

# CASO PRÁCTICAS

Pizarra de navegación

**Interfaces  
Humano Maquina  
IHM – DSIC  
UPV  
Curso 2025-2026**

## Índice

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

## 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. Esta herramienta servirá para que el profesorado pueda mostrar sobre el proyector como se resuelven los diferentes tipos de ejercicios y también permitirá a los alumnos poder practicar la resolución de estos ejercicios al disponer la herramienta de conjunto de problemas, adicionalmente se facilitará al alumno la evolución de su aprendizaje.

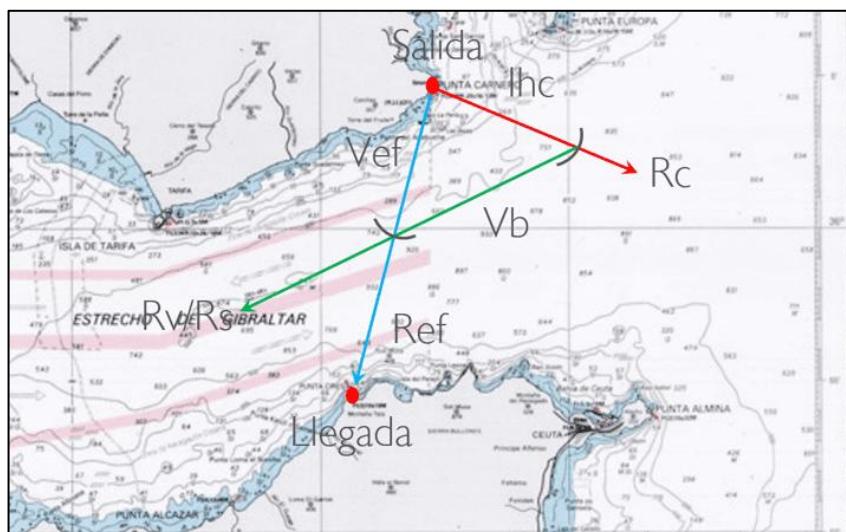
A continuación, describimos como es el proceso de realización de los ejercicios de navegación sobre la carta náutica:

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  $l = 35^{\circ}45,0'N$   $L = 006^{\circ}09,8'W$  navegamos al rumbo efectivo Ref =  $042^{\circ}$  durante una hora, hasta la nueva situación  $l = 35^{\circ} 52,6'N - L = 006^{\circ} 01,5'W$   
¿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,
- dibujar circunferencias o arcos a partir de un punto.

- tomar distancias entre dos puntos,
- medir ángulos,
- 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 tomar distancias, y un transportador de ángulos cuadrado con el que medir ángulos.



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.

Es muy importante enfatizar que el examen se realiza con una carta en papel y con las herramientas descritas, es por ello por lo que esta aplicación debería de simular este proceso de la manera más eficiente posible para que el uso de la aplicación, en lugar de la carta en papel, no penalice al alumno a la hora de realizar el examen.

Podemos diferenciar dos grupos de funcionalidad que debe de integrar la aplicación a desarrollar. En un primer grupo tenemos las funciones que nos permiten gestionar los datos de los alumnos (registro, login, visualización de histórico,) y de los problemas disponibles en la base de datos, y un segundo grupo que abarca las funciones de edición que puede realizar el alumno sobre la carta.

Seguidamente se detallan los escenarios de uso que se han obtenido tras el análisis de requisitos del sistema y que se deben de utilizar para realizar el diseño conceptual y posterior diseño físico. Todo este proceso debe de quedar documentado al igual que la implementación de la aplicación. Se aconseja utilizar un repositorio similar a github para mantener toda esa documentación.

Estos escenarios se han agrupado en dos categorías acorde a lo descrito anteriormente.

## 2. Escenarios generales

### 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.

Para 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](mailto:jgarcia@gmail.com)”. Finalmente, introduce una fecha de nacimiento. Como no le apetece buscar un avatar en ese momento, se queda con el que se le ofrece por defecto.

Tras introducir los datos Juan el sistema los valida y se da cuenta de que no cumplen con alguna de las restricciones. Tras avisarlo Juan modifica los datos para que sean aceptado y consigue terminar el registro con éxito.

Las verificaciones que debe de realizar el sistema son:

- Que el nombre de usuario no es utilizado.
- Que el nombre contenga entre 6 y 15 caracteres o dígitos sin espacios, pudiendo usar guiones o sub-guiones.
- Que 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 (!@#\$%&\*()-+=).
- Que el correo electrónico tenga un formato válido
- Que el usuario tenga más de 16 años.

### 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 están 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 iniciar su sesión. Juan ha realizado 4 ejercicios durante el tiempo que ha tenido la sesión abierta, esta información se guarda en el sistema y queda disponible para su consulta.

