



FABRICIO RIBEIRO BUENO YAMAMOTO

WEBCAMELÔ

Relatório apresentado à Universidade Federal do ABC como parte dos requisitos para aprovação na disciplina MC0037 - Programação para Web do Curso de Bacharelado em Ciência da Computação.

Professora: Vera Nagamuta

Santo André - SP

2016

Sumário

1 Introdução.....	3
2 Desenvolvimento.....	4
3 Simulação e Testes	8
4 Conclusões	13
5 Referências Bibliográficas	14

1 Introdução

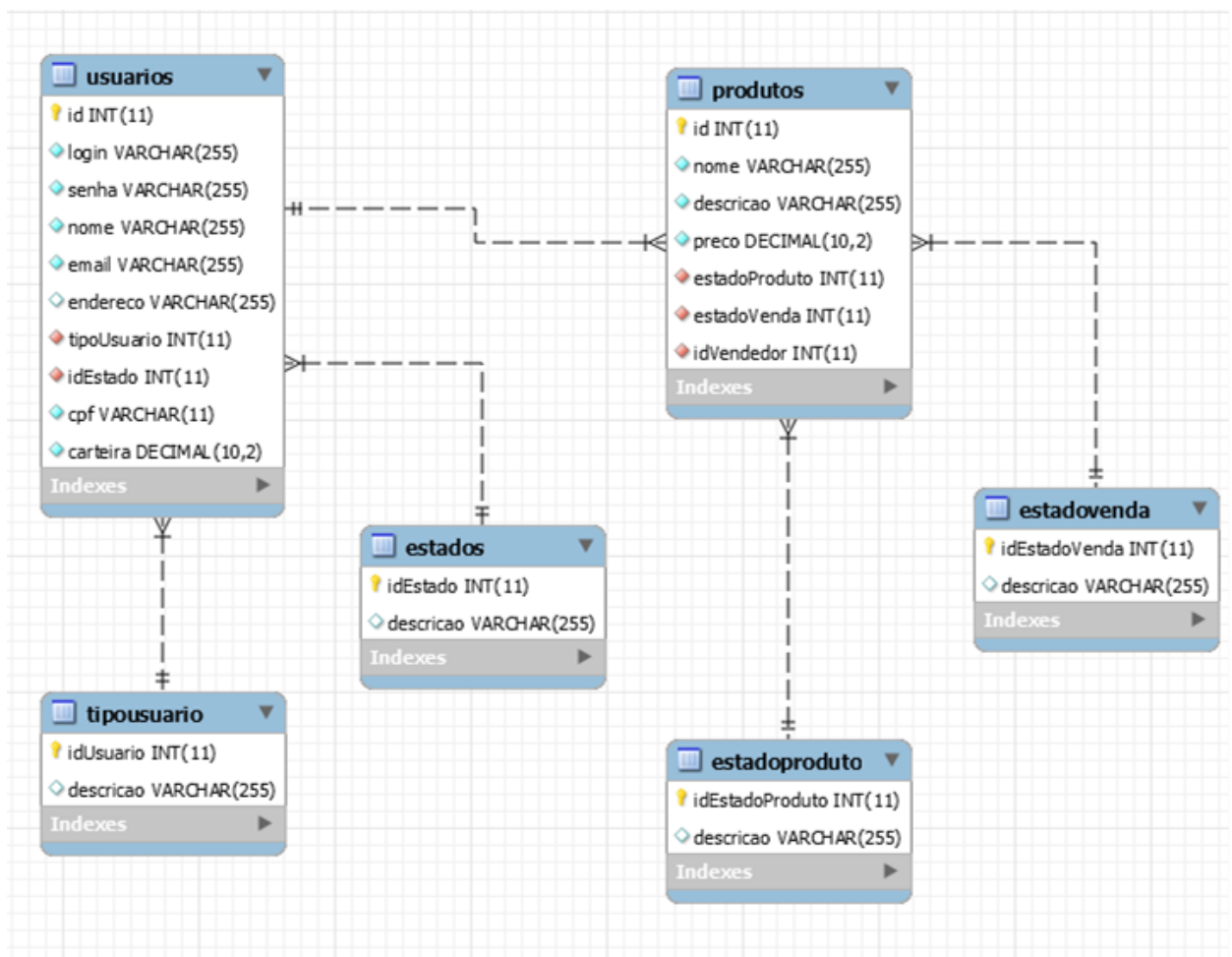
Esse projeto tem como objetivo a criação de uma Web Application em que será possível a venda e compra de mercadorias, de forma simplificada, entre usuários. A partir desse programa o usuário poderá logar no sistema, colocar seus produtos a venda, e comprar produtos de outros usuários.

