

SSC0103 - Programação Orientada a Objetos

Prof. Adenilso da Silva Simão

Trabalho 4

Entrega: 27/06/2015

Grupos de no máximo 3 pessoas

Simulador de supermercado online

Implemente um sistema que simula um **supermercado online**. O sistema deve:

1. Implementar uma aplicação para o **cliente** e outra para o **servidor**.
2. Na aplicação cliente o sistema deve:
 - a. **Cadastrar novo usuário** no servidor (nome, endereço, telefone, email, ID, senha);
 - b. Permitir que usuário faça o **login** no servidor (com ID e senha);
 - c. Recuperar a **lista de produtos** do servidor (obter os dados do servidor);
 - d. **Listar os produtos** disponíveis e indisponíveis (mostrar os dados recuperados);
 - e. Permitir que o usuário requisiute uma **notificação** para os produtos indisponíveis desejados. **Quando o estoque for abastecido e atualizado** no servidor o usuário automaticamente recebe uma notificação por email (JavaMail API) **(PONTO EXTRA)**.
 - f. **Permitir efetuar a compra de produtos**. Verificar no momento da compra se existe no estoque a quantidade requisitada de produtos. Se não, avisar o usuário quais produtos estão em falta no momento (pela interface apenas). A operação de compra deve ser sincronizada com o servidor, pois duas transações podem estar ocorrendo ao mesmo tempo.
3. Na aplicação servidor o sistema deve:
 - a. Permitir o cadastro de novos produtos (formulário com campos: nome, preço, validade, fornecedor);
 - b. Listar os produtos em estoque e os esgotados;
 - c. Realizar a atualização do estoque de produtos já existentes e notificar o cliente (se um produto estava em falta e o fornecedor enviou mais produtos, quando estes chegarem atualize a quantidade no estoque e mande um email para o cliente avisando que o produto está disponível. Quando um produto está em falta o sistema pede ao fornecedor);
 - d. Permitir que vários clientes se conectem ao mesmo tempo no sistema (threads);
 - e. Permitir a geração de um relatório em PDF com informações sobre as vendas do dia e do mês **(PONTO EXTRA)**;
 - f. Estabelecer a conexão entre cliente e servidor via socket.

O sistema deve ler e recuperar as informações em arquivos, no formato CSV.

Entrega

Todas as classes e interfaces devem ser colocadas **pacotes** (*package*). Incluir um arquivo de **README**, com **instruções** de como executar o programa junto com seu funcionamento. Os pacotes devem ser exportados como um **.jar executável** contendo o **código fonte** e o **README**.

Deve-se criar um projeto no **github**, criar uma **tag** para a versão do trabalho a ser avaliar.

Pontos extra

- 1 - Interface Gráfica em JavaFX (1.5)
- 2 - Teste em JUnit (1.5)
- 3 - Geração de relatórios PDF (1.5)
- 4 - Uso de pelo menos 1 padrão de projeto e descrever no README. (1.5)
- 5 - Notificação de Produtos Indisponíveis / JavaMail API (1.5)

A geração de relatórios em PDF (uso de bibliotecas extras). Ex.: itext - <http://itextpdf.com/> e <http://sourceforge.net/projects/itext/files/>

Envio de notificações por email (uso de bibliotecas extras). Ex.: java.mail e <https://cloud.google.com/appengine/docs/java/mail/usingjavamail> e <http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/javamail/javamail.html>
