Tapioca Maker

# 1. Descrição do Projeto

O dono de uma tapiocaria irá cadastrar ingredientes que podem ser usados para fazer uma tapioca (além do nome e da descrição do ingrediente, esse cadastro deve incluir o preço que ele cobrará por cada 100 gramas desse ingrediente e se ele está disponível no estoque). Além de cadastrar ingredientes, o dono da pizzaria poderá cadastrar receitas de tapioca, juntando os ingredientes que já foram cadastrados. O preço de uma tapioca précadastrada é definido pela soma dos preços de seus ingredientes (lembrando que em uma tapioca pode ir mais que 100 gramas de um mesmo ingrediente). Para cadastrar tapiocas e ingredientes o dono da tapiocaria deve passar por um processo de autenticação. Os clientes da tapiocaria, entretanto, não precisam logar no sistema para comprar. Eles podem comprar as tapiocas pré-cadastradas ~~ou montar sua própria tapioca~~. Para fechar o pedido, ele informa as suas informações de contato (celular e e-mail) e os dados do seu cartão de crédito. Se o preço do ingrediente for atualizado, automaticamente todas as tapiocas que usam aquele ingrediente tem seu preço reajustado. Se o dono da tapiocaria sinaliza que o ingrediente não está mais disponível no estoque, todas as tapiocas que o usam devem ficar indisponíveis para os próximos clientes, até que o ingrediente volte a ficar disponível no estoque.

# 2. Requisitos Funcionais

1. Cadastrar administrador - como o administrador é um usuário especial e único no sistema, quando o programa for iniciado pela primeira vez, se não houver nenhum administrador cadastrado a única funcionalidade disponível deve ser a de cadastro de administrador. O sistema deve registrar, pelo menos, um login e uma senha para o administrador.
2. Redefinir a senha – caso esqueça a senha, o sistema deve fornecer ao administrador a possibilidade de redefinir a senha, utilizando seu e-mail. Ele recebe uma senha provisória e, após usá-la, o sistema pedirá para ele cadastrar a nova senha.
3. Cadastrar ingrediente – o administrador do sistema pode cadastrar ingredientes que serão usadas para preparar as tapiocas. No cadastro do ingrediente, ele deve informar um nome, um valor calórico, o preço por 100 gramas desse ingrediente e se ele se encontra disponível ou não no estoque.
4. Editar ingrediente – o administrador do sistema pode editar as informações do ingrediente. Ao atualizar o valor calórico ou o preço, todas as tapiocas já cadastradas também vão sofrer uma atualização. Ao editar o preço do ingrediente, todas as tapiocas que incluem aquele ingrediente que forem compradas, a partir desse momento, passam a custar o novo preço. Se a disponibilidade for alterada, da mesma forma, tapiocas que usam aquele ingrediente passam a ficar disponíveis ou indisponíveis para serem comparadas, dependendo no novo status do produto.
5. Deletar ingrediente – só podem ser deletados ingredientes que não estão sendo usados em nenhuma tapioca.
6. Listar ingredientes (com filtro) – o sistema deve possibilitar o administrador de visualizar a lista de ingredientes cadastrados. Essa lista deve oferecer a possibilidade de filtros: por nome, por estado, por faixa de preço e por faixa calórica.
7. Cadastrar tapioca – os clientes da tapiocaria podem comprar tapiocas disponíveis no cardápio. Para isso, o administrador deve ter a possibilidade de cadastrar tapiocas, montando elas com ingredientes disponíveis no sistema. O preço da tapioca é automaticamente calculado com base no preço dos ingredientes que a compõem, assim como o valor calórico dela. O cadastro da tapioca deve dar a possibilidade de colocar um nome na tapioca.
8. Listar tapiocas (com filtro) – o sistema deve oferecer a opção de visualizar todas as tapiocas que já foram cadastradas. Essa visualização é importante para que o cliente possa fazer seu pedido. Além disso, essa listagem deve oferecer a opção de um filtro por nome (cujo nome é composto por algum vocábulo fornecido) ou por ingrediente (i.e., listar tapiocas que possuem um ou mais ingredientes específicos).
9. Visualizar tapioca – durante a visualização da lista, o usuário tem a opção de selecionar uma tapioca para vê-la com mais detalhes. Caso o usuário seja um cliente, ele pode fazer o pedido dessa tapioca, colocando-a no carrinho de compras. Caso o usuário seja o dono da tapiocaria, ele pode usar essa visualização para excluir a tapioca do cardápio.
10. Excluir tapioca – o dono da tapiocaria pode apagar tapiocas previamente cadastradas, tirando-as do cardápio. Essas tapiocas, entretanto, continuarão a aparecer nos relatórios, mas não serão mais exibidas (nem para os clientes, nem para o administrador).
11. Realizar pedido de uma tapioca do cardápio – o cliente da tapiocaria pode adicionar tapiocas em um carrinho de compras e, ao final, fazer o pedido com essas tapiocas. Ao fazer o pedido, ele deve passar as informações sobre onde a tapioca deverá ser entregue e informações para contato (e-mail e celular). Nesse momento, o status do pedido é em aberto.
12. Enviar e-mail com os dados do pedido – o dono da tapiocaria e o cliente recebem um e-mail com as informações sobre o pedido que foi feito.
13. Listar todos os pedidos (com filtro) – ao usar o sistema, o dono da tapiocaria pode visualizar a lista de pedidos que já foram feitos (deve haver a opção de filtrar os pedidos pelo status).
14. Visualizar um pedido – o administrador pode visualizar com mais detalhes as informações de um pedido e alterar seu status para fechado, indicando que ele já foi entregue.
15. Enviar mala direta a todos os clientes que já fizeram pedidos – o dono da tapiocaria tem a opção de enviar uma mala direta a todos os clientes que já fizeram algum pedido, para informar de promoções ou avisar sobre o funcionamento da tapiocaria.
16. Gerar relatório, em PDF, com a lista de tapiocas ordenadas por popularidade – o dono da tapiocaria deve ser capaz de facilmente identificar quais são as tapiocas que os clientes mais pedem, assim, esse relatório deve incluir a quantidade de vezes que cada tapioca fez parte de pedidos e o valor total arrecadado por cada tapioca, juntando todas as vezes que fez parte de um pedido.

