DESPLIEGUE DEL SOFTWARE SODISPOL

1) Descargar la versión de Tomcat 8.0.
2) Asegurarse de que el puerto configurado para Tomcat sea el **8090**.
3) Abrir el archivo de configuración de Tomcat **"server.xml"** y colocar la siguiente línea.

   ```
   <Connector SSLEnabled="true" acceptCount="100" clientAuth="false"
   disableUploadTimeout="true" enableLookups="false"
   keystoreFile="C:/Users/usuario/.keystore" keystorePass="changeit"
   maxHttpHeaderSize="8192" maxSpareThreads="75" maxThreads="150"
   minSpareThreads="25" port="8443"
   protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https"
   secure="true" sslProtocol="TLS"/>
   ```

   En donde la ruta resaltada anteriormente corresponde al usuario administrador que se encuentre usando la máquina.
4) Copie el archivo **.keystore** en la ruta del usuario especificada en el paso anterior.
5) Copie el archivo **cacerts** en la ruta **JAVA_HOME/jre** y reemplace el archivo en caso de que ya exista uno con el mismo nombre.

   Nota: Estos archivos contienen el certificado y el almacén de claves necesarios para poder integrar el CAS de Espol con nuestro proyecto, puesto que CAS utiliza el protocolo de seguridad https.

6) Abra el navegador Chrome y escriba la siguiente ruta **"localhost:8090".**
7) Seleccione la opción Manager App y si no se ha modificado el archivo **"users.xml"** de la carpeta en donde se encuentra tomcat, el usuario y password serán **admin.**
8) A continuación despliegue el archivo **SodispolSoftware.war** y ejecútelo**.**

Configurar Servidor de CAS con SpringSecurity

1) En el archivo "spring_security_CAS.xml" se agrega la siguiente configuración, especificando el rol de autorización para los usuarios, y la dirección de logout del CAS que se desee integrar.

```
<sec:http entry-point-ref="casEntryPoint" auto-config="true" path-type="ant">
    <sec:custom-filter before="CAS_FILTER" ref="casSingleSignOutFilter"/>
    <sec:custom-filter after="CAS_FILTER"  ref="casFilter"/>
    <sec:intercept-url pattern="/*" access="ROLE_DOCTOR"/>
    <sec:intercept-url pattern="/administradores/*" access="ROLE_ADMIN"/>
    <sec:intercept-url pattern="/usuarios/*" access="ROLE_DOCTOR"/>
    <sec:intercept-url pattern="/home.xhtml" access="ROLE_DOCTOR"/>
    <!--<sec:logout logout-success-url="https://compa:8443/cas/logout" invalida
    <sec:logout logout-success-url="https://auth.espol.edu.ec/logout"
                invalidate-session="true" logout-url="/logout" />
</sec:http>
```

2) A continuación se configura el dominio y la ruta principal de nuestra aplicación.

```
<bean id="serviceProperties" class="org.springframework.security.cas.ServiceProperties">
    <property name="service" value="http://localhost:8090/SodispolSoftware/j_spring_cas_security_check"/>
    <property name="sendRenew" value="false"/>
</bean>
```

3) Se especifica la ruta del servidor de CAS.

```
<bean id="casEntryPoint"
      class="org.springframework.security.cas.web.CasAuthenticationEntryPoint">
    <!--<property name="loginUrl" value="https://compa:8443/cas/login"/>-->
    <property name="loginUrl" value="https://auth.espol.edu.ec/login"/>
    <property name="serviceProperties" ref="serviceProperties"/>
</bean>
```

4) Se añade ciertas etiquetas para poder integrar spring con CAS, así como la clase de validación de tickets que se usará.

```
<bean id="passwordEncoder"
      class="org.springframework.security.authentication.encoding.ShaPasswordEncoder"/>
<bean id="casSingleSignOutFilter"
      class="org.jasig.cas.client.session.SingleSignOutFilter" />
<bean id="casFilter"
      class="org.springframework.security.cas.web.CasAuthenticationFilter">
    <property name="authenticationManager" ref="authenticationManager"/>
</bean>

<sec:authentication-manager alias="authenticationManager">
    <sec:authentication-provider ref="casAuthenticationProvider" />
</sec:authentication-manager>

<!--<bean id="allAuthenticatedUserService" class="com.spring.myuserdetail.MyUserService"/

<bean id="casAuthenticationProvider"
      class="org.springframework.security.cas.authentication.CasAuthenticationProvider">
    <property name="userDetailsService" ref="userServices"/>
    <!--<property name="userDetailsService" ref="allAuthenticatedUserService"/>-->
    <property name="serviceProperties" ref="serviceProperties"/>
    <property name="ticketValidator">
        <bean class="org.jasig.cas.client.validation.Cas20ServiceTicketValidator">
            <!--<constructor-arg index="0" value="https://compa:8443/cas"/>-->
            <constructor-arg index="0" value="https://auth.espol.edu.ec"/>
        </bean>
    </property>
    <property name="key" value="my_password_for_this_auth_provider_only"/>
</bean>
```

5) Se añade la fuente de usuarios que usarán nuestra aplicación, en nuestro caso los usuarios se encuentran en el archivo **“users.properties”**.

```
<bean id="wrappingFilter" class="org.jasig.cas.client.util.HttpServletRequestWrapperFilter" />
<sec:user-service properties="/WEB-INF/users.properties" id="userServices"/>
```

6) A continuación se muestra la manera en la que se deben añadir los usuarios.

```
rmaya=,ROLE_DOCTOR
ricardo=,ROLE_ADMIN
```