



Esta práctica se podrá realizar en equipos de hasta 2 alumnos (pertenecientes al mismo grupo de prácticas) y su entrega es obligatoria.

## Instrucciones

Los casos de usos 1 - 8 son de carácter obligatorio, para que el trabajo pueda ser evaluado esos 8 casos deben estar implementados, con sus correspondientes pruebas automatizadas (no se evaluarán los casos de uso que no contengan pruebas), la puntuación máxima a la que se opta por implementar estos 8 casos de forma perfecta es de 7 sobre 10.

Los casos de uso 9 – 15 son de carácter opcional, la puntuación máxima a la que se opta por implementar los 15 casos de forma perfecta es de 10 sobre 10.

\*Defensa del trabajo presentado: aquellos alumnos que sean requeridos deberán realizar una defensa presencial del trabajo en el laboratorio de prácticas, consistente en la implementación de nuevos casos de uso, la nota de la práctica se verá condicionada a la correcta implementación de esos casos de uso.

## Casos de uso

### 1 Público: registrarse como usuario

Los usuarios deben poder registrarse en la aplicación aportando un email, nombre y una contraseña (que deberá repetirse dos veces y coincidir entre sí).

El email del usuario no podrá estar repetido en el sistema, se debe informar al usuario de los errores en el proceso de registro.

### 2 Público: iniciar sesión

Suministrando su email y contraseña, un usuario podrá autenticarse ante el sistema. Sólo los usuarios que proporcionen correctamente su email y su contraseña podrán iniciar sesión con éxito.

En caso de que el inicio de sesión fracase, será necesario mostrar un mensaje de error indicando el problema.

En caso de que el inicio de sesión sea correcto se debe dirigir al usuario a la vista que “lista todos los usuarios de la aplicación”

### 3 Usuario registrado: listar todos los usuarios de la aplicación

Se visualizará en una lista todos los usuarios de la aplicación. Para cada usuario se mostrará su nombre y email.

La lista debe incluir un sistema de paginación y mostrar 5 usuarios por página.

Debe incluirse una opción de menú principal, visible solo para usuarios en sesión que permita acceder al listado de todos los usuarios de la aplicación.

### 4 Usuario registrado: buscar entre todos los usuarios de la aplicación

Incluir un sistema que permita realizar una búsqueda por nombre y email de usuario. El cuadro de búsqueda contendrá un único campo de texto, la cadena introducida en ese campo utilizará para buscar coincidencias tanto en el campo nombre como en el email de los usuarios. Por ejemplo, si escribimos la cadena “mar” deberá retornar usuarios en los que la cadena “mar” sea parte de su nombre o su email.

El resultado de la búsqueda debe ser una lista que muestre las coincidencias encontradas, esta lista debe incluir un sistema de paginación y mostrar 5 usuarios por página.



## **5 Usuario registrado: enviar una invitación de amistad a un usuario**

Junto a la información de cada usuario presente en la vista de “listar todos los usuarios de la aplicación” debe aparecer un botón con el texto “agregar amigo”. Al pulsar el botón el usuario en sesión esta enviando una invitación de amistad a ese usuario.

## **6 Usuario registrado: listar las invitaciones de amistad recibidas**

Se visualizará en una lista todas las invitaciones de amistad recibidas por el usuario identificado en sesión. Para cada invitación se mostrará el nombre del usuario de la aplicación que solicito la amistad.

Debe incluirse una opción de menú principal, visible solo para usuarios en sesión que permita acceder al listado de todas las invitaciones de amistad recibidas.

La lista de invitaciones debe incluir un sistema de paginación y mostrar 5 invitaciones por página.

## **7 Usuario registrado: aceptar una invitación recibida**

Junto a cada invitación mostrada en la lista de invitaciones de amistad recibidas se debe incluir un botón con el texto “aceptar”, al pulsar este botón la invitación de amistad debe desaparecer de la lista de invitaciones y el usuario que la envió pasará a ser un amigo del usuario en sesión y viceversa (A es amigo de B, B es amigo de A).

## **8 Usuario registrado: listar los usuarios amigos**

Se visualizará en una lista todos los usuarios amigos del usuario en sesión. Para cada usuario se mostrará su nombre y email.

La lista de amigos debe incluir un sistema de paginación y mostrar 5 usuarios por página.

Debe incluirse una opción de menú principal, visible solo para usuarios en sesión que permita acceder al listado de todos los amigos del usuario en sesión.

## **- Opcionales - - - - -**

## **9 Usuario registrado: crear una nueva publicación**

Se debe incluir una nueva vista con un formulario para que un usuario en sesión pueda crear una nueva publicación. Las publicaciones consistirán en un título y un texto, al crear la publicación se debe almacenar: que usuario realizo la publicación, la fecha en la que se creó, el título y el texto asociados a la publicación.

Debe incluirse una opción de menú principal, visible solo para usuarios en sesión que permita acceder al formulario de crear publicaciones.

## **10 Usuario registrado: listar mis publicaciones**

Se visualizará en una lista con todas las publicaciones realizadas por el usuario en sesión. Para cada publicación se mostrará: la fecha en la que se creó y el título de la publicación.

