

CASO PRÁCTICAS

Navegación digital

**Interfaces
Persona
Computador
IPC – DSIC
UPV
Curso 2024-2025**



Índice

1.	Caso de Estudio	2
2.	Escenarios sobre el usuario	3
2.1.	Registrarse en la aplicación.....	3
2.2.	Autenticarse.....	4
2.3.	Cerrar sesión.....	4
2.4.	Realizar un problema.....	4
2.5.	Modificar perfil.....	4
2.6.	Mostrar resultados.....	5
3.	Escenarios sobre la carta.....	5
3.1.	Realizar zoom	5
3.2.	Marcar un punto.....	5
3.3.	Trazar una línea	5
3.4.	Trazar un arco	5
3.5.	Anotar texto	6
3.6.	Cambiar el color de una marca.....	6
3.7.	Eliminar una marca.....	6
3.8.	Limpiar la carta	6
3.9.	Desplazar el transportador para medir ángulos. (Tomar un ángulo / trazar línea con un ángulo)	6
3.10.	Tomar una distancia en la carta.....	7
3.11.	Marcar extremos de un punto en la carta	7
4.	Recursos para realizar la práctica	7
5.	Modelo de datos	7
	User	7
	Session.....	8
	Problem.....	8
	Answer	8
6.	Persistencia de los datos	8
7.	Proyecto base.....	9
8.	Instrucciones de Entrega	10

I. Caso de Estudio

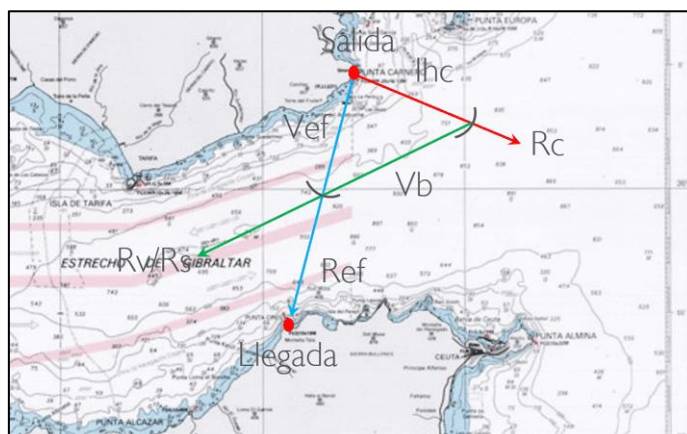
Se desea desarrollar una herramienta para facilitar la preparación de los ejercicios de navegación sobre la carta náutica del estrecho de Gibraltar en los exámenes de patrón de embarcación de recreo. A continuación, describimos como es este proceso.

Los exámenes de esta titulación son de tipo test, al alumno se le propone un enunciado de un problema y se les dan 4 respuestas, siendo una de ellas cierta.

32.- Desde la situación $I = 35^{\circ}45,0'N$ $L = 006^{\circ}09,8'W$ navegamos al rumbo efectivo $Ref = 042^{\circ}$ durante una hora, hasta la nueva situación $I = 35^{\circ}52,6'N$ $L = 006^{\circ}01,5'W$
¿Qué distancia hemos navegado entre las dos situaciones?

- A) $d = 12,6'$
- B) $d = 10,2'$
- C) $d = 9,2'$
- D) $d = 11,2'$

El alumno utiliza la carta náutica para hacer los cálculos necesarios y así poder escoger la respuesta correcta.



Sobre la carta náutica el alumno puede hacer las siguientes operaciones:

- marca puntos,
- trazar líneas entre dos puntos,
- dibujar circunferencias cuyo centro es uno de estos puntos.
- tomar distancias entre dos puntos, después traslada esta medida sobre la escala vertical de la carta para obtener una distancia real en millas
- escribir sobre la carta cálculos intermedios o anotaciones útiles para resolver el problema.

Para todo ello dispone de una carta, un lápiz, una goma de borrar, una regla para trazar líneas y tomar distancias entre dos puntos, un compás con el que dibujar arcos, y un transportador de ángulos cuadrado.



Así pues, la aplicación a desarrollar trata de digitalizar este proceso con el fin de evitar la carta náutica en papel. De esta manera el alumno podrá practicar con su ordenador, aunque el examen final se realizará siempre con la carta en papel y con las herramientas descritas.

