Introdução ao Desenvolvimento Web

SPRING SECURITY

PROF. PABLO VARGAS

Tópicos Abordados

- Spring Security
- Instalação do Spring Security.
- Configuração do Spring Security.

Spring Security

- É um dos projetos do Spring Framework que visa facilitar o desenvolvimento da parte de segurança.
- Existe desde 2003 e utilizado por diversas organizações mundiais.
- Possui uma estrutura de autenticação e controle de acesso poderosa e altamente personalizável.
  - No projeto, estaremos restringindo o acesso através das pastas e o tipo de usuário.
- Proteção contra ataques como fixação de sessão, clickjacking, falsificação de solicitação entre sites, etc.
- Integração da API de servlet.

Spring Security

- https://www.youtube.com/watch?v=ptcjeehUbz8&t=118s

Instalação do Spring Security

- Crie um novo projeto do tipo aplicação web com o nome Spring Security.
  - Selecione o framework JSF e adicione o mesmo código do web.xml do projeto anterior (MVC).
  - Crie os arquivos XHTML da pasta Web Pages igual do projeto anterior (MVC).
  - Crie as classes Usuario, UsuarioDAO, UsuarioDAOHibernate e UsuarioRN igual do projeto anterior (MVC).
  - Crie as classes HibernateUtil, DAOException, DAOFactory, RNException, UtilException igual do projeto anterior (MVC).
  - Crie as classes UsuarioBean e ConexãoHibernateFilter igual do projeto anterior (MVC).
- Extrai os arquivos lib.zip e adicione as biblioteca ao projeto.

Instalação do Spring Security

- Crie o arquivo XML de configuração do hibernate.
  - Obs: a propriedade url pode ter que alterar o value="jdbc:mysql://localhost/idw?useSSL=false"

```
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
        <property name="dialect">org.hibernate.dialect.MySQL5InnoDBDialect</property>
        <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="connection.url">jdbc:mysql://localhost/idw</property>
        <property name="connection.username">root</property>
        <property name="connection.password">root</property>
        <property name="current_session_context_class">thread</property>
        <property name="hibernate.hbm2ddl.auto">update</property>
        <mapping class="usuario.Usuario"/>
    </session-factory>
</hibernate-configuration>
```

Configuração do Spring Security

- Adicione, no arquivo web.xml, os trechos abaixo para configurar o spring security.

```
<!-- Spring Security -->
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
        /WEB-INF/applicationContext.xml
        /WEB-INF/applicationContext-security.xml
    </param-value>
</context-param>
```

Configuração do Spring Security

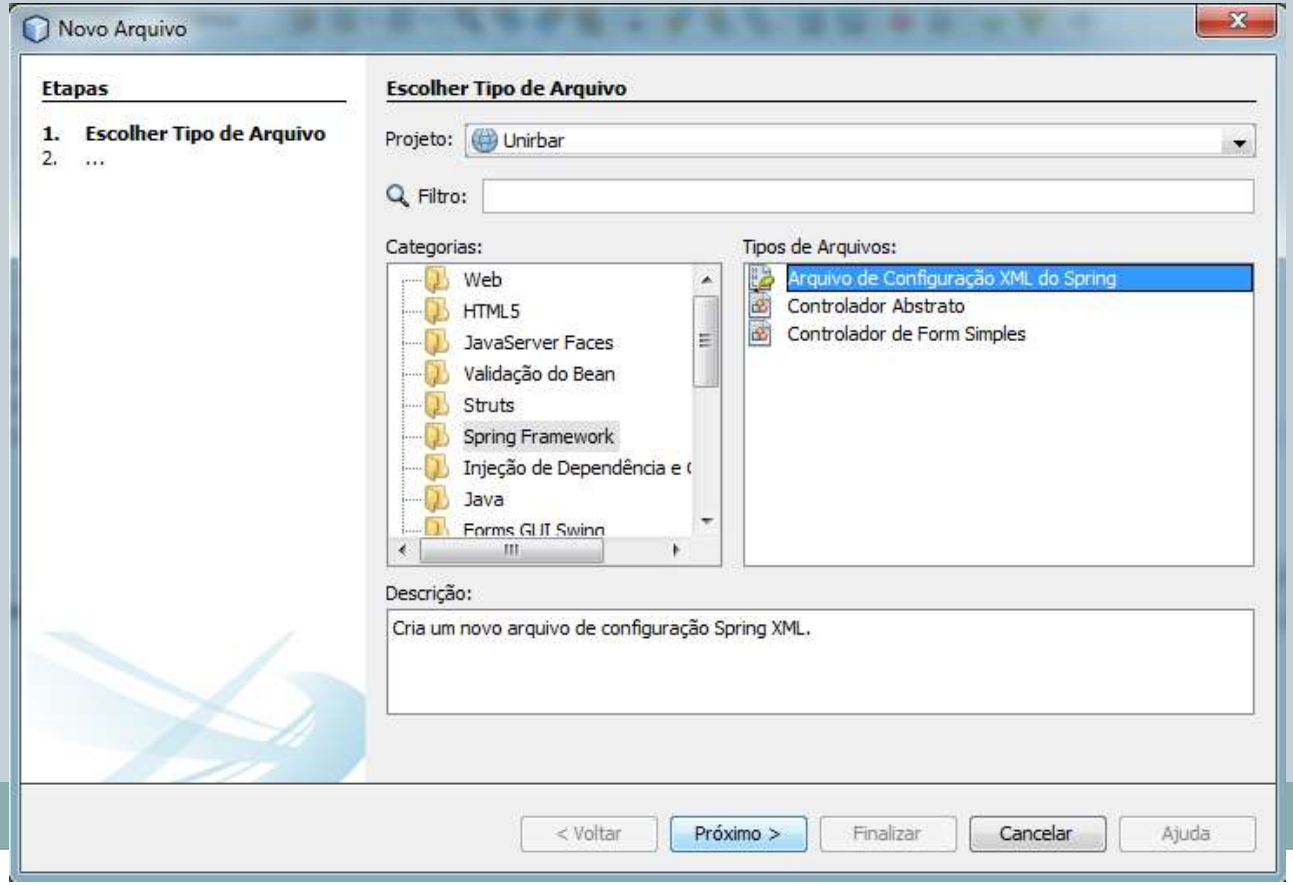
```
<filter>
    <filter-name>springSecurityFilterChain</filter-name>
    <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>

<filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

Configuração do Spring Security

- Crie um arquivo de configuração do Spring com o nome “applicationContext” na pasta WEB-INF.



Configuração do Spring Security

- Obs: a propriedade url pode ter que alterar o value="jdbc:mysql://localhost/idw?useSSL=false"

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

    <bean id="idwDataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource" >
        <property name="url" value="jdbc:mysql://localhost/idw" />
        <property name="driverClassName" value="com.mysql.jdbc.Driver" />
        <property name="username" value="root" />
        <property name="password" value="root" />
   </bean>

</beans>
```

