

Solución Desafío - Sitio protegido con credenciales

Crear clase Role en carpeta model

```
package cl.desafiolatam.challengesecurity2.model;

public enum Role {

    ADMIN, CLIENT

}
```

Crear clase Users en carpeta model

```
package cl.desafiolatam.challengesecurity2.model;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

import java.io.Serializable;

@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
public class Users implements Serializable {

    private String email;
    private String password;
    private Role role;

}
```

Crear interfaz UsersMapper en carpeta mapper

```
package cl.desafiolatam.challengesecurity2.mapper;
import cl.desafiolatam.challengesecurity2.model.Users;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Param;
import org.apache.ibatis.annotations.Select;
@Mapper
public interface UserMapper {

    @Select("select * from users where email = #{email}")
    Users findByEmail(@Param("email") String email);

}
```

Crear clase AuthService en carpeta service

```
package cl.desafiolatam.challengesecurity2.service;

import cl.desafiolatam.challengesecurity2.mapper.UserMapper;
import cl.desafiolatam.challengesecurity2.model.Users;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.GrantedAuthority;
import
org.springframework.security.core.authority.SimpleGrantedAuthority;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetails;
import
org.springframework.security.core.userdetails.UserDetailsService;
import
org.springframework.security.core.userdetails.UsernameNotFoundException;
import
org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Service;

import java.util.ArrayList;
import java.util.List;
```

```
@Service
public class AuthServiceImpl implements UserDetailsService {

    @Autowired
    private UserMapper userMapper;

    @Autowired
    private BCryptPasswordEncoder encoder;

    @Override
    public UserDetails loadUserByUsername(String email) throws
UsernameNotFoundException {
        Users users = userMapper.findByEmail(email);
        if (users == null) throw new
UsernameNotFoundException(email);
        List<GrantedAuthority> authorities = new ArrayList<>();
        authorities.add(new
SimpleGrantedAuthority(users.getRole().toString()));
        return new User(users.getEmail(),
            users.getPassword(),
            authorities);
    }
}
```

Crear clase AdminController en carpeta controller

```
package cl.desafiolatam.challengesecurity2.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class AdminController {

    @GetMapping("/admin")
    public ModelAndView home() {
        ModelAndView modelAndView = new ModelAndView("admin");
        return modelAndView;
    }

}
```

Crear clase ClientController en carpeta controller

```
package cl.desafiolatam.challengesecurity2.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class ClientController {

    @GetMapping("/client")
    public ModelAndView home() {
        ModelAndView modelAndView = new ModelAndView("client");
        return modelAndView;
    }

}
```

Crear template admin.html en thymeleaf dentro de resources/templates

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">

  <head>
    <title>Inicio Administrador</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.m
in.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x9Jv
oRxT2MZw1T" crossorigin="anonymous">
  </head>

  <body>
    <ul class="nav justify-content-end">
      <li class="nav-item">
        <a class="nav-link active" href="#">Administrador</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" th:href="@{/logout}">Cerrar sesion</a>
      </li>
    </ul>
    <p>Bienvenido!</p>
  </body>
</html>
```

Crear template client.html en thymeleaf dentro de resources/templates

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">

  <head>
    <title>Inicio Cliente</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.m
in.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x9Jv
oRxT2MZw1T" crossorigin="anonymous">
  </head>

  <body>
    <ul class="nav justify-content-end">
      <li class="nav-item">
        <a class="nav-link active" href="#">Cliente</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" th:href="@{/logout}">Cerrar sesion</a>
      </li>
    </ul>
    <p>Bienvenido!</p>
  </body>
</html>
```

Crear clase WebSecurityConfig en carpeta configuration

```
package cl.desafiolatam.challengesecurity2.configuration;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.A
uthenticationManagerBuilder;
import
org.springframework.security.config.annotation.web.builders.HttpSecurity
;
import
org.springframework.security.config.annotation.web.configuration.EnableW
ebSecurity;
import
org.springframework.security.config.annotation.web.configuration.WebSecu
rityConfigurerAdapter;
import
org.springframework.security.core.userdetails.UserDetailsService;
import
org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

@EnableWebSecurity
@Configuration
public class WebSecurityConfig extends WebSecurityConfigurerAdapter
{

    @Autowired
    private UserDetailsService userDetailsService;

    @Override
    public void configure(AuthenticationManagerBuilder auth) throws
Exception {

auth.userDetailsService(userDetailsService).passwordEncoder(passwordEnco
der());
    }

    @Override
    public void configure(HttpSecurity http) throws Exception {
```

```
        http.csrf().disable()
            .authorizeRequests()
            .antMatchers("/admin/**").hasAuthority("ADMIN")
            .antMatchers("/client/**").hasAuthority("CLIENT")
            .antMatchers("/login").permitAll()
            .anyRequest().authenticated()
            .and()
            .formLogin()
            .loginPage("/login")
            .failureUrl("/login?error=true")
            .usernameParameter("email")
            .passwordParameter("password")
            .defaultSuccessUrl("/client")
            .and()
            .exceptionHandling()
            .accessDeniedPage("/recurso-prohibido");
    }

    @Bean
    public BCryptPasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }
}
```


Crear clase LoginController en carpeta controller

```
package cl.desafiolatam.challengesecurity2.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class LoginController {

    @GetMapping("/login")
    public ModelAndView login(){
        return new ModelAndView("login");
    }

}
```

Crear template login.html en carpeta thymeleaf

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">

    <head>
        <title>Iniciar Sesión</title>
        <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.m
in.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x9Jv
oRxT2MZw1T" crossorigin="anonymous">
    </head>
    <body>

        <div class="container">
            <div class="row">
                <div class="col-sm-4">
```

```
        </div>
        <div class="col-sm-4">
            <form th:action="@{/login}" method="post"
class="form-signin">
                <h3 class="form-signin-heading"
th:text="Credenciales"></h3>
                <br/>
                <input type="text" id="email" name="email"
th:placeholder="Email"
                    class="form-control"/> <br/>
                <input type="password" th:placeholder="Password"
                    id="password" name="password"
class="form-control"/> <br/>

                <div align="center" th:if="${param.error}">
                    <p style="font-size: 20; color:
#FF1C19;">Verifica las credenciales.</p>
                </div>
                <button class="btn btn-lg btn-primary btn-block"
name="Submit" value="Login" type="Submit"
                    th:text="Login">
                </button>
            </form>

        </div>
        <div class="col-sm-4">
        </div>
    </div>
</div>
</body>
</html>
```

Crear controlador y template 403.html de error personalizado para recurso prohibido

Controlador

```
package cl.desafiolatam.challengesecurity2.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class ErrorController {

    @GetMapping("/recurso-prohibido")
    public ModelAndView recurso03() {
        return new ModelAndView("error/403");
    }

}
```

html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:th="http://www.thymeleaf.org">

  <head>
    <title>Recurso protegido</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.m
in.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x9Jv
oRXT2MZw1T" crossorigin="anonymous">
  </head>

  <body>
    <ul class="nav justify-content-end">
      <li class="nav-item">
        <a class="nav-link" th:href="@{/logout}">Cerrar sesion</a>
      </li>
    </ul>
    <h3>Recurso prohibido</h3>
  </body>
</html>
```