DR3

Disciplina Regular 3

Desenvolvimento Web com Java EE

Graduação em Engenharia de Software - 2020

DR3

Etapa 6 Aula 1

Segurança de Aplicações Web

Para acompanhar pelo Moodle você deve estudar a etapa 5

DR3

Etapa 6 Aula 1 - Competências

- **Competências** Trabalhadas Nesta Etapa:
  - **Estabelecer o gerenciamento de estado em aplicações Java Web**:
    - Compreender os conceitos sobre sessão e seu ciclo de vida**.**
    - Implementar Autenticação e Autorização declarativa (JEE).

DR3

Etapa 6 Aula 1 - Checklist

- **Checklist DR3:**
  - Ter revisado os slides e **os exercícios**, principalmente **JPA** e **CRUDRepository**.
  - Ter implementado exemplos de CRUD com os relacionamentos 1-N e N-M.
    - Construção de telas com JSP + JSTL e EL.
    - Construção de Controllers com validações do Spring MVC.
    - Construção de Repositories do JPA com inserir, alterar, excluir e as consultas.

DR3

Gerenciamento de Sessão

Gerenciamento de Sessão

DR3

- O protocolo Http é desconectado, ou seja, após o envio da resposta, a conexão é fechada.
- No entanto, é comum o acesso a várias páginas que compõem uma mesma operação (caso de uso), como uma operação de compra.
- Uma sessão (conversa) é o conjunto de acessos (requisições e respostas) interligados.

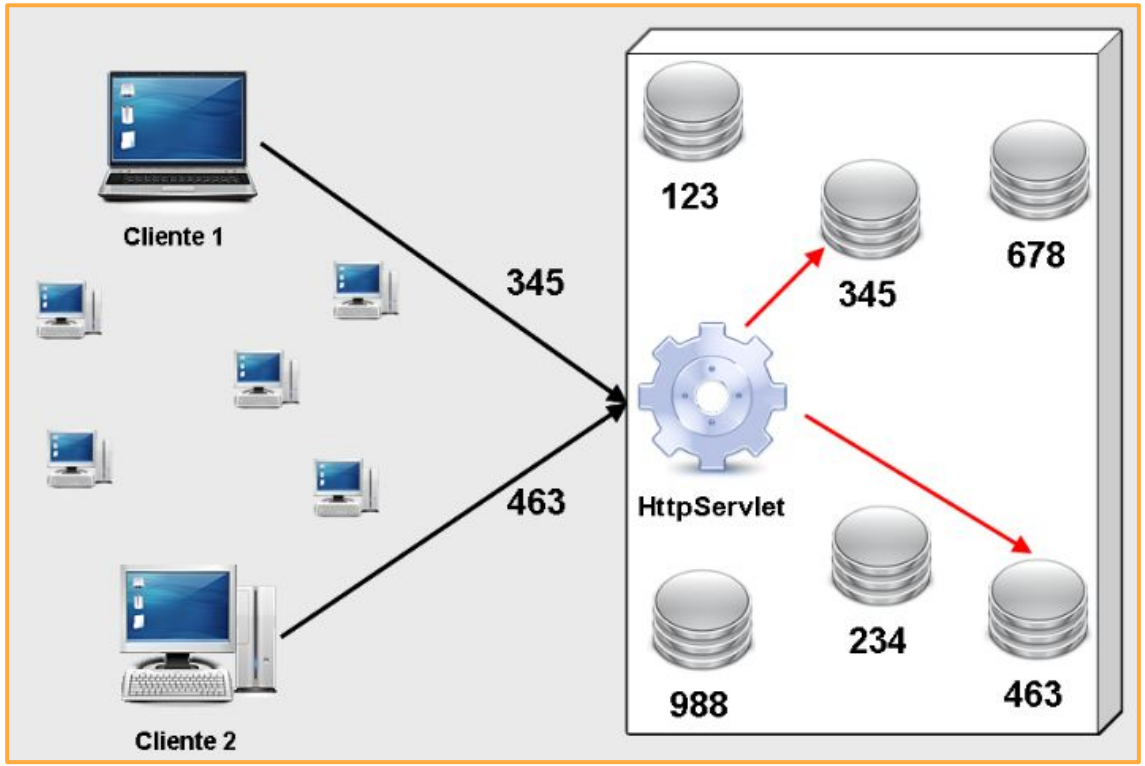
Gerenciamento de Sessão

DR3

- Para que seja possível criar sessões no protocolo Http é necessário verificar a qual usuário pertence cada requisição que chega ao servidor.
- A solução consiste em:
  - Associar cada requisição recebida com uma sessão existente no servidor.
  - Se for uma nova sessão, gerar um identificador único.
  - Recuperar dados da sessão (produtos comprados, por exemplo) de um conjunto (lista, tabela hash etc.).
  - Enviar a identificação da sessão para o cliente.

