

Programação Orientada a Objetos

Bacharelado em Ciência da Computação

Prof. Dr. Eduardo Takeo Ueda

3º Trabalho

Unidade Básica de Saúde

Data de entrega: 19 de maio de 2014

No Brasil, uma unidade de atendimento de saúde ou UBS (Unidade Básica de Saúde), é onde encontram-se médicos, enfermeiros e dentistas para atender à população de um bairro ou de uma determinada região. Para auxiliar o trabalho desses profissionais da área de saúde você deve desenvolver um sistema simples em que pacientes podem agendar consultas com um médico ou dentista e ao mesmo tempo médicos, enfermeiros e dentistas podem ter acesso ao PEP (Prontuário Eletrônico do Paciente) para consultar e adicionar informações (não deve ser possível alterar). Sua tarefa consiste de 3 etapas:

- (a) elaboração do diagrama de casos de uso utilizando a ferramenta de modelagem *ASTAH*;
- (b) produção do diagrama de classes, também com a ferramenta *ASTAH*;
- (c) e por último implementação do sistema com a linguagem de programação Java.

Todos os usuários devem ter acesso ao sistema através de um **login** e **senha**. Um paciente pode agendar consulta com um médico ou um dentista e também acessar (apenas consultar) as informações do seu PEP, pois é possível que ele queira saber resultados de exames, por exemplo. Uma vez que um paciente conseguiu uma consulta com um médico, no dia e horário agendados ele deve passar antes por um enfermeiro que vai medir a pressão arterial e peso para adicionar estas informações ao PEP. Eventualmente o enfermeiro pode acrescentar alguma observação que considere relevante. No caso do paciente ter agendado consulta com um dentista não existe necessidade de passar por um enfermeiro.

Quando o médico atender o paciente ele basicamente fará um *checklist* de itens que podem indicar algum tipo de problema de saúde. Dependendo das respostas do paciente o médico pode considerar ele saudável ou encaminhar para exames médicos detalhados. Todas estas informações devem ser acrescentadas ao PEP, mas uma vez salvas não podem ser alteradas. Você deve pesquisar, preferivelmente com algum

profissional de saúde, para descobrir qual deve ser o conjunto mínimo de itens questionados para um atendimento médico com um mínimo de qualidade. O mesmo vale para possíveis exames que podem ser pedidos ao paciente e devem constar no PEP.

No caso da consulta ser com um dentista o atendimento é mais simples. O dentista da UBS deve realizar apenas procedimentos não complexos como limpeza e/ou obturação dos dentes. De qualquer forma o dentista deve registrar o que foi feito no PEP, e também não deve alterar nenhuma das informações salvas anteriormente, preservando a integridade do PEP.

Para implementação (em Java) e execução do sistema você deve considerar que existem pelo menos 2 pacientes, 2 enfermeiros, 2 médicos e 2 dentistas. Cada paciente deve poder escolher qual o médico ou dentista ele deseja agendar consulta, mas não importa qual enfermeiro vai atender ele. Alguns cuidados devem ser tomados na hora da implementação, por exemplo, não permitir que 2 pacientes agendem consulta com o mesmo médico (ou dentista) no mesmo dia e horário.

Para que exista **persistência** das informações do PEP, assim como cadastros com login e senha dos usuários, você deve utilizar arquivos **CSV** (Comma-Separated Values), pois é uma forma muito fácil de armazenar dados que podem ser manipulados por praticamente qualquer editor de textos e/ou planilhas.

O desenvolvimento de interface gráfica não é obrigatório, mas é altamente **recomendado** e aqueles que implementarem esta funcionalidade terão um **bônus de até 1 ponto** na nota do trabalho, dependendo da qualidade do *layout* produzido.

Também vale **até 1 ponto** de **bônus** a aplicação de pelo menos 1 dos padrões de projeto (*Design Patterns*) do **GoF** (*Gang of Four*). Porém, é necessário explicar em detalhes como o padrão foi aplicado no desenvolvimento do sistema.

Recomendações importantes

1. O trabalho pode ser feito individualmente ou em grupo de dois alunos, mas lembre-se que **não é permitido** o mesmo grupo do 1º ou 2º trabalho.
2. O grupo que for composto por membros que já fizeram o 1º ou 2º trabalho juntos será punido com nota 0 (**ZERO**).
3. Produza um pequeno **relatório** descrevendo os testes que foram feitos e indicando como o professor da disciplina poderá reproduzi-los para conseguir os mesmos resultados.
4. Você deve submeter no BlackBoard um arquivo .rar (ou .zip), contendo os diagramas de caso de uso e diagramas de classe (produzidos com o ASTAH), o

código-fonte Java (implementado no ambiente Windows com NetBeans), arquivos CSV, o relatório (item 3) e um arquivo README (com instruções de como executar seu código); e identificado com a concatenação do seu primeiro nome com seu sobrenome. Por exemplo, meu nome é Eduardo Takeo Ueda, então devo submeter um arquivo chamado EduardoUeda.rar (ou .zip).

5. Se você por acaso julgar que é(são) necessário(s) outro(s) arquivo(s) além dos exigidos neste enunciado, poderá adicionar ao arquivo .rar, desde que justifique no arquivo README porque foi preciso fazer isso.
6. Não esqueça de deixar comentários explicando muito bem seu código-fonte, pois isso será levado em conta na correção do trabalho.
7. Códigos-fonte que forem confirmados como **plágio** receberão nota 0 (**ZERO**) no trabalho.
8. Não serão aceitos trabalhos entregues por e-mail, apenas pelo sistema Black-Board.
9. Não serão aceitos trabalhos atrasados, ou seja, depois da data limite de entrega.
10. Não deixe para fazer o trabalho na última hora, como na véspera da data de entrega.

Bom trabalho!