Contenido

Parte I - Aplicación para la gestión de usuarios

Enunciado

Tomando como ejemplo la aplicación de gestión de tareas, desarrolla una aplicación que nos permita **gestionar usuarios**.

De cada usuario se necesita almacenar la información que se detalla a continuación. Hay que tener en cuenta las restricciones indicadas en cada campo.

- **Nombre:** Cadena de texto entre 5 y 20 caracteres (campo requerido).
- **Apellidos:** Cadena de texto entre 5 y 20 caracteres (campo requerido).
- **Dni:** Cadena de texto de longitud 9 (campo requerido).
- **Correo electrónico:** Cadena de texto de formato email (campo requerido).
- **Código postal:** Campo numérico de longitud 5 (campo requerido).
- **Teléfono móvil:** Cadena de texto de longitud mínima 9 (campo requerido).
- **Fecha de nacimiento:** Campo de tipo fecha (campo requerido).
- **Password:** Cadena de texto (validación: ver parte II) (campo requerido).

Las restricciones debes validarlas en las vistas que se plantean a continuación. En la parte II de este enunciado, se especifican las validaciones que deberás hacer en la parte del servidor.

La aplicación necesitará **gestionar usuarios**. Para ello, necesitaremos poder llevar a

cabo las **operaciones CRUD** (Create, Read, Update y Delete) junto con la de **búsqueda** por nombre/email.

Recuerda que deberás:

- Crear la entidad `Usuario` con los atributos correspondientes y situarla en el paquete `model.entities`
- Crear el repositorio de usuarios `UsuariosRepository` y situarlo en el paquete `model.repositories`
- Crear el controlador `UsuarioController` y las acciones necesarias. Sitúalo en el paquete `controller`.
- Crear las `vistas` establecidas en el siguiente apartado.

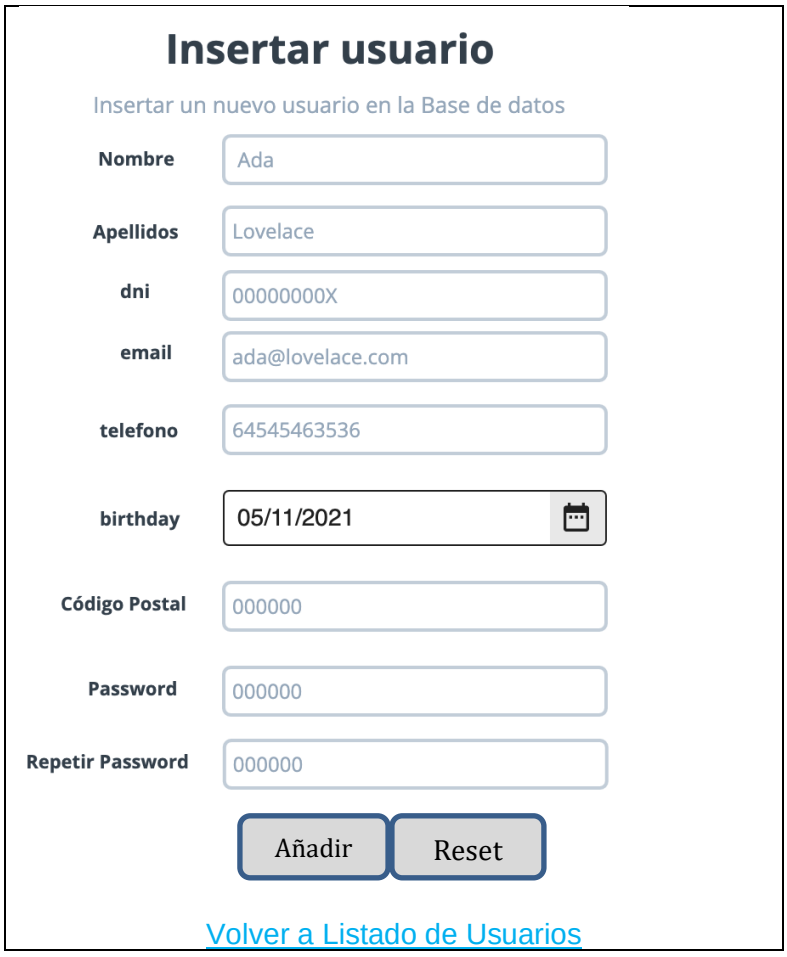
C/ Societat Unió Musical, 8 - 03802 Alcoi
Tel. 966 52 76 60 - Fax 966 52 76 61
03012165.secret@gva.es
www.cipfpbatoi.es