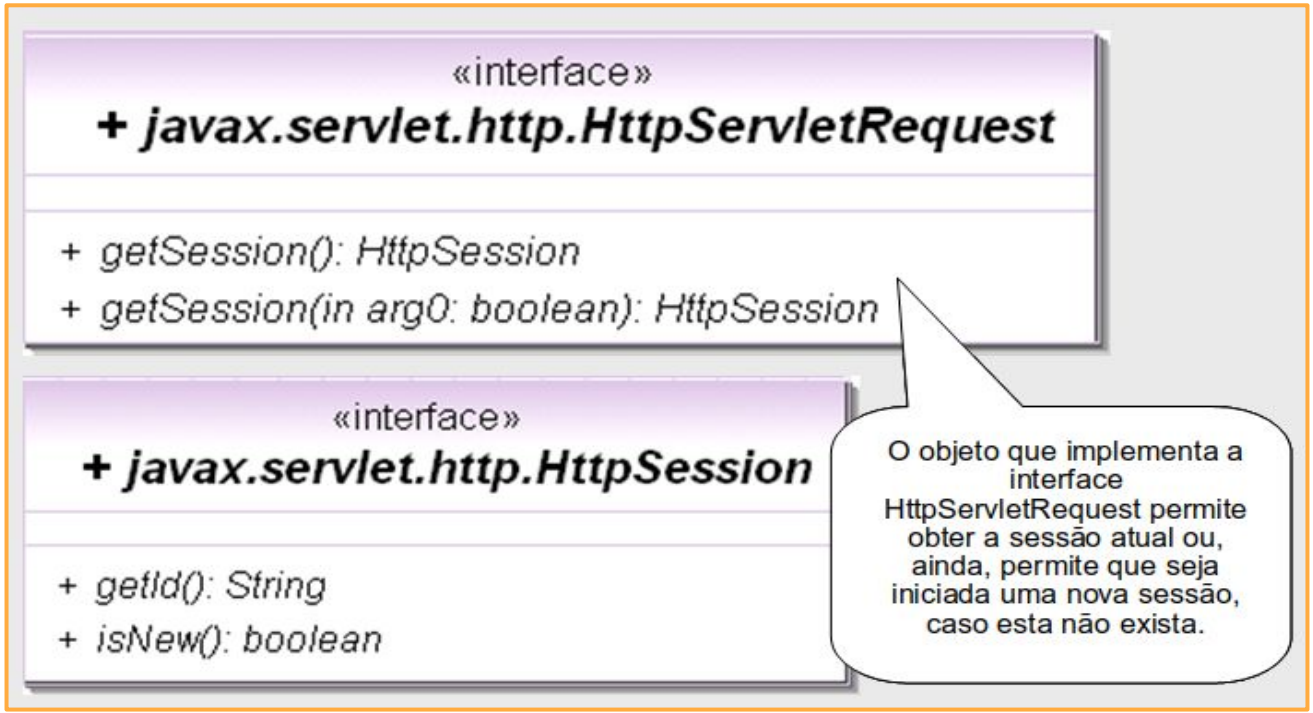
DR3

Sessões no Container



O container passa para o servlet a “sessão” correspondente a cada cliente toda vez que for realizada uma requisição.

Interface HttpSession



Usando Sessões

DR3

- Para recuperar ou criar uma sessão usa-se o método getSession do objeto request:

```
HttpSession session = request.getSession();
```

- Esse método retorna a sessão associada com a requisição atual ou cria uma nova, se for necessário.
- Para verificar se uma sessão acabou de ser criada é usado o método isNew.

Atributos de Sessão

- Atributos com este tipo de escopo podem ser acessados apenas pelas requisições que pertençam à sessão onde estão armazenados.
- O uso de atributos de sessão é similar aos de requisição e de aplicação:

Exemplo

- Quando o usuário se autentica no sistema, sua conta é armazenada na sessão:

```
HttpSession sessao = request.getSession();
   if (sessao.isNew()) {
      sessao.setAttribute("login", dadosLogin);
   }
```

Finalização da Sessão

- Uma sessão pode ser finalizada pela chamada ao método **invalidate** ou por inatividade.
- Para alterar o tempo de inatividade usa-se o método **setMaxInactiveInterval**.

Sessão e Contexto em JSP

DR3

- Não é necessário criar a sessão: toda página JSP tem, por default, uma sessão.

- Para inserir ou recuperar objetos basta indicar o escopo desejado:

```
<jsp:useBean scope="session" id="" class="" />
  <jsp:useBean scope="application"  ... />
```

Sessão e Contexto em JSTL + EL

DR3

- Para acessar os objetos de sessão e contexto:

```
pageContext.session
pageContext.servletContext
```

- Para alterar ou remover um atributo da sessão ou do contexto usam-se as tags da biblioteca core de JSTL:

```
set
remove
```

Exercício

DR3

- Crie um projeto web com duas páginas: index.jsp e index_1.jsp.
  - Uma página contém um link para a outra.
  - Usando EL, mostre o ID da sessão e se ela é nova ou não, em ambas as páginas.
  - Crie um servlet LogoutServlet que feche a sessão atual e redirecione para o index.jsp.
  - Crie um link no index.jsp para invocar o servlet.