Foi escolhido por desenvolver essa Web Application pois após uma análise das possibilidades a serem desenvolvidas, essa foi a de maior agrado, pois seria possível adquirir diversos conhecimentos ao desenvolvê-la. Além disso, facilitar o processo de venda e compra de produtos é algo de um grande interesse.

2 Desenvolvimento

O projeto foi desenvolvido de acordo com a evolução do curso de Programação para Web, em sua versão atual e final utiliza o framework Spring MVC. Este framework auxilia na criação e manutenção de projetos web dinâmicos. Na parte de View é utilizado o conjunto JSP, JavaScript, Bootstrap e CSS. Enquanto na parte de Controller e Model são utilizados Servlets e classes com conexões JDBC.

Para melhor explicação do funcionamento do projeto, primeiramente será apresentado nosso esquema de banco de dados, feito a partir do MySQL Workbench, depois uma breve descrição das páginas, juntamente com as classes.



Esquemático do Banco de Dados

A partir deste esquema é possível averiguar as chaves primárias (key) e chaves estrangeiras (vermelhas), além das relações entre as tabelas.

É possível também verificar se é um campo “not null” ou não, os campos “not null” possuem um losango preenchido antes do nome do atributo.

Para melhor entendimento do modelo, descreverei brevemente o papel de cada tabela.

- usuarios: Tabela responsável por armazenar os dados dos usuários em geral, nela são armazenados dados como login, senha, nome, email, endereço, estado e cpf. Além de armazenar a quantidade de dinheiro que o usuário possui na carteira.
 - estados: Tabela auxiliar, responsável por relacionar cada nome de estado a um respectivo id.
 - tipousuario: Tabela auxiliar, responsável por relacionar um tipo de usuário (administrador ou comum) a um respectivo id.
- produtos: Tabela responsável por armazenar os dados dos produtos anunciados pelos usuários, nela são armazenados dados como o nome, descrição do produto, preço, estado do produto (novo ou usado), estado da venda (a venda ou vendido), além do id do vendedor, que se relaciona com a tabela usuarios.
 - estadovenda: Tabela auxiliar, responsável por relacionar o estado de venda a um respectivo id.
 - estadoproduto: Tabela auxiliar, responsável por relacionar o estado do produto a um respectivo id.

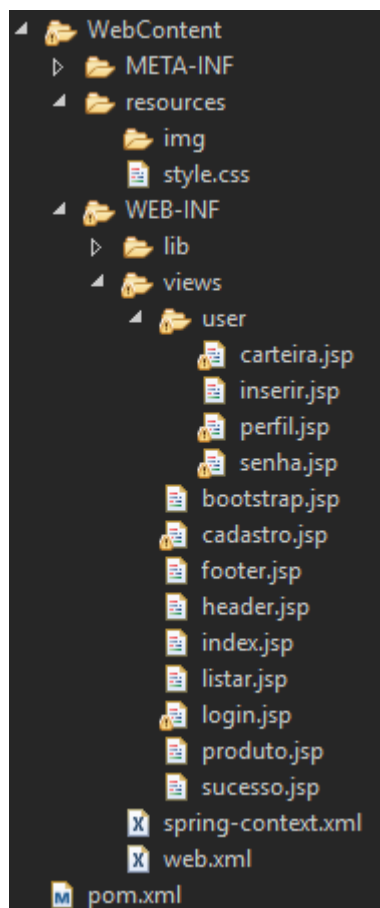
Explicadas as tabelas, agora poderemos nos aprofundar mais nas classes e páginas existentes dentro do projeto.

