



PRÁCTICA 1 DE ENTREGA – SPRING

Extraordinaria adelantada

MY SOCIALNETWORK

Contenido

INSTRUCCIONES	2
ENUNCIADO	2
REQUISITOS OBLIGATORIOS	3
1. PÚBLICO: REGISTRARSE COMO USUARIO	3
2. USUARIO REGISTRADO: INICIO DE SESIÓN	3
3. USUARIO REGISTRADO: FIN DE SESIÓN	4
4. USUARIO REGISTRADO: LISTADO DE USUARIOS	4
5. USUARIO REGISTRADO: BUSCAR USUARIOS	4
6. USUARIO REGISTRADO: ENVIAR INVITACIÓN DE AMISTAD	5
7. USUARIO REGISTRADO: LISTAR LAS INVITACIONES DE AMISTAD RECIBIDAS	5
8. USUARIO REGISTRADO: ACEPTAR UNA INVITACIÓN	5
9. USUARIO REGISTRADO: LISTADO DE AMIGOS	5
10. INTERNACIONALIZACIÓN DE TODAS LAS VISTAS	6
11. SEGURIDAD	6
12. INFORME OBLIGATORIO	6
REQUISITOS OPCIONALES	7
13. USUARIO REGISTRADO: CREAR UNA NUEVA PUBLICACIÓN	7
14. USUARIO REGISTRADO: LISTADO DE MIS PUBLICACIONES	7
15. USUARIO REGISTRADO: LISTADO DE PUBLICACIONES DE UN AMIGO	7
16. USUARIO REGISTRADO: CREAR UNA PUBLICACIÓN CON UNA FOTO ADJUNTA	8
17. ADMINISTRADOR: LISTADO DE USUARIOS	8
18. ADMINISTRADOR: BORRADO MÚLTIPLE DE USUARIOS	8
PRUEBAS AUTOMATIZADAS	9
EL PROYECTO A ENTREGAR Y EL PROTOCOLO DE PRUEBA	9
ASPECTO GENERALES	9
1. SEGURIDAD	9
2. ARQUITECTURA	9
3. ENTREGA	10
4. FECHA MÁXIMA DE ENTREGA	10
EVALUACIÓN	10

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



Instrucciones

Los requisitos del 1 - 12 son de carácter obligatorio, para que el trabajo pueda ser evaluado esos 14 requisitos deben estar implementados, con sus correspondientes pruebas automatizadas (***no se evaluarán los casos de uso que no contengan todas sus pruebas***), la puntuación máxima a la que se opta por implementar estos 14 requisitos de forma perfecta es de 7 sobre 11.

Los casos de uso 13 – 18 son de carácter opcional, la puntuación máxima a la que se opta por implementar los N casos de forma perfecta es de 11 sobre 11.

***Prueba de autoría:** Todos los alumnos que presenta en **extraordinaria** 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.

Nota: ***TODOS Los alumnos que se presentan en extraordinaria deberán realizar este trabajo de forma individual obligatoriamente y su entrega también es obligatoria.***

Enunciado

Se trata de desarrollar una aplicación Web de tipo red social en el que existirán perfiles de usuario de tipo: Público (Anónimo), Usuario Registrado (Administrador y Usuario Estándar). Los requisitos de cada uno de los perfiles serán:

REQUISITOS OBLIGATORIOS (Hasta un 7.0)

Público. Las opciones públicas serán:

1. Registrarse como usuario del sistema. (0.5 puntos)

Usuario Registrado: Un usuario Administrador o Estándar podrá:

2. Iniciar sesión. (0.25 puntos)
3. Salir de sesión. (0.25 puntos)

Usuario Estándar. Los usuarios estándares tendrán las siguientes capacidades:

4. Ver listado de usuarios de la red social. (0.5 puntos)
5. Buscar usuarios por diferentes campos de la mismo. (0.5 puntos)
6. Enviar invitación de amistad. (0.5 puntos)
7. Ver el Listado de invitaciones de amistad recibida. (0.5 puntos)
8. Aceptar una invitación amistad. (0.5 puntos)
9. Ver listado de amigos. (0.5 puntos)

Internacionalización (Hasta 1.0 puntos)

10. Internacionalización páginas. (1.0 punto)

Seguridad. (Hasta 1.0 puntos)

11. Vulnerabilidad de recursos y acciones (1,0 puntos)

Informe obligatorio. (Hasta 1 puntos)

12. Escribir un informe obligatorio del proyecto. (1,0 puntos)

OPCIONALES (Hasta 4 puntos)

13. Crear una nueva publicación. (1.0 punto)



14. Ver el listado de mis publicaciones (0.5 punto)
15. Ver el listado de las publicaciones de un amigo (0.5 punto)
16. Crear una publicación con foto adjunta (1.0 punto)

Usuario Administrador. El administrador podrá:

17. Ver el Listado de los usuarios del sistema. (0.5 puntos)
18. Borrar múltiples usuarios del sistema. (0.5 puntos)

Requisitos OBLIGATORIOS

1. Público: registrarse como usuario

Los usuarios deben poder registrarse en la aplicación aportando un email, nombre, apellidos 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.

