



## Sistemas Distribuidos e Internet

### Curso 2022/2023

# PRÁCTICA 1 DE ENTREGA – SPRING MY WALLAPOP

<b>CONSIDERACIONES SOBRE LA EVALUACIÓN DE REQUISITOS .....</b>	<b>2</b>
1. COEVALUACIÓN HOLÍSTICA .....	2
<b>FECHA MÁXIMA DE ENTREGA.....</b>	<b>3</b>
<b>INTRODUCCIÓN Y RESUMEN DE LOS REQUISITOS.....</b>	<b>3</b>
<b>REQUISITOS OBLIGATORIOS .....</b>	<b>4</b>
1. PÚBLICO: REGISTRARSE COMO USUARIO .....	4
2. USUARIO REGISTRADO: INICIAR SESIÓN .....	4
3. USUARIO REGISTRADO: FIN DE SESIÓN .....	5
4. USUARIO ADMINISTRADOR: LISTADO DE USUARIOS DEL SISTEMA.....	5
5. USUARIO ADMINISTRADOR: BORRADO MÚLTIPLE DE USUARIOS .....	5
6. USUARIO REGISTRADO: DAR DE ALTA UNA NUEVA OFERTA .....	5
7. USUARIO REGISTRADO: LISTADO DE OFERTAS PROPIAS.....	6
8. USUARIO REGISTRADO: DAR DE BAJA UNA OFERTA .....	6
9. USUARIO REGISTRADO: BUSCAR OFERTAS .....	6
10. USUARIO REGISTRADO: COMPRAR UNA OFERTA .....	7
11. USUARIO REGISTRADO: VER EL LISTADO DE OFERTAS COMPRADAS.....	7
12. USUARIO REGISTRADO: ESTABLECER CONVERSACIÓN POR UNA OFERTA .....	7
13. USUARIO REGISTRADO: VER EL LISTADO DE CONVERSACIONES .....	8
14. INTERNACIONALIZACIÓN DE TODAS LAS VISTAS.....	8
15. SEGURIDAD .....	8
16. INFORME Y CATÁLOGO DE CASOS DE PRUEBA .....	10
<b>REQUISITOS OPCIONALES.....</b>	<b>12</b>
17. USUARIO REGISTRADO: ELIMINAR UNA CONVERSACIÓN .....	12
18. USUARIO REGISTRADO: MARCAR UNA OFERTA COMO DESTACADA .....	12
19. USUARIO REGISTRADO: CREAR OFERTAS CON UNA IMAGEN ADJUNTA .....	12
<b>PRUEBAS FUNCIONALES AUTOMATIZADAS .....</b>	<b>14</b>
<b>DATOS DE PRUEBA Y TIEMPO DE EJECUCIÓN DE PRUEBAS .....</b>	<b>14</b>
<b>ASPECTOS TRANSVERSALES EN LA EVALUACIÓN .....</b>	<b>14</b>
1. ARQUITECTURA DE LA PRÁCTICA .....	14
2. OTROS ASPECTOS.....	15
<b>PROCESO DE ENTREGA Y PROTOCOLO DE PRUEBA .....</b>	<b>15</b>



## Consideraciones sobre la evaluación de requisitos

A lo largo del enunciado se detallan los requisitos obligatorios y opcionales de la práctica de entrega. Para cada requisito, se detalla una funcionalidad y las pruebas automatizadas que deben implementarse correctamente para validarlo.

Para que el trabajo sea evaluado por el profesorado, todos los **requisitos obligatorios (del 1 al 16) deben estar implementados, así como todas las pruebas automatizadas asociadas al mismo**. La puntuación máxima a la que se opta por implementar a la perfección estos **16 requisitos es de 8.0 puntos**. La puntuación establecida para cada requisito es una puntuación máxima, **pudiendo llegar a ser negativa**, afectando a la puntuación total de la práctica. La **responsabilidad** de que no se cumpla lo especificado en este párrafo es, **en cualquier tipo de caso, grupal y no individual**.