Após a modelagem do banco de dados, foi criado um escopo de como se transformaria a ideia de uma loja online para a prática. Primeiramente seria necessária a criação de classes que representassem nossas tabelas do banco de dados (Produto.java e Usuario.java), e então focamos na parte de View, do nosso MVC.

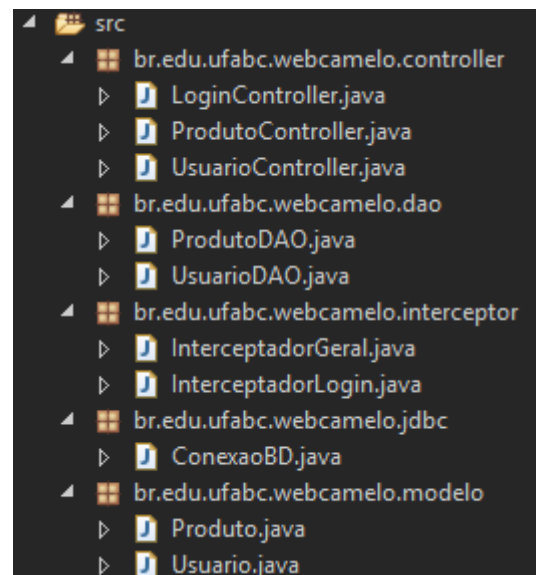
Todas as páginas do site possuem um menu, esse menu pode levar o usuário para as seguintes páginas:

- index: homepage do WebCamelô
- listar: listagem de produtos
- login: login no sistema
- cadastro: cadastro no sistema
- user/inserir: anunciar produtos
- user/carteira: carteira
- user/perfil: editar perfil

As páginas que possuem o caminho user na frente só poderão ser acessadas se o usuário estiver logado.



Visualização das páginas



Visualização das classes

Para filtrar o acesso do usuário no caminho user foi utilizado um interceptador (InterceptadorLogin.java) que verifica se o usuário está logado

somente nessas páginas, caso ele não esteja, o usuário é redirecionado para a página de login.

Além disso, foi utilizado outro interceptador para verificar e enviar informações do login do usuário entre as páginas (InterceptadorGeral.java).

Todas as informações que devem ser passadas do View são passadas por um Controller, e então manipuladas pelo Model, para que o mesmo possa retornar uma resposta para o Controller, e este para o View.

Sabendo disso, explicarei brevemente as funções de cada um dos Controllers neste projeto.

- LoginController.java: Responsável por repassar as informações de login para verificação no Model (UsuarioDAO.java).
- ProdutoController.java: Responsável por repassar informações de produtos, tais como compra, anúncio e venda para o Model (ProdutoDAO.java).
- UsuarioController.java: Responsável por repassar informações dos usuários, tais como edições no perfil, mudança de senha e alterações no fundo da carteira para o Model. (UsuarioDAO.java).

Ao receber cada uma das informações, nosso Model é responsável por manipulá-las, no nosso caso é necessário uma conexão com o banco de dados, que feita a partir da classe ConexaoBD.java e também pela pool de conexões gerenciada pelo HikariCP.

Para fazer tudo isso funcionar, primeiramente é necessário ter certeza de que a informação passada do View para o Controller esteja correta, para isso, em todas as páginas jsp são utilizados formulários validados com JavaScript.

Além disso, para o usuário se sentir confortável, foi utilizado também o Bootstrap juntamente com o CSS, fazendo com que o site seja intuitivo e confortável de se navegar, como podemos ver a seguir na parte de Simulação e Testes.

