



Universidad
Carlos III de Madrid

WeSweat: Geolocalización social de deportistas

TRABAJO DE FIN DE GRADO

Autor: Carlos Muñoz Villar

Tutor: Antonio Berlanga de Jesús

Resumen

El proyecto desarrollado a lo largo de estas páginas corresponde a la elaboración de la empresa WeSweat, S.L. El objetivo con el que se constituye esta compañía es el de crear una gama de productos para simplificar la creación de eventos deportivos a través de la generación de redes de deportistas, tanto individuales como colectivos, facilitando en último término su organización.

El principal aliciente para la creación de esta empresa radica en la identificación de dos problemáticas fundamentales existentes en el día a día de los deportistas habituales. El primer problema es la falta de integrantes a la hora de alcanzar el aforo mínimo necesario para llevar a cabo una sesión deportiva, como puede ser un partido en el caso de deportes de equipo. Esta falta normalmente es generada por incompatibilidades de horarios. El segundo problema radica en el desconocimiento de los aficionados a la práctica de un deporte de si existen personas con sus mismos intereses deportivos y características en la zona en que residen o llevan a cabo dicho deporte. Ambas problemáticas suelen llevar a la pérdida de interés por el ejercicio en cuestión, que en última instancia pueden potenciar el abandono del mismo.

La gran cantidad de deportistas no conectados en conjunción con la penetración de dispositivos móviles inteligentes y servicios basados en la nube representan una gran oportunidad de negocio para el desarrollo de una aplicación móvil y web que solucione las problemáticas detectadas mediante la creación de una gran red de deportistas geolocalizada.

Una vez identificada la oportunidad de negocio y propuesta la idea para explotarla, se pretenderá a lo largo de este documento describir el proceso completo de desarrollo del producto móvil WeSweat. Se comenzará analizando el entorno que rodea e influye en la empresa. Se describirá el proceso de análisis, diseño e implementación de cada uno de los componentes de la arquitectura del producto, y se detallarán además lo concerniente a temas legales y fiscales, para concluir desarrollando un plan económico financiero de previsión para la empresa.

TABLA DE CONTENIDO

1	Introducción	14
1.1	Contexto Actual.....	14
1.2	Objetivos	16
2	Análisis del estado del arte	17
2.1	Usuarios	17
2.2	Smartphones.....	18
2.3	Proveedores.....	20
2.4	Plataforma en cliente.....	22
2.4.1	Arquitectura de Android	22
2.5	Plataformas en servidor	25
2.5.1	Parse.com.....	25
2.5.2	Heroku.....	26
2.6	Competidores.....	26
2.6.1	Amenaza de entrada de nuevos competidores	26
2.6.2	Intensidad de la competición actual.....	27
2.6.3	Aplicaciones similares	28
3	Análisis del Sistema.....	29
3.1	Definición del Sistema.....	29

3.1.1	Alcance del sistema.....	29
3.1.2	Limitaciones y restricciones del sistema.....	30
3.2	Requisitos de Usuario	30
3.2.1	Requisitos de capacidad.....	33
3.2.2	Requisitos de restricción.....	42
3.3	Casos de uso.....	44
3.3.1	Diagrama de casos de uso.....	53
3.4	Requisitos de software.....	53
3.4.1	Requisitos funcionales	55
3.4.2	Requisitos de rendimiento.....	66
3.4.3	Requisitos de interfaz.....	68
3.4.4	Requisitos de operación.....	71
3.5	Matriz de trazabilidad	73
4	Diseño del sistema	75
4.1	Alternativas de diseño	75
4.2	Arquitectura del sistema.....	77
4.2.1	Servidor A.....	78
4.2.2	Servidor B.....	81
4.2.3	Cliente	83

4.2.4	Interfaz de usuario	87
5	Entorno de operación	94
5.1	entorno de trabajo en cliente: Alternativas y Selección	94
5.2	Configuración del entorno de trabajo en cliente.....	94
5.2.1	Configuración de dependencias adicionales del SDK de Android.....	95
5.2.2	Configuración del proyecto.....	95
5.2.3	Gradle e Inclusión de dependencias	96
5.2.4	Inclusión de módulos	98
5.2.5	Inclusión de ficheros .jar	98
5.2.6	Descripción de librerías utilizadas.....	98
5.2.7	Sistema de control de versiones	100
6	Planificación y Presupuesto	102
6.1	Planificación	102
6.1.1	Análisis	102
6.1.2	Diseño	103
6.1.3	Implementación y Pruebas	103
6.1.4	Gantt	103
6.2	Presupuesto	104
7	Conclusiones y trabajos futuros.....	105

7.1	Trabajos futuros.....	105
7.2	Conclusiones:	106
8	Marco legal, Forma Jurídica y Fiscalidad.....	107
8.1	Forma jurídica	107
8.2	Procedimiento legal	107
8.3	Fiscalidad.....	108
8.4	Cuadro Resumen.....	109
8.5	Patentes	110
8.6	Protección de datos	111
9	Glosario	111
10	Bibliografía	113
11	Anexo I: Análisis económico - financiero	115
11.1	Ingresos, costes e inversión	115
11.1.1	Ingresos.....	115
11.1.2	Costes.....	119
11.1.3	Inversión.....	121
11.2	Pérdidas y ganancias.....	122
11.3	Flujos de caja.....	126
11.4	Valoración de la Empresa.....	131

11.5	Balance	136
11.6	Sensibilidad	137
11.7	Indicadores financieros	142
12	Anexo II: Estudio de descarga de aplicaciones móviles	143
13	Anexo III: Ciclo de vida de las aplicaciones móviles	147
13.1	Introducción	147
13.2	Etapas del ciclo de vida	147
13.3	Cuantificación	148
14	Anexo IV: Diagrama de Gantt.....	150
15	ANEXO V: Introducción y conclusiones en inglés.....	152
15.1	Abstract.....	153
15.2	Objectives.....	154
15.3	Motivation.....	156
15.4	Document structure.....	157
15.4.1	Introduction	157
15.4.2	State of the art	157
15.4.3	System Analysis.....	158
15.4.4	System Design	158
15.4.5	Operation environment	158

15.4.6	Planning and estimated Budget	159
15.4.7	Legal	159
15.4.8	Conclusions and further work.....	159
15.4.9	Bibliography	159
15.4.10	Annexes.....	160
15.5	Conclusions and further work.....	161
15.5.1	Future work.....	161
15.5.2	Conclusions	162

Índice de tablas e ilustraciones

Tabla 1: Evolución de propietarios de dispositivos móviles	19
Tabla 2: Valoración de MBaaS	20
Tabla 3: Plantilla de requisitos de usuario	31
Tabla 4: Características de un requisito	33
Tabla 5: RC_01	34
Tabla 6: RC_02	34
Tabla 7: RC_03	35
Tabla 8: RC_04	35
Tabla 9: RC_05	36
Tabla 10: RC_06	37
Tabla 11: RC_07	37
Tabla 12: RC_08	38
Tabla 13: RC_09	38
Tabla 14: RC_10	39
Tabla 15: RC_11	40
Tabla 16: RC_12	40
Tabla 17: RC_13	41
Tabla 18: RC_14	41
Tabla 19: RR_01	42
Tabla 20: RR_02	43
Tabla 21: RR_03	43

Tabla 22: RR_04	44
Tabla 23: Plantilla de casos de uso	45
Tabla 24: CU1	46
Tabla 25: CU2	46
Tabla 26: CU3	47
Tabla 27: CU4	47
Tabla 28: CU5	48
Tabla 29: CU6	49
Tabla 30: CU7	49
Tabla 31: CU8	50
Tabla 32: CU9	50
Tabla 33: CU10	51
Tabla 34: CU11	51
Tabla 35: CU12	52
Tabla 36: RF_01	55
Tabla 37: RF_02	56
Tabla 38: RF_03	57
Tabla 39: RF_04	57
Tabla 40: RF_05	58
Tabla 41: RF_06	59
Tabla 42: RF_07	59
Tabla 43: RF_08	60
Tabla 44: RF_09	61
Tabla 45: RF_10	61

Tabla 46: RF_11.....	62
Tabla 47: RF_12.....	63
Tabla 48: RF_13.....	63
Tabla 49: RF_14.....	64
Tabla 50: RF_15.....	65
Tabla 51: RF_16.....	65
Tabla 52: RF_17.....	66
Tabla 53: RP_01.....	67
Tabla 54: RP_02.....	67
Tabla 55: RP_03.....	68
Tabla 56: RI_01.....	69
Tabla 57: RI_02.....	70
Tabla 58: RI_03.....	70
Tabla 59: RO_01.....	72
Tabla 60: RO_02.....	72
Tabla 61: Matriz de trazabilidad	75
Tabla 62: Resumen de obligaciones fiscales	110
Tabla 63: Ingresos anuales.....	119
Tabla 64: amortización mensual	121
Tabla 65: Inversión y amortización	122
Tabla 66: Cuenta de resultados	126
Tabla 67: Flujo de caja del proyecto	129
Tabla 68: Entrada de Deuda y FP	130
Tabla 69: Deuda	130

Tabla 70: Flujos de caja del accionista	131
Tabla 71: Betas de sectores similares	132
Tabla 72: Beta cualitativa	133
Tabla 73: betas y tasas de descuento	135
Tabla 74: VAN y TIR	135
Tabla 75: Suposiciones	138
Ilustración 1: Comparativa de práctica deportiva (Género y compañía).....	15
Ilustración 2: Ratio descargas gratuitas / de pago.....	17
Ilustración 3: Evolución de sensores en dispositivos móviles.....	18
Ilustración 4: Cuota de mercado de SO móviles	21
Ilustración 5: Arquitectura del SO Android	25
Ilustración 6: Aplicaciones con más de 50.000 descargas	27
Ilustración 7: Diagrama de casos de uso.....	53
Ilustración 8: Comunicación Cliente - Servidor A.....	76
Ilustración 9: Arquitectura del sistema.....	77
Ilustración 10: Instantánea de datos almacenados en Parse (Usuarios).....	79
Ilustración 11: Instantánea de datos almacenados en Parse (Actividades)	80
Ilustración 12: Diseño relacional de la base de datos.....	81
Ilustración 13: Comunicación Cliente – Servidor B.....	82
Ilustración 14: Arquitectura modulada en cliente	85
Ilustración 15: Flujo de datos: Petición.....	86
Ilustración 16: Flujo de datos: Respuesta	86
Ilustración 17: Material Design Guidelines: Ejemplo.....	88

Ilustración 18: Capturas de vista principal.....	89
Ilustración 19: Capturas de vista filtro	90
Ilustración 20: Capturas de vista Login / Registro	91
Ilustración 21: Captura de vista actividad.....	92
Ilustración 22: Capturas de vista crear actividad	93
Ilustración 23: Creación de nuevo proyecto en AS.....	96
Ilustración 24: Fichero de compilación Gradle del proyecto	97
Ilustración 25: inicialización del repositorio Git (1)	100
Ilustración 26: inicialización del repositorio Git (2)	101
Ilustración 27: Histórico de commits en Git.....	102
Ilustración 28: Gantt (reducido).....	103
Ilustración 29: Valores iniciales WeSweat Standard.....	117
Ilustración 30: Valores iniciales WeSweat Pro.....	117
Ilustración 31: Evolución del número de usuarios.....	118
Ilustración 32: Ingresos mensuales.....	119
Ilustración 33: Evolución de beneficios	126
Ilustración 34: Previsión de TIR sin TV	138
Ilustración 35: Previsión de TIR con TV	139
Ilustración 36: Previsión de VAN sin TV	139
Ilustración 37: Previsión de VAN con TV	140
Ilustración 38: Sensibilidad de indicadores de viabilidad	141

1 INTRODUCCIÓN

A día de hoy el uso de los dispositivos móviles inteligentes o smartphones abarca la práctica totalidad de aficionados a la práctica deportiva.

Existen multitud de aplicaciones deportivas para los dispositivos móviles, pero ninguna propuesta explota el factor social en el ámbito deportivo.

Nuestra propuesta de valor consiste en facilitar y fomentar la práctica de deporte mediante cooperación entre deportistas mediante su Smartphone. A través de nuestro producto, aficionados a practicar cierto deporte podrán encontrar a personas con similares gustos y comunicarse con ellas, pudiendo proponer eventos deportivos públicos visibles geográficamente al resto de deportistas que forman parte de nuestra red. Éstos podrán informarse en detalle de cada evento, comunicarse con los organizadores y decidir si asisten al evento.

1.1 CONTEXTO ACTUAL

Es innegable que en la actualidad nos hallamos ante una nueva fase del desarrollo humano, en la que los dispositivos electrónicos juegan un papel esencial y protagonista. Tal es la velocidad con la que estos elementos se han introducido en nuestras vidas, que a veces no nos damos cuenta de la relación tan estrecha que con ellos tenemos. Pocas son las horas del día en las que no llevamos uno en el bolso, bolsillo e incluso muñeca [1].

