



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de una App Android para facilitar la
comunicación entre organizaciones y estudiantes

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Valls Llopis, Luis

Tutor/a: Pelechano Ferragud, Vicente

Cotutor/a externo: PATELLA, MARCO

CURSO ACADÉMICO: 2021/2022

Resumen

El proyecto busca desarrollar una aplicación completamente funcional que sea de gran ayuda tanto para los estudiantes erasmus a la llegada a sus destinos, como para las diferentes organizaciones internacionales para comunicar y estar al tanto de eventos, viajes y quedadas. Para lograr esto la aplicación contará con distintas pantallas, unas “oficiales” donde las organizaciones publiquen sus actividades y otras de “la comunidad” donde los propios estudiantes podrán publicar eventos, consejos, anuncios, etc. Así mismo la App incluirá apartados típicos de una App de turismo para conocer la ciudad de destino (mapa, ubicaciones de interés, guías, etc.).

El proyecto se desarrollará aplicando la metodología ágil, obteniendo una versión ejecutable al final de cada sprint. La tecnología elegida será la de Kotlin junto a JetpackCompose como lenguaje de programación y Android Studio como IDE. A su vez se hará uso de la plataforma ServerLess de Google Firebase para proporcionar el servicio en la nube.

Palabras clave: Kotlin, JetpackCompose, Firebase, Android, Ágil.



Abstract

The project seeks to develop a fully functional application that is of great help both for Erasmus students upon arrival at their destinations, and for different international organizations to communicate and keep up to date with events, trips and meetups. To achieve this, the application will have different screens, some “official” ones where the organizations publish their activities and others from “the community” where the students themselves will be able to publish events, tips, announcements, etc. Likewise, the App will include typical sections of a tourism app to get to know the destination city (map, locations of interest, guides, etc.).

The project will be developed applying the agile methodology, obtaining an executable version at the end of each sprint. The chosen technology will be Kotlin together with JetpackCompose as programming language and Android Studio as Ide. In turn, the ServerLess platform of Google Firebase will be used to provide the service in the cloud.

Keywords: Kotlin, JetpackCompose, Firebase, Android, Agile.

Tabla de contenidos

1. Introducción	10
1.1. Motivación	10
1.2. Objetivos.....	11
1.3. Impacto Esperado	11
1.4. Metodología.....	11
2. Estado del arte.....	14
2.1 Aplicaciones similares	14
2.1.1 Páginas propias de las organizaciones.....	14
2.1.2 Aplicaciones móviles de eventos	16
2.1.3 Aplicaciones erasmus específicas.....	17
2.1.4 Conclusiones	18
3. Estudio del problema	19
3.1. Solución Actual.....	19
3.2. Estudio de viabilidad	19
3.2.1. Viabilidad temporal	19
3.2.2. Viabilidad técnica / operativa	20
3.3. Modelo de negocio [13].....	21
3.3.1. Modelo Premium:.....	21
3.3.2. Modelo Freemium:.....	21
3.3.3. Modelo Mixto:	22
3.3.4. Conclusiones y modelo elegido:	22
3.3 Conclusiones	22
4. Diseño de la solución.....	23
4.1. Matriz Dafo.....	23
4.2. Lean Canvas:.....	25
4.3. Mapa de características:	25
4.4. Modelo de dominio	26
4.5. Mock ups.....	27
.....	27
.....	28
4.7. Gestión de alcance del trabajo	31
4.8. Patrones de diseño.....	31



4.8.1. Patrón Singleton	32
4.8.2. Patrón Inyección de dependencias [16]	32
5. Desarrollo de la solución propuesta.....	33
5.1. Sprint 0	33
5.1.1. Estimación.....	33
5.1.2. Realización.....	34
5.1.3. Finalización.....	34
5.2. Sprint 1	35
5.2.1. Estimación	35
5.2.2. Realización	36
5.2.3. Finalización	37
5.3. Sprint 2.....	39
5.3.1. Estimación	39
5.3.2. Realización	40
5.3.3. Finalización	41
5.4. Sprint 3.....	43
5.4.1. Estimación	43
5.4.2. Realización	43
5.4.3. Finalización	44
5.5. Sprint 4.....	46
5.5.1. Estimación	46
5.5.2. Realización	47
5.5.3. Finalización	48
6. Implantación	50
6.1. Versión de lanzamiento Erasmus Events.....	50
6.2. Lanzamiento en la Play Store	55
7. Pruebas.....	57
7.1. Pruebas unitarias.....	57
7.2 Pruebas de integración.....	57
7.3 Pruebas funcionales	57
7.4 Pruebas integrales.....	58
7.5. Pruebas de aceptación.....	58
7.6. Pruebas de rendimiento.....	59
7.6.1 Prueba rendimiento Pixel 4:	60
7.6.2 Prueba rendimiento Nexus 5:.....	60
7.7. Pruebas de humo.....	61

7.9. Pruebas con usuarios	61
7.9.1. Etapa interna	61
7.9.2. Etapa abierta	62
7.8. Automatización de las pruebas	64
8. Conclusiones.....	65
9. Trabajos futuros	66
9.1. Versión IOS	66
9.2. Nuevas funcionalidades	67
9.3. Automatización de tareas.	67
10. Referencias	68



Tabla de figuras

Ilustración 1 - Página web versión escritorio ESN	15
Ilustración 2 - Página web versión móvil ESN	15
Ilustración 3 Apps de eventos en PlayStore	16
Ilustración 4 - Capturas pantalla App eventos EventBrite.....	17
Ilustración 5 - Capturas pantalla App eventos Fever	17
Ilustración 6 - Capturas pantalla App erasmus, Erasmus in Czech Republic.....	18
Ilustración 7 - Capturas pantalla App erasmus, Erasmus Barcelona.....	18
Ilustración 8 - Porcentaje de sistemas operativos en smartphones [8]	20
Ilustración 9 - Matriz DAFO.....	23
Ilustración 10 - Tabla Lean Canvas	25
Ilustración 11 - Mapa de características	26
Ilustración 12 - Modelo de dominio	26
Ilustración 13 - MockUp Ventana Explorar.....	27
Ilustración 14 - MockUp ventana detalles evento	28
Ilustración 15 - MockUp ventana eventos guardados	28
Ilustración 16 - MockUp ventana eventos pagados.....	29
Ilustración 17 - MockUp ventana inicio sesión	29
Ilustración 18 - MockUp ventana publicar evento	30
Ilustración 19 - MockUp ventana perfil.....	30
Ilustración 20 - MockUp Ventana Filtros	31
Ilustración 21 - UTs incluidas Sprint 0	33
Ilustración 22 - Gráfica Burndown (en puntos) Sprint 0.....	34
Ilustración 23 - Gráfica Burndown (en horas) Sprint 0.....	34
Ilustración 24 - Detalles Sprint 0	34
Ilustración 25 - Gráfica diferencia tiempo estimado/real Sprint 0.....	35
Ilustración 26 - UTs incluidas Sprint 1.....	35
Ilustración 27 - Gráfica Burndown (en puntos) Sprint 1.....	36
Ilustración 28 - Gráfica Burndown (en horas) Sprint 1	36
Ilustración 29 - Detalles Sprint 1.....	37
Ilustración 30 - Gráfica Esfuerzo en UT por actividad Sprint 1	38
Ilustración 31 - Gráfica diferencia horas estimadas/real Sprint 1	38
Ilustración 32 - UTs incluidas Sprint 2	39
Ilustración 33 - Gráfica Burndown (en puntos) Sprint 2	40
Ilustración 34 - Gráfica Burndown (en horas) Sprint 2	40
Ilustración 35 - Detalles Sprint 2	41
Ilustración 36 - Gráfica Esfuerzo en UT por actividad Sprint 2.....	41
Ilustración 37 - Gráfica tiempo estimado/real Sprint 2	42
Ilustración 38 - UTs incluidas Sprint 3	43
Ilustración 39 - Burndown (en horas) Sprint 3.....	43
Ilustración 40 - Burndown (en puntos) Sprint 3	44
Ilustración 41 - Detalles Sprint 3.....	44
Ilustración 42 - Gráfica Esfuerzo en UT por actividad Sprint 3.....	45
Ilustración 43 - Gráfica tiempo estimado/real Sprint 3.....	45
Ilustración 44 - UTs incluidas Sprint 4	46

Ilustración 45 - Gráfica Burndown (en puntos) Sprint 4	47
Ilustración 46 - Gráfica Burndown (en horas) Sprint 4	47
Ilustración 47 - Detalles Sprint 4	48
Ilustración 48 - Gráfica Esfuerzo en UT por actividad Sprint 4	48
Ilustración 49 - Gráfica tiempo estimado/real Sprint 4.....	49
Ilustración 50 - Captura ventana de inicio.....	50
Ilustración 51 - Captura ventana explorar . Ilustración 52 - Captura ventana explorar al buscar	51
Ilustración 53 - Captura categoría "Party" de eventos	52
Ilustración 54 - Captura ventana mapa.....	52
Ilustración 55 - captura ventana eventos guardados	53
Ilustración 56 - Captura ventana publicar evento con foto.....	53
Ilustración 57 - Captura ventana publicar evento, selector de fecha	53
Ilustración 58 - Captura ventana publicar evento.....	53
Ilustración 59 - Captura ventana editar perfil al modificar el avatar.....	54
Ilustración 60 - Captura ventana editar perfil	54
Ilustración 61 - Captura ventana Perfil	54
Ilustración 63 - Captura ventana editar evento	55
Ilustración 62 - Captura ventana editar evento al eliminar un evento	55
Ilustración 64 - Captura ventana mis eventos	55
Ilustración 65 - Ejemplos Pruebas Aceptación	59
Ilustración 66 - Resultado prueba rendimiento Pixel 4.....	60
Ilustración 67 - Prueba rendimiento Nexus 5	60
Ilustración 68 - Respuesta encuesta usuarios – Rapidez.....	62
Ilustración 69 - Respuesta encuesta usuarios - Facilidad de uso.....	62
Ilustración 70 - Respuesta encuesta usuarios - Interfaz Gráfica	63
Ilustración 71 - Respuesta encuesta usuarios - Satisfacción	63
Ilustración 72 - Respuesta encuesta usuarios - Opinión general.....	63
Ilustración 73 - Gráfica porcentaje smartphones con IOS por países.....	66
Ilustración 74 - UTs restantes en Backlog.....	67



1. Introducción

Imagine el inicio de una experiencia erasmus. Llega a un país nuevo, con un idioma que seguramente no acabe de entender, una ciudad que no conoce y un curso por empezar. Las primeras semanas y meses le bombardean constantemente con información, sitios que visitar, actividades que realizar, viajes a los que registrarse...

Todo esto le crea una sensación de no saber muy bien que está haciendo, y sobre todo ese sentimiento de que está olvidando y dejando por el camino muchas cosas importantes.

Aquí es donde entra este proyecto, trabajo que busca cumplir 3 objetivos.

1. Ofrecer una única App, donde los estudiantes erasmus puedan encontrar recogida y organizada toda la información que necesiten durante su estancia. Así como una herramienta para conocer gente y poder disfrutar al máximo de la experiencia.
2. Demostrar que como graduado en ingeniería informática, se han adquirido todos los conocimientos necesarios para desarrollar un proyecto para “el mundo real”, obteniendo como resultado no un entregable más, si no un software listo para ser lanzado y comercializado.
3. Obtener conocimiento en tecnologías de vanguardia. Este es el motivo por el que la App ha sido desarrollada utilizando Kotlin y JetpackCompose, a pesar de que este lenguaje apenas es usado por grandes empresas debido a que es “muy nuevo”. No obstante, al trabajar e indagar en él estamos evitando un problema muy común en los graduados en informática y es el de empezar tu carrera laboral “ya obsoletos”.

1.1. Motivación

La motivación por este proyecto surge en distintas etapas de mi vida estudiantil. La primera, cumpliendo con el segundo y el tercer objetivo fue la de decidir que quería realizar un proyecto completo. Un proyecto que incluyera desde la fase de planificación hasta la de lanzamiento, para poder demostrar que como graduado era capaz de desarrollar un producto de calidad y a la vez crear un perfil profesional de cara al futuro, especializándome en tecnologías modernas y que seguramente dentro de unos años tendrán un peso mucho mayor que el actual. Esto se corrobora observando la tendencia al alza año tras año del mercado de dispositivos móviles y de software para los mismos.

La segunda fase, la de porque realizar una aplicación para estudiantes erasmus, surge por la propia experiencia personal. Durante el segundo semestre del curso 2021-2022 he realizado mi estancia erasmus en Bolonia, Italia. Donde, desde el primer día me sentí abrumado en cuanto a las actividades ofertadas. Solo en esta ciudad ya compiten 2 asociaciones erasmus con los propios eventos de la universidad, por lo que cada semana era muy tedioso descubrir que realizaba que organización, donde, el precio, la disponibilidad, los horarios... Y no se trata de un caso aislado, la gran mayoría de

estudiantes tienen el mismo sentimiento. Por lo que la aplicación viene a remediar esto, viene a solucionar un problema existente y que se repite año tras año.

1.2. Objetivos

Como se ha comentado en la introducción, este proyecto busca cumplir 3 objetivos, el de ofrecer una App que sea beneficiosa para los estudiantes erasmus, desarrollar un producto de calidad que demuestren todas las habilidades adquiridas durante el estudio del grado en Ingeniería informática y adquirir conocimientos y experiencia en nuevas tecnologías.

Partiendo de estos objetivos, consideraremos que la realización de este trabajo ha sido un éxito cuando se logre lanzar al mercado la aplicación terminada y esta consiga el recibimiento esperado.

Se utilizarán como métricas el número de descargas que se obtengan durante los primeros meses del lanzamiento, el número de eventos publicados por los usuarios, la retención de usuarios y el número de llamadas al servidor.

Analizando estos resultados podremos determinar si los objetivos han sido cumplidos o no.

1.3. Impacto Esperado

Se espera que, con el lanzamiento de esta aplicación, miles de estudiantes la utilicen a lo largo de su experiencia Erasmus a la vez que facilite el trabajo a las organizaciones para llegar a su público objetivo. Por lo que el impacto esperado sería el que gran parte de los estudiantes contarán con la aplicación instalada en sus dispositivos y la utilizarán de forma constante creando una comunidad viva y creciente.

1.4. Metodología

La metodología utilizada para la realización de este proyecto ha sido la Metodología Ágil [1].

¿En qué consiste la metodología Ágil? ¿Qué es ser Ágil?

A diferencia de otras metodologías tradicionales, en el desarrollo ágil de software se tiene un enfoque más moderno y dinámico, donde nuestra capacidad de trabajo se mide en personas no recursos. Un equipo ágil debe ser autoorganizado, no existe una figura de “jefe” que organice el trabajo, si no que los miembros del equipo son quienes reparten las tareas; cross-functional, es decir, la fuerza de un equipo es la suma de las habilidades personales de sus integrantes; no contar con una figura de jefe, si no de líder, un miembro que sea un facilitador o que se encargue de las comunicaciones externas; debe existir una estrecha comunicación y ser un equipo constante y pequeño.

Estas serían a grandes rasgos algunas cualidades de la metodología ágil, pero centrándonos en el desarrollo, la principal diferencia respecto a otras metodologías es el



enfoque en la toma de decisiones. Siempre se busca un desarrollo iterativo e incremental, por lo que los requisitos evolucionan y cambian según el proyecto avanza, adaptándose a las necesidades de este. Esto se aleja bastante de metodologías tradicionales como podría ser la “waterfall” (En cascada) donde el desarrollo se lleva a cabo siguiendo etapas siguiendo un riguroso orden y donde los requisitos y las especificaciones iniciales no están predispuestas para cambiarse.

Como se ha dicho anteriormente, el desarrollo ágil busca la flexibilidad y la agilidad. Gracias a esto se ha convertido en la metodología más utilizada hoy en día, siendo sus principales variantes las siguientes:

-Kanban: metodología de trabajo caracterizada en dividir el trabajo en porciones (unidades de trabajo) y organizarlas en un tablero. Este tablero a su vez suele dividirse en tareas pendientes, en curso y finalizadas. Pudiendo consultar en todo momento de forma muy visual el flujo de trabajo.

-Scrum: muy similar a Kanban, en cuanto a la forma de dividir los requisitos y tareas pequeñas tareas. Estas tareas se dividen en bloques cortos, usualmente entre 2 y 4 semanas, para conseguir una versión completa al final de cada iteración. Cada uno de estos bloques suele denominarse Sprint y sus etapas son: planificación, ejecución, reunión diaria y demostración de resultados. En Scrum solo encontramos tres roles dentro de un equipo siendo estos el Product Owner (representa la parte cliente), Scrum Máster (nexo entre el product owner y el equipo de desarrollo) y los desarrolladores (representa el equipo técnico que realiza el trabajo).

-Lean: pensada para equipos de desarrollo pequeños y tareas de corta duración.

-Programación extrema (XP): metodología que considera las relaciones interpersonales como la clave del éxito. Se centra en conseguir un buen ambiente de trabajo y un feedback constante del cliente. Sus 12 pilares son: diseño sencillo, testing, refactorización y programación con estándares, propiedad colectiva del código, programación en parejas, integración continua, entregas semanales, cliente in situ, entregas frecuentes y planificación.

En cada contexto de trabajo se pueden encontrar diferentes metodologías, es importante encontrar la que más se adapte al nuestro. Después de analizar las distintas opciones y estudiar su funcionamiento, para la realización de este proyecto se decidió optar por la metodología **Tune-Up** [2].

¿En qué consiste la metodología **Tune-UP**?

Tune-up ha sido desarrollada por el grupo ISSI (Ingeniería de Software y Sistemas de Información) del departamento DSIC de la UPV con el objetivo de obtener una metodología que incorpore los principales elementos de las anteriormente nombradas. Tune-up ofrece la posibilidad de trabajar siguiendo Scrum, Kanban, Lean o Extreme Programming a la vez que muchas características asociadas a la gestión de proyectos no ofrecidas por las metodologías y herramientas ágiles, lo que permite complementar la forma de trabajo.

La configuración de la metodología Tune-UP utilizada durante el desarrollo del proyecto nos ofrece los siguientes elementos:

-Línea de trabajo. Durante las primeras fases del proyecto se pensó en trabajar siguiendo dos líneas de trabajo, una que siguiera el desarrollo de la app para Android y otra siguiendo el desarrollo para IOS. Esta idea fue rápidamente rechazada debido a la falta de tiempo ya que optar por esta opción hubiera aumentado considerablemente el alcance del proyecto dejándolo fuera de los recursos reservados a la realización del TFG. Por lo que finalmente solo existe una Línea de trabajo asociada al desarrollo de ErasmusEvents.

-Colaboradores. Al tratarse de un trabajo de fin de grado individual, el equipo de trabajo está formado por una única persona, por lo que los distintos roles recaen sobre ella, cumpliendo a la vez de Desarrollador, Product Owner y Scrum Master.

-Unidades de Trabajo (UT). Ítems en los que se divide el trabajo. Estos elementos son los que deben ejecutar los desarrolladores en el marco de una Línea de Trabajo. Durante la realización del proyecto encontraremos de 3 tipos:

- Nuevo requisito: un nuevo requisito a ser implementado.
- Corrección de fallo: corrección de algún fallo detectado en algún requisito ya implementado. Clasificados en función de su nivel de severidad (Leve, Moderado, Severo, Crítico).
- Otras tareas: tareas no relacionadas con la programación del producto, sino más bien con la planificación de este.

-Backlog. Se trata del “contenedor” donde todo el trabajo no asignado queda “almacenado”. Salvo excepciones por urgencia, todas las unidades de trabajo se crean en el Backlog y son movidas a los distintos Sprints conforme sean asignadas.

-Sprint. El termino Sprint hace referencia a una iteración del trabajo, es decir, un “contenedor” de UTs asociado a un periodo de tiempo. Durante la realización del proyecto se ha optado por Sprints de 4 semanas de duración.

-Workflows. Cada tipo de UT tiene asociado un Workflow, que nos indica el proceso para abordarla, es decir, las diferentes fases o tareas que hay que realizar para finalizar dicha UT. Durante el desarrollo se han utilizado los siguientes 3 Workflows:

- WF Desarrollo. Para UT cuya realización implica cambios en el comportamiento del producto.
- WF Refactoring. Para UT cuya realización no implica cambios en el comportamiento del producto, pero si en su estructura interna.
- WF Tarea. Para UT cuya realización no implica cambios en el comportamiento del producto ni en su estructura interna.

Otro elemento de gran relevancia durante la realización del proyecto ha sido la **gestión de alcance del trabajo**, elemento muy importante ya que nos va a marcar como se configuren los distintos Sprints. Para gestionar el alcance de cada Sprint o del Proyecto se han utilizado estimaciones. Primero estimando en Horas Ideales de Programación



(HIP), equivalentes a las horas ininterrumpidas que se estiman necesarias para realizar la programación de una UT con Workflow de Desarrollo o Refactoring, junto a una estimación por Puntos que nos indicará el peso orientativo de cada UT.

2. Estado del arte

Existen cientos de aplicaciones en el mercado pensadas en la gestión de eventos. Tan extendido es este mercado que incluso existe una categoría dentro de la PlayStore exclusivamente para aplicaciones de eventos. Sin embargo, muchas de estas aplicaciones o bien cubren un sector muy amplio y genérico, por lo que cuesta encontrar lo que deseas, o bien son muy concretas y específicas (creadas para un único evento o una categoría en especial, por ejemplo, conciertos). Por lo que ErasmusEvents viene a dar solución a un sector muy específico, el de los estudiantes Erasmus.

2.1 Aplicaciones similares

Encontramos aplicaciones que pueden ofrecer servicios similares, como son el ejemplo de las páginas propias de las distintas organizaciones de estudiantes erasmus o algunas específicas para cada destino, como por ejemplo Erasmus Barcelona, pensada para esta ciudad. Sin embargo, tras una exhaustiva búsqueda, no se ha encontrado una aplicación que ofrezca lo mismo que la desarrollada en este proyecto.

2.1.1 Páginas propias de las organizaciones

El principal catálogo de actividades para los erasmus lo encontramos en las páginas webs propias de las organizaciones de estudiantes. El principal problema de estas es su interfaz gráfica, muy anticuada y tosca de utilizar. Además de que solo cuentan con una versión web que no se adapta correctamente a los dispositivos móviles. Vemos el caso de la página de la ESN de Bolonia [3].



