

SALTOUR

Soluciones tecnológicas para una mejora de la
experiencia turística

Memoria del Proyecto



**VNiVERSIDAD
D SALAMANCA**

Proyecto de Fin de Máster en Ingeniería Informática

Tutores:

Ángel Luis Sánchez Lázaro

Ana Belén Gil González

Alumno:

Miguel Cabezas Puerto

D. Ángel Luis Sánchez Lázaro, y Dña. Ana Belén Gil González, profesores del Departamento de Informática y Automática de la Universidad de Salamanca.

CERTIFICAN:

Que el trabajo de desarrollo titulado “**Saltour**” ha sido realizado bajo su dirección por **Miguel Cabezas Puerto**. Y para que así conste a todos los efectos oportunos.

En Salamanca, 1 de septiembre de 2022

D. Ángel Luis Sánchez Lázaro

Dña. Ana Belén Gil González

Departamento de Informática y Automática

Universidad de Salamanca

Memoria del proyecto

| | |
|--|-----------|
| RESUMEN | 7 |
| CASTELLANO | 7 |
| INGLÉS..... | 7 |
| INTRODUCCIÓN | 8 |
| ESTADO DEL ARTE | 8 |
| <i>Contexto social</i> | 8 |
| <i>Gamificación y turismo</i> | 9 |
| APLICACIÓN..... | 15 |
| <i>Caso de estudio</i> | 15 |
| <i>Breve descripción</i> | 16 |
| CONTENIDO Y ESTRUCTURA..... | 17 |
| OBJETIVOS DEL PROYECTO | 18 |
| OBJETIVOS DEL SISTEMA | 18 |
| <i>Objetivos funcionales</i> | 18 |
| <i>Objetivos no funcionales</i> | 19 |
| OBJETIVOS PERSONALES..... | 19 |
| CONCEPTOS TEÓRICOS | 20 |
| ¿QUÉ ES GAMIFICACIÓN?..... | 20 |
| ¿QUÉ ES APRENDIZAJE BASADO EN RETOS? | 20 |
| ¿QUÉ ES UNA API?..... | 20 |
| ¿QUÉ ES UN FICHERO XML? | 21 |
| ¿QUÉ ES <i>FOOTPRINT</i> ?..... | 22 |
| ¿QUÉ ES UN <i>MOCKUP</i> ? | 22 |
| TÉCNICAS Y HERRAMIENTAS UTILIZADAS..... | 22 |
| ORIENTACIÓN A OBJETOS | 23 |
| ANDROID STUDIO | 26 |
| HERRAMIENTAS AUXILIARES..... | 27 |
| <i>Microsoft Office Word</i> | 27 |
| <i>Tables Generator</i> | 28 |
| <i>MarvelApp</i> | 29 |
| <i>Visual Paradigm</i> | 29 |
| <i>EZEstimate</i> | 30 |
| <i>Microsoft Project</i> | 30 |
| <i>Paint</i> | 30 |
| TECNOLOGÍAS EMPLEADAS | 30 |
| JAVA | 30 |
| C# | 34 |
| VUFORIA-UNITY | 35 |
| APIS Y PLATAFORMAS | 35 |
| <i>API Google</i> | 35 |
| <i>Firebase</i> | 36 |

| | |
|---|------------|
| ASPECTOS RELEVANTES DEL DESARROLLO | 37 |
| INTRODUCCIÓN | 37 |
| METODOLOGÍA DE TRABAJO | 37 |
| <i>Teoría de Proceso Unificado</i> | <i>38</i> |
| <i>Fases del Proceso Unificado</i> | <i>38</i> |
| ESTIMACIÓN DE COSTE Y ESFUERZO | 39 |
| PLANIFICACIÓN TEMPORAL DEL PROYECTO | 40 |
| ESPECIFICACIÓN DE REQUISITOS..... | 42 |
| <i>Roles en el proyecto</i> | <i>43</i> |
| <i>Descripción del sistema.....</i> | <i>43</i> |
| <i>Objetivos.....</i> | <i>43</i> |
| <i>Requisitos del sistema.....</i> | <i>43</i> |
| <i>Matrices de rastreabilidad.....</i> | <i>48</i> |
| <i>Propuesta de arquitectura.....</i> | <i>49</i> |
| ANÁLISIS | 50 |
| <i>Modelo de dominio.....</i> | <i>50</i> |
| <i>Vista de interacción.....</i> | <i>51</i> |
| <i>Propuesta de arquitectura.....</i> | <i>52</i> |
| DISEÑO..... | 53 |
| <i>Arquitectura de capas</i> | <i>54</i> |
| <i>Subsistemas de diseño.....</i> | <i>55</i> |
| <i>Clases de diseño.....</i> | <i>56</i> |
| <i>Realizaciones de caso de uso-diseño</i> | <i>56</i> |
| <i>Modelo de despliegue</i> | <i>57</i> |
| <i>Diseño de datos.....</i> | <i>58</i> |
| <i>Diseño de interfaces.....</i> | <i>58</i> |
| MANUAL DEL PROGRAMADOR | 66 |
| <i>Descripción del funcionamiento de la aplicación.....</i> | <i>66</i> |
| <i>Aspectos relevantes del desarrollo.....</i> | <i>66</i> |
| <i>Casos de estudio relevantes</i> | <i>67</i> |
| MANUAL DEL USUARIO | 78 |
| <i>Instalación.....</i> | <i>78</i> |
| <i>How to de la aplicación.....</i> | <i>78</i> |
| CONCLUSIONES Y AMPLIACIONES..... | 97 |
| CONCLUSIONES | 98 |
| <i>Compleción de objetivos.....</i> | <i>98</i> |
| <i>Transversales</i> | <i>99</i> |
| LÍNEAS DE TRABAJO FUTURAS | 100 |
| BIBLIOGRAFÍA | 101 |

| | |
|--|----|
| ILUSTRACIÓN 1 EVOLUCIÓN DE LAS PUBLICACIONES EN SCOPUS..... | 9 |
| ILUSTRACIÓN 2 ÁREA DE PUBLICACIÓN | 10 |
| ILUSTRACIÓN 3 ÁREA DE PUBLICACIÓN 2020-2022 | 11 |
| ILUSTRACIÓN 4 GAMIFICACIÓN Y SUS CONCEPTOS RELACIONADOS..... | 12 |
| ILUSTRACIÓN 5 ESQUEMA API..... | 21 |
| ILUSTRACIÓN 6 LOGO XML | 22 |
| ILUSTRACIÓN 7 PROGRAMACIÓN ORIENTADA A OBJETOS | 25 |
| ILUSTRACIÓN 8 ANDROID STUDIO | 27 |
| ILUSTRACIÓN 9 MICROSOFT OFFICE WORD | 28 |
| ILUSTRACIÓN 10 TABLES GENERATOR..... | 28 |
| ILUSTRACIÓN 11 MARVELAPP..... | 29 |
| ILUSTRACIÓN 12 VISUAL PARADIGM | 29 |
| ILUSTRACIÓN 13 MICROSOFT PROJECT | 30 |
| ILUSTRACIÓN 14 PAINT | 30 |
| ILUSTRACIÓN 15 JAVA | 33 |
| ILUSTRACIÓN 16 GOOGLE API..... | 36 |
| ILUSTRACIÓN 17 FIREBASE | 37 |
| ILUSTRACIÓN 18 ESQUEMA PLANIFICACIÓN TEMPORAL PU..... | 39 |
| ILUSTRACIÓN 19 PLANIFICACIÓN TEMPORAL..... | 42 |
| ILUSTRACIÓN 20 PROPUESTA DE ARQUITECTURA | 49 |
| ILUSTRACIÓN 21 DIAGRAMA DE CLASES..... | 50 |
| ILUSTRACIÓN 22 DIAGRAMA DE SECUENCIA..... | 52 |
| ILUSTRACIÓN 23 PROPUESTA ARQUITECTURA ANÁLISIS..... | 53 |
| ILUSTRACIÓN 24 ARQUITECTURA DE CAPAS..... | 55 |
| ILUSTRACIÓN 25 SUBSISTEMA DE DISEÑO | 55 |
| ILUSTRACIÓN 26 CLASE DE DISEÑO USUARIO | 56 |
| ILUSTRACIÓN 27 REALIZACIÓN CU DISEÑO REGISTRAR USUARIO | 57 |
| ILUSTRACIÓN 28 MODELO DE DESPLIEGUE | 57 |
| ILUSTRACIÓN 29 DIAGRAMA ENTIDAD-RELACIÓN..... | 58 |
| ILUSTRACIÓN 30 PANTALLA DE LOGIN Y REGISTRO..... | 60 |
| ILUSTRACIÓN 31 PANTALLA PRINCIPAL..... | 61 |
| ILUSTRACIÓN 32 PANTALLA OLVIDO DE CONTRASEÑA..... | 61 |
| ILUSTRACIÓN 33 PANTALLA JUGAR..... | 62 |
| ILUSTRACIÓN 34 PANTALLA PERFIL DE USUARIO | 63 |
| ILUSTRACIÓN 35 PANTALLA ESTADÍSTICAS INDIVIDUALES..... | 63 |
| ILUSTRACIÓN 36 PANTALLA DE ESTADÍSTICAS GLOBALES | 64 |
| ILUSTRACIÓN 37 PANTALLA DE AYUDA | 65 |
| ILUSTRACIÓN 38 VALIDACIÓN EMAIL Y CONTRASEÑA | 68 |
| ILUSTRACIÓN 39 LLAMADA VALIDACIÓN EMAIL | 68 |
| ILUSTRACIÓN 40 VALIDAR EMAIL..... | 68 |
| ILUSTRACIÓN 41 EXPRESIÓN REGULAR EMAIL | 68 |
| ILUSTRACIÓN 42 LANZAMIENTO EVENTO REGISTRO..... | 69 |
| ILUSTRACIÓN 43 ACTUALIZACIÓN DE PERFIL..... | 69 |
| ILUSTRACIÓN 44 ALMACENAMIENTO DE FOTO | 70 |
| ILUSTRACIÓN 45 ALMACENAMIENTO EN FIREBASE | 70 |
| ILUSTRACIÓN 46 VERIFICACIÓN DE USUARIOS | 71 |
| ILUSTRACIÓN 47 FLUJO ALTERNATIVO..... | 71 |
| ILUSTRACIÓN 48 VALIDACIÓN TOKEN GOOGLE..... | 72 |
| ILUSTRACIÓN 49 COMPROBACIÓN EXISTENCIA USUARIO | 72 |
| ILUSTRACIÓN 50 USUARIO EXISTENTE SIN RETOS..... | 72 |

| | |
|--|----|
| ILUSTRACIÓN 51 CACHÉ EN SHARED PREFERENCES..... | 73 |
| ILUSTRACIÓN 52 PARADA INTENCIONADA DE TIEMPO | 73 |
| ILUSTRACIÓN 53 COMPROBACIÓN SERVICIO CONTEO ACTIVO | 73 |
| ILUSTRACIÓN 54 ACTUALIZACIÓN DE DATOS | 74 |
| ILUSTRACIÓN 55 HILO DE ACTUALIZACIÓN | 74 |
| ILUSTRACIÓN 56 ACTUALIZACIÓN DE DATOS USUARIO REGISTRADO | 74 |
| ILUSTRACIÓN 57 LANZAMIENTO DE REALIDAD AUMENTADA..... | 75 |
| ILUSTRACIÓN 58 RECUPERACIÓN COLECCIÓN USUARIO | 75 |
| ILUSTRACIÓN 59 ACTUALIZACIÓN DE TIEMPOS | 75 |
| ILUSTRACIÓN 60 ALMACENAMIENTO EN COLECCIÓN..... | 75 |
| ILUSTRACIÓN 61 ALMACENAMIENTO EN CACHÉ..... | 76 |
| ILUSTRACIÓN 62 LLAMADA A REALIDAD AUMENTADA | 76 |
| ILUSTRACIÓN 63 INTERACCIÓN CON LA REALIDAD AUMENTADA..... | 77 |
| ILUSTRACIÓN 64 REGISTRO..... | 79 |
| ILUSTRACIÓN 65 REGISTRO INVÁLIDO..... | 79 |
| ILUSTRACIÓN 66 CORREO NO VERIFICADO TRAS REGISTRO | 80 |
| ILUSTRACIÓN 67 CORREO DE VERIFICACIÓN | 80 |
| ILUSTRACIÓN 68 CORREO VERIFICADO..... | 81 |
| ILUSTRACIÓN 69 LOGIN | 82 |
| ILUSTRACIÓN 70 LOGIN INCORRECTO..... | 82 |
| ILUSTRACIÓN 71 LOGIN CON GOOGLE | 83 |
| ILUSTRACIÓN 72 OLVIDO CONTRASEÑA | 83 |
| ILUSTRACIÓN 73 CORREO DE RESETEO DE CONTRASEÑA..... | 84 |
| ILUSTRACIÓN 74 CONTRASEÑA RESETEADA..... | 84 |
| ILUSTRACIÓN 75 INICIO | 85 |
| ILUSTRACIÓN 76 PANTALLA JUGAR CON RETO ACTIVO | 86 |
| ILUSTRACIÓN 77 PANTALLA JUGAR RETO NO COMPLETADO | 86 |
| ILUSTRACIÓN 78 PANTALLA JUGAR RETOS COMPLETADOS Y NO INICIADO..... | 87 |
| ILUSTRACIÓN 79 PANTALLA ESCANEO QR | 87 |
| ILUSTRACIÓN 80 MAPA SIN UBICACIÓN ACTIVA..... | 88 |
| ILUSTRACIÓN 81 MAPA CON UBICACIÓN ACTIVA | 88 |
| ILUSTRACIÓN 82 MI PERFIL | 89 |
| ILUSTRACIÓN 83 EDICIÓN DE PERFIL..... | 89 |
| ILUSTRACIÓN 84 ESTADÍSTICAS INDIVIDUALES | 90 |
| ILUSTRACIÓN 85 ESTADÍSTICAS GLOBALES | 91 |
| ILUSTRACIÓN 86 PERFIL OTRO JUGADOR | 91 |
| ILUSTRACIÓN 87 COMPARTIR APP..... | 92 |
| ILUSTRACIÓN 88 COMPARTIR APP FACEBOOK | 93 |
| ILUSTRACIÓN 89 PREGUNTAS FRECUENTES | 94 |
| ILUSTRACIÓN 90 AYUDA PERSONALIZADA..... | 94 |
| ILUSTRACIÓN 91 MENÚ LATERAL..... | 95 |
| ILUSTRACIÓN 92 CERRAR SESIÓN..... | 96 |
| ILUSTRACIÓN 93 REALIDAD AUMENTADA | 97 |

Resumen

Castellano

En un mundo en el que el uso de las tecnologías es ya una forma forma de vida y en el que poco a poco el formato analógico está quedando relegado en todos los ámbitos y edades, la cultura no puede quedarse atrás ya que esta nutre al ser humano y ha sido durante toda su historia una pieza angular en el logro del progreso. Por ello Saltour busca fomentar el turismo en de cualquier ciudad de una forma alternativa y llamativa, especialmente para generaciones jóvenes, tanto en población autóctona como foránea. Para ello desarrolla una propuesta de productos turísticos a través del descubrimiento de la ciudad basada en retos.

Como caso de estudio, se pretende desarrollar en base a la propuesta una aplicación móvil para Android que permita descubrir diferentes puntos de la ciudad de Salamanca a través de diversos desafíos propuestos a toda la comunidad de jugadores

Palabras clave: gamificación, aprendizaje basado en retos, aplicación móvil, realidad aumentada.

Inglés

In a world in which the use of technologies is already a way of life and in which the analog format is gradually being relegated in all surroundings and ages, culture cannot be left behind since it nourishes the human being and has been throughout its history a cornerstone in the achievement of progress. For this reason, Saltour seeks to promote tourism in any city in an alternative and attractive way, especially for young generations, both in the native population and abroad. To do this, it develops a proposal for tourism products through the discovery of the city based on challenges.

As a case study, it is intended to develop, based on the proposal, a mobile application for Android that allows discovering different points of the city of Salamanca through various challenges proposed to the entire community of players.

Keywords: gamification, challenge based learning, mobile app, augmented reality.

Introducción

El presente documento recoge la memoria del proyecto Fin de Máster titulado “Saltour” en la titulación de Máster en Ingeniería Informática. Ha sido realizado por el alumno Miguel Cabezas Puerto durante medio año, desde abril de 2022 hasta septiembre de 2022.

Estado del arte

Contexto social

La sociedad hoy en día sigue evolucionando, lo que hace unas décadas se hacía de forma analógica mediante la cumplimentación de formularios, instancias, escritos... Pasó a realizarse, poco a poco, haciendo uso de la tecnología que nos abrió Internet, de forma online, tanto es así que las administraciones debieron adaptar su forma de trabajar para pasar a tener un modelo híbrido entre lo presencial y lo telemático, así podemos ver trámites cada día realizados en mayor medida a distancia (declaración de la renta, sellado de la tarjeta de empleo, altas en la seguridad social...). La sociedad se está digitalizando más y más, haciendo del mundo un lugar conectado independientemente de la región geográfica donde nos encontremos. Esta tendencia es la que se espera seguir en los años venideros.

Sumado a ello, en los últimos años, la población ha transformado el uso de los dispositivos móviles, de meros instrumentos para llamadas, a instrumentos sofisticados y adaptables a cualquier tipo de tarea, desde ocio (juegos, redes sociales) hasta herramienta de trabajo (p.ej. herramienta de planos en arquitectura). Todo lo que anteriormente se repartía en diferentes dispositivos e instrumentos, hoy se aglutina en un dispositivo móvil. Por ello que, cada vez hay más gente que dispone de teléfonos móviles de última generación y las aplicaciones se han convertido en bienes de gran consumo en este mercado.

Es por todo ello que la cultura no puede quedarse atrás ya que nutre al ser humano y ha sido durante toda su historia una pieza angular en el logro del progreso. Muestra de este progreso que se quiere ver en el sector para hacerlo más atractivo a todos los públicos podemos encontrar la gamificación de todo tipo de juegos clásicos de inteligencia, virtualización de museos, aparición de visitas online, etc.

Gamificación y turismo

Estudios previos

2013-2019

Partiendo de la investigación realizada por Noelia Araújo, Bárbara Barroso, Rogério Azevedo Gomes y Lucília Cardoso en su publicación relacionada con la gamificación en el turismo (Araújo, Barroso, Azevedo Gomes, & Cardoso, 2019), se observa que se trata de un área de estudio reciente. Así los primeros trabajos aparecen, según la base de datos Scopus (Elsevier B.V, s.f.), en el año 2013, centrándose el 35% de los trabajos totales en el año 2017.

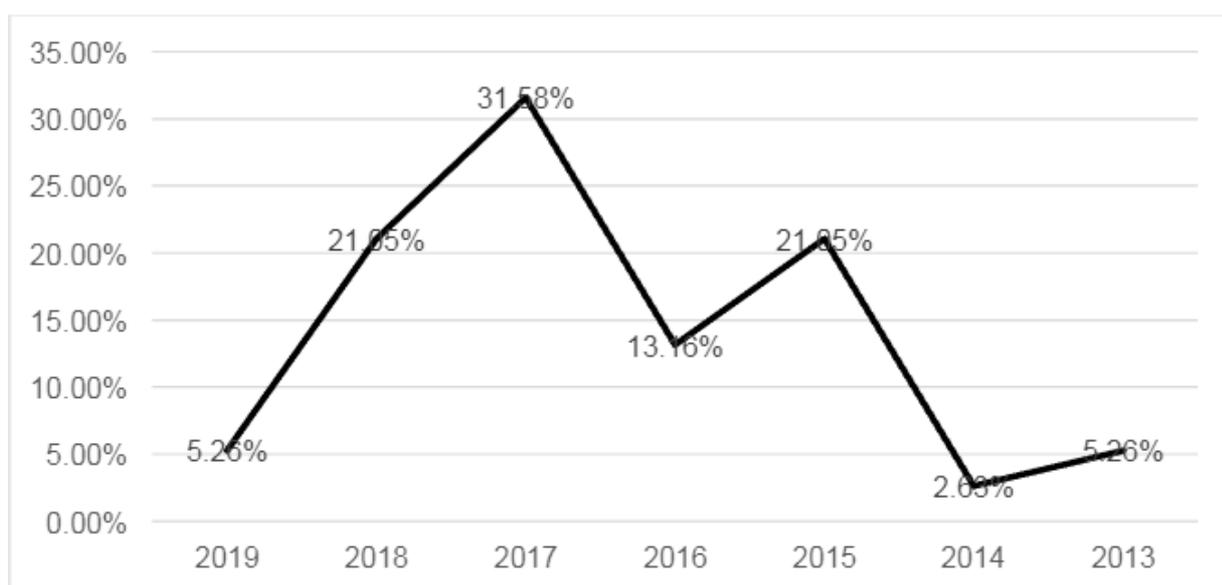


Ilustración 1 Evolución de las publicaciones en Scopus

Los artículos de este campo se publican fundamentalmente en revistas del área de las ciencias de la computación seguidas por las ciencias sociales y en menor medida las ciencias ambientales.

| Gamification+tourism | |
|--|-------|
| Área | % |
| Computer Science | 36,62 |
| Social Sciences | 17,81 |
| Business, management, account | 16,44 |
| Environmental Sciences | 8,22 |
| Economics | 4,11 |
| Engineering | 4,11 |
| Mathematics | 4,11 |
| Art and Humanities | 4,11 |
| Energy | 2,74 |
| Neuroscience | 1,37 |
| Physics and Astronomy | 1,37 |
| *Un mismo trabajo puede pertenecer a más de un área. | |

Ilustración 2 Área de publicación

Así, los autores del informe realizan una segunda etapa de investigación para conocer los aspectos y subtemas mayormente tratados en base al análisis de la frecuencia de aparición de palabras en 40 trabajos seleccionados. Así, *gamification* está presente en las palabras clave del 60% de los trabajos y turismo en el 42,5%. Le siguen en tercer lugar *game* o *games*, en el 35% y a continuación *mobile*, usado en las palabras clave del 22,5% de los trabajos estudiados. A estos tópicos clave se añaden *city*, *augmented*, *experience*, *application*, *marketing* o *location*. Con ello claramente se puede observar que en los últimos años los estudios se centran en la aplicación de tecnología móvil y realidad aumentada a la promoción del turismo en sus diferentes destinos, es decir, la potenciación de un turismo experiencial.

Por otra parte, en lo que respecta a los países investigadores en estas áreas, se posiciona España como primer país, seguido de Portugal, Reino Unido y China.

2020-Actualidad

En lo que respecta a estos tres últimos años, se ha realizado un estudio propio basado en el de los años anteriores para confirmar si la tendencia en el mundo de la gamificación en el turismo continúa en alza. Así, basándose de nuevo en las publicaciones registradas en la base de datos Scopus, se ha procedido a la búsqueda del número de publicaciones en las cuales la palabra gamificación (en inglés y español) está contenida en su título, *abstract* o palabras claves. Así, encontramos un total de 1662 publicaciones para el año 2020, 1884 publicaciones para el año 2021 y un total de 1020 trabajos para el año en curso.

Analizando de nuevo las áreas, en el conjunto de los tres años, en las que se publican estos trabajos de nuevo encontramos a las ciencias de la computación en primera posición, seguida de las ciencias sociales y un nuevo ámbito respecto a los años anteriores como es la ingeniería. Esto podría significar un proceso de transformación en realidad, en proyectos tangibles, de las propuestas teóricas. En la siguiente tabla podemos observar las principales áreas de publicación en temática de gamificación.

| Gamification/Gamificación | |
|-------------------------------------|----------|
| Área | % |
| Computer Science | 31,06 |
| Social Science | 21,25 |
| Engineering | 15,02 |
| Mathematics | 6,40 |
| Medicine | 6,29 |
| Business, Management and Accounting | 6,05 |
| Decision Sciences | 4,68 |
| Arts and Humanities | 2,68 |
| Psychology | 2,54 |
| Others | 4,03 |

Ilustración 3 Área de publicación 2020-2022

En lo relativo al origen de la publicación, en estos tres últimos años ha pasado a la cabeza Estados Unidos, con 561 publicaciones, relegando a España al segundo lugar, con 466. Completa el top 3 Alemania con 359. A esta le siguen Reino Unido y China con el mismo número de publicaciones (286 cada una), Portugal (206), Indonesia (175), Italia (172) y Brasil (159). Cierra esta lista de diez India con 154 publicaciones.

A continuación, se consultan aquellas palabras que más frecuentemente aparecen junto a gamificación bien en los títulos o palabras claves. El resultado, mostrado en la siguiente tabla, nos hace ver la estrecha relación que guarda la temática con el mundo educativo y la búsqueda de la motivación en este a través de la aplicación de nuevas tecnologías. Para el estudio, se han tomado los tres últimos años y un total de 13943 apariciones entre todas las palabras. A partir de ellas se han seleccionado las 10 primeras cuyo significado varía respecto a las demás (así, por ejemplo, se ha contado *Human* pero no *Humans*). Pueden parecer porcentajes ínfimos, pero, debido a la flexibilidad del lenguaje, podemos encontrar palabras clave distintas cuando en verdad en origen representan el mismo concepto (singular/plural, palabras compuestas, términos compuestos,

derivaciones, etc.). Lo que cabe destacar es que en las diez primeras posiciones aparecen términos vinculados a la educación a través de nuevas tecnologías para buscar la motivación en el alumnado y en general en el ser humano.

| Palabra | Apariciones | % |
|---------------------|--------------------|----------|
| Students | 676 | 4,90 |
| Motivation | 535 | 3,84 |
| e-Learning | 459 | 3,29 |
| Human | 451 | 3,23 |
| Serious games | 301 | 2,16 |
| Learning systems | 292 | 2,09 |
| Education | 289 | 2,07 |
| Virtual Reality | 252 | 1,81 |
| Teaching | 195 | 1,40 |
| Game-based learning | 189 | 1,36 |

Ilustración 4 Gamificación y sus conceptos relacionados

A mayores, para consolidar la idea explicada anteriormente en base a datos de la relación de la gamificación con el mundo educativo y las nuevas tecnologías, se han analizado algunas de las publicaciones encontradas, reafirmando con ellas dicha idea. Entre las más recientes, con fecha de este año en curso, podemos encontrar las recogidas en el siguiente apartado.

Trabajos académicos actuales

Se citan tres trabajos seleccionados bien por su importancia, bien por su estrecha relación con el caso de estudio planteado en el presente proyecto y explicado en sucesivos apartados.

