

Pruebas Software

Desarrollo de Software

Curso 2022-2023

3º Ingeniería Informática GP6



**UNIVERSIDAD
DE GRANADA**

Plantilla para realizar la planificación de las pruebas	3
Visualización de comentarios y gestión de filtros.....	3
Simulación de pesca.....	3
Capacidad de gestión de usuarios.....	4
Capacidad de gestión de puntos.....	4
Eficiencia.....	5
Capacidad de gestión de barcos.....	5
Seguridad en el sistema.....	5
Evolución del sistema.....	6
Ética y compromiso.....	6
Disponibilidad.....	7
Criterio de parada general	7
Análisis de pruebas	8
Diseño de pruebas	10
Patrón de pruebas Page Object.....	12
Resultado de los tests.....	12
ANEXO - Cambios de la práctica anterior	13
Nuevo diagrama de clases.....	13
Informe sobre el trabajo realizado por cada miembro del grupo.....	14
Responsables.....	14
Jose Luis Rico Ramos - Experto.....	14
Miguel Tirado Guzmán - Director de proyecto.....	14
David Martínez Díaz - Director de Pruebas.....	14
Equipos.....	15
Sub-equipo de pruebas:.....	15
Sub-equipo de desarrollo:.....	15
Conclusiones	16

Plantilla para realizar la planificación de las pruebas.

Importante: Para no repetir el mismo texto a lo largo de este apartado a continuación se establecen algunos de los elementos que se quiere en todas las pruebas. Sucede igual con los criterios de parada, definiendo uno general al final de este apartado además de los propios en las tablas.

Para todas las siguientes tablas se validará previamente la concordancia de las necesidades de los interesados en el sistema y las partes/capacidades a probar.

Visualización de comentarios y gestión de filtros

Qué se quiere probar y cómo. Requisitos funcionales, no funcionales, hay que ver que se cumplan todas las necesidades de los interesados, mirando puntos de vista. RF	Capacidad para la visualización correcta y amigable de comentarios y gestión de filtros de comentarios por parte de los usuarios a través de pruebas de unidad así como de visualización de dicha unidad.
Quién lo hará y cuándo	Lo realizará el siguiente miembro del equipo de pruebas : <ul style="list-style-type: none">● Raúl Morgado Saravia● Elena Ortega Contreras● Ruben Rosales Tapia En la clase de aula pequeña el día 25 de abril .
Establecer criterios de finalización de prueba	Además de los criterios genéricos: Cuando se encuentren errores graves (como incorrecta visualización de los comentarios, no aplicación de los filtros...) Ó se compruebe que funciona adecuadamente

Simulación de pesca

Qué se quiere probar y cómo	Capacidad para la simulación de sesiones de pesca mediante creación de pruebas software.
Quién lo hará y cuándo	Lo realizará el siguiente miembro del equipo de pruebas : <ul style="list-style-type: none">● Raúl Morgado Saravia● Elena Ortega Contreras● Ruben Rosales Tapia En la clase de aula pequeña el día 25 de abril .
Establecer criterios de finalización de prueba	Criterios genéricos definidos previamente.

Capacidad de gestión de usuarios

Qué se quiere probar y cómo	Capacidad de gestión de usuarios tanto desde el punto de vista del administrador del sistema como del propio usuario a través de codificación de pruebas así como auditando el código fuente.
Quién lo hará y cuándo	Lo realizará el siguiente miembro del equipo de pruebas : <ul style="list-style-type: none">● Raúl Morgado Saravia● Elena Ortega Contreras● Ruben Rosales Tapia En la clase de aula pequeña el día 25 de abril .
Establecer criterios de finalización de prueba	Criterios genéricos definidos previamente.

Capacidad de gestión de puntos

Qué se quiere probar y cómo	Capacidad de gestión de puntos tanto desde el punto de vista del administrador del sistema como del propio usuario a través de codificación de pruebas así como auditando el código fuente.
Quién lo hará y cuándo	Lo realizará el siguiente miembro del equipo de pruebas : <ul style="list-style-type: none">● Raúl Morgado Saravia● Elena Ortega Contreras● Ruben Rosales Tapia En la clase de aula pequeña el día 25 de abril .
Establecer criterios de finalización de prueba	Además de los criterios genéricos: Cuando se encuentren errores graves como incorrecta entrada de parámetros.

