Exercício 5 - CRUD

Este exercício deve ser feito como **alteração do Exercício** feito anteriormente. As alterações a serem adicionadas estão realçadas e as alterações a serem removidas estão ~~tachadas~~.

Agora todas as classes Java deverão estar dentro dos pacotes conforme indicado nas descrições abaixo.

Deve-se usar o banco de dados já criado contendo uma tabela chamada **tb_usuario** contendo os campos **id_usuario** (PK, inteiro), **login_usuario** (VARCHAR(50)), **senha_usuario** (VARCHAR(50) e **nome_usuario** (VARCHAR(100)).

Deve-se ter pelo menos 3 usuários nessa tabela. Esta tabela será usada para efetuar login no sistema.

Deve-se adicionar uma tabela chamada **tb_cliente** contendo os campos **id_cliente** (PK, inteiro), **cpf_cliente** (CHAR(11)), **nome_cliente** (VARCHAR(100)), **email_cliente** (VARCHAR(100), **data_cliente** (DATE), **rua_cliente** (VARCHAR(100)), **nr_cliente** (INTEGER), **cep_cliente** (CHAR(8)), **cidade_cliente** (VARCHAR(100)), **uf_cliente** (CHAR(2)).

Adicionar pelo menos 10 clientes nessa tabela.

Link para básico do BOOTSTRAP:
https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp

1) index.html deve ser substituída por index.jsp

Alterações:

- Deve-se incluir BOOTSTRAP para formatação desta página
- Deve receber um parâmetro no escopo da requisição (chamado "msg"). Se este parâmetro for passado, deve ser apresentado em vermelho, como uma mensagem de aviso, junto com a página de login. Se não for passado, somente a página de login é apresentada.

No rodapé da página deve ser colocada a seguinte mensagem: **"Em caso de problemas contactar o administrador: "** juntamente com o e-mail armazenado no escopo da aplicação.

2) com.ufpr.tads.web2.servlets.StartupServlet

É uma servlet que deve ser inicializada no Startup da aplicação. Dentro do seu método **init()**, deve-se armazenar no escopo da aplicação (ServletContext) uma instância preenchida de um ConfigBean, com o nome "configuracao".

3) com.ufpr.tads.web2.servlets.LoginServlet

É uma servlet que verifica o login do usuário.
Se o par login/senha estiver no banco de dados:

- Preenche uma instância de LoginBean com os dados do usuário
- Armazena o LoginBean na sessão (indicando que o usuário está logado);
- Redireciona para a servlet **portal.jsp**.

Se o par login/senha não estiver no banco de dados:

- ~~Redireciona para **erro.jsp** (via *forward*), passando como parâmetro a mensagem de erro a ser apresentada (parâmetro "msg") e o nome da página para retornar (parâmetro "page", que no caso será "index.html")~~
- Redireciona para **index.jsp** (via *forward*), passando como parâmetro a mensagem de erro a ser apresentada (parâmetro "msg") "**Usuário/Senha inválidos."**

4) PortalServlet deve ser substituída por uma JSP : portal.jsp

Alterações:

- Deve-se incluir BOOTSTRAP para formatação desta página

A primeira coisa que **portal.jsp** faz é verificar se o usuário está logado.

- Abra um código scriptlet (<% … %>)
- O JSP já possui um objeto **session** instanciado, basta acessá-lo
- Se não estiver logado:
    - ~~Redireciona para **erro.jsp** (via *forward*), passando como parâmetro a mensagem de erro a ser apresentada (parâmetro "msg") e o nome da página para retornar (parâmetro "page", que no caso será "index.html")~~
    - Redireciona para **index.jsp** (via *forward*), passando como parâmetro a mensagem de erro a ser apresentada (parâmetro "msg") "**Usuário deve se autenticar para acessar o sistema."**
- Se o usuário estiver logado:
    - Obtenha o objeto **LoginBean** da sessão usando **<jsp:useBean />**
    - Apresenta uma tela com o nome do usuário logado
    - Deve-se incluir um MENU contendo as opções:
        - **Cadastro de Clientes**: que direciona para a servlet **ClientesServlet**

        - **Sair**: que direciona para a servlet **LogoutServlet**
    - ~~Nesta tela apresente um link para **LogoutServlet**~~

No rodapé da página deve ser colocada a seguinte mensagem: **"Em caso de problemas contactar o administrador: "** juntamente com o e-mail armazenado no escopo da aplicação.