# 3. Algumas das Classes do Sistema (sugestões)

Sugere-se que o sistema inclua as classes:

* Administrador – representa o usuário que é capaz de fazer a inserção de informações dentro do sistema.
* Ingrediente – representa os recheios que poderão compor uma tapioca.
* Tapioca – uma tapioca é composta por ingredientes e poderá fazer parte de pedidos. Como as características da tapioca (preço, valor calórico, etc.) podem mudar, ao longo do tempo, sugere-se que os perdidos “guardem” uma cópia da tapioca no seu estado atual (um clone).
* Cliente – o cliente é o usuário que é capaz de adicionar tapiocas no carrinho para fazer um pedido.
* Pedido – corresponde as informações do cliente e das tapiocas que o cliente está comprando.

# 4. Como começar esse projeto

A partir do segundo estágio:

* Comece criando as classes que vão representar as entidades que serão manipuladas no projeto, com seus atributos, métodos (construtores, gets e sets, equals, toString, etc.).
* [sugestão] Crie uma classe chamada “Central”. Adicione a essa classe uma lista de Ingredientes, Tapiocas, Pedidos e quaisquer outras informações que você julgue necessário persistir. Essa classe reunirá todas as informações que precisaremos manipular e “guardar” no sistema.
* Crie, na classe “Central”, métodos para cadastrar um administrador, trocar senha do administrador, cadastrar ingrediente, recuperar informações de um ingrediente, cadastrar uma nova tapioca, recuperar as informações de uma tapioca, etc. Essas funcionalidades da “Central” estão diretamente relacionadas a cada uma das funcionalidades que seu projeto terá. A implementação dessas funcionalidades deve ser feita da forma mais simples possível; esses métodos não devem realizar impressão no console nem solicitar dados do usuário via console. Dê preferência para que cada método receba toda informação que você acha que ele precisará como parâmetro de entrada.
* Se você optar por fazer a persistência de dados usando XSTream, pesquise sobre a biblioteca e sobre como salvar e recuperar objetos com ela.
  + (parte 1). Ex. Crie uma classe que receba um objeto do tipo ingrediente e salve ele em uma arquivo xml. Verifique se o arquivo criado representa seu objeto. Depois, mude essa classe para receber, ao invés de um ingrediente, uma “Central”.
  + (parte 2). Ex. Crie uma classe que recupere do arquivo gravado o objeto correspondente.
* Crie os métodos para imprimir no console a informação dos relatórios. Em seguida, pesquise sobre como gerar os relatórios no formato pedido.
* Crie um método para enviar um e-mail para alguém. O método recebe o texto do e-mail e o destinatário.
* *Faça um design de como você imagina que a navegação do seu projeto deva funcionar. Desenhem, nesse planejamento, cada tela que o projeto terá e como se chegará de uma tela à outra (navegação entre telas). Para facilitar o projeto, imagine que a Janela Principal do projeto possui uma barra de menus que oferece acesso a todas as funções do projeto.*
* ESCREVA O RELATÓRIO FINAL.
* Apresente o projeto.