No es necesario incluir sistema de paginación en este listado.

Debe incluirse una opción de menú principal, visible solo para usuarios en sesión que permita acceder al a la lista de publicaciones de un usuario.



## **11 Usuario registrado: listar las publicaciones de un usuario amigo**

Se modificará la vista que muestra la lista con los amigos del usuario en sesión para poder acceder a las publicaciones de un amigo. Al pulsar sobre el nombre del amigo se nos debe dirigir a una lista donde se visualicen todas sus publicaciones. Para cada publicación se mostrará: la fecha en la que se creó y el título de la publicación y el texto.

No es necesario incluir sistema de paginación en este listado.

## **12 Usuario registrado: crear una publicación con una foto adjunta**

Modificar el caso de uso 9, para que un usuario pueda incluir una fotografía al agregar una nueva publicación. El fichero de la fotografía debe ser subido al servidor utilizando un formulario (la foto puede ser renombrada en el servidor para que tenga el mismo nombre que la id de la publicación).

La fotografía debe ser opcional, puede incluirse en la publicación o no.

Modificar el caso de uso 11, al listar las publicaciones de un usuario amigo debe aparecer la foto adjunta (si es que la tiene) junto al resto de datos de la publicación.

## **13 Público: iniciar sesión como administrador**

Se debe incluir una vista especial /admin/login que permita iniciar sesión exclusivamente a usuarios de tipo administrador. El usuario con perfil administrador indicará su email y contraseña. Sólo los usuarios que tengan perfil administrador y proporcionen correctamente su email y su contraseña podrán iniciar sesión con éxito.

Debe haber usuarios con perfil administrador podemos indicar que un usuario es administrador incluyendo un campo extra en sus datos. Se puede crear un usuario administrador de prueba e insertarlo desde código o directamente en la base de datos, no hace falta que se puedan registrar administradores a través de la aplicación.

En caso de que el inicio de sesión fracase, será necesario mostrar un mensaje de error indicando el problema.

En caso de que el inicio de sesión sea correcto se debe dirigir al usuario a la vista que “lista todos los usuarios de la aplicación”

## **14 Consola de administración: listar todos los usuarios de la aplicación**

Un usuario identificado con perfil de administrador debe poder acceder a una lista en la que figuren todos los usuarios de la aplicación. Para cada usuario se mostrará su email y nombre.

No es necesario incluir sistema de paginación en este listado.

## **15 Consola de administración: Consola de administración: eliminar usuario**

En la lista anterior donde figuran todos los usuarios de la aplicación se debe agregar un botón con el texto “Eliminar” junto al nombre de cada usuario. Al pulsar el botón de eliminar se debe eliminar toda la información relativa al usuario, sus datos, sus publicaciones (sí es que las tiene) y debería dejar de figurar como amigo de otros usuarios.

## **Pruebas automatizadas**

Se deberá suministrar un proyecto Java JUnit con un mínimo de pruebas unitarias empleando el framework Selenium (Se suministrará en el campus virtual un proyecto plantilla de ejemplo).

Las clases de equivalencia mínimas (válidas e inválidas) que deberán realizarse serán:



- 1.1 [RegVal] Registro de Usuario con datos válidos.
- 1.2 [RegInval] Registro de Usuario con datos inválidos (repetición de contraseña invalida).
- 2.1 [InVal] Inicio de sesión con datos válidos.
- 2.2 [InInVal] Inicio de sesión con datos inválidos (usuario no existente en la aplicación).
- 3.1 [LisUsrVal] Acceso al listado de usuarios desde un usuario en sesión.
- 3.2 [LisUsrInVal] Intento de acceso con URL desde un usuario no identificado al listado de usuarios desde un usuario en sesión. Debe producirse un acceso no permitido a vistas privadas.
- 4.1 [BusUsrVal] Realizar una búsqueda valida en el listado de usuarios desde un usuario en sesión.
- 4.2 [BusUsrInVal] Intento de acceso con URL a la búsqueda de usuarios desde un usuario no identificado. Debe producirse un acceso no permitido a vistas privadas.
- 5.1 [InvVal] Enviar una invitación de amistad a un usuario de forma valida.
- 5.2 [InvInVal] Enviar una invitación de amistad a un usuario al que ya le habíamos invitado la invitación previamente. No debería dejarnos enviar la invitación, se podría ocultar el botón de enviar invitación o notificar que ya había sido enviada previamente.
- 6.1 [LisInvVal] Listar las invitaciones recibidas por un usuario, realizar la comprobación con una lista que al menos tenga una invitación recibida.
- 7.1 [AcepInvVal] Aceptar una invitación recibida.
- 8.1 [ListAmiVal] Listar los amigos de un usuario, realizar la comprobación con una lista que al menos tenga un amigo.
- 9.1 [PubVal] Crear una publicación con datos válidos.
- 10.1 [LisPubVal] Acceso al listado de publicaciones desde un usuario en sesión.
- 11.1 [LisPubAmiVal] Listar las publicaciones de un usuario amigo
- 11.2 [LisPubAmiInVal] Utilizando un acceso vía URL tratar de listar las publicaciones de un usuario que no sea amigo del usuario identificado en sesión.
- 12.1 [PubFot1Val] Crear una publicación con datos válidos y una foto adjunta.
- 12.1 [PubFot2Val] Crear una publicación con datos válidos y sin una foto adjunta.
- 13.1 [AdInVal] Inicio de sesión como administrador con datos válidos.
- 13.2 [AdInInVal] Inicio de sesión como administrador con datos inválidos (usar los datos de un usuario que no tenga perfil administrador).
- 14.1 [AdLisUsrVal] Desde un usuario identificado en sesión como administrador listar a todos los usuarios de la aplicación.
- 15.1 [AdBorUsrVal] Desde un usuario identificado en sesión como administrador eliminar un usuario existente en la aplicación.
- 15.2 [AdBorUsrInVal] Intento de acceso vía URL al borrado de un usuario existente en la aplicación. Debe utilizarse un usuario identificado en sesión pero que no tenga perfil de administrador.