A scoping review of gamification for mental health in children: Uncovering its key features and impact

Este trabajo parte de la idea de que la situación mental en la etapa de la juventud condiciona el resto de la vida. En él se estudia el tratamiento de problemas mentales no orgánicos/fisiológicos en esta etapa de la vida a través de la gamificación mediante dispositivos móviles y ordenadores llegando a la conclusión de que a través de estos juegos los jugadores adquieren unas mayores habilidades para hacer frente a sus problemas en la vida real a medida que avanzan en ellos. (Huiting, 2022)

Gamification Applications in E-learning: A Literature Review

Este trabajo, el más citado en relación con la temática durante el presente año, investiga los beneficios y desafíos documentados de las aplicaciones de gamificación en la educación, obteniendo como resultado que la gamificación se toma cada día más como una herramienta útil para lograr un entorno de aprendizaje más atractivo. De igual forma el estudio obtuvo que los elementos que mejor se conjugan con este método de aprendizaje son los sistemas de puntuaciones y clasificatorios, así como los niveles en los juegos. (Salem, Noori, & Ozdamli, 2022)

Gamification as a methodological strategy at the University. The case of BugaMAP: students' perceptions and evaluations

Se elige este estudio ya que, a diferencia de los anteriores, se trata de un análisis con usuarios reales tras el uso de una aplicación real, BugaMap, que busca la gamificación a través de juego de rol en el que los jugadores deben tomar decisiones estratégicas para la gestión y mejora de una compañía aseguradora. (Mapfre, s.f.)

En el estudio se recoge la opinión de 144 estudiantes universitarios en áreas de las ciencias sociales tras haber sido sometidos a la experiencia buscando obtener su grado de satisfacción, así como la utilidad que le vieron. Los resultados son prometedores y apoyan la efectividad del aprendizaje basado en juegos ya que los estudiantes valoraron positivamente aquello que el juego les ha aportado para su aprendizaje. Muchos de ellos indicaron que sus competencias blandas o *soft skills* relacionadas con el trabajo en equipo y toma de decisiones fueron altamente mejoradas y puestas a prueba. (González Limón, Rodríguez Ramos, & Padilla Carmona, 2022)

Evolución y proyectos empresariales actuales

En lo que respecta a los orígenes de la temática, según lo indicado por Pablo Díaz Luque y Cynthia Correa en su publicación acerca de la gamificación y los destinos turísticos (Díaz Luque & H.W. Correa, 2017), se remonta a 2011 cuando la Junta Nacional de Turismo de Irlanda lanzó un juego social llamado Ireland Town a través de la red social Facebook. También se destacan las iniciativas del Gobierno de Noruega que, en colaboración con la empresa Innovation Norway, pasó a ofrecer dos juegos, Holmenkollen Ski Jump y Trysil Twintip, en el sitio web oficial de viajes del país. Otro juego mencionado es el del Gobierno Brasileño, con el juego Brasil Quest lanzado en 2012 por la Junta de Turismo de Brasil para promover las 12 ciudades sede de la Copa Mundial de la FIFA en 2014, y para motivar el compromiso de los turistas para con esas sedes. En los años venideros y hasta la actualidad podemos encontrar otras propuestas que han logrado llegar a ser tangibles y que se recogen a continuación.

Proyectos nacionales

En primer lugar, como caso reseñable y extendido a nivel nacional en cuanto a implantación se encuentra Tourisfy (Tourisfy, s.f.) , una *startup* que ofrece experiencias turísticas en diversas ciudades españolas (Sevilla, Barcelona, Benidorm, Málaga...) a través de la gamificación y retos siguiendo diversos itinerarios.

Por otro lado, a nivel local, Salamanca cuenta con una *startup* de reciente creación, DareMapp (DareMapp Software S.L, 2022), la cual ofrece una aplicación para dispositivos móviles en la cual se pueden crear retos de diversa índole, entre los que se encuentran aquellos vinculados a puntos de interés turístico, en los que se obtiene una recompensa si son completados. Esta aplicación, además de basarse en uno de los pilares principales de este proyecto, la gamificación, se centra también, así como lo hacía Tourisfy, en el otro pilar, el aprendizaje basado en retos.

Proyectos en el mundo

En el extranjero podemos encontrar la dinamización de las visitas al Palacio Schönbrunn de Viena a través de la *app* Schoenbrunn Secret Tale, también la guía a través de la ciudad escocesa de Edimburgo mediante la aplicación móvil Edinburgh World Heritage City App (Edimburg World Heritage, s.f.) basada en la exploración de cuatro recorridos y medición de los logros conseguidos por los diferentes grupos de usuarios de la *app*.

Justificación

Una vez estudiado el contexto social, así como las tendencias predominantes en la actualidad y propuestas llevadas a cabo, se elige la realización de este proyecto frente a la necesidad de cubrir algunas carencias detectadas en las diferentes propuestas encontradas. La realidad actual es que existen *apps* similares que gamifican pero con coste y en las que obligan a seguir un itinerario (véase Tourisfy). También existen *apps* de realidad aumentada pero centradas bien en mundos no reales o cuyo objetivo no es promover el turismo (como la no mencionada pero conocida Pokémon Go) o centradas en un turismo concreto como pueden ser aplicaciones ad-hoc para museos, hoteles, monumentos en concreto...

Este proyecto propone un aprendizaje turístico basado en retos y apoyado en la gamificación sin la necesidad de pagar por ello (al encontrarse en las primeras fases) ni la obligación de ceñirse a recorridos concretos. Esto se propone debido a la necesidad de hacer al usuario aún más partícipe y activo en el juego. Otorgándole la libertad de elección del orden de compleción de los retos se consigue que el usuario sea guía y dueño del juego logrando con esto una mayor sensación de

control y poder a la par que fomentando la toma de decisiones y asunción de responsabilidades en un entorno más creativo.

En resumen, por un lado, el proyecto está impulsado por la falta de opciones reales y por otro, respaldado por la idea de motivar a la población, especialmente la joven, al descubrimiento y gusto por el turismo cultural de una forma amena y divertida. Si todos los estudios teóricos en el marco de la educación apuntan que la gamificación a través de nuevas tecnologías hace más ameno y motivante el temario educativo en las aulas, ¿por qué no ampliar este concepto a una educación informal y a la adquisición de conocimiento complementaria durante el tiempo libre de la población como es el turismo?

Aplicación

Caso de estudio

El proyecto consiste en una aplicación para dispositivos Android que pretende fomentar el turismo en una ciudad de una forma alternativa y llamativa.

Se ha elegido el turismo ya que es un ámbito en el que la gamificación no está del todo extendida, pero en el que, gracias a su carácter asociado al descubrimiento, se puede trasladar muy bien el sistema de puntuaciones, retos y clasificaciones que tan buenos frutos viene dando en el ámbito educativo. Es así ya que estos dos ámbitos, educativo y turístico, parten de la de idea central de la adquisición de conocimientos culturales.

A su vez, se han elegido los dispositivos móviles inteligentes como medio ya que, en base a los estudios comentados en anteriores apartados, se ha visto que el uso de estos refuerza la cercanía a la población al contar la gran mayoría con al menos uno. Además, son el medio habitualmente usado y más representativo en el proceso de transformación digital que está experimentando la sociedad.

Por otro lado, se ha elegido el sistema operativo Android ya que, proporcionalmente, el público objetivo será más amplio. Esto es así ya que, a nivel general, los usuarios de Android representan, con fecha mayo de 2022, el 67,03% de los usuarios de dispositivos móviles. (Flat 101, 2022).

En cuanto al público objetivo se ha elegido como primario personas jóvenes comprendidas entre los 16 y los 35 años. Esto es debido a que es en ellas donde se ha detectado esa lejanía y despegó hacia el turismo cultural debido al carácter anticuado en el que se viene realizando (visitas guiadas con mucha información donde prima la escucha pasiva, contemplación de museos, tours

fotográficos...). Además, es esta franja de edad la que está predispuesta a un uso más ágil y naturalizado de la tecnología, en concreto la móvil, ya que han nacido y crecido en una era digital. Como público secundario se apunta hacia la población adulta en edad de trabajar, es decir, comprendidos entre los 36 y los 65 años, menos diana a medida que aumenta la edad. No se ha descartado a esta franja de edad ya que representan el grueso de la población, sin embargo, no es en ellos donde las técnicas de gamificación apliquen más a la perfección (como pasaba en la población juvenil) ya que estos no ven con desagrado seguir realizando turismo a la manera convencional, su única, por lo general, es el precio que esto supone. Y es precisamente por ello que no se les descarta como público, a través del juego, aunque no se adapten a la parte competitiva, podrán inspeccionar destinos sin necesidad de contratar una guía turística a la cual no tienen tiempo o dinero que invertirle. En definitiva, personas que no ven con malos ojos el uso de la aplicación, pero tampoco supone una mejora cualitativa en sus vidas.

Por último, la población anciana, de 65 años o más, representa el público negativo del proyecto. Personas que no usarán por norma general la aplicación ya que debido a su decalaje digital les supone una mayor dificultad que seguir realizando turismo a la manera tradicional. Es por ello que los diseños, así como las decisiones tomadas en el desarrollo del proyecto, no han tenido en cuenta las dificultades o necesidades de las que dispone este grupo de población.

Por otra parte, en cuanto a la nacionalidad y lengua materna de los usuarios no se ha descartado ningún público. Esto es debido a que se considera el turismo una actividad global por lo que pueden estar dispuestos a utilizar la aplicación cualquier persona independientemente de su condición de raza y/o idioma. Es por ello que el proyecto se plantea ser multi-idioma, al menos en los principalmente hablados mundialmente como son el inglés, francés, español y alemán.

En relación a las capacidades físicas y sensoriales, se ha centrado la aplicación en personas sin ningún tipo de disfuncionalidad sensorial (ceguera, sordera...). Por ello, de nuevo, no se ha tenido en cuenta las posibles dificultades en estos grupos de población por lo que, en una primera versión, no sería inmediato el uso por parte de estos.

Por último, se ha elegido la ciudad de Salamanca como aquella donde empezar ya que, aparte de ser la ciudad donde vive el desarrollador, es considerada una de las ciudades con mayor patrimonio de la región y aún no existe una propuesta similar que explote esta ventaja.

Breve descripción

El proyecto consiste en una aplicación para dispositivos móviles Android que pretende fomentar el turismo en generaciones jóvenes a través del aprendizaje basado en retos y la gamificación.

Para lograr este sistema de aprendizaje se elabora una propuesta de productos turísticos de la ciudad a descubrir, Salamanca, basado en desafíos y sistemas clasificatorios.

Así pues, en la aplicación, el usuario, a mayores de poder registrarse, *loguearse* y modificar los datos de su cuenta (foto, nickname, contraseña...), deberá completar una serie de retos distribuidos por la ciudad. Estos retos consisten en encontrar una serie de códigos QR situados junto a puntos de interés tales como monumentos, esculturas y museos. Para ello el usuario es guiado hasta ellos y, una vez llegado al lugar, deberá encontrar el código QR asociado a dicho punto. Hecho esto aparece una realidad aumentada con la que puede interactuar explicándole la historia de dicho lugar junto con algún dato curioso. Con todo esto el usuario puede también consultar un ranking de otros jugadores en función del número de retos completados y el tiempo tardado, así como su posición dentro de dicho ranking. Del mismo modo, el usuario recibe notificaciones semanales para revisar la clasificación y notificaciones puntuales en caso de abrirse nuevos retos, fomentando así la participación de la comunidad. Por último, cabe destacar que, de forma alternativa, el usuario puede usar la aplicación de forma anónima para la realización de los retos sin entrar a formar parte del sistema de clasificación ni puntuación.

Contenido y estructura

Este documento detalla los aspectos más relevantes de las distintas fases de desarrollo del proyecto. Para mayor profundidad en alguna de ellas se incluyen diferentes anexos adjuntos.

En la primera sección, se explican los objetivos principales que el proyecto debe cumplir. Además, se especifican cuáles son los objetivos personales que se quieren alcanzar en la realización. En segundo lugar, en las secciones segunda y tercera, se incluyen algunos conceptos teóricos que facilitarán una mejor comprensión de la aplicación.

A continuación, en la cuarta sección, se exponen las principales técnicas y herramientas utilizadas en el proyecto. También se explican los recursos y APIs externas más importantes utilizadas durante la implementación.

Más adelante, en la quinta sección, se detallan los aspectos más relevantes acerca del desarrollo de la aplicación en cada una de sus fases, por lo que resulta de suma importancia. La gran mayoría de estos aspectos se encuentran resumidos, no obstante, se pueden encontrar con total profundidad detallados en cada uno de los anexos.

Para finalizar, se presentan las conclusiones obtenidas en la realización del proyecto, así como líneas futuras que se podrían desarrollar para una mayor funcionalidad de la aplicación y algunas de las referencias más importantes utilizadas. Cabe destacar que sólo se añaden las más relevantes,

puesto que para el desarrollo del proyecto se ha debido estudiar y consultar mucha información, la mayoría relativa a Android y su forma de desarrollo.

Objetivos del proyecto

En este apartado quedan recogidos los objetivos que deberá cumplir la aplicación, así como los objetivos personales que se pretenden alcanzar con su realización.

Objetivos del sistema

El siguiente listado detalla los objetivos principales que debe satisfacer la aplicación.

Objetivos funcionales

Localizar la información sobre los puntos de interés

Para los puntos de interés de la ciudad, facilitar al usuario la ubicación exacta de estos, así como una guía para que este pueda encontrarlos

Proporcionar al usuario un sistema de clasificación y puntuaje

Otro objetivo principal de la aplicación es gamificar el turismo a través del almacenaje y visualización de una clasificación tanto propia como de terceros jugadores por parte de cada usuario

Guardar la información en remoto

Permitiendo así al usuario visualizar toda la información del resto de jugadores siempre que tenga conexión a Internet

Especificar configuraciones de usuario

Permitir al usuario establecer su propio *nickname*, contraseña, foto de perfil, etc.

Ayuda al usuario

Proporcionar al usuario una forma directa de contacto con el desarrollador de la aplicación para posibles consultas, dudas o malfuncionamientos mediante el rellenado automático de un correo electrónico donde solo se deberá especificar el motivo de este.

Desarrollar una realidad aumentada interactiva

Poder transmitir datos interesantes de los diversos puntos de interés a través de una guía turística disponible en cualquier horario y de forma gratuita al estar representada mediante una realidad aumentada visible desde el propio dispositivo móvil del usuario. Además, se pretende que esta sea accesible al mayor número de colectivos mediante la emisión las indicaciones e información tanto por medio visual como auditivo.

Objetivos no funcionales

Facilitar al usuario el descubrimiento de puntos de interés de la ciudad

Uno de los objetivos principales de esta aplicación es proporcionar al usuario una herramienta sencilla que, mediante la guía y geolocalización, otorgue al usuario una forma sencilla de encontrar diferentes puntos de interés de la ciudad.

Soporte multilingüe

Mostrar el contenido estático de la aplicación (menús, filtros de búsqueda, cabeceras, etc.) en el idioma que el usuario tenga configurado su dispositivo móvil.

Aplicación móvil de calidad

Aparte de todos los objetivos anteriores, la aplicación debe resultar de calidad, por lo que dispondrá de un buen diseño gráfico, un buen rendimiento e intentará tender a un sencillo modo de uso para el usuario. Además, deberá ofrecerle algún tipo de ayuda para que la complejidad de la fase de aprendizaje resulte mínima.

Objetivos personales

Encontrarse en la finalización de cualquier máster supone que, en muchos de los casos, además de poder continuar con tu vida estudiantil, posiblemente ya se haya tenido contacto con el mundo laboral y se busca en los estudios un puente hacia una mejora profesional

La elección de esta tecnología es debido a la importancia y popularidad de la que actualmente gozan las aplicaciones móviles en todos los ámbitos (ocio, salud, laboral, social...).

El tema elegido se debe a que, aparte de resultar de actualidad por la necesidad de acercar el turismo y la cultura a la población ante el avance de la cultura de “usar y tirar” y del consumo rápido y digital (sobre todo en la población adolescente), este proyecto resulta de suficiente complejidad como para poder aprender cada ámbito del desarrollo de una aplicación móvil en la

tecnología Android (interfaz, uso de bases de datos, almacenamiento persistente de información, trasvase de información, procedimientos de transformación de datos, etc.). Todo ello me permitirá afrontar una vía laboral futura como desarrollador de mayor categoría, no solo en esta tecnología debido también al contacto con el paradigma y las dificultades de desarrollo de una aplicación funcional más allá del ámbito educativo universitario.

Conceptos teóricos

¿Qué es gamificación?

Se entiende por *gamificación* la integración de dinámicas de juego adaptados al aula en entornos no lúdicos con finalidad de aprender, potenciar la concentración, el esfuerzo y otros valores positivos comunes a los juegos. Así sus principales objetivos son crear de un vínculo entre la persona y el contenido trabajado, motivar a la persona en el aprendizaje del contenido, así como optimizar y recompensar al individuo en aquellas tareas en las que no hay ningún incentivo más que el propio aprendizaje. (Junta de Castilla y León, s.f.),

¿Qué es aprendizaje basado en retos?

Enfoque pedagógico que involucra activamente al estudiante en una situación problemática real, relevante y de vinculación con el entorno, la cual implica la definición de un reto y la implementación de una solución. Sus principales objetivos motivar a los alumnos de forma que aprenden haciendo, promover la creatividad, el trabajo en equipo y la toma de decisiones conjunta, desarrollar un pensamiento crítico y analítico entre los alumnos y mejorar la resiliencia y la adaptación en los alumnos (Consejería de Educación, Junta de Castilla y León, s.f.)

¿Qué es una API?

Una API es un conjunto de conceptos, protocolos, definiciones y lógicas que se utilizan para el desarrollo e integración de diverso software en aplicaciones. Sus siglas en inglés significan interfaz de programación de aplicaciones. (Fernández, 2019)

Las API permiten que unas aplicaciones se comuniquen con otras sin necesidad de saber su lógica interna flexibilizando y simplificando el diseño y la administración, así permitiendo el ahorro de costes monetarios y temporales

Debido a que simplifican la forma en que los desarrolladores integran los elementos de las aplicaciones nuevas en una arquitectura actual, las API permiten la colaboración entre el equipo comercial y el de desarrollo.

Las API, como puede verse en la ilustración 1, son un medio simplificado para conectar una infraestructura existente a través del desarrollo de aplicaciones nativas de la nube, pero también permiten compartir datos con clientes y otros usuarios externos.

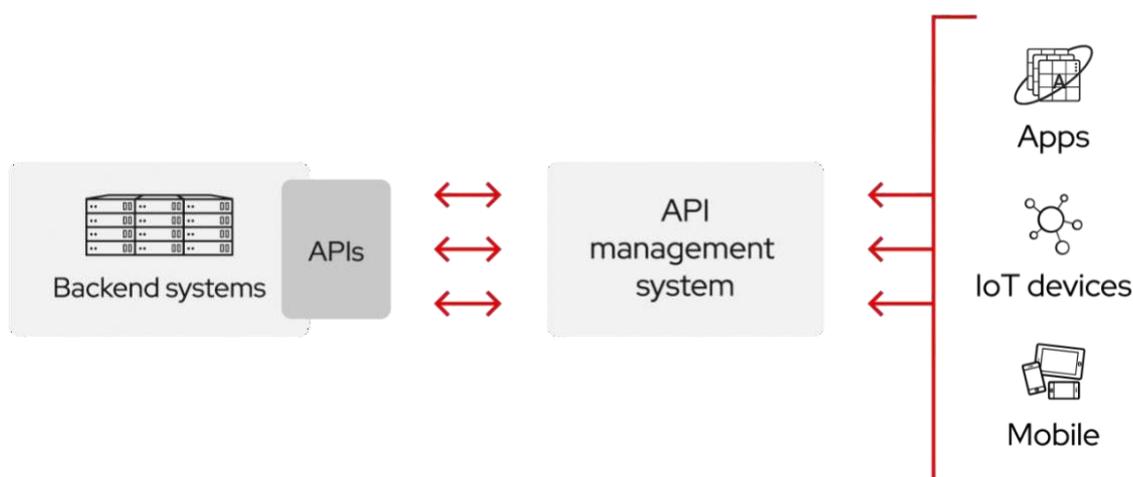


Ilustración 5 Esquema API

¿Qué es un fichero XML?

XML, siglas en inglés de eXtensible Markup Language, es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. XML permite dar soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí sin necesidad de conocer la estructura previamente, es decir, es autocontenido. Se basa en etiquetas para indicar la estructura del documento y los pares atributo-valor contenidos en este. Presenta ventajas respecto a los documentos HTML, ya que permite añadir etiquetas personalizadas a objetos para definir y diferenciar los datos que estos contienen. (Wikidat, s.f.)

Entre sus principales ventajas:

- ◇ Es extensible: basta añadir etiquetas para ampliar el documento una vez diseñado.
- ◇ El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan bugs y se acelera el desarrollo de aplicaciones.
- ◇ Si un tercero decide usar un documento creado en XML, es sencillo entender su

estructura y procesarla. Se puede comunicar los datos aplicaciones diferentes ya que representa una forma de *marshalling* (representación genérica de los datos para que todo el mundo los entienda). Así, su forma autocontenida permite no conocer la estructura de cada documento previamente.

- ◇ Transforma datos en información, pues se les añade un significado concreto y los asociamos a un contexto, con lo cual tenemos flexibilidad para estructurar documentos.



Ilustración 6 Logo XML

¿Qué es *footprint*?

Se trata del espacio ocupado por un dispositivo. En el caso del software se habla de la huella de memoria o footprint como la cantidad de memoria que necesita el programa cuando se ejecuta. (Glosario Terminología Informática, s.f.)

¿Qué es un *mockup*?

Es un diseño digital de una aplicación. Se utilizan en la fase de diseño inicial para visualizar ideas y así coordinar el establecimiento de requisitos entre el cliente y el equipo de desarrollo de una manera más sencilla y visual. Además, permiten realizar pruebas de usabilidad sin ningún esfuerzo previo de desarrollo incentivando la búsqueda de alternativas y la proposición de mejoras por parte de los usuarios. (Ryte Wiki, s.f.)

Técnicas y herramientas utilizadas

En este apartado se presentan las metodologías, las herramientas de desarrollo, los recursos software y APIs que se han utilizado para la realización del proyecto en su totalidad, desde la especificación de requisitos hasta la implementación del sistema.

Orientación a objetos

Debido a que la aplicación está destinada a dispositivos móviles Android, el desarrollo siguiendo el paradigma orientado a objetos resulta de obligado seguimiento. Es así ya que el SDK de Android está implementado en Java cuyo paradigma es el orientado a objetos. Esta imposición no resulta ninguna desventaja, sino todo lo contrario ya que este paradigma es ampliamente utilizado por lo que la documentación abunda en la red. Además, aporta muchas ventajas frente a otros métodos de programación como puede ser el paradigma estructurado.

Los objetos son entidades constituidas por un estado representado por datos y un comportamiento indicado a través de procedimientos, todo ello contenido en un mismo componente, a diferencia de los paradigmas estructurados donde datos y procedimientos se encuentran separados y con menor relación. Los objetos compuestos por la misma estructura de datos y comportamiento se agrupan en lo que se conoce como clase. Por tanto, se considera que cada objeto es una instancia de la clase a la que pertenece.

Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos. Como características más reseñables del paradigma orientado a objetos podemos destacar (Wikibooks, s.f.):

- ◇ **Abstracción:** denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un agente abstracto que puede realizar trabajo, informar y cambiar su estado, y comunicarse con otros objetos en el sistema sin revelar cómo se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos, y, cuando lo están, una variedad de técnicas es requerida para ampliar una abstracción. El proceso de abstracción permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.
- ◇ **Encapsulamiento:** significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden

este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.

- ◇ Modularidad: se denomina modularidad a la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las restantes partes. Estos módulos se pueden compilar por separado, pero tienen conexiones con otros módulos. Al igual que la encapsulación, los lenguajes soportan la modularidad de diversas formas.
- ◇ Principio de ocultación: cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz a otros objetos que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas; solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no puedan cambiar el estado interno de un objeto de manera inesperada, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.
- ◇ Polimorfismo: comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O, dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama asignación tardía o asignación dinámica. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.
- ◇ Herencia: las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza

y facilita el polimorfismo y el encapsulamiento, permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo. Esto suele hacerse habitualmente agrupando los objetos en clases y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple.

- ◇ **Recolección de basura:** la recolección de basura o garbage collector es la técnica por la cual el entorno de objetos se encarga de destruir automáticamente, y por tanto desvincular la memoria asociada, los objetos que hayan quedado sin ninguna referencia a ellos. Esto significa que el programador no debe preocuparse por la asignación o liberación de memoria, ya que el entorno la asignará al crear un nuevo objeto y la liberará cuando nadie lo esté usando.

En la mayoría de los lenguajes híbridos que se extendieron para soportar el Paradigma de Programación Orientada a Objetos como C++ u Object Pascal, esta característica no existe y la memoria debe desasignarse expresamente. En la ilustración 4 se puede observar las principales características del paradigma explicadas anteriormente.

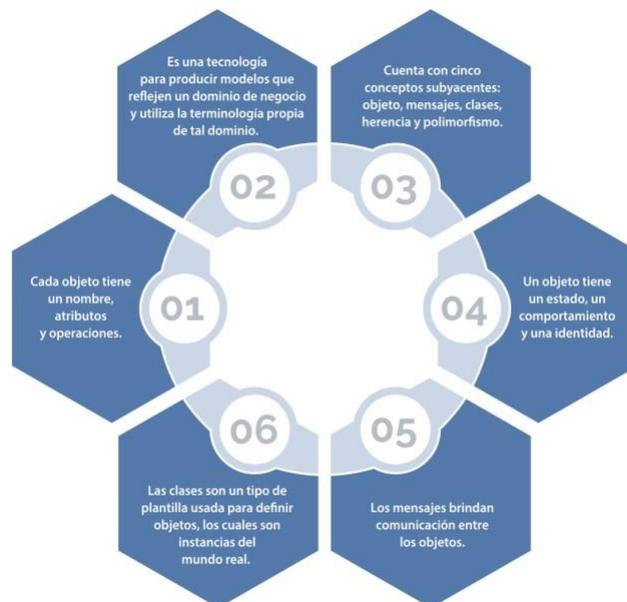


Ilustración 7 Programación orientada a objetos

Android Studio

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Reemplazó, en 2013, a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, macOS y GNU/Linux. Ha sido diseñado específicamente para el desarrollo de Android.