DR3

Segurança Declarativa

DR3

Autenticação & Autorização

- **Autenticação** é a funcionalidade do sistema que permite estabelecer a identidade de um usuário ou instituição.

- No Tomcat o mecanismo de autenticação é implementado através de uma janela do próprio browser ou através de um formulário criado pelo desenvolvedor.

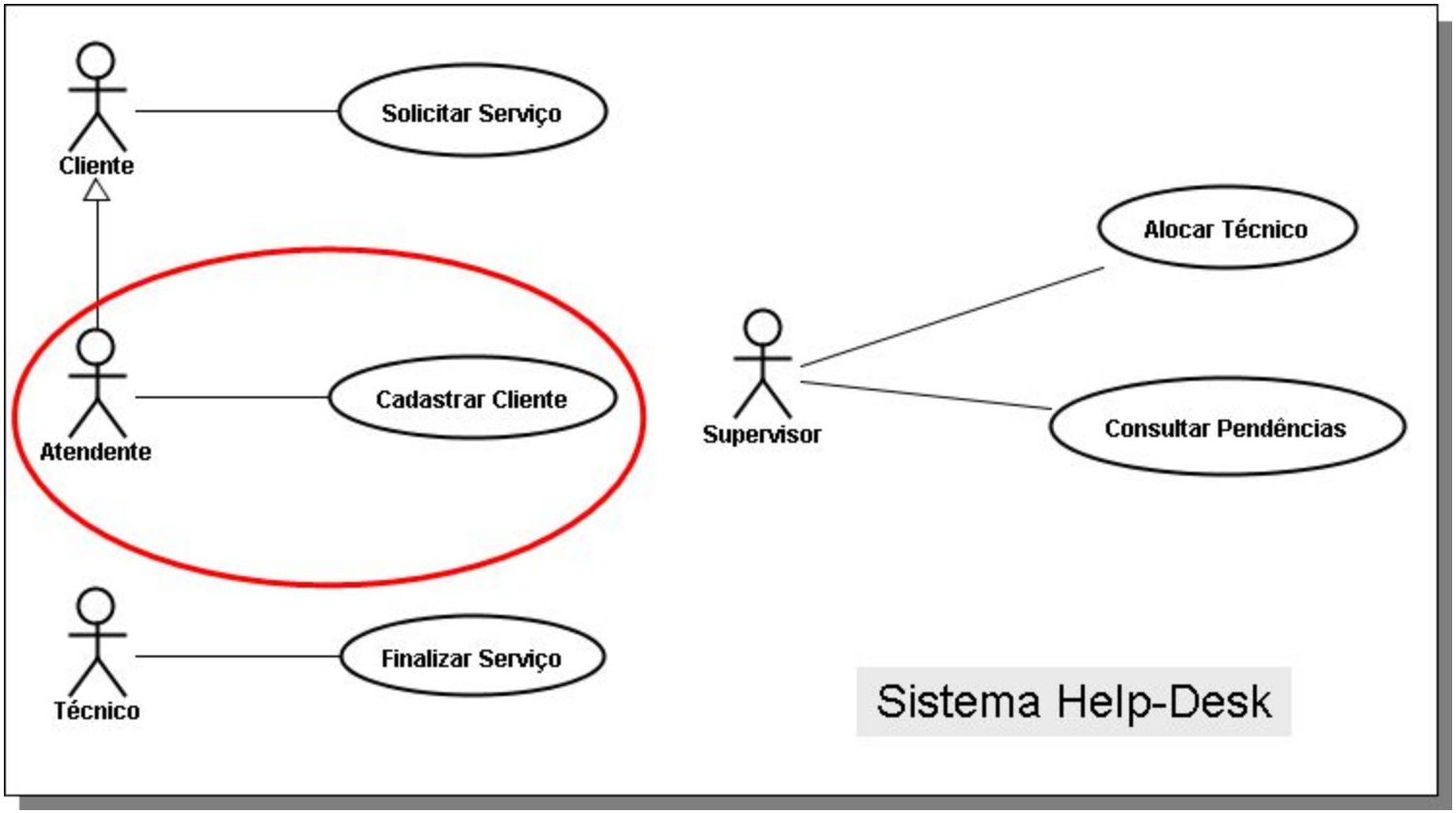
- A configuração do mecanismo de autenticação é declarativa através do **Deployment Descriptor**.