1 Simulação e Testes

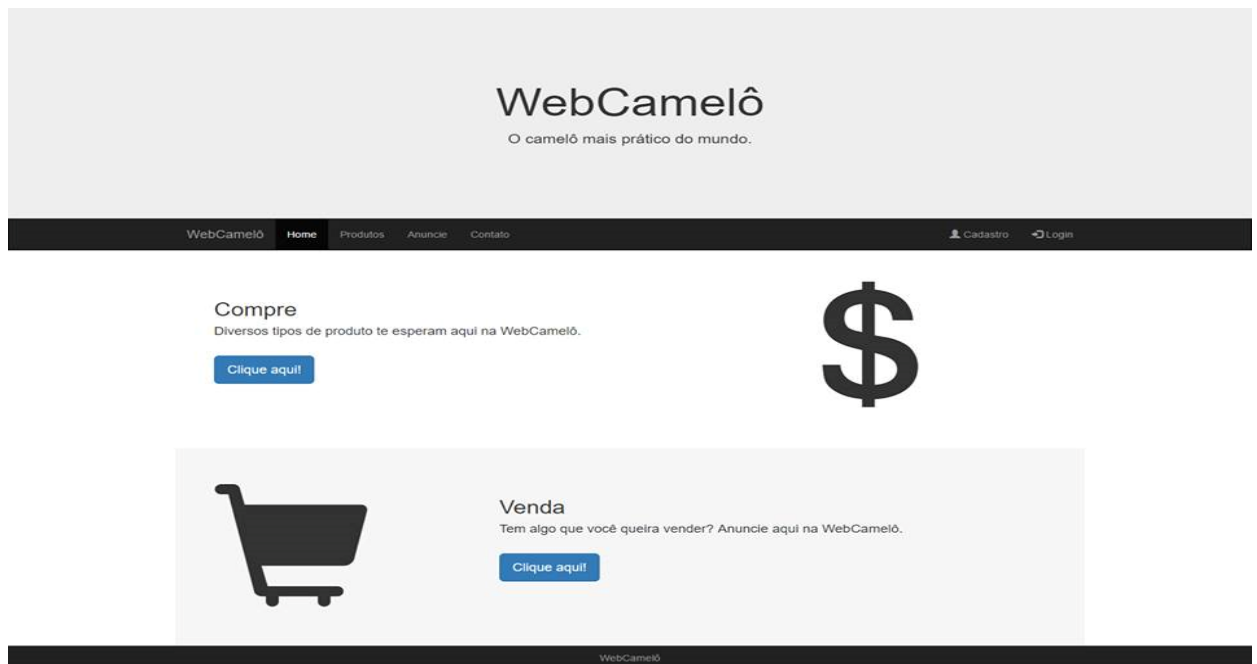


Figura 1

Após o usuário acessar a aplicação, irá se deparar com a tela inicial, mostrada na figura 1. Caso seja o primeiro acesso, o usuário deverá se cadastrar. Se o usuário já possuir cadastro, poderá apenas logar.



Figura 2



The image shows the registration page of the WebCamelô website. At the top, the logo 'WebCamelô' is displayed with the tagline 'O camêlô mais prático do mundo.' Below the logo is a navigation bar with links for 'WebCamelô', 'Home', 'Produtos', 'Ajuda', and 'Contato'. On the right side of the navigation bar are links for 'Cadastro' and 'Login'. The main content area is divided into two sections. The left section, titled 'Cadastro', contains a form with fields for 'Login:', 'Senha:', 'Confirmar:', 'Nome:', 'Email:', 'UF:', 'End:', and 'CPF:', each followed by a text input field. A 'Cadastrar' button is at the bottom of the form. The right section, titled 'Já é cadastrado?', contains a 'Faça o login' button and a piggy bank icon.

Figura 3

Ao clicar no link para se cadastrar, o usuário irá se deparar com um formulário, mostrado na figura 3, que deverá ser preenchido com os dados do usuário.



The image shows the user profile page of the WebCamelô website. At the top, the logo 'WebCamelô' is displayed with the tagline 'O camêlô mais prático do mundo.' Below the logo is a navigation bar with links for 'WebCamelô', 'Home', 'Produtos', 'Ajuda', and 'Contato'. On the right side of the navigation bar are links for 'Meu perfil', 'Sair', and 'Logout'. The main content area is divided into two sections. The left section, titled 'Meu perfil', contains a form with fields for 'Nome:', 'Login:', 'Senha:', 'CPF:', 'Email:', 'End:', and 'UF:', each followed by a text input field. A 'Salvar' button is at the bottom of the form. The right section contains a pencil icon.

Figura 4

Como ilustrado na figura 4, o usuário após ter seu cadastro feito poderá personalizar o seu perfil, caso necessite fazer futuras modificações.