Estuvo en etapa de vista previa de acceso temprano a partir de la versión 0.1, en mayo de 2013, y luego entró en etapa beta a partir de la versión 0.8, lanzada en junio de 2014. La primera compilación estable, la versión 1.0, fue lanzada en diciembre de 2014.

Desde el 7 de mayo de 2019, Kotlin es el lenguaje preferido de Google para el desarrollo de aplicaciones de Android.³ Aun así, Android Studio admite otros lenguajes de programación, como Java y C++.

Entre las principales características de la versión estable actual podemos encontrar:

- ◇ Soporte para construcción basada en Gradle.
- ◇ Refactorización específica de Android y arreglos rápidos.
- ◇ Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
- ◇ Integración de ProGuard y funciones de firma de aplicaciones.
- ◇ Plantillas para crear diseños comunes de Android y otros componentes.
- ◇ Un editor de diseño enriquecido que permite a los usuarios arrastrar y soltar componentes de la interfaz de usuario
- ◇ Soporte para programar aplicaciones para Android Wear.
- ◇ Soporte integrado para Google Cloud Platform, que permite la integración con Firebase Cloud Messaging (antes 'Google Cloud Messaging') y Google App Engine.
- ◇ Un dispositivo virtual de Android que se utiliza para ejecutar y probar aplicaciones.
- ◇ Renderizado en tiempo real
- ◇ Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.

Un ejemplo de la apariencia del IDE puede observarse en la siguiente ilustración.

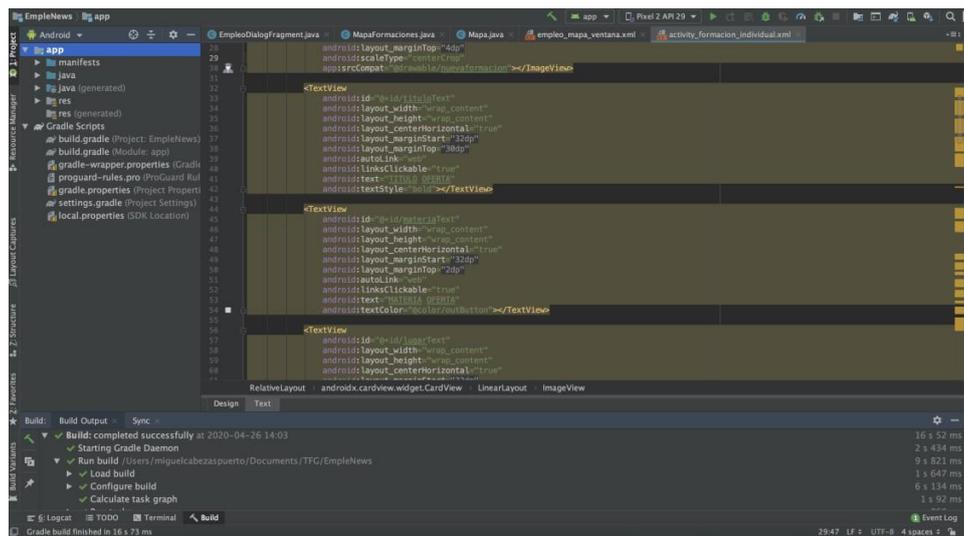


Ilustración 8 Android Studio

Herramientas auxiliares

En este apartado se detallan las diversas herramientas que han dado soporte durante las diferentes fases del proyecto ordenadas según la pertenencia a las mismas.

Microsoft Office Word

Herramienta usada para la redacción de la documentación asociada al proyecto (memoria y anexos). En la siguiente ilustración puede observarse la interfaz de la herramienta.

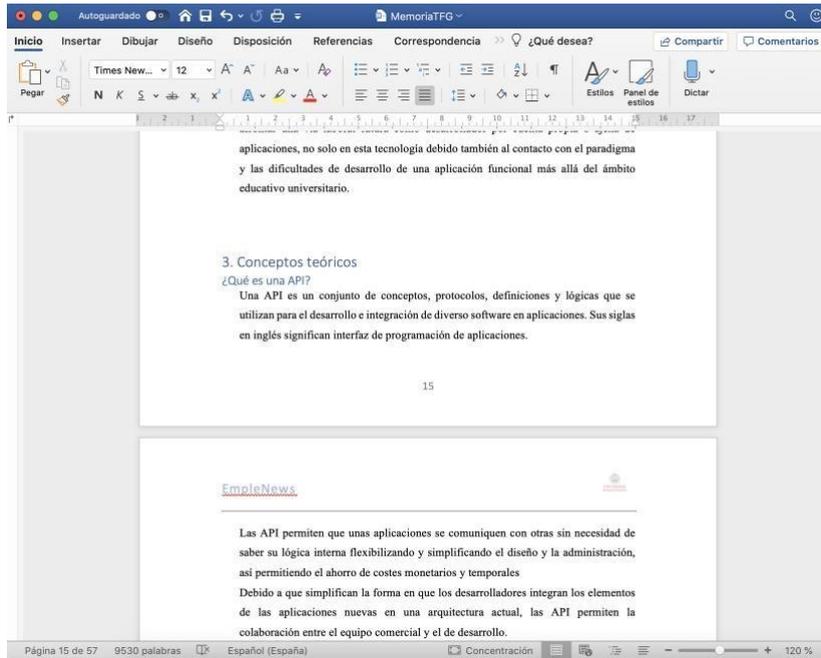


Ilustración 9 Microsoft Office Word

Tables Generator

Herramienta web usada para la realización de las matrices de rastreabilidad del anexo II

Elicitación de requisitos.

Para su manejo basta con introducir gráficamente el nº de filas y columnas que se desean en una tabla, así como su contenido. Una vez diseñada la tabla, se genera un fichero HTML que, abriéndolo en cualquier navegador, muestra la tabla diseñada. En la ilustración se indica la interfaz para la construcción de una tabla.

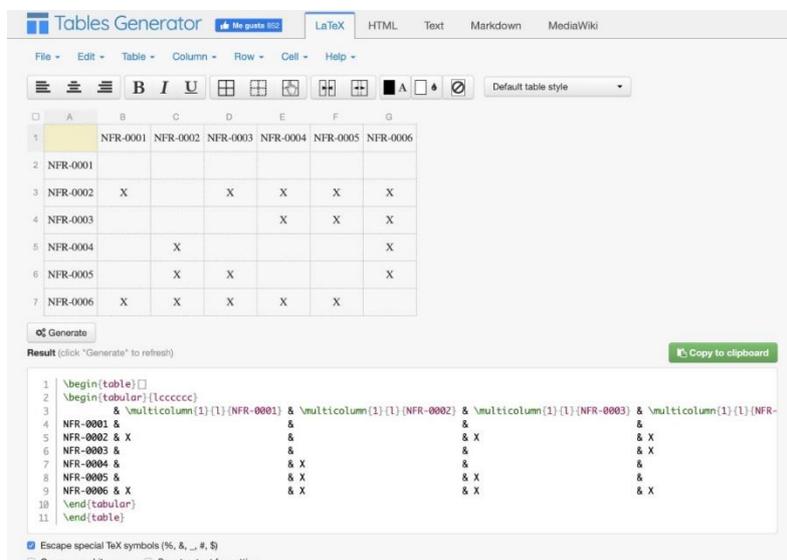


Ilustración 10 Tables Generator

MarvelApp

Herramienta web utilizada en el diseño de la interfaz mostrado en el anexo IV *Diseño*. MarvelApp es una herramienta para realizar prototipos con la que le podemos dar vida a diseños, para que sean funcionales y así, poder mostrárselos a los usuarios, clientes o al equipo de desarrollo. En la ilustración 9 se muestra el ejemplo del desarrollo de un *mockup*

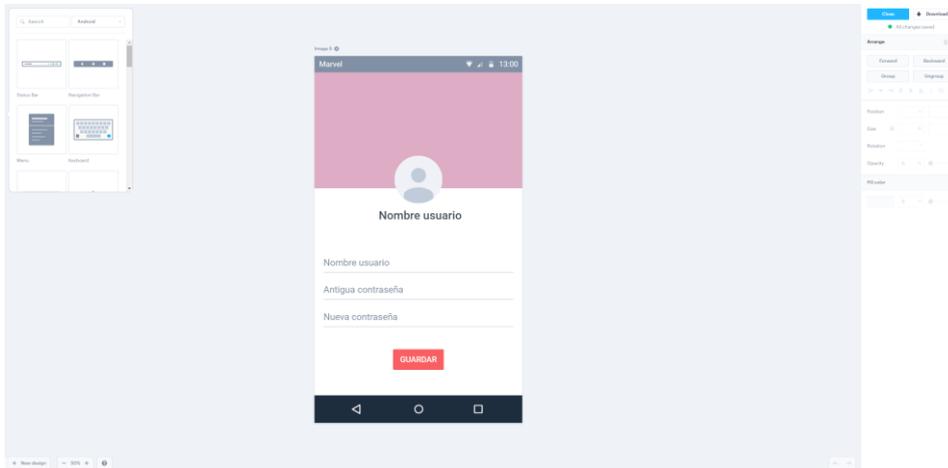


Ilustración 11 MarvelApp

Visual Paradigm

Herramienta necesaria para realizar todos los diagramas mostrados en los anexos del proyecto. Estos diagramas resultan útiles para una mejor comprensión del proyecto en cada una de sus fases. En la ilustración 10 puede observarse la interfaz de la herramienta.

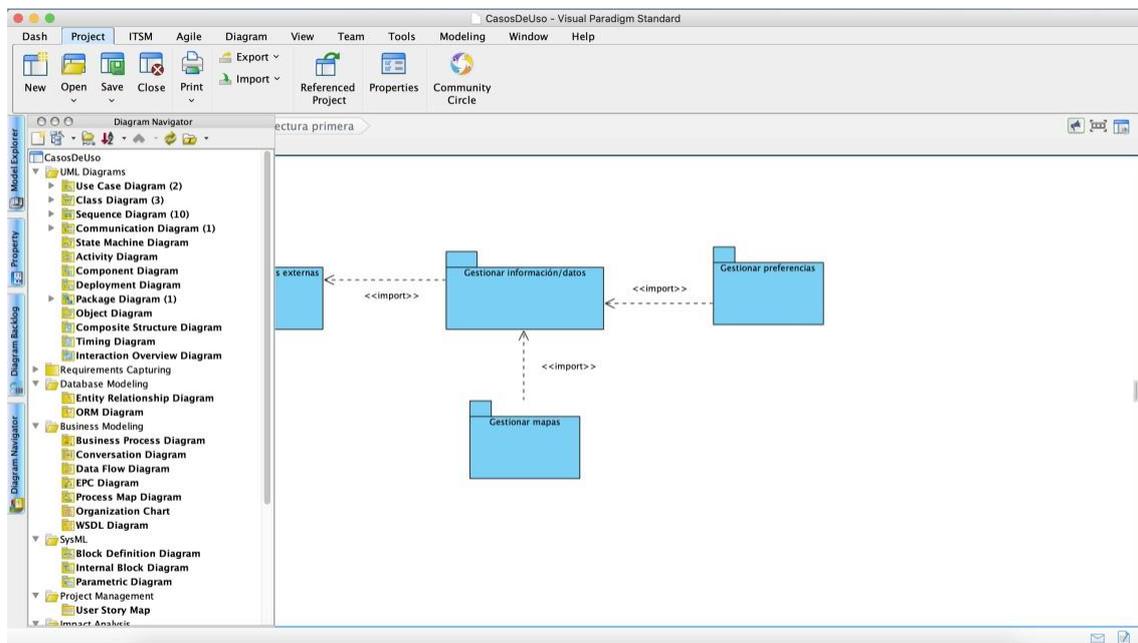


Ilustración 12 Visual Paradigm

EZEstimate

Aplicación utilizada para el cálculo del esfuerzo mediante puntos de caso de uso realizado en el anexo I Planificación temporal.

Microsoft Project

Programa usado para la realización de la planificación temporal del proyecto del anexo I. Con él se ha realizado el Diagrama de Gantt donde poder visualizar las fases del proyecto con sus dependencias y recursos asignados. Además, se utilizado para elaborar el calendario de trabajo.



Ilustración 13 Microsoft Project

Paint

Herramienta gráfica auxiliar en el desarrollo del logotipo de la aplicación.



Ilustración 14 Paint

Tecnologías empleadas

Java

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. La Plataforma Java se compone de un amplio abanico de tecnologías, cada una de las cuales ofrece una parte del complejo de desarrollo o del entorno de ejecución en tiempo real, engloba, entre otros, el lenguaje de programación Java, la API de

Java, el kit de desarrollo de aplicaciones (Software Developme Kit), la máquina virtual de Java (Java Virtual Machine) y el JRE (Java Runtime Environment).

Java, como lenguaje, es uno de los más populares debido a una serie de características que lo convierten en una herramienta eficaz para el desarrollo de aplicaciones tanto de escritorio como web. Destacan las siguientes:

- ◇ **Simplicidad:** Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ no es un lenguaje conveniente por razones de seguridad, pero C y C++ son los lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje. Java elimina muchas de las características de otros lenguajes como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el garbage collector (reciclador de memoria dinámica). No es necesario preocuparse de liberar memoria, el reciclador se encarga de ello y como es de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que limita en mucho la fragmentación de la memoria.
- ◇ **Orientación a objetos:** Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo. Las plantillas de objetos son llamadas, como en C++, clases y sus copias, instancias. Estas instancias, como en C++, necesitan ser construidas y destruidas en espacios de memoria. Java incorpora funcionalidades inexistentes en C++ como por ejemplo, la resolución dinámica de métodos. Esta característica deriva del lenguaje Objective C, propietario del sistema operativo Next. En C++ se suele trabajar con librerías dinámicas (DLLs) que obligan a recompilar la aplicación cuando se retocan las funciones de su interior. Este inconveniente es resuelto por Java mediante una interfaz específica llamada RTTI (RunTime Type Identification) que define la interacción entre objetos excluyendo variables de instancias o implementación de métodos. Las clases en Java tienen una representación en el runtime que permite a los programadores interrogar por el tipo de clase y enlazar dinámicamente la clase con el resultado de la búsqueda.

- ◇ **Carácter distribuido:** Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales. Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando.
- ◇ **Robusto:** Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria. También implementa los arrays auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo empleado en el desarrollo de aplicaciones Java. Además, para asegurar el funcionamiento de la aplicación, realiza una verificación de los ByteCodes, que son el resultado de la compilación de un programa Java. Es un código de máquina virtual que es interpretado por el intérprete Java. No es el código máquina directamente entendible por el hardware, pero ya ha pasado todas las fases del compilador: análisis de instrucciones, orden de operadores, etc., y ya tiene generada la pila de ejecución de órdenes. Java proporciona por tanto comprobación de punteros y límites de arrays, excepciones y verificación de bytecodes.
- ◇ **Arquitectura neutral:** el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (run-time) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado. En la ilustración 5 se indica el proceso de compilado y posterior ejecución del código.

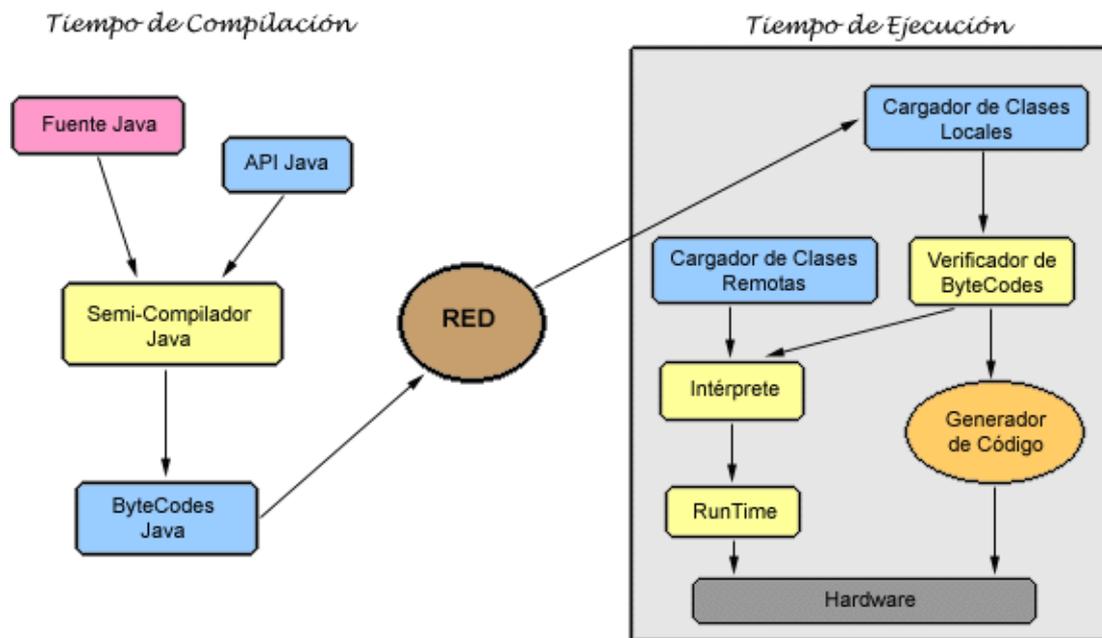


Ilustración 15 Java

- ◇ Seguro: las aplicaciones de Java resultan extremadamente seguras, ya que no acceden a zonas delicadas de memoria o de sistema, con lo cual evitan la interacción de ciertos virus. Java no posee una semántica específica para modificar la pila de programa o la memoria libre o utilizar objetos y métodos de un programa sin los privilegios del kernel del sistema operativo. Además, para evitar modificaciones por parte de los crackers de la red, implementa un método ultraseguro de autenticación por clave pública. El Cargador de Clases puede verificar una firma digital antes de realizar una instancia de un objeto. Por tanto, ningún objeto se crea y almacena en memoria, sin que se validen los privilegios de acceso. Es decir, la seguridad se integra en el momento en que se interpreta, con el nivel de detalle y de privilegio que sea necesario. Con código compilado no sería posible establecer estos niveles de seguridad entre dos objetos pertenecientes al mismo proceso, porque al compartir un único espacio de direcciones, el código de uno podría acceder tranquilamente a cualquier dirección del otro.
- ◇ Portable: más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre enteros y además, enteros de 32 bits en complemento a 2. Además, Java construye sus interfaces de usuario a través de un sistema abstracto

de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, Pc o Mac.

- ◇ Compilado e interpretado a la vez: Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, que son semejantes a las instrucciones de ensamblador. Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real.
- ◇ Multihilo: Java permite muchas actividades simultáneas en un programa. El beneficio de ser multihilo consiste en un mejor rendimiento interactivo y mejor comportamiento en tiempo real. Aunque el comportamiento en tiempo real está limitado a las capacidades del sistema operativo subyacente (Unix, Windows, etc.) de la plataforma, aún supera a los entornos de flujo único de programa (single-threaded) tanto en facilidad de desarrollo como en rendimiento.
- ◇ Dinámico: Java se beneficia todo lo posible de la tecnología orientada a objetos. Java no intenta conectar todos los módulos que comprenden una aplicación hasta el mismo tiempo de ejecución. Las librerías nuevas o actualizadas no paralizarán la ejecución de las aplicaciones actuales -siempre que mantengan el API anterior. Java también simplifica el uso de protocolos nuevos o actualizados. Si el sistema ejecuta una aplicación Java sobre la red y encuentra una pieza de la aplicación que no sabe manejar, es capaz de traer automáticamente cualquier pieza que el sistema necesite para funcionar.

C#

Evolución realizada por Microsoft tomando lo mejor de los lenguajes C y C++, a los que ha continuado añadiéndole funcionalidades y sintaxis de otros lenguajes, como Java. Lo orientó a objetos para toda su plataforma NET (tanto Framework como Core), y con el tiempo adaptó las facilidades de la creación de código que tenía otro de sus lenguajes más populares, Visual Basic, haciéndolo tan polivalente y fácil de aprender como éste, sin perder ni un ápice de la potencia original de C. En el presente trabajo se ha empleado para la codificación de los diferentes scripts para personalizar la interacción entre el usuario y los objetos en realidad aumentada diseñados en Unity-Vuforia.

Vuforia-Unity

Vuforia es una plataforma de desarrollo de aplicaciones de Realidad Aumentada (AR) y Realidad Mixta (MR) multiplataforma, con seguimiento robusto y rendimiento en una variedad de hardware (incluyendo dispositivos móviles y monitores de realidad mixta montados en la cabeza (HMD) como Microsoft HoloLens). La integración de Unity en Vuforia permite crear aplicaciones y juegos de visión para Android e iOS utilizando un flujo de trabajo de creación de arrastrar y soltar. (Unity Technologies, 2018)

Se ha empleado esta plataforma, en su integración con el motor de desarrollo de Unity, para la implementación de la realidad aumentada. Estos detalles se pueden consultar en el manual del programador contenido en el anexo V.

Principalmente se ha empleado su técnica de marcadores y detección de imágenes para representar una animación que detecte la pulsación a través de los eventos ofrecidos por el motor de desarrollo de Unity.

APIs y plataformas

En este apartado se especifican las APIs y plataformas utilizadas en el proyecto para facilitar el desarrollo de determinadas funcionalidades.

API Google

La API de Google es un conjunto de interfaces de programación de aplicaciones (API) desarrolladas por Google que permiten la comunicación con los servicios de Google y su integración con otros servicios. Ejemplos de ello son Search, Gmail, Translate o GoogleMaps. Las aplicaciones de terceros pueden usar estas API para aprovechar o ampliar la funcionalidad de los servicios existentes. Para poder hacer uso de ella es preciso poseer una cuenta de Google y cumplimentar un proceso de autenticación y autorización usando el protocolo OAuth 2.0. Para empezar, es necesario obtener credenciales de la consola de desarrolladores. Luego, la aplicación cliente, nuestra aplicación de empleos, por ejemplo, puede solicitar un token de acceso del Servidor de Autorización de Google y, posteriormente, usar ese token para autorizarse cuando accede a un servicio API de Google.

Para este proyecto se ha hecho uso de los servicios de localización de Google concretamente el de Google Maps Android API, Google Maps Geocoding API, Google Maps Geolocation

API y Google Places API for Android. Todas ellas para poderdesarrollar la localización de las diversas ofertas de empleo y formación en un mapa de Google embebido dentro de la aplicación. En la ilustración siguiente se muestra la consola de desarrollo de la API.

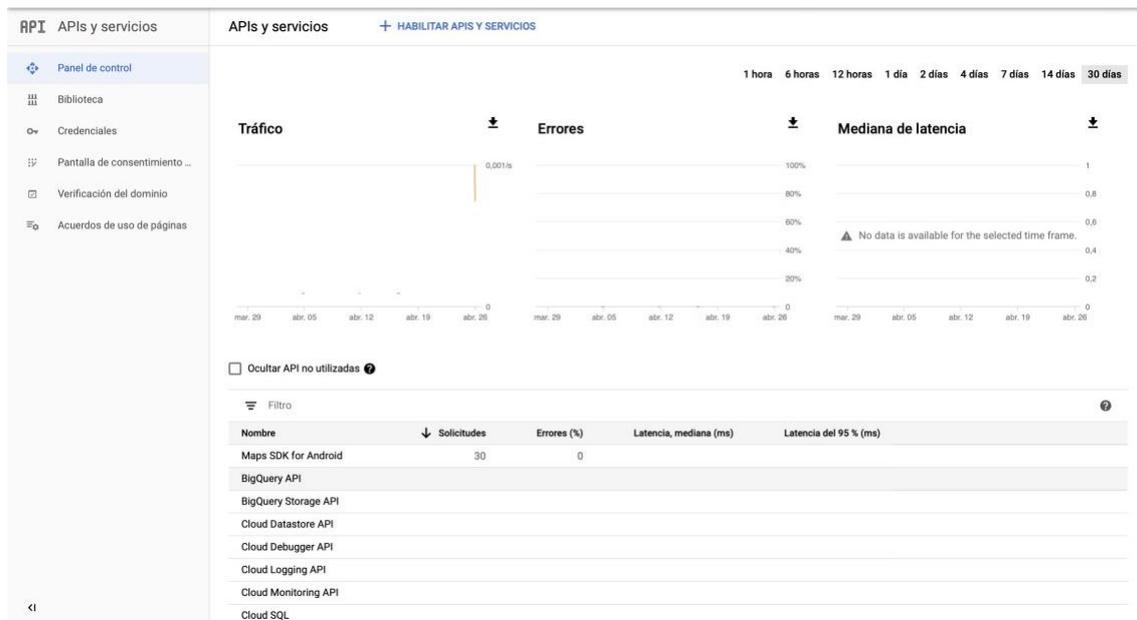


Ilustración 16 Google API

Firestore

Firestore de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Su función esencial es hacer más sencilla la creación de tanto aplicaciones webs como móviles y su desarrollo, procurando que el trabajo sea más rápido, pero sin renunciar a la calidad requerida. Sus herramientas son variadas y de fácil uso, considerando que su agrupación simplifica las tareas de gestión a una misma plataforma. Las finalidades de las mismas se pueden dividir en cuatro grupos: desarrollo, crecimiento, monetización y análisis. Es especialmente interesante para que los desarrolladores no necesiten dedicarle tanto tiempo al *backend*, tanto en cuestiones de desarrollo como de mantenimiento.

Para este proyecto se ha hecho uso de sus servicios de autenticación tanto vía email y contraseña como mediante la autenticación de Google, servicios de almacenamiento con Firestore, servicios de envío de notificaciones asíncronas mediante Cloud Messaging así como almacenamiento y recuperación de datos mediante una base de datos no relacional como Firestore. En la ilustración se muestra una visión general de la plataforma.

