

# Registro y Autenticación por pasos

## Índice

|                                    |   |
|------------------------------------|---|
| 1. Objetivo .....                  | 2 |
| 1.1. Registro por pasos.....       | 2 |
| 1.2. Autenticación por pasos.....  | 2 |
| 2. Requisitos funcionales .....    | 2 |
| 2.1. Registro por pasos.....       | 2 |
| 2.2. Autenticación por pasos.....  | 5 |
| 3. Tests unitarios.....            | 6 |
| 4. Comprobaciones a realizar ..... | 6 |
| 5. Recomendación .....             | 7 |
| 6. Requisitos generales .....      | 7 |

## 1. Objetivo

Esta pequeña aplicación consta de dos partes diferenciadas, pero relacionadas.

- Un registro por pasos
- Una autenticación o login

La ventana inicial de la aplicación será la de login/autenticación, desde la que habrá un acceso al proceso de registro, bien mediante un enlace, bien mediante un botón.

### 1.1. Registro por pasos

Se pretende simular el registro en un determinado servicio, en el que se irán recabando los datos del usuario en **tres pasos**, es decir, tres ventanas independientes, cada una con su formulario. Finalmente se mostrará, en una cuarta ventana, un resumen con toda la información recabada en los tres pasos previos.

- En el **primer paso** se introducirán los **datos de usuario**.
- En el **segundo paso** se introducirán los **datos personales**.
- En el **tercer paso** se introducirán los **datos profesionales**.
- En la **ventana resumen** se mostrará **toda la información recabada** en los pasos previos en **formato de texto** (no dentro de campos de formulario).

Al terminan este último paso de resumen, se podrá guardar el usuario en el sistema.

### 1.2. Autenticación por pasos

Consiste en registrarse en el sistema en dos pasos:

- En el primer paso se introduce el usuario.
- En el segundo pase se introduce la clave.

Una vez introducidas unas credenciales correctas, se accede al área personal donde se da la bienvenida al usuario y se informa del número de visitas que lleva hasta ese momento.

## 2. Requisitos funcionales

### 2.1. Registro por pasos

1. **Pasos o ventanas.** Se irán registrando datos del usuario en varios pasos, para finalmente guardarlo todos en una colección de usuarios existente en el sistema.

- a. En el primer paso, **Datos del usuario**, se introducirán:

| Campo         | Tipo de datos | Tipo HTML |
|---------------|---------------|-----------|
| nombre        | Texto         | text      |
| clave         | Texto         | password  |
| confirmaClave | Texto         | password  |

- b. En el segundo paso, **Datos personales**, se introducirán:

| Campo               | Tipo de datos  | Tipo HTML       |
|---------------------|--|-----------------|
| tratamiento_deseado | Colección de tratamientos (Señora, Señor, Señorita, Señorito, Caballero, Caballera, Amazona, Amazono...) | select simple   |
| apellido            | Texto  | text            |
| fechaNacimiento     | Fecha  | text            |
| edad                | Número entero  | text            |
| genero              | Colección de géneros   | radio           |
| casado_pareja       | Cierto/falso   | checkbox        |
| hijos               | Cierto/falso   | checkbox        |
| nacionalidades      | Colección de países  | select multiple |

- c. En el tercer paso, **Datos profesionales**, se introducirán:

| Campo        | Tipo de datos  | Tipo HTML |
|--------------|--|-----------|
| departamento | Colección de departamentos del esquema Scott de Oracle | select    |
| salario      | Número real  | text      |
| comentarios  | Texto  | textarea  |

2. **Validaciones.** En cada paso se llevará a cabo algunas validaciones sencillas, no permitiéndose pasar al siguiente hasta que se cumplan, e informando al usuario del problema para que pueda subsanarlo.

La primera validación que se debe pasar es que cada campo contenga un dato del tipo de datos esperado (ej. La fecha de nacimiento tiene que ser una fecha válida, el salario debe ser un número real positivo...).

- a. En el primer paso, se debe cumplir:
- el usuario no puede existir ya en el sistema
  - la contraseña y su confirmación sean iguales y no sean la cadena vacía.

- b. En el segundo paso se debe cumplir:
      - i. La fecha de nacimiento tiene que ser de hace más de 18 años.
      - ii. Se deben elegir al menos dos nacionalidades.
    - c. En el tercer paso se debe cumplir:
      - i. Se debe rellenar el salario que debe ser un número mayor que el SMI actual en España.
  3. **Navegación entre ventanas.** Existirá una barra de navegación ubicada en la parte superior que, en todo momento, permitirá pasar al paso que se desee. Al acceder a ese paso, si el usuario hubiera guardado información previamente, se mostrará dicha información. Al moverse entre dos pasos mediante la barra de navegación no se guarda ninguna información que se hubiera podido cumplimentar en el formulario del paso origen.
  4. **Registro de datos de un paso.** En cada ventana del proceso de registro existirá un botón **Grabar paso X** (para  $x \in \{1..3\}$ ) que grabará (registrará en la sesión) los datos introducidos por el usuario en ese paso, y avanzará al siguiente paso. En el paso 3, al grabar se pasará al paso resumen donde se muestra toda la información introducida por el usuario.