Figura 5



Figura 6

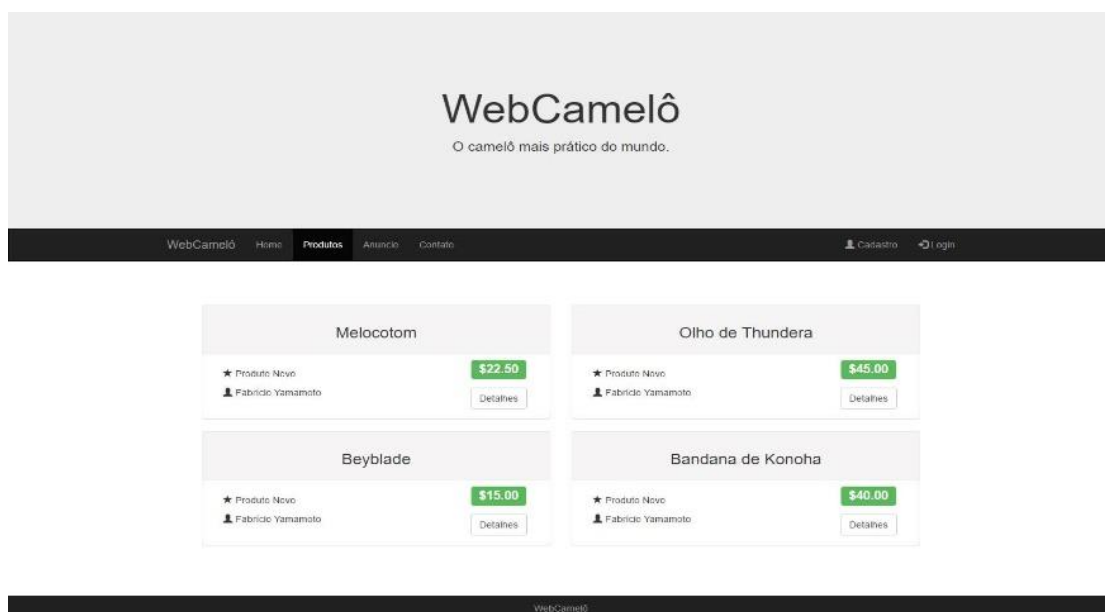
Para realizar suas compras, o usuário deverá inserir dinheiro em sua carteira, podendo assim efetivar suas compras.



The image shows the 'WebCamelô' website header with the logo and tagline 'O camêlô mais prático do mundo.'. Below the header is a navigation bar with links: WebCamelô, Home, Produtos, Anuncie, and Contato. On the right of the navigation bar, there is a user profile icon for 'fabricio.yamamoto', a balance of '\$100.00', and a 'Deslogar' button. The main content area features a section titled 'Anuncie seu produto!' with a sub-header 'Anuncie seu produto!'. Below this is a form with the following fields: 'Nome:' with a placeholder 'Digite o nome do produto', 'Descrição:' with a placeholder 'Descreva o produto' and a text area, 'Preço:' with a placeholder 'Digite o preço', and 'Estado:' with radio buttons for 'Novo' (selected) and 'Usado'. At the bottom of the form is an 'Anunciar' button. The footer of the page is a dark bar with the 'WebCamelô' logo.

Figura 7

Para anunciar seu produto, é necessário preencher um pequeno formulário com informações sobre seu produto, essas informações são Nome, descrição, preço e estado, como demonstrado na figura 7.



The image shows the 'WebCamelô' website header with the logo and tagline 'O camêlô mais prático do mundo.'. Below the header is a navigation bar with links: WebCamelô, Home, Produtos, Anuncie, and Contato. On the right of the navigation bar, there is a user profile icon for 'Cadastro' and a 'login' button. The main content area displays a grid of four product listings. Each listing includes the product name, a star icon, the status 'Produto Novo', the seller 'Fabricio Yamamoto', the price, and a 'Detalhes' button. The products and their prices are: Melocotom (\$22.50), Olho de Thundera (\$45.00), Beyblade (\$15.00), and Bandana de Konoha (\$40.00). The footer of the page is a dark bar with the 'WebCamelô' logo.

Figura 8

A sessão Produtos permite que o usuário veja todos os produtos anunciados, como demonstrado na figura 8.