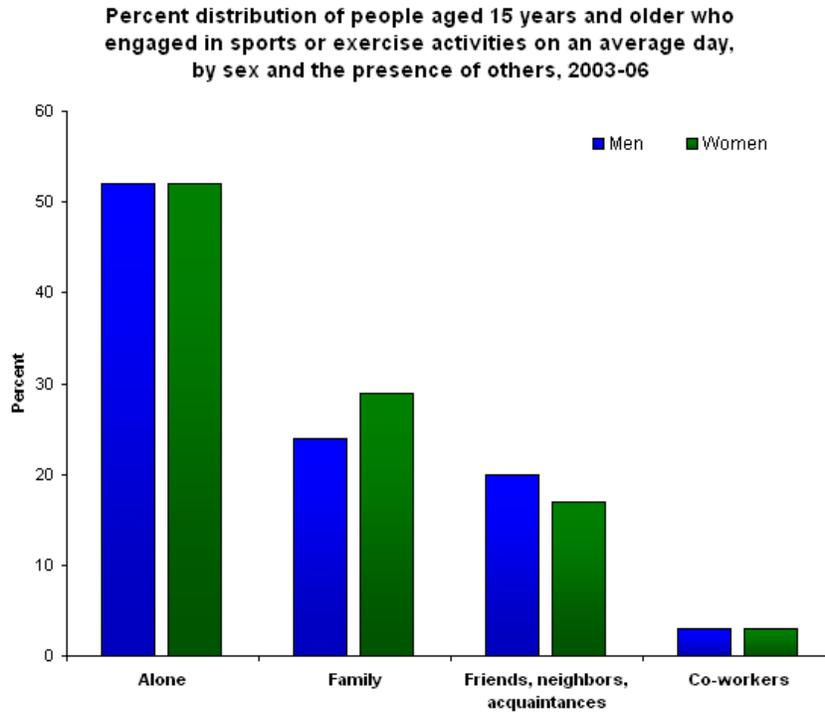


Ilustración 1: Comparativa de práctica deportiva (Género y compañía)

Por otra parte, los hábitos de práctica deportiva son esenciales para nuestra salud, y conviene destacar que más de la mitad de los individuos que practican deporte lo hacen en solitario. En la era de tecnología en que vivimos es innegable que las interacciones sociales vía telemática han tomado parte en esta cifra.

La tecnología del software aplicada al deporte goza de gran aceptación hoy en día. Sin embargo, se ha identificado una oportunidad de negocio en el ámbito de la interconexión de deportistas tanto de carácter individual como colectivo.

La gran cantidad de deportistas no conectados en conjunción con la penetración de dispositivos móviles inteligentes y servicios basados en la nube representan una gran oportunidad de negocio para el desarrollo de una aplicación móvil y web que solucione las problemáticas detectadas mediante la creación de una gran red de deportistas geolocalizada.

Este proyecto surge con la idea de modificar el proceso, sacando partido de nuestros dispositivos móviles para encontrar compañeros con quien practicar deporte. Las posibilidades que esto ofrece son elevadas.

En ocasiones a los aficionados al deporte nos surgen distintas cuestiones que este proyecto pretende responder:

- Soy aficionado a este deporte, pero no conozco a nadie que lo practique alrededor.
- Soy un gran profesional de este deporte, y mis conocidos no suponen un reto para mí.
- Me encantaría iniciarme en este deporte, pero, ¿cómo empezar?

1.2 OBJETIVOS

Conocido el contexto que influye en la actividad de esta empresa, a continuación se revelan los objetivos a alcanzar a lo largo de la realización del proyecto. Estos objetivos determinarán el grado de cercanía entre el trabajo realizado y el inicialmente deseado, y con tal fin se analizará el nivel de consecución de los mismos en las etapas finales del trabajo.

- **Revelar el contexto y situación de las variables de entorno de la empresa:** Tanto desde el punto de vista empresarial como tecnológico, se pretenderá describir el estado del arte de cada componente con potencial influencia en la ejecución del proyecto. Estas variables abarcaran por tanto desde las tecnologías a utilizar en cliente y servidor hasta los proveedores, competidores o potenciales clientes del producto.
- **Estudiar alternativas de diseño y obtener una conclusión óptima:** Se pretende detectar y analizar las distintas vías de consecución del proyecto en términos de diseño. Tras analizar las ventajas e inconvenientes de cada una, uno de los objetivos es seleccionar una arquitectura de diseño que optimice el funcionamiento de cada componente.
- **Describir el proceso de construcción y proporcionar un prototipo del cliente:** Un objetivo esencial a alcanzar es revelar los detalles surgidos durante la implementación del diseño elegido. Además, la consecución ideal de este objetivo desemboca en ilustrar un prototipo del cliente, las diferentes vistas que lo conforman y el funcionamiento de su lógica de negocio en comunicación con los servidores.
- **Servir como guía para similares desarrollos en el futuro:** A través del detallado proceso seguido a lo largo del proyecto se pretende convertir este documento en una fuente de información útil a consultar para la toma de decisiones acertadas en futuros proyectos desarrollados por terceros.

2 ANÁLISIS DEL ESTADO DEL ARTE

2.1 USUARIOS

Dada la saturación del mercado de aplicaciones móviles la capacidad de negociación de los clientes es elevada. Se considerará en primer término como potenciales clientes a todos aquellos individuos que a) dispongan de un Smartphone y b) practican con asiduidad al menos un deporte. A nivel nacional, un 63,2% de residentes en España dispone de un Smartphone , y en torno a un 40% practica habitualmente al menos un deporte [2]. En concreto esta cifra se hace favorable en tanto que son dos componentes relacionadas y siguen un patrón similar si se realiza una segmentación por edades.

En cuanto a la relación entre adquisiciones del producto y precio del mismo, sin lugar a dudas el cliente de aplicaciones móviles es extremadamente sensible al precio. A continuación puede observarse una relación entre descargas gratuitas y de pago con carácter histórico, así como una previsión de su evolución en los próximos años [3]:

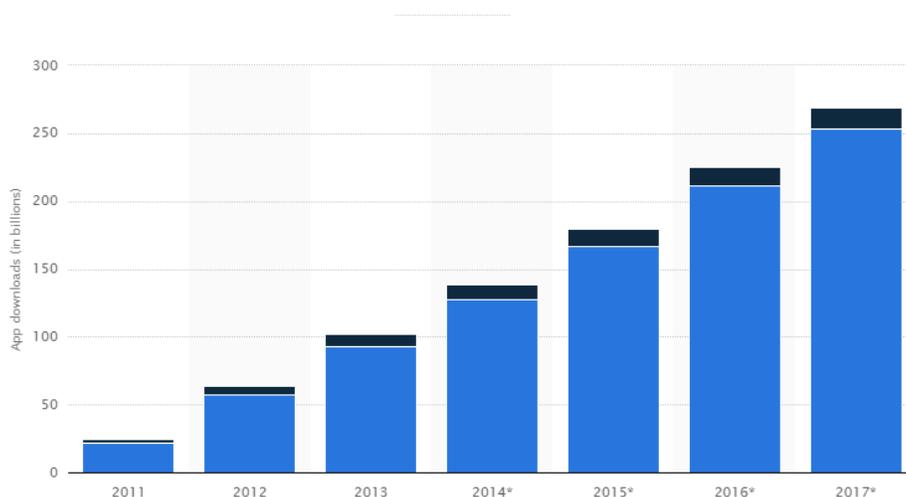


Ilustración 2: Ratio descargas gratuitas / de pago

Como puede observarse, el ratio de descargas de aplicaciones de pago con respecto al total de adquisiciones apenas alcanza el 6% [3] . Si bien esta proporción se incrementa anualmente, se trata de un

dato muy reducido, a tener en cuenta especialmente a la hora de definir el modelo de negocio de una aplicación móvil como producto.

2.2 SMARTPHONES

Los smartphones son dispositivos telefónicos de carácter móvil con capacidades de manejo de datos y ejecución de tareas similares a las de un ordenador. En su gran mayoría son táctiles y disponen de puertos para su conexión a internet vía Wi-Fi o conexión de datos. Además habitualmente cuentan con una serie de sensores y puertos muy variada, entre los que se encuentran con frecuencia GPS o Bluetooth.

Es este último aspecto, los sensores, uno de los que han gozado de mayor importancia en la generación más reciente de dispositivos móviles. La industria del smartphone ha hecho hincapié en la introducción de innovadores sensores [4]:

- Lectores de huella dactilar: con objetivos de seguridad.
- Giroscopios y acelerómetros: Si bien ya existentes en generaciones más antiguas de smartphones, la importancia que toman para el tracking de movimiento de usuarios ha potenciado sus capacidades y desarrollo.
- Monitores de salud: Pulsímetros o sensores de temperatura son los más comunes.

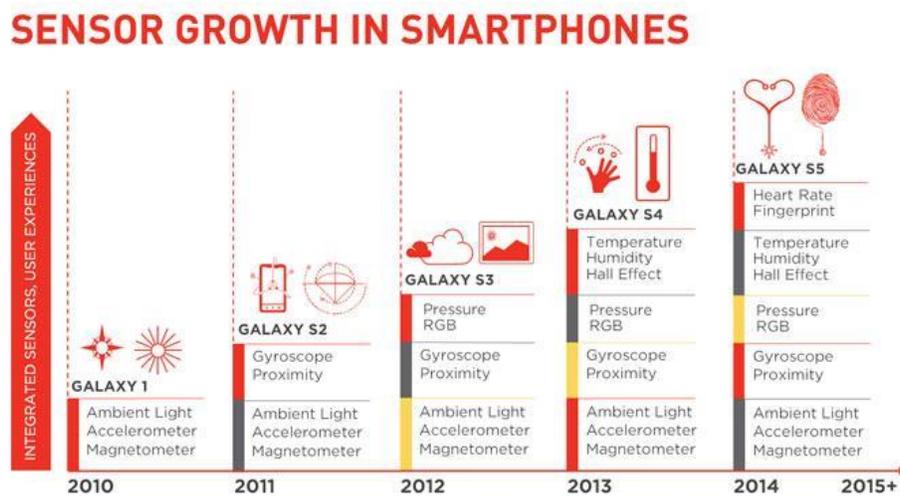


Ilustración 3: Evolución de sensores en dispositivos móviles

El sistema operativo hace uso de esta gran cantidad de receptores de información para proporcionar al usuario herramientas de utilidad en diversos aspectos de su vida.

Es de destacar que en territorio americano, el 90% de los adultos disponen de un teléfono móvil, siendo casi un 60% smartphones. Las estadísticas en Europa toman un cariz muy similar. Con ello el alcance de las aplicaciones móviles para estos dispositivos gozan de un alcance potencial enorme [5].

Cell owners in 2014		Smartphone owners in 2014	
<i>Among adults, the % who have a cell phone</i>		<i>Among adults, the % who have a cell phone</i>	
	Have a cell phone		Have a smartphone phone
All adults	90%	All adults	58%
Sex			
a Men	93 ^b	a Men	61
b Women	88	b Women	57
Race/ethnicity*			
a White	90	a White	53
b African-American	90	b African-American	59
c Hispanic	92	c Hispanic	61 ^b
Age group			
a 18-29	98 ^{cd}	a 18-29	83 ^{bcd}
b 30-49	97 ^{cd}	b 30-49	74 ^{cd}
c 50-64	88 ^d	c 50-64	49 ^d
d 65+	74	d 65+	19
Education level			
a High school grad or less	87	a High school grad or less	44
b Some college	93 ^b	b Some college	67 ^b
c College+	93 ^b	c College+	71 ^b
Household income			
a Less than \$30,000/yr	84	a Less than \$30,000/yr	47
b \$30,000-\$49,999	90	b \$30,000-\$49,999	53
c \$50,000-\$74,999	99 ^{ab}	c \$50,000-\$74,999	61 ^b
d \$75,000+	98 ^{ab}	d \$75,000+	81 ^{abc}
Community type			
a Urban	88	a Urban	64 ^c
b Suburban	92	b Suburban	60 ^c
c Rural	86	c Rural	43

Source: Pew Research Center Internet Project Survey, January 9-12, 2014. N=1,006 adults. Note: Percentages marked with a superscript letter (e.g., ^a) indicate a statistically significant difference between that row and the row designated by that superscript letter, among categories of each demographic characteristic (e.g., age).

* The results for race/ethnicity are based off a combined sample from two weekly omnibus surveys, January 9-12 and January 23-26, 2014. The combined total n for these surveys was 2,008; n=1,421 for whites, n=197 for African-Americans, and n=236 for Hispanics.