5) com.ufpr.tads.web2.servlets.LogoutServlet

~~Efetua o logout do usuário e apresenta uma tela contendo:~~

- ~~Uma mensagem indicando que o usuário saiu do sistema~~
- ~~Um link para **index.html**~~

Efetua o logout do usuário e direciona, via **forward** para **index.jsp**, passando como parâmetro a mensagem **"Usuário desconectado com sucesso"**

6) ErroServlet deve ser substituída por uma JSP : erro.jsp

Esta JSP recebe dois parâmetros no escopo da requisição: "msg" e "page".

Apresenta a mensagem de erro passada como parâmetro (parâmetro de nome "msg") no request e um link para o recurso cujo nome também foi passado como parâmetro (parâmetro de nome "page").

No rodapé da página deve ser colocada a seguinte mensagem: "Em caso de problemas contactar o administrador: " juntamente com o e-mail armazenado no escopo da aplicação.

Alterações:

- Deve-se incluir BOOTSTRAP para formatação desta página

7) com.ufpr.tads.web2.servlets.ClientesServlet

Inicialmente verifica se o usuário está logado. Se não estiver, redireciona via **forward** para **index.jsp** passando como parâmetro a mensagem **"Usuário deve se autenticar para acessar o sistema"** (parâmetro "msg").

Esta servlet busca todos os clientes da base de dados, como um **List<Cliente>**, e os coloca no escopo da requisição. Em seguida, efetua **forward** para **clientesListar.jsp**.

8) com.ufpr.tads.web2.beans.Cliente

Um Java Bean que contém os dados da tabela **tb_cliente**. Este bean deve ser usado juntamente com as classes DAO para acessar o banco de dados.

9) clientesListar.jsp

Inicialmente verifica se o usuário está logado. Se não estiver, redireciona via **forward** para **index.jsp** passando como parâmetro a mensagem **"Usuário deve se autenticar para acessar o sistema"** (parâmetro "msg").

Esta tela apresenta o mesmo formato que **portal.jsp**, alterando somente seu conteúdo. Deve apresentar, em forma de tabela, os dados de Cliente: CPF, Nome e E-mail. Adicionalmente, em cada linha, deve-se apresentar 3 pequenas imagens representando as ações: Visualizar, Alterar e Remover; com os seguintes links:

- Visualizar: para a Servlet **VisualizarClienteServlet?id=10** (onde id=10 é o id do cliente naquela linha)
- Alterar: para a Servlet **FormAlterarClienteServlet?id=10** (idem)
- Remover: para a Servlet **RemoverClienteServlet?id=10** (idem)

Esta tela também deve apresentar um botão de Novo, com um link para a servlet **FormNovoClienteServle**t.

10) com.ufpr.tads.web2.servlets.VisualizarClienteServlet

Inicialmente verifica se o usuário está logado. Se não estiver, redireciona via **forward** para **index.jsp** passando como parâmetro a mensagem **"Usuário deve se autenticar para acessar o sistema"** (parâmetro "msg").

Executa os seguintes passos:

- Recebe um parâmetro (via parameter, GET) com o Id do cliente a ser mostrado
- Busca no banco de dados o objeto do cliente
- Adiciona o objeto no escopo da requisição
- Efetua forward para **clientesVisualizar.jsp**

11) com.ufpr.tads.web2.servlets.RemoverClienteServlet

Inicialmente verifica se o usuário está logado. Se não estiver, redireciona via **forward** para **index.jsp** passando como parâmetro a mensagem **"Usuário deve se autenticar para acessar o sistema"** (parâmetro "msg").

Executa os seguintes passos:

- Recebe um parâmetro (via parameter, GET) com o Id do cliente a ser removido
- Remove o cliente do banco de dados
- Efetua forward para **ClientesServlet**

12) com.ufpr.tads.web2.servlets.FormAlterarClienteServlet

Inicialmente verifica se o usuário está logado. Se não estiver, redireciona via **forward** para **index.jsp** passando como parâmetro a mensagem **"Usuário deve se autenticar para acessar o sistema"** (parâmetro "msg").