Los **requisitos opcionales (del 17 – 19) permiten alcanzar una puntuación máxima de 11.0 puntos**, siempre y cuando se implementen todos a la perfección a nivel de funcionalidad y pruebas automatizadas. En perfección, se incluyen aspectos tales como: validación, registro en log e internacionalización. **La puntuación mínima de un ejercicio opcional será de 0.0 puntos**.

**Prueba de autoría:** Aquellos alumnos que sean requeridos por el profesorado, 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. El listado de alumnos y fecha de dicha prueba se publicará en el CV con suficiente antelación.

### 1. Coevaluación holística

La calificación final asignada a cada integrante de un equipo (Nota Individual del Estudiante, NIE) se calculará mediante el método denominado coevaluación holística: coevaluación, porque cada alumno va a valorar el rendimiento y comportamiento de sus compañeros, así como de uno mismo, y holístico porque esa valoración deberá ser una medida global (actitud, cumplimiento de plazos, ...) de sus compañeros en lo que compete al desarrollo de esta práctica.

Con el fin de simplificar ese cálculo cada alumno asignará una valoración entre varios niveles de posibles de Ciudadanía de Equipo (CE) a sus compañeros. Los valores de la escala Likert de CE van desde:

- “EXCELENTE” (100%): Contribución muy destacada y constante en el trabajo de equipo, con un rendimiento sobresaliente, hasta
- NO MOSTRADO (0%): No jugó un papel efectivo en el trabajo en equipo y/o asistencia y compromisos virtualmente inexistentes.

De esta forma cada alumno emitirá un listado de N valoraciones siendo un equipo de N miembros. Con la matriz de NxN valores de un equipo, el profesor correspondiente calculará para cada alumno de dicho equipo el Factor Individual de Coevaluación (FIC). De esta forma se calculará la Nota Individual de cada Estudiante (**NIE**) a partir del FIC y de la Nota obtenida por el Equipo en la práctica entregada (**NE**):

$$NIE = FIC * NE$$

Se enviará un mensaje desde el CV indicando cual será el procedimiento tanto para enviar los niveles de CE. Además, en clase se explicará con algo más de detalle cómo se realizará el cálculo del FIC.



## Fecha máxima de entrega

La fecha límite de entrega el día **domingo 19/03/2022 a las 23:55**. No se recibirá ningún trabajo fuera de plazo, ni por otra vía que no sea el campus virtual.

## Introducción y resumen de los requisitos

El objetivo de la práctica es desarrollar una aplicación Web basada en la compra-venta de artículos, similar a Wallapop. 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 8.0 puntos)

**Público.** Las opciones públicas serán:

1. Registrarse como usuario del sistema. (0.25 puntos)

**Usuario Registrado:** Un usuario Administrador o Estándar podrá:

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

**Usuario Administrador:** Un usuario Administrador podrá:

4. Consultar el listado de los usuarios del sistema. (0.25 puntos)
5. Borrar múltiples usuarios del sistema. (0.5 puntos)

**Usuario Estándar.** Un usuario Estándar podrá:

6. Dar de alta una nueva oferta (0.5 puntos).
7. Consultar el listado de ofertas propias (0.25 puntos).
8. Dar de baja una oferta propia (0.5 puntos).
9. Buscar ofertas por diferentes campos. (0.5 puntos)
10. Comprar una oferta. (0.5 puntos)
11. Consultar el listado de ofertas compradas. (0.5 puntos)
12. Establecer una conversación por una oferta. (0.75 puntos)
13. Ver el listado de conversaciones. (0.5 puntos)

### Internacionalización (Hasta 0.5 puntos)

14. Internacionalización de vistas. (0.5 puntos)

### Seguridad. (Hasta 1.0 puntos)

15. Vulnerabilidad de recursos y acciones. (1.0 puntos)

### Informe obligatorio. (Hasta 1.0 puntos)

16. Escribir un informe obligatorio del proyecto. (1.0 puntos)

### REQUISITOS OPCIONALES (Hasta 3.0 puntos)

17. Eliminar una conversación (0.50 puntos).
18. Marcar una oferta como destacada (1.0 puntos).
19. Crear ofertas con imagen adjunta (1.50 puntos).