Eficiencia

Qué se quiere probar y cómo	Eficiencia del sistema, sobre todo en operaciones de gran volumen el sistema, como pueden ser: en gestión de BD, comentarios...
Quién lo hará y cuándo	Lo realizará el siguiente miembro del equipo de pruebas : <ul style="list-style-type: none">● Raúl Morgado Saravia● Elena Ortega Contreras● Ruben Rosales Tapia. En la clase de aula pequeña el día 2 de mayo .
Establecer criterios de finalización de prueba	Cuando el sistema resista el 90% de las pruebas de carga realizadas

Capacidad de gestión de barcos

Qué se quiere probar y cómo	Capacidad de gestión y creación de barcos a través de codificación de pruebas así como auditando el código fuente.
Quién lo hará y cuándo	Lo realizará el siguiente miembro del equipo de pruebas : <ul style="list-style-type: none">● Raúl Morgado Saravia● Elena Ortega Contreras● Ruben Rosales Tapia En la clase de aula pequeña el día 2 de mayo
Establecer criterios de finalización de prueba	Cuando se encuentren errores graves como incorrecta entrada de parámetros.

Seguridad en el sistema

Qué se quiere probar y cómo	Capacidad de seguridad del sistema, a través de la auditoración del código fuente y controlando el uso de patrones y estilos arquitectónicos que favorezcan esta capacidad.
Quién lo hará y cuándo	Lo realizará el siguiente miembro del equipo de pruebas : <ul style="list-style-type: none">● Raúl Morgado Saravia● Elena Ortega Contreras● Ruben Rosales Tapia

	En la clase de aula pequeña el día 2 de mayo .
Establecer criterios de finalización de prueba	Cuando se encuentren fallos graves en este ámbito , a través de test de ataques al sistema.

Evolución del sistema

Qué se quiere probar y cómo	Capacidad de evolución del sistema, a través de la auditoración del código fuente y controlando el uso de patrones y estilos arquitectónicos que favorezcan esta capacidad
Quién lo hará y cuándo	Lo realizará el siguiente miembro del equipo de pruebas : <ul style="list-style-type: none"> ● Raúl Morgado Saravia ● Elena Ortega Contreras ● Ruben Rosales Tapia En la clase de aula pequeña el día 9 de mayo
Establecer criterios de finalización de prueba	Cuando se detecte un mal uso de estilos arquitectónicos / patrones.

Ética y compromiso

Qué se quiere probar y cómo	Validación de la ética y compromiso del sistema así como la escalabilidad, observando las partes visibles comprobando que los productos cumplen de manera responsable y segura.
Quién lo hará y cuándo	Lo realizará el siguiente miembro del equipo de pruebas : <ul style="list-style-type: none"> ● Raúl Morgado Saravia ● Elena Ortega Contreras ● Ruben Rosales Tapia En la clase de aula pequeña el día 9 de mayo .
Establecer criterios de finalización de prueba	Si se determina algún fallo en la satisfacción de esta propiedad.

Disponibilidad

Qué se quiere probar y cómo	Validación de la capacidad de Disponibilidad del sistema a través de distintas pruebas de resistencia
Quién lo hará y cuándo	Lo realizará el siguiente miembro del equipo de pruebas : <ul style="list-style-type: none">● Raúl Morgado Saravia● Elena Ortega Contreras● Ruben Rosales Tapia En la clase de aula pequeña el día 9 de mayo .
Establecer criterios de finalización de prueba	Cuando se logre una disponibilidad del 90% que se estuvo monitoreando.

Criterio de parada general

Para todas aquellas pruebas codificadas: Cuando se establezca una relación de cantidad de pruebas y errores encontrados suficientemente razonables para llegar a la conclusión de que no existen más errores. Si se encuentran errores, cuando se observe un error persistente o un error suficientemente grave como para perjudicar determinadamente el buen funcionamiento del sistema.

Análisis de pruebas

Nota: Cuando no se cumple alguna condición al intentar hacer efectiva una operación, se debe mostrar un mensaje de error que informe de ello.