Ilustración 2 - Página web versión móvil ESN

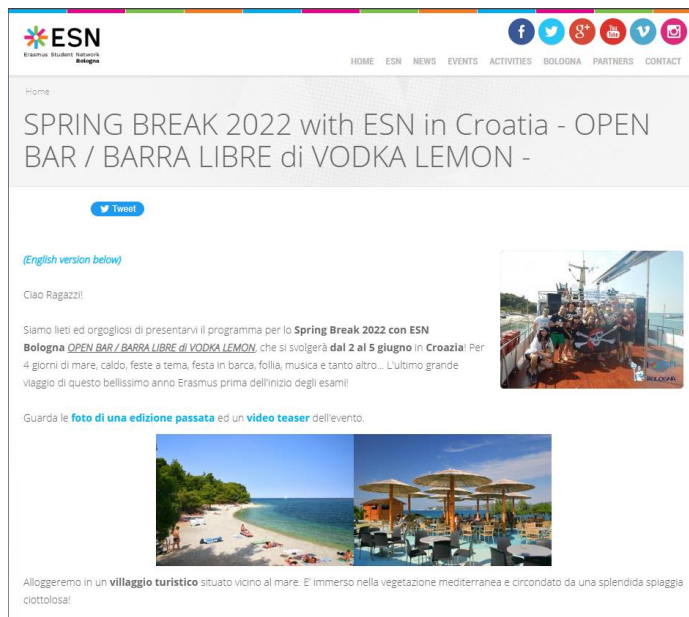


Ilustración 1 - Página web versión escritorio ESN

Aquí observamos como en este caso, las imágenes no se muestran correctamente en la versión móvil de página web, desplazando el texto. Además de contar con una tosca navegación por la página, poco intuitiva y difícil de utilizar, debido a que cuando fue desarrollada no se tuvo en cuenta el mercado móvil y dista mucho de contar con un diseño adecuado para esta plataforma.

2.1.2 Aplicaciones móviles de eventos

Aquí encontramos ejemplos de aplicaciones de eventos. En este caso sí que encontramos software de calidad para dispositivos móviles. Aplicaciones realizadas de forma nativa para estos dispositivos, que cuentan con una navegación optimizada y un diseño agradable. Estas aplicaciones, si bien ofrecen un servicio similar al de Erasmus Events, no están pensadas para estudiantes, si no para un público muy general, por lo que servirán de inspiración sobre cómo debe funcionar nuestro proyecto. Podríamos decir que nuestro producto es una especificación de este tipo de aplicaciones ya existentes.

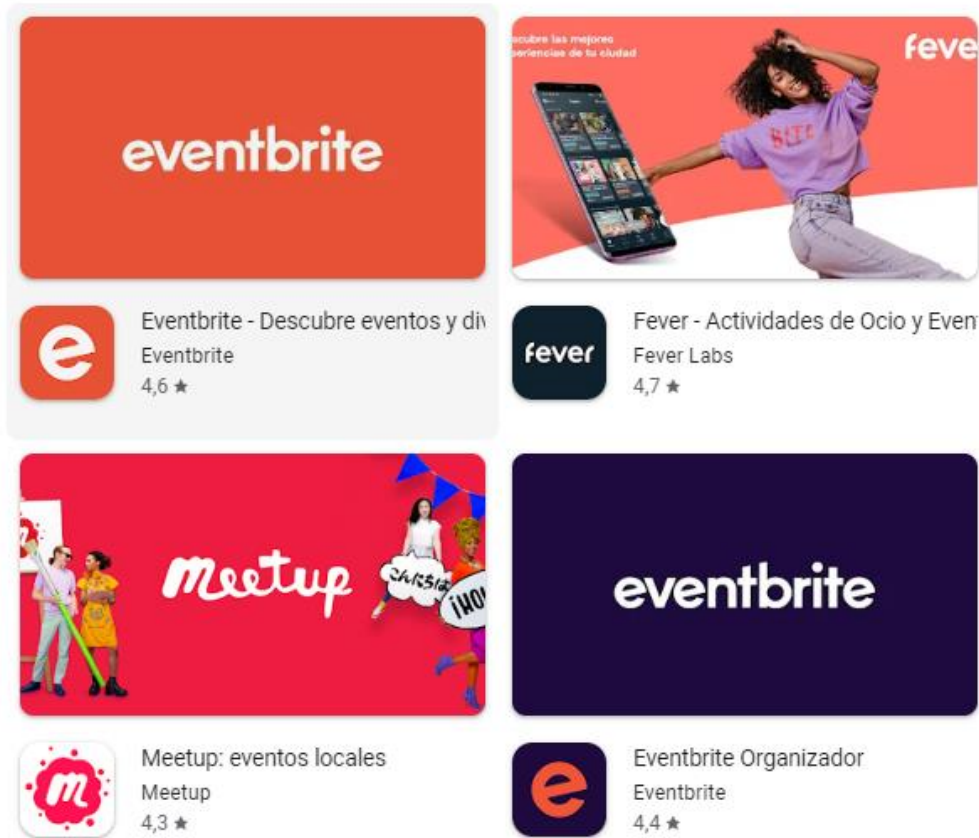


Ilustración 3 Apps de eventos en PlayStore

Eventbrite [4]

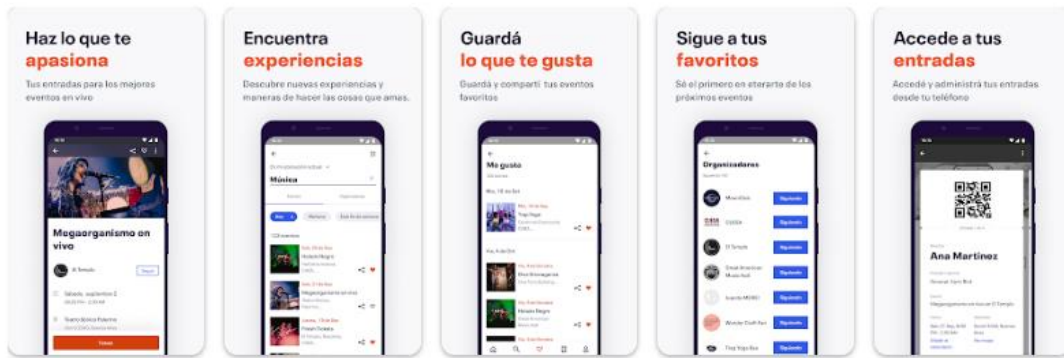


Ilustración 4 - Capturas pantalla App eventos EventBrite

Fever [5]

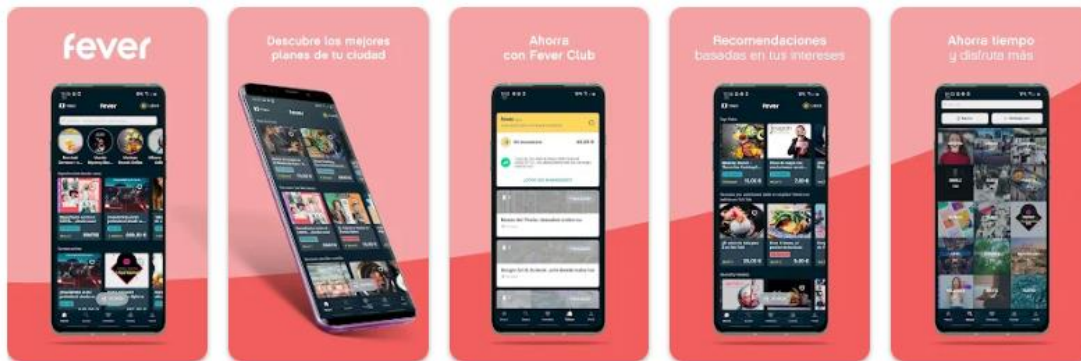


Ilustración 5 - Capturas pantalla App eventos Fever

2.1.3 Aplicaciones erasmus específicas

Observamos aquí dos ejemplos de aplicaciones cuyo objetivo sería el mismo que el de nuestro proyecto, pero que nuevamente cuentan con un diseño anticuado o una navegación tosca. A su vez también se limitan a un solo destino erasmus, son específicas para una ciudad, por lo que no llegan a abarcar tanto público como Erasmus Events.

Erasmus in Czech Republic [6]

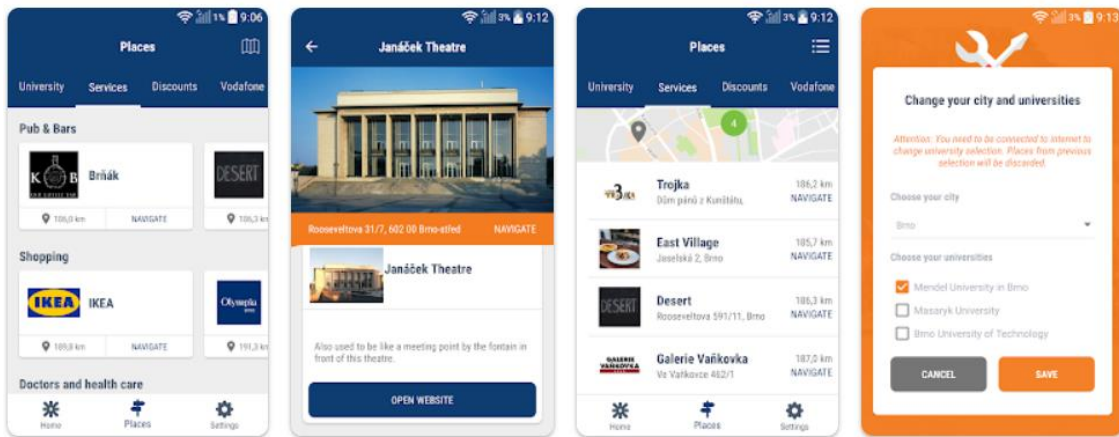


Ilustración 6 - Capturas pantalla App erasmus, Erasmus in Czech Republic

Erasmus Barcelona [7]



Ilustración 7 - Capturas pantalla App erasmus, Erasmus Barcelona

2.1.4 Conclusiones

Observando las alternativas del mercado podemos concluir que, aunque si podemos encontrar aplicaciones con funcionalidades u objetivos similares, no encontramos una que nos ofrezca todo lo que buscamos en ErasmusEvents, hecho que deja vía libre para que este proyecto alcance el éxito y tenga cabida en el mercado. Por ejemplo, en el caso de Erasmus Barcelona, no contamos con una sección de eventos oficiales, ni un mapa donde poder consultar lugares de interés. Por el contrario, en Erasmus in Czech Republic si encontramos el mapa o la sección de eventos, pero los usuarios no disponen de la posibilidad de publicar sus propios eventos.

3. Estudio del problema

3.1. Solución Actual

Los principales inconvenientes de las aplicaciones disponibles son que no se encuentran centralizadas, es decir, cada organización es independiente, con sus propias páginas, eventos y socios. Por lo que los usuarios finales, los estudiantes, encuentran difícil el hecho de estar al tanto de todas sus oportunidades.

Con la solución desarrollada se pretende solucionar todo esto, agrupar en una sola App toda la información que las distintas organizaciones y estudiantes puedan aportar, permitiendo estar en todo momento al corriente de todo.

3.2. Estudio de viabilidad

Durante el Sprint o del proyecto, Sprint en el que se realizaron todas las acciones necesarias para iniciar el proyecto (modelo de negocio, documentación, investigación, etc.) Se realizó también el estudio de viabilidad en función de 2 variables, tiempo y recursos técnicos/operativos.

3.2.1. Viabilidad temporal

Realizando un estudio sobre la viabilidad temporal se decidió dividir el proyecto en 4 Sprints, los cuales se configuraron de la siguiente forma:

- Sprint 0, duración de 3 semanas, 15h
- Sprint 1, duración de 4 semanas, 40h
- Sprint 2, duración de 4 semanas, 40h
- Sprint 3, duración de 4 semanas, 40h
- Sprint 4, duración de 3 semanas, 40h

Teniendo en cuenta que el esfuerzo esperado para la realización del TFG es de 300 horas aproximadas, encontramos que a cada Sprint le corresponden 75h de trabajo. El tiempo de estos Sprints están expresado en horas ideales de programación por lo que a los mismos hay que sumar los tiempos de investigación sobre tecnologías, redacción de la memoria y demás trabajo.

Debido a esto al Sprint 0 se le han asignado menos horas de tareas, dejando así alrededor de unas 60 horas de investigación sobre la tecnología. Horas que se dedicarán a aprender a utilizar tanto el IDE como el lenguaje de programación, así como el Framework, control de versiones, etc.

El resto de Sprints tienen alrededor de 35h que serán invertidos en complementar este aprendizaje sobre la tecnología, así como en la redacción de la memoria del proyecto.



Se espera que la realización del proyecto se lleve a cabo entre los meses de febrero a junio, siendo asignado un sprint a cada mes y comprendiendo un total de 21 semanas, siendo el tiempo de dedicación semanal esperado de 15h.

3.2.2. Viabilidad técnica / operativa

En una primera instancia se buscaba desarrollar la App tanto para Android como para IOS, idea que fue rápidamente desechada debido a que el alcance del proyecto aumentaba de forma que se convertía en inviable teniendo en cuenta los recursos operativos.

Por esta razón y teniendo en cuenta la viabilidad técnica, finalmente se optó por desarrollar el proyecto exclusivamente para Android, ya que el desarrollo en IOS hubiera supuesto tener a disposición del alumno un dispositivo Mac donde poder ejecutar el IDE de Apple. A su vez, como podemos observar en la siguiente gráfica, más del 80% de dispositivos móviles cuentan con el sistema operativo Android, por lo que se abarcará un mayor número de posibles usuarios.

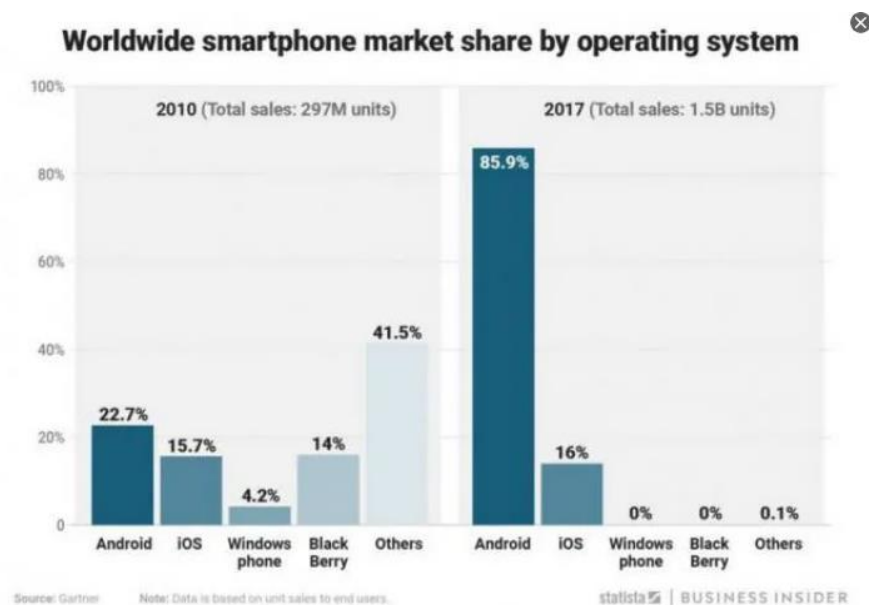


Ilustración 8 - Porcentaje de sistemas operativos en smartphones [8]

Finalmente, todas las herramientas utilizadas son gratuitas u ofertan una versión libre, lo que permite el desarrollo del proyecto sin requerir un desembolso monetario inicial.

Las herramientas seleccionadas para el desarrollo del proyecto son las siguientes:

-**Android Studio [9]**, entorno de desarrollo oficial para la plataforma Android lanzado por Google en 2014.

-**GitHub [10]**, controlador de versiones más popular para el desarrollo de software. Propiedad de Microsoft y reconocida como la plataforma más importante de colaboración para proyectos de código abierto.

-**Firestore [11]**, plataforma para el desarrollo de aplicaciones web y móviles ubicada en la nube, ofrece un conjunto de herramientas para la creación y sincronización de proyectos en la nube. Propiedad de Google.

-**Figma [12]**, herramienta de prototipado y edición de gráficos vectoriales. Su versión gratuita permite diseñar la interfaz gráfica y obtener renderizados que nos permitan prototipar nuestra aplicación.

3.3. Modelo de negocio [13]

Con el creciente aumento en el consumo de Apps año tras año, se han desarrollado nuevos entornos y modelos de negocio para monetizar esos productos software.

Aunque muchas empresas buscan aplicaciones para sus negocios con el fin de crear una marca y fidelizar clientes, utilizando estas como estrategias de marketing, otras desarrollan software con el fin de comercializarlo, como es el caso de ErasmusEvents.

Los 3 modelos de negocio que encontramos actualmente en prácticamente todas las aplicaciones monetizadas son: El modelo premium, modelo Freemium, modelo Mixto.

3.3.1. Modelo Premium:

Popular entre aplicaciones que aportan un servicio de gran beneficio para el usuario, o se distancian de la competencia al ser más eficientes, innovadoras o deseadas.

En este modelo, los clientes pagan un precio inicial para adquirir la App y poder hacer uso de esta. El precio de este tipo de aplicaciones oscila entre 0,99 y 4,99€.

3.3.2. Modelo Freemium:

El más extendido en el mundo de aplicaciones móviles, consiste en permitir la descarga de forma gratuita del software y luego realizar una serie de acciones que permitan generar ingresos.

Estas acciones suelen consistir en compras in-App, mediante micro pagos para adquirir servicios, ventajas, funcionalidades exclusivas. Las principales formas de monetización Freemium son:

-**Compras in-App**, como en plataformas de comercio electrónico, donde la aplicación es un canal para conseguir las ventas buscadas, videojuegos, en los que habitualmente se realizan pagos para adquirir ventajas sobre el resto de los jugadores, contenido o cosméticos, etc.

-**Pago por suscripción**. En este modelo los clientes pagan una tarifa bien semanal, mensual, anual... Para poder hacer uso de la App. El precio de estas tarifas no suele superar los 10€ mensuales y estos productos



suelen estar libres de publicidad. Este método es muy utilizado por empresas que ofrecen paquetes de servicios o contenidos en internet.

-Publicidad in-App. Modelo de negocio sencillo, se ofrece acceso a la App de forma gratuita y se obtienen beneficios gracias a la publicidad que se muestra a los usuarios que navegan por la aplicación. Sin embargo, hay que ser cautos a la hora de ubicar esta publicidad, ya que los anuncios pueden generar una gran insatisfacción en la opinión de los clientes. Hay que comprobar en que espacios, formatos, momentos y frecuencias mostrar estas herramientas de monetización para que tengan éxito y no perjudiquen el uso de la aplicación.

-Venta de datos. En este caso, los usuarios obtienen los servicios de la aplicación al autorizar que sus datos sean recopilados y vendidos por la empresa propietaria del software. No es una técnica muy aceptada socialmente, no obstante, en muchos casos las aplicaciones combinan esta técnica junto a otras formas de monetización para maximizar los beneficios.

3.3.3. Modelo Mixto:

Este modelo está pensado para maximizar los beneficios de los productos software. Consiste en la combinación de los distintos modelos presentados anteriormente. Encontramos ejemplos de esto en muchas de las aplicaciones más populares de la Play store, como por ejemplo Spotify, donde se combina el modelo Freemium de descarga gratuita y monetización gracias a publicidad in-App, además de paquetes de suscripción premium.

3.3.4. Conclusiones y modelo elegido:

Analizando los modelos anteriores se ha concluido la utilización de un modelo Freemium con publicidad in-App para ErasmusEvents. Esto ha sido así debido a que con esta aplicación no se busca generar ingresos, y con su monetización solo se busca cubrir los gastos derivados de los servicios en la nube que se ofrece.

Por esto en ErasmusEvents no encontraremos tarifas abusivas, ni suscripciones premium, ni beneficios a ciertos usuarios ni nada por el estilo. Solo se situarán anuncios en momentos específicos para generar lo necesario como para cubrir los costes de mantenimiento de la App.

3.3 Conclusiones

En este apartado hemos podido concluir que es importante conocer los recursos y tiempo disponibles para la realización de un proyecto, ya que muchas veces no disponemos de lo necesario para llevar a cabo un desarrollo que nos habíamos propuesto inicialmente.

4. Diseño de la solución

Una vez identificados los requisitos y objetivos de la aplicación, durante el primer sprint del proyecto (Sprint 0) se ha llevado a cabo la fase de diseño de la solución, esto incluye la creación de una matriz Dafo, un gráfico Lean Canvas, un mapa de características, un modelo de dominio y el diseño de los Mock-Ups de las distintas funcionalidades de la App.

A su vez también se ha establecido el modelo de negocio que seguirá el proyecto cuando alcance su fase de despliegue.

4.1. Matriz Dafo [20]

La matriz Dafo es un método de análisis que nos ayuda a conocer la situación actual de nuestro proyecto, saber de qué elementos disponemos y cuáles serán nuestras necesidades. A su vez podemos obtener una visión de las fallas existentes con las que deberemos lidiar. De esta forma obtenemos de forma sintetizada un diagnóstico de las debilidades, amenazas, fortalezas y oportunidades de nuestro producto.

Las oportunidades y amenazas corresponden a los aspectos externos, como pueden ser la situación económica nacional, la evolución de la moda, etc. Son factores que debemos prever para lograr adaptarnos y evitar futuros desastres.

Por otra parte, encontramos las fortalezas y debilidades, aspectos que la empresa puede controlar o modificar. Hacen referencia a factores internos, como pueden ser los conocimientos y experiencias personales del mercado, la materia prima utilizada, etc. Es importante conocer estos elementos para aportar un buen crecimiento al proyecto.

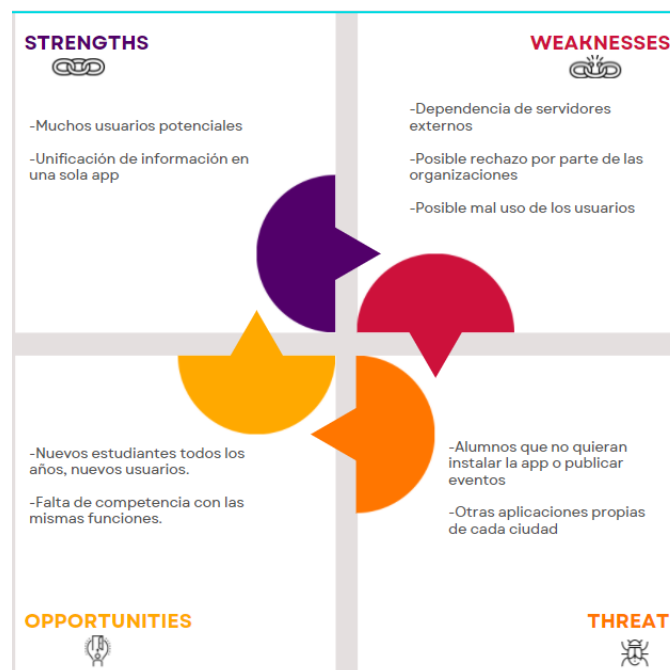


Ilustración 9 - Matriz DAFO

Observando la matriz podemos identificar cuáles son los puntos fuertes y débiles de nuestro proyecto.

