



Trabalho Unidade 1

Valor: 5,0

Entrega: 16/04 (não será adiado)

Trabalho em duplas.

Crie um vídeo de demonstração mostrando o site funcionando para cada caso de uso. Envie o link do vídeo e do projeto no GITHUB através do SIGAA.

Descrição

Desenvolver uma aplicação Web usando banco de dados. O website deve contemplar funcionalidades básicas como cadastro de clientes, login de cliente, login de administrador, cadastro de produtos e carrinho de compras, de forma que todas essas informações serão armazenadas em um banco de dados. A figura 1 apresenta os casos de uso de uso que devem ser implementados. A seguir detalhes sobre cada um dos casos.

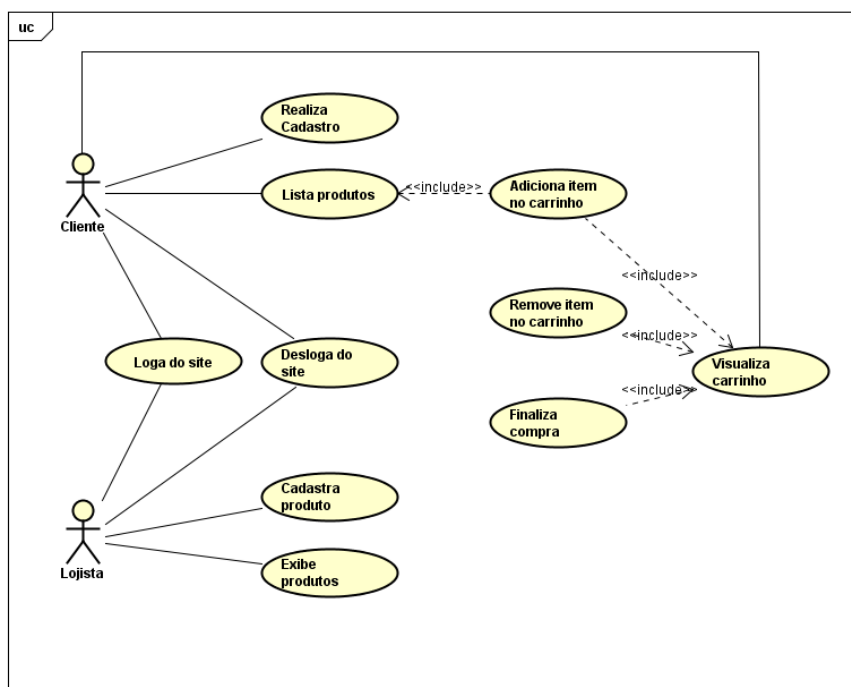


Figura 1. Casos de uso do website.

Realiza Cadastro: O cliente poderá fazer um cadastro no site informando nome, email e senha. O email serve como login no site. Uma lista prévia de clientes pode ser cadastrada no banco de dados conforme a tabela 1 (ver dica 1).

Tabela 1. Clientes.

Nome	Email	Senha
João Pedro	jp2017@uol.com.br	12345jaum
Amara Silva	amarasil@bol.com.br	amara82
Maria Pereira	mariape@terra.com.br	145aektm

Loga do Site e Desloga do site: Clientes e Logistas podem tentar se logar no site. Logar se corresponde ao ato de verificar que os dados de login e senha informados são compatíveis com o email e senha registrados. É preciso identificar qual dos clientes ou qual dos lojistas está autenticado. A aplicação web deverá ser capaz de identificar se o usuário autenticado é um cliente ou um logista e apresentar apenas os casos de uso respectivos de cada usuário. A ação de Deslogar do site deve destruir a sessão (HttpSession) do usuário autenticado. A tabela 2 apresenta uma lista de Logistas que poderão estar previamente cadastrados no banco de dados.

Tabela 2. Lojistas.

Nome	Email	Senha
Taniro Rodrigues	tanirocr@gmail.com	123456abc
Lorena Silva	lore_sil@yahoo.com.br	12uhuuu@

Lista Produtos: Aplicação web deverá apresentar uma lista de produtos que no banco de dados do servidor e associar a cada elemento dessa lista de produtos um **link** para adicionar (**Adicionar Item no Carrinho**) o produto no “Carrinho de compras” ver figura 2.

Lista Produtos				
Nome	Descrição	Preço	Estoque	Carrinho
Mesa	Uma mesa de computador	10	0	Sem estoque
Lapis	Lapis B2 grafite	15	0	Sem estoque
Caneta	Caneta BIC azul	7	1	Adicionar
Telefone	Telefone fixo antigo	8	1	Adicionar
Computador	Computador Intel	1000	1	Adicionar
Ver Carrinho				

Figura 2. ListaProdutosController

Visualiza Carrinho: A aplicação deverá manter também uma segunda página web que deve listar os produtos que foram adicionados ao “Carrinho de compras”, associando cada produto a um link para remover (**Remover item do carrinho**) ou adicionar (**Adicionar Item no Carrinho**) mais um produto do carrinho. Os itens adicionados ao carrinho de compras deverão ser armazenados em Cookies com tempo de duração de 48hs. Ou seja, se o cliente retornar ao website em menos de 48hs o seu carrinho de compras deverá ser restaurado. A figura 2 apresenta um exemplo de como deve ser a página para listar produtos (ListaProdutosController) e a figura 3 apresenta um exemplo da página do “Carrinho de compras” (VerCarrinhoController). Estes serão os únicos Controllers que retornam views (HTML) do sistema. Lembre-se que VerCarrinhoController só poderá ser exibido se o carrinho existir (se houverem Cookies), se um usuário tentar acessar VerCarrinhoController sem ter um carrinho, ele deve ser redirecionado (SendRedirect) para o Controller **ListaProdutosController**.