## Requisitos OBLIGATORIOS

### 1. Público: Registrarse como usuario

Los usuarios deben poder registrarse en la aplicación introduciendo: 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.

Cada usuario comenzará con una cartera por valor de 100 Euros. Este dato se gestionará internamente sin necesidad de un campo de formulario.

#### 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: 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.

Cuando un usuario inicia sesión con éxito, debe mostrarse en cada una de las vistas su email y, junto a este, el valor de su cartera.

A continuación, se detalla cómo se debe proceder para cada uno de los perfiles existentes.

#### Caso 1: Usuario con perfil de administrador

- **Sólo existirá un usuario administrador en el sistema con email [admin@email.com](mailto:admin@email.com) y contraseña admin.** 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.
- En caso de que el inicio de sesión sea correcto se debe redirigir al usuario a la vista: “listado de todos los usuarios de la aplicación” [Requisito Obligatorio 4].

#### Caso 2: Usuario Estándar

- En caso de que el inicio de sesión sea correcto se debe redirigir al usuario a la vista “listado de ofertas propias” [Requisito Obligatorio 7].

#### 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

Incluir en el menú de navegación una opción que permita finalizar la sesión, enviando al usuario al formulario de Inicio de sesión. Este botón solo se mostrará exclusivamente si un usuario se ha autenticado previamente.

#### Pruebas Funcionales

[Prueba9] Hacer clic 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 Administrador: Listado de usuarios del sistema

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

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

### 5. Usuario Administrador: Borrado múltiple de usuarios

En la vista del [Requisito Obligatorio 4] 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, véase: datos, ofertas, conversaciones, etcétera. Un usuario Administrador no podrá borrarse a sí mismo.

#### Pruebas Funcionales

[Prueba12] Ir a la lista de usuarios, borrar el primer usuario de la lista, comprobar que la lista se actualiza y dicho usuario desaparece.

[Prueba13] Ir a la lista de usuarios, borrar el último usuario de la lista, comprobar que la lista se actualiza y dicho usuario desaparece.

[Prueba14] Ir a la lista de usuarios, borrar 3 usuarios, comprobar que la lista se actualiza y dichos usuarios desaparecen.

### 6. Usuario registrado: Dar de alta una nueva oferta

Un usuario identificado con perfil de Usuario Estándar, debe poder crear una oferta proporcionando: título descriptivo de la oferta, detalle textual de la oferta, fecha de alta de la oferta (puede ser la del sistema) y cantidad solicitada en euros.

Los tamaños y tipos de estos campos quedan a criterio del alumno. Deben existir validaciones del lado del servidor para cada uno de los siguientes valores: título, detalle y precio.



### Pruebas Funcionales

[Prueba15] Ir al formulario de alta de oferta, rellenarla con datos válidos y pulsar el botón Enviar. Comprobar que la oferta sale en el listado de ofertas de dicho usuario.

[Prueba16] Ir al formulario de alta de oferta, rellenarla con datos inválidos (precio negativo) y pulsar el botón Enviar. Comprobar que se muestra el mensaje de campo inválido.

## 7. Usuario registrado: Listado de ofertas propias

Un usuario identificado con perfil de Usuario Estándar debe poder acceder a una lista en la que figuren todas sus ofertas. Para cada oferta se mostrará: texto descriptivo de la oferta, detalle de la oferta y cantidad solicitada en euros.

No es necesario incluir sistema de paginación en esta vista.

### Pruebas Funcionales

[Prueba17] Mostrar el listado de ofertas para dicho usuario y comprobar que se muestran todas las que existen para este usuario.

## 8. Usuario registrado: Dar de baja una oferta

En el listado de ofertas propias, un Usuario Estándar podrá dar de baja una oferta. Para cada una de éstas, se presentará un botón/enlace “Eliminar” que, al hacer clic, eliminará la oferta y la información relativa a la misma.

Un usuario no podrá dar de baja una oferta que pertenezca a otro usuario.