Vistas a desarrollar

1. Inserción de usuario

Cuando el usuario se inserte correctamente, nos enviará a la página de listado en la que mostraremos todos los usuarios del sistema.

Añade a la vista un enlace que permita volver a la vista *Listado de usuarios.*

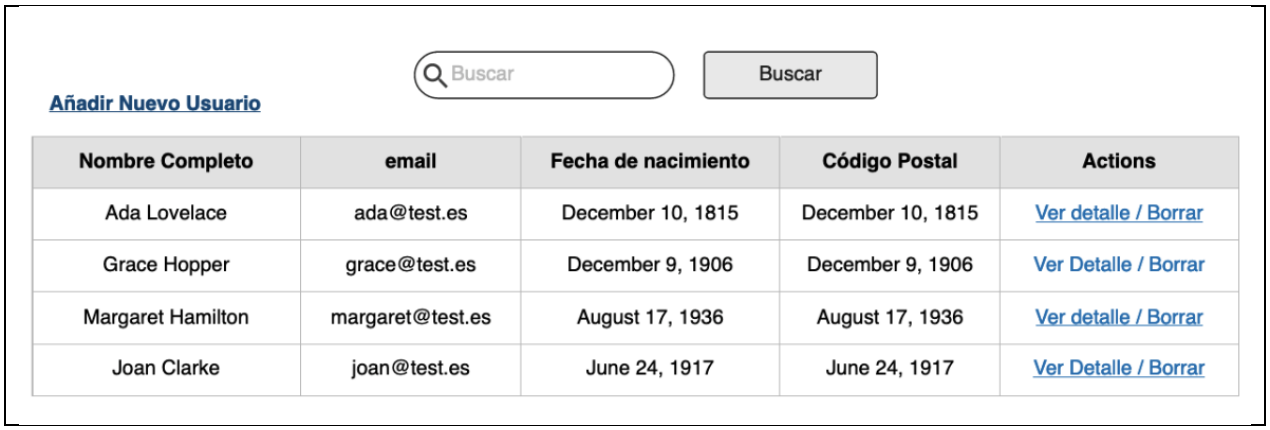


2. Listado de usuarios

El listado de usuarios se mostrará en forma de tabla.
Debe contener un formulario de búsqueda en la parte superior que, al enviarlo, filtrará (buscará) el listado de los usuarios que se van a visualizar (puedes hacerlo a partir del nombre, del correo o de ambos)**.** Si no hay resultados se mostrará el texto “No hay resultados”.

Además, al igual que ocurría en el listado de tareas, cada fila del listado debe tener las opciones para visualizar el detalle de un usuario (accederá a la vista Detalle de usuario) o borrarlo (lo eliminará y se mantendrá en la vista actual).

En la parte superior se dispondrá de un enlace que accede a la vista *Inserción de usuarios*



3. Detalle de usuario

El detalle del usuario se visualizará mostrando todos los datos en un formulario similar al de inserción de usuario, pero en este caso, ningún campo será editable. La fecha de nacimiento debe mostrarse en formato español.

Se dispondrá de un enlace que permita volver a la vista *Listado de usuarios*.

C/ Societat Unió Musical, 8 - 03802 Alcoi
Tel. 966 52 76 60 - Fax 966 52 76 61
03012165.secret@gva.es
www.cipfpbatoi.es

4. Modificación de un usuario (opcional)

La modificación implica **añadir un nuevo enlace (Editar) en cada fila del listado de usuarios**, de manera que permita acceder a la edición de un usuario ya creado.