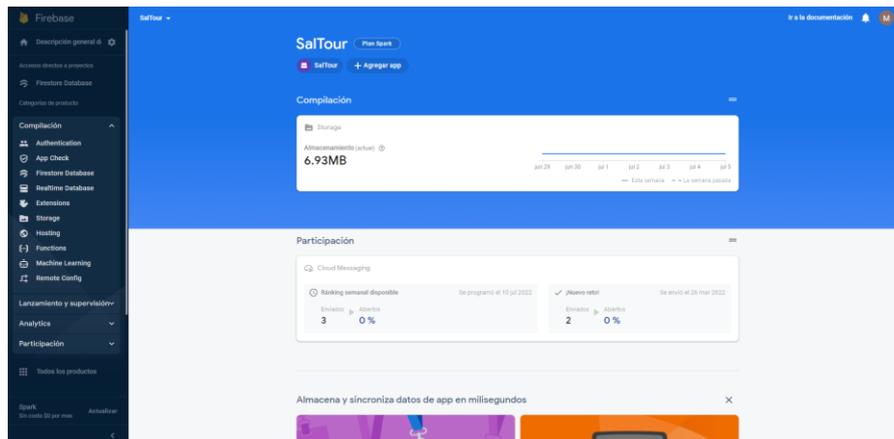


Ilustración 17 Firebase

Aspectos relevantes del desarrollo

Introducción

Este apartado enumera los aspectos más relevantes de las diferentes fases del proyecto. (Departamento de Informática de la universidad de Valladolid, s.f.) (Grupo de Investigación GRIAL, s.f.) (Moreno García, Peñalvo, & José) (Moreno García, García, & JoséFrancisco, Apuntes Ingeniería del Software II) (García) (Jiménez, 2002)

Metodología de trabajo

Para el desarrollo del trabajo de fin de grado se siguió el marco de trabajo pautado por elProceso Unificado. (Jacobson, 2000)

El Proceso Unificado es más que un simple proceso [Jacobson et al., 2000], es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

Se ha elegido este marco de trabajo debido a que, dado gran tamaño del proyecto resulta de mucha utilidad para desarrollarlo de forma sencilla, clarificando desde el principio las ideas y pudiendo organizar todo el proceso en tareas realizadas sucesivamente siguiendo un calendario y guían el refinamiento del producto final, siempre asegurando una alta calidad y el cumplimiento de los requisitos.

Teoría de Proceso Unificado

El Proceso Unificado (PU) es un marco de trabajo flexible y basado en componentes. En él no existe un proceso universal ya que puede extenderse y especializarse para una gran variedad de sistemas de software. Además, permite una gran variedad de estrategias de ciclo de vida, la definición de diferentes conjuntos de productos, actividades y encargados de las mismas.

Aunque sin duda, la característica fundamental es que es un proceso iterativo e incremental ya que se basa en la ampliación y el refinamiento del sistema mediante una serie de desarrollos cortos (mini proyectos de 2 a 6 semanas, cada iteración reproduce el ciclo de vida a menor escala).

Así, las características del Proceso Unificado son:

- ◇ Iterativo e incremental: dado que el tamaño del proyecto resulta importante resulta necesario dividirlo en varias partes más simples realizadas de forma incremental a través de flujos de trabajo. Cada uno de estos flujos seguirá una serie de fases para su realización y cada fase contendrá varias iteraciones.
- ◇ Dirigido por los casos de uso: estos nos indican de una forma precisa quiénes interactuarán en el proyecto y en qué partes de este. Se necesitarán también para determinar todos los requisitos los objetivos, los datos, las restricciones y la funcionalidad del sistema.
- ◇ Centrado en la arquitectura: es necesario realizar una arquitectura donde se encuentre organizada toda la estructura y que determine el flujo de la funcionalidad del sistema. Esta dependerá de factores como la plataforma de ejecución, la disponibilidad de componentes reutilizables, consideraciones de instalación, requerimientos no funcionales, etc. La arquitectura es importante ya que nos ayudará a ver la realización del diseño completo con una claridad y flexibilidad en cambios futuros y reutilización.

Fases del Proceso Unificado

El Proceso Unificado se repite a lo largo de una serie de ciclos de desarrollo que constituyen la vida de un sistema. Cada ciclo de desarrollo concluye con una versión entregable del producto y consta de cuatro fases:

- ◇ Inicio: se define el alcance del proyecto y se desarrollan los casos de negocio
- ◇ Elaboración: se planifica el proyecto, se especifican en detalle la mayoría de

los casos de uso y se diseña la arquitectura del sistema

- ◇ Construcción: se construye el producto
- ◇ Transición: El producto se convierte en versión beta, se corrigen problemas y se incorporan mejoras sugeridas en la revisión

Cada ciclo, como se menciona anteriormente, consta de varias iteraciones. El resultado de cada iteración constituye una versión ejecutable no orientada a la entrega conocida como hito secundario. En la ilustración siguiente se puede ver el esquema gráfico de las fases explicadas.

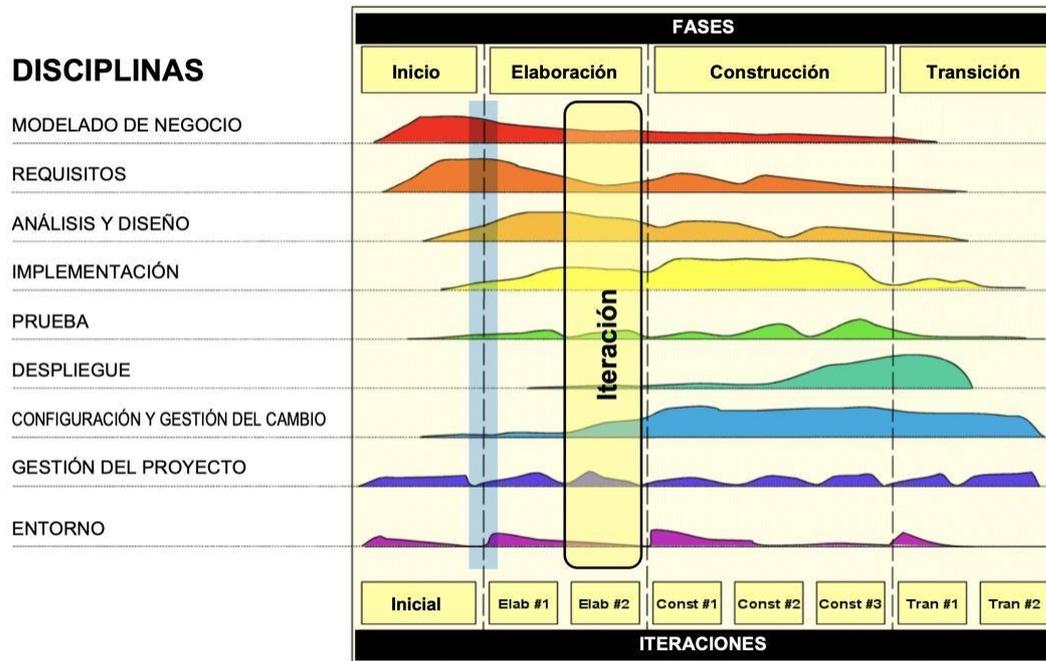


Ilustración 18 Esquema planificación temporal PU

Estimación de coste y esfuerzo

Para la realización de la estimación del coste y esfuerzo del proyecto se ha empleado la métrica de puntos de casos de uso. Esta técnica permite evaluar la funcionalidad representada en forma de casos de uso y así poder estimar el esfuerzo de desarrollo considerando actores, escenarios y factores técnicos y de entorno. Para ello se requiere del cálculo de tres variables:

- ◇ UUCP (Unadjusted Use Case Points)
 - UUCW (Unadjusted Use Case Weight)
 - UAW (Unadjusted Actor Weight)
- ◇ TCF (Technical Complexity Factor)
- ◇ ECF (Environment Complexity Factor)

Como punto a destacar en la presente memoria se puede resaltar la asignación de un menor número de horas a cada caso de uso (15 horas establecidas frente a las 20 horas por defecto) debido al mayor refinamiento en su elicitación. Además, aquellos factores técnicos asociados al conocimiento del lenguaje y del paradigma de programación móvil se han visto atenuados debido a que ya se poseía cierto conocimiento previo al haber recibido tanto formación como práctica (mediante el trabajo fin de grado) en ello.

Así pues, el resultado de este cálculo es de e 1011 horas, para un solo trabajador, es decir, tomando que el desarrollador trabaja de media 5 horas al día cada día de la semana natural, son 202,176 días naturales, que transformado a semanas resultan 28 semanas lo que equivale, contando 4 semanas por mes, 7 meses. Todo ello contando que no se toma la semana laboral sino la natural ni festivos ya que el estudiante podrá invertir horas en el proyecto sin seguir ningún convenio laboral.

En cuanto al precio que esto supondría 32221,80€, siguiendo los datos ofrecidos por Malt, es decir, una tarifa media de 31,875€ la hora para un desarrollador *Android freelance* (Malt, 2022). Este cálculo no es del todo real ya que este proyecto no solo cuenta con labores de desarrollo sino también de análisis, diseño, elicitación de requisitos, gestión de proyectos... En definitiva, áreas que no se ciñen solo a un puesto de desarrollador. Tampoco se tiene en cuenta en qué momento se realizan las horas (fines de semana, horario nocturno...) ni la categoría profesional de quien realiza el proyecto.

Todos estos detalles apoyados en una breve explicación teórica, así como valores concretos dentro del proyecto pueden verse detallados en el anexo I *Planificación temporal*. El cálculo de las horas de esfuerzo requeridas para el proyecto se ha calculado de forma automática introduciendo los valores explicados en el anexo en la herramienta EZEstimate.

Planificación temporal del proyecto

Planificar todas las tareas a realizar es el primer aspecto a tener en cuenta para el desarrollo de la planificación temporal, ya que su identificación marcará el tiempo necesario y el éxito del desarrollo del proyecto.

Como se puede observar en el diagrama de Gantt mostrado en las siguientes ilustraciones, cada una de estas tareas resulta imprescindible para la realización del proyecto. Un diagrama de Gantt es una representación gráfica de las tareas y subtareas que conforman el proyecto sobre una escala temporal. Las actividades se muestran en forma de barra y su longitud representa el tiempo de dicha tarea o subtarea.

Cabe destacar que la planificación temporal del proyecto difiere en una iteración respecto a lo marcado por defecto en el proceso unificado en la fase de inicio ya que la idea surge del propio alumno siendo una de las primeras experiencias de este en la proposición de unas líneas de actuación desde cero. Por ello esta diferencia en el número de iteraciones se prolonga a la fase de elaboración, contando esta con una más respecto al estándar, esto también es debido a una mayor fragmentación de los casos de uso y apartados en los que se ideó dividir el proyecto. Esto se arrastra también a la fase de construcción en la que se cuenta con cinco iteraciones frente a las tres del estándar. Además de lo comentado, se tomó así para poder abordar la integración de cada una de las partes con una granularidad más fina ya que esta representa el punto crítico para que la aplicación funcione en su totalidad.

Así, en resumen, en la primera etapa de inicio se abordan cuestiones asociadas a la elicitación de requisitos e ideación de la arquitectura que se deseaba seguir en el proyecto y que le diese forma estable. A su vez, se probaron funcionalidades similares en diferentes dispositivos para saber qué requerimientos mínimos (en cuanto a hardware y software) se deseaban exigir a los usuarios para que la aplicación final funcionase de forma fluida. En este punto cabe destacar que se descartó recorrer el camino de implementar una realidad aumentada mediante los sistemas ofrecidos por Google ya que estos requerían de versiones Android que, generalmente, la población actualmente, como desde el perfil de usuario como desde el perfil de desarrollador software, no maneja al tratarse de las últimas versiones.

En la siguiente etapa, elaboración, se centraron los esfuerzos en implementar pequeñas funcionalidades aisladas de la aplicación probando el rendimiento en el huésped seleccionado así como el estudio de la mejor forma para el almacenamiento persistente (aquí surgió la disyuntiva explicada en el manual del programador de elegir entre almacenamiento relacional no disponible gratuitamente pero más adaptado al contexto de la aplicación o un almacenamiento no relacional gratuito del que se poseía menor nivel de conocimiento).

En la tercera etapa, construcción, en cada una de las iteraciones se implementaron cada una de las grandes áreas de la aplicación (geolocalización, persistencia de datos y perfiles de usuario, realidad aumentada y ayuda al usuario) así como la integración por fases entre ellas (comenzando por la persistencia y geolocalización, continuando con estas y la realidad aumentada para terminar incorporando las ayudas al usuario). Cabe destacar que en esta etapa ya comenzaron pruebas con usuarios finales, bien a través de prototipos bien con funcionalidades aisladas, para obtener su opinión en las fases más tempranas de desarrollo.

Finalmente, en la etapa de transición se realizaron pruebas del conjunto de la aplicación con usuarios finales (detalladas en el anexo VI) así como pruebas de resiliencia para detectar hasta

qué versiones (superiores e inferiores) era la aplicación capaz de funcionar con fluidez, siendo la versión mínima la indicada en el manual del usuario y la máxima tres versiones superiores. Para más detalles sobre la planificación temporal se recomienda ver el Anexo I Planificación temporal. Se puede observar un ejemplo de las primeras tareas de la planificación temporal en la ilustración siguiente.

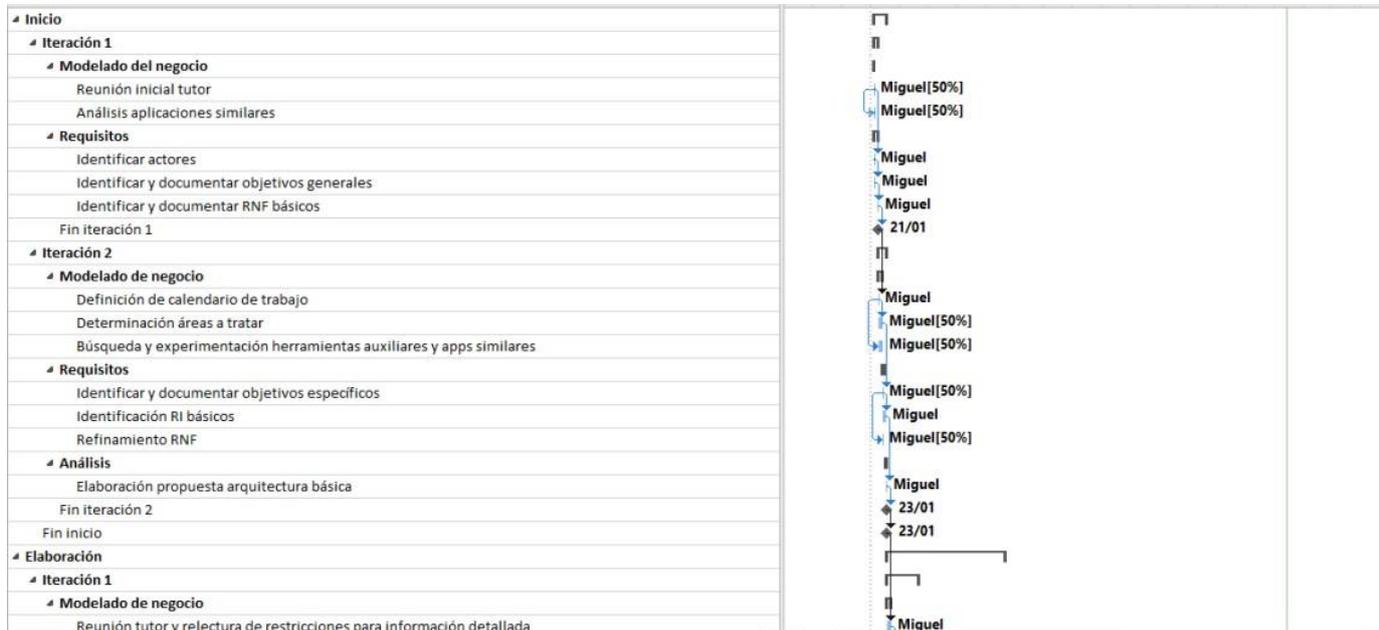


Ilustración 19 Planificación temporal

Especificación de requisitos

El documento de Especificación de Requisitos del Sistema, contenido en el Anexo II, se ha elaborado siguiendo la Metodología de Elicitación de Requisitos de Durán y Bernárdez. Con ella se estudian los requisitos del usuario y se crea un documento que debe ser aceptado tanto por el cliente como por el equipo de desarrollo.

El anexo está organizado de la siguiente manera:

- ◇ Roles en el proyecto
- ◇ Descripción del sistema
- ◇ Objetivos
- ◇ Requisitos funcionales
- ◇ Requisitos no funcionales
- ◇ Requisitos de información

- ◇ Casos de uso
- ◇ Matrices de rastreabilidad
- ◇ Propuesta de arquitectura

En los siguientes apartados se explica brevemente cada uno de los puntos anterior. Para un mayor detalle se recomienda acudir a los anexos.

Roles en el proyecto

El alumno, Miguel Cabezas Puerto, designado con los roles de desarrollador, cliente y usuario final. Los tutores, así como la hermana del desarrollador cubrirán el rol de clientes.

Descripción del sistema

Aplicación para dispositivos Android que pretende fomentar el turismo en una ciudad de una forma alternativa y llamativa, especialmente para generaciones jóvenes, tanto en población autóctona como foránea. Para ello se elaborará una propuesta de productos turísticos a través del descubrimiento de la ciudad basada en retos. Como caso de estudio, se pretende desarrollar en base a la propuesta una aplicación móvil para Android que permita descubrir diferentes puntos de la ciudad de Salamanca a través de diversos desafíos propuestos a toda la comunidad de jugadores.

Objetivos

Los objetivos del sistema son:

- ◇ Gestión de usuarios
- ◇ Gestión de puntuaciones
- ◇ Localización de puntos de interés
- ◇ Gestión de realidad aumentada

Requisitos del sistema

Los requisitos del sistema se dividen en requisitos de información, requisitos no funcionales y requisitos funcionales o casos de uso. Se mostrará un ejemplo de cada uno de ellos mostrando

la plantilla seguida. Para mayor detalle se recomienda consultar el Anexo II Especificación de requisitos.

Requisitos de información

Datos que el sistema debe contener. Así, debido a la naturaleza del proyecto donde únicamente se basa en el almacenamiento de datos de personas (IRQ-0001 Usuario) y de los desafíos que ha completado y existentes en el juego, se han establecido solamente dos, aquel relacionado con los datos del usuario que desempeña el rol de jugador en la aplicación y otro más relacionado con la información tanto retos y desafíos que va completando como de los existentes en el juego (IRQ-0002 Retos) compuesto de un nombre, unas coordenadas y alguna información de interés histórico del punto en el que se halla.

| | |
|--------------------------|--|
| IRQ-0001 | Usuario |
| Versión | 1.0 (01/03/2022) |
| Autores | Miguel Cabezas Puerto |
| Fuentes | |
| Dependencias | OBJ-0001 Gestionar puntuaciones OBJ-0002 Gestionar información de usuario OBJ-0003 Gestionar retos IRQ-0002 Reto NFR-0001 Privacidad de la información NFR-0002 Sistema operativo Android NFR-0003 Tiempo de aprendizaje NFR-0004 Tiempo de respuesta NFR-0005 Interfaces gráficas |
| Descripción | El sistema deberá almacenar la información correspondiente al usuario |
| Datos específicos | <ul style="list-style-type: none"> ● Correo ● Contraseña ● Nickname |

| | | |
|--------------------------------|--|---------------|
| | <ul style="list-style-type: none"> ● Foto de perfil ● Puntuación | |
| Tiempo de vida | Medio | Máximo |
| | 3 año(s) | 5 año(s) |
| Ocurrencias simultáneas | Medio | Máximo |
| | 1 | 1 |
| Importancia | | |
| Urgencia | | |
| Estado | | |
| Estabilidad | Alta | |
| Comentarios | | |

Tabla 1 IRQ-0001 Usuario

Requisitos no funcionales

Definen propiedades emergentes del sistema, tales como el tiempo de respuesta, las necesidades de almacenamiento, la fiabilidad, etc. Pueden especificar también la utilización de una herramienta CASE en particular, un lenguaje de programación o un método del desarrollo. Dado que se trata de una aplicación utilizada fundamentalmente por jóvenes y en la que se quiere que su dificultad técnica pase a un segundo plano cobran importancia aquellos requisitos no funcionales relacionados con el corto tiempo de aprendizaje (NFR-0003), el menor tiempo de respuesta posible de la aplicación (NFR-0004), la integración de interfaces gráficas sencillas y familiares (NFR-0005). Por otra parte, debido a que es una aplicación abierta a turistas propios como foráneos, debe incluir un soporte multi-idioma (NFR-0006).

A mayores, al tratarse de una aplicación únicamente para Android es lógico que este represente un requisito no funcional (NFR-0002).

Por último, debido al avance de la ciberdelincuencia y el valor creciente de los datos personales, es requisito que los introducidos por el usuario y que resultan más sensibles (en este caso la contraseña de su perfil) sean de completa privacidad (NFR-0001)

| | |
|---------------------|---|
| NFR-0003 | Tiempo de aprendizaje |
| Versión | 1.0 (01/03/2022) |
| Autores | Miguel Cabezas Puerto |
| Fuentes | |
| Dependencias | <p>OBJ-0001 Gestionar puntuaciones</p> <p>OBJ-0002 Gestionar información de usuario</p> <p>OBJ-0003 Gestionar retos</p> <p>OBJ-0004 Localizar puntos de interés</p> <p>IRQ-0001 Usuario</p> <p>IRQ-0002 Reto</p> <p>NFR-0004 Tiempo de respuesta</p> <p>NFR-0005 Interfaces gráficas</p> <p>NFR-0006 Multi-idioma</p> |
| | |
| Descripción | El tiempo de aprendizaje del manejo del sistema por un usuario deberá ser menor a una semana. |
| Importancia | Importante |
| Urgencia | Hay presión |

| | |
|--------------------|-----------------|
| Estado | En construcción |
| Estabilidad | Alta |
| Comentarios | |

Tabla 2 NFR-0003 Tiempo de aprendizaje

Casos de uso

El proyecto centra sus casos de uso en cuatro grandes áreas. La primera de ellas es la gestión de los retos, esto quiere decir, su ubicación sobre los mapas, el registro del tiempo invertido en ellos, su modo de detección y el modo en que se inicia y se detienen sus tiempos. Para cada uno de estos escenarios se plantea un caso de uso.

El segundo gran bloque es el de los usuarios. Este se centra en una gestión CRUD de su información. Así se elicitán casos de uso para el registro, autenticación, cierre de sesión y actualización de la información de usuario asociada sus datos personales (nombre de usuario, contraseña...). Queda fuera de este CRUD el borrado de usuarios ya que en una primera versión se encargaría el desarrollador, a modo de soporte, de borrarlos manualmente. Con esto se pretende tener un mayor control de los usuarios activos hasta lanzar el producto a una segunda fase más extensiva.

En tercer lugar, el bloque de ayuda al usuario, en este se detallan los casos de uso vinculados al envío de correos y la organización de las preguntas frecuentes.

Por último, derivado de los retos, dado que se ha llevado un nivel de granularidad más fino en la elicitación de requisitos, se encuentra la gestión de las puntuaciones. Esto es la posibilidad de consultar las puntuaciones globales de la que dispone el usuario y todas las acciones que se disparan (recuento de puntuaciones individuales, ordenación, actualización...), para ello se crea un caso de uso vinculado (UC-0011)

Se muestra como ejemplo el caso de uso asociado al registro de un nuevo usuario. Para más detalle se recomienda ver el anexo II

| | |
|---------------------|---|
| UC-0006 | Registrar usuario |
| Versión | 1.0 (01/03/2022) |
| Autores | Miguel Cabezas Puerto |
| Fuentes | |
| Dependencias | OBJ-0002 Gestionar información de usuario IRQ-0001 Usuario |

| | | |
|-------------------------|--|---|
| | NFR-0001 Privacidad de la información NFR-0002 Sistema operativo Android NFR-0003 Tiempo de aprendizaje NFR-0004 Tiempo de respuesta NFR-0005 Interfaces gráficas NFR-0006 Multi-idioma | |
| Descripción | El sistema debe permitir el registro de un nuevo usuario | |
| Precondición | Ninguna | |
| Secuencia normal | Paso | Acción |
| | 1 | Un actor de tipo ACT-01 (<i>Usuario</i>) selecciona la opción de registrar nuevo usuario |
| | 2 | El sistema solicita la información al usuario, en particular la indicada en el IRQ-0001 (<i>Usuario</i>). |
| | 3 | El usuario introduce los datos correspondientes |
| | 4 | El sistema almacena los datos. |
| | 5 | El usuario confirma el registro. |
| Postcondición | Ninguna | |
| Excepciones | Paso | Acción |
| | 3 | Si el sistema detecta algún error en la validez de los datos volverá al paso 2. |
| Estabilidad | Alta | |
| Comentarios | Ninguno | |

Tabla 3 UC-0006 Registrar usuario

Actores

Los actores que desempeñarán los casos de uso serán el usuario y el sistema.

Matrices de rastreabilidad

Permiten establecer las relaciones entre los requisitos y los objetivos, así como entre requisitos, de una forma visual. El sistema es altamente desacoplado entre casos de uso, es decir, cada caso de uso guarda relación a lo sumo con otro (generalmente de su mismo bloque),

esto se ha pensado así para ganar independencia entre funcionalidades desde el primer momento y poder disponer de ellas incluso en el caso que el resto fallase. Por otra parte, el sistema es altamente dependiente de los requisitos no funcionales ya que, debido al carácter social y para usuarios no técnicos de la aplicación, se valora que se pueda usar de una forma rápida tanto de aprendizaje como de respuesta por parte del sistema. Esto incentivará a su recomendación a terceros.

Todos estos detalles se pueden ver recogidos en el Anexo II

Propuesta de arquitectura

Divide al sistema en los cuatro grandes paquetes ya detectados en los casos de uso, pero con un mayor nivel de abstracción y centrándose en una funcionalidad más técnica. Así, dentro del paquete “Gestionar información/datos” estarían contenidos los casos de uso detallados anteriormente relacionados con el almacenamiento y recuperación de la información acerca de los retos y usuarios. Dentro del paquete “Gestionar conexiones externas” se englobaría la consulta a servidores externos de puntuaciones y el envío de correos. En el paquete “Gestionar mapas” se englobarían los casos de uso relacionados con la muestra de los retos sobre mapas. Por último, en el paquete “Gestionar configuraciones” se incluirían los casos de uso relacionados con la configuración del usuario

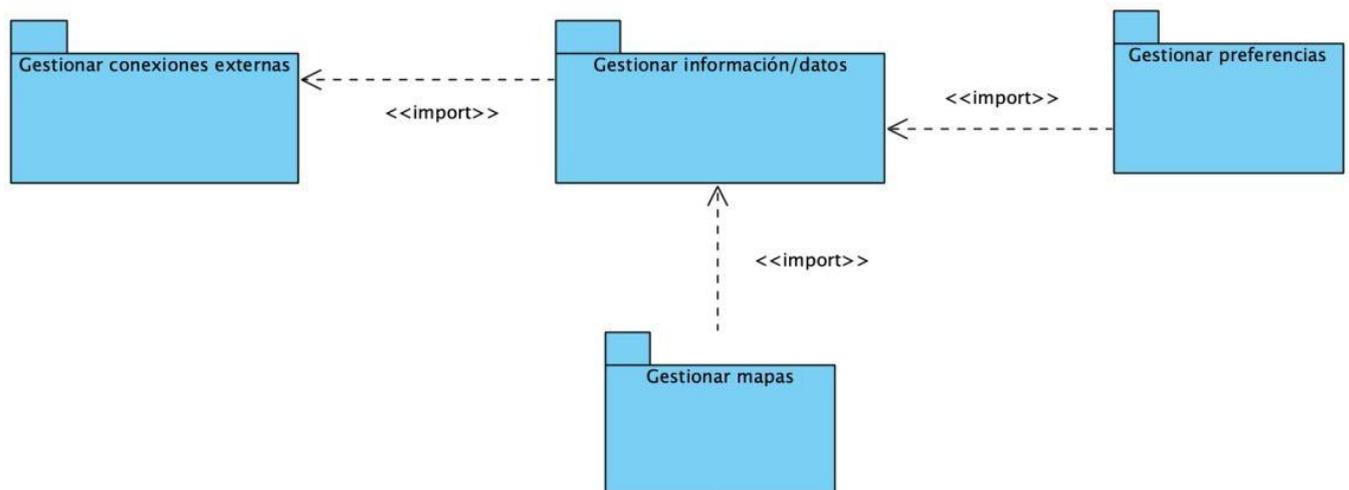


Ilustración 20 Propuesta de arquitectura

Análisis

En el anexo III Análisis podemos encontrar la primera representación técnica del sistema mediante la mezcla de formatos en texto y diagramas con el objetivo de representar los requisitos del software, las funciones y el comportamiento.