Por una parte, partiendo por las **fortalezas** podemos observar cómo al tratarse de una App para estudiantes erasmus, estos cambiarán año tras año, por lo que contaremos con un flujo constante de nuevos usuarios que accederán a la plataforma e interaccionarán con ella, a la vez de no solo los erasmus pueden disfrutar de ella, sino que otros estudiantes locales pueden utilizar la app para conocer gente, practicar idiomas u obtener información de interés de destinos que les interesen. Otra fortaleza sería la de la unificación que se busca con este proyecto, ya que ante la duda de si utilizar esta aplicación u otra similar, esta ofrecerá un contenido más completo que otro software más genérico o específico.

Analizando las **debilidades**, la más significativa sería la dependencia de servidores externos, en nuestro caso los de la plataforma Firebase, ofrecidos por Google. Esto es debido a que este servicio parte de una tarifa gratuita, pero conforme aumenta el número de usuarios también aumenta el precio de esta, por lo que conforme aumente el número de descargas se deberán buscar formas de financiación para hacer frente al coste. También debemos prever un posible rechazo por parte de las organizaciones, las cuales se nieguen a publicar sus eventos en nuestra plataforma, teniendo que buscar métodos alternativos para incluir estos eventos. A su vez el mal uso de los usuarios será algo inevitable, por lo que hay que tener esto en cuenta de cara a implementar restricciones o soluciones para mitigar estos problemas.

En cuanto a las **oportunidades**, nos encontramos con un mercado que se renueva año tras año y que crece de forma considerable. Cada vez son instaladas más aplicaciones en nuestros dispositivos gracias al aumento de potencia y memoria de estos que junto al flujo constante de nuevos estudiantes nos ofrece posibilidades ilimitadas de crecimiento. También la falta de competencia que ofrezca los mismos servicios que nuestra App nos brinda de una gran oportunidad estratégica.

Finalmente, en cuanto a las **amenazas**, los estudiantes que no hagan uso de nuestro software o utilicen otro similar pueden afectar negativamente en nuestro alcance, ya que si no se logra crear una comunidad activa la App carecerá de contenido y como consecuente de interés.

4.2. Lean Canvas [21]

Lean Canvas es una herramienta pensada para proyectos de emprendimiento que permite visualizar modelos de negocio. Es muy recomendada para startups y se centra en generar nuevas ideas y llevarlas a prototipado para verificar si el mercado necesita la solución que se propone. Es útil ya que nos permite identificar los problemas más importantes que se encuentran en el mercado, que soluciones se precisan, los canales a través de los cuales llegar a los usuarios finales, las métricas claves, como se llegarán a los ingresos necesarios...

PROBLEM <small>List your top 1-3 problems.</small> <p>Es muy difícil para los nuevos estudiantes erasmus estar al día de todas las actividades que se organizan en su ciudad de destino, por lo que muchas veces pierdes la oportunidad simplemente por no enterarte de donde o cuando hay algún evento.</p>	SOLUTION <small>Outline a possible solution for each problem.</small> <p>Una aplicación para smartphones que permita estar al día de forma cómoda y sencilla. A su vez dando la posibilidad a los estudiantes de publicar sus propios eventos para conocer gente.</p>	UNIQUE VALUE PROPOSITION <small>Single, clear, compelling message that states why you are different and worth paying attention.</small> <p>Todos los eventos reunidos en una sola aplicación. -Viajes oficiales -Fiestas -Viajes extraoficiales -Descuentos -Eventos -Quedadas</p>	UNFAIR ADVANTAGE <small>Something that cannot easily be bought or copied.</small> 	CUSTOMER SEGMENTS <small>List your target customers and users.</small> <ul style="list-style-type: none"> -Estudiantes erasmus -Organizaciones erasmus -Empresas con publico erasmus -Bares e negocios de la restauración
	KEY METRICS <small>List the key numbers that tell you how your business is doing.</small> <ul style="list-style-type: none"> -Nuevos usuarios / mes -Descargas app / mes -Publicaciones / mes 		CHANNELS <small>List your path to customers (inbound or outbound).</small> <p>Anuncios en redes sociales Difusiones en grupos erasmus</p>	
COST STRUCTURE <small>List your fixed and variable costs.</small> <ul style="list-style-type: none"> -Coste servicios en la nube (variable) -Desarrolladores (fijo) -Publicidad (fijo) 		REVENUE STREAMS <small>List your sources of revenue.</small> <p>Ingresos por publicidad</p>		

Ilustración 10 - Tabla Lean Canvas

4.3. Mapa de características:

En esta sección se desarrolló el mapa de características, consistente en plasmar con “post-its” el resultado de una lluvia de ideas. Estas características podrán más tarde corresponder a Unidades de trabajo para convertirse en funcionalidades de la App.

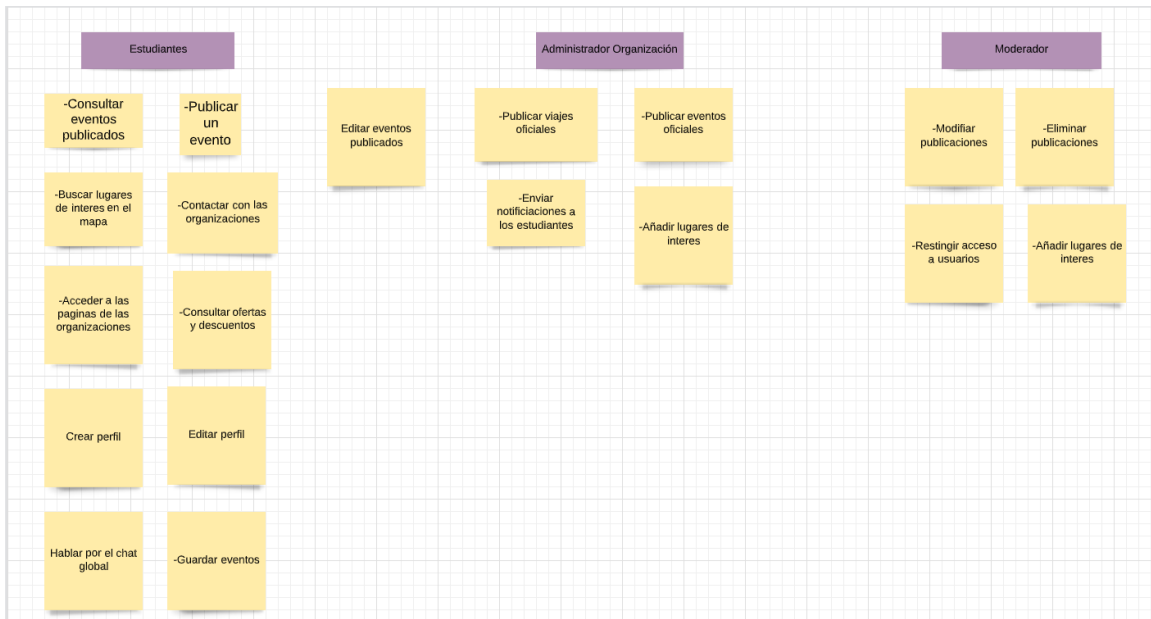


Ilustración 11 - Mapa de características

4.4. Modelo de dominio

La realización de un modelo de dominio en nuestro proyecto nos dota de un mapa conceptual que represente el vocabulario y los conceptos claves de un proyecto. A su vez también se representan las relaciones entre todas las entidades implicadas.

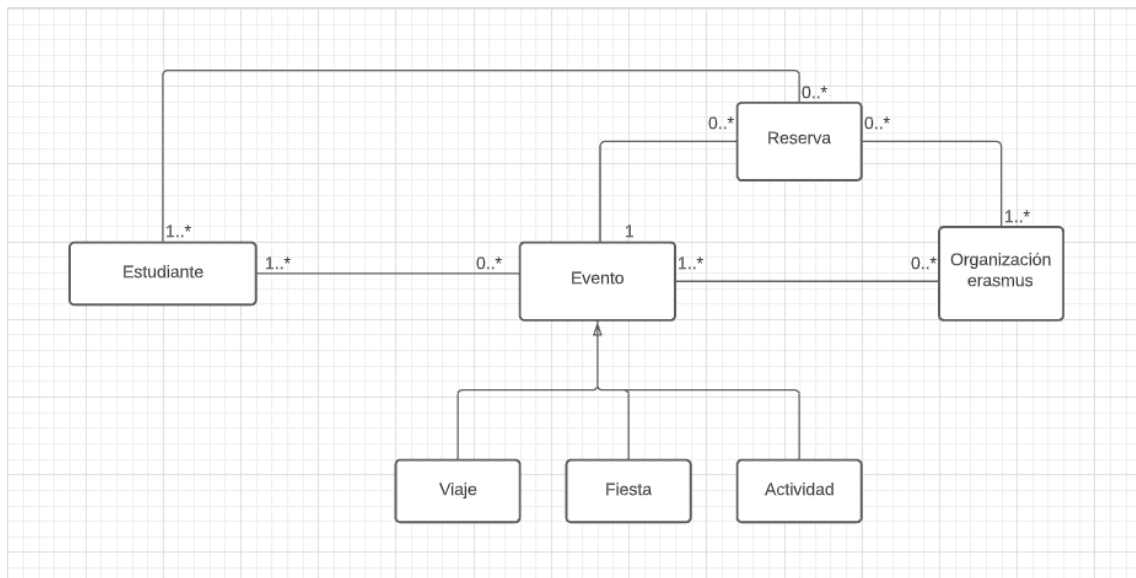


Ilustración 12 - Modelo de dominio

4.5. Mock ups [22]

Los Mock ups son una forma de mostrar al mundo como se verá nuestro producto una vez finalizado. Realizar un MockUp consiste en fingir un montaje real que nos permita crear nuestra identidad visual.

Los MockUps son muy útiles en el desarrollo de una aplicación móvil ya que nos pueden servir primero para obtener la aprobación del cliente, a la vez que nos ayudará mucho a defender la objetividad de nuestra propuesta, ya que estamos aplicando un diseño a cada uno de los escenarios pensados. En nuestro caso nos han servido como para decidir qué elementos del mapa de características deben convertirse en funcionalidades, a la vez estos MockUps han servido como “guía” a la hora de realizar la programación de la interfaz gráfica.

Los diseños han sido creados utilizando la herramienta Figma [12] y se ha usado como inspiración distintos mockups publicados de forma abierta por distintos diseñadores gráficos en la plataforma Dribbble [14].

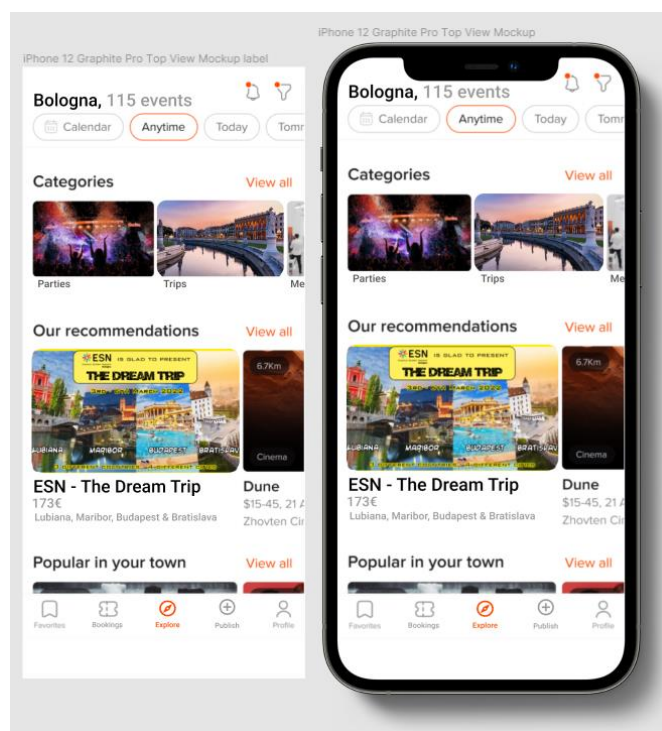


Ilustración 13 - MockUp Ventana Explorar

En la ventana “Explorar”, la primera que encontrará el usuario una vez inicie sesión en la aplicación, encontramos unos filtros rápidos en la parte superior donde indicar el periodo de tiempo en el que queremos buscar eventos, seguido de una serie de categorías donde especificar nuestra búsqueda. En las distintas secciones encontramos eventos clasificados por popularidad, por organización, por ciudades, etc.

Finalmente, en la parte inferior de la pantalla contamos con una barra de navegación que nos permite movernos por las distintas ventanas.

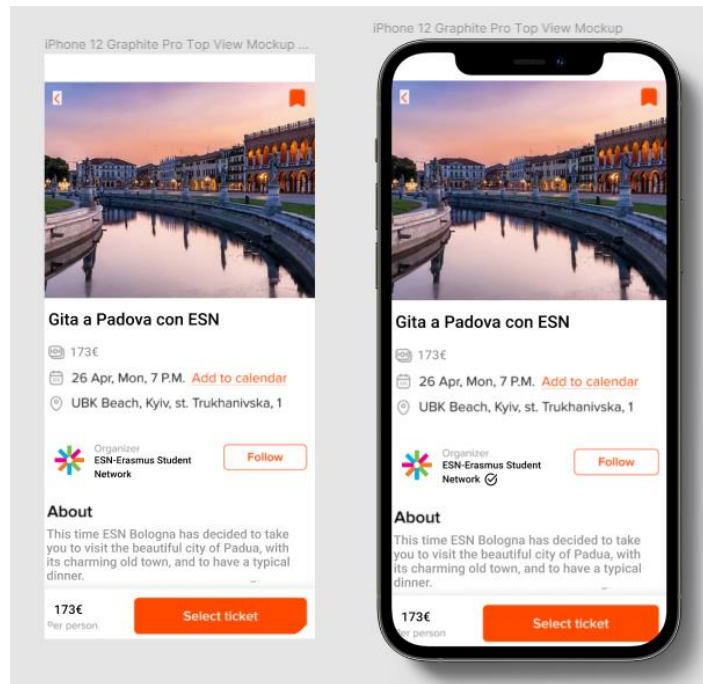


Ilustración 14 - MockUp ventana detalles evento

Una vez accedamos a un evento esta será la pantalla que visualizaremos, donde podemos encontrar detalles de este como el precio, la fecha en la que se llevará a cabo, la ubicación, el organizador, una breve descripción y si lo permite el evento, un enlace para realizar la reserva o el pago en caso de que fuera necesario.

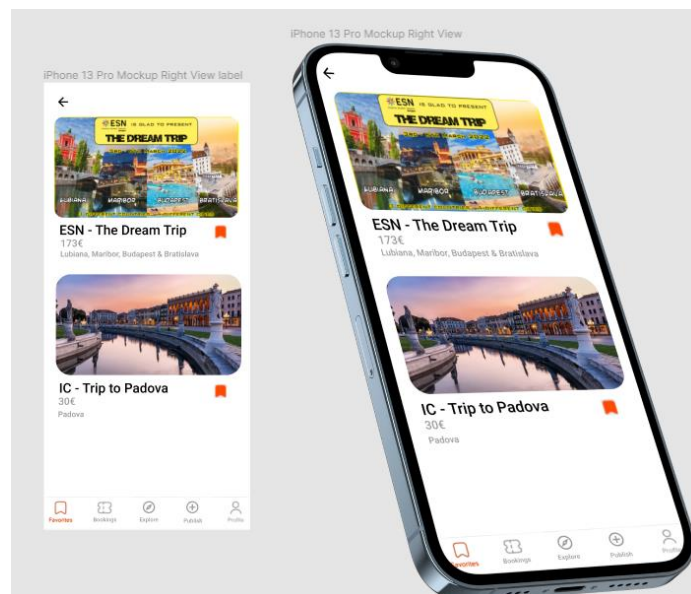


Ilustración 15 - MockUp ventana eventos guardados

Una muy buena funcionalidad de la app es la de permitir guardar los eventos que nos interesen para tenerlos a mano más rápidamente en un futuro. En este MockUp podemos ver como se espera que funcione esta mecánica, el usuario guarda un evento con el icono naranja, quedando este en una sección específica en la barra de navegación inferior.

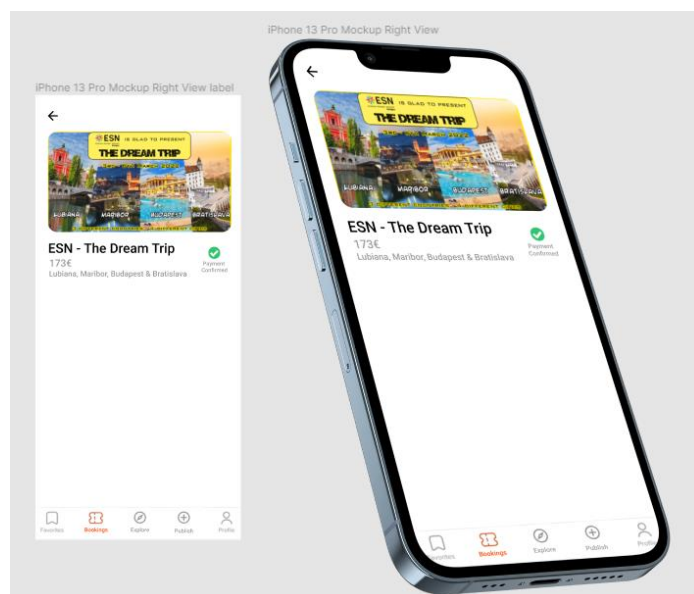


Ilustración 16 - MockUp ventana eventos pagados

A la vez que los eventos guardados, la App también debe mantener un registro de los eventos ya reservados o pagados como es el caso del que se muestra en el MockUp.

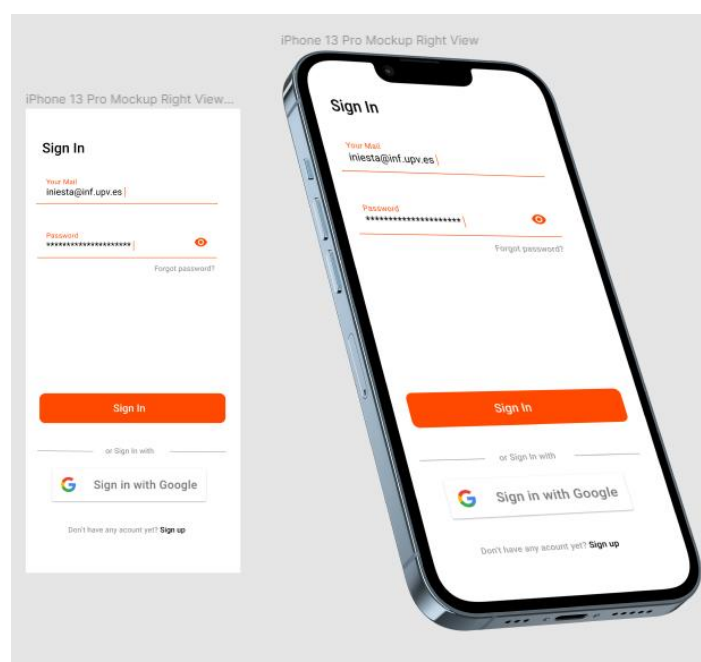


Ilustración 17 - MockUp ventana inicio sesión

En este MockUp encontramos una sencilla ventana de inicio de sesión donde el usuario bien puede identificarse utilizando su correo y contraseña o su cuenta de Google.

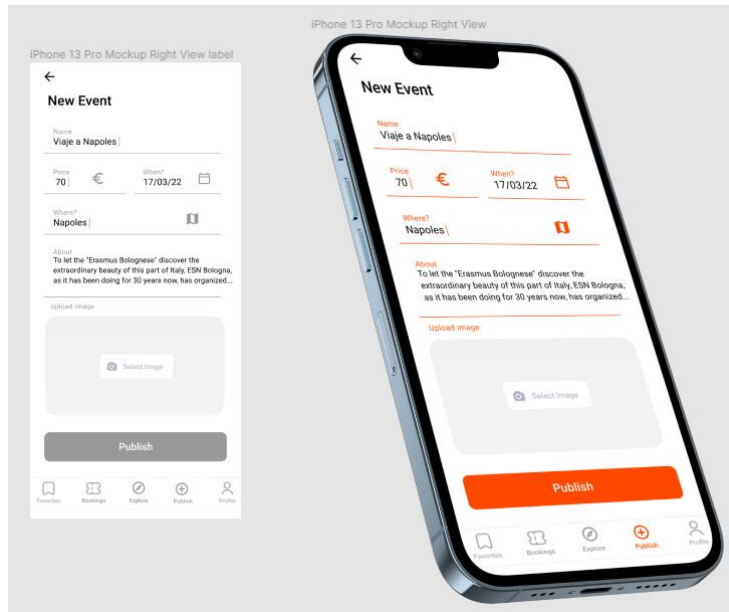


Ilustración 18 - MockUp ventana publicar evento

Otra de las funcionalidades principales de ErasmusEvents es la de ofrecer la posibilidad a los estudiantes de que publiquen sus propios eventos. En este MockUp podemos ver como se realizaría esta acción, pidiendo al usuario que introduzca los datos del evento junto a una imagen para su posterior publicación.

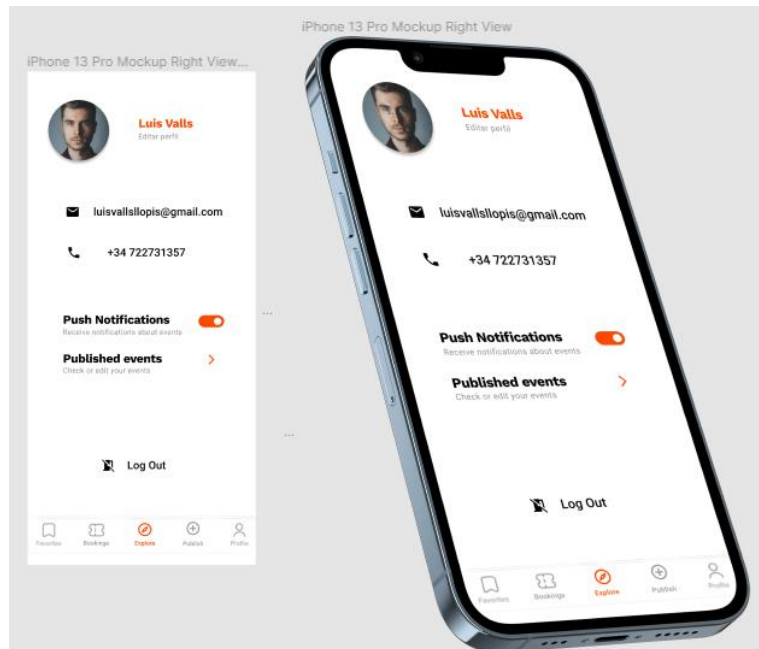


Ilustración 19 - MockUp ventana perfil

En el siguiente MockUp vemos el diseño de la ventana de perfil de usuario, donde se muestran algunos datos como el correo o el teléfono, junto con la opción de activar o desactivar las notificaciones, acceder a los eventos publicados, cerrar sesión o modificar el perfil.

