



Nombre y apellidos:

DNI:

| Examen Tercera Evaluación | |
|----------------------------------|-------------------------------------|
| Módulo | Contornos de Desenvolvemento |
| Fecha | 29/05/2025 |
| Hora | 16:00 (llamamiento 15:45) |
| Ciclo | DAM |
| Régimen | Adultos |
| Modalidad | Presencial |
| Duración | 2h 15' |
| Puntuación máxima | 7 |

Calificación

| Parte 1 - 2.5 puntos | | | Parte 2 - 4.5 puntos | | | Total |
|-----------------------------|-------------|-------------|-----------------------------|-------------|-------------|--------------|
| T1.1 | T1.2 | T1.3 | T2.1 | T2.2 | T2.3 | |
| 1 | 0.75 | 0.75 | 1.25 | 2 | 1.25 | 7 |
| | | | | | | |



Instrucciones generales

El examen se realizará bajo el **sistema operativo Windows** para realizar todas las actividades propuestas en él.

Tendrás que entregar una memoria, en formato .pdf donde harás las explicaciones y capturas de pantalla de la ejecución de cada una de las preguntas. El nombre del fichero tendrá el formato

apellido_nombre_examen_cd_t3.pdf

Es muy importante recoger en dicho informe todo aquello que se solicite en el enunciado de cada una de las tareas, para que éstas sean evaluables. Puedes utilizar como plantilla para dicho informe el enunciado del examen, y contestar después de cada una de las preguntas.

Para poder realizar alguno de los ejercicios, se proporcionan algunos ficheros adicionales. Puedes copiar todo el conjunto de ficheros para los ejercicios desde la tarea del aula virtual de la entrega del examen.

Además, también tendrás que entregar ficheros adicionales en algunos de los ejercicios. Se indicará en cada uno de ellos qué debes entregar.

Todos los archivos, incluida la memoria realizada, deben añadirse a un archivo comprimido llamado

apellido_nombre_examen_cd_t3.zip

Dentro de ese archivo .zip debes añadir la memoria en pdf y una carpeta de archivos por cada ejercicio, siguiendo el esquema **indicado al final del examen**. El archivo resultante tendrás que entregarlo en la tarea correspondiente del aula virtual.

Podrás responder al examen tanto en castellano como en gallego.

Temática

Aplicación para identificación de fauna y flora mediante IA, geolocalización y notificación de especies invasoras o peligrosas.

PARTE 1: Diagramas de comportamiento (60 min)

La empresa BioLoc quiere desarrollar una aplicación móvil llamada BioLocFinder para identificación de fauna y flora mediante IA, geolocalización y notificación de especies invasoras o peligrosas.

Se quiere que BioLocFinder permita a los ciudadanos:

- Registrarse (e iniciar sesión) como **usuario estándar** para:
 - Fotografiar fauna o flora y recibir una identificación automática mediante IA.
 - Opcionalmente, **tras la identificación**, registrar la localización de la especie detectada.
 - Notificar la presencia de especies peligrosas o invasoras a las autoridades (haciendo uso de los casos de uso previos)
- Acceder a un mapa de avistamientos (sin necesidad de registro).
- Consultar información sobre especies (sin necesidad de registro).
- Consultar una sección FAQ (Frequently Asked Questions) (sin necesidad de registro).
- Inscribirse como **usuario experto** para:
 - Validar avistamientos de especies.
 - Colaborar con las autoridades en la identificación de especies peligrosas o invasoras a partir de las notificaciones enviadas por los usuarios estándar.
- Los **administradores** de la aplicación podrán:
 - Gestionar los usuarios (dar de alta/baja, modificar datos).
 - Actualizar la sección FAQ (Frequently Asked Questions) de la aplicación.

Tarea 1.1 (1 punto)

Realiza el **diagrama de casos de uso con Draw.io** que represente a los **actores y casos de uso** descritos en el enunciado. Recuerda utilizar la **notación UML** adecuada, así como modelar correctamente las **relaciones** entre los actores y los casos de uso (incluyendo generalización, inclusión y extensión si fuera necesario).