PEW RESEARCH CENTER

Tabla 1: Evolución de propietarios de dispositivos móviles

2.3 PROVEEDORES

En el desarrollo de aplicaciones móviles no existe el concepto de proveedor habitual, si bien identificamos a las empresas que proveen de servicios en la nube para la gestión e interconexión de clientes. Concretamente se pueden distinguir las siguientes categorías

2.3.1.1 Empresas de Mobile Backend as a Service:

Proveen como servicio la vinculación en la nube, servicios analíticos, integración con redes sociales y gestión de usuarios. A tener en cuenta para la selección de proveedor, además de los costes, por su carácter de servicio tecnológico ha de valorarse el número de usuarios que utilizan el servicio así como la percepción media que los usuarios tienen de sus características [6]:

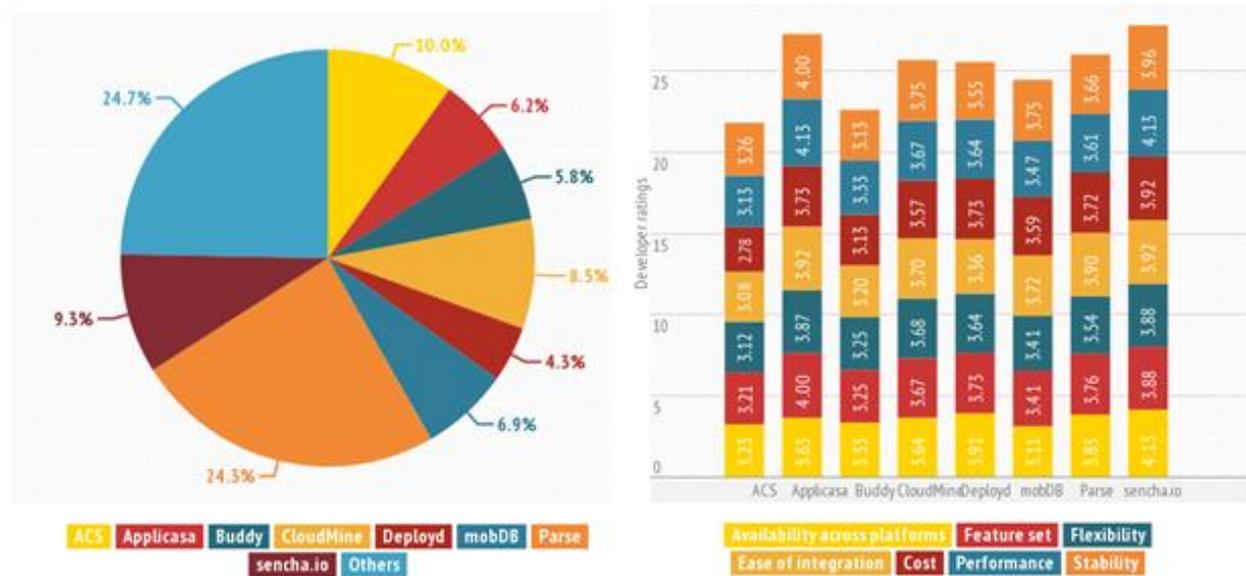


Tabla 2: Valoración de MBaaS

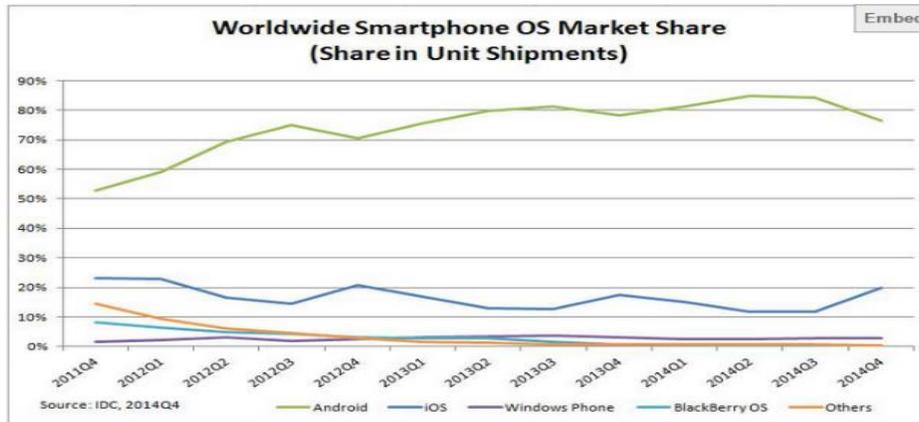
Como puede observarse en las ilustraciones anteriores, en términos de cuota de mercado y preferencia de uso, Parse es el más utilizado con diferencia y cuenta con el soporte de una gran empresa, Facebook. Es importante destacar que **su poder de negociación es reducido**, puesto que la variedad de alternativas que proveen de los mismos servicios es extensa.

2.3.1.2 Empresas de Platform as a Service

Proveen como servicio la infraestructura y plataforma para el desarrollo del backend. El concepto es similar al hosting. Destacan entre ellas Heroku, AWS y Google App Engine. Permite mayor capacidad de modificación en comparación con MBaaS, a la par que mayor cantidad de trabajo, puesto que el backend. Al igual que en las anteriores, **el poder de negociación en su papel de proveedor es escaso.**

2.3.1.3 Empresas proveedoras del Sistema Operativo

Proveen la API sobre la que se desarrollará el producto en clientes. En el caso de nuestra empresa se trata de Google para el caso de desarrollo en Android y Apple en el desarrollo IOS. Concretamente nos proveerá de nuevas herramientas para el diseño mediante actualizaciones de su SDK, tanto de carácter gratuito como de pago.



Period	Android	iOS	Windows Phone	BlackBerry OS	Others
Q4 2014	76.6%	19.7%	2.8%	0.4%	0.5%
Q4 2013	78.2%	17.5%	3.0%	0.6%	0.8%
Q4 2012	70.4%	20.9%	2.6%	3.2%	2.9%
Q4 2011	52.8%	23.0%	1.5%	8.1%	14.6%

Source: IDC, 2014 Q4

Ilustración 4: Cuota de mercado de SO móviles

Como se puede ver en la ilustración anterior, en este caso el poder de negociación es algo más elevado: La gran base de usuarios de las dos principales compañías, proveedoras de sistema operativo móvil, en torno al 97% de smartphones mundiales [7], hace de sus precios un componente muy poco elástico. Sin embargo, cabe destacar que dichos precios no han variado excesivamente en el tiempo de vida de ambos sistemas operativos.

2.4 PLATAFORMA EN CLIENTE

La plataforma seleccionada para el desarrollo del cliente Android ¹es un sistema operativo móvil basado en el núcleo de Linux. Aunque en origen su hardware objetivo fueron los dispositivos móviles, a día de hoy alcanzan un gran paquete de dispositivos, entre los que encontramos tablets, televisiones, relojes, vehículos e incluso gafas de realidad aumentada. Su diseño se centra en dispositivos móviles que cuentan con pantalla táctil.

Android fue adquirido en 2005 por Google, y a día de hoy cuenta con la mayor cuota de mercado de todos los sistemas operativos orientados a dispositivos móviles. El 78% de los dispositivos móviles actuales funcionan con sistema operativo Android, y las predicciones sugieren que esta cifra no hará sino aumentar [3].

Es de destacar además que Android es el sistema operativo que cuenta con el mayor número de aplicaciones instalables del mercado [8].

La principal fuente de éxito de este sistema operativo frente a los de su competencia es el reducido coste de adquisición e implementación que tiene para las empresas productoras de dispositivos móviles, así como su capacidad de personalización y adaptación a las necesidades de dichas empresas. Además, la naturaleza “open-source” de la que goza este sistema operativo le ha permitido la construcción de una gran comunidad de desarrolladores que colaboran generando nuevas características, mejorando las existentes, e incluso adaptando el sistema operativo a dispositivos que de forma oficial no tienen capacidad para soportarlo.

2.4.1 Arquitectura de Android

Desde el punto de vista de diseño y arquitectura, Android se compone de 4 capas fundamentales [9]:

Kernel Linux: La capa base es un kernel Linux modificado con ciertos cambios de arquitectura. En Android cumple una funcionalidad similar a la que cumple en otros sistemas operativos que cuentan con un kernel Linux para gestionar el núcleo del mismo. Muy resumidamente, provee de drivers y software para comunicar y controlar el hardware del dispositivo. Además actúa como capa de abstracción entre el

¹ Como se analiza en detalle en el plan de negocio realizado paralelamente a este proyecto, se pretende desarrollar la aplicación para el resto de sistemas operativos móviles principales: IOS y Windows Phone.

hardware y el resto de capas de software. La popularidad y extensión de este tipo de kernel permite que la portabilidad del sistema operativo sea una tarea relativamente sencilla.

Librerías: La siguiente capa de Android son las librerías nativas. Esta capa permite al dispositivo manejar diferentes tipos de data. Las librerías nativas están desarrolladas en c o c++ y son específicas al hardware que complementan. Las más destacadas son las siguientes:

- **Surface Manager:** Su funcionalidad principal es la de componer los elementos de navegación que habrán de estar disponibles en la pantalla en cada momento, así como gestionar las diferentes ventanas correspondientes a las distintas aplicaciones en ejecución en cada momento.
- **Media Framework:** Conjunto de códecs que permiten la reproducción y grabación de multimedia en distintos formatos.
- **SQLite:** librerías de soporte para el sistema gestor de bases de datos SQLite, principal componente de persistencia de datos en Android
- **WebKit:** Utilizado para renderizar contenido HTML en el navegador.
- **OpenGL:** Utilizado para renderizar contenido grafico 2D o 3D en la pantalla.

Android Runtime: Se compone de dos elementos, la Dalvik Virtual machine y las librerías Core Java:

- **Dalvik Virtual Machine:** Es un tipo de máquina virtual de java usada en dispositivos Android para ejecutar aplicaciones. Está optimizada para entornos con poca capacidad de procesamiento y memoria reducida. Al contrario de las JVM habituales, no ejecuta ficheros .class, sino que ejecuta ficheros .dex. Estos ficheros se generan en tiempo de compilación y precisamente mejoran su rendimiento en los entornos de capacidad reducida nombrados anteriormente. Permite además la generación de múltiples instancias de la máquina virtual, lo que aporta mayor aislamiento y seguridad.
- **ART:** Art es la nueva máquina virtual introducida por Google y que desde Lollipop (Android 5.0) sustituye completamente a la DVM. Cuenta con un gran número de ventajas frente a Dalvik, como compilación AOT (Ahead of Time) y grandes mejoras en el recolector de basura, que elevan significativamente el rendimiento del Sistema operativo y sus aplicaciones.
- **Core Java:** Conjunto de librerías ligeramente diferente a las habituales Java SE y ME, pero que proveen de la mayor parte de funcionalidad contenida en éstas.

Framework de Aplicaciones: Capa con la que las aplicaciones interactúan directamente. Los paquetes incluidos gestionan la funcionalidad básica del teléfono, como gestión de recursos, gestión de llamadas de voz... Desde el punto de vista del desarrollador de aplicaciones, provee de un sistema de herramientas que permite el desarrollo de aplicaciones. Destacan los siguientes paquetes de herramientas:

- Activity Manager: Encargado de gestionar el ciclo de vida de las Activities.
- Content Providers: Gestionan la compartición de datos entre aplicaciones.
- Telephony Manager: Gestiona las llamadas de voz, con lo que se requerirá su uso en aplicaciones que requieran esta funcionalidad.
- Resource Manager: Maneja ciertos recursos a utilizar en nuestra aplicación.

Aplicaciones: Componen la capa más alta de la arquitectura, y es donde nuestras aplicaciones van a alojarse. Todas las aplicaciones que podemos encontrar en el habitual lanzador de aplicaciones de nuestro dispositivo se encuentran en esta capa.

En la ilustración siguiente puede observarse de manera gráfica la arquitectura del sistema operativo Android:

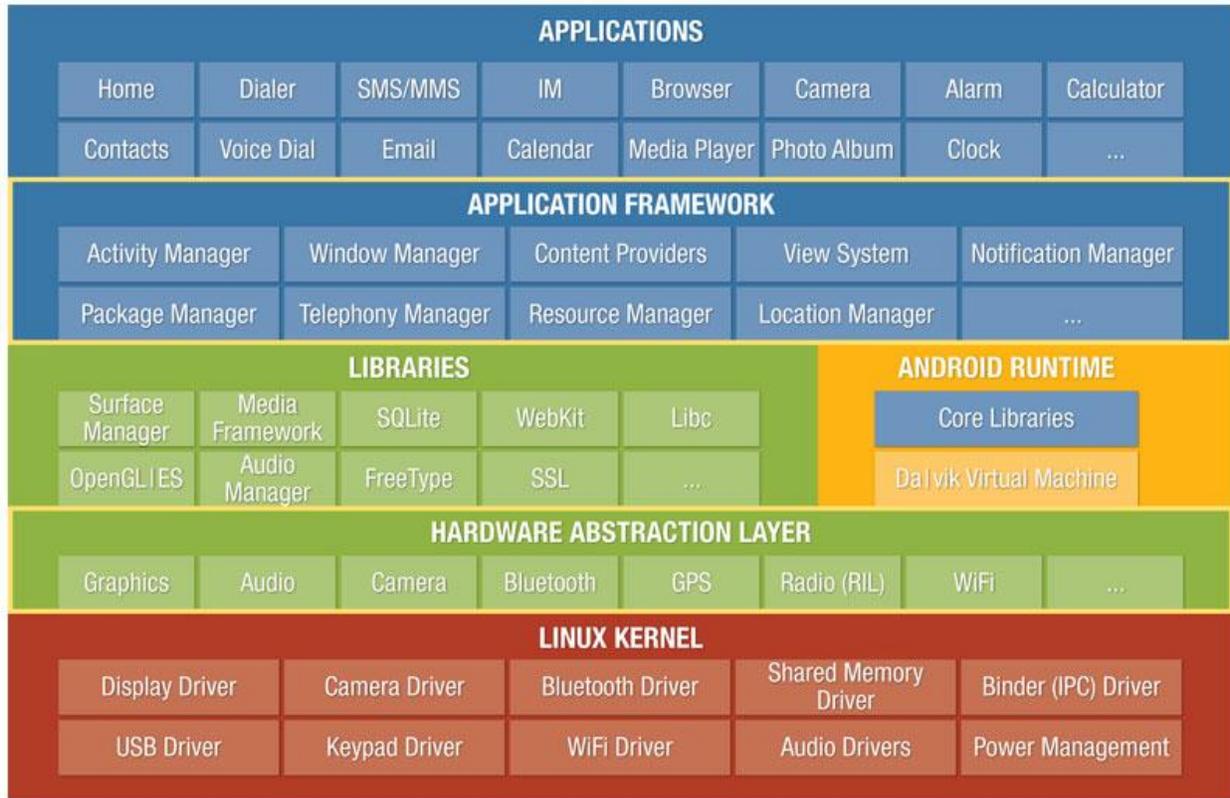


Ilustración 5: Arquitectura del SO Android

2.5 PLATAFORMAS EN SERVIDOR

2.5.1 Parse.com

Parse es una plataforma que abstrae en gran medida al programador de la ingeniería de sistemas habitualmente requerida para la creación de un backend funcional, para que este pueda centrarse en la presentación de los datos. Mediante un reducido número de líneas de código permite realizar operaciones CRUD en la nube. Dispone de librerías adaptadas disponibles para las principales opciones de desarrollo en cliente: IOS, OSX, Android, Windows Phone, Windows, Javascript, .NET, Unity, PHP y Xamarin, así como una API REST que cubre cualquier plataforma con capacidad de realizar peticiones HTTP. Dispone además de un gran catálogo de características.