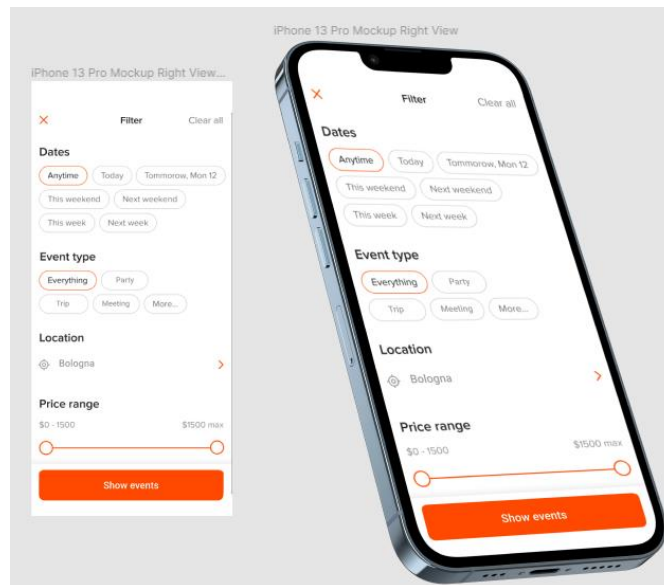


Ilustración 20 - MockUp Ventana FilTROS

En este MockUp vemos la ventana de filtros, ventana desde la cual podemos establecer una serie de parámetros para filtrar los eventos que van a ser mostrados.

4.7. Gestión de alcance del trabajo

Al tratarse de un proyecto Ágil, la gestión del trabajo en un sprint se realizará en base a estimaciones, utilizando la métrica de Horas Ideales de programación.

Durante la realización de cada sprint se reestiman constantemente los tiempos de desarrollo asociados a cada Ut de forma que se puede comprobar la coherencia entre el esfuerzo determinado inicialmente y el real para terminar el trabajo. Con estas actualizaciones podemos modificar el alcance en función de los recursos con lo que contemos en cada momento para avanzar cumpliendo los horarios y plazos establecidos.

De esta forma, si durante una reestimación el esfuerzo necesario supera al esperado podemos modificar el alcance del sprint reduciendo el número de UTs o realizando un MVP (mínimum viable producto), técnica que reduce una funcionalidad para obtener una versión ejecutable, aunque sea menos completa que lo que se esperaba inicialmente. De igual manera, si durante las reestimaciones observamos que se dispone de más tiempo durante el sprint que podrán añadir nuevas UTs al mismo para su desarrollo.

4.8. Patrones de diseño

Para el desarrollo de ciertas funcionalidades de la aplicación se prevé hacer uso de patrones de diseño.

¿Que son estos patrones?[15]

Los patrones de diseño o *design patterns* en inglés, son unas “plantillas” que nos dicen como abordar problemas comunes presentes en el desarrollo de un producto software. Estas soluciones tratan de ser las más apropiadas para el problema en cuestión.

Existen 3 tipos de patrones de diseño, los patrones creacionales, estructurales y de comportamiento, clasificados de esta forma en función del tipo de solución que aporten.

Los patrones creacionales proporcionan mecanismos que aumenten la flexibilidad y la reutilización de código en la creación de objetos, los estructurales facilitan métodos eficientes de cómo tratar a las composiciones de clases y las estructuras de objetos. Finalmente, los patrones de comportamiento, que nos ayudan a mejorar la comunicación entre objetos de clase.

Existe una gran cantidad de estos patrones de diseño, a continuación, vamos a detallar los que se prevé que sean utilizados en nuestra app y como se espera que funcionen.

En concreto los patrones que se espera utilizar son el patrón Singleton y el de Inyección de dependencias.

4.8.1. Patrón Singleton

El patrón Singleton busca asegurar que solo existe una instancia de una clase concreta. Esta instancia única tiene una accesibilidad global, lo que permite que cualquier otra clase que necesite hacer uso de esta pueda acceder a ella sin necesidad de crear una instancia nueva.

Su implementación se consigue estableciendo el constructor como privado, de esta forma, al no se accesible desde fuera de la clase, es otra función la que se encarga de facilitar la instancia o de crearla la primera vez que se llame a la misma.

Para garantizar el acceso global se establece como una instancia estática.

Se prevé la utilización de este patrón en las clases responsables de la conexión con la base de datos, ofreciendo junto al patrón de inyección de dependencias un repositorio único con el cual realizar todas las operaciones relacionadas con la misma de una forma sencilla y globalizada.

4.8.2. Patrón Inyección de dependencias [16]

Muchas veces nos encontramos el caso de que una clase necesita una referencia a otra. Esta necesidad se denomina “dependencia” y nos presenta el caso de una clase donde para ejecutarse necesita una instancia de otra clase.

Obtener este objeto que necesita es esencial para el funcionamiento del programa y esto se puede hacer creando e inicializando su propia instancia, obtenerla de otro lugar, de un ViewModel, por ejemplo; o recibéndola como parámetro.

Esta tercera opción es la inyección de dependencias. Este enfoque busca tomar las dependencias de una clase y proporcionarlas a las demás que puedan necesitarlas.

Este patrón se prevé utilizar junto al Singleton, encargado de crear una instancia única de la clase “repository” responsable de la conexión con la base de datos, para transmitir esta instancia entre todas las clases que la requieran.

5. Desarrollo de la solución propuesta

ErasmusEvents será desarrollada siguiendo una metodología Ágil, en concreto la Tune-UP. Este desarrollo quedará dividido en 5 Sprints de 1 mes de duración cada uno, lo que llevará un desarrollo de 5 meses. Al final de cada Sprint se obtendrá una versión ejecutable del producto.

Para la gestión y seguimiento de los Sprints se hará uso de la herramienta Worki [17], donde se crearán las UTs y se registrarán los tiempos, avances y modificaciones de estas.

5.1. Sprint 0

El sprint 0 o puesta en marcha busca permitir al equipo preparar la infraestructura de trabajo, poblar de contenido el Backlog, preparar el conjunto de UTs que se implementaran en el Sprint 1, crear los diseños de la App, etc.

5.1.1. Estimación

Recordando la fase de análisis, el Sprint 0 tiene una duración estimada de 3 semanas, o lo que es lo mismo de 19 días. La fecha de inicio estimada fue el 09/02/2022 y la de finalización real el 28/02/2022.

Orden	UT	Creación	Finalización	Línea de trabajo	Sprint	Puntos	Estimación (horas)	Workflow	Proyecto
1010	3 - Realizar mapa de características	09/02/2022	17/02/2022	TFG Luis Valls	Sprint 0	3	1	WF Tarea	Proyecto TFG Luis Valls
1020	5 - Lean Canvas	09/02/2022	12/02/2022	TFG Luis Valls	Sprint 0	1	0.5	WF Tarea	Proyecto TFG Luis Valls
1025	2 - Poblar el backlog	09/02/2022	17/02/2022	TFG Luis Valls	Sprint 0	5	0.5	WF Tarea	Proyecto TFG Luis Valls
1030	4 - Modelo de dominio	09/02/2022	18/02/2022	TFG Luis Valls	Sprint 0	2	0.75	WF Tarea	Proyecto TFG Luis Valls
1040	7 - Preparar ide i control de versiones	09/02/2022	24/02/2022	TFG Luis Valls	Sprint 0	20	3	WF Tarea	Proyecto TFG Luis Valls
1050	26 - Crear mockUps	09/02/2022	28/02/2022	TFG Luis Valls	Sprint 0	40	6	WF Tarea	Proyecto TFG Luis Valls
1060	6 - Crear y poblar base de datos	09/02/2022	28/02/2022	TFG Luis Valls	Sprint 0	20	3	WF Tarea	Proyecto TFG Luis Valls

Ilustración 21 - UTs incluidas Sprint 0

En la siguiente imagen observamos la configuración del sprint con las UTs que lo componen junto a sus estimaciones tanto en horas como en puntos. Vemos como cada UT de este sprint corresponde a una acción para la puesta en marcha del proyecto.

5.1.2. Realización

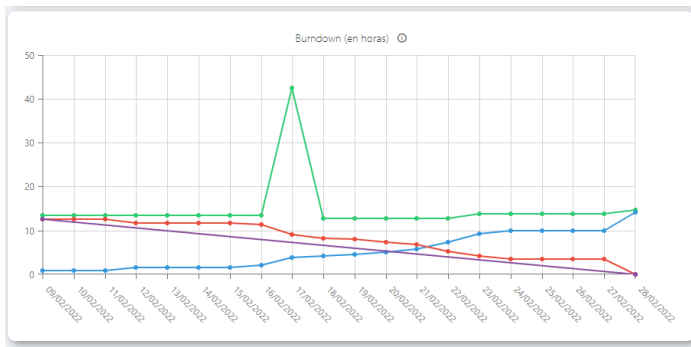


Ilustración 23 - Gráfica Burndown (en horas) Sprint 0

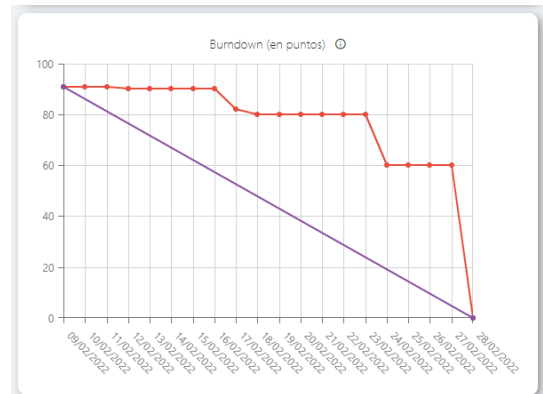


Ilustración 22 - Gráfica Burndown (en puntos) Sprint 0

Observando ambas graficas Burndown [Ilustración 23 - Ilustración 22], tanto en horas como en puntos podemos ver como avanzó el trabajo durante el transcurso del sprint.

En el Burndown por horas podemos observar que la estimación da un salto de una forma muy exagerada, esto se debe a un error a la hora de reestimar una actividad por lo que no se debe tener en cuenta este valor.

Podemos observar como el trabajo fue constante a lo largo de todo el sprint, ya que la gráfica en horas sigue una progresión casi lineal, acercándose a la ideal (línea morada). En cuanto a la gráfica en puntos podemos ver los momentos donde se finalizaron las diferentes UTs, encontrando el mayor número de estas en los últimos días del sprint.

5.1.3. Finalización

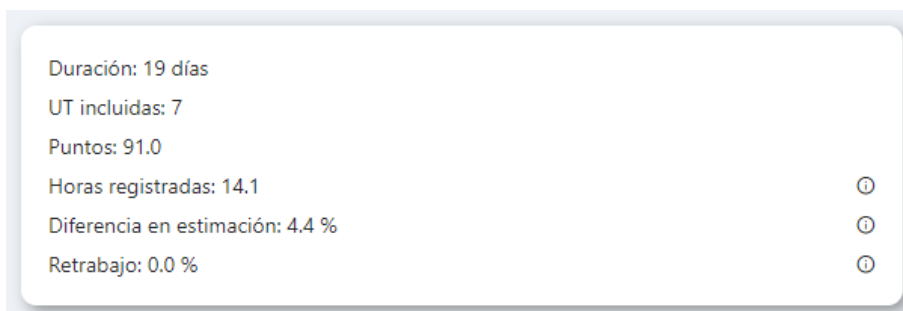


Ilustración 24 - Detalles Sprint 0

El sprint 0 finalizó el 28/02/22 y tuvo una duración real de 19 días. A la finalización del sprint 0 encontramos todo preparado para empezar a programar funcionalidades para nuestra aplicación.

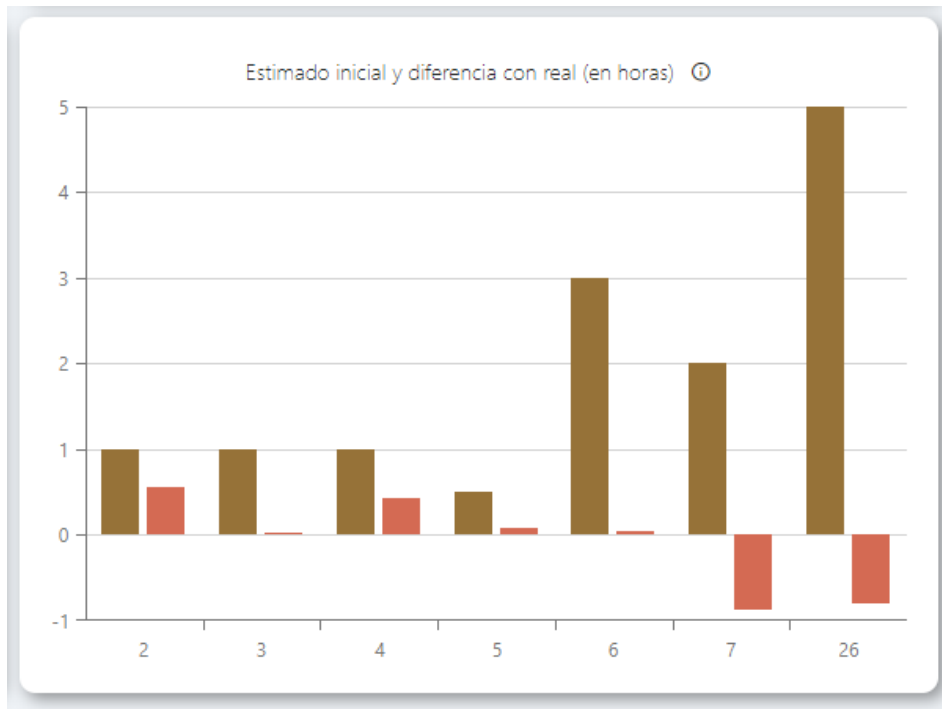


Ilustración 25 - Gráfica diferencia tiempo estimado/real Sprint 0

Observando la gráfica que nos muestra la diferencia entre el tiempo estimado y el real [Ilustración 25] podemos ver como el tiempo de realización de cada UT se ha acercado al estimado, exceptuando la primera UT (poblar backlog), donde el tiempo real fue inferior al estimado (estimado inicial 1h, real 27 minutos) o las dos últimas UTs, preparar IDE/Control de versiones y Crear MockUps, donde el tiempo real superó al estimado.

5.2. Sprint 1

El Sprint 1 corresponde al primer “sprint de programación”, a partir de este punto se empiezan a programar las diferentes funcionalidades de la App.

5.2.1. Estimación

El Sprint 1 una duración estimada de 4 semanas y unas 40 horas de programación. A continuación, se muestra una lista de las UTs que se esperan completar, siendo la suma de todas sus estimaciones de 37 horas, dejando margen para futuras reestimaciones al alza.

UT	Creación	Finalización	Línea de trabajo	Sprint	Puntos	Estimación (horas)	Workflow	Proyecto
33 - Barra navegación inferior	28/02/2022	02/03/2022	TFG Luis Valls	Sprint 1	3	2.5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
10 - Publicar evento	09/02/2022	21/03/2022	TFG Luis Valls	Sprint 1	13	12	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
32 - Mostrar lista de eventos	28/02/2022	08/03/2022	TFG Luis Valls	Sprint 1	20	10	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
35 - Consultar datos Perfil	28/02/2022	31/03/2022	TFG Luis Valls	Sprint 1	8	5.5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
34 - Consultar detalles evento	28/02/2022	31/03/2022	TFG Luis Valls	Sprint 1	5	6	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
8 - Consultar eventos publicados	09/02/2022	31/03/2022	TFG Luis Valls	Sprint 1	3	4.2	WF Desarrollo con Sprints	Proyecto TFG Luis Valls

Ilustración 26 - UTs incluidas Sprint 1

5.2.2. Realización

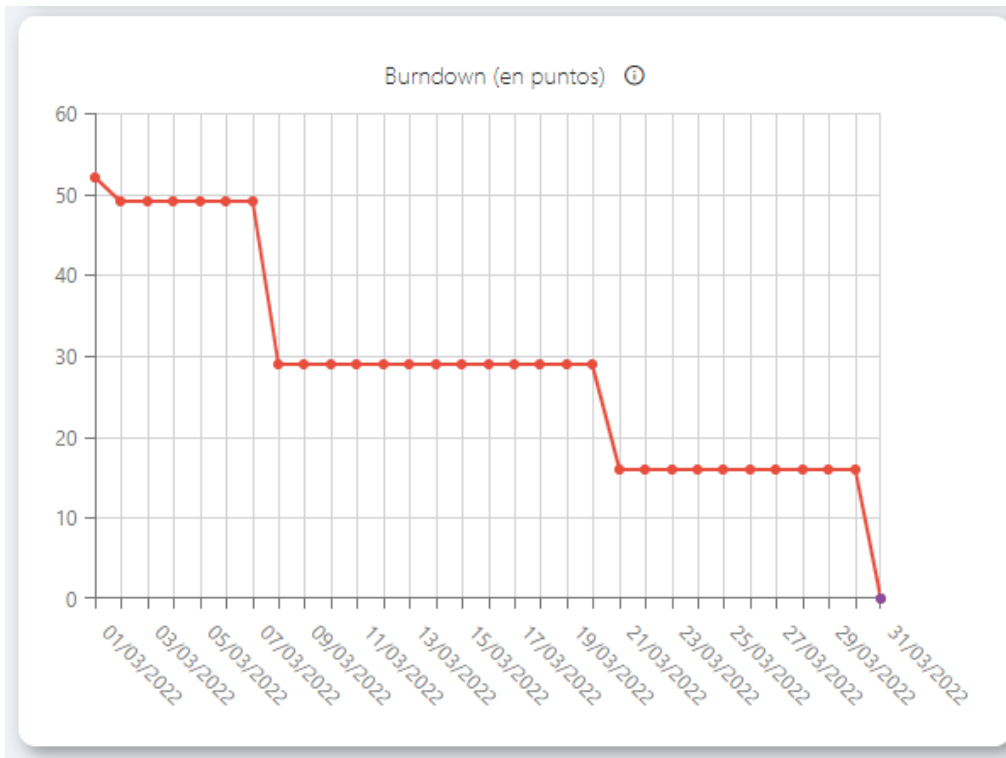


Ilustración 27 - Gráfica Burndown (en puntos) Sprint 1

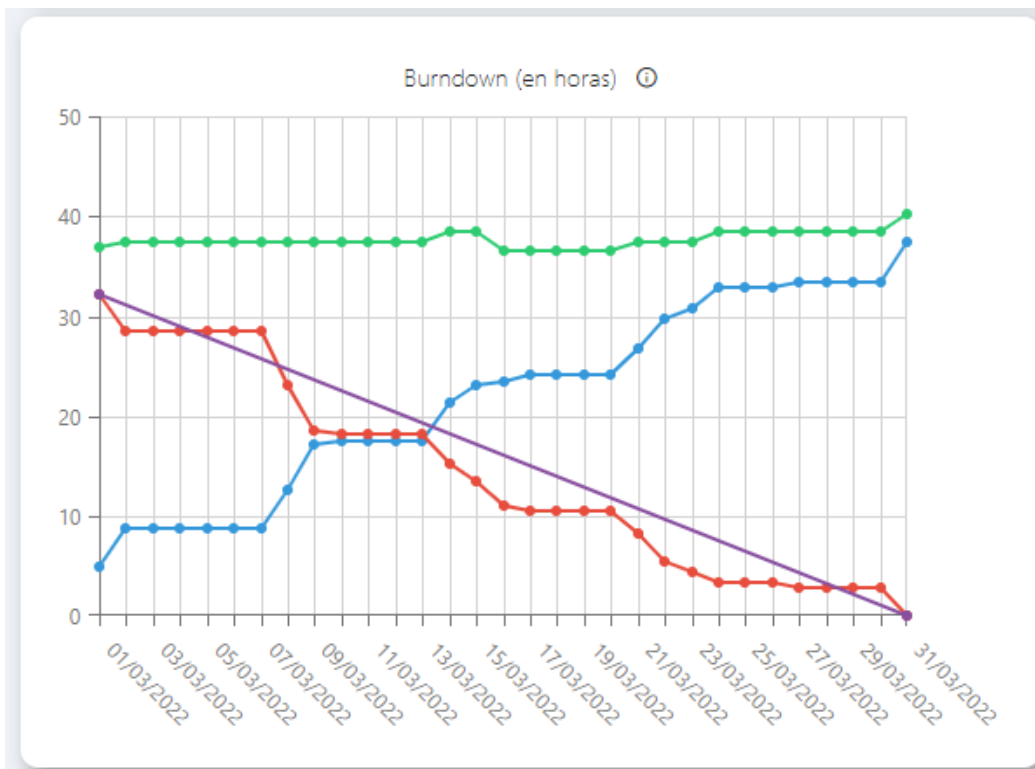


Ilustración 28 - Gráfica Burndown (en horas) Sprint 1

Observando las gráficas Burndown [Ilustración 27 – Ilustración 28], podemos ver cómo fue la realización del sprint.

Vemos como cerca de la mitad la estimación en horas aumentó, para luego volver a disminuir, manteniéndose bastante regular con ligeros aumentos. Finalmente, las horas estimadas al final del sprint superaron a las iniciales en 3 horas y 20 minutos.

No obstante, el esfuerzo real fue de 37 horas y 22 minutos, muy cerca de la estimación inicial, indicándonos que las últimas estimaciones al alza eran erróneas no fue necesario un esfuerzo extra para terminar el trabajo.

También vemos como el esfuerzo en horas fue bastante constante durante el transcurso de los días.

5.2.3. Finalización

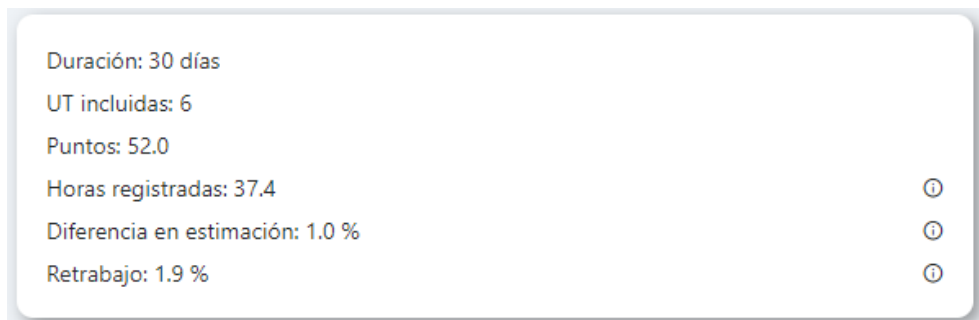


Ilustración 29 - Detalles Sprint 1

El Sprint tuvo una duración real de 30 días y un esfuerzo registrado de 37,4 horas. Superando la estimación inicial en un 1%.

También encontramos que fue necesario el retrabajo en alguna UT para poder solucionar problemas encontrados en las pruebas de aceptación, pruebas detalladas de forma más extensa en el apartado 7, pruebas.

