Problema 2: Nova unidade do SAMU

Ao visitar a sua cidade natal, durante as férias do final de ano, você entrou em contato com um vereador (seu tio) que está super animado com a nova unidade do SAMU (Serviço de Atendimento Móvel de Emergência) que está prestes de ser inaugurada na cidade. Entretanto, ele disse que a unidade requer um software para gerenciar os atendimentos. Muito orgulhoso do sobrinho que está fazendo Engenharia de Computação na UEFS, ele pergunta se você não seria capaz de desenvolver esse sistema para a prefeitura. Ele vai logo dizendo: "você sabe que os nossos recursos são poucos, mas você não negaria um pedido ao seu tio querido, não é mesmo?".

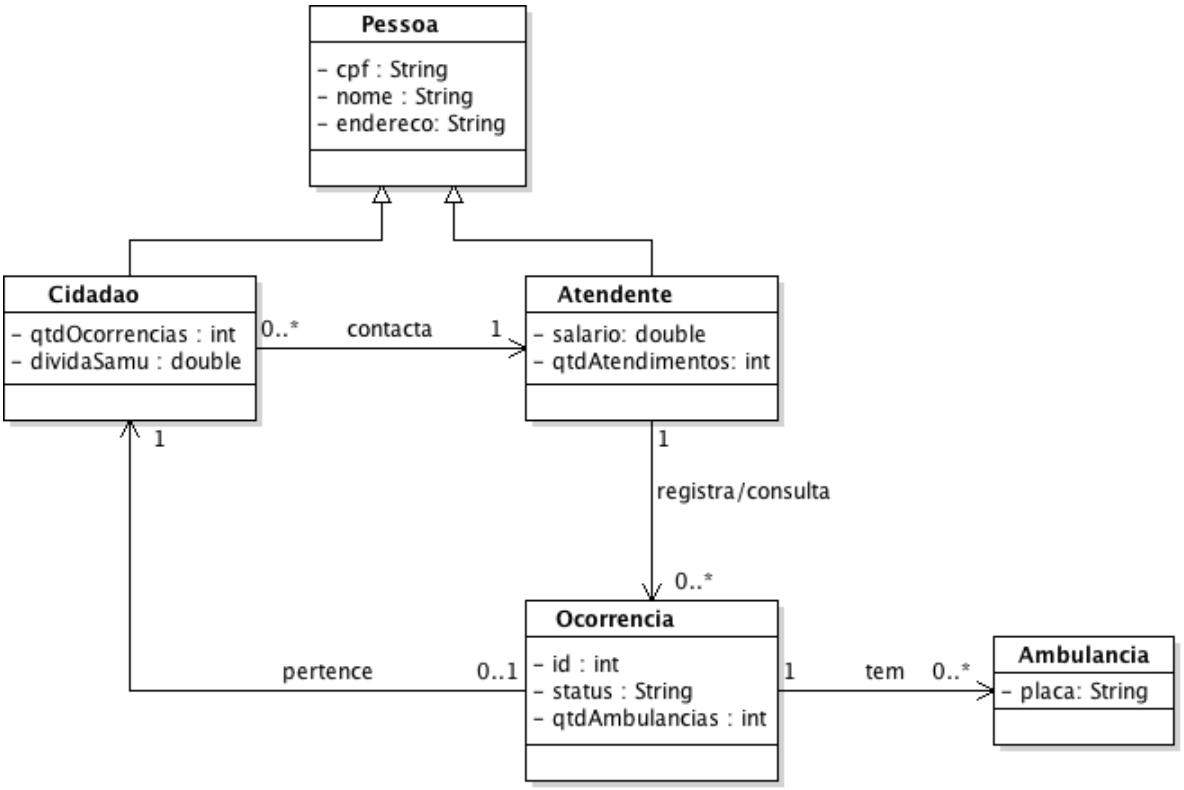
Se sentindo intimado pelo tio a resolver o problema, você acaba topando. O seu tio então começa a descrever um histórico sobre a implantação desta unidade de saúde na cidade: "A prefeitura foi contemplada com uma unidade do SAMU a três anos atrás e, por questões financeiras, o espaço alocado para a unidade foi um terreno bem comprido doado por um cidadão nobre da cidade. O engenheiro da prefeitura, alocado para construir o prédio, optou por erguer uma garagem bem comprida com capacidade para estacionar até 10 ambulâncias, uma atrás da outra. A garagem possui apenas uma porta para entrada e saída dos veículos. Logo, a primeira ambulância a entrar é necessariamente a última a sair. Este prédio também tem uma sala com capacidade para cinco atendentes, cada atendente fica próximo de um aparelho telefônico para atender as chamadas. Para reduzir custos, a prefeitura recebeu uma autorização para cobrar R$10,00 por cada ocorrência finalizada. Desta forma, a unidade não vai onerar tanto os gastos municipais".

Empolgado, o seu tio então continua: "O que precisamos é de um sistema para cadastrar os cidadãos, atendentes e ambulâncias, registrando as ocorrências da população. Ao ligar para a unidade, o cidadão pode solicitar um novo atendimento ou verificar o status de um atendimento solicitado. O atendente que está a mais tempo parado deve ser alocado para atender ao chamado, caso não tenha nenhum atendente disponível, o sistema deve solicitar ao cidadão que ligue mais tarde. Assim que um atendente realiza um atendimento, ele fica disponível para atender outro chamado. Para atender a uma ocorrência, um atendente pode solicitar uma ou mais ambulâncias. As ambulâncias mais próximas da porta da garagem devem ser alocadas para atender o chamado. Além disso, você deve observar que cada ocorrência tem um identificador único, a identificação do cidadão que solicitou o pedido e o status da ocorrência".