- Gestión y visualización de datos desde consola de administración web.

- Gestión de estadísticas (Número de instalaciones, estadísticas de uso, peticiones realizadas,
- Notificaciones Push

Aunque su uso básico es gratuito, dispone de planes de pago para aplicaciones que por su volumen de usuarios requieren un uso mayor de sus servicios. Esta plataforma fue adquirida por Facebook en 2013.

2.5.2 Heroku

Heroku es una “Platform as a service” que soporta distintos lenguajes de programación. A diferencia de Parse, no ofrece la funcionalidad completa de servidor, sino que únicamente abstrae del montaje de la plataforma, siendo el programador el que desarrolla íntegramente la lógica del backend. Desde su creación en Junio de 2007, ha extendido su soporte de lenguajes a Java, Node.js, Scala, Clojure y Python (no documentado) PHP. Aunque supone mayor esfuerzo por parte del desarrollador en comparación con Parse, hay ciertos aspectos que hacen que su uso en determinadas situaciones sea preferible. Destaca especialmente el coste de la escalabilidad: el crecimiento de la aplicación no obliga a suscripción a un plan de pagos.

2.6 COMPETIDORES

2.6.1 Amenaza de entrada de nuevos competidores

El sector de aplicaciones deportivas sociales no es un sector ampliamente explotado, especialmente a nivel nacional. Existen aplicaciones que aportan soluciones parciales pero ninguna acaba de resolver la búsqueda de actividades deportivas interesantes alrededor del usuario basándose en sus preferencias.

En cuanto a las barreras de entrada, en primera instancia el sector requiere una inversión inicial muy reducida para crear un producto y distribuirlo. Sin embargo el proceso de promoción dada la gran cantidad de competidores y sustitutivos sí que requiere de inversión. El **conseguir una comunidad de deportistas disponible en los primeros estadios del producto** es esencial, ya que si el usuario no encuentra actividades deportivas cuando comienza a utilizar la aplicación desechará nuestro producto dada la facilidad con que este es sustituido.

Por otro lado las economías de escala pueden alcanzarse de forma sencilla en este mercado, en tanto que una vez implantada, por ejemplo, en una comunidad autónoma, la adaptación del producto a nivel

nacional tiene un coste muy reducido, incluso a nivel internacional no supone costes elevados más allá de los necesarios para su promoción.

Es de especial interés para generar barreras de entrada ante posibles competidores establecer **colaboraciones y acuerdos con distintas asociaciones y federaciones deportivas**. Al ser el primer producto que cubre esta necesidad no se tiene conocimiento de acuerdos similares previamente existentes.

2.6.2 Intensidad de la competición actual

Para el estudio actual se considerarán como competidoras todas las **empresas productoras de aplicaciones móviles circunscritas en categorías iguales o similares a nuestro producto**. En concreto: *Sports, Health and Fitness, Social, Lifestyle, Entertainment*. El porcentaje medio de aplicaciones exitosas es de en torno a un 5,75%, mientras que el de aplicaciones de baja calidad es de un 14.25%. Los cálculos realizados en detalle se encuentran en el Anexo II.

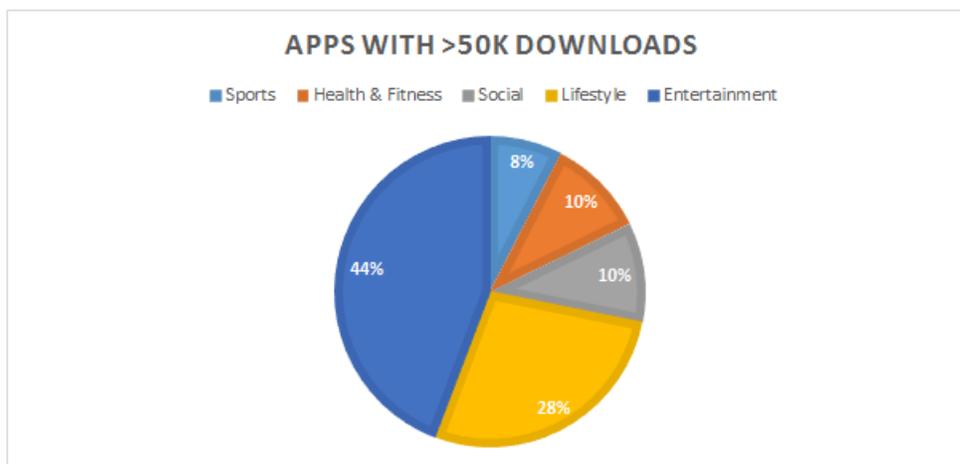


Ilustración 6: Aplicaciones con más de 50.000 descargas

Concretamente, como competidores cercanos, se considerarán todas aquellas aplicaciones catalogadas como "Sport". El porcentaje de éxito en este caso es de en torno a un 4% de las aplicaciones existentes, mientras que el de aplicaciones de baja calidad es de un 13%.

2.6.2.1 Concentración

Existen innumerables empresas de diferente tamaño produciendo aplicaciones en estas categorías. Existen casi 300000 aplicaciones catalogables como competidoras (ver anexo II), y el promedio de creación de productos por compañía desarrolladora es de en torno a 5 [10]. Por tanto, se compite por espacio en el Smartphone de clientes con en torno a 60000 empresas.

2.6.2.2 Diversidad

La competencia es ampliamente diversa. Existen miles de aplicaciones que aportan soluciones complejas, parciales o no suficientemente desarrolladas que pretenden solucionar el problema de la práctica social del deporte. Por otro lado los competidores existentes tienen su base en decenas de países, en los que habitualmente focalizan su mercado. Sin embargo, aunque no se centren en el mercado español, han de tenerse en cuenta ya que pueden cubrir algunas de las funciones que nuestro producto plantea si llegan a tener penetración en el mercado nacional.

2.6.2.3 Diferenciación

El grado de diferenciación en este mercado dada la saturación del mismo es muy elevada. Los productos existentes son muy diferentes, aportando diferentes funcionalidades y soluciones parciales al problema planteado, pero ninguno hasta el momento planteando la comunicación y descubrimiento de nuevas actividades deportivas en la manera en que nuestro producto pretende hacerlo.

2.6.3 Aplicaciones similares

2.6.3.1 Runtastic / Endomondo / Strava

Tres ejemplos de productos que dominan el mercado de aplicaciones deportivas en la actualidad. De manera generalizada cumplen un doble propósito: por un lado el de mantener un registro de actividades deportivas realizadas por el usuario y por otro el de hacer de ellas una herramienta social con la que mostrar las hazañas deportivas llevadas a cabo por el usuario a sus contactos. No suponen un competidor directo puesto que, aunque su segmento de mercado es el de las aplicaciones móviles deportivas, no pretenden la interconexión de deportistas con el fin de que éstos practiquen deporte juntos.

2.6.3.2 *Timpik*

Timpik es una de las pocas aplicaciones existente en el mercado con funcionalidad similar a la propuesta en este proyecto. Permite, previo registro en el cual se establecen preferencias de deporte, encontrar eventos deportivos relacionados con tus gustos en la zona indicada. Como ventajas, cuenta con versión para varios sistemas operativos móviles así como con sitio web. Sin embargo, la aplicación ha sido desarrollada en código no nativo (han optado por código web), con lo que su rendimiento es reducido especialmente en móviles de bajas especificaciones. Cuenta con cierto grado de gamificación de perfiles, búsqueda basada en geolocalización y gestión de equipos.

Este proyecto no pretende pues crear algo totalmente novedoso, sino que tiene como objetivo el de mejorar propuestas como las anteriormente mostradas mediante una oferta de características más amplia y presentada de una manera más amigable y funcional. Análisis del sistema

3 ANÁLISIS DEL SISTEMA

3.1 DEFINICIÓN DEL SISTEMA

A lo largo de este apartado se definirán las directrices del sistema a desarrollar en este proyecto. Se pretenderá obtener una visión detallada (al margen de los matices del diseño) de la funcionalidad completa del sistema.

3.1.1 Alcance del sistema

En primer lugar es necesario definir el problema que este sistema pretende resolver. Dado que el proyecto actual basa su propósito en la interacción vía dispositivo móvil de usuarios que practican deportes de equipo, se desarrollará una aplicación móvil y un sistema de servidores para la comunicación, que constará de un servidor para almacenamiento e intercambio de datos y otro dedicado exclusivamente al sistema de mensajería entre participantes en una actividad.

3.1.2 Limitaciones y restricciones del sistema

3.1.2.1 Cliente

- Dado que se desarrolla para el sistema operativo Android, la implementación de la aplicación móvil se basará en lenguaje Java.
- La aplicación móvil ha de poder ser instalada y lanzada en el mayor número de dispositivos Android posible.
- La información transmitida entre cliente y servidor ha de enviarse cifrada
- Todas las funcionalidades representadas y definidas en los requisitos de usuario han de ser implementadas en el cliente.

3.1.2.2 Servidor

- Si se requiere de algún código adicional al ya disponible en la API de Parse, habrá de ser programado en lenguaje Java.
- La información intercambiada ha de ser autenticada.
- La información intercambiada ha de ser cifrada.
- Todas las funcionalidades representadas y definidas en los requisitos de usuario han de ser implementadas en el servidor.

3.2 REQUISITOS DE USUARIO

Los requisitos de usuario han de ser extraídos a partir de las directrices expuestas por el cliente. En este caso, puesto que es un proyecto sin cliente, la generación de requisitos se basa en decisiones propias en colaboración con el tutor de informática, Antonio Berlanga.

Se distribuirán los requisitos en dos secciones: requisitos de capacidad y requisitos de restricción. Se utilizará la siguiente plantilla para su extensión:

Identificador	
Título	

Descripción	
Prioridad	
Necesidad	
Verificabilidad	
Modificabilidad	

Tabla 3: Plantilla de requisitos de usuario

Donde cada uno de los elementos que aparecen cumple la siguiente funcionalidad:

- **Identificador:** parámetro que distingue unívocamente el requisito. Se utilizará la nomenclatura RX_XX para definir el identificador, donde
 - **RX:** identificador del tipo de requisito, donde RC hace referencia a requisitos de capacidad y RR a requisitos de restricción
 - **XX:** numeración del requisito
- **Título:** texto concreto y de corta extensión que define el requisito.
- **Descripción:** extensión clara del contenido requisito. No ha de dar lugar a duda o ambigüedad.
- **Prioridad:** Define la trascendencia del requisito. Se dispondrá de 3 categorías de prioridad
 - **Alta:** La importancia de este requisito es máxima, de tal manera que su desarrollo será prioritario frente a otros de categoría inferior.
 - **Media:** Su importancia es parcialmente destacable y por tanto han de ser satisfechos con carácter mandatorio, si bien con una relevancia menor que los de prioridad alta.
 - **Baja:** La importancia de este requisito es reducida y por tanto se sitúan a la cola de prioridad en el proceso de desarrollo.
- **Necesidad:** Parámetro que definirá la utilidad del requisito en el sistema. Se dispondrá de 3 categorías de necesidad:

- **Alta:** Se tratará de un requisito imprescindible y su implementación por tanto será imperativa.
- **Media:** La implementación de este requisito es conveniente pues mejora la condición del sistema, pero no es mandatorio su desarrollo.
- **Baja:** Este requisito no ha de ser cumplido con carácter obligatorio.
- **Estabilidad:** Parámetro que define la variabilidad a la que el requisito estará sujeto durante el proceso de desarrollo. Se dispondrá de 3 categorías de estabilidad:
 - **Alta:** Se tratará de un requisito invariable, con lo que no podrá ser modificado a lo largo del desarrollo
 - **Media:** Cambios en el proyecto podrán dar lugar a variación en este requisito
 - **Baja:** Este requisito puede sufrir modificaciones de manera habitual a lo largo del desarrollo.
- **Verificabilidad:** parámetro que define cualitativamente la capacidad de comprobación de cumplimiento del requisito. Se dispondrá de 2 categorías de verificabilidad:
 - **Alta:** El cumplimiento del requisito es comprobable de manera sencilla.
 - **Baja:** Es costoso comprobar el cumplimiento del requisito.

La siguiente tabla resume las características que han de componer un requisito para maximizar su utilidad.

Característica	Explicación
Unitario	El requisito hace referencia a una única cosa
Completo	El requisito es declarado en completitud en un solo lugar,
Consistente	El requisito no contradice ningún otro requisito especificado y es consistente con cualquier documentación de autoridad externa.
Atómico	El requisito no contiene conjunciones. Si se considera que debe utilizarlas significa que el requisito debe ser dividido en requisitos separados

Trazable	El requisito especifica total o parcialmente una necesidad de negocio según lo declarado por el cliente.
Actual	El requisito no queda obsoleto a causa del paso del tiempo.
No ambiguo	El requisito expresa su contenido de manera clara sin necesidad de lenguaje enrevesado, acrónimos, metáforas y demás estructuras literarias. Expresa hechos objetivos y no opiniones subjetivas, estando sujeto a una única interpretación. Frases negativas o compuestas así como verbos sustantivos o adjetivos imprecisos han de ser evitados a la hora de redactar requisitos.
Verificable	La implementación del requisito ha de ser implementable a través de métodos de inspección, demostración, test instrumentados o análisis

Tabla 4: Características de un requisito

Dada la estructura de requisitos y sus características, los siguientes apartados describirán los requisitos de capacidad y restricción recogidos para nuestro sistema.