Una vez registrado un usuario será autenticado automáticamente pasando a la vista de opciones de usuario registrado.

Pruebas Funcionales

[Prueba1] Registro de Usuario con datos válidos.

[Prueba2] Registro de Usuario con datos inválidos (email vacío, nombre vacío, apellidos vacíos).

[Prueba3] Registro de Usuario con datos inválidos (repetición de contraseña inválida).

[Prueba4] Registro de Usuario con datos inválidos (email existente).

2. Usuario Registrado: Inicio de 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”

Caso 1: Usuario con perfil de administrador

- Sólo existirá un usuario administrador en el sistema con email admin@email.com y contraseña admin. 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 sea correcto se debe redirigir al usuario a la vista que contenga las opciones del **perfil Administrador**. Ejemplo: dirigir al usuario a la vista que “lista todos los usuarios de la aplicación”

Caso 2: Usuario Estándar



- En caso de que el inicio de sesión sea correcto se debe dirigir al usuario a la vista que contenga con las opciones del **perfil Usuario**. Ejemplo: dirigir al usuario a la vista que “lista todos los usuarios de la aplicación”

Pruebas Funcionales

[Prueba5] Inicio de sesión con datos válidos (administrador).

[Prueba6] Inicio de sesión con datos válidos (usuario estándar).

[Prueba7] Inicio de sesión con datos inválidos (usuario estándar, campo email y contraseña vacíos).

[Prueba8] Inicio de sesión con datos válidos (usuario estándar, email existente, pero contraseña incorrecta).

3. Usuario Registrado: Fin de sesión

Disponer una opción de menú que permita finalizar la sesión enviando al usuario al formulario de Inicio de sesión. Este botón solo se mostrará si un usuario se ha autenticado previamente.

Pruebas Funcionales

[Prueba9] Hacer click en la opción de salir de sesión y comprobar que se redirige a la página de inicio de sesión (Login).

[Prueba10] Comprobar que el botón cerrar sesión no está visible si el usuario no está autenticado.

4. Usuario registrado: Listado de usuarios

Se visualizará en una lista todos los usuarios de la aplicación (excepto los usuarios administradores y el usuario autenticado). 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.

Pruebas Funcionales

[Prueba11] Mostrar el listado de usuarios y comprobar que se muestran todos los que existen en el sistema.

5. Usuario registrado: Buscar usuarios

Incluir un sistema que permita realizar una búsqueda por nombre, apellidos 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, apellidos 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.

Pruebas Funcionales

[Prueba12] Hacer una búsqueda con el campo vacío y comprobar que se muestra la página que corresponde con el listado usuarios existentes en el sistema.

[Prueba13] Hacer una búsqueda escribiendo en el campo un texto que no exista y comprobar que se muestra la página que corresponde, con la lista de usuarios vacía.



[Prueba14] Hacer una búsqueda con un texto específico y comprobar que se muestra la página que corresponde, con la lista de usuarios en los que el texto especificados sea parte de su nombre, apellidos o de su email.

6. Usuario registrado: Enviar invitación de amistad

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

Pruebas Funcionales

[Prueba15] Desde el listado de usuarios de la aplicación, enviar una invitación de amistad a un usuario. Comprobar que la solicitud de amistad aparece en el listado de invitaciones (punto siguiente).

[Prueba16] Desde el listado de usuarios de la aplicación, enviar una invitación de amistad a un usuario al que ya le habíamos enviado 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.

7. 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 solicitó 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.

Pruebas Funcionales

[Prueba17] Mostrar el listado de invitaciones de amistad recibidas. Comprobar con un listado que contenga varias invitaciones recibidas.

8. Usuario registrado: Aceptar una invitación

Junto a cada invitación mostrada en la lista de invitaciones de amistad recibidas se debe incluir un botón/enlace con el texto “aceptar”, al pulsar este un botón/enlace 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).

Pruebas Funcionales

[Prueba18] Sobre el listado de invitaciones recibidas. Hacer click en el botón/enlace de una de ellas y comprobar que dicha solicitud desaparece del listado de invitaciones.

9. Usuario registrado: Listado de amigos

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

Pruebas Funcionales

[Prueba19] Mostrar el listado de amigos de un usuario. Comprobar que el listado contiene los amigos que deben ser.