Este anexo cuenta con:

- ◇ Modelo de dominio
- ◇ Vista de interacción
- ◇ Propuesta de arquitectura

Modelo de dominio

Formado por un diagrama de clases y un glosario. Este último contiene información relativa a las clases y atributos que componen el diagrama.

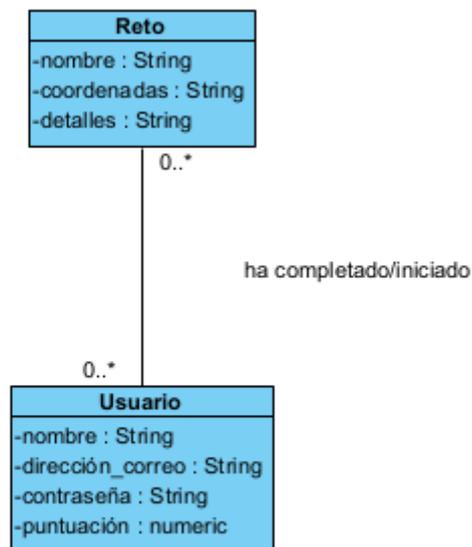


Ilustración 21 Diagrama de clases

Partiendo de los requisitos de información, se establece un sistema en el que el usuario y los retos se relacionen. Esto es así ya que en la aplicación final se desea que el usuario pueda tener una serie de retos asociados (aquellos que ha iniciado, que ha completado, que aún no empezó...), a su vez, esta lista de retos existentes podrá pertenecer a varios usuarios ya que se trata de un sistema competitivo. En cuanto a los atributos de cada clase, se basan en los que se requirieron a través de los requisitos de información llevándolos a un nivel de detalle mayor en

el caso del usuario. A modo de ejemplo se muestra la definición de la clase Usuario. Representa la información en la aplicación de la persona que la utiliza y está registrada.

- ◇ nombre. *Nickname* del usuario. Es una cadena de texto alfanumérico
- ◇ contraseña. Clave de acceso a la aplicación. Es una cadena de texto.
- ◇ correo electrónico. Dirección email vinculada a la cuenta. Es una cadena de texto alfanumérica.
- ◇ puntuación. Media del tiempo invertido por el usuario en los retos completados. Es un valor numérico

Vista de interacción

Representa cómo interaccionan los objetos que forman el sistema previamente definido.

Esta pretende representar, en un primer nivel, los flujos que la aplicación final seguirá. Así pues, ya que posteriormente se parte de un modelo MVC, se representan en los diagramas aquellos componentes que reaccionan cuando el usuario lanza una acción. Para este proyecto los flujos recogidos serán:

- ◇ Inicio y parada del tiempo: estos flujos serán disparados cuando el usuario, a través de la interfaz quiera pausar o reanudar alguno de los retos existentes.
- ◇ Guardado del tiempo: este flujo dependerá de los anteriores, en concreto de la parada de tiempos. Es por ello que es lanzado por el propio sistema.
- ◇ CRUD de usuarios: este conjunto de flujos reaccionará siempre a una acción lanzada por el usuario asociada a cada funcionalidad (registro, actualización, consulta de perfil, autenticación y cierre de sesión). Es por ello que el flujo comenzará en la interfaz para finalmente afectar a la entidad de la modelo asociada a la información del usuario.
- ◇ Mostrado de retos en el mapa: dentro de la gestión de los retos, cobra un papel importante el poder visualizarlos para su posterior descubrimiento. Este punto, en este nivel, es el asociado al inicio del juego con uno de los retos, por ello surge como respuesta a la interacción del usuario con la interfaz.
- ◇ Consulta de puntuaciones: flujo asociado al cuatro gran bloque descrito en los casos de uso. De nuevo consecuencia de la interacción del usuario con la interfaz

Todos ellos se presentarán mediante diagramas de secuencia como el mostrado a continuación

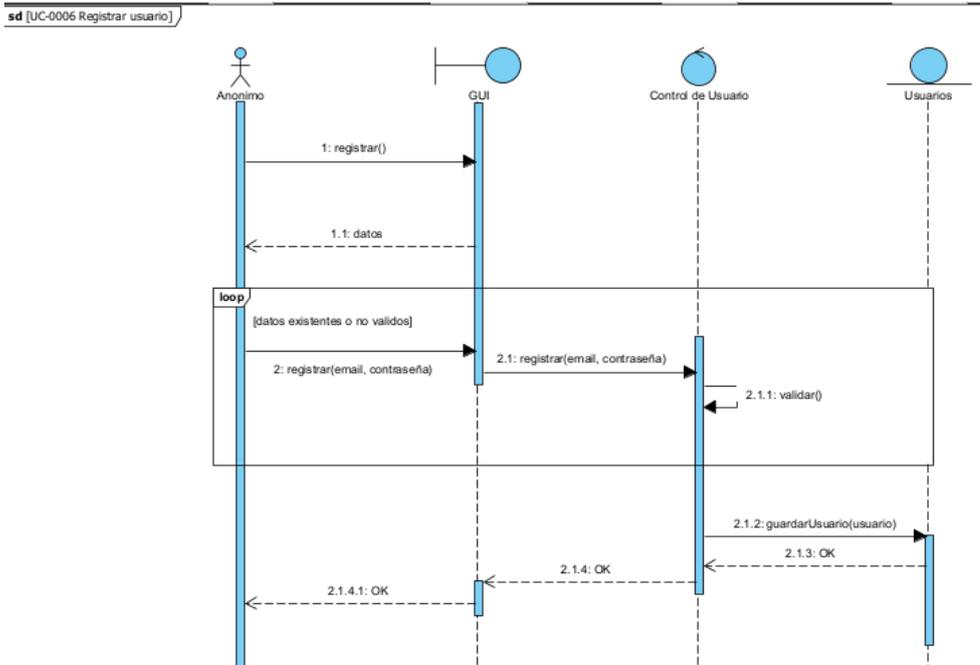


Ilustración 22 Diagrama de secuencia

Propuesta de arquitectura

Muestra de qué manera interactúan los objetos interfaz, control y entidad. Se deriva de las diferentes interacciones entre los controladores y entre estos las interfaces y los modelos. Así se puede ver que aparece un paquete respecto a la propuesta arquitectónica anterior. Esto es debido a que se detecta la necesidad de comprobar si el dispositivo cuenta con conexión a la red para realizar el resto de las funcionalidades ya que estas deben registrarse en remoto para que la aplicación cumpla con su carácter de tiempo real (bajo tiempo de respuesta) y global. A mayores podemos observar qué retos y usuarios, para la vinculación de información representada en el diagrama de clases, deben interactuar a través de sus controladores. De igual modo ocurre entre los retos y el controlador encargado de situarlos sobre el mapa. Se ha decidido desacoplar por si en un futuro se quisiesen geolocalizar otras entidades que no fuesen retos. Por último, destacar que el paquete de ayuda queda parcialmente desvinculado del resto al no tener ninguna dependencia (ya mostradas en las matrices de rastreabilidad) respecto a ellos tanto en información como en lógica.

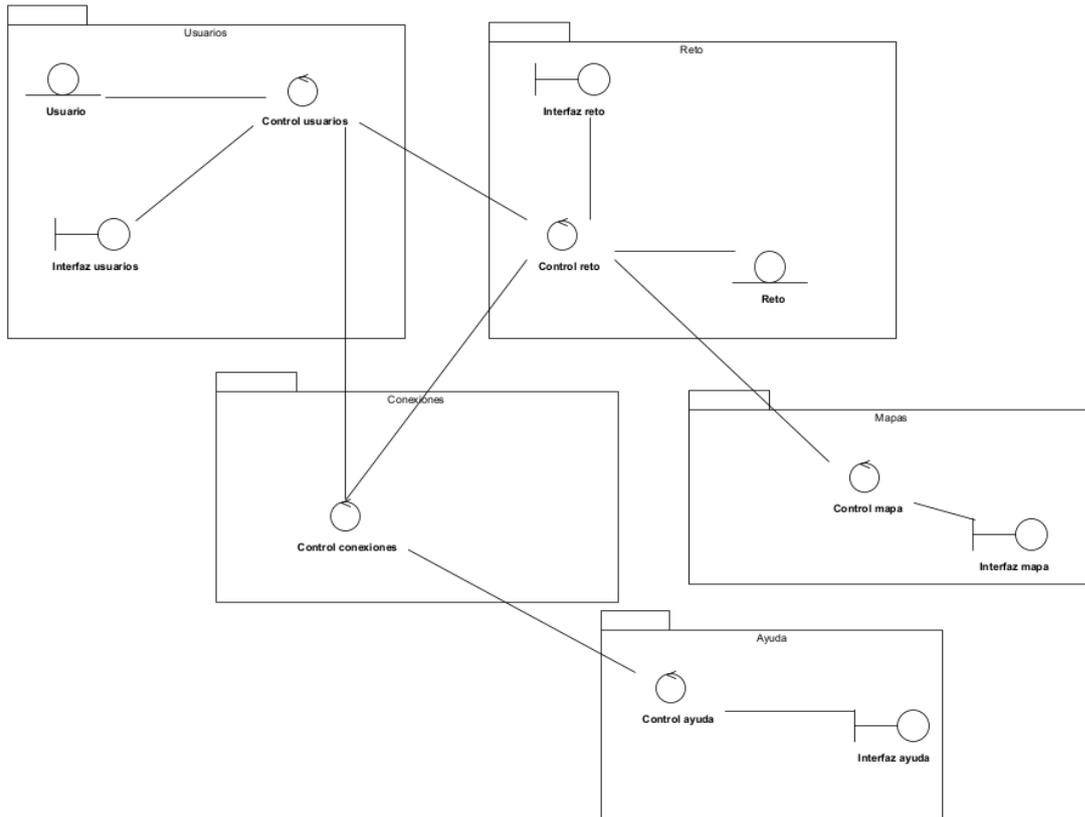


Ilustración 23 Propuesta arquitectura análisis

Diseño

En el anexo IV *Diseño* podemos encontrar la búsqueda de soluciones a los requisitos de usuario de cara a la implementación. Es decir, se muestra el paso del proyecto desde el dominio del problema que se trata en el análisis al dominio de la solución. En esta fase están contenidos:

- ◇ Subsistemas de diseño
- ◇ Clases de diseño
- ◇ Realizaciones de caso de uso-diseño
- ◇ Arquitectura de capas
- ◇ Modelo de despliegue
- ◇ Diseño de datos
- ◇ Diseño de interfaces

Cabe destacar que para el diseño se han empleado el patrón MVC (Modelo- Vista- Controlador). Se ha elegido así ya que además de ser el patrón propuesto para el desarrollo

móvil, permite tener múltiples vistas sincronizadas del mismo modelo, punto crucial de la aplicación ya que debe mostrar información actualizada y en tiempo real tanto propia como del resto de jugadores. Además, gracias a Android se supera el aumento de la dificultad técnica en el desarrollo ya que las vista se realizan a través de ficheros XML que indican cuál debe ser el aspecto de la aplicación y mayormente los eventos a los que reaccionar, dejando los ficheros Java para la implementación del modelo y controlador

Por otra parte, se emplea el patrón DAO (*Data Access Object*). Con él se consigue encapsular las consultas y modificaciones de los modelos, se ha realizado así para liberar a los controladores de la lógica de acceso a base de datos, es decir, para los controladores resultará transparente el motor de almacenamiento subyacente de tal forma que si este pasase de relacional a no relacional o viceversa, solo habría que modificar los DAO logrando con ello un mayor nivel de desacoplamiento y menor dependencia entre los modelos de dato y la lógica de negocio. Para un mayor detalle se recomienda acudir al anexo IV Diseño.

Arquitectura de capas

Muestra los artefactos más relevantes del modelo de diseño a través de carpetas distribuidas en capas o niveles diferentes. Como se puede observar en ella se integran los paquetes detallados en las anteriores propuestas arquitectónicas junto con los patrones arquitectónicos elegidos. Así todos los paquetes contarán con una implementación de controladores y a su vez la información de los modelos de los diferentes paquetes será aislada mediante el acceso a través de DAO.

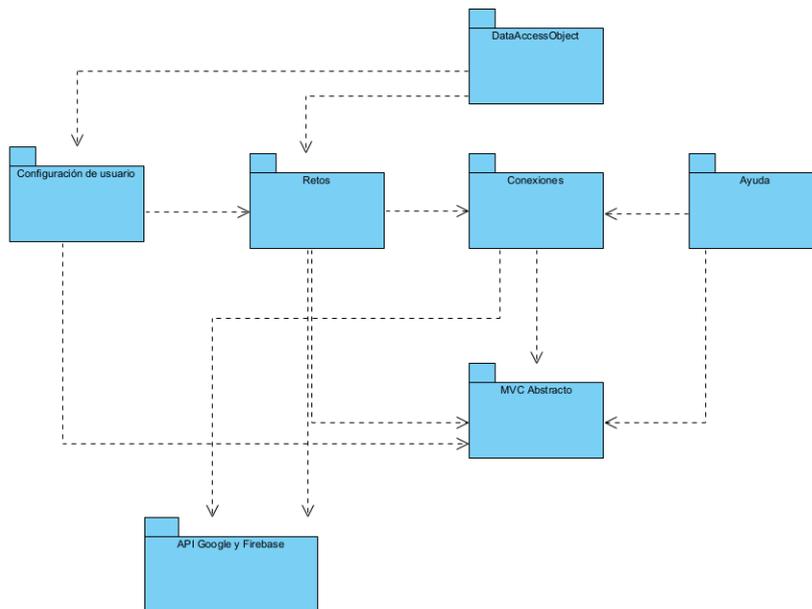


Ilustración 24 Arquitectura de capas

Subsistemas de diseño

Trata de organizar los artefactos que vamos a tratar en el diseño en piezas más manejables. En este apartado se detallan de forma teórica y práctica los patrones empleados.

La arquitectura del proyecto sigue los patrones MVC y DAO explicados en profundidad en el anexo III. En las siguientes ilustraciones se puede observar el patrón abstracto seguido. Se recomienda ver el anexo III para su desarrollo

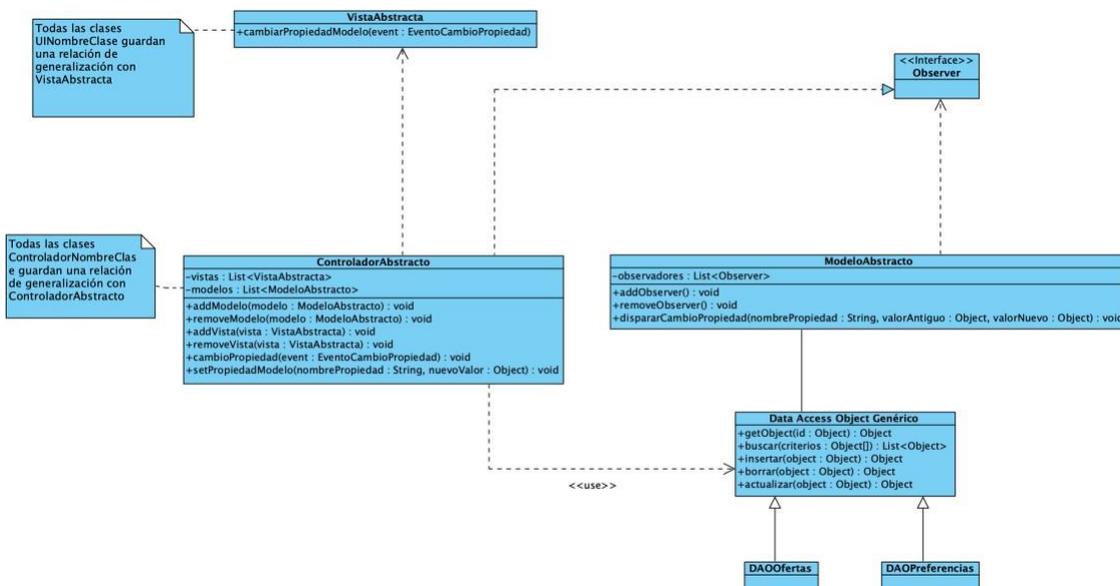


Ilustración 25 Subsistema de diseño

Clases de diseño

Formado por un diagrama de clases de diseño que, a diferencia del modelo de análisis, contendrá las operaciones y atributos necesarios para dar el salto al nivel de representación. A modo de ejemplo se puede observar uno de los paquetes de clases de diseño, así como la definición de una de estas clases.

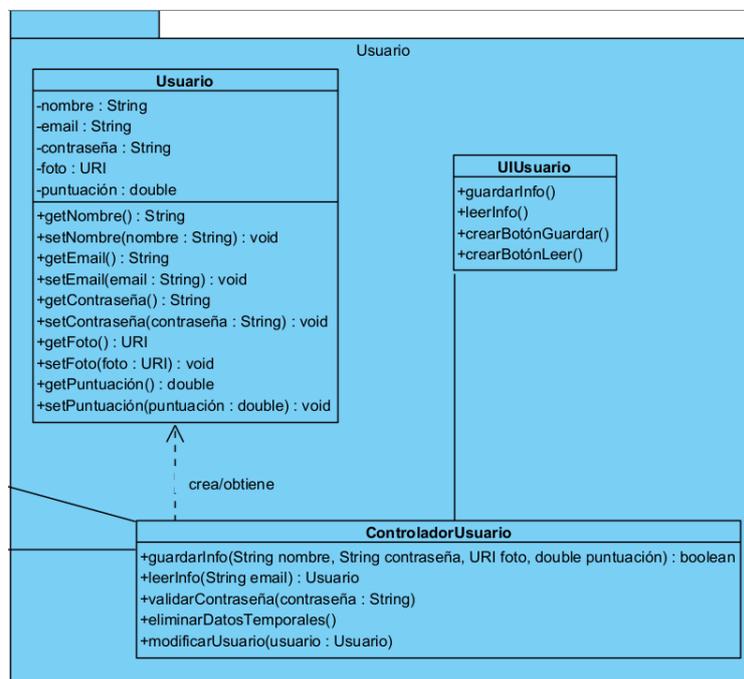


Ilustración 26 Clase de diseño Usuario

Realizaciones de caso de uso-diseño

Se trata de una colaboración en el modelo de diseño, es decir, cómo se realiza un caso de uso específico. Esto se modelará a través de los diagramas de actividad, de estado y de secuencia de dichos casos de uso. En la siguiente ilustración se observa el caso de uso relacionado con registrar un usuario

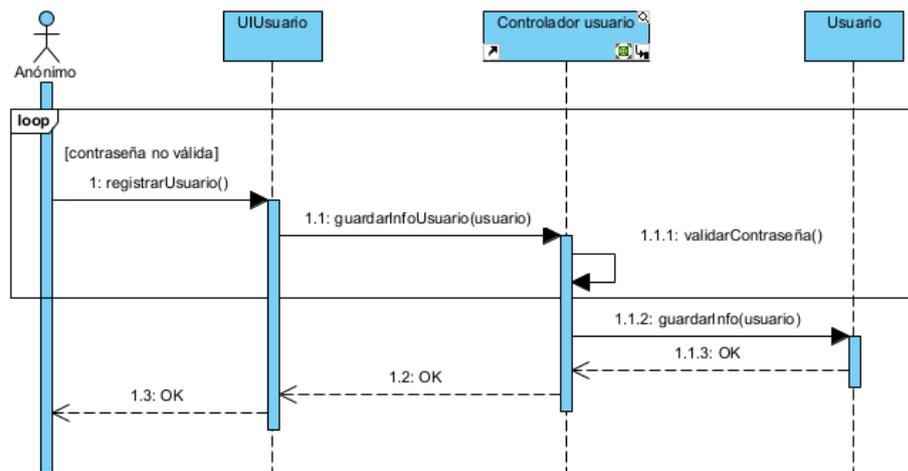


Ilustración 27 Realización CU diseño registrar usuario

Modelo de despliegue

Describe la distribución física del sistema en términos del modo de distribución de la funcionalidad entre los nodos. Así pues, al tratarse de una aplicación móvil con un servidor externo no gestionado propiamente, contará con un único nodo propio contenedor de la aplicación y toda su lógica, conectado a otro nodo gestionado por terceros y ya levantado. De igual manera, este nodo externo puede recibir peticiones de varios nodos propios, cada uno de ellos representando un dispositivo móvil con la aplicación corriendo.

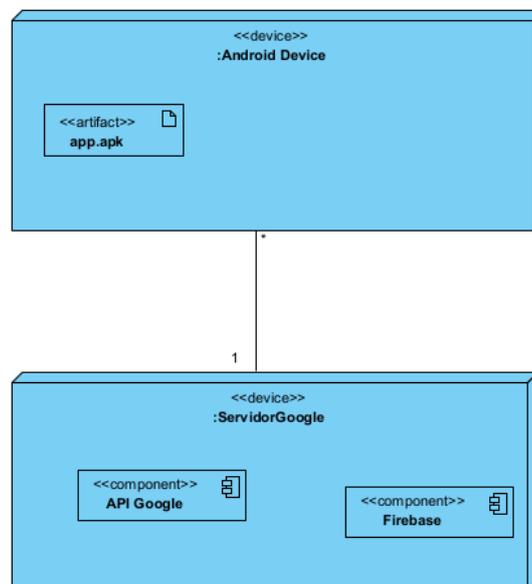


Ilustración 28 Modelo de despliegue

Diseño de datos

Detalla cómo se han almacenado y tratado los datos requeridos en los requisitos y desarrollados en el resto de los modelos. Así, como resulta natural, existen dos clases, asociadas a la información que se desea manejar, una para usuarios y otra para retos. Como la relación entre ellas es múltiple, ya que un usuario puede tener en su haber (en curso o terminados) diversos retos y estos retos a su vez pueden aparecer vinculados a varios usuarios (al tratarse de un juego competitivo), aparece una clase intermedia que recoja todas esas posibles combinaciones.

Se recomienda acudir al anexo III para una mayor información acerca de los atributos de cada clase, a modo ilustrativo se muestra el diagrama entidad-relación.

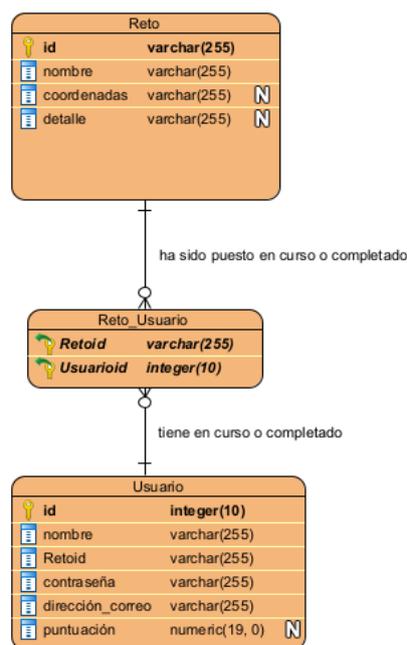


Ilustración 29 Diagrama entidad-relación

Diseño de interfaces

Indica el diseño elegido para el desarrollo de las interfaces y vistas de usuario mediante el uso de mockups desarrollados con el software MarvelApp.

Se ha tratado de un proceso iterativo alimentado por las opiniones de la gente en las diversas pruebas de interfaz de usuario tanto con prototipos funcionales como únicamente estéticos. Esto se realizó así ya que en los propios requerimientos no funcionales del proyecto aparecía que su interfaz fuera sencilla de utilizar consiguiéndose esto únicamente a través de la prueba con personas reales ajenas al proyecto

Fondos y colores

Los colores que emplea la aplicación constituyen uno de los primeros filtros para el usuario. Los tonos empleados indican qué se quiere transmitir con la aplicación y simbolizan uno de los focos más influyentes sobre la interpretación del usuario.

Se ha hecho uso de tonalidades rojizas y granates para el diseño de la aplicación.

Según la teoría y psicología del color, el granate simboliza la pasión, el atrevimiento, la energía y la estimulación (Canva, 2022), sensaciones que se desean potenciar para así fomentar la competitividad dentro de la aplicación que conllevaría un mayor uso y un disfrute del turismo alternativo obedeciendo la ley de consumo rápido y competición que impera en las sociedades de hoy en día. Además, normalmente se emplean dichas tonalidades para estimular a las personas a tomar decisiones rápidas lo cual refuerza la idea anteriormente expuesta.

Disposición de elementos

Basándose en la distribución en aplicaciones estándares para Android, experiencias de usuarios en dichas aplicaciones y opiniones de gente a lo largo del proceso de desarrollo, se ha ido modificando la interfaz para finalmente quedar constituido el diseño por los elementos que posteriormente se detallan.

Cabe destacar que la pantalla de visualización del reto sobre el mapa no se diseña ya que se ha tomado como decisión que se visualice sobre la aplicación externa Google Maps. Esto es así para que la curva de aprendizaje del usuario sea menor ya que este ya dispone de dicha aplicación en el móvil, posiblemente la haya usado con anterioridad por tanto los controles, así como la estética le resultarán familiares.

De igual modo, la pantalla de escaneo del código QR no se diseña ya que se abrirá la cámara del propio dispositivo volviendo a la aplicación una vez detectado y analizado el QR.