3.2.1 Requisitos de capacidad

Identificador	RC_01
Título	Registro
Descripción	El usuario ha de poder registrarse en la aplicación
Prioridad	Alta
Necesidad	Alta

Verificabilidad	Alta
Modificabilidad	Baja

Tabla 5: RC_01

Identificador	RC_02
Título	Autenticación
Descripción	El usuario ha de poder autenticarse en la aplicación
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 6: RC_02

Identificador	RC_03
Título	Desconexión
Descripción	El usuario ha de poder desconectarse en la aplicación

Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 7: RC_03

Identificador	RC_04
Título	Consultar actividades deportivas
Descripción	El usuario podrá consultar actividades deportivas a través de la aplicación
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 8: RC_04

Identificador	RC_05
Título	Dar de alta actividad deportiva
Descripción	El usuario autenticado podrá dar de alta una actividad deportiva desde la aplicación
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 9: RC_05

Identificador	RC_06
Título	Dar de baja actividad deportiva
Descripción	El usuario autenticado podrá dar de baja una actividad deportiva desde la aplicación
Prioridad	Media
Necesidad	Media
Verificabilidad	Alta

Modificabilidad	Baja
------------------------	------

Tabla 10: RC_06

Identificador	RC_07
Título	Suscribirse a una actividad
Descripción	El usuario autenticado suscribirse a una actividad existente.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 11: RC_07

Identificador	RC_08
Título	Abandonar una actividad
Descripción	El usuario autenticado abandonar una actividad existente a la que se encuentra suscrito.

Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 12: RC_08

Identificador	RC_09
Título	Consultar actividades creadas
Descripción	El usuario autenticado puede consultar las actividades que ha creado
Prioridad	Media
Necesidad	Media
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 13: RC_09

Identificador	RC_10
Título	Consultar actividades suscritas
Descripción	El usuario autenticado puede consultar las actividades a las que se ha suscrito
Prioridad	Media
Necesidad	Media
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 14: RC_10

Identificador	RC_11
Título	Filtrar búsqueda de actividades deportivas
Descripción	El usuario podrá filtrar su búsqueda de actividades deportivas por deporte, fecha y nivel
Prioridad	Media
Necesidad	Media
Verificabilidad	Alta

Modificabilidad	Alta
------------------------	------

Tabla 15: RC_11

Identificador	RC_12
Título	Ajustar radio de búsqueda de actividades
Descripción	El usuario podrá determinar un rango de distancia desde la ubicación seleccionada a la que buscar actividades
Prioridad	Media
Necesidad	Media
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 16:RC_12

Identificador	RC_13
Título	Especificar localización de búsqueda de actividades

Descripción	El usuario podrá seleccionar una localización de búsqueda desde la cual buscar actividades deportivas
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 17: RC_13

Identificador	RC_14
Título	Comunicación entre participantes en actividades
Descripción	El usuario podrá comunicarse vía chat con los usuarios que participan de una actividad deportiva
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 18: RC_14

3.2.2 Requisitos de restricción

Identificador	RR_01
Título	Conexión a internet
Descripción	Se requerirá conexión a internet para todas las acciones que requieran de intercambio de información con el servidor
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 19: RR_01

Identificador	RR_02
Título	Versión Android
Descripción	Se requerirá un dispositivo móvil con una versión de Android igual o superior a 4.0.3
Prioridad	Alta

Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 20: RR_02

Identificador	RR_03
Título	Acceso a localización
Descripción	Se requerirá acceso al sistema de geolocalización del dispositivo para acciones de búsqueda de actividades alrededor de ubicación actual.
Prioridad	Media
Necesidad	Media
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 21: RR_03

Identificador	RR_04
----------------------	-------

Título	Idioma
Descripción	Los idiomas de la interfaz serán inglés (por defecto) o español (si su dispositivo está configurado en ese idioma).
Prioridad	Media
Necesidad	Media
Verificabilidad	Alta
Modificabilidad	Baja

Tabla 22: RR_04

3.3 CASOS DE USO

Un **caso de uso** es una descripción de los pasos que han de realizarse para llevar a cabo un proceso en la aplicación. Los personajes o entidades que participarán en un caso de uso se denominan actores. Concretamente en este caso, un caso de uso definirá una secuencia de interacciones entre el sistema (la aplicación) y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

Tras completar la extracción y definición de requisitos, pueden generarse los casos de uso de la aplicación. Utilizaremos la siguiente plantilla para desarrollar cada caso de uso:

Caso de Uso	
Actores involucrados	
Objetivo	

Precondiciones	
Postcondiciones	
Requisitos relacionados	

Tabla 23: Plantilla de casos de uso

Donde:

- **Caso de uso:** identifica del caso de uso, lo define de manera unívoca.
- **Actores:** personajes o entidades que participarán en un caso de uso.
- **Objetivo:** propósito final del caso de uso. Ha de ser no ambiguo y de extensión reducida.
- **Precondiciones:** estado en que se encuentra el sistema antes de ejecutarse el caso de uso
- **Postcondiciones:** estado en que queda el sistema tras la ejecución del caso de uso
- **Requisitos relacionados:** requisitos de usuario que guardan relación con el caso de uso

Encontramos dos tipos de actores en la aplicación actual

- **Usuario anónimo:** Aquel usuario que no ha realizado autenticación con el sistema
- **Usuario autenticado:** Aquel usuario que si ha realizado autenticación con el sistema

Con todo ello, se han definido los siguientes casos de uso:

Caso de Uso	Realizar el registro
Actores involucrados	Usuario anónimo
Objetivo	Crear una cuenta de usuario con la que realizar autenticación con el sistema

Precondiciones	Usuario sin cuenta de usuario en el sistema
Postcondiciones	Usuario con cuenta de usuario en el sistema
Requisitos relacionados	RC01

Tabla 24: CU1

Caso de Uso	Realizar la autenticación
Actores involucrados	Usuario anónimo
Objetivo	Acceder como usuario identificado al sistema para poder realizar acciones como tal.
Precondiciones	Usuario no autenticado en el sistema
Postcondiciones	Usuario autenticado en el sistema
Requisitos relacionados	RC02

Tabla 25: CU2

Caso de Uso	Realizar la desconexión
Actores involucrados	Usuario autenticado

Objetivo	Dejar de acceder como usuario identificado al sistema.
Precondiciones	Usuario autenticado en el sistema
Postcondiciones	Usuario no autenticado en el sistema
Requisitos relacionados	RC03

Tabla 26: CU3

Caso de Uso	Buscar actividades alrededor del usuario
Actores involucrados	Usuario autenticado, Usuario anónimo
Objetivo	Encontrar actividades deportivas alrededor de la posición actual del usuario
Precondiciones	El usuario ha concedido acceso a su ubicación
Postcondiciones	Se mostrará en el mapa las actividades encontradas alrededor del usuario
Requisitos relacionados	RC04, RC11, RC12

Tabla 27: CU4

Caso de Uso	Buscar actividades alrededor de localización específica
Actores involucrados	Usuario autenticado, Usuario anónimo
Objetivo	Encontrar actividades deportivas alrededor de la posición definida por el usuario
Precondiciones	
Postcondiciones	Se mostrará en el mapa las actividades encontradas alrededor de la ubicación seleccionada
Requisitos relacionados	RC04, RC11, RC12, RC13

Tabla 28: CU5

Caso de Uso	Crear una actividad
Actores involucrados	Usuario autenticado
Objetivo	Crear una actividad deportiva a partir de los parámetros definidos por el usuario
Precondiciones	El usuario ha realizado la autenticación
Postcondiciones	El sistema almacenará la actividad creada por el usuario.

Requisitos relacionados	RC05
--------------------------------	------

Tabla 29: CU6

Caso de Uso	Eliminar una actividad
Actores involucrados	Usuario autenticado
Objetivo	Eliminar una actividad deportiva creada
Precondiciones	El usuario ha realizado la autenticación y ha creado la actividad
Postcondiciones	El sistema eliminará la actividad del usuario.
Requisitos relacionados	RC06

Tabla 30: CU7

Caso de Uso	Suscribirse a una actividad
Actores involucrados	Usuario autenticado
Objetivo	Unirse a una actividad deportiva existente en el sistema

Precondiciones	El usuario ha realizado la autenticación y ha encontrado una actividad
Postcondiciones	El usuario autenticado quedará registrado como partícipe de la actividad
Requisitos relacionados	RC07, RC04

Tabla 31: CU8

Caso de Uso	Eliminar la suscripción a una actividad
Actores involucrados	Usuario autenticado
Objetivo	Dejar de estar suscrito a una actividad deportiva existente en el sistema
Precondiciones	El usuario ha realizado la autenticación y se encuentra suscrito a la actividad
Postcondiciones	El usuario autenticado deja de estar registrado como partícipe de la actividad
Requisitos relacionados	RC08

Tabla 32: CU9

Caso de Uso	Consultar historial de actividades creadas
--------------------	--

Actores involucrados	Usuario autenticado
Objetivo	Consultar actividades creadas por el usuario
Precondiciones	El usuario ha realizado la autenticación
Postcondiciones	Se mostrará una lista de actividades creadas por el usuario
Requisitos relacionados	RC09

Tabla 33: CU10

Caso de Uso	Consultar historial de actividades en las que se participa
Actores involucrados	Usuario autenticado
Objetivo	Consultar actividades en las que el usuario participa
Precondiciones	El usuario ha realizado la autenticación
Postcondiciones	Se mostrará una lista de actividades en las que participa el usuario
Requisitos relacionados	RC10

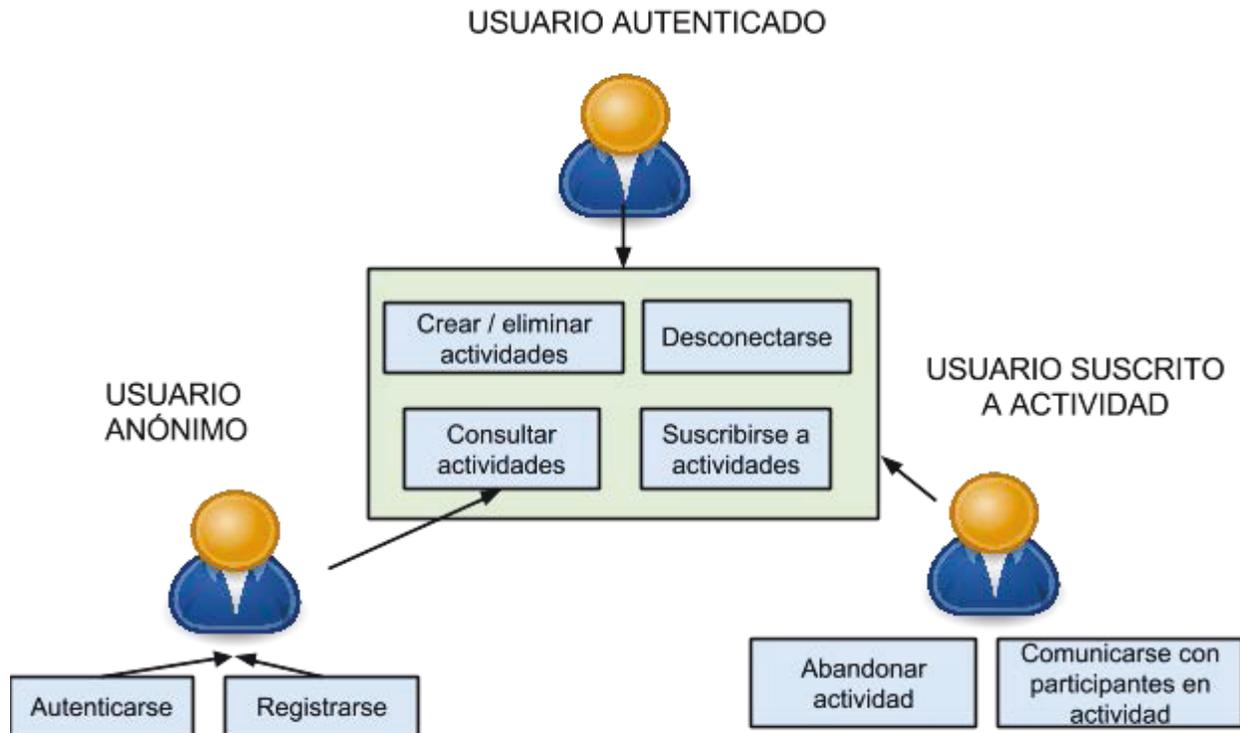
Tabla 34: CU11

Caso de Uso	Enviar mensaje
--------------------	----------------

Actores involucrados	Usuario autenticado
Objetivo	Enviar mensaje a usuarios suscritos a una actividad en la que participa el usuario
Precondiciones	El usuario ha realizado la autenticación y se encuentra suscrito a una actividad
Postcondiciones	El resto de usuarios suscritos recibirán el mensaje enviado.
Requisitos relacionados	RC14