Tenemos pues dos grupos de funciones de la aplicación a desarrollar. En primer lugar, hay un conjunto de funciones que facilitan al alumno enunciados de problemas con su correspondiente batería de respuestas, y que a su vez recogerá los aciertos y errores que cometa el alumno y mantendrá un histórico de sus acciones. Para ello el usuario se registrará y creará un perfil de usuario, cada vez que inicie una sesión la aplicación le propondrá problemas que el usuario deberá de resolver. Al finalizar la sesión se guardarán los aciertos y errores que se hayan cometido durante la sesión. Esta información almacenada se podrá visualizar a posteriori para ver la evolución en el aprendizaje del alumno.

Por otra parte, se dotará a la aplicación de la funcionalidad necesaria para que pueda pintar sobre una carta digital: puntos, líneas, arcos, y escribir anotaciones. Con la herramienta queremos dotar de un entorno digital similar al entorno físico en el que el alumno realizará el examen, es decir disponer de las mismas herramientas físicas que utilizará el día del examen, pero en formato digital. También podrá el alumno calcular ángulos y tomar distancias

Seguidamente se detallan los escenarios de uso que se han obtenido tras el análisis de requisitos del sistema y que deben utilizarse para diseñar e implementar adecuadamente la aplicación requerida. Estos escenarios se han agrupado en dos categorías acorde a lo descrito anteriormente.

2. Escenarios sobre el usuario

2.1.Registrarse en la aplicación

Juan se quiere preparar el examen de patrón de embarcación de recreo. En la academia que se ha apuntado, el profesor utiliza una herramienta digital que está disponible para descargar. Tras instalarla y para poder acceder al conjunto de problemas resueltos disponibles accede a la opción de registrarse.

El sistema le solicita que introduzca un nombre de usuario, que será utilizado para identificarle, una cuenta de correo electrónico válida, una contraseña, su fecha de nacimiento y si lo desea, una imagen para ser utilizada como avatar. Juan introduce como nombre de usuario “jgarcia”. Como contraseña, elige “passPER21!” y proporciona la cuenta de correo: “jgarcia@gmail.com”. Finalmente, introduce una fecha de nacimiento 14/03/2005. Como no le apetece buscar un avatar en ese momento, se queda con el que se le ofrece por defecto. Tras introducir toda la información pedida, solicita finalizar el proceso.

El sistema comprueba los datos introducidos y le indica que ese nombre ya está siendo utilizado y debe seleccionar un nuevo valor que contenga entre 6 y 15 caracteres o dígitos si espacios, pudiendo usar guiones o sub-guiones. Juan decide seleccionar el *nickname* “jpgarcia”.

El sistema comprueba que el resto de datos introducidos son válidos. La contraseña contiene entre 8 y 20 caracteres, incorpora al menos una letra en mayúsculas y minúsculas, así como algún dígito y algún carácter especial (!@#\$%&*()-+=). El correo electrónico tiene un formato válido y el usuario tiene más de 16 años. El sistema registra al usuario y le informa de ello.

2.2. Autenticarse

Juan tiene un rato libre y piensa en hacer unos ejercicios de navegación antes de ir a la academia. Abre la aplicación y selecciona la opción de autenticarse en el sistema. El sistema le solicita que introduzca su nombre de usuario o *nickname* y su contraseña. Juan introduce su *nickname* y su contraseña. El sistema comprueba que existe un usuario con esos datos y lo autentifica dándole acceso al resto de funciones.

2.3. Cerrar sesión

Juan está cansado de hacer ejercicios, y Susana quiere probar a ver si es capaz de hacer bien 4 ejercicios seguidos, en lugar de cerrar la aplicación, Juan cierra su sesión para que Susana pueda arrancar la suya.

2.4. Realizar un problema

Juan, tras autenticarse, le pide al sistema que le proponga un problema. Hoy no quiere pensar mucho y escoge que el sistema le proponga un problema de manera aleatoria. El sistema le propone un problema, le muestra el enunciado y las 4 respuestas. Antes de mostrar las respuestas el sistema siempre las ordena de manera aleatoria.

Después de realizar sus cálculos sobre la carta, Juan marca la respuesta que considera la correcta y le pide al sistema que verifique la solución.

Otro día Juan le pedirá al sistema que le muestre la lista de problemas y escogerá uno él mismo.