10. Internacionalización de todas las vistas

Internacionalizar todas las vistas implementadas.

Pruebas Funcionales

[Prueba20] Visualizar al menos cuatro páginas en Español/Inglés/Español (comprobando que algunas de las etiquetas cambian al idioma correspondiente). Ejemplo, Página principal/Opciones Principales de Usuario/Listado de Usuarios.

11. Seguridad

Deberá diseñarse adecuadamente la política de seguridad con Spring Security para que no existan situaciones de vulnerabilidad en el acceso a recursos/acciones de usuarios registrados.

Pruebas Funcionales

[Prueba21] Intentar acceder sin estar autenticado a la opción de listado de usuarios. Se deberá volver al formulario de login.

[Prueba22] Intentar acceder sin estar autenticado a la opción de listado de invitaciones de amistad recibida de un usuario estándar. Se deberá volver al formulario de login.

[Prueba23] Estando autenticado como usuario estándar intentar acceder a una opción disponible solo para usuarios administradores (Se puede añadir una opción cualquiera en el menú). Se deberá indicar un mensaje de acción prohibida.

12. Informe OBLIGATORIO

Se deberá entregar un informe técnico en formato PDF así como un catálogo de casos de prueba en formato XLSX (que a su vez es formulario de autoevaluación). Se suministran dos plantillas que deberán tomarse como base para la elaboración de dichos documentos:

- Una plantilla Word para el informe (sdi-entrega1-n.docx). (Se entregará en formato PDF).
- Una plantilla Excel para los casos de prueba (sdi-entrega1-n.xlsx). (Se entregar en formato Excel).

Ambos documentos deberán ser renombrados cambiando la n por el código GIT de los miembros del equipo (ej. sdi-entrega1-801-804.pdf, sdi-entrega1-801-804.xlsx).

El contenido del informe deberá ser el siguiente:

- En la portada se deberá incluir:
 - Los datos personales de los autores: Nombre y apellidos, Emails de ambos, y Código ID GIT de cada uno y
 - URL del proyecto en AWS.
- Mapa de Navegación.
- Una descripción clara y detallada aquellos aspectos técnicos y/o diseño que considere relevantes.
- Cualquier otra información necesaria para una descripción razonablemente detallada de lo entregado y su correcto despliegue y ejecución (Versiones de java, hsqldb, spring, etc.).

La plantilla suministrada para cumplimentar los casos de prueba realizados presenta dos hojas:



- Una (denominada Pruebas) con una serie de tablas, de las cuales el alumno deberá rellenar aquella indicada en amarillo (Casos de Prueba) y donde deberá reflejar los casos de prueba que haya diseñado, el estado de dicho caso de prueba y una explicación/aclaración sobre el mismo en caso de fallo o no haber rellenado el caso. También en el caso de incluir casos de prueba extra deberá incluir en la columna explicación/aclaración en que consiste dicha prueba.
- Una segunda hoja (Instrucciones) con las instrucciones para cumplimentar la hoja primera.

Requisitos OPCIONALES

13. 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 realizó 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.

Pruebas Funcionales

[Prueba24] Ir al formulario crear publicaciones, rellenarla con datos válidos y pulsar el botón Submit. Comprobar que la publicación sale en el listado de publicaciones de dicho usuario.

[Prueba25] Ir al formulario de crear publicaciones, rellenarla con datos inválidos (campo título vacío) y pulsar el botón Submit. Comprobar que se muestra el mensaje de campo obligatorio.

14. Usuario registrado: Listado de mis publicaciones

Se visualizará en una lista 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.

Pruebas Funcionales

[Prueba26] Mostrar el listado de publicaciones de un usuario y comprobar que se muestran todas las que existen para dicho usuario.

15. Usuario registrado: Listado de publicaciones de un 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.

Pruebas Funcionales

[Prueba27] Mostrar el listado de publicaciones de un usuario amigo y comprobar que se muestran todas las que existen para dicho usuario.



[Prueba28] Utilizando un acceso vía URL u otra alternativa, tratar de listar las publicaciones de un usuario que no sea amigo del usuario identificado en sesión. Comprobar que el sistema da un error de autorización.

16. Usuario registrado: Crear una publicación con una foto adjunta

Modifica el formulario de crear publicaciones 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 listado de publicaciones, 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.

Pruebas Funcionales

[Prueba29] Desde el formulario de crear publicaciones, crear una publicación con datos válidos y una foto adjunta. Comprobar que en el listado de publicaciones aparece la foto adjunta junto al resto de datos de la publicación.

[Prueba30] Crear una publicación con datos válidos y sin una foto adjunta. Comprobar que la publicación se a creado con éxito, ya que la foto no es obligaría.