Pantalla de login y registro

La pantalla de login y registro tratará de ser lo más parecida al estándar seguido por las aplicaciones Android, esto se hace así para que la curva de aprendizaje del usuario sea menos costosa al estar ya familiarizado debido a experiencias pasadas.

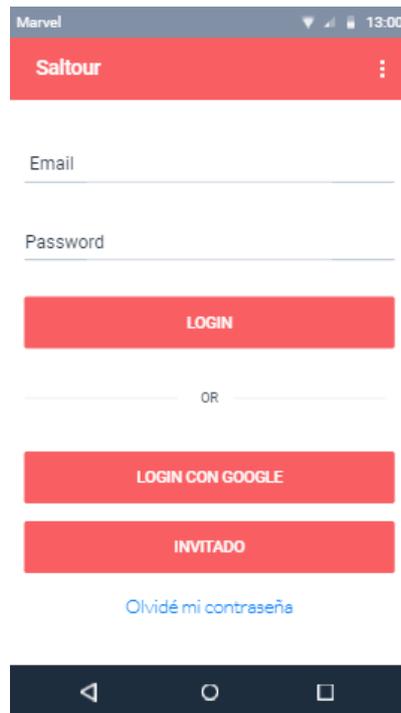


Ilustración 30 Pantalla de login y registro

Pantalla principal

Se ha elegido un formato simple y a través de un gráfico que crezca mediante animación para dar en un simple vistazo toda la información general de una forma estética y llamativa para el usuario.

Tanto en esta como en el resto de pantallas se ha elegido un menú lateral hamburguesa desplegable común en aplicaciones Android y con el que se ocupa un tamaño ínfimo de pantallas, evitando crear otra pantalla al efecto a la que tendría que navegar continuamente el usuario para visualizar las diferentes funcionalidades de la aplicación. Así, mediante el despliegue, se ahorran pasos y tiempo de navegación que podría hacer que el usuario abandonase la aplicación.

En este menú lateral, se primará la agrupación de funcionalidades similares, separado cada bloque por cierta distancia, para así facilitar su distinción al usuario.

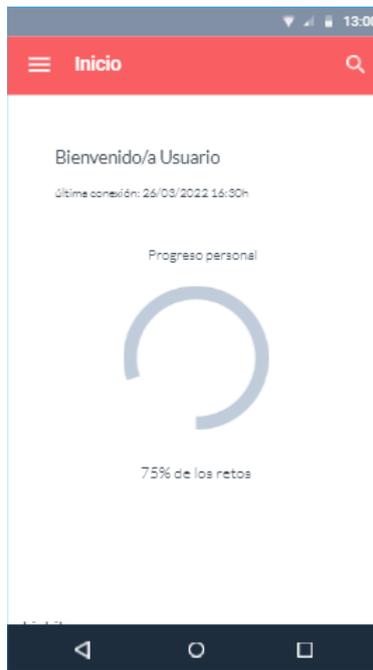


Ilustración 31 Pantalla principal

Pantalla olvido de contraseña

Se ha decidido tener un único cuadro de texto donde introducir el correo de recuperación y un botón ya que cuando el usuario acuda a esta pantalla es probable que no esté tranquilo debido a que ha olvidado su contraseña. Por ello los estímulos recibidos tienen que ser pocos y directos.

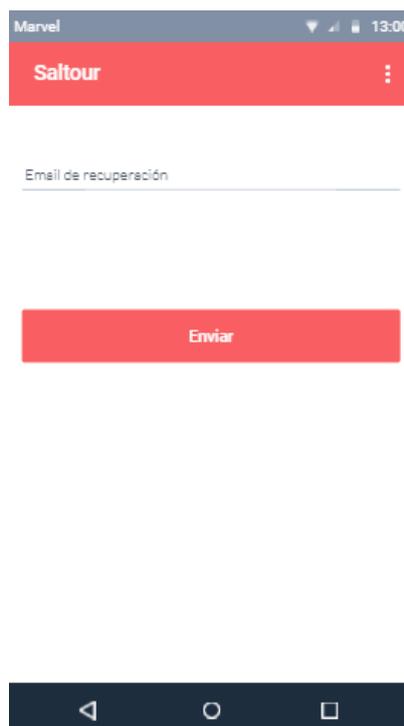


Ilustración 32 Pantalla olvido de contraseña

Pantalla jugar

Un listado con todos los retos (nombre y foto descriptiva). Al pulsar en cada uno de ellos se comenzará a jugar a encontrarlo activando el conteo de tiempo. El color de los retos dependerá de su grado de compleción, asignando una escala (verde, amarillo, sin color) para representar retos completados, en curso o sin empezar. Se han seleccionado estos colores ya que generalmente se utilizan para estos fines.

Respecto a los botones, el asociado a detener el tiempo, así como el de escanear QR únicamente aparecerán si el usuario tiene un reto con conteo de tiempo activo. Así por una parte se evita un mal uso de la aplicación por sobrecarga de información innecesaria.

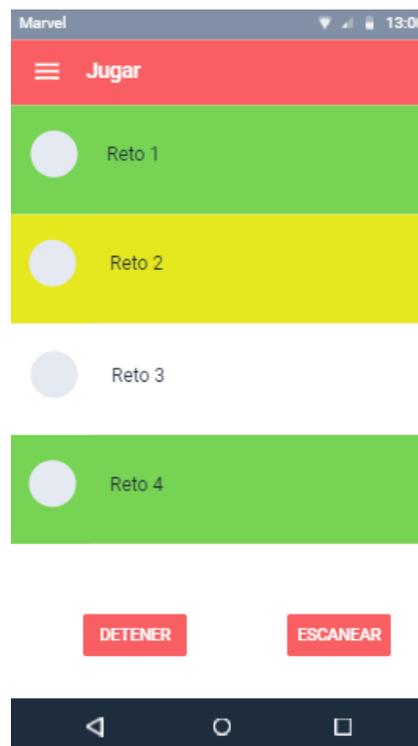


Ilustración 33 Pantalla jugar

Pantalla perfil de usuario

Formada por elementos editables al pulsar sobre ellos (nombre de usuario, contraseña y foto de perfil). Esta información es ampliable con tantos parámetros e información se recopile en la aplicación acerca del usuario.

Disposición lineal para que el usuario pueda seguir un paso tras otro en la edición de su perfil.

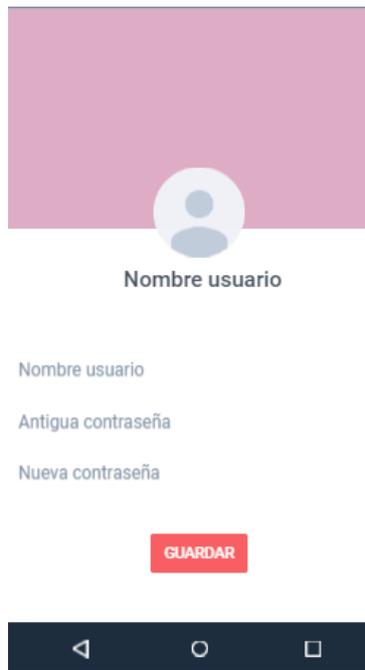


Ilustración 34 Pantalla perfil de usuario

Pantalla de estadísticas

Pantallas separadas en función de su alcance: una para el progreso individual y otra para clasificaciones generales. Respecto al progreso individual los tics se automarcarán a medida que el usuario vaya completando retos no siendo estos editables por su parte.

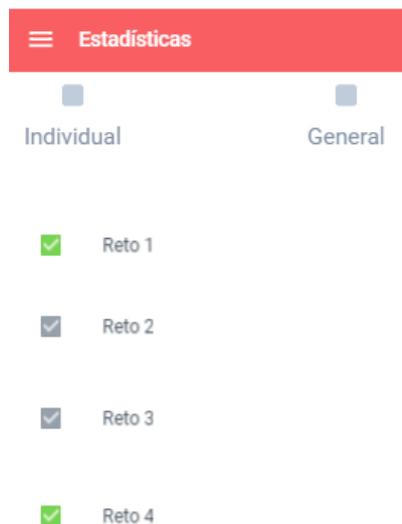


Ilustración 35 Pantalla estadísticas individuales



Ilustración 36 Pantalla de estadísticas globales

Pantalla de ayuda

Pese a que no aparezca en la ilustración, cada una de las preguntas consiste en un desplegable en el que el encabezado es la propia pregunta y su contenido la respuesta. Además, se situará un *link* en la parte inferior para contactar con los desarrolladores vía correo electrónico en caso de que el usuario tenga alguna duda no recogida en el momento en este apartado. Cabe destacar que, siguiendo las recomendaciones para desarrolladores de Android, se suprimirán en todas las pantallas el botón de volver dejando esta funcionalidad en manos de los botones de retorno ya disponibles en los dispositivos.

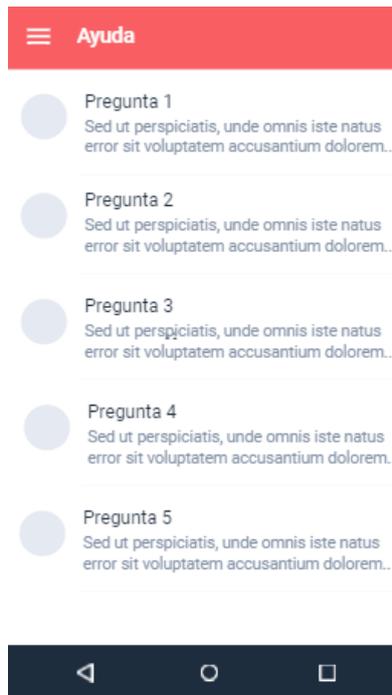


Ilustración 37 Pantalla de ayuda

Pantalla de realidad aumentada

No se adjunta imagen debido a que el software encontrado y gratuito no dispone de dicha opción. Si embargo, en las decisiones de diseño se incluye que el modelo 3D que represente la guía turístico contenga colores vivos para que así creen contraste con el fondo que generalmente será la propia realidad por lo que sus colores variarán en función del momento del día, la cantidad de iluminación, la calidad de la cámara del dispositivo... Además, las indicaciones mediante texto se situarán sobre un fondo blanco para asegurar que se distingan del fondo y en un tamaño de letra 14 para que sea visible en un dispositivo móvil sin necesidad de forzar la vista, pero a la vez no se salga de los límites de una pantalla estándar.

Tipografía

Al igual que el color, la tipografía también despierta sensaciones en los lectores, por lo que, de la misma forma, juega un papel importante en la aplicación.

Tanto para la parte desarrollada en Android (Roboto) como para la parte de Unity (Arial), se han elegido tipografías sans-serif. El motivo principal radica en que son las mejores para textos cortos y están especialmente indicadas para visualizaciones en pantallas. Además, transmiten seguridad y alegría (Flores, s.f.), que son sensaciones buscadas mediante el uso de nuestra aplicación.

Manual del programador

En el anexo V se detallan los aspectos relevantes para una mejor comprensión del código implementado, así como matices que se han modificado en la implementación respecto al diseño. Además, se explican las diversas pruebas realizadas para una optimización del sistema.

Descripción del funcionamiento de la aplicación

El funcionamiento de la aplicación, así como cada uno de los flujos disponibles en esta puede verse recogido mediante imágenes y explicaciones a través del manual para el usuario contenido en el anexo VII y explicado posteriormente. A modo de resumen cuando el usuario inicia la aplicación se comprueba si está logueado en la misma, en caso contrario se le pide bien el proceso de autenticación bien el proceso de registro en caso de que así lo desee. El proceso de registro se realizará en dos pasos, uno primero en el que deberá cumplimentar un formulario con su correo y contraseña válidos para posteriormente recibir un código de confirmación en dicho correo con el que pasará a formar parte del sistema. Una vez dentro del sistema podrá autenticarse. Pasada la autenticación aparecerá la pantalla de inicio del usuario con su porcentaje de retos completados, así como fecha de última conexión e imágenes decorativas de diversos puntos de la ciudad. A mayores, dispondrá de un menú lateral con diferentes funcionalidades como personalizar su perfil, consultar preguntas frecuentes, ver la clasificación y datos de otros participantes, así como el grueso de la aplicación: encontrar mediante guía los diferentes puntos de interés cronometrando el sistema el tiempo desde el inicio del reto hasta su cumplimentación (marcada por el escaneo de un código QR) o por parada voluntaria del tiempo por parte del usuario.

Aspectos relevantes del desarrollo

En el Anexo V se detallan todos los aspectos del desarrollo de la aplicación, se recomienda acudir a él para una mayor profundidad (Gómez, s.f.) (Google Developers, s.f.) (Google Developers, s.f.) (StackOverflow, s.f.). Entre ellos se puede destacar:

- ◇ Uso de Firebase Firestore como almacenamiento de datos en un modelo no relacional clave-valor.
- ◇ Uso de la API de Google para la geolocalización de retos. Para ello es necesario autenticarse con una cuenta de Google en su consola, crear unas credenciales para el proyecto e implementar los métodos requeridos para la ubicación de coordenadas.

- ◇ Uso de XML y recursos de Android para la interfaz. La interfaz se ha diseñado a través de diversos ficheros XML asociados a actividades Java con el rol de Vista. Para una mayor simplicidad a la hora de aplicar cambios sobre la Vista se han empleado los recursos de Android, es decir, los parámetros asociados a dimensiones, contenido estático multi-idioma de la aplicación, colores e imágenes, se recogen en ficheros XML dependientes de esos parámetros para que así las modificaciones pertinentes resulten de inmediata aplicación.
- ◇ Uso de Unity con Vuforia para el desarrollo e integración de la realidad aumentada en la aplicación.

Casos de estudio relevantes

A continuación, se exponen tres flujos que por su mayor complejidad técnica se deben destacar en este proyecto como son el registro de nuevos usuarios, detención de tiempos y la interacción con la realidad aumentada

Registro de usuarios

Este flujo presenta dos variantes, una primera en la que los usuarios mediante correo y contraseña se pueden registrar en la aplicación teniendo que verificar la dirección email dada y una segunda opción en la que haciendo uso de su cuenta de Google pueden acceder a la aplicación sin necesidad de verificar su correo ni indicar ninguna contraseña.

Registro con usuario y contraseña

El flujo se dispara, como se explica en el manual de usuario, cuando el usuario pulsa el botón de registro. Una vez pulsado se la redirige a una pantalla donde debe introducir una dirección de correo y una contraseña.

La primera comprobación que se hace es que tanto la dirección de correo como la contraseña no estén vacíos y esta última contenga al menos 8 caracteres, de lo contrario se volverá a pedir al usuario rellenar los campos de registro.

```

if(usuario_text.isEmpty()){
    Toast.makeText(context,"You must provide an email",Toast.LENGTH_LONG).show();
    usuario.setBackgroundResource(R.drawable.borderojo);
    return;
}
if(contrasena_text.isEmpty()){
    Toast.makeText(context,"You must enter a password",Toast.LENGTH_LONG).show();
    contrasena.setBackgroundResource(R.drawable.borderojo);
    return;
}
if(contrasena_text.length()<8){
    Toast.makeText(context,"The new password must contain at least 8 characters",Toast.LENGTH_LONG).show();
    contrasena.setBackgroundResource(R.drawable.borderojo);
    return;
}
}

```

Ilustración 38 Validación email y contraseña

A continuación, se pasa a validar que la dirección de correo sea válida a través de una expresión regular provista por Java consistente en que el texto contenga entre 1 y 156 caracteres alfanuméricos, el signo de arroba y posteriormente entre 1 y 64 caracteres alfanuméricos seguidos de un punto y finalizar con entre 1 y 25 caracteres alfanuméricos. De no ser así, se vuelve a pedir los datos al usuario.

```

if(!validarEmail(usuario_text)){
    Toast.makeText(context,"Invalid email",Toast.LENGTH_LONG).show();
    usuario.setBackgroundResource(R.drawable.borderojo);
    return;
}

```

Ilustración 39 Llamada validación email

```

/**
 * Comprueba que la dirección email introducida casa con el patrón correcto determinada por las expresiones regulares provistas por Java
 * @param email Dirección de correo a validar
 * @return Validez del correo
 */
private boolean validarEmail(String email) {
    Pattern pattern = Patterns.EMAIL_ADDRESS;
    return pattern.matcher(email).matches();
}

```

Ilustración 40 Validar email

```

public static final Pattern EMAIL_ADDRESS
    = Pattern.compile(
        "[a-zA-Z0-9\\+\\.\\_\\%\\-\\!]{1,256}" +
        "\\@" +
        "[a-zA-Z0-9][a-zA-Z0-9\\-]{0,64}" +
        "(" +
        "\\." +
        "[a-zA-Z0-9][a-zA-Z0-9\\-]{0,25}" +
        ")+"
    );

```

Ilustración 41 Expresión regular email

Una vez pasadas las validaciones se procede a realizar la llamada al método provisto por Firebase para registrar un usuario con email y contraseña. Para ello se recupera en la variable *mAuth* la sesión recientemente creada y a través de su método *createUserWithEmailAndPassword* se procede al registro. Este método es asíncrono por lo que se crea un escuchador que reaccione a la finalización del evento de registro (*addOnCompleteListener*).

```
mAuth.createUserWithEmailAndPassword(usuario_text,contrasena_text)
    .addOnCompleteListener(activity, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if(task.isSuccessful()){
                mAuth.getCurrentUser().sendEmailVerification()
                .addOnCompleteListener(new OnCompleteListener<Void>() {
                    ..
                })
            }
        }
    })
```

Ilustración 42 Lanzamiento evento registro

En el supuesto de resultar exitosa la tarea, se extrae el texto previo al signo de arroba del correo y se almacena como nombre por defecto del usuario, además se recoge una imagen por defecto para establecerla de perfil. Todo ello se introduce en un objeto de petición de cambio de datos representado por la clase *UserProfileChangeRequest*. Hecho esto, se recupera el usuario con la sesión que se acaba de crear (*currentUser*) y se llama a su método para actualizar datos con el objeto de petición creado anteriormente. De nuevo, como todos los métodos ofrecidos por Firebase, se trata de una funcionalidad asíncrona por lo que se establece un escuchador.

```
public void onComplete(@NonNull Task<Void> task) {
    if(task.isSuccessful()){
        int posArroba=mAuth.getCurrentUser().getEmail().toString().indexOf("@");
        String nombreUsuario=mAuth.getCurrentUser().getEmail().toString().substring(0,posArroba);
        Log.d( tag: "EMAIL",nombreUsuario);
        UserProfileChangeRequest profileUpdates = new UserProfileChangeRequest.Builder()
            .setDisplayName(nombreUsuario)
            .setPhotoUri(Uri.parse("https://example.com/jane-q-user/profile.jpg"))
            .build();
        final FirebaseUser currentUser = mAuth.getCurrentUser();
        currentUser.reload();
        currentUser.updateProfile(profileUpdates)
            .addOnCompleteListener(new OnCompleteListener<Void>() {
```

Ilustración 43 Actualización de perfil

En caso de tener éxito la tarea, se recupera la instancia del repositorio donde se almacenarán las imágenes de perfil de los usuarios (identificadas por su correo seguido de la extensión jpg) a través de la clase *StorageReference*. Una vez recuperada se sube la foto de perfil del usuario mediante la clase *UploadTask*.

```

public void onComplete(@NonNull Task<Void> task) {
    if (task.isSuccessful()) {
        FirebaseStorage storage = FirebaseStorage.getInstance();
        StorageReference storageRef = storage.getReference();
        StorageReference userImageRef = storageRef.child("images/" + currentUser.getEmail().toString() + ".jpg");
        Uri file = Uri.parse("android.resource://com.miguelcabezas.tfm.saltour/" + R.drawable.profile_icon);
        UploadTask uploadTask = userImageRef.putFile(file);
    }
}

```

Ilustración 44 Almacenamiento de foto

Para finalizar, se recupera la instancia de base de datos no relacional donde se almacena la información del usuario y los retos. Una vez recuperada se conforma el documento que relacionará al usuario con los retos que posee y el tiempo invertido. Los campos de este documento serán el email del usuario, el tiempo invertido en los retos y un mapa que relacione el nombre del reto con el tiempo que lleva invertido en él. Para marcar un reto como finalizado se le concatenará una C en el final valor del tiempo invertido.

Conformado el documento, se almacenará haciendo uso del método `db.collection("users").document(currentUser.getEmail()).set(newUser)` en el que en la colección de documentos denominada *users*, se recupera un documento con nombre el email del usuario (cada usuario tendrá su documento dentro de la colección) y en él se establecerán los valores comentados anteriormente mediante un mapa de claves identificativas de lo almacenado y valores aquellos previamente explicados representados en la variable *newUser*.

```

FirebaseFirestore db = FirebaseFirestore.getInstance();
Map<String, Object> newUser = new HashMap<>();
newUser.put("email", currentUser.getEmail());
newUser.put("challengesCompleted", 0);
newUser.put("totalTime", 0);
newUser.put("challengesAndTime", new HashMap<String, String>());
db.collection("users").document(currentUser.getEmail()).set(newUser);

```

Ilustración 45 Almacenamiento en Firebase

Creado el usuario, Firebase se encarga automáticamente de enviar el correo de verificación sin el cual el usuario no podrá iniciar sesión. Esto es así debido a que se ha marcado la opción de verificación de usuarios.

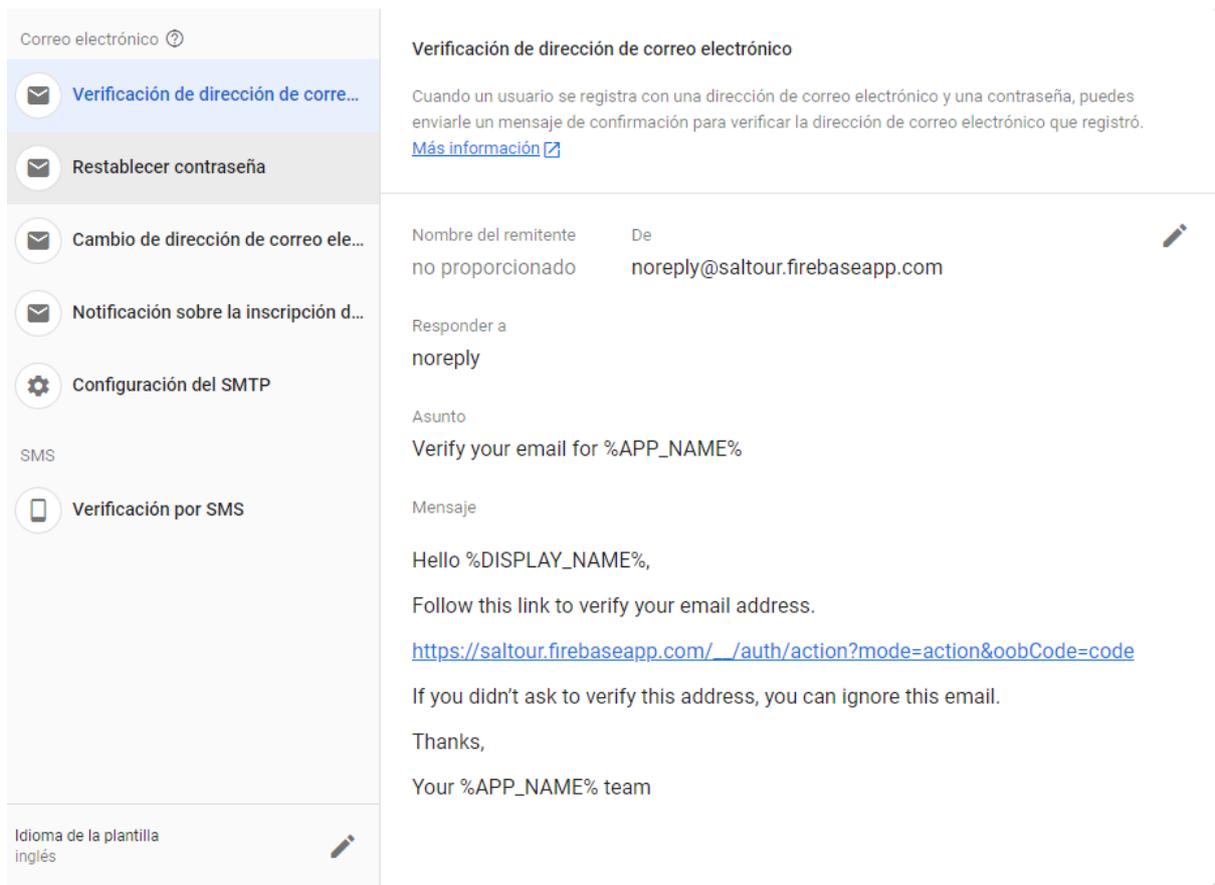


Ilustración 46 Verificación de usuarios

En caso de error en cualquiera de las tareas asíncronas explicadas anteriormente o existencia previa del usuario, se notificará al usuario manteniéndole en la pantalla de registro.

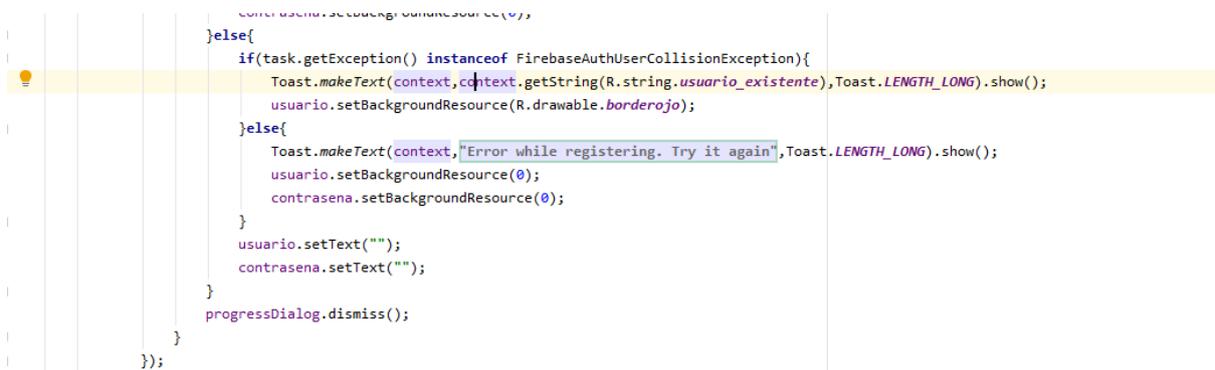


Ilustración 47 Flujo alternativo

Registro con Google

El flujo se desencadena cuando el usuario decide iniciar sesión mediante el botón de inicio de sesión con cuenta de Google que puede verse en el manual del usuario.