Elemento a probar	Condiciones	Datos requeridos
Dar de alta usuario	<ul style="list-style-type: none"> Comprobar que el usuario no existe ya. Comprobar que todos los campos estén rellenos. Comprobar que el correo no esté en uso. Comprobar que el nick no esté en uso. Contraseña ≥ 8 caracteres. Salida: Confirmación de los cambios realizados. 	Correo electrónico, nombre, contraseña, nick.
Gestión de cuenta	<ul style="list-style-type: none"> Acceso identificado al proceso. Comprobación del dato (o datos) modificados. Salida: Confirmación de los cambios realizados. 	Identificador de cuenta y contraseña, correo electrónico, nick.
Añadir comentario	<ul style="list-style-type: none"> Acceso identificado al proceso. Comprobación de longitud del comentario. Post-procesamiento del comentario (aplicar filtro) Salida: Confirmación de los cambios realizados. 	Cadena de texto.
Crear barco	<ul style="list-style-type: none"> Acceso identificado al proceso. Comprobar que sea correcto el tipo de barco. Comprobar que el usuario dispone de puntos suficientes para construir dicho barco. Pedir confirmación de la acción antes de hacerse efectiva la creación. Salida: Confirmación de los cambios realizados. 	Tipo de barco.
Gestión de barco	<ul style="list-style-type: none"> Acceso identificado al proceso. Comprobar que el usuario dispone de puntos suficientes para realizar dicho cambio. Pedir confirmación de la acción antes de hacerse efectivo el cambio. Salida: Confirmación de los cambios realizados. 	Identificador de barco, tipo de barco.
Gestión de filtro	<ul style="list-style-type: none"> Acceso identificado al proceso como gestor de BD. Si se va a añadir o modificar un filtro, 	Palabras que se desea filtrar.

	comprobar que los datos introducidos sean correctos. <ul style="list-style-type: none"> • Salida: Confirmación de los cambios realizados. 	
Gestión de peces	<ul style="list-style-type: none"> • Acceso identificado al proceso como gestor de BD. • Si se va a añadir o modificar un pez, comprobar que los datos introducidos sean correctos. • Pedir confirmación de la acción antes de hacerse efectivo el cambio. • Salida: Confirmación de los cambios realizados. 	Tipo de pez.
Gestión de puntos	<ul style="list-style-type: none"> • Acceso identificado al proceso como gestor de BD. • Si se van a añadir o modificar puntos a un objeto, comprobar que sean de tipo entero con valor positivo. • Salida: Confirmación de los cambios realizados. 	Número de puntos.
Pescar	<ul style="list-style-type: none"> • Acceso identificado al proceso como gestor BD. • Usuario tenga un barco para pescar. 	Barco.
Ética y compromiso	<ul style="list-style-type: none"> • Usuario sea mayor de edad 	Edad del usuario
Seguridad	<ul style="list-style-type: none"> • Encapsular los datos de todas las clases del modelo. 	N/A
Acceso identificado	<ul style="list-style-type: none"> • Comprobar que el nombre o el correo de usuario/gestor existe. • Comprobar que la contraseña es correcta. 	Nombre o email, contraseña.
Eficiencia	<ul style="list-style-type: none"> • Comprobar que los fragmentos de código no sean susceptibles de refactorización. 	Código.

Diseño de pruebas

Casos de pruebas (ordenados por prioridad)	Entornos de prueba	Datos de Prueba	Relación con las condiciones de prueba
Disponibilidad	Disponibilidad de monitorización	Rendimiento del servidor	Necesario para acceder a la BD y prestar servicio
(Dar de alta) Comprobar usuario = Comprobar correo electrónico + Comprobar contraseña + Comprobar edad + Comprobar nombre	Aplicación (entorno de desarrollo)	Correo electrónico, edad del usuario, nombre, contraseña	Comprobación de un usuario por sus atributos se necesita: <ol style="list-style-type: none"> 1. Comprobar si el dominio de correo es válido. 2. Comprobar contraseña sea \geq 8 caracteres 3. Comprobar su edad sea \geq 18 4. Comprobar que el nombre de usuario no esté en uso. <p>Todos son necesarias se haga a la vez para comprobar al usuario.</p>
Comprobar cambios cuenta (Gestión cuenta) = Comprobar datos + Comprobar id	Aplicación	Identificador de cuenta y datos a modificar.	Para gestionar la cuenta es necesario: 1. Comprobar el id de la cuenta. 2. Comprobar que los datos estén en el formato correcto.
(Crear barco) Comprobar creacion = Comprobar puntos + Comprobar tipo barco + comprobar confirmación	Aplicación	Tipo barco	Comprar un barco con puntos y acto seguido pedir una confirmación. <ol style="list-style-type: none"> 1. Comprobar puntos \geq precio barco. 2. Comprobar qué tipo de barco existe. 3. Comprobar confirmación satisfactoria.
Pescar = Comprobar barco	Aplicación	Tipo barco	Pescar requiere que se especifique el barco a pescar, para ello se debe comprobar barco (que exista)

