

## Pasos sugeridos a seguir para iniciar con Proyectos Web:

1. Instalación:
  - a. CONFIGURAR VARIABLES→ Maven (Ver instructivo “Instalación Maven”) e instalar plugin Spring para seguir utilizando NetBeans.
  - b. SPRING→ **Instalar el plugin de Spring para NetBeans** o Instalar directamente Spring Tool Suite (quien tenga MAC debe utilizar esa opción preferentemente)
2. Ver Video 1. “Fundamentos Web”  
<https://www.youtube.com/watch?v=zWSUoUG3XEE&list=PLgwlfcqa5h3x8HAea7s3DXv5CvjEfR4uG>
3. Ver Video 2. “Configuración Spring”. Solo verlo para comprender como funciona lo que hacemos en Punto 4 de esta guía.  
<https://www.youtube.com/watch?v=-eXHYtDntnU&list=PLgwlfcqa5h3x8HAea7s3DXv5CvjEfR4uG&index=2>
4. Crear un proyecto Spring (Ver instructivo “Paso 2”). Aquí también se indica como instalar el plugin necesario.
5. Crear la base de datos que voy a utilizar para mi proyecto.
6. Edito mi archivo pom.xml (Modificar la versión a 2.5.4). Guardar cambios
7. Editar archivo application.properties (Ver instructivo “Paso 3”).
8. Si quiero acá... para verificar que funciona y escucha el puerto de forma correcta, creo un controlador para ver una página modelo:
  - a. Crear una carpeta para destinarla a los controladores. (Leer en la teoría que significa y el uso de los controladores, para comprender como funciona). Recordar que toda carpeta debe nacer de la carpeta raíz de nuestro proyecto para evitar futuras complicaciones.
  - b. Crear una clase llamada PortalControlador
  - c. Incorporar la anotación, sobre el nombre de la clase @Controller
  - d. Copiar este método:

```
@GetMapping("/")  
  
public String index(){  
  
    return "index.html";  
  
}
```

(Esto me permite, mapear que cuando entre a mi página principal lo dirige a la vista index.html)

- e. Crear un archivo del tipo html, en la carpeta “templates” (OtherSources / src/main/resources) un archivo llamado index con alguna estructura html. EJ:

```
<html>  
    <head>
```

```

<title>TODO supply a title</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
</head>
<body>
<p> LOGRADO CON EXITO </p>
</body>
</html>

```

9. Guardar cambios y correr mi aplicación. SPRING BOOT RUN (Esto aparece en la ventana Navigator, si no me aparece en NetBeans → Pestaña Windows de NetBeans → Navigator)
10. Abrir en mi navegador **localhost:8080** (mi servidor local)
11. AVISO: Hasta no configurar la seguridad de mi proyecto, al ingresar al navegador, me pedirá un usuario y contraseña:
  - a. Opción 1: Utilizo las credenciales provistas por Spring
    - i. Usuario: user
    - ii. Token: Al correr el programa, me incorpora una cadena de caracteres. Copiar y pegar (Token que veo en mi Output )
  - b. Opción 2: Configuro la seguridad de mi proyecto. Ver apartado 13 / Inciso d de este instructivo.
12. Continuar viendo los videos. Se sugiere ver Primera lista de reproducción completa.  
<https://www.youtube.com/playlist?list=PLgwlfqga5h3x8HAea7s3DXv5CvjEfr4uG>
13. Maqueto mi proyecto, todas las carpetas contenedoras deben nacer de mi carpeta base, es decir la que contiene el main, BOTON DERECHO, CREAR PAQUETE, e ir nombrando cada uno (entidades / controladores / etc):
  - a. Creo mis entidades con sus anotaciones correspondientes. Recordar que deben estar todas las entidades en un mismo paquete que las contenga.
  - b. Creo mis repositorios con sus anotaciones correspondientes. Recordar que deben estar todos los repositorios en un mismo paquete que los contenga.
  - c. Corro el proyecto, para asegurarme que no tengo errores y se creen las tablas de forma correcta.
  - d. Crear una clase para configurar la Seguridad de mi proyecto. Esta ya les queda completa para futuros métodos “de login”
    - i. Ubicación: SourcePackages
    - ii. Paquete Nombre: configuraciones

iii. Nombre Clase: ConfiguracionesSeguridad

(Pegar el siguiente código en la clase ConfiguracionesSeguridad – Importar lo necesario)

```
@Configuration
@EnableWebSecurity
@EnableGlobalMethodSecurity(prePostEnabled=true)
@Order(1)
public class ConfiguracionesSeguridad extends WebSecurityConfigurerAdapter{

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .authorizeRequests()
                .antMatchers("/css/**", "/js/**", "/img/**", "/*").permitAll()

            .and().formLogin()
                .loginPage("/login") // Que formulario esta mi login
                .loginProcessingUrl("/logincheck")
                .usernameParameter("username") // Como viajan los
datos del logueo
                .passwordParameter("password")// Como viajan los
datos del logueo

                .defaultSuccessUrl("/inicio") // A que URL viaja
                .permitAll()
            .and().logout() // Aca configuro la salida
                .logoutUrl("/logout")
                .logoutSuccessUrl("/login?logout")
                .permitAll().and().csrf().disable();
    }
}
```

- e. Luego comenzar con servicios y controladores. **ACLARACION:** Es conveniente, método que declaro, probarlo con su circuito correspondiente.

14. Recordar que tienen proyectos modelos, descargar e ir viendo según avanza con los contenidos, existe en este repositorio un README donde aclara que trabaja con cada proyecto:

<https://github.com/agustinfiorde/Spring>