### Pruebas Funcionales

[Prueba18] Ir a la lista de ofertas, borrar la primera oferta de la lista, comprobar que la lista se actualiza y que la oferta desaparece.

[Prueba19] Ir a la lista de ofertas, borrar la última oferta de la lista, comprobar que la lista se actualiza y que la oferta desaparece.

## 9. Usuario registrado: Buscar ofertas

Incluir un sistema que permita realizar una búsqueda de ofertas por su título. El cuadro de búsqueda contendrá un único campo de texto. Por ejemplo, si escribimos la cadena “coch” deberá devolver ofertas en las que la cadena “coch” sea parte de su título. Si la cadena está vacía deberá mostrar un listado completo con todas las ofertas existentes en el sistema.

En el listado resultante de la búsqueda, para cada oferta se mostrará: título, texto descriptivo de la oferta, detalle de la oferta y cantidad solicitada (en euros). A la derecha de cada oferta, se mostrará un enlace o botón “**Comprar**” si la oferta está disponible para la compra o bien el texto “**Vendido**” si la oferta ya ha sido vendida.

Esta lista debe incluir un sistema de paginación y mostrar 5 ofertas por página.

[Prueba20] Hacer una búsqueda con el campo vacío y comprobar que se muestra la página que corresponde con el listado de las ofertas existentes en el sistema

[Prueba21] 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 ofertas vacía.



## 10. Usuario registrado: Comprar una oferta

Sobre el listado resultante de una búsqueda de ofertas, un usuario podrá comprar una oferta haciendo clic en el botón “Comprar” correspondiente. Sólo se permite la compra de una oferta si:

- El contador de dinero del Usuario es igual o superior al precio de la misma.
- La oferta que desea comprar no está ya vendida.
- La oferta que desea comprar no es suya.

Al comprar una oferta, se deberán realizar las siguientes acciones:

- Comprobar en el servidor que se cumplen las condiciones anteriores.
  - Si no se cumple alguna de las condiciones anteriores, se mostrará un mensaje de error identificativo al usuario.
- Decrementar el contador de dinero del comprador en el precio que tenga la oferta.
- Marcar la oferta como comprada para evitar que vuelva a ser vendida.

### Pruebas Funcionales

[Prueba22] Sobre una búsqueda determinada (a elección del desarrollador), comprar una oferta que deja un saldo positivo en el contador del comprador. Comprobar que el contador se actualiza correctamente en la vista del comprador.

[Prueba23] Sobre una búsqueda determinada (a elección del desarrollador), comprar una oferta que deja un saldo 0 en el contador del comprador. Comprobar que el contador se actualiza correctamente en la vista del comprador.

[Prueba24] Sobre una búsqueda determinada (a elección del desarrollador), intentar comprar una oferta que esté por encima de saldo disponible del comprador. Y comprobar que se muestra el mensaje de saldo no suficiente.

## 11. Usuario registrado: Ver el Listado de ofertas compradas

Un usuario deberá disponer de una opción que le muestre el listado de ofertas que haya comprado mostrando para cada oferta los datos: título, detalle, precio y el email del vendedor.

No es necesario incluir sistema de paginación en este listado.

### Pruebas Funcionales

[Prueba25] Ir a la opción de ofertas compradas del usuario y mostrar la lista. Comprobar que aparecen las ofertas que deben aparecer.

## 12. Usuario registrado: Establecer conversación por una oferta

Tomando como punto de partida el listado resultante de una búsqueda de ofertas, un usuario podrá hacer clic en un botón/enlace “Conversación”. Las conversaciones son únicas para cada oferta. Es decir, el usuario A tendrá múltiples conversaciones con el usuario B, siempre y cuando sean para ofertas diferentes.

La primera vez que un usuario envíe un mensaje a una oferta se creará dicha conversación y, a partir de ahí, los mensajes enviados por ambos lados se irán incorporando a la conversación. Para cada mensaje se mostrará y almacenará: el autor del mensaje, la fecha/hora del mensaje y el texto del mismo (similar a WhatsApp).





Una propuesta de interfaz, sería un pequeño formulario donde escribir el mensaje y una tabla donde se muestran los mensajes de uno y otro.