Tabla 35: CU12

3.3.1 Diagrama de casos de uso



(Aclaración: Los usuarios autenticados pueden realizar todas las acciones del cuadro verde, mientras que los usuarios anónimos sólo pueden realizar la acción "Consultar actividades")

Ilustración 7: Diagrama de casos de uso

3.4 REQUISITOS DE SOFTWARE

Los requerimientos de software delimitan la actuación del sistema, y se derivan tanto de los requerimientos de usuario como de los casos de uso. Se dividen con carácter general en dos categorías principales [11]:

- Requisitos funcionales: especifican la función del sistema y/o de sus componentes
- Requisitos no funcionales: especifican criterios que pueden utilizarse para valorar la operación de un sistema y/o de sus componentes. Engloban por tanto todo requisito que no describe funcionalidad del sistema, dividiéndose habitualmente en diversas, subcategorías,

Los elementos que compondrán cada requisito serán los siguientes

- **Identificador:** parámetro que distingue unívocamente el requisito. Se utilizará la nomenclatura RX_XX para definir el identificador, donde
 - **RX:** identificador del tipo de requisito, donde:
 - RF: Requisito funcional
 - RP: Requisito de Rendimiento (Performance) (No funcional)
 - RI: Requisito de Interfaz. (No funcional)
 - RO: Requisito de Operación. (No funcional)
 - RD: Requisito de documentación (No funcional)
 - RS: Requisito de seguridad. (No funcional)
 - **XX:** numeración del requisito
- **Título:** texto concreto y de corta extensión que define el requisito.
- **Descripción:** extensión clara del contenido requisito. No ha de dar lugar a duda o ambigüedad.
- **Prioridad:** Define la trascendencia del requisito. Se dispondrá de 3 categorías de prioridad
 - **Alta:** La importancia de este requisito es máxima, de tal manera que su desarrollo será prioritario frente a otros de categoría inferior.
 - **Media:** Su importancia es parcialmente destacable y por tanto han de ser satisfechos con carácter mandatorio, si bien con una relevancia menor que los de prioridad alta.
 - **Baja:** La importancia de este requisito es reducida y por tanto se sitúan a la cola de prioridad en el proceso de desarrollo.
- **Necesidad:** Parámetro que definirá la utilidad del requisito en el sistema. Se dispondrá de 3 categorías de necesidad:
 - **Alta:** Se tratará de un requisito imprescindible y su implementación por tanto será imperativa.
 - **Media:** La implementación de este requisito es conveniente pues mejora la condición del sistema, pero no es mandatorio su desarrollo.
 - **Baja:** Este requisito no ha de ser cumplido con carácter obligatorio.
- **Estabilidad:** Parámetro que define la variabilidad a la que el requisito estará sujeto durante el proceso de desarrollo. Se dispondrá de 3 categorías de estabilidad:
 - **Alta:** Se tratará de un requisito invariable, con lo que no podrá ser modificado a lo largo del desarrollo
 - **Media:** Cambios en el proyecto podrán dar lugar a variación en este requisito

- **Baja:** Este requisito puede sufrir modificaciones de manera habitual a lo largo del desarrollo.
- **Verificabilidad:** parámetro que define cualitativamente la capacidad de comprobación de cumplimiento del requisito. Se dispondrá de 2 categorías de verificabilidad:
 - **Alta:** El cumplimiento del requisito es comprobable de manera sencilla.
 - **Baja:** Es costoso comprobar el cumplimiento del requisito.
- **Requisito Origen:** Referenciará el requisito de usuario que da lugar a este requisito de software.

3.4.1 Requisitos funcionales

Los requisitos funcionales definen la actuación y función del sistema.

Identificador	RF_01
Título	Registro
Descripción	EL sistema permitirá al usuario registrarse en el sistema a través de la aplicación móvil.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja
Requisito Origen	RC01, RR_01

Tabla 36: RF_01

Identificador	RF_02
Título	Autenticación Convencional
Descripción	EL sistema permitirá al usuario autenticarse en el sistema a través de la aplicación móvil mediante usuario y contraseña.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja
Requisito Origen	RC02, RR_01

Tabla 37: RF_02

Identificador	RF_03
Título	Autenticación Social
Descripción	EL sistema permitirá al usuario autenticarse en el sistema a través de la aplicación móvil mediante su cuenta de Facebook
Prioridad	Alta

Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja
Requisito Origen	RC02, RR_01

Tabla 38: RF_03

Identificador	RF_04
Título	Cierre sesión
Descripción	El sistema permitirá al usuario cerrar la sesión en la aplicación móvil
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja
Requisito Origen	RC_03, RR_01

Tabla 39: RF_04

Identificador	RF_05
Título	Consulta de actividades cercanas
Descripción	El sistema permitirá al usuario consultar en un mapa las actividades cercanas a su posición actual.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja
	RC_04, RR_03, RR_01

Tabla 40: RF_05

Identificador	RF_06
Título	Consulta de actividades en localización seleccionada
Descripción	El sistema permitirá al usuario consultar en un mapa las actividades cercanas a una localización seleccionada manualmente.
Prioridad	Media

Necesidad	Media
Verificabilidad	Alta
Modificabilidad	Baja
Requisito Origen	RC_04, RC_13, RR_01

Tabla 41: RF_06

Identificador	RF_07
Título	Filtro de consulta de actividades
Descripción	El sistema permitirá al usuario filtrar actividades por deporte, distancia, nivel y fechas de inicio y fin.
Prioridad	Media
Necesidad	Media
Verificabilidad	Alta
Modificabilidad	Alta
Requisito Origen	RC_11, RC_12, RR_01

Tabla 42: RF_07

Identificador	RF_08
Título	Visualización de Actividad
Descripción	El sistema permitirá al usuario acceder a una actividad consultada y ver sus detalles: Descripción, fechas, nivel y usuarios participantes.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja
Requisito Origen	RC_04, RR_01

Tabla 43: RF_08

Identificador	RF_09
Título	Dar de alta actividad
Descripción	El sistema permitirá al usuario dar de alta una actividad geolocalizada.
Prioridad	Alta

Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja
Requisito Origen	RC_05, RR_01

Tabla 44: RF_09

Identificador	RF_10
Título	Dar de baja actividad
Descripción	El sistema permitirá al usuario dar de baja una actividad geolocalizada.
Prioridad	Media
Necesidad	Media
Verificabilidad	Alta
Modificabilidad	Baja
Requisito Origen	RC_06, RR_01

Tabla 45: RF_10

Identificador	RF_11
Título	Participar en actividad
Descripción	El sistema permitirá al usuario suscribirse como participante en una actividad consultada
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja
Requisito Origen	RC_07, RR_01

Tabla 46: RF_11

Identificador	RF_12
Título	Abandonar actividad
Descripción	El sistema permitirá al usuario abandonar una actividad en la que está participando.
Prioridad	Media
Necesidad	Media

Verificabilidad	Alta
Modificabilidad	Baja
Requisito Origen	RC_08, RR_01

Tabla 47: RF_12

Identificador	RF_13
Título	Consultar actividades creadas
Descripción	El sistema permitirá al usuario consultar las actividades que ha creado.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja
Requisito origen	RC_09, RR_01

Tabla 48: RF_13

Identificador	RF_14
Título	Consultar actividades en las que se participa
Descripción	El sistema permitirá al usuario consultar las actividades a las que se ha suscrito como participante.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja
Requisito origen	RC_10, RR_01

Tabla 49: RF_14

Identificador	RF_15
Título	Consultar actividades creadas por otros usuarios
Descripción	El sistema permitirá al usuario consultar las actividades que otros usuarios crean.
Prioridad	Alta

Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja
Requisito origen	RC_04, RR_01

Tabla 50: RF_15

Identificador	RF_16
Título	Consultar actividades suscritas por otros usuarios
Descripción	El sistema permitirá al usuario consultar las actividades en las que otros usuarios se han suscrito como participantes.
Prioridad	Media
Necesidad	Media
Verificabilidad	Alta
Modificabilidad	Media
Requisito origen	RC_04, RR_01

Tabla 51: RF_16

Identificador	RF_17
Título	Comunicarse con otros participantes
Descripción	El sistema permitirá al usuario enviar mensajes al resto de participantes de una actividad.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	RC_14, RR_01

Tabla 52: RF_17

3.4.2 Requisitos de rendimiento

Estos requisitos establecen cuantitativamente los parámetros de rendimiento que ha de mantener el sistema.

Identificador	RP_01
Título	Tiempo de carga inicial
Descripción	El sistema deberá ser capaz de realizar la carga inicial y mostrar el mapa en un tiempo máximo de 5 segundos
Prioridad	Alta

Necesidad	Alta
Verificabilidad	Media
Modificabilidad	Media
Requisito Origen	RC_04

Tabla 53: RP_01

Identificador	RP_02
Título	Tiempo de consulta de actividades
Descripción	El sistema deberá ser capaz de consultar actividades alrededor de un punto y mostrarlas en el mapa en un tiempo máximo de 10 segundos con una conexión de 3Mbps.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Media
Modificabilidad	Media
Requisito Origen	RC_04, RC_13

Tabla 54: RP_02

Identificador	RP_03
Título	Tiempo de peticiones a servidores
Descripción	El sistema deberá ser capaz realizar un intercambio de datos petición-respuesta con los servidores en un tiempo máximo de 10 segundos con una conexión de 3Mbps
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Media
Modificabilidad	Media
Requisito Origen	

Tabla 55: RP_03

3.4.3 Requisitos de interfaz

Delimitan los métodos e comunicación entre los distintos módulos del sistema.

Identificador	RI_01
Título	Gestión de peticiones de datos

Descripción	Las peticiones de datos se gestionarán por una aplicación web en la plataforma Parse.com
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Alta
Requisito Origen	

Tabla 56: RI_01

Identificador	RI_02
Título	Gestión de Chat
Descripción	El sistema de envío de mensajes se gestionará mediante un servidor de sockets.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Alta

Requisito origen	
-------------------------	--

Tabla 57: RI_02

Identificador	RI_03
Título	Idioma de la interfaz
Descripción	La interfaz de usuario estará disponible en español e inglés.
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Media
Requisito origen	

Tabla 58: RI_03

Identificador	RI_04
Título	Versión Android

Descripción	Se requerirá un dispositivo móvil con una versión de Android igual o superior a 4.0.3
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Alta
Requisito Origen	RR_02

3.4.4 Requisitos de operación

Estos requisitos delimitarán como serán realizadas las tareas del sistema.

Identificador	RO_01
Título	Conexión a internet
Descripción	Se requerirá conexión a internet para todas las acciones que requieran intercambiar información con el servidor.
Prioridad	Alta
Necesidad	Alta

Verificabilidad	Alta
Modificabilidad	Baja
Requisito origen	

Tabla 59: RO_01

Identificador	RO_02
Título	Tiempo operativo
Descripción	El sistema y sus módulos han de estar operativos un 99,9% del tiempo (tres nueves)
Prioridad	Alta
Necesidad	Alta
Verificabilidad	Alta
Modificabilidad	Baja
Requisito origen	

Tabla 60: RO_02

3.5 MATRIZ DE TRAZABILIDAD

A continuación puede observarse la matriz de trazabilidad, que refleja que todos los requisitos de usuario han sido cubiertos por algún requisito de software:

	RC_01	RC_02	RC_03	RC_04	RC_05	RC_06	RC_07	RC_08	RC_09	RC_10	RC_11	RC_12	RC_13	RC_14	RR_01	RR_02	RR_03	RR_04
RF_01	X														X			
RF_02		X													X			
RF_03		X													X			
RF_04			X												X			
RF_05			X	X											X		X	
RF_06				X								X			X			
RF_07										X	X				X			
RF_08				X											X			
RF_09					X										X			
RF_10						X									X			

RF_									
11			X						X
RF_									
12				X					X
RF_									
13					X				X
RF_									
14						X			X
RF_									
15		X							X
RF_									
16		X							X
RF_									
17								X	X
RP_									
01		X							
RF_									
02		X					X		
RF_									
03									
RI_0									
1									
RF_									
RI_0									
2									

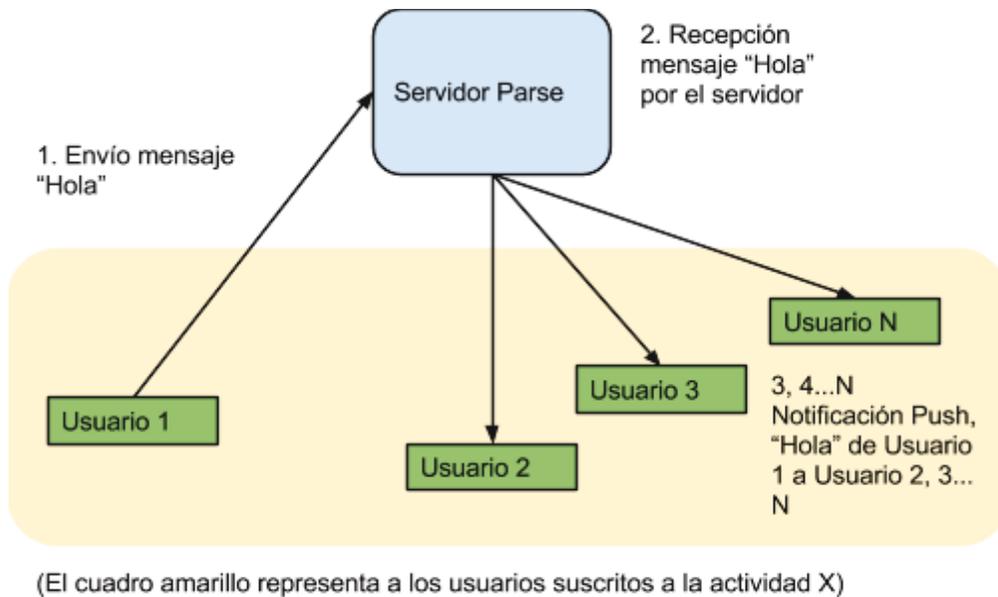


Ilustración 8: Comunicación Cliente - Servidor A

De esta manera, por cada mensaje enviado por un usuario se produciría una llamada al servidor Parse y $N-1$ notificaciones push, donde N representa el número de usuarios participantes de cada actividad. El problema de esto proviene de los planes de pago de Parse, cuya tarificación se basa en el número de notificaciones push y de llamadas al servidor realizadas. Si se hubiera optado por esta opción, dichas llamadas y notificaciones se hubieran disparado, dando lugar a un crecimiento excesivo e innecesario de los costes.