# 5. Interface Gráfica

O aluno deverá desenvolver o projeto para plataforma desktop (utilizando Swing). **O aluno deve desenvolver a interface gráfica do projeto sem a ajuda de programas do tipo clique e arraste.**

# 6. Persistência de Dados

A persistência de dados pode ser implementada de uma dessas formas: utilizando arquivo (binário, xml, json ou txt) ou banco de dados. No caso de persistência em banco de dados, recomenda-se o uso do JPA. No caso de persistência em formato XML, recomenda-se o uso do XStream.

# 7. Sobre os grupos

Os alunos podem optar em fazer o trabalho individualmente ou em dupla. A avaliação de cada aluno levará em conta sua contribuição para o projeto final.

# 8. Relatório

## 8.1. Relatório quinzenal

O aluno deverá entregar A CADA 15 DIAS um relatório com ao menos 10 linhas (fonte times new roman 12) explicando:

* O que cada aluno tem feito no projeto;
* Dificuldades;
* Planejamento.

## 8.2. Relatório final

No dia anterior a apresentação, todas as equipes devem DISPONIBILIZAR O EXECUTÁVEL DO PROJETO PARA MIM, para que eu possa me familiarizar com a aplicação, antes do dia da apresentação.

Na apresentação do projeto, CADA ALUNO deve entregar um relatório final, com pelo menos quatro páginas, descrevendo:

* O que o aluno fez no projeto;
* Quais funcionalidades o grupo conseguiu implementar;
* Quais funcionalidades o grupo não conseguiu implementar e por que;
* Experiência do aluno no desenvolvimento no projeto (o que ele mais gostou, maiores dificuldades, se o projeto foi proveitoso para o seu aprendizado).
* Quais as bibliotecas o grupo utilizou para desenvolver o projeto e porque eles optaram por elas;
* Referências que os alunos consultaram para desenvolver o projeto (links de tutoriais, perguntas em fóruns, etc.);
* Impressão de uma classe que os alunos considerem importante no projeto;
* Documentação dessa classe (JavaDoc).

# 7. Sobre a nota

Para a nota do aluno será levado em consideração a quantidade de requisitos funcionais que ele conseguiu implementar e suas decisões em relação aos requisitos não funcionais. Serão levados em consideração também aspectos relacionados ao código e a postura do aluno, em sala e durante a apresentação.

## 7.1. Dimensões que serão avaliadas

|  |
| --- |
| D1 – CÓDIGO |
| D2 – ABSTRAÇÕES OO |
| D3 – COMPLETUDE E CORRETUDE FUNCIONAL |
| D4 – INTERFACE COM USUÁRIO |
| D5 – TRATAMENTO DE SITUAÇÕES DE ERRO |
| D6 – DIFERENCIAL DO PROJETO |
| D7 – AVALIAÇÃO INDIVIDUAL DO ALUNO |
| D8 – TRABALHO EM GRUPO |
| D9 – DEFESA ORAL |

# 8. Sugestões de Referências

A seguir, algumas sugestões de bibliotecas que vocês podem utilizar no projeto.

## 8.1. Geração de Arquivos em PDF

A própria API de manipulação de arquivos nativa de Java pode ser usada para criar arquivos em PDF mais simples. Entretanto, há uma API chamada iText que oferece a mesma função, também é simples, porém mais completa. Usando o IText, vocês podem gerar relatórios que incluam tabelas, cabeçalhos, etc.

Um Bom Tutorial do IText:  
<http://www.forumweb.com.br/artigo/223/java/usando-a-api-itext-para-gerar-documentos-pdf-em-java>

<https://www.youtube.com/watch?v=jcJehf9mGmc>

<https://www.youtube.com/watch?v=eBRL70rR30I>

## 8.2. Persistência de Dados com XStream

Um Bom Tutorial do XStream:  
<http://deixaeuprogramar.weebly.com/home/criando-e-lendo-arquivo-xml-em-java-com-xstream>

<https://www.youtube.com/watch?v=AW6619JR-RE&feature=youtu.be>

## 8.3. Envio de e-mail

Existem várias APIs para envio de e-mail. Recomenda-se o uso da Java Mail.

Um bom tutorial:

<https://www.devmedia.com.br/enviando-email-com-javamail-utilizando-gmail/18034>

https://youtu.be/PnIx26AGZ7Y