(Añadir comentario) Comprobar comentario = Comprobar texto + Comprobar filtros	Aplicación	Cadena de texto	Añadir un comentario debe comprobarse antes que el texto no esté vacío y a su vez los filtros para aplicarlos.
(Gestión de filtros) Comprobar filtro = Comprobar texto	Aplicación	Cadena de texto	
(Gestión de puntos) Comprobar puntos = Comprobar valor + Comprobar tipo dato	Aplicación	Número	Se tiene que comprobar que los valores sean positivos y que el tipo de dato sea correcto.
(Gestión barcos) Comprobar barco modificar = Comprobar barco + Comprobar puntos + Comprobar confirmación	Aplicación	Tipo barco, número	Gestionar el barco (mejorar) requiere que el barco exista, puntuación sea \geq al límite establecido y una confirmación por parte del usuario.
(Gestión peces) Comprobar peces = Comprobar datos + Comprobar tipo pez	Aplicación	Tipo pez	Comprobar los peces debe de comprobarse que los datos administrados sean correctos los tipos y que el tipo de pez exista.
Seguridad	Aplicación	N/A	Mandatorio para la gestión de puntos y útil para el resto del sistema, logrado a través de la encapsulación.
Eficiencia	Aplicación	Código a comparar.	De gran utilidad para una experiencia óptima.
Evolutivo	Visual Paradigm y herramientas de diagramas	Estructura de la aplicación.	Un sistema evolutivo es necesario en este tipo de aplicaciones dónde la experiencia del usuario puede variar.

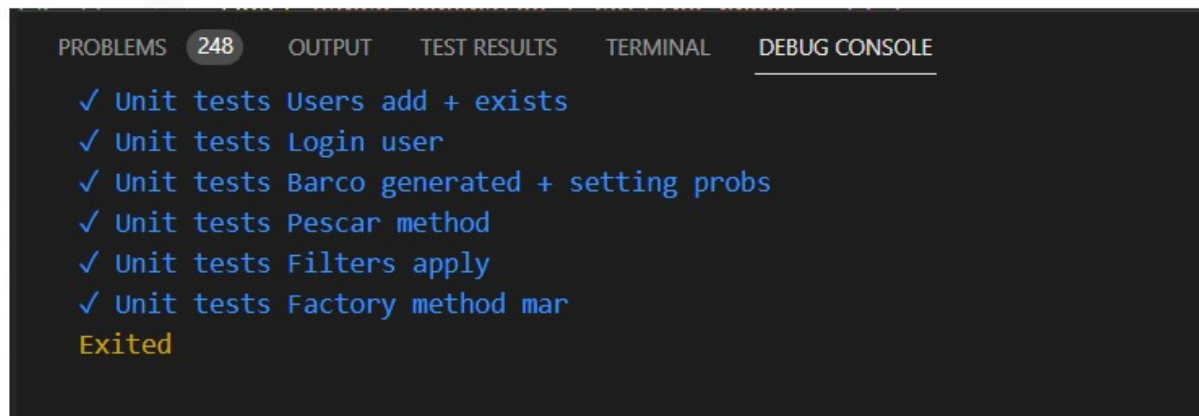
Patrón de pruebas Page Object

Para nuestro proyecto el patrón de diseño será el patrón Page Object.

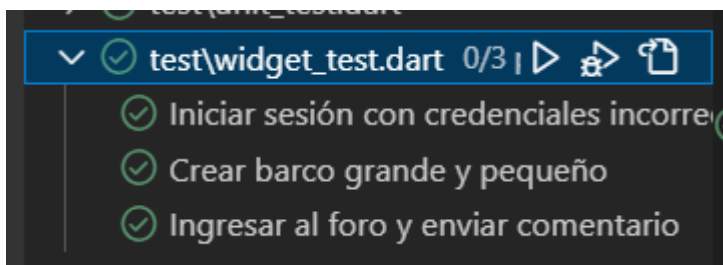
- Este patrón sirve para evitar el código duplicado y el mantenimiento de pruebas por la separación del comportamiento de la página/widget de su implementación enfocándose sólo en el comportamiento.
- Ideal para este proyecto debido a que soporta varias interfaces, en nuestro caso la de la aplicación y la de la web (esta última por confirmar su desarrollo).

Como ventaja de este patrón podemos destacar que cualquier cambio que se produzca en la interfaz de usuario sólo va a afectar al Page Object pero no a los test que ya están implementados.