17. Administrador: listado de usuarios

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, nombre y apellidos.

No es necesario incluir sistema de paginación en este listado.

Pruebas Funcionales

[Prueba31] Mostrar el listado de usuarios y comprobar que se muestran todos los que existen en el sistema.

18. Administrador: Borrado múltiple de usuarios.

En la lista anterior donde figuran todos los usuarios de la aplicación se debe poder seleccionar múltiples usuarios (con un checkbox) y se dispondrá de un botón “**Eliminar**” para confirmar el borrado de todos aquellos seleccionados. Al pulsar el botón de Eliminar se deben eliminar todos los usuarios seleccionados, así como toda la información relativa a los mismos (datos, publicaciones, debería dejar de figurar como amigo de otros usuarios, etc.).

Pruebas Funcionales

[Prueba32] Ir a la lista de usuarios, borrar el primer usuario de la lista, comprobar que la lista se actualiza y dicho usuario desaparece.

[Prueba33] Ir a la lista de usuarios, borrar el último usuario de la lista, comprobar que la lista se actualiza y dicho usuario desaparece.

[Prueba34] Ir a la lista de usuarios, borrar 3 usuarios, comprobar que la lista se actualiza y dichos usuarios desaparecen.



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 se encuentran al final de cada requisito.

El proyecto a entregar y el protocolo de prueba

El proyecto Spring Boot que se deberá entregar deberá el proyecto Spring Boot en el carpeta src/main/java y en la carpeta src/test/java todas las clases de prueba tal cual se ha indicado en la clase sobre Selenium. De este modo este proyecto se puede ejecutar como proyecto Spring Boot (Run As Spring Boot App) o como proyecto de pruebas (Run As JUnit Test).

Para probar cada proyecto el profesor realizará los siguientes pasos:

- a) Lanzar la base de datos hsqldb suministrada por el alumno
- b) Importar y ejecutar la aplicación Spring Boot
- c) Importar y ejecutar el proyecto sdi1-test

Aspecto Generales

1. 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).

2. 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.
- Diseño de la interfaz de la aplicación.

3. Entrega

El número n de cada trabajo será el ID GIT del alumno. Según ese número se deberá crear un proyecto STS Spring Boot/Junit con sus pruebas unitarias (tal cual se ha visto en clase todo en un mismo proyecto) con el nombre **sdi2021-entrega1-n-ext-adel**.

Según lo anterior la entrega consistirá en los siguientes puntos:

- 1) Subir el proyecto a un repositorio GIT con nombre **sdi2021-entrega1-n-ext-adel** e invitar al usuario **sdigithubuniovi** como colaborador. Sobre dichos repositorios deberán realizarse actualizaciones frecuentes para que se pueda hacer un seguimiento del ritmo de trabajo.
- 2) Subir a la tarea del Campus Virtual correspondiente un archivo ZIP (usando el formato ZIP) con el nombre **sdi2021-entrega1-n-ext-adel.zip** (en minúsculas) y que deberá contener en su raíz:
 - El INFORME OBLIGATORIO en formato PDF con nombre **sdi2021-entrega1-n-ext-adel.pdf** y el archivo Excel con el catálogo de casos de prueba (**sdi2021-entrega1-n-ext-adel.xlsx**).
 - El proyecto Spring Boot/Junit en formato carpeta (no comprimido) con el nombre. **\sdi-entrega1-n-ext**
 - Una copia **completa** de la base de datos que utiliza la aplicación (toda la carpeta hsqldb renombrada como hsqldb-n). (Tanto si los datos se generan automáticamente mediante código como si no, es necesario incluir esta carpeta)
 - En resumen, el zip deberá contener en su raíz (MUY IMPORTANTE: el proyecto Spring Boot/Junit deberán renombrarse con la n correspondiente dentro de STS antes de exportarlos para meterlos en el zip):
 - **sdi2021-entrega1-n-ext-adel.pdf** (Archivo PDF suministrado y renombrado cambiado la n).
 - **sdi2021-entrega1-n-ext-adel.xlsx** (Archivo Excel suministrado y renombrado cambiado la n).
 - **sdi2021-entrega1-n-ext-adel** (Carpeta del proyecto Spring Boot/Junit renombrada cambiado la n)
 - **hsqldb-n** (Carpeta con la base de datos renombrada cambiado la n)

4. Fecha máxima de entrega

La fecha límite de entrega el 10 de enero 2021 a las 23:55. No se recibirá ningún trabajo fuera de plazo.

Evaluación

La calificación máxima que se puede obtener será de 11 puntos, en base al siguiente reparto: 7 puntos pertenecen a la funcionalidad obligatoria, 4 puntos a la parte opcional. Se penalizará:

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