Configuração do Spring Security

- Faça o mesmo procedimento para o arquivo “applicationContext-security”, mas com o seguinte código.

```
<?xml version="1.0" encoding="UTF-8"?>
<b:beans xmlns="http://www.springframework.org/schema/security"
         xmlns:b="http://www.springframework.org/schema/beans"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
                             http://www.springframework.org/schema/security
http://www.springframework.org/schema/security/spring-security-3.0.xsd">
    <http>
        <intercept-url pattern="/Admin/**" access="ROLE_ADMINISTRADOR" />
        <intercept-url pattern="/Logado/**" access="ROLE_USUARIO" />
        <form-login login-page="/Publico/login.jsf"
                    always-use-default-target="true" default-target-url="/Logado/principal.jsf"
                    authentication-failure-url="/Publico/login.jsf?login_error=1"  />
        <logout/>
        <remember-me />
    </http>
```

Configuração do Spring Security

```
    <authentication-manager>
        <authentication-provider>
            <password-encoder hash="md5"/>
            <jdbc-user-service data-source-ref="idwDataSource"
                               authorities-by-username-query="SELECT u.login, p.permissao FROM usuario u, usuario_permissao p
                                                              WHERE u.codigo = p.usuario
                                                                AND u.login = ?"
                               users-by-username-query="SELECT login, senha, ativo FROM usuario WHERE login = ?"
            />
        </authentication-provider>
    </authentication-manager>
</b:beans>
```

Configuração do Spring Security

- Adicione o trecho abaixo dentro da tag “body” da página login.xhtml.

```
<h:panelGroup rendered="#{!empty param.login_error}">
    <span style="font-color:red"> Erro ao efetuar o login.</span><br />
    <br/>
    Motivo: #{SPRING_SECURITY_LAST_EXCEPTION.message}
</h:panelGroup>
```

Configuração do Spring Security

```
<form id="login" method="post" action="${request.contextPath}/j_spring_security_check">
    <table>
        <tr><td>Login</td>
            <td><input type='text' name='j_username'/></td></tr>
        <tr><td>Senha</td>
            <td><input type='password' name='j_password'/></td></tr>
        <tr><td></td>
            <td><input type="submit" value="Entrar"/></td>
        </tr>
    </table>
    <script>
        document.getElementById("login").j_username.value = "#{SPRING_SECURITY_LAST_USERNAME}";
    </script>
</form>
```

Configuração do Spring Security

- Crie a pagina principal.xhtml na pasta Logado.
- Adicione o trecho abaixo dentro da tag “body” na página principal.xhtml da pasta Logado.
  - Modifique o cabeçalho e as tags head e body para JSF.

```
<h:body>
    <h1>Sistemas Unirbar</h1>

    <br/>
    <h:form>
        <sec:ifAnyGranted roles="ROLE_ADMINISTRADOR">
            <h:commandLink action="/Admin/principal" title="Administrativo">
                <h:graphicImage library="imagens" name="administrativo16.png" />
            </h:commandLink>
        </sec:ifAnyGranted>
    </h:form>
</h:body>
```

Configuração do Spring Security

- No html adicione a tag “sec”.