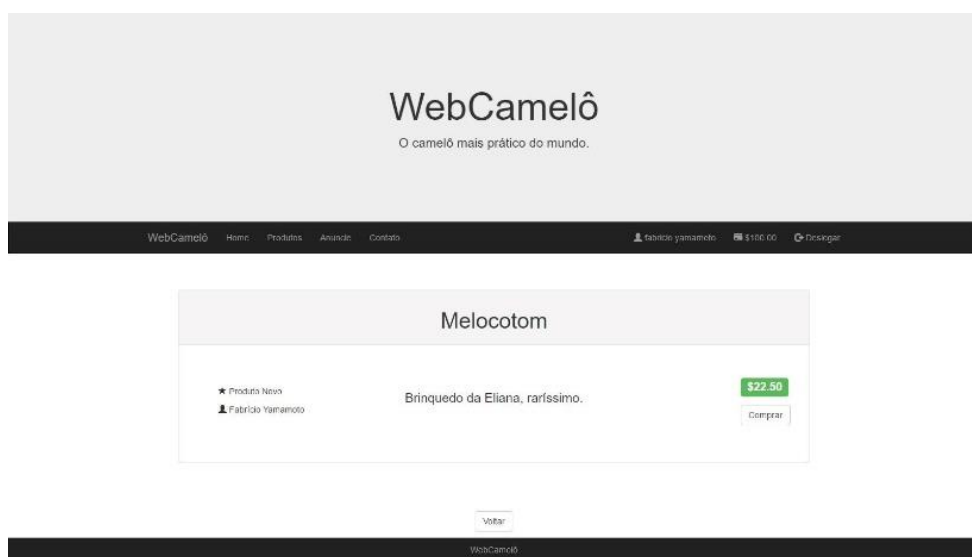


Figura 9

Como vemos na figura 9, ao selecionarmos um produto, o usuário irá para uma página que mostrará toda a descrição do produto, além da opção de Compra.



Figura 10

Depois de efetuada a compra, o usuário receberá uma mensagem em sua tela, como demonstra a figura 10. Assim que a compra é efetuada, o dinheiro é removido de sua carteira e adicionado a carteira do vendedor.

2 Conclusões

1

No começo do projeto foram encontradas dificuldades na criação e desenvolvimento do banco de dados, pois não possuíamos conhecimentos prévios do assunto, mas durante o desenvolvimento do projeto esses conhecimentos foram adquiridos, permitindo que o projeto fosse desenvolvido como o planejado.

Os mecanismos utilizados para desenvolver as classes de venda, compra, e carteira foram bem aprimorados, e tornaram a aplicação bastante acessível para os usuários, e de fácil manutenção de código, caso necessário.

O projeto poderia ser aperfeiçoado na personalização de usuário, podendo ser acrescentadas funções como imagens nos produtos e comentários sobre usuários, que auxiliariam outros utilizadores a selecionar qual seria a melhor opção de compra. Também poderia ser acrescentado ao projeto um sistema de mensagens, para que os compradores pudessem tirar suas dúvidas com o vendedor de forma rápida e fácil.

3 Referências Bibliográficas

Downey, T. Guide to Web Development with Java: Understanding Website Creation. Editora Springer London, 2012.

Sebesta, Robert W. Programming the World Wide Web, 8th edition. Pearson Addison Wesley, 2014.

Goncalves, Antonio. Beginning Java EE 7 (Expert Voice in Java). Apress, 2013.

Deitel, H. M.; Deitel, P.J. - Java Como Programar. 6ª. Edição. Editora PearsonPrentice Hall, 2005.

Ladd, S.; Davison, D.; Devijver, S.; Yates, C. Expert Spring MVC and Web Flow. Editora Apress, 2006.

Carnell, J.; Harrop, R. Pro Apache Struts with Ajax. Editora Apress, 2007.

Mak, G. Spring Recipes: A Problem-Solution Approach. Editora Apress, 2008.

Harrop, R.; Machacek, J. Pro Spring. Editora Apress, 2005.

Fields D.K., Kolb M.A. Desenvolvendo na Web com JavaServer Pages, Editora Ciência Moderna, 2000

Mecenas I. – Java 2 – Fundamentos, Swing e JDBC. Editora Alta Books, 2003.