DR3

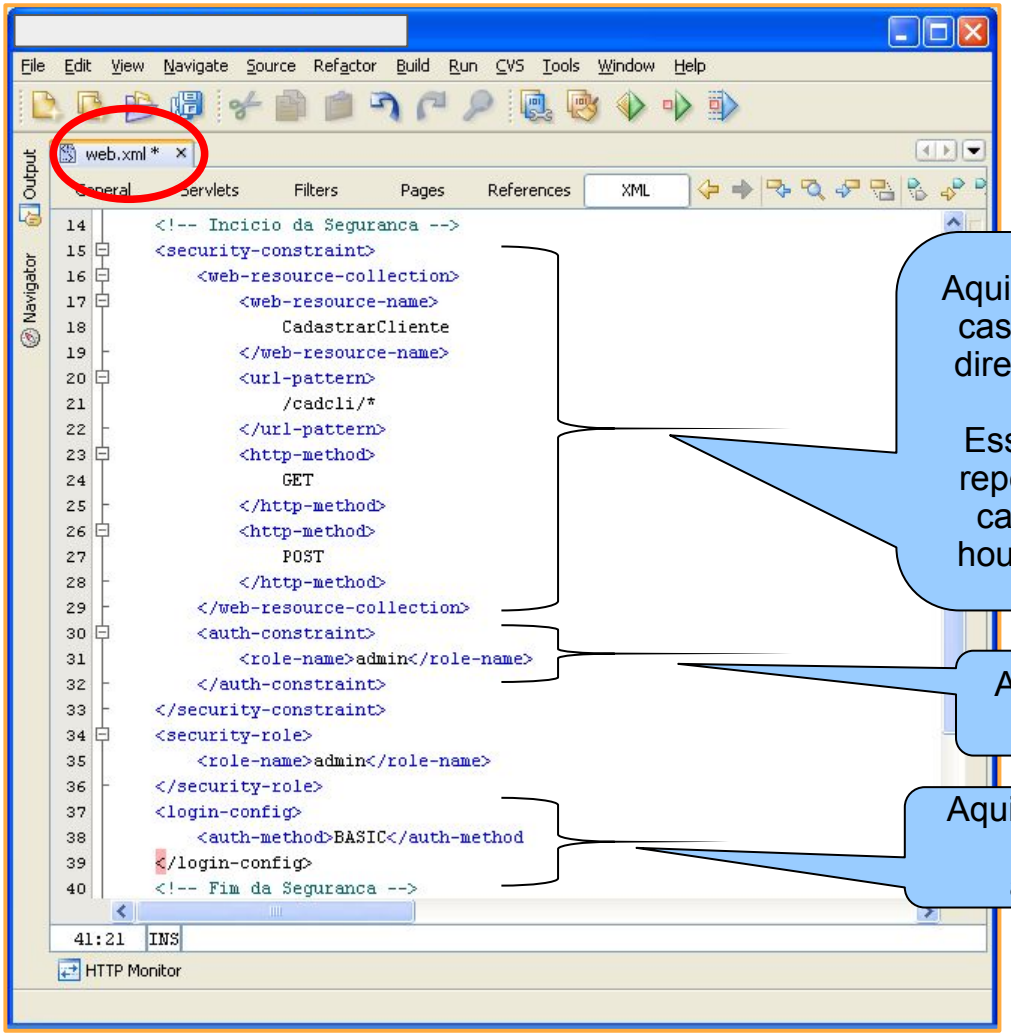
Autenticação & Autorização

- **Autorização** é a funcionalidade de associar os recursos da aplicação aos usuários.
- Devemos usar o modelo UML de **atores X casos de uso** a fim de facilitar a implementação da autorização (RBAC = Role Based Access Control).
- Os atores são os grupos de usuários e os casos de uso as funcionalidades.
- Devemos criar diretórios para colocar os arquivos dos casos de uso. É necessário também refletir essa estrutura de diretório nos mapeamentos dos controllers.

DR3


Cliente
Solicitar Serviço
Atendente
Cadastrar Cliente
Técnico
Finalizar Serviço
Supervisor
Alocar Técnico
Consultar Pendências
Sistema Help-Desk