Debe validarse en el servidor:

- El usuario A no podrá iniciar una conversación en una oferta del usuario A.
- No será posible enviar un mensaje vacío para iniciar una conversación o para continuarla.
- Los únicos usuarios que pueden participar en una conversación son: el usuario que inició la conversación y el dueño de la oferta.

### Pruebas Funcionales

[Prueba26] Sobre una búsqueda determinada de ofertas (a elección de desarrollador), enviar un mensaje a una oferta concreta. Se abriría dicha conversación por primera vez. Comprobar que el mensaje aparece en la conversación.

[Prueba27] Enviar un mensaje a una conversación ya existente accediendo desde el botón/enlace “Conversación”. Comprobar que el mensaje aparece en la conversación.

## 13. Usuario registrado: Ver el listado de conversaciones

Un usuario podrá acceder a su listado de conversaciones abiertas y podrá reanudar cada una de las conversaciones mediante un botón/enlace. El listado mostrará: el email del ofertante, el título de la oferta y el botón/enlace para reanudar la conversación.

### Pruebas Funcionales

[Prueba28] Mostrar el listado de conversaciones ya abiertas. Comprobar que el listado contiene la cantidad correcta de conversaciones.

## 14. Internacionalización de todas las vistas

Todas las vistas de la aplicación deben estar internacionalizadas, tanto a nivel de texto (incluyendo mensajes de validación) como los atributos de las etiquetas HTML que el grupo considere relevantes.

Se contemplan los idiomas español e inglés. Además de la implementación correcta de la internacionalización, el profesorado evaluará el uso correcto de la codificación.

La gramática y ortografía del español será evaluada en cualquier parte de la aplicación y documentación. El nivel del inglés empleado en la internacionalización no será evaluado, siempre y cuando cumpla unos mínimos, ejemplo: No traducir “página” como “página”, “pa” o similares.

### Pruebas Funcionales

[Prueba29] 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.

## 15. 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. Además, se evaluará en este apartado que:

- Se emplea la técnica de autenticación/autorización más adecuada al contexto.





- La existencia de comprobación basada en roles para el acceso a funcionalidades dependientes de rol específico.
- Registro de la actividad de los usuarios en un Logger (véase [org.slf4j.Logger](#)), incluyendo la fecha, la acción y la información que el grupo considere relevante para cada caso.

Casos:

- Registro de todas y cada una de las peticiones recibidas en el controlador correspondiente.
- Registro de las altas de usuarios en el sistema.
- Registro de los inicios de sesión tanto exitosos como fallidos.
- Registro de los cierres de sesión.
- Almacenar en una tabla de la base de datos la información de los cuatro tipos de logs indicados en el punto anterior (peticiones, altas, inicio de sesión con éxito y sin éxito y cierres de sesión.). La información mínima que se deberá almacenar en la base de datos será:
  - Para cada petición:
    - Tipo de log: "PET".
    - Fecha-Hora: Fecha y hora en que se registró el log en formato timestamp.
    - Texto descriptivo: Un texto que incluya el mapping del controlador que recibe la petición, el método http, y los parámetros recibidos si los hubiera.
  - Para cada alta:
    - Lo mismo que para cada petición, pero con tipo de log = "ALTA". En este caso el log aparecerá duplicado: uno por la petición y otro por el alta.
  - Para cada inicio de sesión con éxito:
    - Tipo de log: "LOGIN-EX".
    - Fecha-Hora: Fecha y hora en que se realizó el login en formato timestamp.
    - Texto descriptivo: el username del usuario que ha entrado en sesión.
  - Para cada inicio de sesión sin éxito:
    - Tipo de log: "LOGIN-ERR".
    - Fecha-Hora: Fecha y hora en que se realizó el intento de login en formato timestamp.
    - Texto descriptivo: el username del usuario que intentó entrar en sesión.
  - Para cada vez que el usuario sale de sesión:
    - Tipo de log: "LOGOUT".
    - Fecha-Hora: Fecha y hora en que se realizó el logout en formato timestamp.
    - Texto descriptivo: el username del usuario que salió de sesión.
- El usuario administrador podrá visualizar todos los logs ordenados por fecha-hora de más reciente a más antiguo. El listado podrá ser filtrado por cada tipo de log indicado arriba y el resultado siempre será ordenado por fecha de más reciente a más antiguo.
- El usuario administrador podrá borrar todos los logs con una opción de menú.