En primer lugar, valida que el token asociado a la cuenta de Google seleccionada sea correcto a través del proveedor de credenciales propio de Google y su método *getCredential*

```
private void firebaseAuthWithGoogle(String idToken, final FirebaseAuth mAuth, final Context context, final Activity activity){
    AuthCredential credential = GoogleAuthProvider.getCredential(idToken, accessToken: null);
    mAuth.signInWithCredential(credential)
```

Ilustración 48 Validación token Google

En caso de que sea correcto se comprueba si para dicho email existe un documento en la colección de usuarios y retos. Esto se hace así ya que el usuario puede bien estar registrándose o bien accediendo no por primera vez a través de este método de autenticación.

```
if(task.isSuccessful()){
    final FirebaseUser currentUser = mAuth.getCurrentUser();
    currentUser.reload();
    FirebaseFirestore db = FirebaseFirestore.getInstance();
    final CollectionReference usersRef = db.collection( collectionPath: "users");
    usersRef.whereEqualTo( field: "email", currentUser.getEmail())
        .get()
```

Ilustración 49 Comprobación existencia usuario

En caso de que el documento no exista, se creará de igual forma que se hacía en el registro convencional con correo y contraseña. De lo contrario, se comprobará si su mapa de retos iniciados o completados contiene elementos. De no ser así simplemente se le crea como se explicó en el otro modo de registro.

```
public void onComplete(@NonNull Task<QuerySnapshot> task) {
    if (task.isSuccessful()) {
        Log.e( tag: "TAG", msg: "EXISTE EN BBDD");
        ArrayList<String> items = new ArrayList();
        Map<String, String> challengesAnTime = new HashMap();
        Set<String>set = new HashSet();
        for (QueryDocumentSnapshot document : task.getResult()) {
            Log.e( tag: "Map",document.getData().get("challengesAndTime").toString());
            ObjectMapper oMapper = new ObjectMapper();
            challengesAnTime = oMapper.convertValue(document.getData().get("challengesAndTime"), Map.class);
            for (Map.Entry<String, String> entry : challengesAnTime.entrySet()) {
                Log.e(entry.getKey(), String.valueOf(entry.getValue()));
                set.add(entry.getKey()+"#"+String.valueOf(entry.getValue()));
            }
        }
    }

    if(set.isEmpty()){
        int posArroba=mAuth.getCurrentUser().getEmail().toString().indexOf("@");
        String nombreUsuario=mAuth.getCurrentUser().getEmail().toString().substring(0,posArroba);
        Log.d( tag: "EMAIL",nombreUsuario);
        UserProfileChangeRequest profileUpdates = new UserProfileChangeRequest.Builder()
            .setDisplayName(nombreUsuario)
            .setPhotoUri(Uri.parse("https://example.com/jane-q-user/profile.jpg"))
            .build();
        final FirebaseUser currentUser = mAuth.getCurrentUser();
        currentUser.reload();
        currentUser.updateProfile(profileUpdates)
```

Ilustración 50 Usuario existente sin retos

En cualquiera de los casos, para posteriormente acortar el tiempo de respuesta de la aplicación, se almacena en las preferencias de usuario (en cada inicio de sesión tanto por Google como por el método de usuario y contraseña) mapa de retos que tiene activos, así como el total de

jugadores en el juego a modo de caché. Así, cuando se deban recuperar, por ejemplo, en las estadísticas, no se consume tiempo ni recursos en acceder a remoto continuamente, sino que se recupera de una forma más rápida del propio dispositivo.

```

SharedPreferences myPrefs = context.getSharedPreferences( name: "ChallengesCompleted#" + currentUser.getEmail(), mode: 0);
SharedPreferences.Editor editor = myPrefs.edit();
editor.putStringSet( key: "ChallengesCompleted#" + currentUser.getEmail(), set);
editor.apply();
editor.commit();

usersPref.get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
    @Override
    public void onComplete(@NonNull Task<QuerySnapshot> task) {
        if(task.isSuccessful()){
            SharedPreferences myPrefs = context.getSharedPreferences( name: "ActiveUsers", mode: 0);
            SharedPreferences.Editor editor = myPrefs.edit();
            editor.putLong("ActiveUsers", task.getResult().size());
            editor.apply();
            editor.commit();
        }
    }
}

```

Ilustración 51 Caché en Shared Preferences

Parada de tiempos y actualización de datos

Este flujo tiene dos posibles precedentes, el primero de ellos que el usuario, intencionadamente, antes de terminar el reto desee abandonarlo parando el tiempo. Para ello, pulsa el botón asociado de la pantalla de jugar que se explica en el manual de usuario en próximos apartados. Esto hace que en primer lugar se compruebe si el servicio de conteo de tiempo representado por la clase CountTimeService se encuentra activo

```

btnParar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Log.d( tag: "b_parar", msg: "Parar el tiempo del reto");
        if(isAServiceRunning(CountTimeService.class)){

```

Ilustración 52 Parada intencionada de tiempo

Para ello se mira los servicios que tiene arrancados el sistema del dispositivo. En caso de que el nombre de alguno coincida con el del servicio de conteo, entonces significa que este se encuentra activo.

```

private boolean isAServiceRunning(Class<?> serviceClass) {
    ActivityManager manager = (ActivityManager) getActivity().getSystemService(Context.ACTIVITY_SERVICE);
    for (ActivityManager.RunningServiceInfo service : manager.getRunningServices(Integer.MAX_VALUE)) {
        if (serviceClass.getName().equals(service.service.getClassName()) {
            return true;
        }
    }
    return false;
}

```

Ilustración 53 Comprobación servicio conteo activo

En caso afirmativo se procede a pararlo mediante el método nativo `stopService`. Posteriormente, y siempre en caso que no se trate del usuario invitado, se lanza un hilo que espera un tiempo prudencial de 3 segundos a que el servicio pare para seguidamente actualizar los datos. Para el caso de parada intencionada únicamente se actualizan en local ya que al no completar el reto no afecta al sistema de puntuaciones. En el caso de parada por compleción del reto, se actualizan a mayores los datos en la colección remota de usuarios y retos como se explica en el siguiente punto.

```

if(isAServiceRunning(CountTimeService.class)){
    ActiveChallengeSingleton activeChallengeSingleton = ActiveChallengeSingleton.getInstance();
    getActivity().stopService(new Intent(getContext(), CountTimeService.class));
    v.setVisibility(View.INVISIBLE);
    v.setEnabled(false);
    FirebaseAuth auth = FirebaseAuth.getInstance();
    if(!(auth.getCurrentUser().getEmail().equalsIgnoreCase( anotherString: "Invitado@testsaltour.com"))){
        getActivity().runOnUiThread(esperarYActualizarPreferences( milisegundos: 3000,auth.getCurrentUser().getEmail(),activeChallengeSingleton.getName()));
    }
}

```

Ilustración 54 Actualización de datos

Actualización de datos en remoto

El hilo comentado anteriormente llama al método `esperarYActualizar`. Este simplemente espera durante 3 segundos que pare el servicio y posteriormente actualiza los datos haciendo llamada al método `updateUserData`

```

/**
 * Genera una espera útil para dar tiempo a parar tiempos por el servicio de conteo. Pasado ese tiempo llama al controlador para realizar los guardados en base de datos
 * @param milisegundos milisegundos de espera
 * @param email email del usuario afectado
 * @param challengeName nombre del reto a actualizar
 */
public Runnable esperarYActualizar(final int milisegundos, final String email, final String challengeName) {
    return () -> {
        Handler handler = new Handler();
        handler.postDelayed(() -> {
            Log.e( tag: "ACCIONES", msg: "TRAS 3000ms");
            updateUserData(email,challengeName);
        }, milisegundos);
    };
}

```

Ilustración 55 Hilo de actualización

El método de actualización simplemente llama al controlador asociado para actualizar los datos siempre y cuando se trate de un usuario no invitado, de lo contrario únicamente lanza el paquete contenedor de la realidad aumentada sin registrar tiempos (ya que el usuario invitado no computa en el sistema de clasificación).

```

private void updateUserData(final String emailUser, final String challengeName){

    if(!(emailUser.equalsIgnoreCase( anotherString: "invitado@testsaltour.com"))){
        ProfileSettingsController profileSettingsController = new ProfileSettingsController();
        profileSettingsController.updateUserChallengesAndProcessAR(emailUser,challengeName,getContext(),getActivity());
    }
}

```

Ilustración 56 Actualización de datos usuario registrado

```

}else{
    Log.e( tag: "#####", msg: "TO AR invitado");
    Intent LaunchIntent = getContext().getPackageManager().getLaunchIntentForPackage( packageName: "com.DefaultCompany.CharacterTalking");
    startActivityForResult(LaunchIntent, requestCode: 2);
}

```

Ilustración 57 Lanzamiento de realidad aumentada

El controlador simplemente, haciendo uso del DAO, recupera la información de la colección asociada al usuario que ha completado el reto.

```

public void updateUserChallengesAndProcessAR(final String emailUser, final String challengeName, final Context context, final Activity activity){
    final DaoFirebaseImpl daoFirebase = new DaoFirebaseImpl();
    FirebaseFirestore db = daoFirebase.getDatabaseInstance();
    CollectionReference dbUsers = daoFirebase.getCollectionReference(db, path: "users");

    final DocumentReference userSelected = daoFirebase.getDocumentReference( path: "users", emailUser, db);

```

Ilustración 58 Recuperación colección usuario

Una vez recuperada la colección, selecciona dentro de ella el mapa contenedor del tiempo invertido en cada reto y concatena al recientemente completado una “C” al final del tiempo invertido para así denotar que está completado para futuros procesamientos en el sistema de puntuaciones.

```

public void onComplete(@NonNull Task<DocumentSnapshot> task) {
    DocumentSnapshot document = task.getResult();
    if(document.exists()){
        Map<String, String> challengesAndTime = (Map<String, String>) document.get("challengesAndTime");
        if(challengesAndTime != null && !challengesAndTime.isEmpty() && challengesAndTime.get(challengeName) != null){
            String tiempo = challengesAndTime.get(challengeName);
            challengesAndTime.put(challengeName, tiempo+"C");
        }
        Set<String> set = new HashSet<>();
        for (Map.Entry<String, String> entry : challengesAndTime.entrySet()) {
            Log.e(entry.getKey(), String.valueOf(entry.getValue()));
            set.add(entry.getKey()+"#"+String.valueOf(entry.getValue()));
        }
    }
}

```

Ilustración 59 Actualización de tiempos

Una vez actualizado el mapa se lanza una petición de actualización de la colección de tal forma que se le inserta el nuevo mapa de retos-tiempo, se suma uno al número de retos completados y se actualiza el campo de la relación entre retos completados y tiempo invertido en la que se basa el sistema de puntuaciones (relación entre el tiempo total invertido tanto en retos completados como en retos en curso dividido entre el número total de retos completados, a menor relación, mejor puesto en la clasificación).

```

daoFirebase.updateDocumentField(userSelected, field: "challengesCompleted_totalTime", value: (((Long)document.get("totalTime")))/(((Long)document.get("challengesCompleted")+1));
daoFirebase.updateDocumentField(userSelected, field: "challengesCompleted", value: ((Long)document.get("challengesCompleted")+1));
daoFirebase.updateDocumentField(userSelected, field: "challengesAndTime", challengesAndTime);

```

Ilustración 60 Almacenamiento en colección

Además, a fin de una mayor velocidad en la respuesta, se almacena el nuevo mapa de retos y tiempo en las preferencias de la aplicación para el usuario activo a modo de caché.

```

DaoSharedPreferencesImpl daoSharedPreferences = new DaoSharedPreferencesImpl();
SharedPreferences myPrefs = daoSharedPreferences.getChallengesCompletedSharedPreferences( key: "ChallengesCompleted#" + emailUser, context);
daoSharedPreferences.updateChallengesCompleted(myPrefs, key: "ChallengesCompleted#" + emailUser, set);

```

Ilustración 61 Almacenamiento en caché

Finalmente se llama al paquete para lanzar la realidad aumentada de igual modo que se hacía para el usuario invitado.

```

Intent launchIntent = context.getPackageManager().getLaunchIntentForPackage( packageName: "com.DefaultCompany.CharacterTalking");
launchIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
activity.startActivityForResult(launchIntent, requestCode: 2);

```

Ilustración 62 Llamada a realidad aumentada

Interacción con realidad aumentada

Tras configurar todo lo necesario para poder detectar automáticamente a través de Vuforia los marcadores que hacen aparecer la realidad aumentada (se recomienda ver el anexo V para un mayor detalle), llega el momento de poder interactuar con esta a través de la detección de eventos de pulsación sobre la pantalla. Para este proyecto se ha implementado que al detectar la pulsación de pantalla la animación del modelo 3D así como la explicación tanto por medio auditivo como visual se detenga. En caso de detectar de nuevo una pulsación, este conjunto de animación y audio se reanuda desde el punto donde previamente se detuvo.

Para este caso de estudio tanto la animación como el modelo 3D se han extraído de la página Mixamo (Adobe Systems Incorporated, 2022).

Una vez descargada se incorpora al proyecto mediante *drag and drop*, es decir, arrastrando y soltando en el directorio de carpetas del proyecto. Una vez situado en la escena, se le añade y ajusta, mediante la técnica de ensayo y error, un *Box Collider* con el objetivo de poder interactuar si se pulsa sobre él. Este componente actúa a modo de estructura exterior que detecta pulsaciones, colisiones con otros objetos, etc.

Una vez incorporado el elemento, se añade un *script* al modelo para que, al pulsar sobre el personaje, tanto la animación como el sonido se detengan. En este código, recogido a continuación, mediante el método *OnMouseDown*, se detecta la posición sobre el componente *Box Collider* y, en caso de estar la animación y el sonido activos, se paran. De lo contrario, se reanudan. A mayores, para indicar de una forma visual las indicaciones de la realidad aumentada, se ha incluido un vídeo textual que se pausa y reanuda también con la detección de pulsaciones sobre el modelo 3D de la realidad aumentada.

Todo esto se puede ver en el método *OnMouseDown*, este debido al motor de animación proporcionado por Unity es capaz de detectar la pulsación sobre el modelo 3D a través de la estructura que se le asigna en forma de *Box Collider*. Una vez detectado se hace una llamada

al método `paraSi()`. En él, mediante un marcador se comprueba si la animación se encuentra activa (valor verdadero) o detenida (valor falso). En caso de estar activa se cambia en primer lugar el marcador de valor para posteriormente recuperar el componente de animación del modelo 3D (Animator) y marcarlo como desactivado (logrando con ello pausar la animación). Por último, se recupera el componente de vídeo de la escena (videoPlayer), que ha sido pasado como parámetro al código, y se pausa mediante su método nativo `Pause()`.

Por el contrario, si la animación se encuentra detenida, se cambia el marcador de valor, de nuevo se recupera el componente de animación del modelo 3D activándolo en este caso para por último obtener el componente de vídeo de la escena y reanudarlo mediante su método nativo `Play()`.