Tarea 1.2 (0.75 puntos)

Describe el caso de uso "Notificar especie peligrosa", teniendo en cuenta su funcionamiento:

- Se activa al pulsar un botón "Notificar especie peligrosa" en la pantalla de identificación de especies.
- El usuario selecciona la especie peligrosa de una lista de especies peligrosas o invasoras.
- El usuario añade una fotografía o varias fotografías de la especie.
- El usuario añade la geolocalización de la especie.
- El usuario añade un comentario opcional.

- El usuario pulsa el botón "Enviar notificación".

Ten en cuenta las precondiciones y poscondiciones del caso de uso, así como los posibles flujos alternativos o puntos de inclusión o extensión, dependiendo del diagrama de casos de uso diseñado. Puedes utilizar la siguiente tabla como guía:

| Elemento | Descripción |
|------------------------|--|
| Nombre del caso de uso | Notificar especie peligrosa |
| Identificador | CU - 001 |
| Descripción | |
| Actor(es) | Usuario estándar |
| Precondiciones | Ser un ciudadano registrado en la aplicación. Haber fotografiado a una especie de flora y/o fauna y que la identificación recibida sea de peligroso/invasora. El avistamiento debe estar validado por usuarios expertos. |
| Curso normal | El usuario pulsar un botón "Notificar especie peligrosa" Añade una fotografía o varias fotografías de la especie. El usuario añade la geolocalización de la especie. El usuario pulsa el botón "Enviar notificación" |
| Curso alternativo 1 | El usuario puede añadir un comentario a la notificación o no. |
| Postcondiciones | Usuarios expertos y/o autoridades verifican la veracidad de la notificación. |
| Notas | |

Tarea 1.3 (0.75 puntos)

Una notificación de especie peligrosa puede pasar por diferentes **estados** hasta ser validada por un usuario experto y procesada por las autoridades competentes. De esta forma, se han definido los siguientes estados:

- **Pendiente:** Estado inicial de la notificación al ser creada. La notificación ha sido creada pero deben añadirse los datos necesarios a la notificación (especie, fotografía, geolocalización y comentario). En este estado, la notificación puede ser modificada o eliminada.
- **Enviada:** Estado en el que la notificación ha sido enviada a los revisores (usuarios expertos). En este estado, la notificación no puede ser modificada (pero sí eliminada).

- **En revisión:** Estado en el que la notificación ha sido asignada para su revisión a un usuario experto.
- **Validada:** Estado en el que la notificación ha sido validada por un usuario experto. En este estado, la notificación se considera válida y se puede enviar a las autoridades.
- **Rechazada:** Estado en el que la notificación ha sido rechazada por un usuario experto. En este estado, la notificación se considera inválida y no se puede enviar a las autoridades. No se puede deshacer el rechazo y la notificación llega a su fin.
- **Eliminada:** Estado en el que la notificación ha sido eliminada por el usuario (por ejemplo, si se ha creado por error). Puede ser eliminada mientras está en estado pendiente o enviada. No se puede deshacer la eliminación.
- **Enviada a autoridades:** Estado en el que la notificación ha sido enviada a las autoridades para su procesamiento, tras la validación del usuario experto. En este estado, la notificación no puede ser modificada ni eliminada.
- **Procesada:** Una vez la notificación ha sido procesada por las autoridades, se considera que la notificación ha sido gestionada. Es el último estado de la notificación.

Realiza el **diagrama de estados** utilizando **Draw.io** que represente los diferentes estados por los que puede pasar una notificación de especie peligrosa, así como las transiciones entre ellos.

Deberás crear el diagrama de estados en el **mismo archivo** en el que has creado el diagrama de casos de uso, en una pestaña diferente, y deberás entregar dicho archivo en tu entrega final con el nombre *apellidos_nombre_parte1_cd_t3.drawio*.