Ao chegar em Feira de Santana para iniciar o semestre letivo, você entra em contato com um colega da sua cidade natal que já cursou a disciplina de Engenharia de Software. O seu colega então fala o seguinte: "Eu vou te ajudar, mas eu não tenho tempo para programar, portanto eu vou preparar as *user stories* e um conjunto de testes de software".

Ao chegar em casa a noite, você verifica que tem um novo email do seu colega contendo as *user stories*, um modelo conceitual e um arquivo compactado contendo os testes de aceitação para o mesmo. Todo animado, você inicia a leitura das *user stories*, do diagrama e dos testes. Você pensa: "eu vou conseguir fazer esse sistema, mas como o meu tio quer que ele fique pronto logo, eu vou desenvolver uma primeira versão do software utilizando uma interface de linha de comando. Assim, ele fica testando, enquanto eu ganho tempo para aprender a desenvolver uma interface gráfica com maior usabilidade".

Modelo Conceitual:



User Stories:

User Story	Título	Breve Descrição
1	Cadastrar cidadão	Um novo cidadão é cadastrado no sistema
2	Listar cidadãos	Lista todos os cidadãos ordenados pelo nome
3	Consultar dívida	Retorna a dívida atual de um cidadão
4	Pagar dívida	A dívida do cidadão é reduzida em um determinado valor
5	Cadastrar atendente	Um novo atendente é cadastrado e fica disponível para cadastrar/consultar ocorrências
6	Listar atendentes	Lista todos os atendentes ordenados pelo nome
7	Listar atendentes disponíveis	Lista todos os atendentes disponíveis para atender ocorrencias
8	Cadastrar ambulância	Uma nova ambulância é cadastrada e fica disponível para atender as ocorrências em aberto ou as novas ocorrências
9	Listar ambulâncias disponíveis	Uma lista com todas as ambulâncias estacionadas na garagem é exibida na tela, da mais próxima da porta para a mais distante
10	Listar ambulâncias	Uma lista com todas as ambulâncias, ordenadas pela placa da ambulância, é exibida na tela
11	Cadastrar ocorrência	O atendente cadastra a ocorrência no sistema. O status da ocorrência é marcado como “aberto“ e o atendente fica indisponível até que a ocorrência seja atendida
12	Atender ocorrência	Um ambulância é alocada para atender a ocorrência (a ambulância mais próximo da porta da garagem). O status da ocorrência é marcado como “em atendimento“ e atendente fica disponível para cadastrar outra ocorrência
13	Finalizar ocorrência	As ambulâncias relacionadas a ocorrência são estacionadas na garagem, o atendimento é marcado como “finalizado“ e as ambulâncias ficam disponíveis para atender novas ocorrências

14	Consultar ocorrência	Recupera uma ocorrência a partir do seu id
15	Listar ocorrências ativas	Todas as ocorrências ativas (status seja "em aberto "ou "em atendimento") são listadas, da mais antiga para a mais recente

Produto:

Você deve enviar um e-mail com o produto final para o seu tutor até às 12 horas (meio dia) do dia **16/05/2014** (sexta-feira), anexando o arquivo compactado com o código-fonte e os testes. O código fonte deve estar todo documentado utilizando o padrão javadoc. Também deverá ser entregue um relatório técnico **individual** escrito no formato de artigo da SBC (http://sites.ecomp.uefs.br/mip-2014-1/home/tutorial). O relatório deve ter entre quatro e seis páginas (excluindo os anexos das atualizações do modelo conceitual e do diagrama de classes de projeto), contendo:

- Introdução com a apresentação resumida da motivação, contexto e problema;
- Fundamentação teórica;
- Metodologia utilizada;
- Resultados e discussão (com a descrição detalhada da sua solução, declarando as decisões que você teve de tomar e fundamentando-as, sempre que possível);
- Conclusões (lições aprendidas e os pontos positivos e negativos deste projeto).

Você deve entregar o código-fonte em um só arquivo compactado, com todas as classes que você desenvolveu para este projeto. Todas as classes devem estar compilando e implementando as funcionalidades adequadamente. Todos os testes devem estar executando e passando. Todas as classes, atributos e métodos que você criou devem estar documentados utilizando o padrão javadoc. A correção será feita avaliando o código, executando os testes e apreciando a documentação javadoc.

Avaliação:

O problema entregue corresponde a 40% da nota da unidade, o relatório corresponde a 30% e o desempenho nos tutoriais corresponde a 30% da nota.

Calendário:

Aula	**Data**	**Atividade**
1	04/04	Apresentação do Problema
2	08/04	PBL
3	11/04	PBL
4	15/04	PBL
	18/04	*Feriado Semana Santa*
	22/04	*Não haverá sessão*
	25/04	*Feriado Micareta*
5	29/04	PBL
	02/05	*Ponto facultativo*
6	06/05	PBL
7	09/05	PBL
8	13/05	PBL
	16/05	Entrega do problema 2 (código e relatório)