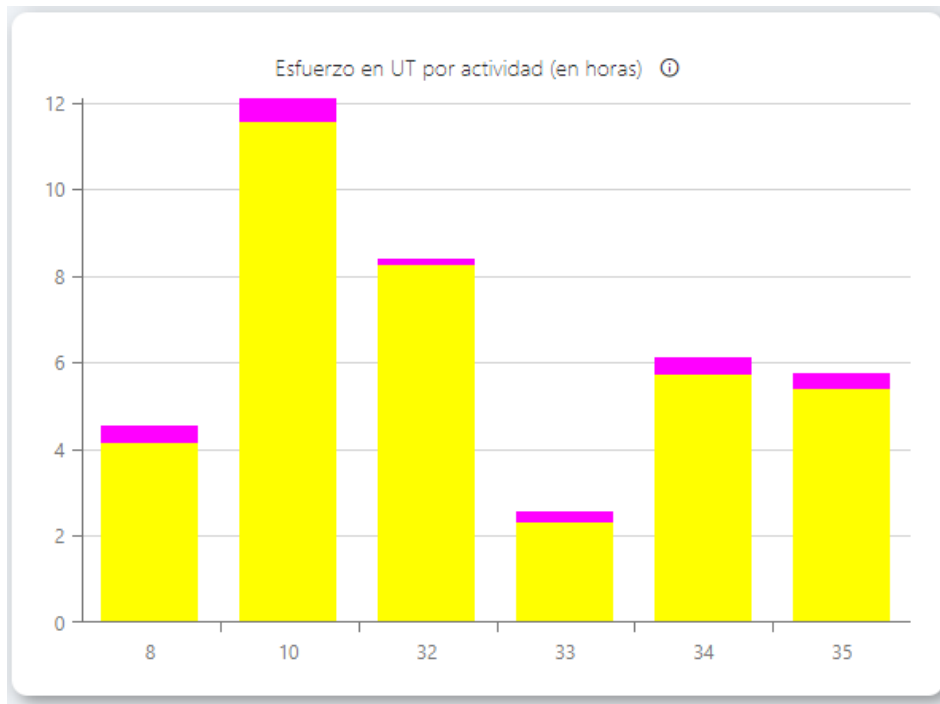


Ilustración 30 - Gráfica Esfuerzo en UT por actividad Sprint 1

En la gráfica de esfuerzo por UTs por actividad [Ilustración 30] vemos el tiempo dedicado a la programación (amarillo) y el esfuerzo dedicado a realizar las pruebas(rosa).

El tiempo de pruebas ha sido de unos 20 minutos de media.

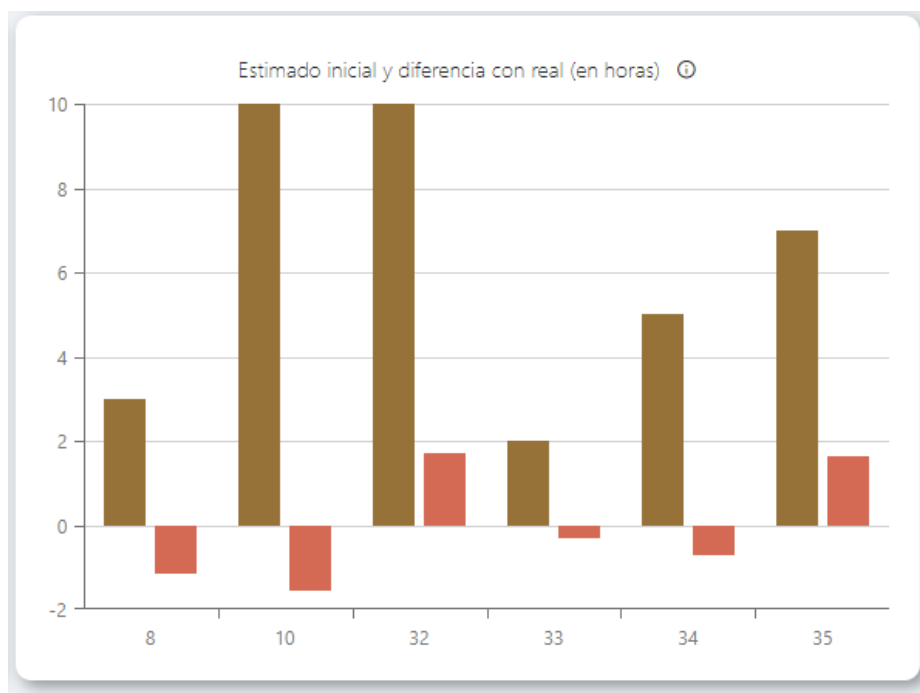


Ilustración 31 - Gráfica diferencia horas estimadas/real Sprint 1

En cuanto a la diferencia de tiempo estimado/real [Ilustración 31], vemos como ninguna UT se aleja de forma notable de su tiempo estimado, mostrándonos que las estimaciones eran bastante fieles a la realidad.

Al final del Sprint 1 ya encontramos una versión ejecutable de ErasmusEvents, una versión aún muy básica que permite a los usuarios navegar entre las diferentes pantallas, algunas de ellas aún por programar; publicar un evento, consultar los datos de nuestro perfil, consultar eventos ya publicados, etc.

5.3. Sprint 2

Empieza el sprint 2. Partiendo de la versión finalizada en el sprint 1 se continuará desarrollando nuevas funcionalidades para nuestro proyecto.

5.3.1. Estimación

Al igual que en el Sprint 1, encontramos un Sprint con una estimación de 4 semanas y un esfuerzo de 40 horas de programación. En este caso la suma de las horas estimadas de las diferentes UTs incluidas y listadas a continuación es de 43 horas.

UT	Creación	Finalización	Línea de trabajo	Sprint	Puntos	Estimación (horas)	Workflow	Proyecto
58 - Instancia unica para cada usuario	01/04/2022	07/04/2022	TFG Luis Valls	Sprint 2	20	6.5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
40 - Mostrar mapa	28/02/2022	11/04/2022	TFG Luis Valls	Sprint 2	20	5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
41 - Puntos de interes en el mapa	28/02/2022	20/04/2022	TFG Luis Valls	Sprint 2	5	5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
24 - Filtros mapa	09/02/2022	20/04/2022	TFG Luis Valls	Sprint 2	5	3	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
36 - Guardar evento	28/02/2022	21/04/2022	TFG Luis Valls	Sprint 2	40	5.5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
37 - Consultar eventos guardados	28/02/2022	26/04/2022	TFG Luis Valls	Sprint 2	20	4.5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
25 - Filtros rapidos eventos	09/02/2022	27/04/2022	TFG Luis Valls	Sprint 2	8	3.55	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
21 - Eliminar publicaciones	09/02/2022	26/04/2022	TFG Luis Valls	Sprint 2	13	2	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
11 - Consultar puntos de interes en el mapa	09/02/2022	30/04/2022	TFG Luis Valls	Sprint 2	5	3	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
9 - Editar evento	09/02/2022	01/05/2022	TFG Luis Valls	Sprint 2	8	5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls

Ilustración 32 - UTs incluidas Sprint 2



5.3.2. Realización

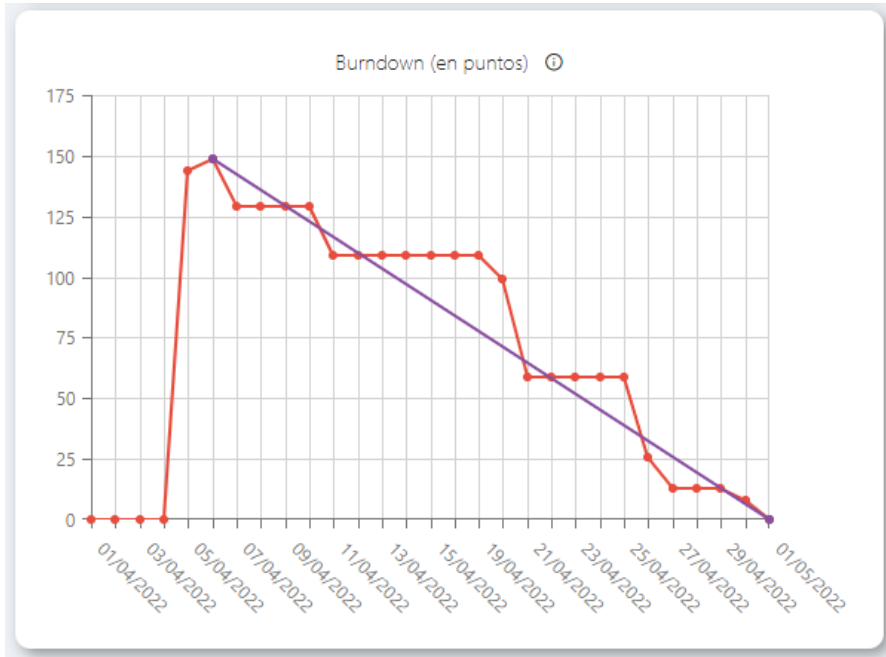


Ilustración 33 - Gráfica Burndown (en puntos) Sprint 2

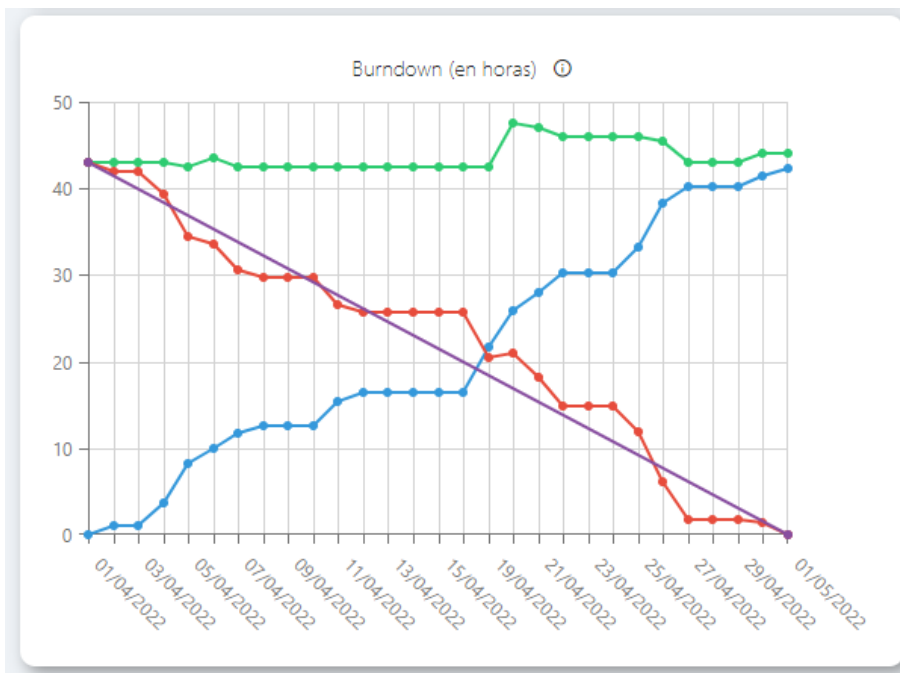


Ilustración 34 - Gráfica Burndown (en horas) Sprint 2

De las gráficas Burndown [Ilustración 33 – Ilustración 34] podemos obtener la siguiente información.

Primero observamos en la gráfica en puntos que durante los 4 primeros días no existen puntos asociados a las UTs, esto es debido a un error ya que estos no se registraron correctamente.

En cuanto al tiempo estimado vemos como se mantiene constante hasta la mitad del sprint, donde aumenta considerablemente para luego volver a bajar acercándose al estimado inicialmente.

El esfuerzo en horas fue constante a lo largo de las diferentes semanas de desarrollo.

5.3.3. Finalización

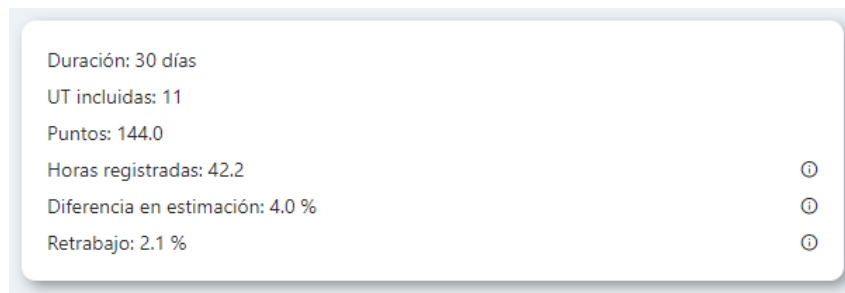


Ilustración 35 - Detalles Sprint 2

El segundo Sprint tuvo una duración de 30 días con un total de 42,2 horas registradas, superando el tiempo estimado del sprint en 2,2 horas y teniendo una diferencia de la estimación inicial del 4%.

En este sprint encontramos un 2.1% de retrabajo.

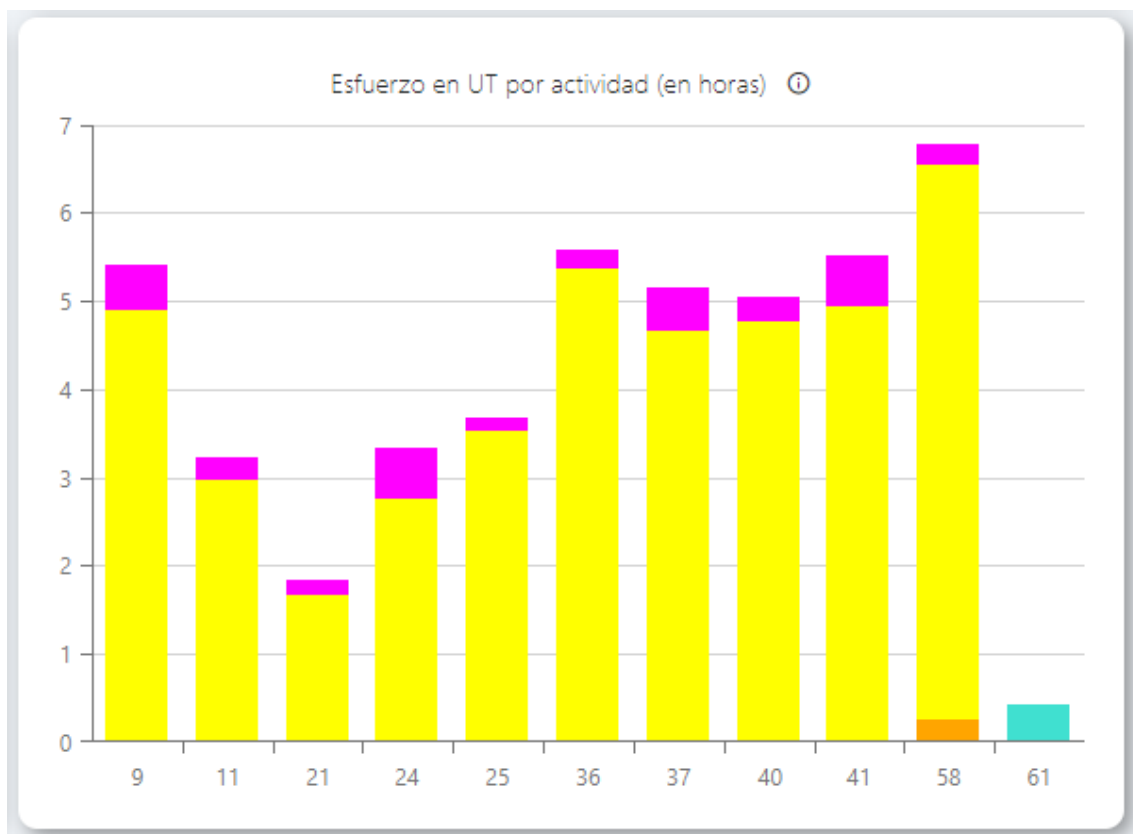


Ilustración 36 - Gráfica Esfuerzo en UT por actividad Sprint 2

En cuanto al esfuerzo en UTs por actividad [Ilustración 36], vemos una dedicación similar a la del Sprint 1.

El tiempo dedicado a pruebas ha sido una media de 25 minutos.

La barra azul corresponde al tiempo de realización de una tarea, en este caso la de preparar el sprint 3, ya que era necesario una nueva especificación de requisitos para un par de UTs.

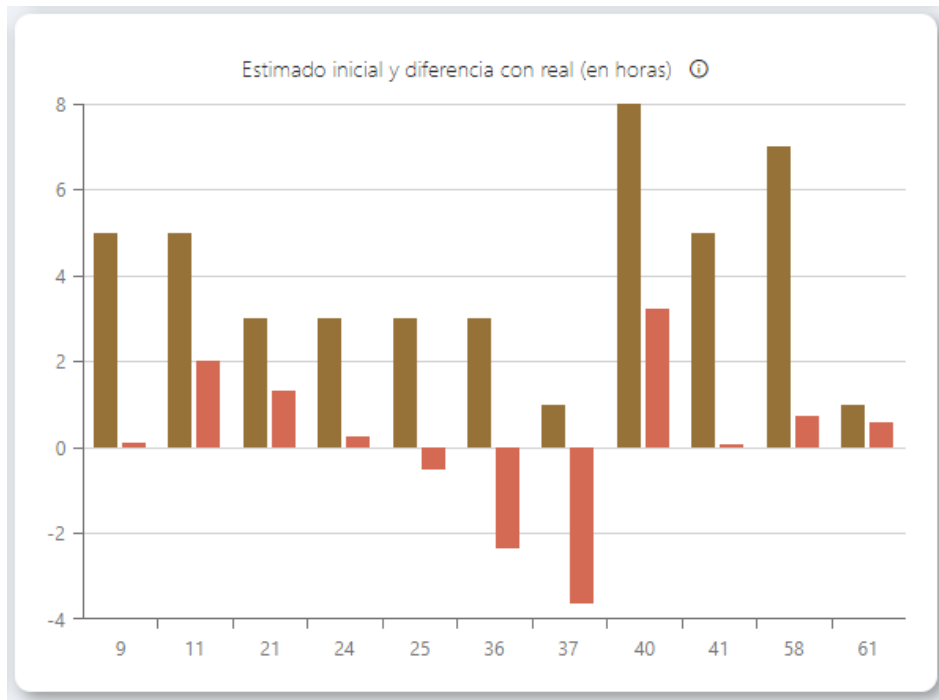


Ilustración 37 - Gráfica tiempo estimado/real Sprint 2

Observando la gráfica de diferencia entre el tiempo estimado inicial y el real [Ilustración 37] observamos grandes diferencias. Vemos como exceptuando 6 UTs cuya diferencia de tiempo real no excede en gran medida la estimada, el resto si tienen grandes diferencias de tiempos. El caso más notable sería el de la UT “consultar eventos guardados”, la cual tenía una estimación inicial de 1 hora y finalmente fueron necesarias 4h y 39 minutos para completarla.

Por suerte no fue necesario reducir el alcance del sprint ya que la mayoría de las UTs fueron completadas en menos tiempo del estimado, dejando cierta holgura para poder completar todo el trabajo dentro del umbral establecido.

Al final del sprint 2 ya encontramos más de la mitad de las funcionalidades completadas. Encontramos una versión usable bastante completa que ya permitirá completar los objetivos de nuestro proyecto de una forma general. No obstante, aún queda trabajo para obtener un software con la calidad esperada.

5.4. Sprint 3

El sprint 3 busca acabar de desarrollar todas las funcionalidades esenciales de ErasmusEvents, dejando para el siguiente y ultimo sprint el trabajo relacionado a la finalización del proyecto.

5.4.1. Estimación

De nuevo encontramos un sprint con una duración estimada de 4 semanas y con un esfuerzo de unas 40 horas de programación.

A continuación, se incluye una lista de las UTs a implementar, las cuales suman unas 45 horas de programación estimadas.

UT	Creación	Finalización	Línea de trabajo	Sprint	Puntos	Estimación (horas)	Workflow	Proyecto
60 - Bug publicar evento confoto (borra el texto)	06/04/2022	02/05/2022	TFG Luis Valls	Sprint 3	3	3	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
62 - Buscar evento	02/05/2022	12/05/2022	TFG Luis Valls	Sprint 3	8	6	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
65 - Editar Perfil Usuario	02/05/2022	09/05/2022	TFG Luis Valls	Sprint 3	5	4.5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
63 - Filtros eventos	02/05/2022	29/05/2022	TFG Luis Valls	Sprint 3	13	6	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
64 - Foto de perfil Usuario	02/05/2022	10/05/2022	TFG Luis Valls	Sprint 3	5	2.5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
67 - Modulo IA	02/05/2022	20/05/2022	TFG Luis Valls	Sprint 3	20	8	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
66 - Anuncios inApp	02/05/2022	31/05/2022	TFG Luis Valls	Sprint 3	13	8	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
14 - Acceder a las paginas de las organizaciones	09/02/2022	30/05/2022	TFG Luis Valls	Sprint 3	5	3	WF Desarrollo con Sprints	Proyecto TFG Luis Valls

Ilustración 38 - UTs incluidas Sprint 3

5.4.2. Realización

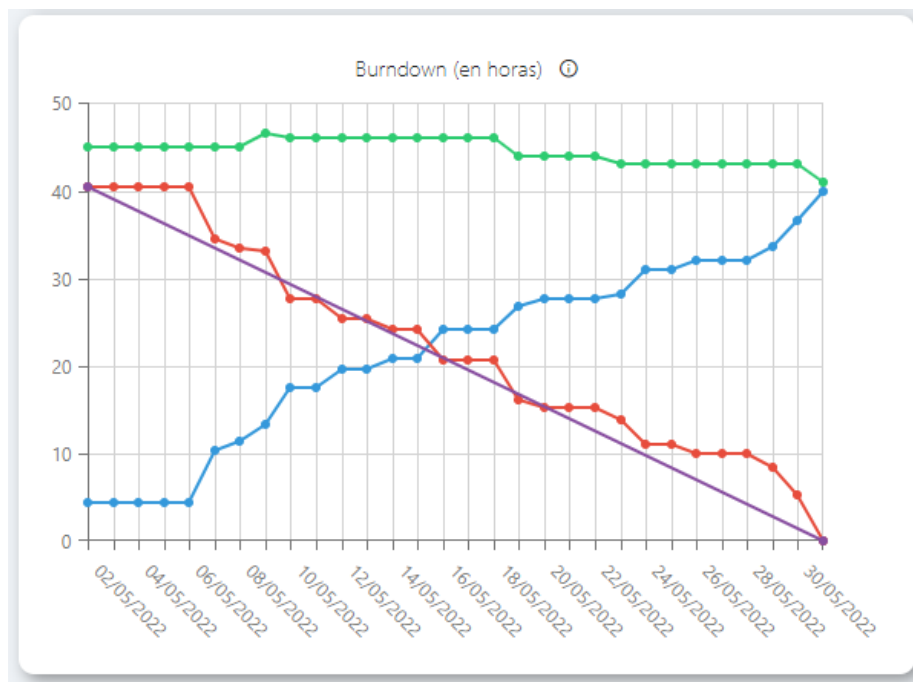


Ilustración 39 - Burndown (en horas) Sprint 3

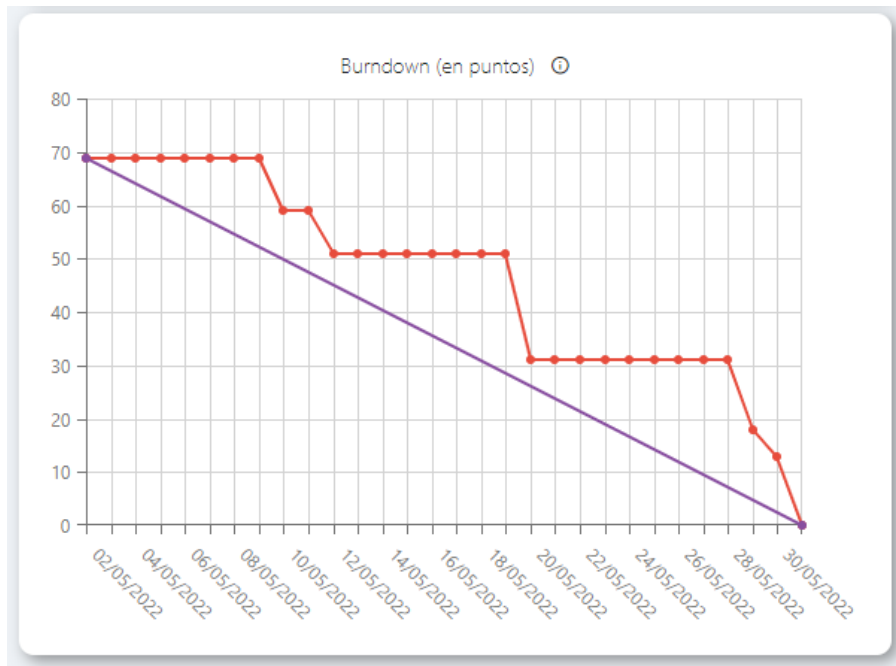


Ilustración 40 - Burndown (en puntos) Sprint 3

Las gráficas Burndown [Ilustración 39 – Ilustración 40] nos muestran como en este sprint el trabajo ha sido muy constante a la vez que la estimación de horas, la cual apenas aumentó durante la primera mitad del Sprint y luego disminuyó hasta equipararse a la estimación inicial. Finalmente, en los últimos días observamos una reducción de horas, quedando por debajo de la estimación final de la inicial.