DR3

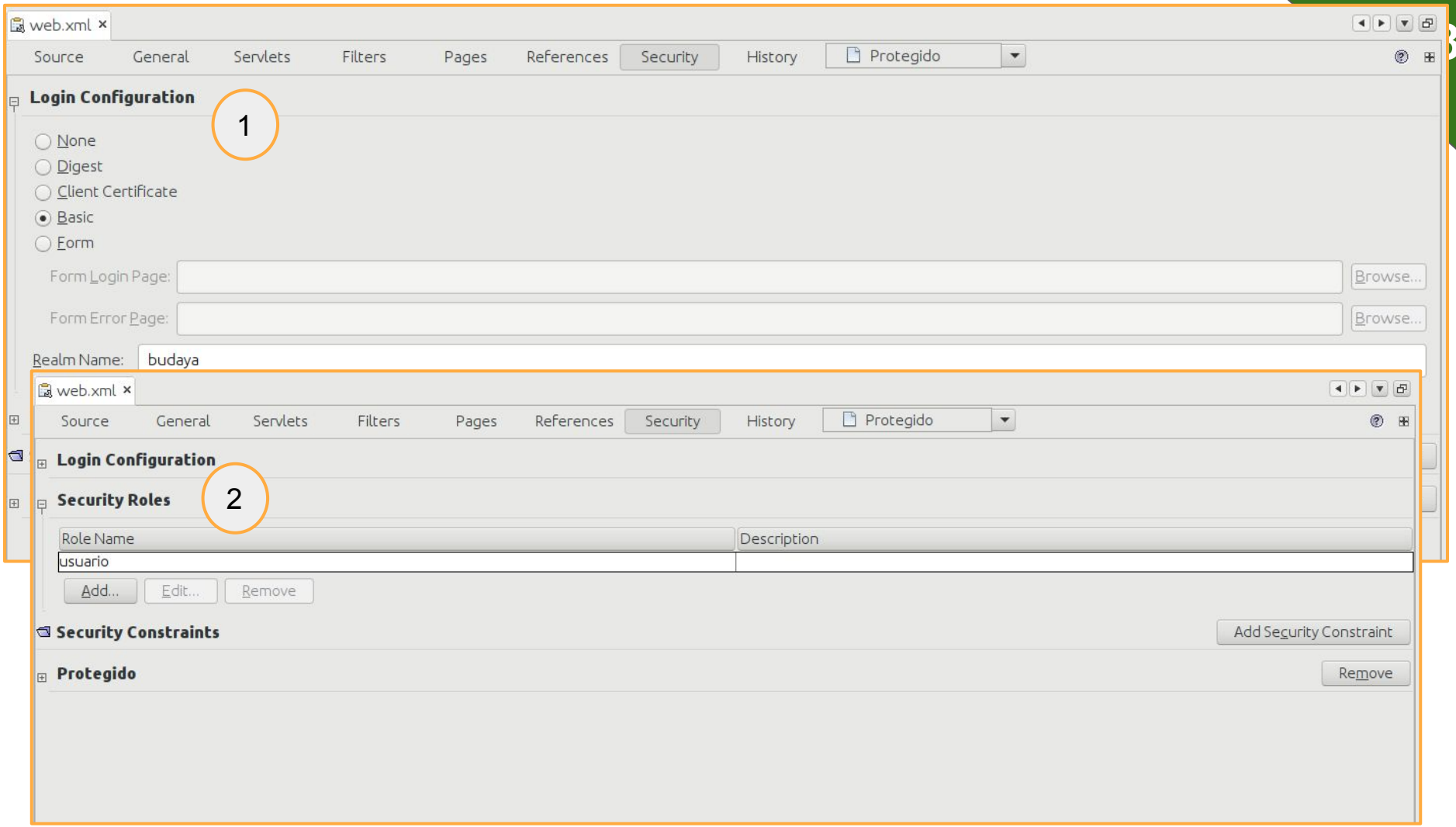


Aqui cadastramos os casos de uso – um diretório para cada.

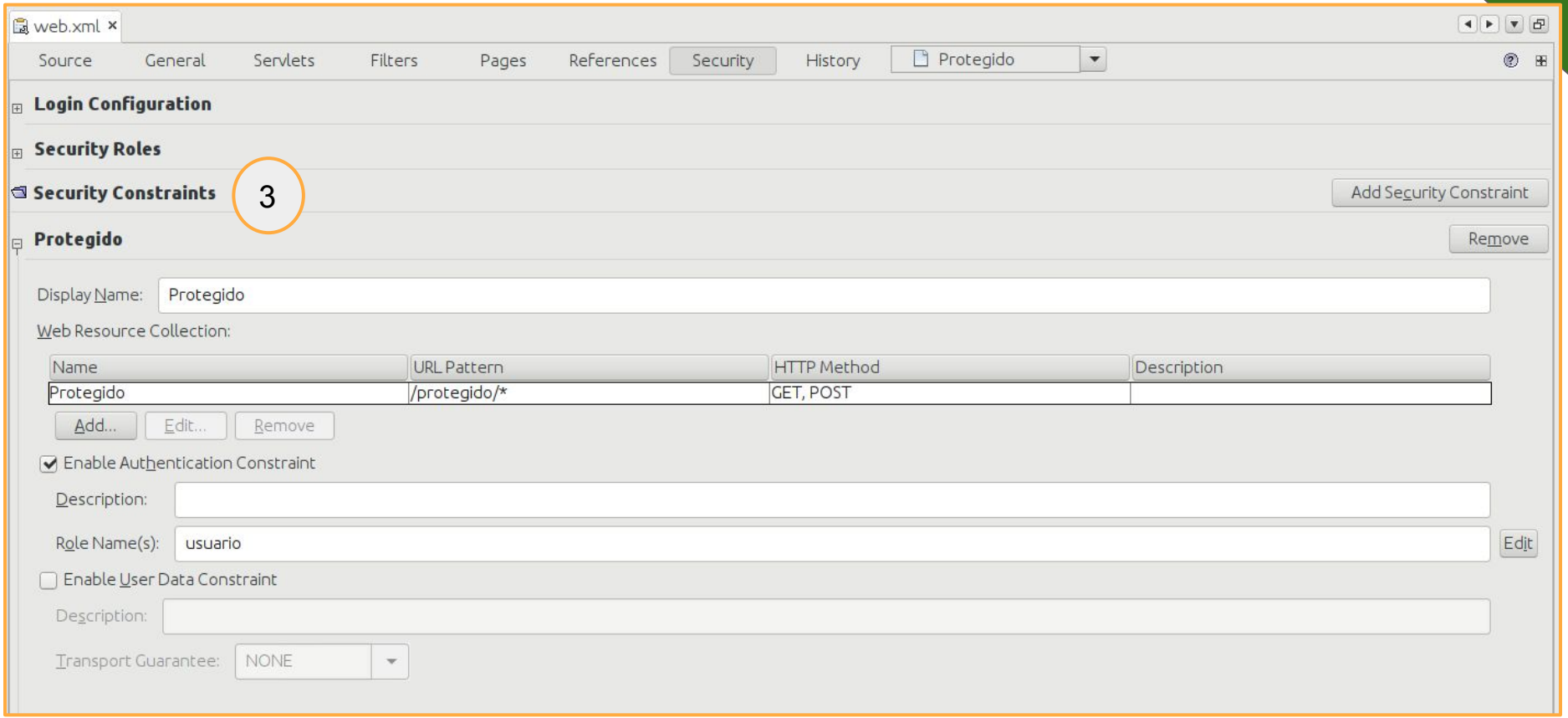
Essa tag pode ser repetida para cada caso de uso que houver no sistema.