Resultado de los tests



```
PROBLEMS 248 OUTPUT TEST RESULTS TERMINAL DEBUG CONSOLE
✓ Unit tests Users add + exists
✓ Unit tests Login user
✓ Unit tests Barco generated + setting probs
✓ Unit tests Pescar method
✓ Unit tests Filters apply
✓ Unit tests Factory method mar
Exited
```

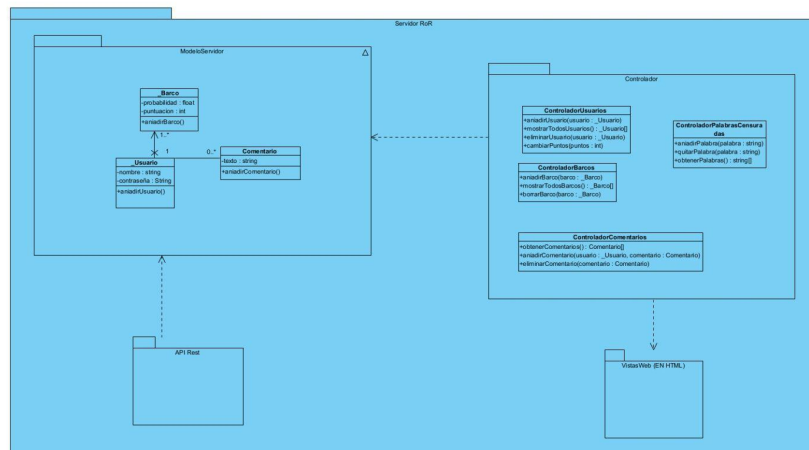
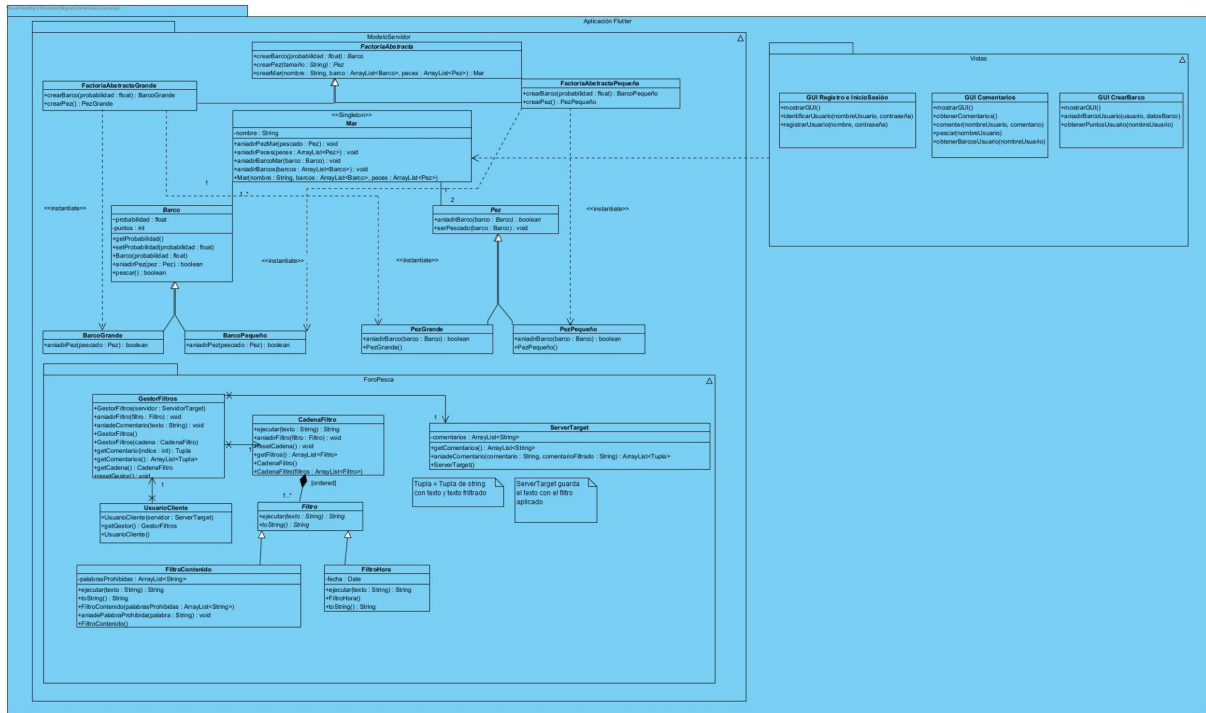


```
test\widget_test.dart 0/3 | ▶ ⏏
  ✓ Iniciar sesión con credenciales incorre
  ✓ Crear barco grande y pequeño
  ✓ Ingresar al foro y enviar comentario
```