2.5. Modificar perfil

Juan no eligió un avatar cuando se dio de alta, pero se ha cansado de aparecer con el avatar por defecto, así que decide acceder a la opción de modificar los datos de su

perfil. El sistema le ofrece la información que tiene actualmente: su nombre de usuario, su contraseña, su dirección de correo electrónico y su fecha de nacimiento, así como el avatar que tiene asignado. Juan se da cuenta que puede modificar cualquier dato excepto su nombre de usuario. Decide cambiar su avatar entre algunas imágenes que tiene en su máquina. Tras realizar el cambio, solicita que se actualice la información.

El sistema comprueba que todos los cambios introducidos cumplen con los requisitos establecidos. La contraseña contiene entre 8 y 20 caracteres, incorpora al menos una letra en mayúsculas y minúsculas, así como algún dígito y algún carácter especial (!@#\$%^&*()-+=). El correo electrónico tiene un formato válido y el usuario tiene más de 16 años. Por tanto, el sistema actualiza la información e informa al usuario.

2.6. Mostrar resultados

Juan quiere saber cómo está evolucionando en la materia así que, después de autenticarse, le pide a la aplicación que le muestre la información sobre los aciertos y errores que ha cometido. El sistema le muestra sus datos de forma estructurada. Juan quiere ver esa información, pero solo en los últimos días, así que filtra los datos a mostrar indicando el día a partir del cual quiere que se muestren los resultados.

3. Escenarios sobre la carta

Al igual que ocurre en el entorno real en el que el usuario dispone de lápices de diferentes colores y puntas de diferente grosor, en las acciones que se van a describir a continuación y que así lo requieran, debe de estar la opción de escoger color y anchura del trazo.

3.1. Realizar zoom

Juan está realizando un ejercicio, pero no ve con detalle el nombre de un faro sobre la carta, así que hace una aproximación al detalle o zoom para acercar la vista. Después de leer el dato, decide volver a otra vista más alejada y hace un zoom para alejar la vista. El zoom se realiza sobre la carta digital.

3.2. Marcar un punto

Juan decide marcar un punto sobre la carta. Utiliza la herramienta de puntos y selecciona la posición sobre la carta. El sistema muestra el punto en la forma y color previamente establecido.

3.3. Trazar una línea

Juan quiere trazar una línea que pasa por dos puntos, así que accede a la herramienta de trazado de líneas. Tras seleccionar el punto de origen y el punto final, la línea queda marcada con el grosor y color preestablecido.

3.4. Trazar un arco

Juan quiere trazar un arco así que accede a la herramienta de trazado de arcos y, tras seleccionar el centro del arco y su radio, este queda marcado con el grosor y color

preestablecido. Este arco será de radio libre o de radio preestablecido, en este caso Juan toma una medida sobre el eje vertical de la carta

3.5. Anotar texto

Juan quiere anotar texto sobre la carta. Selecciona la herramienta de texto y, tras escoger el lugar donde se mostrará, escribe el texto correspondiente. Al finalizar, el sistema muestra el texto con el color y tamaño preestablecido.

3.6. Cambiar el color de una marca

Juan quiere cambiar el color de una marca (punto, línea, arco, texto), así que selecciona el color y la marca, y el sistema muestra la marca con la nueva apariencia.

3.7. Eliminar una marca

Juan no está contento con la última línea pintada, así que la selecciona y la elimina de la carta. En otro momento elimina también un punto, un arco y un texto.

3.8. Limpiar la carta

Juan ha resuelto el problema y quiere limpiar la carta para poder resolver otro problema. Escoge la opción de limpiar y el sistema le presenta una carta sin ninguna marca.

3.9. Desplazar el transportador para medir ángulos. (Tomar un ángulo / trazar línea con un ángulo)

Juan quiere saber el ángulo que forma una línea, para ello selecciona el transportador, desplaza el transportador por la carta hasta situar su centro sobre la línea. Una vez visualiza el punto de corte de la línea sobre el transportador, ya sabe cuál es el ángulo que forma esta línea sobre el norte. Poco después quiere trazar una línea desde un faro con un ángulo determinado. Para ello sitúa el centro del transportador en el faro. Para trazar una línea solo tendrá que marcar un punto en la carta próximo a la medida deseada del transportador

Tomar un ángulo:

Podemos indicar que tomar un ángulo es una tarea compuesta de varios de los escenarios descritos en el punto 3, es decir, para anotar el ángulo de una línea primero tendrá que pintar la línea, después desplazar el transportador y por último anotar el texto que considere. En el siguiente enlace podéis acceder a un video en el que se muestra un ejemplo de esta tarea de alto nivel: [anotar ángulo](#)

Trazar línea ángulo:

De manera similar al caso anterior, en el siguiente video mostramos una tarea de alto nivel que utiliza varios escenarios: [trazar línea ángulo](#)

3.10. Tomar una distancia en la carta

Juan quiere saber la distancia entre dos puntos, para ello utiliza la regla disponible, lleva la regla hasta que une los dos puntos y con la medida tomada mueve la regla hasta la escala vertical de la carta donde de manera manual calcula el número de subdivisiones de la escala que equivalen los centímetros medidos con la regla. Como la medida en centímetros no es relevante, este proceso se puede realizar con un compás, si se abre el compás con las puntas en los dos puntos a medir su distancia, se puede después trasladar el compás hasta la escala vertical de la carta y calcular las subdivisiones de la escala que abarca esta medida.

3.11. Marcar extremos de un punto en la carta

Juan quiere saber la latitud y la longitud de un punto marcado en la carta. Después de seleccionar la herramienta adecuada y seleccionar el punto el sistema, pinta sobre la carta dos líneas ortogonales que pasan por el punto y que cortan perpendicularmente los bordes de la carta. La latitud es el punto de corte sobre la escala vertical y la longitud es el punto de corte sobre la escala horizontal

4. Recursos para realizar la práctica

Para realizar la práctica se proporciona el fichero “carta_nautica.jpg” que se utilizará como base para realizar los diferentes casos de uso.

También se proporciona el fichero “transportador.jpg” como transportador cuadrado de ángulos. Para que este transportador resulte útil, es necesario que se le dote de transparencia. Los nodos en javaFX tienen la propiedad **Opacity** que permite variar su opacidad y hacerlos transparentes. No es obligado utilizar este recurso, el alumno puede utilizar otras estrategias o fuentes para disponer de un transportador de ángulos transparente. También se proporciona el recurso “regla.jpg” como regla que se utilizará para tomar medidas o pintar líneas si es el caso.

5. Modelo de datos

El estudio de los objetos de tarea ha dado como resultado los objetos: User, Session, Problem y Answer.

A continuación, se detalla la implementación en java de estas clases:

User

La clase User permite manejar la información de los usuarios del sistema. Los atributos de User son:

- String **nickName**: nombre de usuario. No puede ser actualizado.
- String **email**: dirección de correo electrónico del usuario.
- String **password**: contraseña del usuario.
- Image **avatar**: imagen de avatar del usuario.
- LocalDate **birthdate**: fecha de nacimiento del usuario.

- `ArrayList<Session> sessions`: una lista con todas las sesiones que ha iniciado el usuario y en la que se guarda el número de problemas resueltos correctamente y el número de problemas no resueltos correctamente.

Session

La clase `Session` almacena la información de cada sesión realizada por el usuario. Los atributos de un `Session` no pueden ser modificados tras haberse creado. Es por ello que el objeto se deberá de crear en el momento que el usuario cierra la sesión, bien de manera voluntaria o porque se cierra la aplicación, es en este momento cuando se conoce el número de aciertos o fallos durante la sesión.

Los atributos de la clase `Session` son:

- `LocalDateTime timeStamp`: día y hora en el que se inicia/registra la sesión.
- `int hits`: número de problemas resueltos correctamente
- `int faults`: número de problemas resueltos incorrectamente

Problem

La clase `Problem` se utiliza para guardar los problemas de navegación. En la base de datos se incluye un conjunto de problemas por lo que no es necesario que creéis objetos de este tipo. Los campos de la clase son:

- `StringProperty text`: propiedad con el texto del enunciado del problema.
- `ArrayList<Answer> answers`: lista con las cuatro respuestas, objetos del tipo `answer`.

Answer

La clase `Answer` se utiliza para guardar una respuesta de un problema de navegación. En la base de datos se incluye un conjunto de problemas con sus respuestas por lo que no es necesario que creéis objetos de este tipo. Los campos de la clase son:

- `StringProperty text`: propiedad con el texto de la respuesta.
- `BooleanProperty validity`: propiedad con el valor de certeza de la respuesta (es la correcta o no).