Es en este punto en el que entra el segundo servidor, basado en sockets y nodeJS. Como ya se ha analizado, la plataforma servidor en que se aloja (Heroku) basa su plan de precios en la cantidad de recursos usados, lo que elevaría el coste en función de la cantidad de usuarios que utilicen la aplicación en lugar de en el número de mensajes que cada uno de ellos necesite enviar.

4.2 ARQUITECTURA DEL SISTEMA

Para comprender la arquitectura propuesta para el desarrollo del sistema se ha elaborado un diagrama a modo resumen de los componentes principales del mismo, que representa además sus dependencias y relaciones.

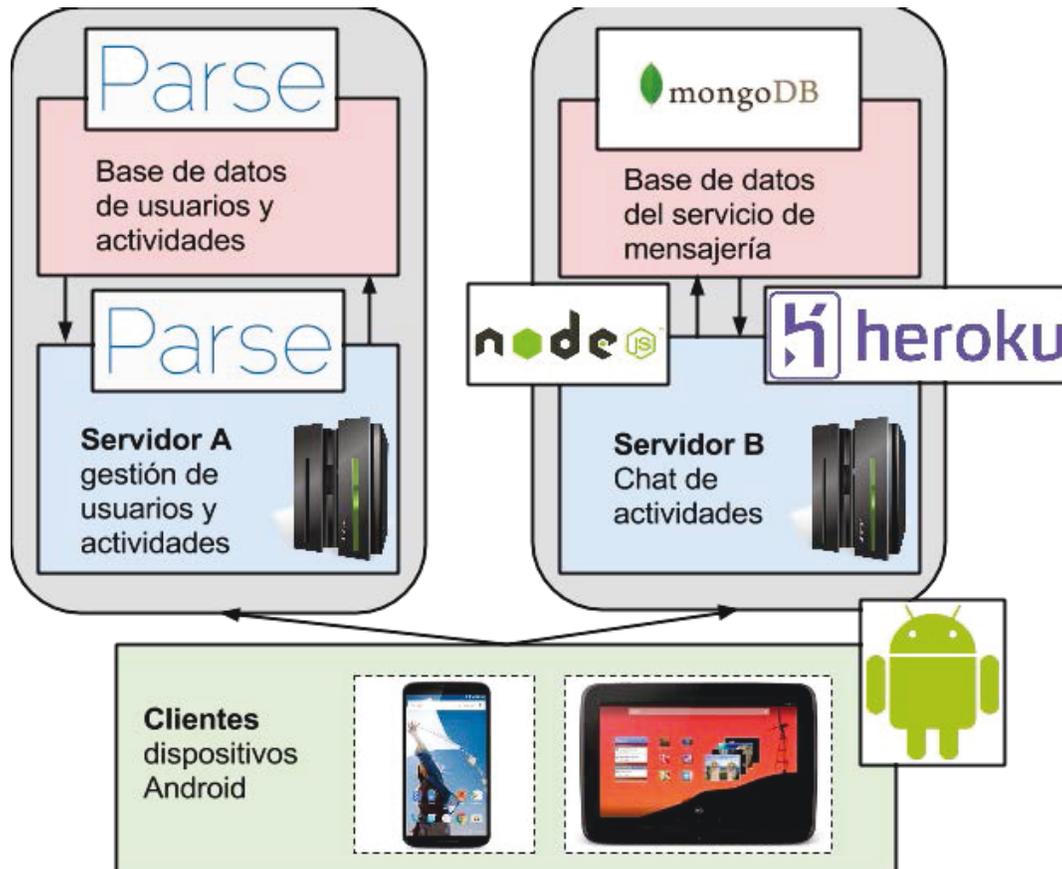


Ilustración 9: Arquitectura del sistema

Como se puede observar en el diagrama, existen 3 componentes principales: dos componentes servidor y uno cliente. Los componentes de servidor se dividen en núcleo (Azul) y modelo (Rojo). Junto a los componentes puede observarse una ilustración representando las plataformas y/o tecnologías seleccionadas para el desarrollo (Parse, Android, mongoDB, nodeJS, Heroku, Android).

El componente servidor etiquetado como servidor A es el encargado de gestionar la persistencia en la nube de usuarios y actividades, así como la relación entre ambos elementos. Al ser Parse el servicio utilizado para este apartado, la gestión de la base de datos es transparente al programador, ofreciéndonos una interfaz de operaciones CRUD. Si bien es conocido que Parse utiliza MongoDB para gestionar los datos, aunque, como se ha comentado, es un factor transparente en este proyecto.

El componente etiquetado como servidor B manejará las salas de chat asociadas a cada actividad. Este servidor se vale de las siguientes herramientas:

- Plataforma Heroku como abstracción de servidor (PaaS).
- La librería de sockets Socket.io para la transmisión de datos.
- Node.js como lenguaje en el servidor.
- MongoDB para almacenar las conversaciones.

Al contrario que ocurre con el servidor A, en este caso toda la gestión deja de ser transparente al programador, con lo que es necesario implementar el funcionamiento al completo. Lo único que se provee como servicio es la plataforma del servidor.

4.2.1 Servidor A

Parse se basa en un modelo de datos relacional, si bien no es un modelo relacional “al uso”. Provee de algunas abstracciones más propias de la programación orientada a objetos. A continuación se realiza una aproximación del modelo de datos utilizando términos propios de los modelos relacionales:

Tabla Users: en ella se almacenan los usuarios y su información. Parse dispone de una API de gestión de usuarios que gestiona las operaciones habituales que tienen que ver con los mismos (registro, autenticación, desconexión y reenvío por correo de contraseñas). En el apartado de cliente se describirá el uso de esta API a través de la clase Java “ParseUser”. Parse además permite además añadir atributos (lo que se entiende en modelo relacional por columnas) adicionales a la tabla de usuarios. En este proyecto se han añadido los siguientes:

- **fbEmail, fbId, fbName:** campos para almacenar información relacionada con Facebook, de los cuales se hace uso cuando el usuario opta por este mecanismo de autenticación.

- **joined_exercises²**: enfocándolo desde un punto de vista de base de datos relacional, este campo es una relación de tipo oneToMany con actividades deportivas, concretamente a las que el usuario se ha suscrito.

objectid	username	password	authData	authData	createdAt	Date	e...	updatedAt	Date	ACL	ACL	email	String	joined_exercises	R...
iiRaXmfKUr	Fyx5Orkq...	(hidden)	Facebook: 880914...		Jan 23, 2015, 19...		(u...	Jan 25, 2015, 23:04		Public R...		(undefined)			View Relations
obrQC6YIaZ	oA2uw1D8...	(hidden)	Facebook: 102041...		Jan 24, 2015, 12...		(u...	Jan 24, 2015, 17:40		Public R...		(undefined)			View Relations
XaysgEU2...	mnwYdDa...	(hidden)	Facebook: 918305...		Jan 24, 2015, 22...		(u...	Jan 24, 2015, 23:12		Public R...		(undefined)			View Relations
apl20gUr3	carlospiles	(hidden)	(undefined)		Jan 24, 2015, 12...		(u...	Jan 24, 2015, 12:17		Public R...		kalospiles@g...			View Relations

Ilustración 10: Instantánea de datos almacenados en Parse (Usuarios)

Tabla Exercises: en ella se almacenan las actividades deportivas existentes en el sistema. La abstracción en cliente para operaciones CRUD de tablas corrientes (no de tipo usuario) se lleva a cabo mediante la API definida en la clase ParseObject. En el apartado de cliente se describirá el uso de esta API. En este caso los campos definidos son los siguientes

- **date_start, date_end:** campos de tipo fecha que identifican hora de inicio y finalización de la actividad deportiva
- **title, description:** título y breve descripción de la actividad deportiva.
- **creator:** enfocándolo desde un punto de vista de base de datos relacional, este campo es una relación de tipo ManyToOne con un usuario, el creador de la actividad. Un usuario puede crear varias actividades pero una actividad solo puede tener un creador.
- **sport:** código numérico que identifica el deporte que se practica en esta actividad.

² Para todas las referencias a actividades deportivas en código se ha optado por el nombre “exercises”, puesto que Activity es una palabra reservada en Android y puede dar lugar a confusión.

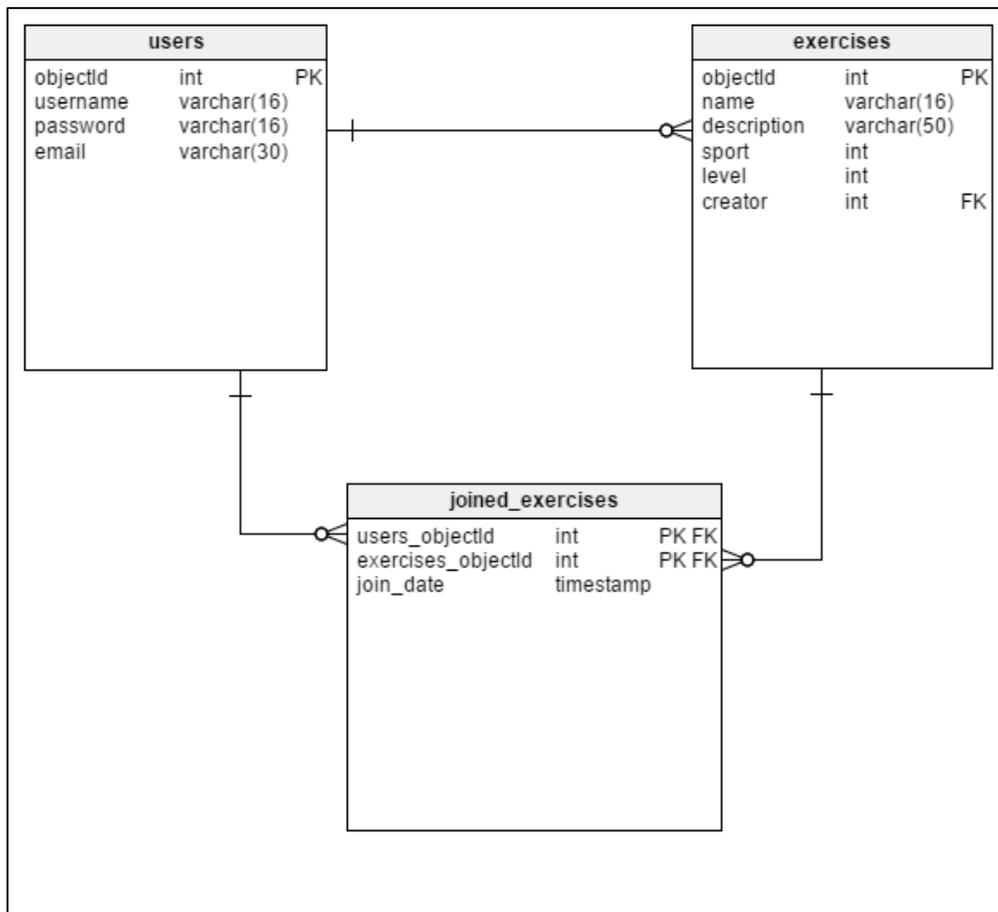


Ilustración 12: Diseño relacional de la base de datos

4.2.2 Servidor B

Como ya se ha comentado, se ha optado por Heroku como PaaS, seleccionando entre los lenguajes de programación el framework basado en Javascript para desarrollo de aplicaciones de servidor Node.js. Este framework permite construir aplicaciones de servidor de manera rápida, sencilla y escalable. Se han utilizado además diversos sub-frameworks para la construcción del backend:

- **ExpressJs:** framework web minimalista para la gestión de peticiones HTTP
- **Socket.IO:** framework para la comunicación en tiempo real basada en eventos (mediante la utilización de sockets). Mediante él se envían y reciben mensajes.
- **Mongoose:** framework para la conexión con la base de datos MongoDB

El funcionamiento de este servidor es sencillo: Cuando un cliente accede a la sala de mensajería de un chat y envía un mensaje, se envía mediante la librería de Socket.IO en cliente el mensaje a todos los usuarios que tienen abierto el socket y que participan de esa actividad.

