

**Universidade Estadual de Feira de Santana**  
**EXA863 MI Programação - 2019.2**

**Prof.:** Rodrigo Tripodi Calumby, Carlos Alberto Rodrigues, Rafael Tosta e Moab Rodrigues

**Projeto 1: Precisamos de uma Unidade de Pronto Atendimento (UPA) na UEFS!**

Considerando a ausência de um sistema de pronto atendimento no Campus da Universidade Estadual de Feira de Santana, a equipe da administração superior busca alternativas para viabilizar a implantação de um serviço deste tipo na universidade. Ao buscar mais informações sobre a situação, você descobriu que um dos desafios do projeto é a implantação de um sistema para gerenciamento dos atendimentos e realização de exames médicos.

Dada a importância do projeto você prontamente candidata-se para fornecer um software para permitir que o atendimento seja totalmente informatizado. Sendo este um projeto de grande porte, e para permitir a implantação do serviço o quanto antes, a equipe solicita que você entregue em curto prazo uma versão do sistema com as funcionalidades centrais e que permita sua utilização para os serviços básicos.

Você inicia uma conversa com O Sr. Ragnar Lothbrok, responsável pela coordenação do serviço, que logo define que um dos pontos mais críticos é a ordem de atendimento dos pacientes que chegarão ao setor de pronto atendimento. Nesta primeira versão do sistema, o atendimento é realizado considerando apenas a ordem de chegada dos pacientes. Você considera isso relativamente simples, mas o Sr. Ragnar, lembra que por lei, pacientes com direito à prioridade são colocados no início da fila de atendimentos, claro, logo após os demais pacientes prioritários.

Neste momento, o Pró-Reitor de Atenção à Comunidade, Prof. Ecbert Ealhmunding, entra na sala com sorriso no rosto e conta que 2 médicos foram cedidos pela Prefeitura de Feira para trabalhar na UEFS imediatamente e que posteriormente chegarão outros! Antes mesmo que ele deixe a sala, o Sr. Ragnar recebe uma ligação da Assessoria Especial de Informática (AEI), era o coordenador de novos sistemas, o Sr. Ivar Ragnarsson, lembrando que os novos sistemas da UEFS devem atender a altos padrões de qualidade e ser bastante fáceis de entender e de manter, incluindo uma modelagem adequada e algo que ele chamou de “unit test”.

Junto ao Sr. Ragnar e os demais colaboradores vocês conseguem elaborar uma lista de requisitos que são apresentados na Tabela 1.

Código	Título	Descrição
1	Inclusão de novo médico	Um novo médico está disponível e então novos pacientes podem ser adicionados na sua fila de atendimento. Deve-se registrar o nome e número de CRM do médico.
2	Editar dados do médico	Dado o CRM de um médico, deve ser possível editar seu nome.

3	Encaminhamento de paciente	Um paciente que chega ao hospital é adicionado na fila do médico que tiver menos pacientes aguardando atendimento. Se o paciente for prioritário ele vai para a frente da fila, logo após os demais prioritários. Deve-se registrar o número de matrícula do paciente.
4	Listar fila de pacientes	Dado o CRM de um médico, deve ser possível listar os números de matrícula dos pacientes na sua fila e na ordem em que devem ser atendidos.
5	Realizar atendimento	O paciente é imediatamente removido da fila de atendimentos. O médico atende um paciente, quando pode então solicitar alguns exames.
6	Solicitação de exames	Ao atender um paciente, o médico pode solicitar exames. De modo a organizar o atendimento posterior, os exames são incluídos nas listas de exame de cada paciente e na lista geral de atendimento para exames.
7	Listagem de exames	Dado o tipo de exame, deve ser possível listar o nome de todos os pacientes aguardando para realizá-lo, considerando a exata ordem que eles foram inseridos na fila e indicar a quantidade de pessoas <b>que precisam daquele exame</b> .
8	Listar atendimentos realizados por um médico	Dado o CRM um médico, deve ser possível listar o nome de todos os pacientes que já foram atendidos por ele.
9	Listar exames solicitados	Dado o <b>número de matrícula</b> de um paciente, o sistema deve mostrar todos os exames solicitados.
10	Realizar exame	<b>Dado um tipo de exame, o próximo paciente na fila de exames que precise daquele tipo de exame é atendido. Um paciente só sai da fila após todos os seus exames terem sido realizados. Sua posição na fila não é alterada entre um exame e outro.</b>
11	Testes automatizados	Todos os requisitos do sistema devem poder ser extensivamente testados e isso deve ser feito de modo automatizado. Os testes devem ser devidamente descritos para que os próximos desenvolvedores entendam rapidamente o que exatamente cada teste verifica em relação ao funcionamento do sistema.

Alguns minutos depois você recebe um e-mail do Sr. Floki Boatbuilder, responsável pela gerência de manutenção de sistemas e criterioso como sempre, indicando que devem ser fornecidos testes automatizados suficientes para garantir uma perfeita validação da implementação do sistema. Ele deixa claro que os testes não devem verificar apenas casos básicos simples, mas principalmente casos extremos, como, por exemplo, quando a fila está vazia e vai receber o primeiro exame ou paciente, quando uma fila de atendimento é totalmente esvaziada, quando em um processo de busca um médico ou paciente não é encontrado, quando um paciente prioritário precisa ser incluído na fila de atendimentos em que outros

prioritários já estão, a tentativa de realizar um tipo de exame em um paciente para o qual aquele tipo de exame não foi solicitado, etc.

## Produto

Você deve enviar um e-mail com o produto final para o seu tutor até às 10 horas da manhã do dia 25/11/2019, anexando o arquivo compactado com o **Código-Fonte, o Modelo Conceitual e os Testes**. Para cada requisito devem ser fornecidos pelo menos 3 testes diferentes com suas respectivas descrições (incluídas no código).

A entrega após este horário implicará em um desconto de 2 pontos na nota do produto. Após 24 horas de atraso, será descontado mais 1 ponto na nota. O problema não será mais recebido após 48hs de atraso. O produto entregue corresponde a 70% da nota do problema e o desempenho nos tutoriais corresponderá a 30% da nota.

Você deve entregar, individualmente, 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 rodando e passando. Todos as classes, atributos e métodos que você criou devem estar documentados utilizando o padrão javadoc. A correção será feita a partir do código, rodando os testes de unidade e avaliando a documentação javadoc.

Atenção! Antes de enviar o produto, certifique-se de ter lido o documento “Evitando cópias indevidas em trabalhos acadêmicos” disponível no [https://sites.google.com/a/ecompu.uefs.br/mip-20182/home/tutorial/plagio\\_e\\_citacoes.doc](https://sites.google.com/a/ecompu.uefs.br/mip-20182/home/tutorial/plagio_e_citacoes.doc) e de ter adicionado a declaração de autoria do código nas classes desenvolvidas.