ANEXO - Cambios de la práctica anterior

- Ahora cada barco tiene su propia puntuación
- El mar pasa a tener simplemente 2 peces (uno de cada tipo) en lugar de un array de peces
- Ya no existe relación entre peces y barcos

Nuevo diagrama de clases



Informe sobre el trabajo realizado por cada miembro del grupo

Responsables

Las tareas de los responsables se han repartido conforme se especifica en el guión de prácticas, aunque dejando libertad para apoyarnos en la realización de las tareas. De esta forma, si alguno de los responsables tenía dudas sobre algún concepto o quedaba atrasado en sus tareas por ser de mayor complejidad, los demás responsables le ayudamos para mantener todas las tareas al mismo ritmo.

Jose Luis Rico Ramos - Experto

Tal y como especifica el guión, se ha encargado de dar la explicación del temario en la primera sesión de aula pequeña, siendo esta de gran importancia y ayuda tanto para el equipo de desarrollo como para el de pruebas además de para el director de proyecto y el director de pruebas.

Miguel Tirado Guzmán - Director de proyecto

Se ha encargado de la redacción del presente documento (a excepción de las partes de análisis, diseño de pruebas y aplicación del patrón desarrollados por el equipo de pruebas). Además, apoya en la resolución de dudas de los equipos de desarrollo en conceptos de la aplicación así como en las dudas sobre ciertos conceptos y tablas del equipo de pruebas.

David Martínez Díaz - Director de Pruebas

Se encarga de la planificación de las pruebas y de resolver casi todas las dudas que surgen con flutter al conocerlo muy bien. Además, también apoya en la resolución de dudas de ambos equipos.

Equipos

Para esta práctica hemos dividido al equipo en 2 grupos, el equipo de pruebas y el equipo de desarrollo. El equipo de desarrollo se encarga de hacer el código de la práctica, en función a lo decidido en la práctica anterior en función de los distintos requisitos funcionales y no funcionales, mientras que el equipo de pruebas se encargaba de realizar las pruebas a los distintos bloques de código.

Sub-equipo de pruebas:

{ Raúl Morgado Saravia - Elena Ortega Contreras - Ruben Rosales Tapia }

Ellos han sido el equipo encargado de realizar el análisis de pruebas, seleccionando los elementos para probar lo que se han descrito en los puntos de vista anteriormente descritos y de las perspectivas del DA que se han especificado en la práctica 3, donde se han identificado las condiciones de prueba para nuestro proyecto de la “Red Social de Barcos”, y datos requeridos para las pruebas.

Por otro lado, también se han encargado de realizar los análisis correspondientes, para cada tipo de prueba agrupando los requisitos a probar y priorizándolos.

Por último, han utilizado el patrón de pruebas Page Object, bastante eficiente para la lógica de la interfaz de usuario de la aplicación, que facilita el mantenimiento de las pruebas y permite que los cambios en la interfaz de usuario no afecten a las pruebas.

Sub-equipo de desarrollo:

{ Diego Velázquez Ortuño - Gador Romero Prieto - Sergio Muñoz Gomez }

Ellos han sido el equipo encargado del desarrollo del código de la aplicación flutter en función de lo requerido, tal y como se especificaba en los criterios de requisitos funcionales y no funcionales descritos en la anterior práctica.

En este caso contamos con una aplicación flutter mezclando 2 de las aplicaciones desarrolladas en la práctica 2:

- Sistema de gestión de barcos (astillero)
- Filtros de intercepción

De esta forma contaremos con una aplicación con un foro mientras que contamos también con las funcionalidades de la otra aplicación, con una futura implementación de servicio de registros de usuarios donde estos pueden pescar, ganar puntos y comprar más barcos además de las interacciones en el foro anteriormente mencionado.

Conclusiones

La realización de esta práctica se ha llevado a cabo intentando seguir el ritmo establecido en el guión de prácticas, y colaborando entre todos los miembros del equipo debatiendo sobre los diferentes aspectos tanto de los tests (desde el cómo rellenar las tablas hasta los criterios de parada, el qué se quería probar...) como de la propia aplicación, sobre el uso/no uso de ciertos patrones, cambios en el código...

Consideramos por tanto que hemos adquirido los conocimientos necesarios establecidos en el temario y se ha logrado aplicar de forma correcta en la práctica.