## Pruebas Funcionales

[Prueba30] Intentar acceder sin estar autenticado a la opción de listado de usuarios. Se deberá volver al formulario de login.

[Prueba31] 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.

[Prueba32] Estando autenticado como usuario estándar intentar acceder a una opción disponible solo para usuarios administradores (Añadir menú de auditoria (visualizar logs)). Se deberá indicar un mensaje de acción prohibida.

[Prueba33] Estando autenticado como usuario administrador visualizar todos los logs generados en una serie de interacciones. Esta prueba deberá generar al menos dos interacciones de cada tipo y comprobar que el listado incluye los logs correspondientes.

[Prueba34] Estando autenticado como usuario administrador, ir a visualización de logs, pulsar el botón/enlace borrar logs y comprobar que se eliminan los logs de la base de datos.

## 16. Informe y catálogo de casos de prueba

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). **Todos estos ficheros son OBLIGATORIOS.** Se suministran dos plantillas que deberán tomarse como base para la elaboración de dichos documentos:

- Una plantilla Word para el informe (sdi2223-entrega1-n.docx). (Entregar en formato PDF).
- Una plantilla Excel para los casos de prueba sdi2223-entrega1-n.xlsx). (Entregar en formato Excel).

**Ambos documentos deberán ser renombrados cambiando la n por el código del grupo asignado al equipo (ej. sdi2223-entrega1-22.pdf, sdi2223-entrega1-22.xlsx).**

El contenido del informe deberá ser el siguiente:

- En la portada se deberá incluir:
  - Los datos personales de los autores: Nombre y apellidos, IDGIT, Emails de UNIOVI, Código ID GIT de cada uno y código del equipo.
- Diagrama del modelado de las entidades y sus relaciones. Debe incluirse también un texto que explique las relaciones y las decisiones más relevantes que ha tomado el grupo en el modelaje.
- Diagrama de navegabilidad con texto explicativo. Un diagrama de navegabilidad no se compone de capturas de pantalla de la interfaz. Pueden usarse estas como apoyo en el texto explicativo, pero nunca como sustitutivo.
- Una descripción clara y detallada aquellos aspectos técnicos y/o diseño que se consideren relevantes a nivel de servicios, controladores, repositorios, vistas y sus relaciones. El objetivo no es explicar que son las capas y otros aspectos vistos en teoría. El objetivo es desarrollar qué decisiones se han tomado en base a lo visto en teoría.
- Cualquier otra información necesaria para una descripción razonablemente detallada de lo entregado y su correcto despliegue y ejecución (Versiones de java, hsqldb, spring, etc.). **Obligatorio usar las versiones de los softwares usados en clase.**
- Conclusión individual de cada miembro del grupo. Esto incluye la aportación en el desarrollo de la práctica, así como las dificultades encontradas durante el mismo. También qué ventajas y desventajas



se han detectado a nivel técnico y a nivel de trabajo en grupo. Una conclusión individual incompleta, repercutirá negativamente en la nota individual del alumno.

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

### 17. Usuario registrado: Eliminar una conversación

Sobre el listado anterior se podrá borrar una conversación incluyendo un botón/enlace “Eliminar” para cada conversación. Al pinchar sobre éste, se eliminará dicha conversación y todos los mensajes relacionados.

Una conversación solamente puede ser eliminada por los usuarios implicados en la conversación.

#### Pruebas Funcionales

[Prueba35] Sobre el listado de conversaciones ya abiertas. Pinchar el enlace Eliminar de la primera y comprobar que el listado se actualiza correctamente.

[Prueba36] Sobre el listado de conversaciones ya abiertas, pulsar el enlace Eliminar de la última y comprobar que el listado se actualiza correctamente.