Cabe destacar que el componente de vídeo de la escena se trata de un objeto a modo de plano en el cual se reproduce un vídeo en el que aparece a través de texto aquello que se reproduce simultáneamente en audio (ver el siguiente apartado de manual de usuario). Este vídeo se ha hecho a través de una página que lee un texto y va resaltando progresivamente las palabras sobre las que pasa.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.Video;
5
6  public class StopAnimation : MonoBehaviour
7  {
8      public GameObject objectStop;
9
10
11     public GameObject quad;
12
13     VideoPlayer videoPlayer;
14
15     bool animacionActiva = false;
16
17     public void paraSi(){
18         if(animacionActiva == true){
19             animacionActiva = false;
20             objectStop.GetComponent<Animator>().enabled = false;
21             videoPlayer.Pause();
22             Debug.Log("[StopAnimation-paraSi()]-Animated character stopped successfully");
23         }
24         else{
25             animacionActiva = true;
26             objectStop.GetComponent<Animator>().enabled = true;
27             videoPlayer.Play();
28             Debug.Log("[StopAnimation-paraSi()]-Animated character (re)started successfully");
29         }
30     }
31     // Start is called before the first frame update
32     void Start()
33     {
34         Debug.Log("[StopAnimation-Start()]-Animated motion started successfully");
35         videoPlayer = quad.GetComponent<VideoPlayer>();
36         objectStop.GetComponent<Animator>().enabled = false;
37     }
38
39     // Update is called once per frame
40     //void Update()
41     /*{
42         Debug.Log("update");
43     }*/
44
45     public void OnMouseDown()
46     {
47         Debug.Log("[StopAnimation-onMouseDown()]-Interaction with AR animated character detected");
48         paraSi();
49     }
50

```

Ilustración 63 Interacción con la realidad aumentada

Manual del usuario

Instalación

En primer lugar, previo a su uso el usuario deberá instalarse la aplicación bien a través de la importación de apk como se indica en el anexo VII ya que la aplicación no se encuentra todavía disponible en la tienda de aplicaciones. Para poder hacerlo su dispositivo deberá disponer de una serie de requisitos y recursos mínimos detallados a continuación:

- ◇ Requisitos hardware mínimos
 - 3G o WiFi: ya que tanto la recuperación de información global como la actualización de la propia y localización de retos requiere de acceso a servidores online. Además, se requerirá para otras funcionalidades como el envío de correos y una mejor localización del propio dispositivo en el mapa.
 - Geolocalizador en el dispositivo: se requerirá activar la ubicación del dispositivo en caso de querer mostrar la propia localización del usuario respecto al reto iniciado.
 - 20MB, al menos, de memoria disponible para almacenar la aplicación y los datos fundamentales de esta.
- ◇ Requisitos software mínimos
 - Android versión 5.0, como mínimo, para un correcto funcionamiento de todas las funcionalidades.

How to de la aplicación

Los flujos de funcionamiento, también recogidos en el anexo VII como manual de usuario, abarcan desde el registro del usuario hasta la compleción de retos pasando por la modificación del perfil, visualización de clasificaciones, envío de correos, etc.

Registro

En esta pantalla el usuario podrá introducir un email y contraseña para crearse una cuenta. Se validará mostrando un mensaje en caso de existir ya el usuario o no cumplir con las normas de contraseña (al menos ocho caracteres). En caso de registro correcto el sistema mandará un correo de verificación al usuario para que confirme su cuenta. Una vez verificada podrá entrar a ella.



SalTour

Email

Contraseña

REGISTRARSE

Ilustración 64 Registro



SalTour

Email

Contraseña

REGISTRARSE

Usuario ya existente

Ilustración 65 Registro inválido

Si un usuario registrado intenta iniciar sesión si haber confirmado su correo, se le indica

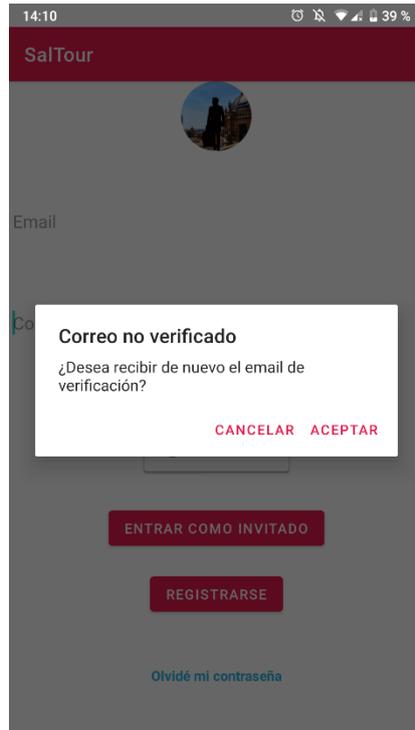


Ilustración 66 Correo no verificado tras registro

Deberá confirmar su dirección de correo a través el enlace enviado al correo asociado a su nueva cuenta. Una vez hecho podrá acceder a su cuenta.

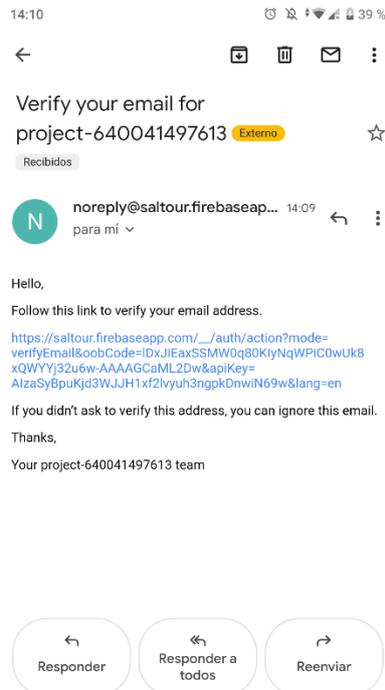


Ilustración 67 Correo de verificación

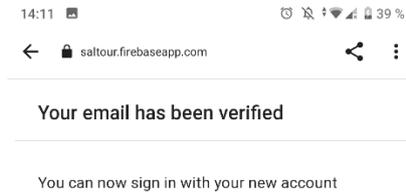


Ilustración 68 Correo verificado

Login

Al lanzar la aplicación aparecerá la siguiente pantalla en la que el usuario podrá entrar a la aplicación bien como invitado (opción en la que no podrá visualizar ni personalizar su perfil ni ver estadísticas), con email y contraseña o con su cuenta de Google (en esta última opción el registro se realizará de forma simultánea al inicio de sesión en caso de no disponer de cuenta ya creada).

Por otra parte, podrá registrarse (siendo llevado a otra pantalla) u obtener un correo de recuperación de contraseña en caso de haber sido olvidada.



SalTour

Email

Contraseña

ENTRAR

Iniciar sesión

ENTRAR COMO INVITADO

REGISTRARSE

[Olvidé mi contraseña](#)

Ilustración 69 Login



SalTour

Email

Contraseña

ENTRAR

Iniciar sesión

ENTRAR COMO INVITADO

REGISTRARSE

[Olvidé mi contraseña](#)

Usuario y/o contraseña incorrecto

Ilustración 70 Login incorrecto

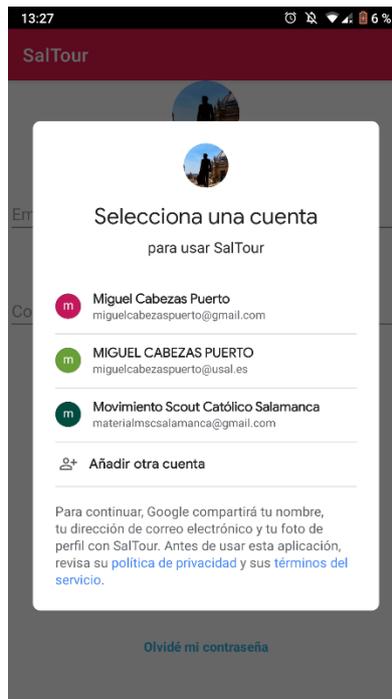


Ilustración 71 Login con Google

Recuperación de contraseña

En esta pantalla el usuario podrá introducir un email para el envío de los pasos para recuperar su contraseña. Una vez introducido le llegarán a dicho email.



Ilustración 72 Olvido contraseña

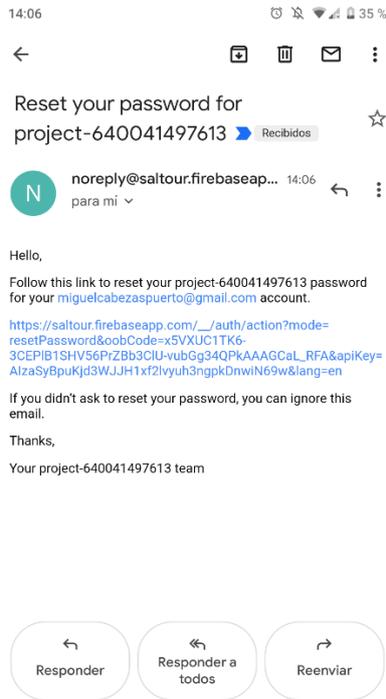


Ilustración 73 Correo de reseteo de contraseña

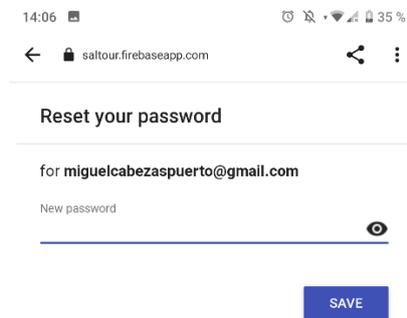


Ilustración 74 Contraseña reseteada

Inicio

En esta pantalla el usuario será recibido y podrá ver por una parte una serie de imágenes relacionadas con la ciudad mientras que por otra visualizará en un gráfico circular su progreso en la aplicación en cuanto al número de retos completados.



Ilustración 75 Inicio

Jugar

En esta pantalla el usuario puede ver el listado de todos los retos. Estos estarán marcados con diferentes colores en función de si están completados por el usuario (verde), en progreso (amarillo) o sin empezar (blanco). De igual modo, aquellos retos no completados, dispondrán de un botón de *play* sobre el cual el usuario podrá pulsar. Una vez pulsado se mostrará un mapa con la localización del reto en el que, siguiendo el diseño de Google Maps, el usuario podrá ser guiado hasta él desde su posición.

A mayores en esta pantalla aparecen dos botones. Uno de ellos “Escanear”, estará activo siempre, pulsándolo se abrirá la cámara con la que el usuario podrá enfocar a códigos QR para tratar de finalizar el reto. Por otro lado, el botón “Detener”, visible solo cuando el usuario tenga algún reto en curso, permitirá al usuario detener el conteo de tiempo del reto en curso en caso

que así lo desee por no encontrarlo o querer interrumpirlo sin haberlo encontrado (para, por ejemplo, comenzar otro)



Ilustración 76 Pantalla jugar con reto activo

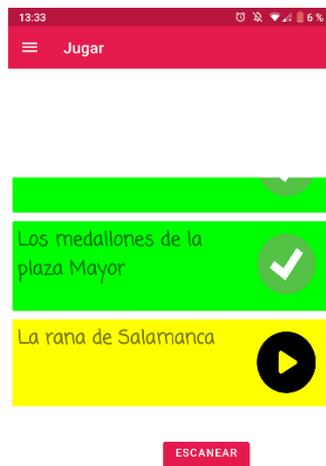


Ilustración 77 Pantalla jugar reto no completado

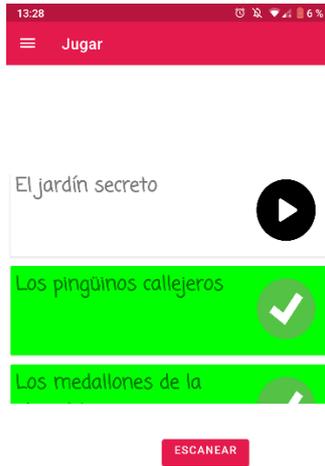


Ilustración 78 Pantalla jugar retos completados y no iniciado



Ilustración 79 Pantalla escaneo QR

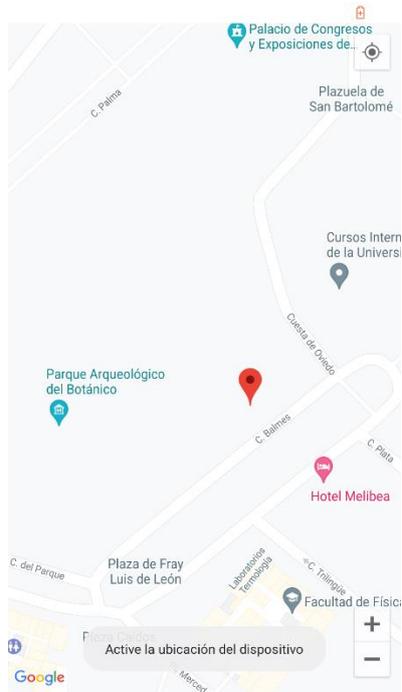


Ilustración 80 Mapa sin ubicación activa



Ilustración 81 Mapa con ubicación activa

Mi perfil

En esta pantalla el usuario puede tanto ver como modificar su nombre, contraseña y foto de perfil. Para modificar su foto de perfil deberá pulsar sobre la actual abriéndose el selector de

imágenes de su dispositivo. Para poder modificar nombre y/o contraseña deberá pulsar sobre el lápiz. Una vez hecho pedirá estos datos. Para guardar los cambios bastará con pulsar sobre el botón guardar.



Ilustración 82 Mi perfil



Ilustración 83 Edición de perfil

Estadísticas

Por un lado, en esta pantalla el usuario podrá ver sus estadísticas individuales en la pestaña “Individual”. En ella, ampliando los desplegables podrá ver tanto el tiempo invertido en los retos completados como el invertido en sus retos en curso.

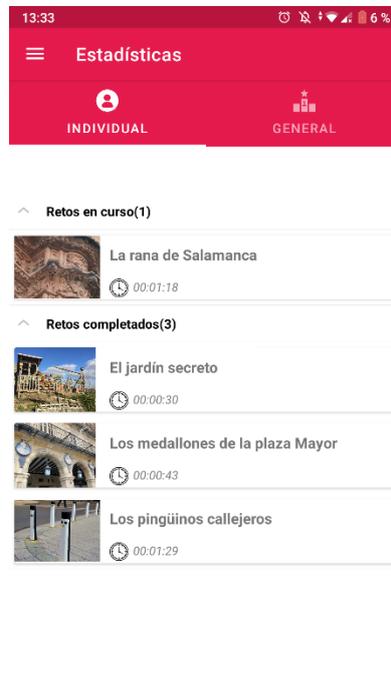


Ilustración 84 Estadísticas individuales

Por otro lado, mediante la pestaña “General” podrá ver una clasificación global de los jugadores activos en función de la relación tiempo invertido-retos completados. En esta podrá también ver el número de jugadores activos en el juego (al menos un reto iniciado) y la propia posición dentro del *ránking*.

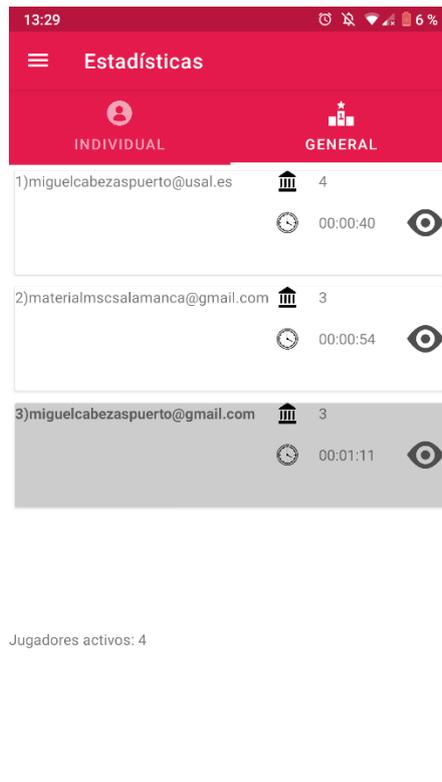


Ilustración 85 Estadísticas globales

Además, pulsando en el símbolo del ojo de jugadores ajenos, podrá ver información de su perfil en detalle como la foto, el número de retos superados, la posición, así como el tiempo desglosado de cada reto superado por éste.



Ilustración 86 Perfil otro jugador

Compartir

Mediante esta pantalla el usuario, pulsando en cada uno de los logos, podrá compartir el enlace de la aplicación a la Play Store en diferentes medios. Con esto se pretende dar visibilidad a la aplicación.

Al pulsar cada logo se enlazar  con su aplicaci n correspondiente (en caso de disponer de ella) y se escribir  una publicaci n/mensaje en esta ya predeterminado que el usuario podr  personalizar.



Ilustraci n 87 Compartir app

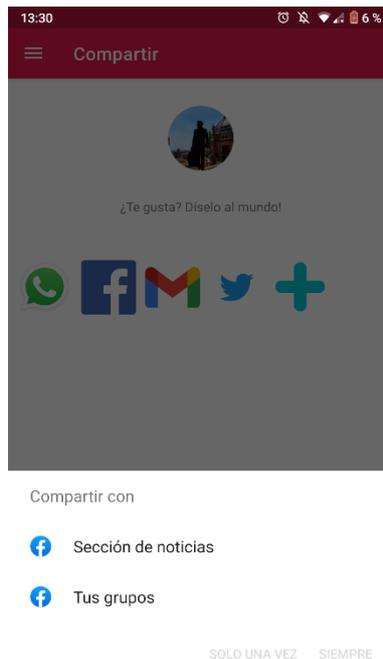


Ilustración 88 Compartir app Facebook

Ayuda

En ella aparecen una serie de preguntas frecuentes con sus respuestas (basta pulsar en el desplegable de la pregunta para ver la respuesta). La ayuda se realiza así para dotar al usuario de una guía sencilla, directa y práctica sin necesidad de seguir un guion.

A mayores, pulsando sobre el *link* de la parte inferior de la pantalla, se abrirá el gestor de correo del usuario con un mensaje con asunto y destino predeterminados (el equipo de desarrollo) para el caso en que el usuario quiera plantear una duda fuera del alcance de las preguntas frecuentes. La intención es que una vez recopiladas una serie de dudas estas se formulen y pasen a formar parte del repositorio de preguntas frecuentes.

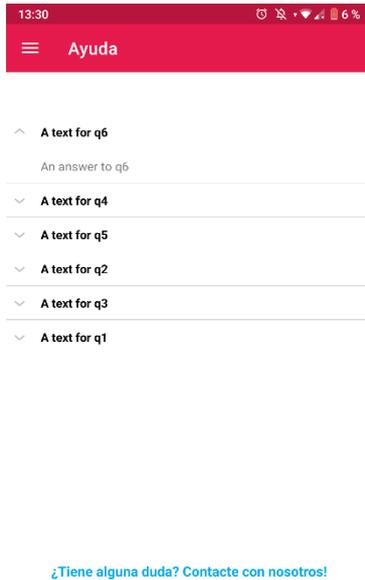


Ilustración 89 Preguntas frecuentes

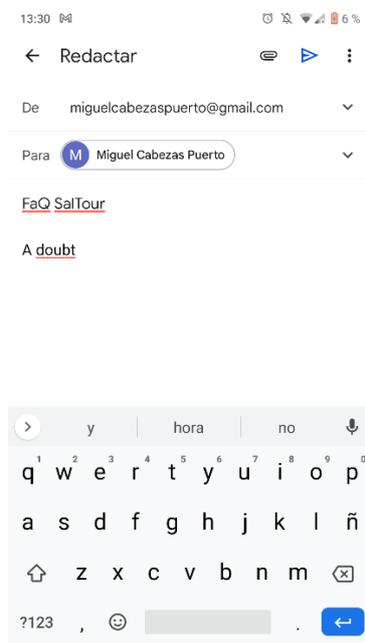


Ilustración 90 Ayuda personalizada

Manú lateral

En cada una de las pantallas aparece un menú desplegable en la parte superior izquierda. Este es un menú hamburguesa, se ha decidido elegir este ya que su uso es común en la mayoría de aplicaciones Android por lo que el usuario estará familiarizado con su uso. Además, su distribución hace que no entorpezca la visualización de las pantallas.

Para abrirlo bastará con deslizar a la derecha o pulsar sobre sus líneas mientras que para cerrarlo será suficiente con deslizar a la izquierda o pulsar fuera de él.

Una vez abierto contendrá las diferentes funcionalidades de la aplicación. Para ir a cada una basta con pulsar encima de ella.

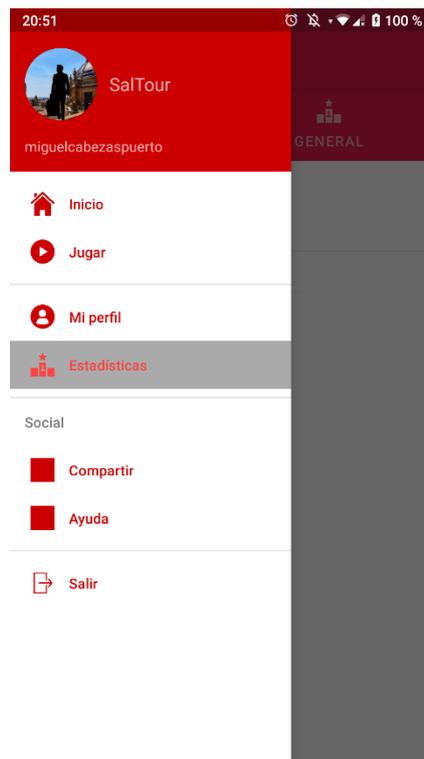


Ilustración 91 Menú lateral

Cerrar sesión

Mediante el botón salir del menú lateral se permitirá al usuario cerrar sesión de su cuenta. Una vez pulsado se lanzará un mensaje de confirmación. En caso de cancelarlo se mantendrá la sesión activa, de lo contrario se cerrará la sesión del usuario rediriéndolo a la pantalla de login.

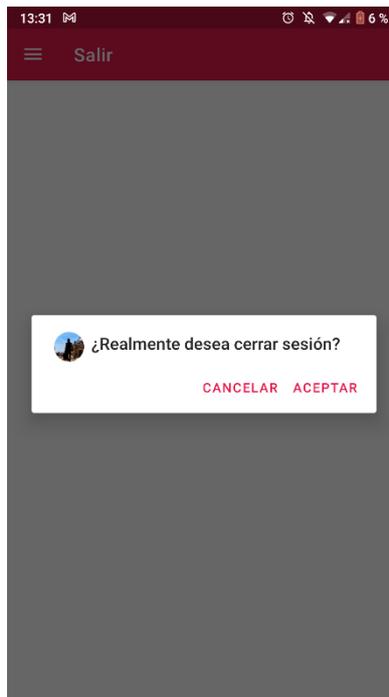


Ilustración 92 Cerrar sesión

Realidad aumentada

En esta pantalla el usuario deberá detectar el token que hace saltar la realidad aumentada (generalmente situado junto al código QR). Una vez detectado aparecerá una guía turística que mediante texto y voz explicará una breve historia del punto de interés que se acaba de encontrar asociado al reto. Esta guía podrá ser parada y reanudada cuantas veces se desee mediante la pulsación sobre ella (un toque para voz y texto, otro toque lo reanuda y así sucesivamente).

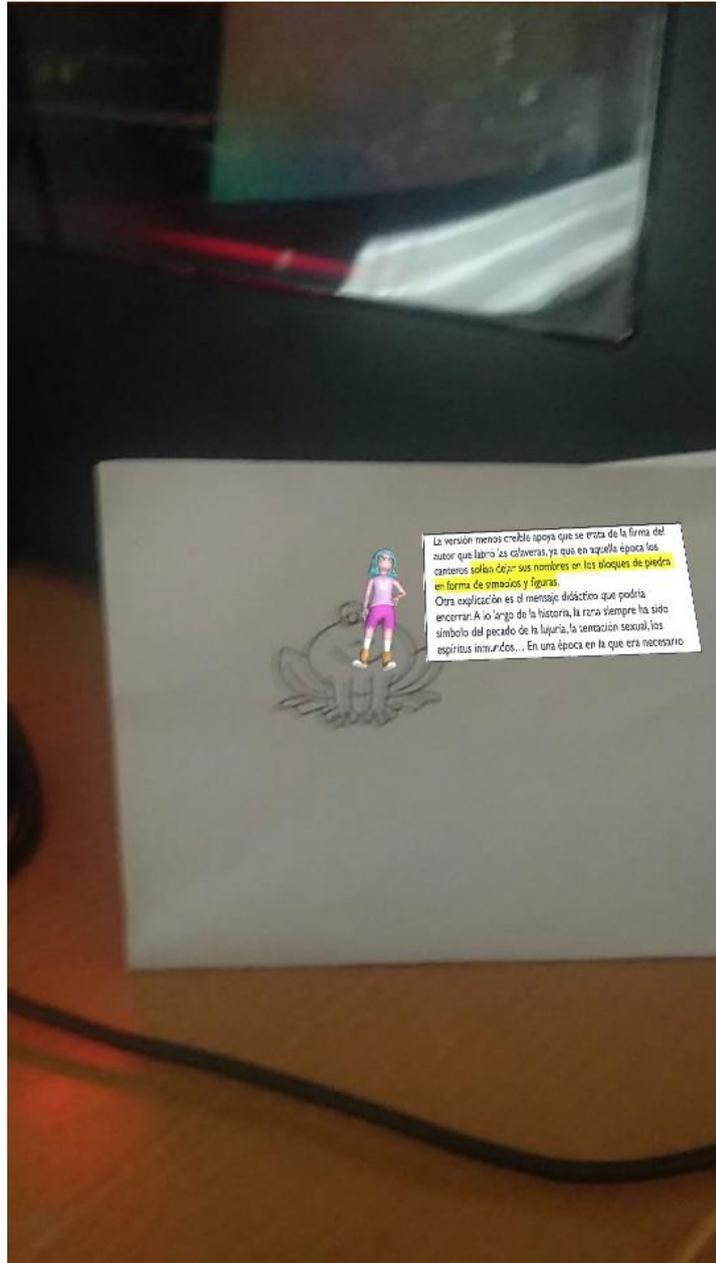


Ilustración 93 Realidad aumentada

Conclusiones y ampliaciones

En este apartado se explican las conclusiones obtenidas tras el desarrollo completo del proyecto. Además, se presentan sugerencias y vías de mejora para posibles versiones futuras de la aplicación.

Conclusiones

Compleción de objetivos

Tras la terminación del proyecto, se puede afirmar que se han cubierto todos los objetivos y requisitos plasmados en su inicio.

- ◇ Facilitar al usuario el descubrimiento de puntos de interés de la ciudad: aunque se ha centrado el caso de estudio en una única ciudad y una serie de puntos de demostración, su ampliación resultaría trivial al ser en verdad una especie de listado.
- ◇ Proporcionar al usuario un sistema de clasificación y puntuaje: a través del uso de una plataforma en remoto se ha conseguido dotar a la aplicación de un carácter cercano al tiempo real en la variación de puntuaciones y clasificación tanto propia como ajena
- ◇ Localizar la información sobre los puntos de interés: de forma familiar al usuario medio al realizarse a través de otra aplicación comúnmente usada como es Google Maps
- ◇ Guardar la información en remoto
- ◇ Desarrollar una realidad aumentada interactiva: aunque representa un punto claro de mejora tanto en integración (que sea nativo a la aplicación y no un paquete externo) como en interacción, se ha logrado representar de forma visual una guía turística con voz y texto pausables mediante la interacción mediante pulsaciones a la misma.
- ◇ Especificar configuraciones de usuario: aunque resulta ampliamente ampliable, se ha conseguido al menos poder cambiar el *nickname*, contraseña, así como imagen de perfil visible al resto de usuarios
- ◇ Soporte multilingüe: se ha logrado adaptar el contenido estático de la aplicación a tres lenguas fácilmente ampliables como se indica en próximos apartados. Resulta ítem de mejora ampliar a otros idiomas a mayores del castellano la guía por voz de la realidad aumentada.
- ◇ Ayuda al usuario: mediante el contacto vía correo electrónico con el equipo de desarrollo, automatizando la información relativa a la versión de la aplicación y de Android, el usuario puede plantear sus problemas. Además de la

disposición de un apartado de ayuda con preguntas frecuentes y un manual de usuario con los pasos a seguir para disfrutar de las diferentes funcionalidades.

Por todo ello, se ha pretendido que la aplicación resulte de utilidad a la par que intuitiva.

Entre los logros destacados se pueden mencionar:

- ◇ Disponibilidad de una base de datos remota y accesible por todos los usuarios en tiempo real para consultar las puntuaciones.
- ◇ Posibilidad de ubicar los diversos puntos de interés
- ◇ El uso de base de datos en la nube permite aligerar de carga tanto de almacenamiento como de procesado a los dispositivos móviles, así como dotar al juego de modalidad multijugador-competitiva y carácter cercano al tiempo real
- ◇ Desarrollo e integración de una realidad aumentada con la que poder interactuar. Resultando la aplicación más atractiva y novedosa en el sector.
- ◇ Las fases de pruebas resultaron cruciales para el desarrollo del proyecto permitiendo la mejora continua de esta. Además, las pruebas con usuarios se presentaron como significantes ya que ayudaron al acercamiento de la aplicación a sus destinatarios principales.

Transversales

En lo relativo a los dos pilares conceptuales sobre los que se apoya el proyecto, la gamificación y el aprendizaje basado en retos, se puede decir que el proyecto aborda un primer paso hacia la dinamización juvenil en materia de educación lúdica. Así, ofrece un aprendizaje acorde al estilo de vida basado en el consumismo y la recompensa por logros al que tiende cada día la sociedad.

De este modo, mediante la interacción con otros usuarios, tecnologías actuales como la móvil y la realidad aumentada, se consigue cumplir en los jóvenes aquellos objetivos teóricos que persigue el aprendizaje basado en retos como son motivar al individuo a la forma de aprender haciendo, promover la creatividad a la hora de resolver los retos (buscando caminos alternativos, conexiones entre retos para invertir un menor tiempo...) así como favorecer la toma de decisiones y responsabilidades y mejorar la resiliencia y adaptación de las personas, fundamentalmente jóvenes, a las emociones tales como frustración.

Todo ello tomando como herramienta la gamificación debido a que esta, implementada a través de dispositivos extendidos en la población, mayormente en los jóvenes, como son los dispositivos móviles, convierte al proyecto en una mezcla adaptada y cercana a la realidad actual, promoviendo con ello el turismo y una visión más cercana y novedosa de este, dejando atrás las típicas visitas guiadas protocolarias donde prima una escucha pasiva que termina desenganchando al público. Mediante Saltour, esta escucha y aprendizaje se convierte en activo al ser el usuario el centro sobre el que giran y se abren nuevos caminos, posibilidades y retos.

Líneas de trabajo futuras

La aplicación, a través de las diversas sugerencias captadas en las pruebas con usuarios, se presta a una sencilla ampliación de sus funcionalidades que permitiesen, por un parte, brindar al usuario más facilidades para disfrutar de un turismo asequible y alternativo, y por otra, aumentar el espectro de destinatarios en lo referido al idioma. Entre estas ampliaciones podemos destacar:

- ◇ **Publicador de actualizaciones:** aunque se ha implementado a través del servicio de mensajería de Cloud Message y Notification de Firebase, al ser un ítem de prueba gratuito no funciona todo lo preciso y al instante que se desea. Por ello supone un punto de mejora la creación de un servidor donde publicar novedades y que los dispositivos móviles leyera de este para recibir notificaciones emergentes.
- ◇ **Multi-idioma en contenido dinámico:** a través de configuraciones o la posibilidad de traductores instantáneos de voz como los ofrecidos por los servicios AWS de Amazon poder aumentar el espectro de idiomas en los que la realidad aumentada habla y en los que a su vez aparece el texto para personas con dificultades auditivas.
- ◇ **Base de datos relacional:** disponer de una base de datos en remoto relacional para una mejor estructuración de los datos y unas escrituras más consistentes y *thread safe*. Dejando la actual base de datos no relacional para lecturas rápidas.

Todas estas mejoras se pueden realizar sin afectar en gran medida a las ya disponibles debido al desarrollo modular seguido en la implementación de la aplicación.

Bibliografía

- Adobe Systems Incorporated. (2022). *Mixamo*. Obtenido de <https://www.mixamo.com/#/>
- Araújo, N., Barroso, B., Azevedo Gomes, R., & Cardoso, L. (2019). *La Gamificación aplicada al Sector Turístico: Análisis sistemático sobre la base de datos Scopus*. International Journal of Marketing, Communication and New Media. Recuperado el agosto de 2022, de <https://docplayer.es/161267483-La-gamificacion-aplicada-al-sector-turistico-analisis-sistemico-sobre-la-base-de-datos-scopus.html>
- Canva. (2022). *La psicología del color: el significado de los colores y cómo aplicarlos a tu marca*. Obtenido de https://www.canva.com/es_mx/aprende/psicologia-del-color/
- Consejería de Educación, Junta de Castilla y León. (s.f.). *Educa JCyL*. Obtenido de <https://www.educa.jcyl.es/crol/es/recursos-educativos/aprendizaje-basado-problemas-retos>
- DareMapp Software S.L. (2022). *DareMapp*. Obtenido de <http://daremapp.com/>
- Departamento de Informática de la universidad de Valladolid. (s.f.). Obtenido de <https://www.infor.uva.es/~mlaguna/is1/apuntes/2-requisitos.pdf>
- Díaz Luque, P., & H.W. Correa, C. (2017). *Gamificación y destinos turísticos*. Revista Internacional de Organizaciones. Recuperado el agosto de 2022, de <file:///C:/Users/Miguel/Downloads/Dialnet-GamificacionYDestinosTuristicosClasificacionDeLosJ-6050125.pdf>
- Edimburg World Heritage. (s.f.). *Edimburg World Heritage*. Obtenido de <https://ewh.org.uk/news/>
- Elsevier B.V. (s.f.). *Scopus*. Obtenido de <https://www.scopus.com/home.uri>
- Fernández, Y. (23 de agosto de 2019). *Xataka*. Obtenido de <https://www.xataka.com/basics/api-que-sirve>
- Flat 101. (2022). *Estudio sobre conversión en negocios digitales españoles 2022*. Madrid.
- Flores, J. (s.f.). *SEOptimer*. Obtenido de <https://www.seoptimizer.com/es/blog/psicologia-de-la-tipografia/>
- García, M. N. (s.f.). *Apuntes Gestión de proyectos*.
- Glosario Terminología Informática. (s.f.). *Glosario Terminología Informática*. Obtenido de <http://www.tugurium.com/gti/>
- Gómez, S. (s.f.). *SGOliver*. Obtenido de <https://www.sgoliver.net/>
- González Limón, M., Rodríguez Ramos, A., & Padilla Carmona, M. (2022). *Gamification as a methodological strategy at the University. The case of BugaMAP: students' perceptions and evaluations*. Universidad de Sevilla. Sevilla: Universidad de Sevilla. doi:10.12795/PIXELBIT.90394
- Google Developers. (s.f.). Obtenido de <https://developer.android.com/>
- Google Developers. (s.f.). *Firestore Documentation*. Obtenido de <https://firebase.google.com/docs>
- Grupo de Investigación GRIAL. (s.f.). Obtenido de <https://repositorio.grial.eu/bitstream/grial/1155/1/UML%20->
- Huiting, X. (2022). *A scoping review of gamification for mental health in children: Uncovering its key features and impact*. Singapur: Institute of Mental Health. Recuperado el agosto de 2022
- Jacobson. (2000). *El proceso unificado de desarrollo*.
- Jiménez, A. (2002). *Metodología para la elicitación de requisitos de sistemas software*. Sevilla.

Junta de Castilla y León. (s.f.). *Educa JCYL*. Recuperado el agosto de 2022, de <https://www.educa.jcyl.es/crol/es/recursos-educativos/gamificacion>

Malt. (2022). *Barómetro por profesión*. Obtenido de <https://www.malt.es/t/barometro-tarifas/tech/desarrollador-mobile/desarrollador-android>

Mapfre. (s.f.). *Fundación Mapfre*. Obtenido de <https://www.fundacionmapfre.org/educacion-divulgacion/educacion-financiera-seguros/actividades-educativas/juego-estrategia-seguros-bugamap/>

Moreno García, M., García, P., & JoséFrancisco. (s.f.). *Apuntes Ingeniería del Software II*.

Moreno García, M., Peñalvo, G., & José, F. (s.f.). *Apuntes Ingeniería del software I*.

Muradas, Y. (22 de junio de 2021). *Open Webinars*. Obtenido de <https://openwebinars.net/blog/que-es-firebase-de-google/>

Ryte Wiki. (s.f.). *Ryte Wiki*. Obtenido de <https://es.ryte.com/wiki/Mockup>

Salem, A., Noori, M., & Ozdamli, F. (2022). *Gamification Applications in E-learning: A Literature Review*. Mersin: Department of Computer Information Systems, Near East University.

Sánchez, J. J. (s.f.). *Github*. Recuperado el 16 de febrero de 2021, de <https://gist.github.com/josejuansanchez/7501a91223ef2064167a15908ad5991b>

StackOverflow. (s.f.). *StackOverflow*. Obtenido de <https://es.stackoverflow.com/questions/268777/cómo-agrego-funcionalidad-a-los-botones-de-un-dialogfragment-customizado>

Tourisfy. (s.f.). *Tourisfy!* Obtenido de <https://www.touristfy.com/index.php>

Unity Technologies. (28 de marzo de 2018). *Unity documentation*. Obtenido de <https://docs.unity3d.com/es/2018.4/Manual/vuforia-sdk-overview.html>

Valles, D. M. (1999). *Las Tecnologías de la Información y el turismo*. Madrid: Estudios Turísticos.

Wikibooks. (s.f.). *Wikibooks*. Obtenido de https://es.wikibooks.org/wiki/Programación_Orientada_a_Objeto/Caracter%C3

Wikidat. (s.f.). *Wikidat*. Obtenido de <https://es.wikidat.com/info/extensible-markup-language>