6. Persistencia de los datos

La persistencia de la información se va a realizar a través de una base de datos *SQLite*¹. En el proyecto se proporcionará la librería *IPC2025.jar*. En esta librería se define el modelo de datos descrito en el punto anterior, es decir NO tenéis que codificar las clases descritas en el modelo, tenéis que utilizar las clases ya implementadas en esta librería. Esta librería os permitirá almacenar y recuperar toda la información que necesite persistencia. En concreto la aplicación mantendrá un conjunto de problemas

¹ <https://www.sqlite.org/index.html>

de navegación con sus respuestas. También guardará la información de los usuarios, sus datos y de cada sesión iniciada se guardará también el número de problemas resueltos correctamente y el número de problemas no resueltos adecuadamente.

La librería gestiona todo el acceso a la base de datos, creándola de forma automática si es necesario dentro del directorio de vuestro proyecto, en un fichero denominado “database.db”. Si el sistema crea la base de datos esta estará vacía por lo que no se dispondrán de problemas que mostrar al usuario. Os vamos a dejar una base de datos que contiene un conjunto de problemas y un usuario genérico para poder probar vuestro trabajo, en el caso de que todavía no tengáis definido el registro de usuarios.

Se os proporciona un método capaz de volcar a un fichero de texto el contenido de la base de datos, por si necesitáis verificar su contenido durante el desarrollo del proyecto. No obstante, existen numerosas aplicaciones que os permiten abrir una base de datos *SQLite* y ver el contenido de sus tablas. Por ejemplo, podéis instalaros la aplicación gratuita “[DB Browser for SQLite](https://sqlitebrowser.org/)”².

Esta librería y la documentación para poder utilizarla en el proyecto se publicará en un segundo pdf.

7. Proyecto base

Para realizar el trabajo se proporciona el proyecto de Netbeans, **Poi_UPV**. Este proyecto muestra cómo se puede implementar la funcionalidad de zoom descrita en el escenario de tarea, y se puede utilizar como proyecto base para desarrollar el trabajo (no es obligatorio realizarlo a partir de este proyecto).

Para implementar el zoom se utiliza un *ImageView* que ocupa todo un contenedor. En el *ImageView* se muestra un mapa (similar a la carta náutica que se pide en el trabajo). El contenedor adecuado de *javaFX* para poder aplicar el zoom es el *ScrollPane*. En este proyecto se muestra cómo podemos incluir una imagen de un mapa en este contenedor y después aplicar un escalado. Cuando el tamaño de la imagen desborda el tamaño del contenedor debido a aplicar una función de escalado, aparecen de forma automática los scrolls que permiten movernos por todo el contenedor. Si hemos añadido objetos por detrás del *ImageView*, se mostrará por encima de este; y si aplicamos un escalado al contenedor, también se les aplicará a estos objetos. El escalado no afectará a la posición del nodo dentro del mapa, pues el escalado afecta a los dos nodos por igual, es decir mantendrá su posición (x,y) respecto al (0,0) de la imagen.

Para que esta estrategia funcione de manera adecuada es necesario utilizar objetos del tipo *Group* que no están disponibles en el *SceneBuilder*, el código necesario se incluye en el método *initialize()* de la clase controladora y está explicado mediante comentarios.

Se recomienda utilizar este proyecto como base del trabajo, eliminado o cambiando aquellos elementos que no resulten adecuados para la funcionalidad requerida.

² <https://sqlitebrowser.org/>

8. Instrucciones de Entrega

Los primeros pasos para desarrollar la entrega pasan por la realización del diseño conceptual y posterior diseño físico mediante prototipos de baja fidelidad.

Os recomendamos que reviséis con vuestro profesor de prácticas o de teoría los prototipos que realicéis. Tened en cuenta que en clase de teoría hemos terminado con el diseño conceptual y estamos empezando con el diseño físico, temas necesarios para poder realizar el diseño adecuado de la entrega.

En una publicación posterior os referenciaremos los métodos de la librería IPC2025.jar así como las instrucciones para poder añadir la librería al proyecto.

También os facilitaremos código de ayuda para poder dibujar nodos sobre la carta y/o desplazarlos.

La entrega se realizará en grupos de 3 alumnos pertenecientes al mismo grupo de laboratorio.