## 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 la proponga un problema de manera aleatoria. El sistema le propone un problema, le muestra el enunciado y las 4 respuestas. Por si Juan ya ha realizado este ejercicio con anterioridad, el sistema ordena las respuestas de manera aleatoria antes de mostrarlas.

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 muestra la lista completa de problemas y escogerá uno el 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 electrónica 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 y después actualiza la información.

## 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 de edición sobre la carta

El día del examen el alumno puede utilizar lápices o bolígrafos de diferentes grosores y diferentes colores, esta posibilidad también debería de estar disponible en la aplicación.

## 3.1. 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.

## 3.2. Trazar una línea

Juan quiere trazar una línea une isla Tarifa con Tanger, tras utilizar la herramienta adecuada pinta sobre la carta la línea que une ambos puntos en el color establecido.

### 3.3. Trazar un arco

Juan quiere trazar un arco así que accede a la herramienta de trazado de arcos, el compás y, tras seleccionar el centro traza el arco, este queda marcado con el grosor y color preestablecido.

### 3.4. 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.

### 3.5. 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.

### 3.6. Eliminar una marca

Juan no está contento con la última línea pintada, así que selecciona la goma y elimina la línea de la carta. Seguro que en otro momento elimina algún punto, arco y/o texto.

### 3.7. 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 nueva.

### 3.8. 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 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](#)

### 3.9. Tomar una distancia en la carta

Juan quiere saber la distancia entre dos puntos, para ello utiliza la regla disponible (o el compás), 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. En otra ocasión utilizará el compás, en este caso abre el compás hasta que las puntas toquen los dos puntos y después traslada el compás hasta la escala vertical de la carta y calcular las subdivisiones de la escala que abarca esta medida al igual que hizo con la regla.

### 3.10. Marcar extremos de un punto en la carta

Juan sabe calcular la latitud y longitud de la carta utilizando el compás o la regla, pero quiere cerciorarse de que lo ha hecho correctamente y le pide al sistema que muestre los puntos de corte de un punto dibujado en la carta sobre las escalas horizontales y verticales que se encuentran en los bordes de la carta. Una vez verificados le pide al sistema que los oculte de nuevo

### 3.11. Realizar zoom

Juan está realizando un ejercicio, pero no ve con detalle en 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.

## 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 ejercicios.

También se proporciona un fichero css en el que hay definidos dos estilos, uno que permite mostrar un transportador al asignar el estilo a un nodo y otro que permite mostrar una regla. Para representar estas imágenes se utiliza la potencia del css que permite representar el path de un fichero .svg directamente dentro de un css. Hay otras maneras de representar estos objetos en javafx y no es necesario hacer uso de los que aquí presentamos

Ademas se proporciona un proyecto PoiUPV en el que se implementa la capacidad de hacer zoom sobre una imagen que representa un plano. Este proyecto puede servir como proyecto base para desarrollar la aplicación si así lo consideráis conveniente

## 5. Modelo de datos

Para centrarnos en la asignatura, es decir en el diseño de la interfaz y no en el modelo de datos y de la persistencia de ellos, he desarrollado una librería que da soporte al modelo de datos que debéis de utilizar para desarrollar el proyecto.

Así pues, los objetos de tarea que he considerado son: 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 creáis 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 creáis 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. Librería proporcionada para la persistencia de los datos

Pendiente de realizar y publicar

## 7. Ayudas a la programación

Pendiente de publicar

## 8. Instrucciones de Entrega

Todos los grupos tendrán la misma fecha entrega, el 16 de diciembre.

Un único miembro del grupo sube el fichero zip a la tarea correspondiente, incluyendo en el campo de comentarios los nombres de los miembros del grupo.

- Es necesario que indiquéis la uri del repositorio que habéis utilizado y debo de tener permisos para poder acceder a él.

## 9. Evaluación

- Es importante que quede reflejado el proceso de desarrollo, es por ello que debe de quedar constancia de la documentación generada en el Diseño conceptual, Diseño físico, (prototipos)
- Aquellos proyectos que no compilen o que no se muestren la pantalla principal al arrancar se calificarán con un cero.
- Se deberán incluir los diálogos de confirmación, errores, etc. que se considere necesario.



- Hay que utilizar hojas de estilo para dar formato a la aplicación.
- Para evaluar el diseño de la interfaz de la aplicación se tendrán en consideración **los principios, guías de diseño y demás directrices estudiadas en clase de teoría**.
- Debe ser posible redimensionar la pantalla principal, los controles se ajustarán adecuadamente para usar el espacio disponible.
- Se aplicará la Normativa de Integridad Académica de la UPV y la Normativa de Honestidad Académica de la ETSInf. En la asignatura existen herramientas anti plagio.