5.4.3. Finalización

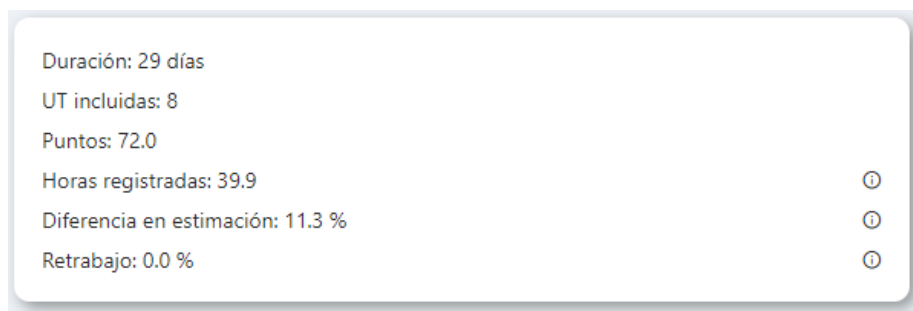


Ilustración 41 - Detalles Sprint 3

El tercer Sprint tuvo una duración de 29 días con un total de 39.9 horas registradas, ajustándose al tiempo estimado inicialmente. Esta vez encontramos una diferencia en la estimación del 11,3% y no fue necesario el retrabajo.

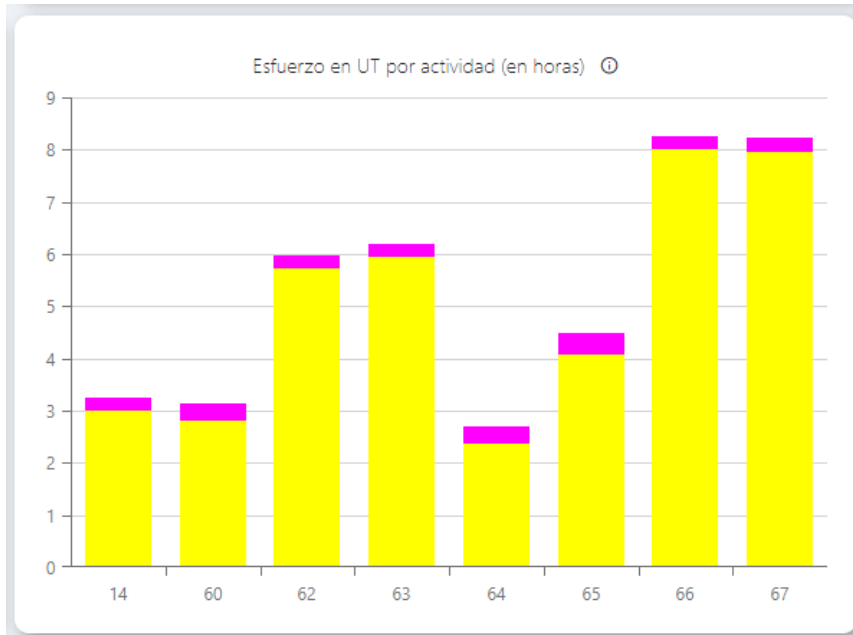


Ilustración 42 - Gráfica Esfuerzo en UT por actividad Sprint 3

En esta ocasión encontramos el tiempo dedicado a pruebas con una media de alrededor de los 15 minutos.

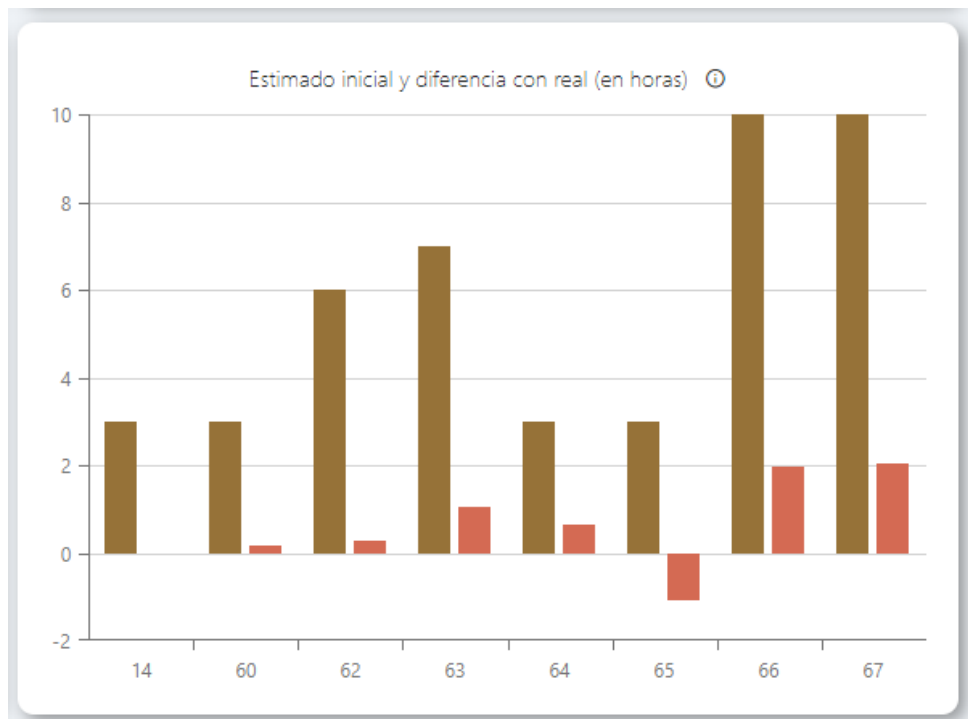


Ilustración 43 - Gráfica tiempo estimado/real Sprint 3

En cuanto a la relación tiempo estimado/real [Ilustración 43], no encontramos grandes variaciones con el tiempo estimado. Quizá podríamos destacar las dos últimas UTs en las cuales el tiempo de real de realización fue un 20% menor del estimado.

Con la finalización del Sprint 3 encontramos cubiertos los objetivos previstos para este proyecto, quedando el Sprint 4 listo para finalizar las pocas funcionalidades que entraban dentro del alcance del TFG e iniciar la fase de implantación.

5.5. Sprint 4

El cuarto y último sprint de este TFG busca llevar a cabo la fase de implantación, desplegando la App en el mercado, así como corregir una serie de errores identificados mediante pruebas y finalizar todas las funcionalidades previstas para la versión de lanzamiento de Erasmus Events.

5.5.1. Estimación

Debido a los plazos de entrega del proyecto, el sprint 4 tiene una estimación que dista del resto de los Sprints. En este caso se esperan unas 3 semanas de desarrollo y unas 40 horas de trabajo, por lo que la dedicación semanal será superior.

Las UTs mostradas a continuación son las incluidas en este Sprint final, estas UTs a excepción de la de “Cambiar ciudad” y “modificar perfil” no corresponden a nuevos requisitos, si no que corresponden bien a la corrección de errores, mejoras o a todo el trabajo necesario para finalizar el proyecto.

UT	Creación	Finalización	Línea de trabajo	Sprint	Puntos	Estimación (horas)	Workflow	Proyecto
75 - Refactoring	03/06/2022		TFG Luis Valls	Sprint 4	8	5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
76 - Registrar App Play store	03/06/2022		TFG Luis Valls	Sprint 4	13	5	WF Tarea	Proyecto TFG Luis Valls
77 - Lanzamiento de pruebas con usuarios	03/06/2022		TFG Luis Valls	Sprint 4	13	7	WF Tarea	Proyecto TFG Luis Valls
78 - Cambiar ciudad	03/06/2022		TFG Luis Valls	Sprint 4	40	10	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
79 - Bug publicar evento con foto	03/06/2022		TFG Luis Valls	Sprint 4	5	5	WF Tarea	Proyecto TFG Luis Valls
80 - Bug foto de perfil	03/06/2022		TFG Luis Valls	Sprint 4		5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
19 - Modificar perfil	09/02/2022	06/06/2022	TFG Luis Valls	Sprint 4	8	5.5	WF Desarrollo con Sprints	Proyecto TFG Luis Valls

Ilustración 44 - UTs incluidas Sprint 4

5.5.2. Realización

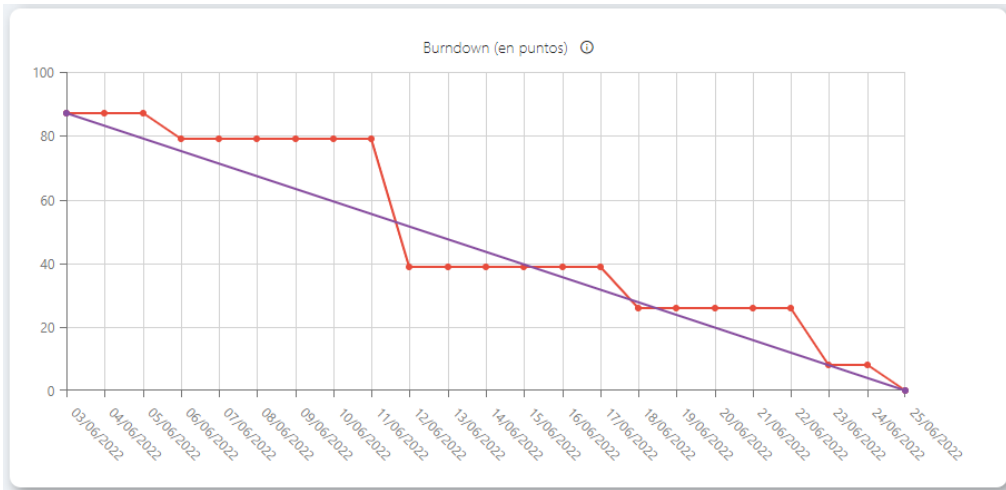


Ilustración 45 - Gráfica Burndown (en puntos) Sprint 4

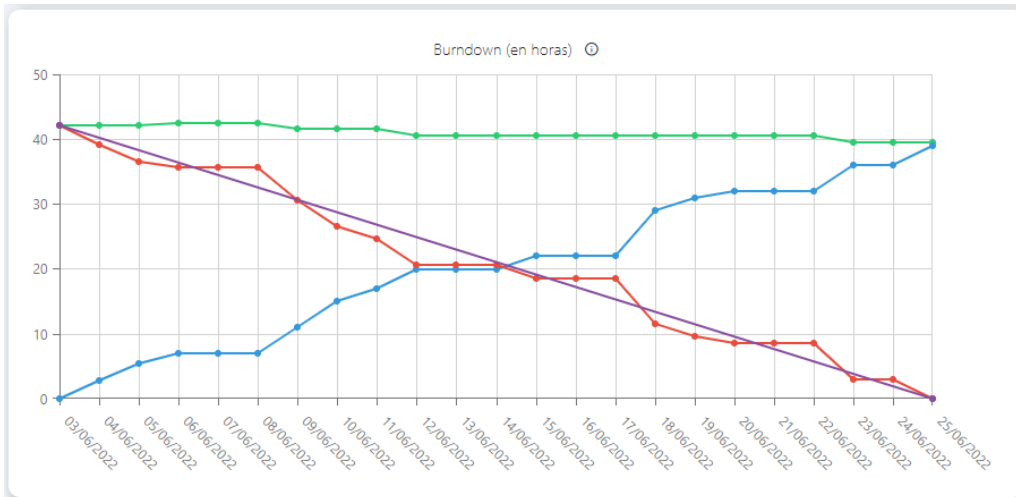


Ilustración 46 - Gráfica Burndown (en horas) Sprint 4

Poco cabe destacar observando ambas graficas Burndown [Ilustración 45- Ilustración 46]. El trabajo durante este Sprint final fue constante y la estimación de horas siguió una tendencia a la baja poco pronunciada, habiendo poca variación en las horas estimadas. Esto es debido al conocimiento previo en la realización de estas tareas, ya se conocía de antemano el tiempo que sería necesario para completar las principales tareas.

5.5.3. Finalización

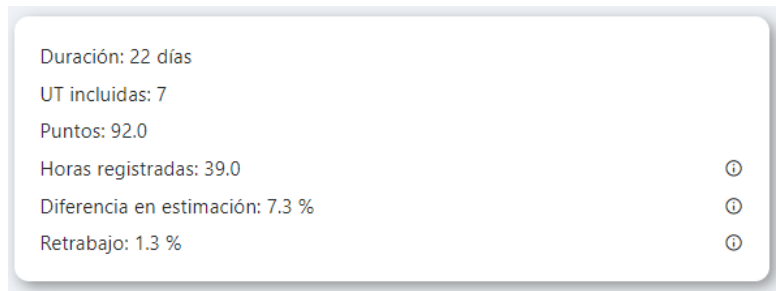


Ilustración 47 - Detalles Sprint 4

El Sprint 4 tuvo una duración real de 22 días y 39 horas de esfuerzo. La diferencia con la estimación inicial fue del 7.3% y el retrabajo correspondió al 1,3% del esfuerzo.

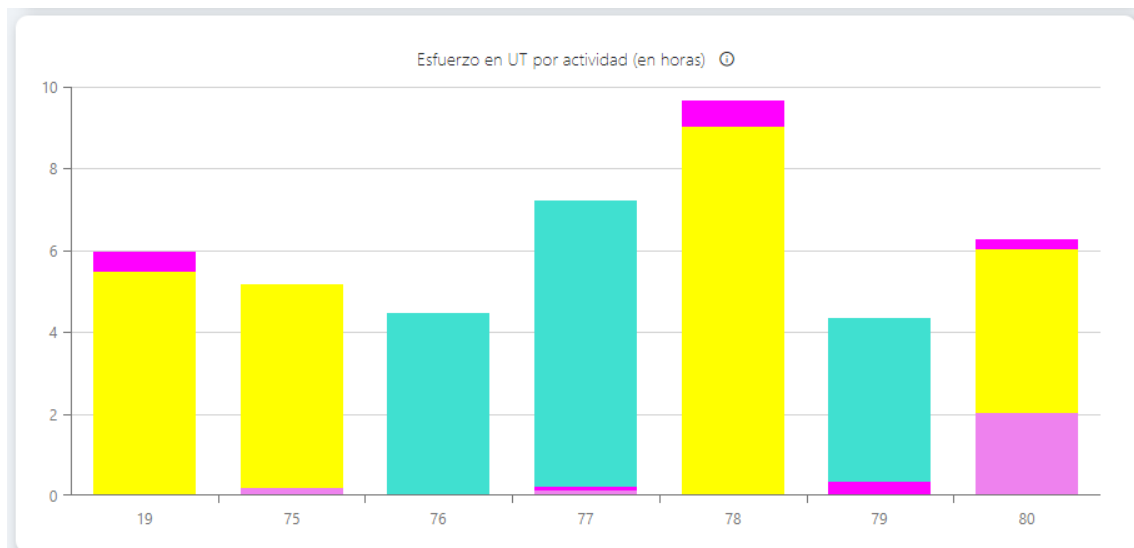


Ilustración 48 - Gráfica Esfuerzo en UT por actividad Sprint 4

En este caso, observando la gráfica esfuerzo en UT por actividad [Ilustración 48], vemos como el tiempo dedicado a pruebas de aceptación es más reducido que en Sprints anteriores, debido a el tipo de funcionalidades que se implementaban, a la vez que las tareas (barras azules) o la UT “Refactoring” no cuentan con pruebas de aceptación asociadas.

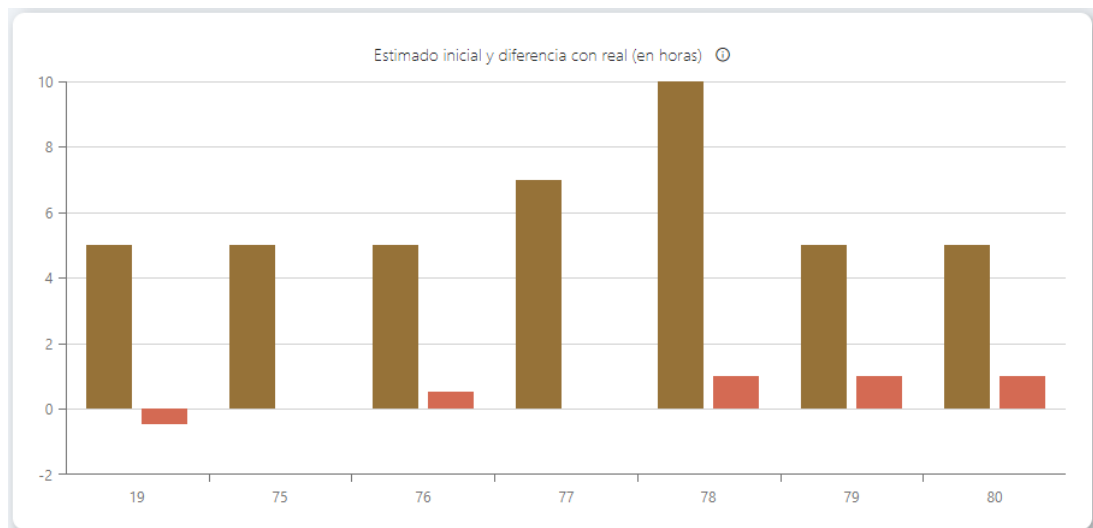


Ilustración 49 - Gráfica tiempo estimado/real Sprint 4

Finalmente vemos que no hay grandes variaciones entre el tiempo estimado inicial con el real [Ilustración 49], siendo el real menor en casi todos los casos.

Una vez concluido el Sprint 4, ya podemos encontrar todos los objetivos de este proyecto conseguidos. Erasmus Events ya se encuentra disponible en el mercado de aplicaciones y está lista para recibir usuarios.

6. Implantación

Llegamos a la fase de implantación. Esta fase se alcanzó durante el transcurso del Sprint 4, existiendo una UT asociada a todo el proceso de publicación y lanzamiento de la App en la PlayStore de Google.

Antes de su lanzamiento como beta abierta disponible para todo el mundo se lanzó una versión de Prueba interna en la cual un grupo de 10 usuarios seleccionados por su conocimiento en aplicaciones dieron un primer vistazo a la versión de lanzamiento de Erasmus Events.

El resultado de estas pruebas internas junto a los resultados de las pruebas abiertas se discutirá en el siguiente apartado (Apartado 7. Pruebas).

En este apartado se mostrará la versión de lanzamiento de ErasmusEvents junto a la explicación del proceso de lanzamiento en la Play Store.

6.1. Versión de lanzamiento Erasmus Events

A continuación, se van a mostrar capturas de pantalla de la versión que ha sido lanzada a la PlayStore con el fin de observar el resultado de este proyecto.

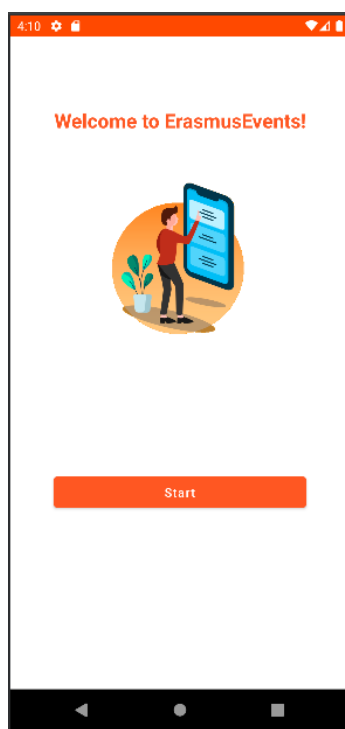


Ilustración 50 - Captura ventana de inicio

Al entrar en la aplicación encontraremos este menú de inicio [Ilustración 50] el cual cuenta con el botón “Start” que al pulsarlo nos abrirá el formulario de inicio de sesión. Este formulario cuenta con funcionalidades como el autocompletar cuenta de nuestro teléfono o el SmartLock [18], funcionalidad que permite gestionar nuestras cuentas por lo que una vez iniciada sesión no será necesario volver a iniciarla a no ser que esta se cierre.