Para cada una de estas pruebas se debe añadir al menos un caso de prueba y en caso querer añadir más casos se numerará la prueba (por ejemplo, RegVal1, RegVal2, RegVal 3...).

En caso de desear incluir más se deberán incluir al final de la clase de pruebas JUnit.



Todas las pruebas deben incluir el click en las opciones de menú correspondientes, etc, etc, igual que lo haría un usuario real.

## El protocolo de prueba

Para probar cada proyecto el profesor realizará los siguientes pasos:

- Lanzar la base de datos suministrada por el alumno (/data/startup.bat).
- Ejecutar la aplicación Spring Boot
- Importará el proyecto sdi2-test en eclipse.
- Ejecutar el proyecto sdi2-test.

## Aspecto Generales

### Seguridad

Deberán tenerse en cuenta los siguientes aspectos de seguridad:

- Emplear la técnica de autenticación/autorización más adecuada a este contexto.
- En caso necesario comprobar el rol de cada usuario para el acceso a funcionalidades dependientes de rol.
- Registrar el acceso de los usuarios y la actividad en un Logger (por ejemplo org.slf4j.Logger ).

### Arquitectura

La aplicación deberá estar obligatoriamente diseñada siguiendo el patrón arquitectónico visto en clase. Usando siempre de forma correcta controladores, servicios, repositorios y vistas. La utilización incorrecta de elementos de esta arquitectura será fuertemente penalizada (por ejemplo, implementar parte de la lógica de negocio de la aplicación en un controlador, acceder a un repositorio desde un controlador, configurar una vista desde un servicio, etc.).

### Otros aspectos que serán evaluados

- Claridad y calidad de la implementación del código Java y cumplimiento de *Java Code Conventions*.
- Calidad de la implementación de las vistas y usando todas las funcionalidades vistas en clase: fragmentos, replace, etc.
- Internacionalización de la aplicación, debe estar disponible en inglés y español.
- Validaciones en los formularios

### Entrega

El número n de cada trabajo deberá ser el UO del alumno (en caso de un integrante) o la concatenación de los UO( en caso de dos ej: UO0000UO0001) Según ese número se deberá crear los proyectos eclipse y las pruebas unitarias con los nombres **sdi1-n** y **sdi1-test-n** respectivamente. Se deberá subir a la tarea correspondiente un archivo ZIP (usando el formato ZIP) con el nombre sdi1-n.zip (en minúsculas) y que deberá contener en su raíz:

- Un documento PDF (con el mismo nombre que el proyecto sdi1-n.pdf) que contenga:



- Una descripción clara y detallada de la implementación de cada uno de los casos de uso implementados.
- Un catálogo de las pruebas unitarias realizadas y descripción sencilla de cada una
- Cualquier otra información necesaria para una descripción razonablemente detallada de lo entregado y su correcto despliegue y ejecución.
- El proyecto Spring Boot en formato carpeta (no comprimido) con el nombre `.\sdi1-n`
- El proyecto Java eclipse con los casos de prueba en formato carpeta (no comprimido) con el nombre `.\sdi1-test-n`
- En resumen, el zip deberá contener en su raíz:
  - `sdi1-n.pdf`
  - `sdi1-n`
  - `sdi1-test-n`

## Fecha máxima de entrega

La fecha de entrega será dependiente del grupo de laboratorio, coincidiendo con el día ANTERIOR en el que se imparte la práctica 7 el grupo al que pertenece el alumno, la hora límite serán las 20:00.

Por ejemplo, el grupo del L1 del Miércoles, deberá realizar la entrega el martes 20/03

## Evaluación

La calificación máxima que se puede obtener será de 11 puntos, en base al siguiente reparto: 7 puntos pertenecen a la funcionalidad obligatoria, 3 puntos a la parte opcional y 1 punto a la documentación.

Se penalizará:

- Que la compilación del código genere “*warnings*”.
- Que se presenten problemas durante el despliegue.