PARTE 2: Refactorización y documentación (75 min)

Como punto de partida para desarrollar **BioLocFinder**, se ha creado un proyecto en Java que incluye las siguientes clases:

- **Usuario.java:** clase que modela a un usuario de la aplicación.
- **UsuarioService.java:** clase que gestiona los usuarios de la aplicación y proporciona las funcionalidades de crear y eliminar usuarios, obtener el número máximo de capturas permitidas para un usuario en concreto, registrar una identificación por parte de un usuario o obtener un listado de usuarios ordenados según un criterio dado.
- **Main.java:** clase que contiene el método main de la aplicación y permite ejecutar la aplicación desde la línea de comandos. En esta versión, la aplicación permite registrar y borrar usuarios, listar usuarios ordenados por ciertos criterios, registrar identificaciones y comprobar el número de identificaciones restantes.

Tarea 2.1 (1.25 puntos)

Crea el archivo **README.md** con la información necesaria para documentar el proyecto en el directorio raíz del proyecto. Deberás **replicar el contenido del pdf adjunto**. Recuerda utilizar la sintaxis de Markdown para dar formato al texto. Evita, en la medida de lo posible, utilizar la



sintaxis HTML si hay una alternativa Markdown nativa.

Tarea 2.2 (2 puntos)

La aplicación desarrollada hasta el momento es funcional pero no está desarrollada siguiendo unas buenas prácticas de desarrollo: incumple algunos de los principios de desarrollo más importantes (DRY, KISS, YAGNI, SOLID) y tiene algunos problemas de diseño que dificultan su mantenimiento y escalabilidad.

Analiza el código en busca de **code smells**. Para cada uno que encuentres:

- Indica qué tipo de smell has detectado
- **Refactorízalo adecuadamente** aplicando la técnica correspondiente (renombrar métodos, extracción de clases, extracción de constantes, creación de jerarquías...)
- **Justifica** brevemente el cambio

Antes de empezar a refactorizar, analiza el código en profundidad y elige los cambios que consideres más adecuados: quizá empieces a cambiar una parte de la aplicación y, tras aplicar otro cambio, el primero ya no sea necesario.

A la hora de aplicar las refactorizaciones, deberás realizar una **captura de pantalla de la funcionalidad** que estés utilizando del IDE en al menos **3 acciones diferentes**.

Importante: no se debe eliminar ningún código, aunque haya métodos o clases que no se utilicen. Aunque el IDE puede proponernos eliminarlo (código muerto), suponemos para el examen que es código que se empleará en el futuro. Recuerda también que no se debe modificar el comportamiento de la aplicación, por lo que no debes cambiar la lógica de negocio.

Tarea 2.3 (1.25 puntos)

Documenta el código refactorizado de las clases **Usuario.java** y **UsuarioService.java** utilizando **comentarios Javadoc**. Asegúrate de que la documentación generada es clara y completa, y que incluye información sobre los métodos, parámetros y excepciones lanzadas.

Genera la documentación en formato HTML utilizando las funcionalidades proporcionadas por IntelliJ. Adjunta una captura de pantalla indicando cómo has generado la documentación HTML desde el IDE.



RESUMEN DE ENTREGA

A modo resumen, deberás entregar, tras realizar los ejercicios, un archivo comprimido con la siguiente estructura

- *apellido_nombre_examen_cd_t3.zip*
 - *apellido_nombre_examen_cd_t3.pdf (memoria)*
 - *parte1/*
 - *apellidos_nombre_parte1_cd_t3.drawio*
 - *parte2/*
 - *apellidos_nombre_parte2_cd_t3_readme.md*
 - *apellidos_nombre_parte2_cd_t3_doc_javadoc/*
 - *...*
 - *index.html*
 - *apellidos_nombre_parte2_cd_t3_app/*
 - *.idea*
 - *...*
 - *out*
 - *...*
 - *src*
 - *[todos los paquetes y archivos java necesarios]*
 - *pom.xml*