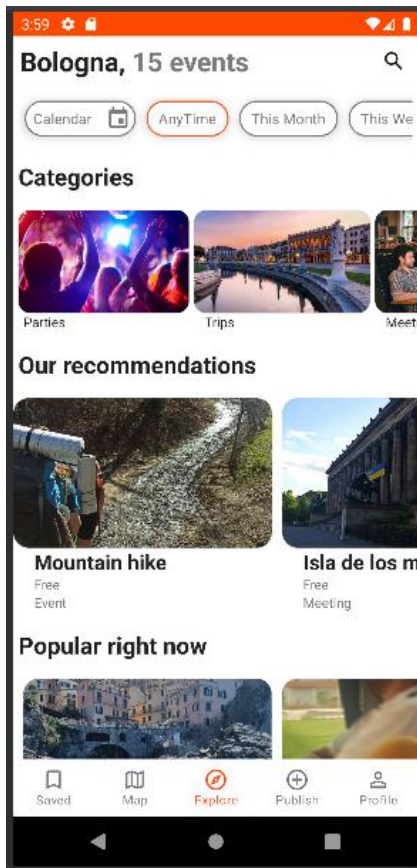


Ilustración 51 - Captura ventana explorar

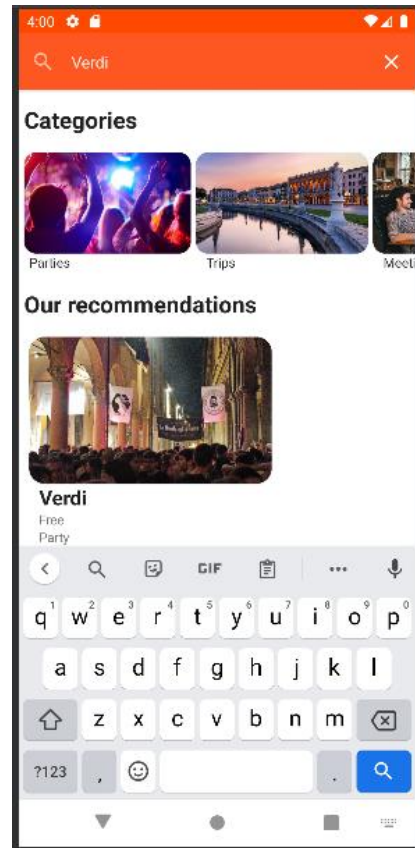


Ilustración 52 - Captura ventana explorar al buscar

Una vez iniciada la sesión encontraremos la ventana “Explore” [Ilustración 51], donde encontramos eventos publicados por los usuarios y algunos filtros rápidos por fechas. También tenemos una sección de categorías donde poder elegir un tipo de eventos en específico.

En esta ventana también disponemos de un buscador [Ilustración 52], que nos permitirá buscar más rápidamente un evento específico.

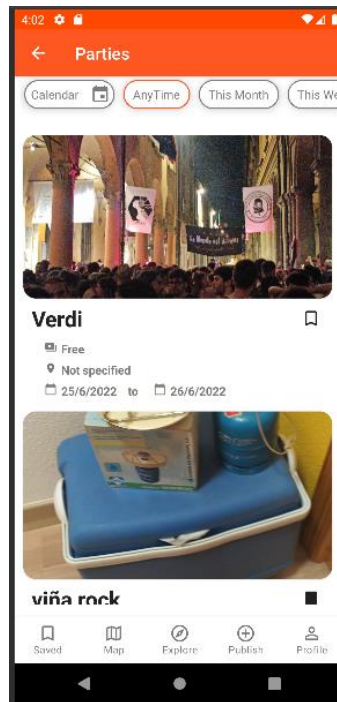


Ilustración 53 - Captura categoría "Party" de eventos

Si accedemos a una categoría específica encontraremos los eventos que cumplan con la misma ordenados en forma de lista [Ilustración 53]. En esta distribución podemos observar más características de los eventos, así como la posibilidad de guardarlos directamente haciendo click sobre el icono de guardado. De nuevo contamos con filtros rápidos por fechas en la parte superior.

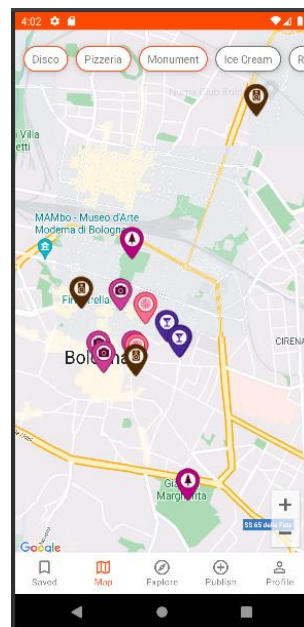


Ilustración 54 - Captura ventana mapa

En la ventana "Map" [Ilustración 54], encontramos un mapa de la ciudad seleccionada donde los marcadores indican puntos de interés. Estos marcadores están clasificados por categorías que podemos modificar a nuestro gusto en la parte superior.

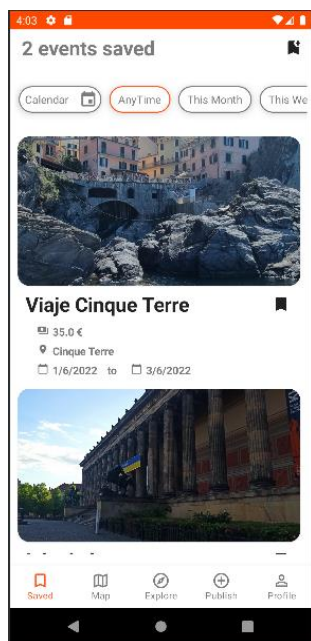


Ilustración 55 - captura ventana eventos guardados

Si accedemos a la ventana “Saved” [Ilustración 55], encontraremos los eventos que han sido guardados, pudiendo acceder a ellos o eliminarlos de nuestra lista. Una vez más disponemos de filtros rápidos para clasificar estos eventos por fechas.

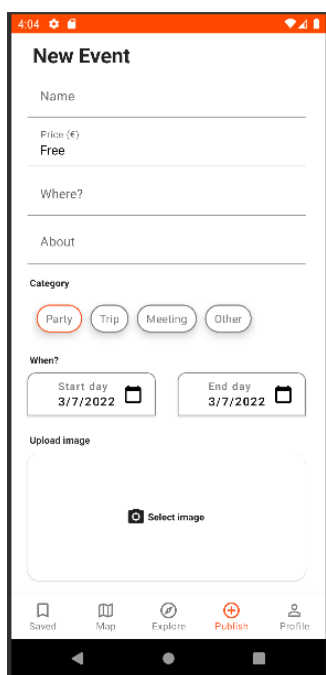


Ilustración 58 - Captura ventana publicar evento

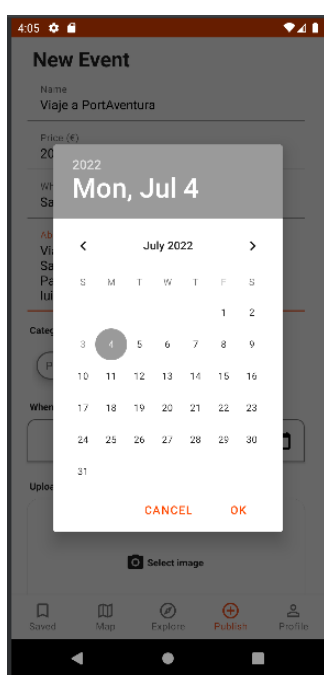


Ilustración 57 - Captura ventana publicar evento, selector de fecha

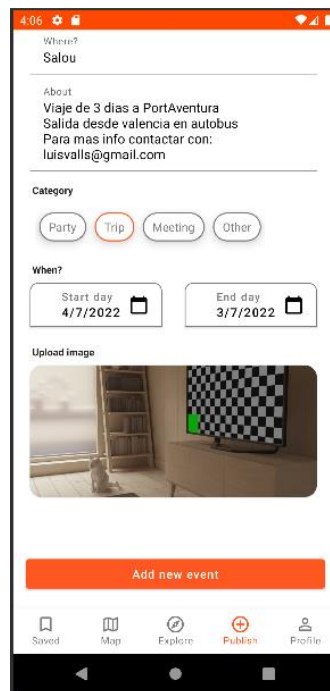


Ilustración 56 - Captura ventana publicar evento con foto

Una de las principales funcionalidades de Erasmus Events es la de publicar eventos. Al acceder a la ventana “Publish” [Ilustración 57], encontramos el formulario para publicar nuestro evento. El único campo obligatorio es el del nombre, el resto pueden no ser introducidos.

Para seleccionar la fecha se ha creado un DatePicker [Ilustración 56], para que el proceso sea más sencillo para los usuarios.

También existe la posibilidad de publicar un evento con una imagen, para esto los usuarios pueden seleccionar la que deseen desde su galería [Ilustración 58].

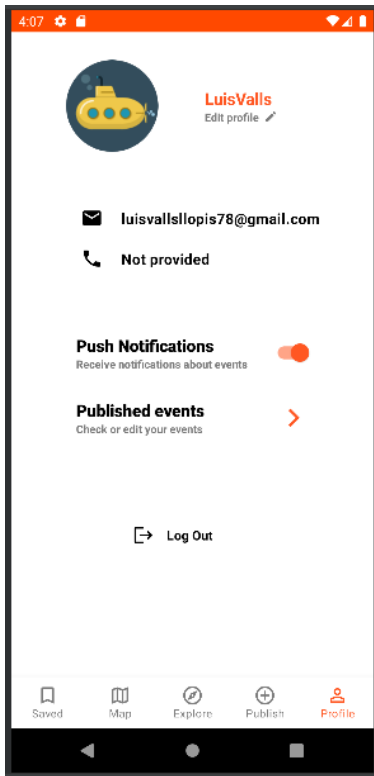


Ilustración 61 - Captura ventana Perfil

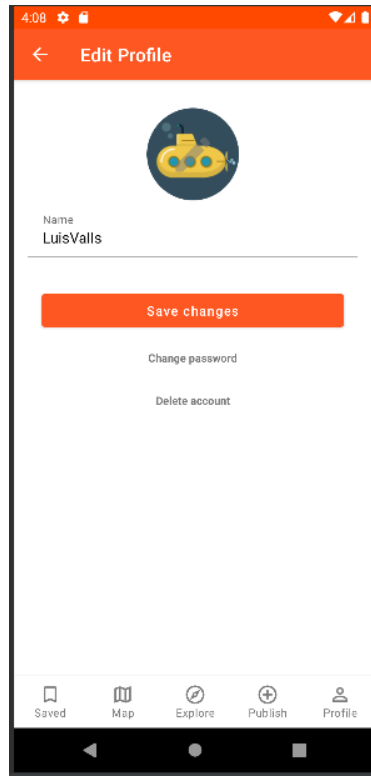


Ilustración 60 - Captura ventana editar perfil

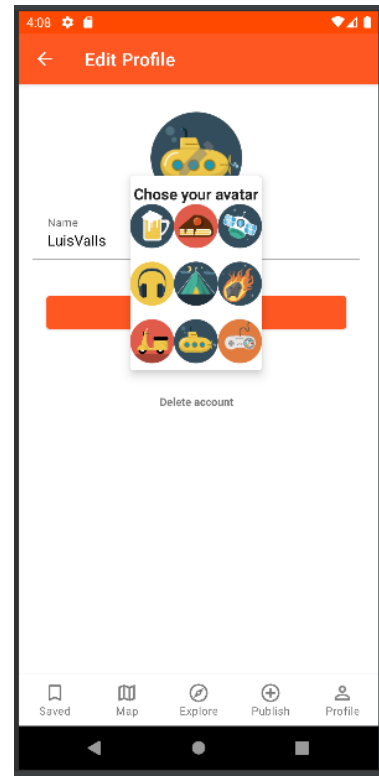


Ilustración 59 - Captura ventana editar perfil al modificar el avatar

Al acceder a la ventana “Profile” [Ilustración 61], encontramos los datos de nuestro perfil proporcionados, en este ejemplo vemos como el usuario no ha querido compartir su numero de teléfono. Desde esta ventana podemos realizar diferentes acciones como modificar nuestro perfil [Ilustración 60] cambiando nuestro nombre, contraseña o avatar [Ilustración 59], así como eliminar nuestra cuenta de forma permanente. También podemos activar o desactivar las notificaciones, acceder a nuestros eventos publicados o cerrar nuestra sesión.

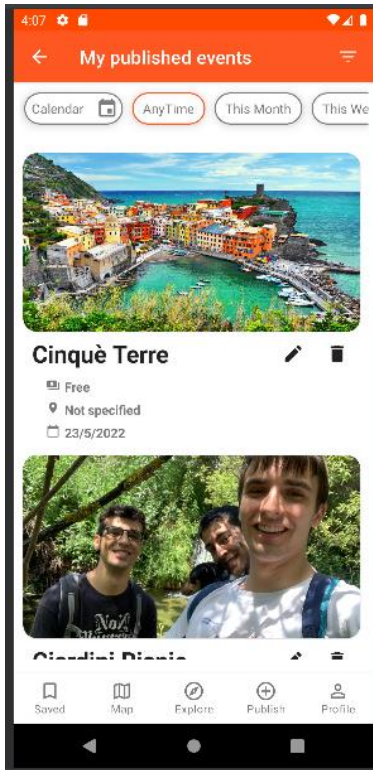


Ilustración 64 - Captura ventana mis eventos

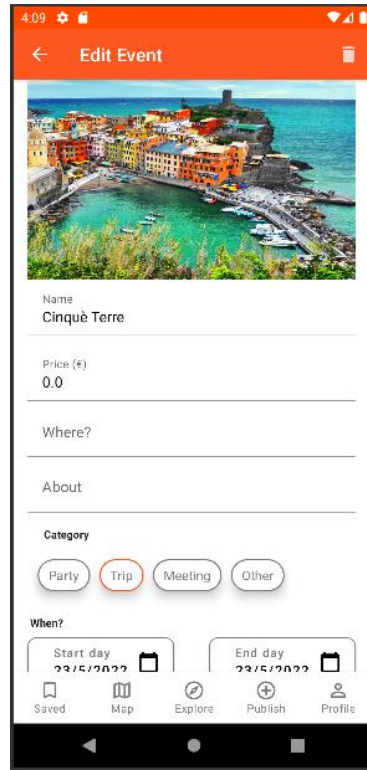


Ilustración 62 - Captura ventana editar evento

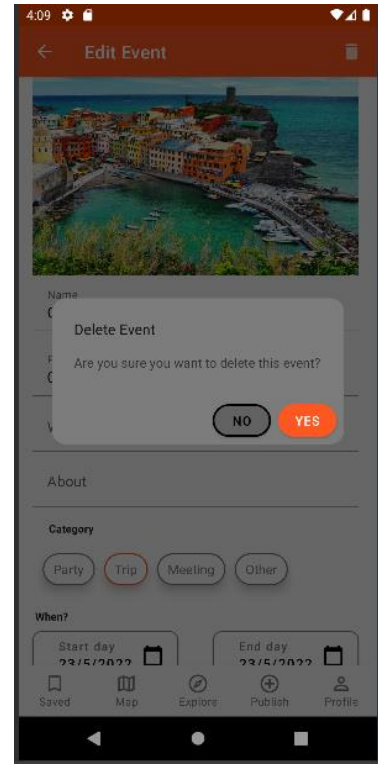


Ilustración 63 - Captura ventana editar evento al eliminar un evento

Al acceder a nuestros eventos publicados observaremos una lista de estos [Ilustración 64], donde podemos consultarlos, editarlos o eliminarlos.

Si deseamos editarlos podemos modificar todos sus valores [Ilustración 63] y luego decidir si guardar los cambios o no.

Si deseamos eliminarlo nos saldrá un dialogo de confirmación donde al pulsar en “yes” el evento será eliminado de la base de datos.

6.2. Lanzamiento en la Play Store

Lanzar una aplicación a producción en la Play Store de Google no es una tarea sencilla, pero si indispensable si se busca llegar a los usuarios del sistema operativo Android, ya que este Marketplace es el más popular de la plataforma abarcando casi el 100% de la cuota de mercado.

Para alcanzar la aprobación por parte de Google se necesita cumplir una serie de requerimientos y realizar las acciones a continuación enumeradas:

1. Establecer una política de privacidad
2. Configurar los permisos de la aplicación
3. Rellenar la información referente a los anuncios in-App
4. Indicar la clasificación de contenido
5. Informar de la audiencia objetivo
6. Rellenar un informe sobre la seguridad de los datos
7. Proporcionar datos de contacto

8. Configurar la Ficha de Play Store
9. Seleccionar Países y regiones de lanzamiento
10. Revisar y lanzar la versión

Una vez estas acciones realizadas se inicia el proceso de verificación de la App por parte de Google, el cual ofrece un veredicto al finalizar indicando si puede ser publicada o no.

El resultado de esta verificación dio el visto bueno para lanzar la versión interna el 12 de mayo, permitiendo al grupo de usuarios seleccionados descargar la App y realizar pruebas.

7. Pruebas

Esta fase tiene un objetivo muy importante en el desarrollo de nuestro producto software y es el de comprobar que el código escrito funcione como se espera. Es por esto que la fase de pruebas ha estado presente durante todo el desarrollo de Erasmus Events, cobrando protagonismo en las fases finales del desarrollo con el lanzamiento de la aplicación.

Es importante recordar que existen dos tipos de pruebas, las manuales y las automatizadas y estas a su vez pueden variar en cuanto a complejidad y tamaño.

Aunque las pruebas automatizadas son mucho más potentes y fiables que las manuales, exigen un gran esfuerzo para ser desarrolladas y debido a la falta de tiempo durante el desarrollo de nuestro proyecto se ha optado por utilizar un conjunto de pruebas manuales para verificar nuestro software.

Los diferentes tipos de pruebas utilizadas en Erasmus Events han sido:

7.1. Pruebas unitarias.

Las pruebas unitarias consisten en probar fragmentos muy pequeños dentro de nuestro código, fragmentos como métodos o funciones individuales.

En ErasmusEvents todos los métodos han sido testeados cuando han sido escritos, utilizando técnicas de caja blanca, donde con un conjunto de valores de entrada corroboramos que realizan la función esperada de ellos.

7.2 Pruebas de integración

Las pruebas de integración buscan verificar que las distintas partes de nuestra aplicación funcionan bien en conjunto, ya que pueden tener un resultado correcto en las pruebas unitarias, pero crear problemas al utilizarse de forma concurrente con otros métodos.

En Erasmus Events se han realizado pruebas de integración cada vez que una versión era lanzada, es decir, al final de cada mes, teniendo especial atención al comportamiento de la base de datos, ya que era el “eslabón más débil” de cara a múltiples usuarios interaccionando con ella a la vez. Por ejemplo, al programar la funcionalidad de subir un evento con foto, se llevaron a cabo múltiples pruebas de integración para comprobar que no teníamos ningún problema al subir un evento, ya fuera uno con imagen o sin imagen.

7.3 Pruebas funcionales

Las pruebas funcionales están enfocadas en observar si se cumplen los requisitos empresariales de un producto software. Solo verifican el resultado obtenido al realizar una acción ignorando los estados intermedios del sistema mientras esa acción se realiza.



A diferencia de las pruebas de integración, que por ejemplo podrían centrarse en verificar que funciona la base de datos, en una prueba funcional se espera un valor específico dictado por los requisitos del producto para dar por buena la prueba.

Este tipo de pruebas se han utilizado durante el desarrollo de nuestra App para verificar el correcto comportamiento de la base de datos. Un ejemplo de estas pruebas era cuando tras algún cambio en alguna clase que interaccione con la base de datos, se subía un evento y se comprobaba que los valores en la base de datos se correspondían con los introducidos.

7.4 Pruebas integrales

Las pruebas integrales están pensadas para replicar el comportamiento de un usuario en un entorno completo. Verifican diferentes flujos de ejecución observando que funcionen según lo esperado.

Estas pruebas han sido realizadas al final de cada sprint, simulando acciones de usuarios al entrar a la aplicación. Han sido útiles ya que han permitido observar cómo al realizar acciones de una determinada forma aparecían errores, errores que han sido etiquetados y corregidos en Sprints posteriores. Un ejemplo sería el error de subir un evento con foto, donde gracias a estas pruebas se descubrió que si se seleccionaba la foto para el evento antes de completar el resto de los campos se producía una excepción de puntero nulo. Este error se pudo corregir y ya se encuentra solucionado.

7.5. Pruebas de aceptación

Las pruebas de aceptación nos permiten saber si nuestro producto satisface los requisitos de cada UT. Estas pruebas han sido registradas antes de realizar cada UT y nos dicen que debe realizar nuestro sistema una vez esta finalice la etapa de programación.

Estas pruebas han sido registradas en Worki, y cada vez que una Ut llegaba a la etapa de Pruebas, estas se ejecutaban y se observaba si el resultado era optimo. De no superar estas verificaciones, la Ut volvía a la etapa de programación para solucionar los posibles fallos.

Estas pruebas estaban divididas en tres grupos referenciando a los 3 tipos de UTs en función de que operaciones realicen. Estos grupos son Navegación, Organización y estudiante.

A continuación, se muestran algunas de estas pruebas, podemos observar el día que fueron ejecutadas y las UTs a las que corresponden.

Nombre PA	Orden	Última modificación de PA			
		Fecha modificación	UT	Sprint	Colaborador
1 - La barra de navegación se muestra correctamente		2/3/2022, 14:22	33	Sprint 1	
2 - Podemos navegar correctamente entre ventanas		2/3/2022, 14:23	33	Sprint 1	
3 - La transición es estable		2/3/2022, 14:23	33	Sprint 1	
5 - Los iconos cambian de color		2/3/2022, 14:25	33	Sprint 1	
20 - El diseño es fiel a los MockUps		31/3/2022, 12:18	34	Sprint 1	
21 - Al clicar en un evento se muestran los detalles del mismo		31/3/2022, 12:18	34	Sprint 1	
22 - La foto que aparece es la del evento		31/3/2022, 12:19	34	Sprint 1	
23 - Eventos sin foto aparece una genérica de la categoría		31/3/2022, 12:19	34	Sprint 1	
24 - EL botón de seguir es clickable		31/3/2022, 12:19	34	Sprint 1	
25 - Para cada usuario se muestran sus eventos		7/4/2022, 14:37	58	Sprint 2	
26 - Cada usuario tiene su perfil individual		7/4/2022, 14:37	58	Sprint 2	

Ilustración 65 - Ejemplos Pruebas Aceptación

Este tipo de pruebas están relacionadas con las Pruebas de Regresión, las cuales fueron ejecutadas al final de cada sprint y consisten en ejecutar todas las pruebas de aceptación previamente superadas para verificar que el sistema sigue funcionando correctamente.

7.6. Pruebas de rendimiento.

Como bien su nombre indica, estas pruebas buscan evaluar el rendimiento de nuestro sistema con una carga de trabajo predeterminada. Estas pruebas son útiles para medir la fiabilidad, velocidad, escalabilidad y capacidad de respuesta de nuestro producto software.

Para la realización de estas pruebas en ErasmusEvents se ha hecho uso de la herramienta Android Profiler, que nos permite ejecutar la aplicación y obtener valores de carga sobre dispositivos de referencia.

Los dispositivos medidos han sido el Pixel 4, con un API 30 de Android, teléfono de gama alta que nos permitirá medir el funcionamiento de la app en un dispositivo de altas prestaciones y el Nexus 5, teléfono de gama alta lanzado en 2013, que al contar con hardware antiguo nos permite observar el funcionamiento de nuestra app en un teléfono de bajas prestaciones.