Una posible apariencia sería:

|  |   |
|--|---|
| <div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 10px;"> <span>1 2 3</span> <span>🏁</span> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <b>Datos personales</b> </div> <div style="margin-bottom: 10px;">       Nombre <input type="text" value="Ramón"/> Apellidos <input type="text" value="Donguillo"/> </div> <div style="margin-bottom: 10px;">       Fecha de nacimiento <input type="text" value="19/04/1950"/> </div> <div style="margin-bottom: 10px;">       Género -&gt; Mujer <input type="radio"/> Hombre <input checked="" type="radio"/> Otro <input type="radio"/> </div> <div style="margin-bottom: 10px;">       Casado o Pareja de hecho <input checked="" type="checkbox"/> Hijos <input checked="" type="checkbox"/> </div> <div style="margin-bottom: 10px;">       Nacionalidades <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">         Española<br/>         Francesa<br/>         Italiana<br/>         Portuguesa       </div> </div> <div style="text-align: right;"> <input type="button" value="Grabar información e ir al paso 2 - Datos profesionales"/> </div> | <div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 10px;"> <span>1 2 3</span> <span>🏁</span> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <b>Datos profesionales</b> </div> <div style="margin-bottom: 10px;">       Departamento <input type="text" value="Marketing"/> </div> <div style="margin-bottom: 10px;">       Salario <input type="text" value="2750.00"/> </div> <div style="margin-bottom: 10px;"> <div style="border: 1px solid #ccc; padding: 5px; min-height: 40px;">         En la venta no hay rival       </div> </div> <div style="margin-bottom: 10px;">       Comentarios <div style="border: 1px solid #ccc; height: 20px;"></div> </div> <div style="text-align: right;"> <input type="button" value="Grabar información e ir al paso 3 - Datos bancarios"/> </div>                        |
| <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px; text-align: center;"> <input type="button" value="Grabar información e ir al resumen final"/> </div> <div style="margin-bottom: 10px;">       Cuenta corriente <input type="text" value="12345678901234567890"/> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <b>Datos bancarios</b> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 10px;"> <span>1 2 3</span> <span>🏁</span> </div>  | <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <b>Datos personales</b> </div> <div style="margin-bottom: 10px;">       Nombre: <b>Ramón</b><br/>       Apellidos: <b>Donguillo</b><br/>       Fecha de nacimiento: <b>1950-04-19</b><br/>       Nacionalidades: <b>Española Portuguesa</b> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <b>Datos profesionales</b> </div> <div style="margin-bottom: 10px;">       Departamento: <b>Marketing</b><br/>       Salario: <b>2750.00</b><br/>       Comentarios: <b>En la venta no hay rival</b> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <b>Datos bancarios</b> </div> <div style="margin-bottom: 10px;">       Cuenta corriente: <b>12345678901234567890</b> </div> <div style="text-align: right;"> <a href="#">Volver al principio</a> </div> |

5. **Descarte de todos los datos introducidos.** Desde el paso resumen se debe poder descartar (eliminar) todos los datos introducidos, volviendo al primer paso y destruyendo la sesión, lo que provocará que la aplicación se “resetee” y se comporte como si no se hubiera almacenado información alguna.