Lista Carrinho				
Nome	Descrição	Preço	Quantidade	Remover
Mesa	Uma mesa de computador	10	1	Remover
Lapis	Lapis B2 grafite	15	1	Remover
Ver Produtos				

Figura 3. VerCarrinhoController

Finaliza compra: Soma todos os produtos e calcula o total a ser pago. Atualiza estoque de produtos, removendo (update no banco) a quantidade necessária segundo a quantidade selecionada no carrinho de compras. Para simplificar, não é necessário manter um histórico de compras do cliente. Será necessário apenas deduzir o produto do estoque. Note que este será o único momento em que o estoque é alterado. Adicionar um item no carrinho não implica na remoção de um item do estoque.

Exibe produtos: O Lojista pode listar os produtos que estão cadastrados no momento e seu respectivo estoque. Ele não poderá colocar produtos em carrinho. A tabela 3 apresenta uma lista inicial de produtos que será inicialmente cadastrada no banco de dados.

Cadastrar produtos: O Lojista pode cadastrar novos produtos no website.

Tabela 3. Produtos.

Nome	Descrição	Preço Unitário	Quantidade
Mesa	Uma mesa de computador.	R\$ 500,00	10
Lápis	Lápis B2 grafite .	R\$ 2,00	50
Computador	Computador I5 16Gb de RAM.	R\$ 1.500,00	2

Dicas

DICA#1. Para implementar a adição e/ou remoção de itens do carrinho será necessário um outro **Controller** que trate a requisição e redirecione uma resposta válida. Por exemplo, considere na Figura 4 que queremos adicionar uma caneta (id de produto 3) ao carrinho. Observe o link que indica os parâmetros que **CarrinhoController**, o Controller que trata o carrinho, deve receber. São passados 2 parâmetros, o ID do produto que está sendo apontado e o comando (add na imagem) indicando que queremos adicionar tal produto ao carrinho (ver código). Após tratar a requisição, CarrinhoController faz um **Dispatcher** para a página que lista os produtos (ListaProdutosController).

Lista Produtos				
Nome	Descrição	Preço	Estoque	Carrinho
Mesa	Uma mesa de computador	10	0	Sem estoque
Lapis	Lapis B2 grafite	15	0	Sem estoque
Caneta	Caneta BIC azul	7	1	Adicionar
Telefone	Telefone fixo antigo	8	1	Adicionar
Computador	Computador Intel	1000	1	Adicionar

[Ver Carrinho](#)

<http://localhost:8080/LojaOnLine/CarrinhoServlet?id=3&comando=add>

Figura 4. Ao manter o mouse sobre o botão adicionar de Caneta, um link é exibido contendo o ID de caneta e o comando "add".

```
String command = request.getParameter("comando");
if (command.equals("add")){
    //adicionar ao carrinho
    //remover do estoque
}else if (command.equals("remove")){
    //remover do carrinho
    //adicionar ao estoque
}
```

Código-fonte 2. Implementação da verificação se é para adicionar ou remover um produto do carrinho.

Carrinho.java

```
package classes;

import java.util.ArrayList;

public class Carrinho {

    public Carrinho(ArrayList<Produto> produtos) {
        super();
        this.produtos = produtos;
    }

    ArrayList<Produto> produtos;

    public ArrayList<Produto> getProdutos() {
        return produtos;
    }

    public void setProdutos(ArrayList<Produto> produtos) {
        this.produtos = produtos;
    }

    public Produto getProduto (int id){

        Produto mp = null;

        for (Produto p : produtos){
            if (p.getId() == id){
                return p;
            }
        }
        return mp;
    }

    public void removeProduto (int id){

        Produto p = getProduto(id);
        produtos.remove(p);
    }

}
```

```
        public void addProduto (Produto p){  
            produtos.add(p);  
        }  
    }  
}
```

Produto.java

```
package classes;  
  
public class Produto {  
  
    public Produto(int id, int preco, String nome, String descricao, int estoque) {  
        super();  
        this.id = id;  
        this.preco = preco;  
        this.nome = nome;  
        this.Descricao = descricao;  
        this.estoque = estoque;  
    }  
    int id;  
    int preco;  
    String nome;  
    String Descricao;  
    int estoque;  
  
    public int getId() {  
        return id;  
    }  
    public void setId(int id) {  
        this.id = id;  
    }  
    public int getPreco() {  
        return preco;  
    }  
    public void setPreco(int preco) {  
        this.preco = preco;  
    }  
    public String getNome() {  
        return nome;  
    }  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
    public String getDescricao() {  
        return Descricao;  
    }  
    public void setDescricao(String descricao) {  
        Descricao = descricao;  
    }  
    public int getEstoque() {  
        return estoque;  
    }  
    public void incrementaEstoque() {  
        this.estoque++;  
    }  
    public void diminuiEstoque() {  
        this.estoque--;  
    }  
}
```