

# Programação Orientada a Objetos

Bacharelado em Ciência da Computação

Prof. Dr. Eduardo Takeo Ueda

Trabalho

## Unidade Básica de Saúde

Data de entrega: 2 de junho de 2016

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. É importante que médicos, enfermeiros e dentistas tenham permissão de alterar apenas as informações que foram adicionados por eles próprios, ou seja, um médico não deve poder alterar o que foi inserido no PEP por um enfermeiro ou dentista, por exemplo. Sua tarefa consiste de 3 etapas:

- (1) elaboração do diagrama de casos de uso utilizando a ferramenta de modelagem *ASTAH*;
- (2) produção do diagrama de classes, também com a ferramenta *ASTAH*;
- (3) 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, peso e temperatura corporal para adicionar estas informações ao PEP. Eventualmente o enfermeiro pode acrescentar alguma observação que considere relevante, como uma possível alergia a um medicamento. 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. Para delimitarmos o escopo deste trabalho consideremos que qualquer um dos médicos irá diagnosticar apenas casos de **dengue**, **chikungunya** e **zika**. Dependendo das respostas do paciente o médico pode considerar ele saudável ou encaminhar para exames médicos mais detalhados. Todas estas informações devem ser acrescentadas ao PEP, para posterior acesso do paciente e dos demais profissionais de saúde. 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 e diagnóstico correto das doenças citadas. 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 também deve registrar o que foi feito no PEP, mas não tem permissão para alterar informações registradas por um médico ou enfermeiro, preservando a integridade do PEP.

Para implementação (em Java) e execução do sistema você deve considerar que existem pelo menos 3 pacientes, 3 enfermeiros, 2 médicos e 2 dentistas cadastrados. 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 a biblioteca **XStream** para serializar objetos para o formato **XML** e vice-versa.

## Recomendações importantes

1. O trabalho pode ser feito individualmente ou em grupo de no máximo dois alunos. No caso de formação de grupo os nomes dos integrantes devem ser explicitados no código-fonte e no arquivo README.
2. 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 XML, 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).
3. 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.
4. Não esqueça de deixar comentários explicando muito bem seu código-fonte, pois isso será levado em conta na correção do trabalho.
5. Códigos-fonte que forem confirmados como **plágio** receberão nota 0 (**ZERO**) no trabalho.
6. Não serão aceitos trabalhos atrasados e nem entregues por e-mail, apenas pelo sistema BlackBoard.
7. Não deixe para fazer o trabalho na última hora, como na véspera da data de entrega.

**Bom trabalho!**