Executa os seguintes passos:

- Recebe um parâmetro (via parameter, GET) com o Id do cliente a ser alterado
- Busca no banco de dados o objeto do cliente
- Adiciona o objeto no escopo da requisição
- Efetua forward para **clientesAlterar.jsp**

13) clientesAlterar.jsp

Inicialmente verifica se o usuário está logado. Se não estiver, redireciona via **forward** para **index.jsp** passando como parâmetro a mensagem **"Usuário deve se autenticar para acessar o sistema"** (parâmetro "msg").

Deve apresentar um formulário com todos os dados do cliente, já preenchidos com os dados recebidos via request, um botão **Alterar** e um botão **Cancelar**.

Ao pressionar **Alterar**, deve-se submeter para **AlterarClienteServlet**. Ao pressionar **Cancelar**, deve-se direcionar para **ClientesServlet**.

14) clientesVisualizar.jsp

Inicialmente verifica se o usuário está logado. Se não estiver, redireciona via **forward** para **index.jsp** passando como parâmetro a mensagem **"Usuário deve se autenticar para acessar o sistema"** (parâmetro "msg").

Deve apresentar todos os dados de um cliente que foi passado no escopo da requisição e um botão **Voltar**.

Ao pressionar **Voltar**, deve-se submeter para **ClientesServlet** para apresentar a lista total de clientes.

15) com.ufpr.tads.web2.servlets.AlterarClienteServlet

Inicialmente verifica se o usuário está logado. Se não estiver, redireciona via **forward** para **index.jsp** passando como parâmetro a mensagem **"Usuário deve se autenticar para acessar o sistema"** (parâmetro "msg").

Recebe todos os dados do cliente passados via POST pelo formulário e realiza uma atualização na base de dados. Então, efetua um redirecionamento para **ClientesServlet**.

16) com.ufpr.tads.web2.servlets.FormNovoClienteServlet

Inicialmente verifica se o usuário está logado. Se não estiver, redireciona via **forward** para **index.jsp** passando como parâmetro a mensagem **"Usuário deve se autenticar para acessar o sistema"** (parâmetro "msg").

Efetua um redirecionamento para **clientesNovo.jsp**.

17) clientesNovo.jsp

Inicialmente verifica se o usuário está logado. Se não estiver, redireciona via **forward** para **index.jsp** passando como parâmetro a mensagem **"Usuário deve se autenticar para acessar o sistema"** (parâmetro "msg").

Deve apresentar um formulário com todos os dados do cliente (limpos para uma adição), um botão **Salvar** e um botão **Cancelar**. Ao pressionar Salvar, deve-se submeter para **NovoClienteServlet**. Ao pressionar Cancelar, deve-se direcionar para **ClientesServlet**.

18) com.ufpr.tads.web2.servlets.NovoClienteServlet

Inicialmente verifica se o usuário está logado. Se não estiver, redireciona via **forward** para **index.jsp** passando como parâmetro a mensagem **"Usuário deve se autenticar para acessar o sistema"** (parâmetro "msg").

Recebe todos os dados do cliente passados via POST pelo formulário e realiza uma inserção na base de dados. Então, efetua um redirecionamento para **ClientesServlet**.

19) com.ufpr.tads.web2.beans.LoginBean

Um Java Bean que contém os dados de login a serem armazenados na sessão. Contém o id do usuário, e o nome do usuário.

20) com.ufpr.tads.web2.beans.ConfigBean

Um Java Bean que contém dados de configuração da aplicação, no caso o e-mail do administrador.

21) com.ufpr.tads.web2.beans.Usuario

Um Java Bean que contém os dados da tabela **tb_usuario**. Este bean deve ser usado juntamente com as classes DAO para acessar o banco de dados.

22) com.ufpr.tads.web2.dao.UsuarioDAO

É a classe DAO de acesso do usuário ao banco de dados.

23) com.ufpr.tads.web2.dao.ConnectionFactory

É a classe que controla a conexão com o banco de dados.

24) com.ufpr.tads.web2.dao.ClienteDAO

É a classe DAO de acesso do usuário ao banco de dados para manipular os dados de clientes.