Aqui cadastramos os atores

Aqui definimos como será feita a autenticação


web.xml
Source
General
Servlets
Filters
Pages
References
Security
History
Protegido
Login Configuration
1
None
Digest
Client Certificate
Basic
Form
Form Login Page:
Browse...
Form Error Page:
Browse...
Realm Name:
budaya
web.xml
Source
General
Servlets
Filters
Pages
References
Security
History
Protegido
Login Configuration
Security Roles
2
Role Name
Description
usuario
Add...
Edit...
Remove
Security Constraints
Add Security Constraint
Protegido
Remove

DR3


web.xml
Source
General
Servlets
Filters
Pages
References
Security
History
Protegido
Login Configuration
Security Roles
Security Constraints
3
Add Security Constraint
Protegido
Remove
Display Name:
Protegido
Web Resource Collection:
Name
URL Pattern
HTTP Method
Description
Protegido
/protegido/*
GET, POST
Add...
Edit...
Remove
Enable Authentication Constraint
Description:
Role Name(s):
usuario
Edit
Enable User Data Constraint
Description:
Transport Guarantee:
NONE

DR3

Autenticação & Autorização

- BASIC.

```
<login-config>
    <auth-method>BASIC</auth-method>
</login-config>
```

- FORM.

```
<login-config>
    <auth-method>FORM</auth-method>
    <form-login-config>
        <form-login-page>login.html</form-login-page>
        <form-error-page>login_erro.html</form-error-page>
    </form-login-config>
</login-config>
```