Los resultados de las pruebas son los siguientes:

7.6.1 Prueba rendimiento Pixel 4:



Ilustración 66 - Resultado prueba rendimiento Pixel 4

Tras la realización de esta prueba donde se han explorado casi todas las funcionalidades de ErasmusEvents podemos concluir que el rendimiento de nuestro producto es bueno y cumple con lo esperado. Observando la gráfica de resultados vemos que en el momento de mayor carga del procesador este llega al 66% y esto corresponde a la acción de publicar un evento, acción pesada ya que debe crear el objeto, subirlo al servidor y mostrar un anuncio publicitario.

No obstante, en el resto de las acciones la carga del procesador no supera el 50% que, junto al uso de memoria, que en ningún caso supera los 500mb, y teniendo en cuenta que un dispositivo moderno cuenta con 6 o 8 GB de memoria; nos indica que nuestra aplicación tiene un rendimiento muy y funcionará sin problemas en teléfonos modernos.

7.6.2 Prueba rendimiento Nexus 5:

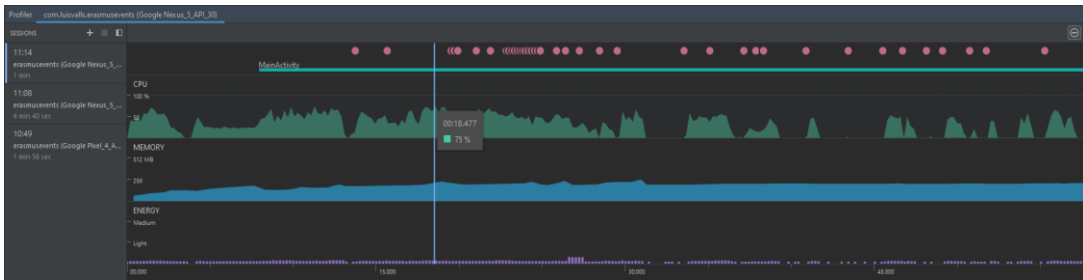


Ilustración 67 - Prueba rendimiento Nexus 5

En el caso del Nexus5, al tratarse de un dispositivo antiguo observamos como la carga del procesador es mayor, llegando a un pico de 75% de carga. No obstante, sigue estando por dentro de los valores aceptados igual que el uso de memoria.

Gracias a estas pruebas de rendimiento podemos concluir que ErasmusEvents tiene un rendimiento muy bueno y podrá funcionar sin problemas en cualquier tipo de dispositivo aumentando nuestro alcance de usuarios.

7.7. Pruebas de humo

Las pruebas de humo son pruebas muy básicas que permiten comprobar que la aplicación “sigue funcionando”. Están pensadas para ejecutarse rápidamente por ejemplo después de cada compilación. De esta forma al modificar algún parámetro podemos asegurarnos de que las principales funciones de nuestros sistemas siguen funcionando correctamente.

Estas pruebas han sido especialmente útiles durante el desarrollo de nuestra app, ya que al ejecutarlas tras cada compilación nos han permitido identificar errores menores que si no se hubiera detectado podrían haber sido un gran problema en pruebas posteriores.

7.9. Pruebas con usuarios

En muchas ocasiones nos encontramos el caso de un producto software bien construido, que pasa sin problema todas las pruebas de funcionamiento y sin embargo es un fracaso. Esto ocurre principalmente porque solo se han tenido en cuenta los requisitos empresariales y no el de los usuarios, al obviar esta parte, una de las principales, podemos encontrarnos con un software que no cumplirá su objetivo.

Por suerte, existen las pruebas con usuarios, pruebas que buscan el contacto directo con usuarios reales que probarán la aplicación instalándola en sus propios dispositivos.

Posteriormente responderán a una encuesta dando su opinión y veredicto.

Durante la fase de implantación de la aplicación se realizó el despliegue en dos etapas. Una etapa inicial de beta interna, donde 10 usuarios cercanos pudieron instalar y probar ErasmusEvents, dando su opinión y consejos de mejora; y una segunda etapa donde la aplicación fue lanzada como beta abierta y donde cualquier usuario podía descargarla y realizar una breve encuesta ofreciendo su veredicto.

7.9.1. Etapa interna

Durante la fase interna el método de investigación fue el de observación y análisis. Debido a que los 10 usuarios seleccionados para realizar esta prueba eran personas cercanas, se pudo observar el grado de eficacia, eficiencia y satisfacción estando presente durante la realización de la prueba y recogiendo los resultados mediante unas breves entrevistas posteriores.

En general el feedback recibido fue muy positivo, todos los usuarios mostraron un gran interés por la aplicación y expresaron su agrado con la misma. Esta fase fue muy útil para identificar una serie de bugs y fallos existentes, fallos ya que fueron rápidamente corregidos. A su vez también sirvieron para realizar una serie de mejoras en la interfaz gráfica, ya que gracias a sus opiniones se modificaron una serie de elementos visuales para mejorar la experiencia de uso.



7.9.2. Etapa abierta

Una vez finalizada la etapa interna, se lanzó una versión beta abierta, donde cualquier persona podía descargar la aplicación y probarla.

Se facilitó un breve formulario de Google donde los usuarios podían responder 5 sencillas preguntas mostrando su nivel de agrado en una escala del 1 al 5. Estas 5 métricas fueron la de rapidez, facilidad de uso, interfaz gráfica, satisfacción y opinión general.

En total se recibió un total de 24 respuestas, el resultado de estas se muestra a continuación:

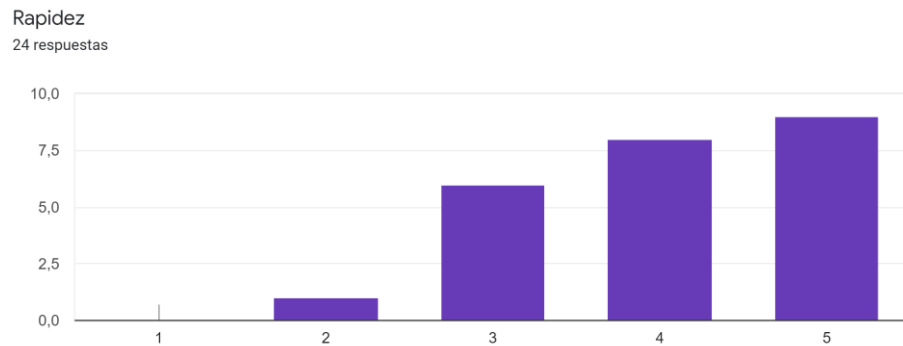


Ilustración 68 - Respuesta encuesta usuarios – Rapidez

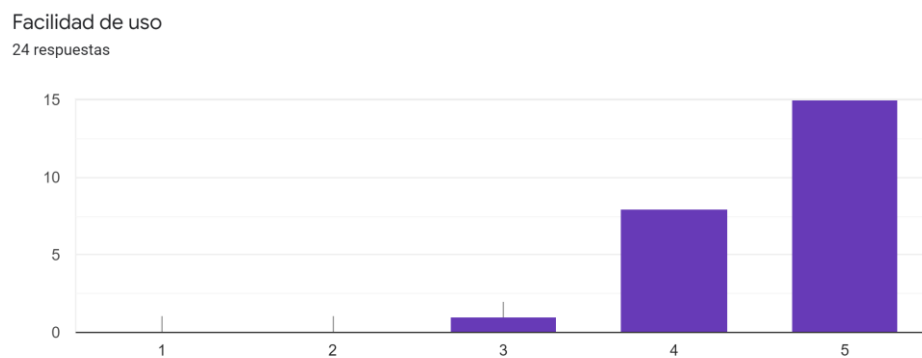


Ilustración 69 - Respuesta encuesta usuarios - Facilidad de uso

Interfaz Grafica

24 respuestas

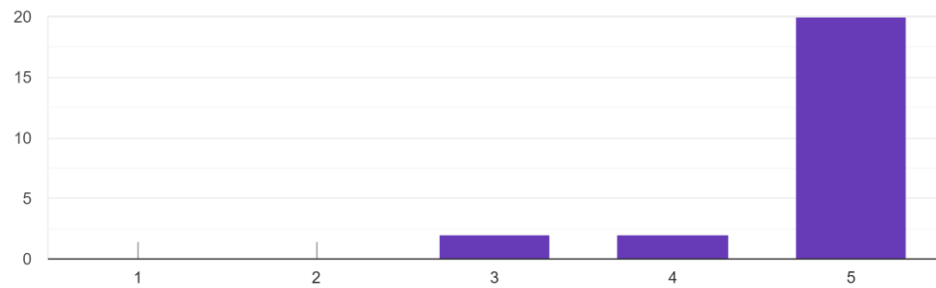


Ilustración 70 - Respuesta encuesta usuarios - Interfaz Gráfica

Satisfacción

24 respuestas

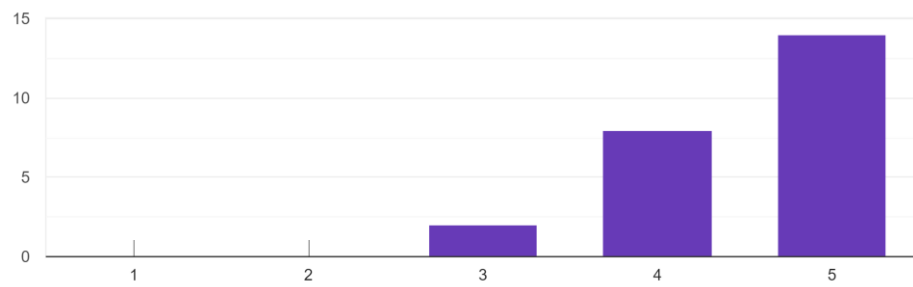


Ilustración 71 - Respuesta encuesta usuarios - Satisfacción

Opinión general

24 respuestas

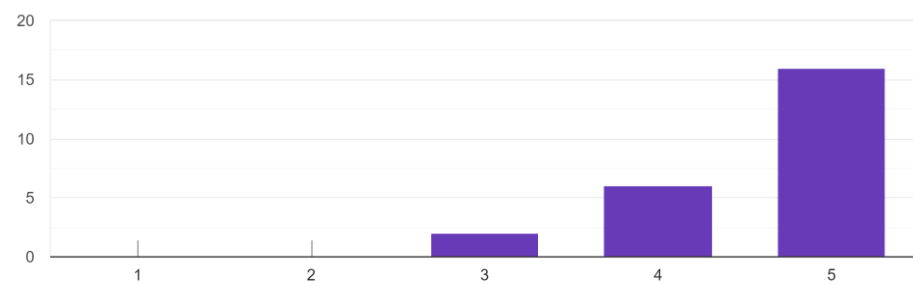


Ilustración 72 - Respuesta encuesta usuarios - Opinión general

Estas pruebas han tenido un gran impacto en la verificación del software ya que además de su principal cometido, también han permitido medir el rendimiento del sistema ofrece en un gran número de dispositivos, obtenido estos valores gracias a las estadísticas que la consola de Google y obtener pruebas integrales reales, ya que los usuarios no han sido guiados para realizar estas pruebas.

Analizando los resultados vemos como la percepción general de los usuarios ha sido excepcional. El único campo que muestra una alerta de cara a su refinamiento futuro es el de rendimiento, y es que en las pruebas de rendimiento ya pudimos observar cómo en ciertas acciones se producen picos de carga del procesador. Estos picos corresponden a la ejecución de métodos muy pesados, quedando como trabajo futuro la optimización de estos.

7.8. Automatización de las pruebas

La parte más complicada de testear mediante pruebas automáticas de una aplicación Android son las pruebas de la interfaz de usuario, no obstante, se pueden realizar sin un verificador humano, ya que esta tarea es automatizable gracias a un conjunto de herramientas disponibles en el mercado que imitan patrones de comportamiento humanos, bien realizando las acciones que se esperarían por un usuario, como realizando otras aleatorias para verificar el correcto funcionamiento del sistema.

No obstante, estas pruebas requieren de un gran esfuerzo inicial para programarlas y lanzarlas, por lo que al no contar con los recursos necesarios esta automatización ha sido aplazada quedando como un trabajo futuro.

8. Conclusiones

Una vez finalizada y lanzada al mercado la versión inicial de ErasmusEvents podemos concluir este proyecto de fin de grado. Esto es así ya que se han alcanzado los objetivos propuestos.

El primer objetivo queda logrado al quedar la aplicación publicada, ya que los estudiantes erasmus del curso 2022-2023 ya podrían hacer uso de esta, facilitando sus estancias.

El segundo objetivo era el de demostrar que como graduado en ingeniería informática era un profesional apto para incorporarme al mercado laboral teniendo el conocimiento suficiente como para desarrollar productos software completos y de calidad. Este objetivo queda cumplido al ofrecer una aplicación al mercado completa y que ha superado todas las pruebas, incluyendo las de rendimiento y las con usuarios. Estas últimas nos muestran como el trabajo cumple las expectativas de los usuarios reales.

Por último, el tercer objetivo queda más que cumplido al cumplirse los dos anteriores. Durante estos 5 meses de desarrollo se ha investigado muchísimo sobre estas nuevas tecnologías, profundizando en cómo realizar acciones de la forma más correcta posible, por lo que actualmente puedo afirmar que dispongo de un conocimiento suficientemente avanzado en este campo.

No obstante, no solo el cumplimiento de estos tres objetivos nos indica el éxito del proyecto. A nivel personal este proyecto ha tenido un fuerte impacto, ya que a sido el primer proyecto “real” en el que me he visto involucrado y profundamente implicado. Es el proyecto al que más horas le he dedicado en mi vida y el que me ha servido para reafirmar mi creencia de la ingeniería del software es la profesión idónea para mí.

9. Trabajos futuros

9.1. Versión IOS

De cara al lanzamiento y comercialización de Erasmus Events al mercado de aplicaciones móviles quedan aún un gran número de acciones por realizar.

La principal y más importante será la del desarrollo de la aplicación para el sistema operativo IOS. Y es que, analizando los datos de ventas de móviles a nivel mundial, en 2017 las ventas de dispositivos de Apple ya suponían el 18'94% del total, aumentando al 24'5% en países como Francia o incluso al 42,6% en Inglaterra (Países europeos que entran en el programa Erasmus).



Ilustración 73 - Gráfica porcentaje smartphones con IOS por países

Esta tendencia nos obliga a adaptar nuestra App a este entorno, no solo por no perder este gran número de usuarios potenciales, sino porque al limitar el alcance del software a una parte de los estudiantes, la comunidad que se espera crear puede no ser suficiente como para que se mantenga a lo largo del tiempo.

Los principales desafíos con los que nos encontraremos en esta fase serán los de empezar de nuevo la fase de Desarrollo de la solución, ya que la programación deberá ser empezada de nuevo en un entorno y lenguaje especializado, por lo que no se podrá reutilizar software. Por otra parte, gracias a ya contar con todas las fases de diseño y especificación el tiempo necesario para la obtención de una versión final lanzada al mercado será sustancialmente menor.

Se espera que de cara al segundo semestre del curso 2022-2023 se pueda contar con una versión tanto IOS como Android disponible en el mercado.

9.2. Nuevas funcionalidades

Una vez lanzada la aplicación, se busca seguir desarrollando funcionalidades y ofreciendo mejoras y contenido a los usuarios.

Se buscará seguir con el desarrollo añadiendo nuevos Sprints para incluir las UTs que quedaron en el backlog por falta de recursos para su implementación. A continuación, se muestran algunas de estas UTs sin asignar.

UT	Creación	Finalización	Línea de trabajo	Sprint	Puntos	Estimación (horas)	Workflow	Proyecto
42 - Modo oscuro disponible	02/03/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls
43 - Opción de elegir idioma	02/03/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls
59 - TextField autocompletable	06/04/2022		TFG Luis Valls	Backlog	5	3	WF Desarrollo con Sprints	Proyecto TFG Luis Valls
15 - Publicar viajes oficiales	09/02/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls
16 - Publicar eventos oficiales	09/02/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls
12 - Contactar con una organización	09/02/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls
18 - Crear perfil usuario	09/02/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls
39 - Reservar evento	28/02/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls
38 - Consultar eventos reservados	28/02/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls
17 - Enviar notificaciones	09/02/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls
20 - añadir lugar de interés	09/02/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls
13 - Consultar ofertas o descuentos	09/02/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls
22 - Restringir acceso a usuarios	09/02/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls
23 - chat global	09/02/2022		TFG Luis Valls	Backlog			WF Desarrollo con Sprints	Proyecto TFG Luis Valls

Ilustración 74 - UTs restantes en Backlog

A su vez también se buscará ampliar y añadir contenido a este Backlog con el feedback de los usuarios, pudiendo añadir funcionalidades deseadas o buscadas por parte de los estudiantes.

9.3. Automatización de tareas.

Como ya se ha comentado en el apartado de Pruebas, debido a la falta de tiempo durante el desarrollo no se consiguió automatizar las pruebas de software como se hubiera deseado, por lo que queda como trabajo pendiente de cara al futuro conseguir automatizar estas pruebas para conseguir ejecutarlas de forma más rápida, eficiente y completa cada vez que se lance una actualización con una nueva versión.

10. Referencias

- [1] José H. Canós, Patricio Letelier y M.^a Carmen Penadés, “Metodologías Ágiles en el Desarrollo de Software” (2003). [En línea]. Disponible: <https://acortar.link/o3KzwM>
- [2] Patricio Letelier, “TUNE-UP Software Process Metodología y herramienta de apoyo para la gestión ágil de proyectos de desarrollo y mantenimiento de software” (2011). [En línea]. Disponible: <https://acortar.link/FTBfQM>
- [3] Página oficial ESN Bolonia. Disponible: <https://www.esnbologna.org/>
- [4] Página en la PlayStore de “Eventbrite”. Disponible: <https://acortar.link/Y2hsyO>
- [5] Página en la PlayStore de “Fever”. Disponible: <https://acortar.link/48wtHW>
- [6] Página en la PlayStore de “Erasmus in Czech Republic”. Disponible: <https://play.google.com/store/apps/details?id=com.nanooq.esn>
- [7] Página en la PlayStore de “Erasmus Barcelona”. Disponible: https://play.google.com/store/apps/details?id=net.manageapps.app_122139
- [8] Juan Antonio Pascual, “Android vs iPhone: la guerra de los smartphones en cifras” (2018). [En línea]. Disponible: <https://computerhoy.com/reportajes/industria/android-vs-iphone-guerra-smartphones-cifras-271447>
- [9] Página oficial Android Studio. [En línea]. Disponible: <https://developer.android.com/studio>
- [10] Página oficial GitHub. [En línea]. Disponible: <https://github.com/>
- [11] Página oficial Firebase. [En línea]. Disponible: <https://firebase.google.com/>
- [12] Página oficial Figma. [En línea]. Disponible: <https://www.figma.com/>
- [13] “Qué modelos de negocio para aplicaciones móviles existen?” (2021) [En línea]. Disponible: <https://www.dimensiona.com/es/modelos-de-negocio-para-aplicaciones-moviles/>
- [14] Página oficial Dribbble. [En línea]. Disponible: <https://dribbble.com/>
- [15] Miriam Martínez, “¿Qué son los patrones de diseño de software? (2020). [En línea]. Disponible: <https://profile.es/blog/patrones-de-diseno-de-software/>
- [16] Documentación oficial Android Developers, “Inyección de dependencias en Android”. [En línea]. Disponible: [https://developer.android.com/training/dependency-injection?hl=es-419#:~:text=La%20inyecci%C3%B3n%20de%20dependencias%20\(DI,Reutilizaci%C3%B3n%20de%20oc%C3%B3digo](https://developer.android.com/training/dependency-injection?hl=es-419#:~:text=La%20inyecci%C3%B3n%20de%20dependencias%20(DI,Reutilizaci%C3%B3n%20de%20oc%C3%B3digo)
- [17] Página oficial Worki. [En línea]. Disponible: <http://www.tuneupprocess.com/>

[18] Yúbal Fernández, “Qué es Google SmartLock, cómo funciona y cómo puedes usarlo” (2019). [En línea]. Disponible: <https://www.xataka.com/basics/que-google-smart-lock-como-funciona-como-puedes-usarlo>

[19] Sten Pittet, “Los distintos tipos de pruebas de software”. [En línea]. Disponible: <https://www.atlassian.com/es/continuous-delivery/software-testing/types-of-software-testing>

[20] Milona González, “Análisis DAFO: Guia” (2022). [En línea]. Disponible: <https://aulacm.com/analisis-dafo-ejemplo-plantilla/>

[21] Javier Megias, “Lean Canvas, un lienzo de modelo de negocio para startups” (2012). [En línea]. Disponible: <https://javiermegias.com/blog/2012/10/lean-canvas-lienzo-de-modelos-de-negocio-para-startups-emprendedores/>

[22] Jaime P. Llasera, “Mockup: qué es y para qué se usa en diseño gráfico” (2021). [En línea]. Disponible <https://imborrable.com/blog/mockup-que-es/>

11. Anexo

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.		X		
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.		X		
ODS 11. Ciudades y comunidades sostenibles.		X		
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.			X	
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.		X		
ODS 17. Alianzas para lograr objetivos.	X			

De los objetivos de desarrollo sostenibles anteriormente mencionados, relacionamos el proyecto con los mismos por las siguientes razones:

- **Igualdad de género.** La idea presentada busca la igualdad de todas las personas, sin crear ningún tipo de restricción por edad, raza o género, lo que fomenta la creación de una comunidad igualitaria y favorece a conseguir una igualdad de género extrapolable a toda la sociedad.
- **Reducción de desigualdades,** ErasmusEvents se encuentra disponible para todos los usuarios independientemente de su origen o de su nivel económico, por lo que al ofrecer los mismos servicios a todos los usuarios ayuda a reducir las desigualdades presentes en la sociedad ya que cualquier persona puede disfrutar de las mismas actividades.
- **Ciudades y comunidades sostenibles,** fomentando en la propia aplicación el transporte público al mostrar información de este, se fomenta que los usuarios lo utilicen ayudando a crear ciudades y comunidades más amigables con el medio ambiente.
- **Acción por el clima,** mediante mensajes a los usuarios y promoción a eventos verdes, ErasmusEvents buscará promover un estilo de vida verde y sostenible.
- **Paz, justicia e instituciones sólidas,** una muy buena forma conseguir la paz es la de crear comunidades que colaboren entre sí creando lazos. Al compartir experiencias y amistades durante la experiencia erasmus los estudiantes mejoran su sentimiento de unidad, favoreciendo unas instituciones más sólidas y fomentando la paz al ver a todo el mundo como iguales.
- **Alianzas para lograr objetivos,** gracias a la comunidad creada durante tu estancia de erasmus, se consigue incentivar la cooperación entre países ya que los jóvenes no verán a las personas de otras nacionalidades como extraños o rivales, si no como amigos en quien confiar. ErasmusEvents facilita esta interacción, por lo que promueve la creación de buenos vínculos entre naciones ya que todas las personas tienen acceso a las mismas actividades sin importar su país.