### 18. Usuario registrado: Marcar una oferta como destacada

Un usuario puede marcar una oferta propia como Destacada pagando 20 euros (cantidad fija sin fecha de caducidad). Obviamente, debe validarse que tenga saldo suficiente para realizar la operación.

Una oferta podrá marcarse como Destacada tanto al crearla (mediante un checkbox en el formulario de creación de una oferta) como mediante un enlace en el listado de ofertas propias del usuario. Una vez destacada, la oferta no podrá volver a destacarse. Tampoco será posible deshacer el cambio.

Todas las ofertas destacadas se mostrarán a todos los usuarios en la vista “listado de ofertas propias” [Requisito Obligatorio 7], independientemente de que pertenezcan al usuario en sesión o no. Estas ofertas se presentarán ANTES del listado de ofertas propias.

#### Pruebas Funcionales

[Prueba37] Al crear una oferta, marcar dicha oferta como destacada y a continuación comprobar: i) que aparece en el listado de ofertas destacadas para los usuarios y que el saldo del usuario se actualiza adecuadamente en la vista del ofertante (-20).

[Prueba38] Sobre el listado de ofertas de un usuario con 20 euros (o más) de saldo, pinchar en el enlace Destacada y a continuación comprobar: que aparece en el listado de ofertas destacadas para los usuarios y que el saldo del usuario se actualiza adecuadamente en la vista del ofertante (-20).

[Prueba39] Sobre el listado de ofertas de un usuario con menos de 20 euros de saldo, pinchar en el enlace Destacada y a continuación comprobar que se muestra el mensaje de saldo insuficiente.

### 19. Usuario registrado: Crear ofertas con una imagen adjunta

Modifica el formulario de crear ofertas [Requisito Obligatorio 6] para que un usuario pueda incluir una imagen al dar de alta una oferta. El fichero de la imagen debe ser subido al servidor utilizando un formulario. El fichero de la imagen puede ser renombrado en el servidor para que tenga el mismo nombre que la id de la oferta.

La imagen debe ser opcional, puede incluirse en la oferta o no.

Modificar las vistas necesarias [Requisitos Obligatorios 7, 9, 11 y 18 (éste solo si se implementó)] para que, en los listados de ofertas, aparezca la imagen adjunta (si es que la tiene) junto al resto de datos de la oferta.



## Pruebas Funcionales

[Prueba40] Desde el formulario de dar de alta ofertas, crear una oferta con datos válidos y una imagen adjunta. Comprobar que en el listado de ofertas propias aparece la imagen adjunta junto al resto de datos de la oferta.

[Prueba41] Crear una oferta con datos válidos y sin una imagen adjunta. Comprobar que la oferta se ha creado con éxito, ya que la imagen no es obligatoria.



## Pruebas funcionales automatizadas

Se deberá suministrar un proyecto Java **JUnit5** en la carpeta `src/test/java` dentro del proyecto Spring Boot entregado con todas y cada una de las pruebas funcionales establecidas en cada uno de los requisitos obligatorios. En el caso de realizar requisitos opcionales, también es necesario implementar y adjuntar las pruebas establecidas.

El grupo puede añadir más pruebas, además de las planteadas en el enunciado, pero deberán incluirla en los ficheros correspondientes y en la documentación.

El desarrollo de las pruebas estará basado en el framework Selenium y JUnit5 vistos durante la asignatura.

## Datos de prueba y tiempo de ejecución de pruebas

Se deberá poblar la base datos al arrancar la aplicación con un mínimo de datos que permite ejecutar las pruebas solicitadas:

- Un único usuario administrador con las siguientes credenciales:
  - Login: [admin@email.com](mailto:admin@email.com)
  - Password: [admin](#)
- Usuarios registrados: al menos 15 (para que permitan al menos un listado de usuarios con 3 páginas de 5 usuarios por páginas). Seguir el patrón de nombrado siguiente:
  - Login [user01@email.com](mailto:user01@email.com), [user02@email.com](mailto:user02@email.com), ....
  - Password: user01
- Publicaciones por usuario: al menos 10 publicaciones por usuario que permitan un listado con páginas 5 publicaciones.