## 2.2. Autenticación por pasos

1. **Pasos o ventanas.** Se irán introduciendo por pasos los datos necesarios para que el usuario se pueda autenticar:
  - a. En el primer paso, **usuario**, aparecerán los campos:
    - i. nombre o login del usuario, de tipo textual.
    - ii. listado de usuarios previamente autenticados, que será un desplegable.Se podrá rellenar a mano el campo texto, o seleccionando uno de la lista de usuarios previamente registrados, lo que producirá que se rellene el campo usuario.
  - b. En el segundo paso, **clave**, aparecerá únicamente el campo contraseña del usuario.
  - c. La tercera ventana, **Área personal**, ya no contendrá ningún formulario y representará la zona de la aplicación a la que el usuario accede una vez autenticado. Habrá un botón/enlace para volver a la ventana de *login*.
2. **Validaciones.** En cada paso se llevará a cabo alguna validación sencilla, no permitiéndose pasar al siguiente hasta que se cumpla, e informando al usuario del problema para que pueda subsanarlo.
  - a. En el primer paso, no se podrá avanzar hasta que se introduzca un usuario existente en la colección de usuarios registrados.
  - b. En el segundo paso, no se podrá avanzar hasta que se introduzca la contraseña del usuario añadido en el primer paso.
3. **Área personal.** Una vez que el usuario se haya autenticado, accederá a su área personal, donde se le dará la bienvenida (**ej.** Bienvenido Sr. Peláez (usuario: alfa)) y se le informará del número de visitas que lleva. Cada vez que un usuario se autentique correctamente, se guardará en el cliente HTTP (nunca en el servidor) el nombre del usuario y el número de autenticaciones válidas que lleva (si es su primera autenticación, el número almacenado será 1). Esto se hará con todos los usuarios del sistema que se vayan autenticando correctamente.
4. **Control de accesos inadecuados.** Se debe impedir que un “usuario avisado” acceda directamente a una ventana intermedia, es decir:
  - a. acceda directamente a la ventana de la contraseña saltándose la introducción del usuario.
  - b. acceda directamente a la ventana del Área personal saltándose la introducción del usuario y su contraseña.

|   |   |
|---|---|
| <b>Autenticación</b><br>Nombre de usuario <input type="text" value="gamma"/><br>Usuarios anteriormente autenticados <input type="text" value="gamma"/> <input type="button" value="v"/><br><input type="button" value="Siguiente"/> | <b>Autenticación</b><br>Nombre de usuario <input type="text" value="inconnu"/> <b>Usuario inexistente en el sistema</b><br>Usuarios anteriormente autenticados <input type="text" value="alfa"/> <input type="button" value="v"/><br><input type="button" value="Siguiente"/> |
| <b>Autenticación</b><br>Contraseña <input type="password" value="....."/><br><input type="button" value="Siguiente"/>   | <b>Autenticación</b><br>Contraseña <input type="password"/><br><b>No existen esas credenciales en el sistema</b><br><input type="button" value="Siguiente"/>  |

**Área personal**  
 Bienvenido **Amazona Peláez** (usuario **gamma**)  
 N° de autenticaciones validas: 13

De izquierda a derecha y de arriba abajo: Formulario de usuario en el que se selecciona un usuario de la lista | Formulario de usuario con error de usuario inexistente | Formulario de contraseña con texto insertado | Formulario de usuario con error de credenciales inválidas | Área personal

### 3. Tests unitarios

Se llevarán a cabo 6 tests unitarios, 3 para cada proceso.

### 4. Comprobaciones a realizar

1. Al ejecutar por primera vez la parte del registro por pasos, la aplicación, se podrá acceder mediante la barra de navegación a todos los pasos, incluido el resumen, de manera que se mostrarán todos los formularios en su estado inicial y el paso resumen sólo con la información que no aparece vacía en la carga inicial como, por ejemplo, el departamento.
2. Una vez que se han ido rellenando y guardando datos en algún paso, al hacer clic en cualquiera de los pasos de la barra de navegación, se recuperará de la sesión la información de ese paso y se visualizaría, incluido el paso resumen.
3. Pruebas. Para cada formulario realizarán tres pruebas de envío:
  - a. Enviarlo sin rellenar ningún campo, es decir, como aparece según se carga.
  - b. Enviarlo con todos los campos rellenos.
  - c. Enviarlo con algunos campos rellenos.

## 5. Recomendación

Para toda página que se cargue en el navegador, visualizar su código fuente (botón derecho en la página ► Ver código fuente) y comprobar que el HTML sea correcto, que no tenga errores. Esto funciona seguro en Mozilla Firefox.

## 6. Requisitos generales

- Cada POJO dispondrá de, al menos:
  - Atributos (anotados, en el caso de requerirlo la lógica de negocio).
  - Un constructor sin parámetros y otro con todos los parámetros.
  - Getters y setters para todos los atributos.
  - Opcionalmente, métodos auxiliares, como toString().
  - Se recomienda el uso del generador de código Lombok.
- Toda **validación de lógica de negocio** se realizará **siempre en lado del servidor**, nunca en el lado del cliente salvo que se solicite expresamente. Por lo tanto, todos los campos textuales en todo formulario serán `<input type="text" ... />`
- Se organizará todo el código fuente .java en paquetes, estructurado según su funcionalidad (ej. **controller**, **model**, **config...**).
- Se organizarán los archivos estáticos en carpetas, según su contenido, de la forma **/css**, **/js**, **/img...**
- Todas las **colecciones estáticas** se guardarán en el archivo **Colecciones.java**.