DR3

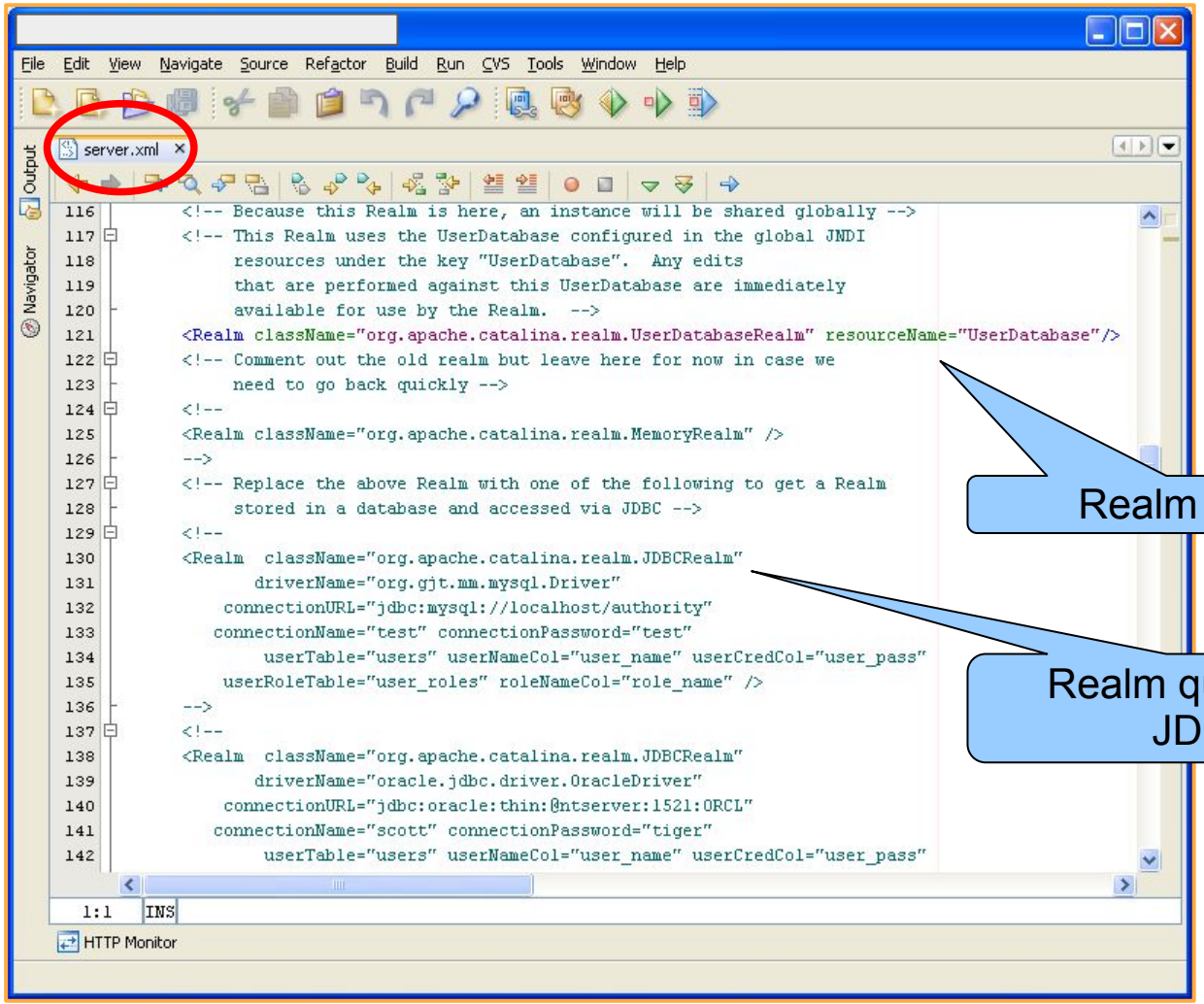
Autenticação & Autorização

- Regras a serem obedecidas quando a autenticação for implementada através de formulário:
  - Campos (input) com nomes obrigatórios:
    - **j_username**
    - **j_password**
  - FORM com ação definida:
    - **j_security_check**

