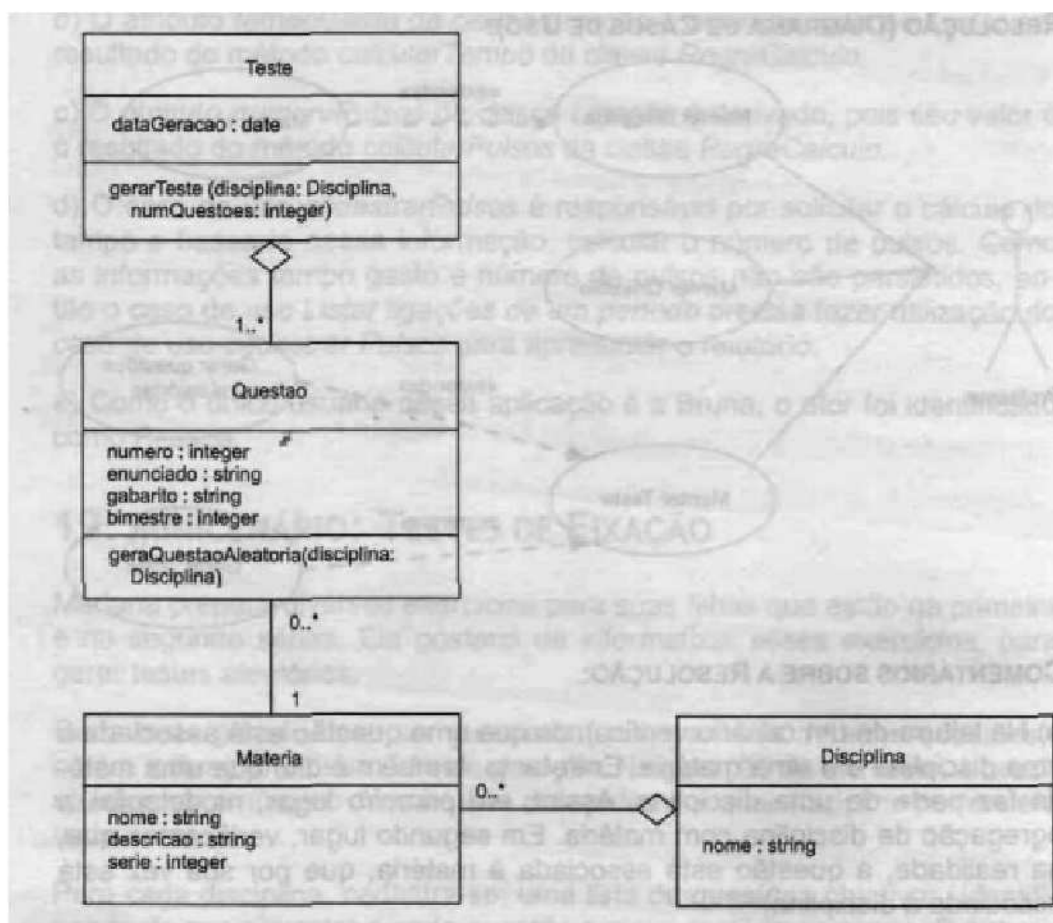
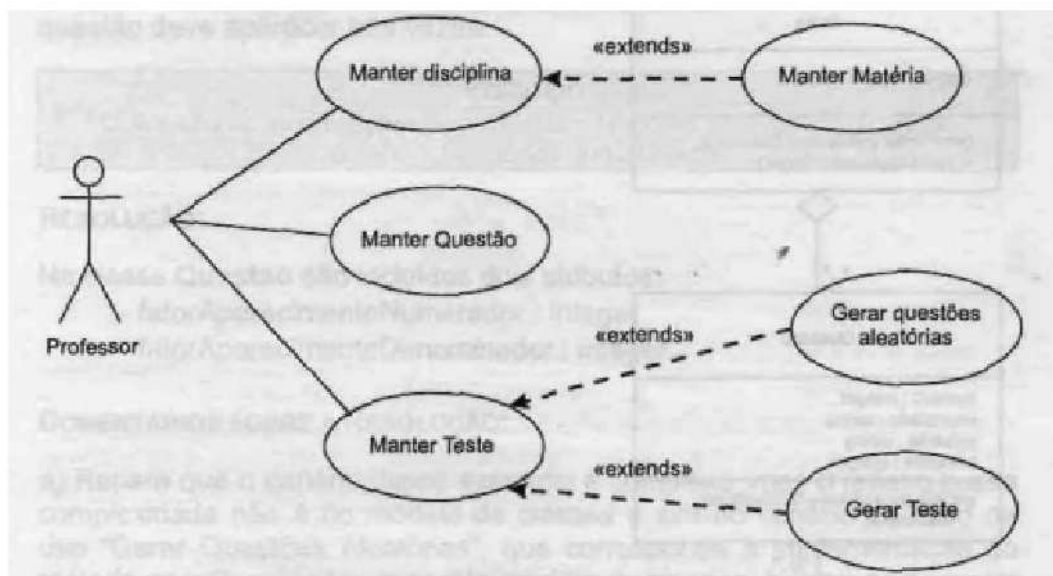
e) Como o único usuário dessa aplicação é a Bruna, o ator foi identificado como *Pessoa*.

Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Pratique Mais

Cenário 03: Testes de Fixação

Orientação de Resposta:



Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

Orientação de Resposta:

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;

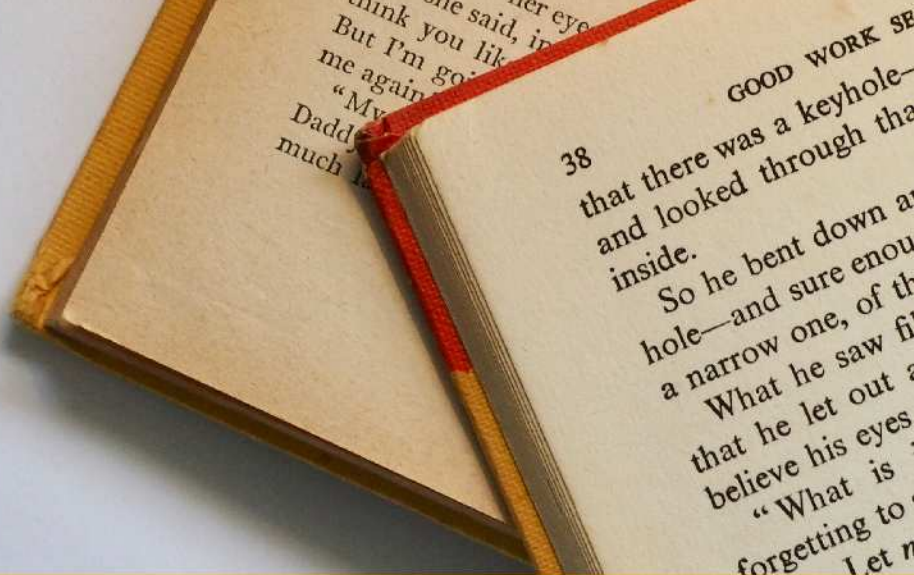
```
SELECT Questão.numero. Questão.enunciado  
FROM Questão  
WHERE Questão.matéria.disciplina.nome = "MATEMÁTICA"  
AND Questão.matéria.serie = 1
```

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.

Fonte: MELLO, C. A. Exercitando Modelagem em UML: 51 exercícios resolvidos. Brasport, 2006.

REFERÊNCIAS

Para melhor embasamento e criação deste material algumas fontes bibliográficas foram utilizadas. Aproveito para lembra-los da importância da pesquisa e leitura contínua para melhor compreensão dos conteúdos. Segue a relação das obras utilizadas:



GUEDES, G. T. A. **UML 2:** uma abordagem prática. São Paulo: Novatec, 2009.

MELLO, C. A. **Exercitando Modelagem em UML:** 51 exercícios resolvidos. Brasport, 2006.

PASCUTTI, M. C. D; FREITAS, J. A; GASPAROTTI, T. T; PEDROSO, V. M. **Engenharia de Software.** Maringá-Pr.: Unicesumar, 2019.



Tutoriais

Para melhor compreensão do uso e instalação das ferramentas de modelagem foram disponibilizados alguns tutoriais:

Astah instalação

<https://youtu.be/xZ95U2eMqGE>

StarUML instalação

<https://youtu.be/2-1dAUR5l6Q>

Modelando com Astah

<https://youtu.be/tmOjdxzGWuA>

Modelando com StarUML

<https://youtu.be/U-l2ODm2Quw>