La vista será igual a la de *Inserción de usuario,* donde **el único campo que no se podrá editar será el *DNI*.**

Se debe disponer de un enlace que permita volver al *Listado de usuarios*.

Parte II - Validación de datos

Enunciado

Uno de los retos que se nos presenta cuando se lleva a cabo la implementación de una nueva aplicación es la validación de los datos de entrada del usuario. El objetivo es que la información que maneje la aplicación sea siempre íntegra y correcta: un email, un número de teléfono, un DNI, un código postal, una contraseña o, incluso, que un texto cumpla con la longitud requerida.

Un ejemplo de ello sería la información que almacenamos en una BD como Postgresql o Mariadb/Mysql. En el momento en el que se define la tabla de usuarios, indicamos las restricciones como son el ***tipo de campo*** *y su* ***longitud.*** *Será el SGBD quien se encargue de validarlas.*

En esta actividad, implementaremos una clase `Validator` con métodos estáticos que nos permitirán validar la entrada de datos de los diferentes programas que vayamos implementando. **Tienes ya una versión de esta clase en el ejemplo resuelto de la gestión de tareas.**

```
public class Validator {

private static final String NUMERIC_REGEXP = "\\d+";

private static final String ALFABETIC_FIRST_UPPERCASE_REGEXP = "^[A-
Z][a-zA-Z]*$";

public static boolean isValidNumberOfLength(String param, int
maxLength) {
  return isNotEmptyOrNull(param) && param.length() <= maxLength &&
  param.matches(NUMERIC_REGEXP);
}

public static boolean isValidText(String param) {
  return isNotEmptyOrNull(param) &&
  param.matches(ALFABETIC_FIRST_UPPERCASE_REGEXP);
}

private static boolean isNotEmptyOrNull(String param) {
  return param != null && !param.isEmpty();
```

C/ Societat Unió Musical, 8 - 03802 Alcoi
Tel. 966 52 76 60 - Fax 966 52 76 61
03012165.secret@gva.es
www.cipfpbatoi.es


```
}

}
```

Tómala como base y añade métodos que nos permitan validar los siguientes campos:

- **Nombre y Apellidos:** Al menos **5 caracteres** siendo el primero en mayúscula
- **DNI:**

  Se considera válido si cumple el siguiente patrón **00000000X** donde:
  - **0** puede ser cualquier número.
  - **X** puede ser **TRWAGMYFPDXBNJZSQVHLCKE** y debe corresponderse con el cálculo de la letra de los números anteriores.
- **Código postal español (00000)**
  - **Contiene exactamente 5 dígitos**
  - Además**:**
    - Las **2 primeras cifras** hacen referencia a la **provincia** y debe ser un numero entre el **01 al 52.**
    - Las **3 últimas cifras** hacen referencia al distrito y debe ser un número entre el **000** al **999.**
- **Número de teléfono móvil**
  - Debe empezar por el prefijo internacional: **0034, +34 o 34.**
  - Seguido del número de móvil: **9 dígitos que empieza obligatoriamente por 6 o 7.**
- **Correo electrónico**
  - Puede contener: letras mayúsculas y minúsculas del alfabeto inglés y números del 0 al 9, pero no empezar por número.
  - Puede contener el carácter punto, pero no al inicio, al final o repetirse consecutivamente.
  - **En la parte del dominio, además de lo anterior tendremos que permitir** guiones "-" pero no al principio o final y tampoco de manera consecutiva.
- **Password**: Cadena de texto entre **5 y 20 caracteres** debe contener al menos letras mayúsculas, minúsculas y un carácter especial

Para validar los tipos de campos se hará uso de **expresiones regulares**, similar a lo realizado en el ejemplo de gestión de tareas con una clase `Validator`.

Aplica estas validaciones en las inserciones y modificaciones de los usuarios mostrando:

1. Los **mensajes de error** correspondientes manteniéndose en el formulario actual con los valores que se introdujeron antes de iniciar la validación (no se debe resetear el formulario si ocurre un error).
2. Si la operación ha sido exitosa (se pudo modificar o insertar un usuario), se mostrará un **mensaje de confirmación** en la vista del listado de usuarios.