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:sec="http://www.springframework.org/security/facelets/tags">
```

Configuração do Spring Security

- Adicione o método abaixo no UsuarioBean.java

```
public String atribuiPermissao(Usuario usuario, String permissao) {

    this.usuario = usuario;

    java.util.Set<String> permissoes = this.usuario.getPermissao();

    if (permissoes.contains(permissao)) {
        permissoes.remove(permissao);
    } else {
        permissoes.add(permissao);
    }
    UsuarioRN usuarioRN = new UsuarioRN();
    usuarioRN.salvar(this.usuario);
    return null;
}
```

Configuração do Spring Security

- Adicione a coluna na página principal.xhtml da pasta Admin.

```
<h:column>
    <f:facet name="header">Permissões</f:facet>
    <h:commandLink action="#{usuarioBean.atribuiPermissao(usuario, 'ROLE_ADMINISTRADOR')}"
                   title="Permissão Administrador">
        <h:graphicImage library="imagens"
                        name="ROLE_ADMINISTRADOR_#{usuario.permissao.contains('ROLE_ADMINISTRADOR')}.png"
                        style="border:0"/>
    </h:commandLink>
    <h:commandLink action="#{usuarioBean.atribuiPermissao(usuario, 'ROLE_USUARIO_VIP')}"
                   title="Permissão Usuário VIP">
        <h:graphicImage library="imagens"
                        name="ROLE_USUARIO_VIP_#{usuario.permissao.contains('ROLE_USUARIO_VIP')}.png"
                        style="border:0"/>
    </h:commandLink>
</h:column>
```

Configuração do Spring Security

- Adicione o trecho abaixo na classe UsuarioRN.

```
public void salvar(Usuario usuario) {

        Integer codigo = usuario.getCodigo();
        if (codigo == null || codigo == 0) {
            usuario.getPermissao().add("ROLE_USUARIO");
                this.usuarioDAO.salvar(usuario);
        } else {
                this.usuarioDAO.atualizar(usuario);
        }
}
```

Configuração do Spring Security

- Crie a classe ContextoBean.java dentro do pacote "web" com o seguinte código.

```
@Named
@SessionScoped
public class ContextoBean implements Serializable {

	private Usuario	usuarioLogado		= null;

	public Usuario getUsuarioLogado() {
		FacesContext context = FacesContext.getCurrentInstance();
		ExternalContext external = context.getExternalContext();
		String login = external.getRemoteUser();

		if (this.usuarioLogado == null || !login.equals(this.usuarioLogado.getLogin())) {

			if (login != null) {
				UsuarioRN usuarioRN = new UsuarioRN();
				this.usuarioLogado = usuarioRN.buscarPorLogin(login);
			}
		}
		return usuarioLogado;
	}

}
```

Configuração do Spring Security

- Na classe UsuarioBean.class, crie uma propriedade, do tipo String, chamada de “senhaMD5”.
- Gere os métodos get e set da propriedade.
- Na página usuario.xhtml:
  - Altere nos campos de “senha” e “confirma senha” o atributo required=“#{empty contextoBean.usuarioLogado}” e redisplay=“false”.

Configuração do Spring Security

Na classe UsuarioBean.class altere o método editar.

```
public String editar(){
    this.senhaMD5 = this.usuario.getSenha();
    return "/Publico/usuario";
}
```

Configuração do Spring Security

Na classe UsuarioDAOHibernate.class altere o método atualizar.

```
public void atualizar(Usuario usuario) {
        this.session.merge(usuario);
}
```

Configuração do Spring Security

Na classe UsuarioBean.class altere o método salvar.

```
public String salvar() {
    FacesContext context = FacesContext.getCurrentInstance();
    String senha = this.usuario.getSenha();
    if (senha != null && senha.trim().length() > 0  && !senha.equals(this.confirmarSenha)) {
        FacesMessage facesMessage = new FacesMessage("A senha não foi confirmada corretamente");
        context.addMessage(null, facesMessage);
        return null;
    }
    if (senha != null && senha.trim().length() == 0) {
            this.usuario.setSenha(this.senhaMD5);
    } else {
            String senhaCripto = DigestUtils.md5DigestAsHex(senha.getBytes());
            this.usuario.setSenha(senhaCripto);
    }
    UsuarioRN usuarioRN = new UsuarioRN();
    usuarioRN.salvar(this.usuario);

    return this.destinoSalvar;
}
```

Configuração do Spring Security

- As configurações com o banco de dados do Spring Security foram passadas para o arquivo applicationContext.xml.
- No arquivo web.xml:
  - O parametro “contextConfigLocation” informa os arquivos de configuração.
  - As tags “filter” e “filter-mapping” permitirão que o spring security intercepte todas as requisições.
  - A tag “listener” indica que o spring security carrega os arquivos de configuração simultaneamente com o servidor.

JAVA EE x Spring

- https://www.youtube.com/watch?v=fWcg8at3VCM

Configuração do Spring Security

- Exercício: Crie uma forma do usuário obter permissão de admin.