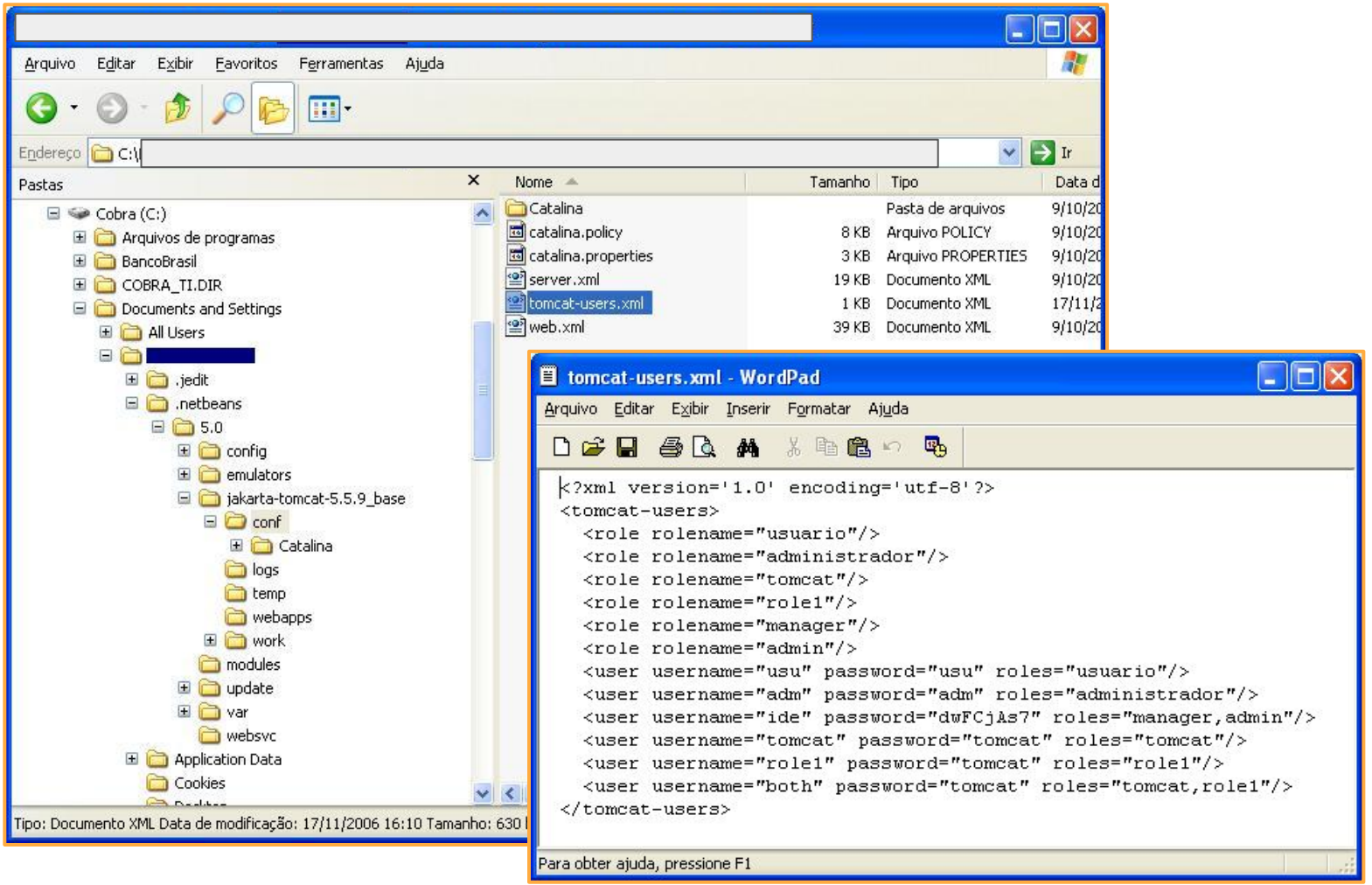
Autenticação & Autorização

- **Security Realm** é o componente de software que protege recursos de uma aplicação, definindo os perfis de quem pode acessá-los.
- Usado para autenticação e autorização.
- Cada servlet container funciona à sua maneira.
- No Tomcat existem diversos tipos, como por exemplo:
  - Classe MemoryRealm ou UserRealm = tomcat-users.xml.
  - Classe JDBCRealm = tabela em banco de dados.

DR3


File Edit View Navigate Source Refactor Build Run CVS Tools Window Help
server.xml
Output
Navigator
116 <!-- Because this Realm is here, an instance will be shared globally -->
117 <!-- This Realm uses the UserDatabase configured in the global JNDI
118 resources under the key "UserDatabase". Any edits
119 that are performed against this UserDatabase are immediately
120 available for use by the Realm. -->
121 <Realm className="org.apache.catalina.realm.UserDatabaseRealm" resourceName="UserDatabase"/>
122 <!-- Comment out the old realm but leave here for now in case we
123 need to go back quickly -->
124 <!--
125 <Realm className="org.apache.catalina.realm.MemoryRealm" />
126 -->
127 <!-- Replace the above Realm with one of the following to get a Realm
128 stored in a database and accessed via JDBC -->
129 <!--
130 <Realm className="org.apache.catalina.realm.JDBCRealm"
131 driverName="org.gjt.mm.mysql.Driver"
132 connectionURL="jdbc:mysql://localhost/authority"
133 connectionName="test" connectionPassword="test"
134 userTable="users" userNameCol="user_name" userCredCol="user_pass"
135 userRoleTable="user_roles" roleNameCol="role_name" />
136 -->
137 <!--
138 <Realm className="org.apache.catalina.realm.JDBCRealm"
139 driverName="oracle.jdbc.driver.OracleDriver"
140 connectionURL="jdbc:oracle:thin:@ntserver:1521:ORCL"
141 connectionName="scott" connectionPassword="tiger"
142 userTable="users" userNameCol="user_name" userCredCol="user_pass"
1:1 INS
HTTP Monitor
Realm default
Realm que utiliza JDBC

DR3


Arquivo Editar Exibir Favoritos Ferramentas Ajuda
Endereço C:\
Ir
Pastas
Cobra (C:)
Arquivos de programas
BancoBrasil
COBRA_TI.DIR
Documents and Settings
All Users
[redacted]
.jedit
.netbeans
5.0
config
emulators
jakarta-tomcat-5.5.9_base
conf
Catalina
logs
temp
webapps
work
modules
update
var
websvc
Application Data
Cookies
Nome
Tamanho
Tipo
Catalina
Pasta de arquivos
catalina.policy
8 KB
Arquivo POLICY
catalina.properties
3 KB
Arquivo PROPERTIES
server.xml
19 KB
Documento XML
tomcat-users.xml
1 KB
Documento XML
web.xml
39 KB
Documento XML
Tipo: Documento XML Data de modificação: 17/11/2006 16:10 Tamanho: 630
tomcat-users.xml - WordPad
Arquivo Editar Exibir Inserir Formatar Ajuda
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
<role rolename="usuario"/>
<role rolename="administrador"/>
<role rolename="tomcat"/>
<role rolename="role1"/>
<role rolename="manager"/>
<role rolename="admin"/>
<user username="usu" password="usu" roles="usuario"/>
<user username="adm" password="adm" roles="administrador"/>
<user username="ide" password="dwFCjAs7" roles="manager,admin"/>
<user username="tomcat" password="tomcat" roles="tomcat"/>
<user username="role1" password="tomcat" roles="role1"/>
<user username="both" password="tomcat" roles="tomcat,role1"/>
</tomcat-users>
Para obter ajuda, pressione F1

Exercício

- Criar um projeto que contenha uma pasta “protegido” onde deve-se colocar um “index.jsp”.
- No “index.jsp” principal, criar um link para o recurso “/protegido/index.jsp”.
- Criar a configuração de autenticação e autorização que solicite o login para acessar o recurso “/protegido/index.jsp”.
  - Implementar a autenticação BASIC.
  - Depois alterar para FORM.