Un aspecto a tener en cuenta es que las pruebas no deben depender del resultado las anteriores. Por ello, debe pensarse en cómo poblar adecuadamente la base de datos para que cada prueba parta de un conjunto de datos conocido. Por otro lado, debe buscarse un compromiso entre este aspecto y el tiempo de ejecución de las pruebas.

## Aspectos transversales en la evaluación

La calificación máxima que se puede obtener será de 11 puntos, en base al siguiente reparto: 8.0 puntos pertenecen a la funcionalidad obligatoria, 3.0 puntos a la parte opcional

Si bien existen aspectos que se evaluarán de forma transversal a todos los requisitos establecidos en el presente enunciado. **Como normal general se considera que, si no se cumplen, penalizan a la nota obtenida en la práctica por el grupo.**

### 1. Arquitectura de la práctica

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 los elementos de esta arquitectura **será fuertemente penalizada**. Ejemplos: 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, etcétera.



## 2. Otros aspectos

A continuación, se enumeran otra serie de buenas prácticas (o requisitos) que penalizarán a la nota total del grupo en caso de incumplimiento:

- 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, etcétera.
- Validaciones correctas del lado del servidor, bien sea mediante peticiones directas o empleando formularios.
- Uso de un formato de URLs consistente entre las diferentes rutas de los controladores.
- Nombrado de ficheros de entrega y formatos correctos.
- Que la compilación del código genere “*warnings*”.
- Que se presenten problemas durante el despliegue.
- El encoding del proyecto debe estar en **formato UTF8** para evitar problemas con las pruebas. **Si se envía el proyecto con otro encoding** y las pruebas fallan, automáticamente los correspondientes **casos de uso no serán evaluados**.

## Proceso de entrega y protocolo de prueba

Según el número asignado a cada grupo, se deberá crear un proyecto Spring Boot/Junit5 con sus pruebas unitarias (tal cual se ha visto en clase todo en un mismo proyecto) con el nombre **sdi2223-entrega1-n**, donde el número n de cada trabajo será el identificador del equipo (publicado en el CV).

Según lo anterior la entrega consistirá en los siguientes puntos:

- 1) Subir el proyecto a un **repositorio privado en GitHub** con nombre **sdi2223-entrega1-n**. 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 de todos los alumnos.
- 2) Subir a la tarea del Campus Virtual correspondiente un archivo ZIP (usando el formato ZIP) con el nombre **sdi2223-entrega1-n.zip** (en minúsculas) y que deberá contener en su raíz:
  - El INFORME OBLIGATORIO en formato PDF con nombre **sdi2223-entrega1-n.pdf** y el archivo Excel con el catálogo de casos de prueba (**sdi2223-entrega1-n.xlsx**).
  - El proyecto Spring Boot/Junit5 en formato carpeta (no comprimido) con el nombre **\sdi2223-entrega1-n** (debe ser una copia literal de la carpeta del proyecto y **NO** de lo subido a Github)
  - 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). Lo recomendable es que los datos se generen automáticamente para evitar problemas de entrega en el campus virtual por el tamaño del fichero comprimido.
  - En resumen, el zip deberá contener en su raíz (MUY IMPORTANTE: el proyecto Spring Boot/Junit deberán renombrarse con la n correspondiente dentro de IntelliJ antes de exportarlos para incluirlo en el zip):
    - **sdi2223-entrega1-n.pdf** (Archivo PDF suministrado y renombrado cambiado la n).
    - **sdi2223-entrega1-n.xlsx** (Archivo Excel suministrado y renombrado cambiado la n).





- sdi2223-**entrega1**-n (Carpeta del proyecto Spring Boot/Junit renombrada cambiado la n)
- hsqldb-n (Carpeta con la base de datos renombrada cambiado la n)

El proyecto Spring Boot que se deberá entregar deberá contener el proyecto Spring Boot en la 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 o como proyecto de pruebas.

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) Ejecutar el proyecto de pruebas.