Además, el mensaje se almacena en la base de datos del servidor. Cuando un cliente abre la aplicación se conecta al socket, envía el MessageId del último mensaje que recibió, y el servidor responde con todos los mensajes posteriores a ese que han sido recibidos en las salas de chat de las actividades a las que el usuario se encuentra suscrito. La interacción vía socket en cliente se desarrollará en posteriores apartados, si bien el siguiente diagrama resume su funcionamiento:

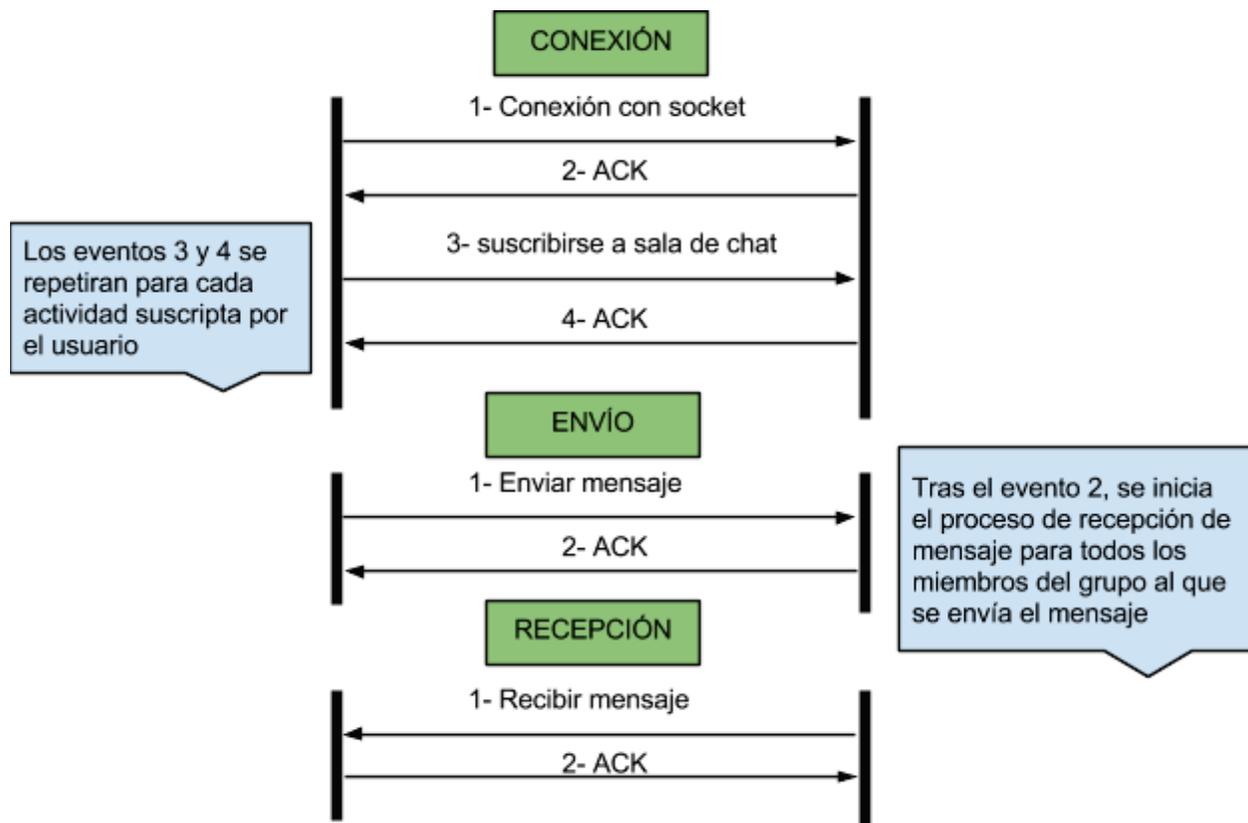


Ilustración 13: Comunicación Cliente – Servidor B

El modelo de datos en este caso es muy sencillo. Simplemente se almacenan las conversaciones cifradas que tienen lugar en las diferentes actividades deportivas. Es de destacar que se ha optado por SGBD MongoDB, apoyado por la API de node.js Mongoose (se definirá en el apartado “Núcleo en Servidor B”).

“Tabla” messages: almacena los mensajes. Cuenta con los siguientes campos

- **MessageId**: campo entero autoincrementado que identifica el mensaje
- **MessageContent**: texto del mensaje.
- **RoomId**: id de la sala de chat, que equivaldrá al identificador unívoco de la actividad en la base de datos del servidor A (ObjectId).

4.2.3 Cliente

Se ha pretendido para el desarrollo de este proyecto hacer uso de una arquitectura limpia y sólida. Una arquitectura que permita un mantenimiento sencillo e independencia entre capas de sistema. Como más adelante se estudia, la API de Android no facilita el uso de una arquitectura por capas en la que datos e interfaz son independientes, puesto que muchos de sus componentes acoplan ambas capas. Diversas son las propuestas realizadas por la comunidad de desarrolladores para alcanzar un modelo de arquitectura que genere un sistema testable y desacoplado. En este proyecto se ha realizado una aproximación reducida de estas propuestas.

4.2.3.1 Vistas: *Arquitectura Model View Presenter*

MVP es un patrón derivado del comúnmente utilizado y extendidamente conocido MVC (Model View controller). Sin embargo MVP ha obtenido por su adaptación a la API de Android gran relevancia para el desarrollo de aplicaciones en esta plataforma.

Esta arquitectura permite separar la capa de presentación de la lógica de la aplicación. Es decir, todo el funcionamiento la interfaz queda separado del modo en que se presenta dicha interfaz en la pantalla. Conviene destacar que se centra únicamente en la capa de presentación, con lo que no es una arquitectura al completo.

4.2.3.2 *Por qué MVP en Android?*

Existe un problema en Android que ha destacado desde su inicio: Todos los elementos de la API relacionados con vistas (Activities, Fragments, Dialogs...), están muy acopladas tanto a la interfaz (Vista) como a la capa de datos (Modelo). Innumerables son las clases relacionadas con Activity en Android que reflejan este acoplamiento. Un ejemplo muy relevante es el CursorAdapter, que mezcla en la misma clase cursores (resultados de acceso a capa de datos) con Adapters (que definen representación de los datos en la vista). La mayoría de las implementaciones de Adapter incurrir en el mismo problema. Por ello se han desarrollado diversos y variados patrones de arquitectura que permiten reducir dicho acoplamiento. En este proyecto se ha decidido utilizar uno de ellos, el patrón Model View Presenter [12].

4.2.3.3 *Componentes en MVP*

- **Presenter:** actúa como un intermediario entre la vista y el modelo. Recupera y accede a los datos del modelo, los transforma y los dispone para su acceso por la vista. Aquí reside la mayor diferencia con el tradicional MVC: El presenter decide que ocurre cuando se interactúa con la vista.
- **View:** Dependiendo de la situación será implementada por una Activity, un Fragment o una View (View como clase Android, no como capa de la arquitectura). Contendrá una referencia al presenter, que podrá ser insertada a través de un inyector de dependencias como Dagger [13]. En este proyecto no se ha utilizado inyector, siendo el programador el encargado de mantener dicha referencia al presenter. La vista únicamente llamará a un método del presenter cada vez que se interactúa con ella (cuando se pulsa un botón, se inserta texto en un campo, etc).
- **Model:** Al igual que en MVC, si las capas de la aplicación se han diseñado correctamente, el modelo será simplemente el método de acceso al dominio de la aplicación o a la lógica de negocio, el proveedor de los datos que se mostraran en la vista.

4.2.3.4 *Dominio y modelo: Casos de uso y datasources*

La mayor parte de arquitecturas propuestas por la comunidad para un desarrollo limpio sugieren la creación de diversos módulos dentro del proyecto. Un esqueleto muy adecuado sería el siguiente [14]:

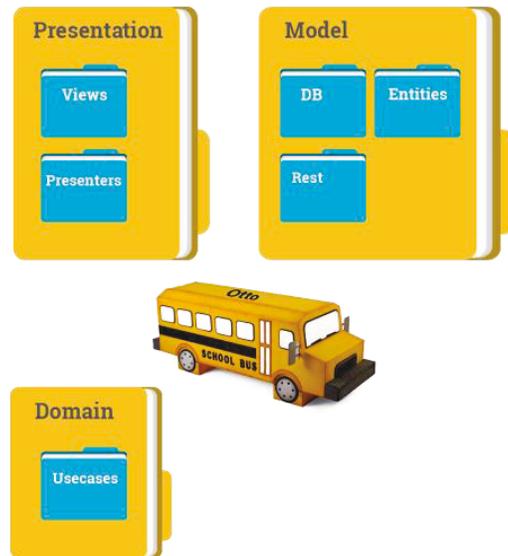


Ilustración 14: Arquitectura modular en cliente

Cada carpeta en este diagrama hace referencia a un módulo.

- La carpeta **presentation** hace referencia a un módulo dependiente de Android, que almacena las vistas y los presentadores, que se describirán en el siguiente apartado (Model View Presenter).
- El módulo **domain** contiene los casos de uso, cada uno representado habitualmente por una clase, y que como no puede ser de otra manera contemplan la lógica de negocio de la aplicación. Serán clases “ejecutables”, en tanto que habrán de implementar un método “ejecutar”, invocado por el presenter, que llevará a cabo cierta lógica de negocio accediendo a las diferentes fuentes de datos (en el módulo del modelo). Se trataría de un módulo Java puro, sin referencias a ninguna clase Android. actuaría como dependencia del módulo Android Presentation.
- La carpeta **model** hace referencia a los datos y a las fuentes de datos, y actúa como dependencia del dominio. Dispone de clases e interfaces con el sufijo DataSource, que representan el encapsulado de acceso a datos y su detalle de implementación (acceso a APIs REST, base de datos, preferencias...)
- El **autobús** representa la comunicación entre módulos, existen diferentes maneras de comunicar cada módulo, entre las que destacan los callbacks y los buses de datos. En este diagrama se nombra Otto [15], una popular librería de Bus de datos Java comúnmente utilizada en Android.

El flujo de información sería el siguiente:

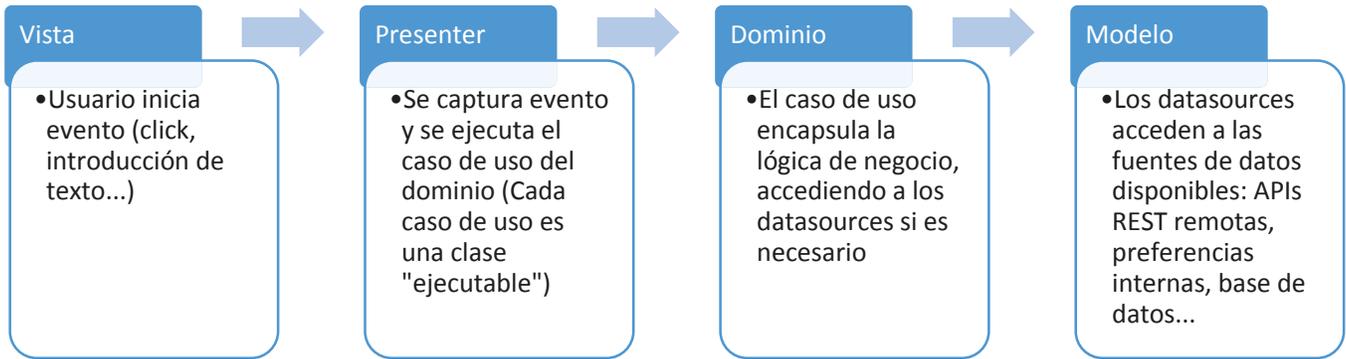


Ilustración 15: Flujo de datos: Petición

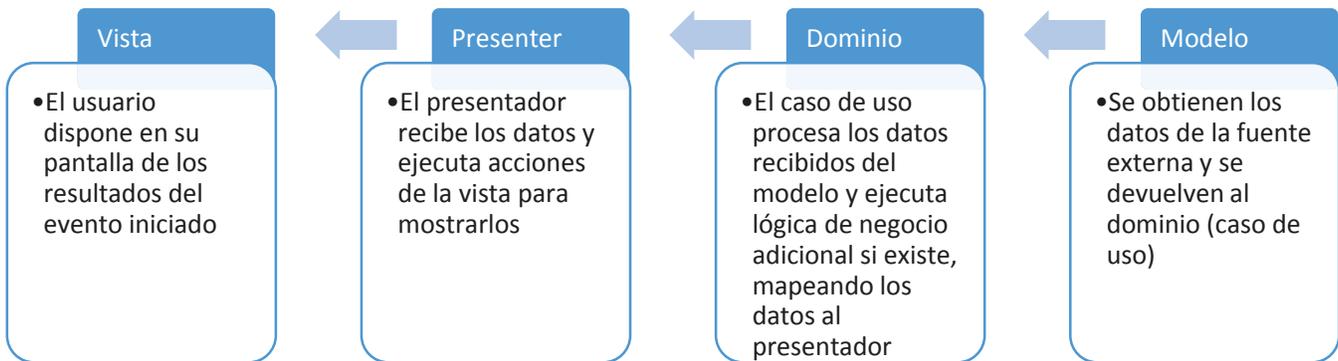


Ilustración 16: Flujo de datos: Respuesta

Entrando en mayor detalle, la estructura de paquetes sería la siguiente:

- **Paquete adapter:** en él se encuentran todos los adapters del proyecto. Los adapters en Android son los encargados de enlazar la capa de datos con numerosos tipos de vistas, como pueden ser listas o grids.
- **Paquete common:** Contiene clases útiles para el resto de paquetes.

- BaseUi y BaseActivity: interfaz y clases base para las actividades del sistema, con referencias a funciones comunes de las mismas (mostrar y ocultar dialogos de carga, mensajes de error, logs de información...)
- Navigator: clase que centraliza los cambios de pantalla y facilita el intercambio de datos entre las mismas.
- Utils: Clase que contiene constantes y funciones de utilidad para el resto de clases.
- SportsApplication: clase que extiende de Application y la convierte en un Singleton, lo que permite proveer de contexto al resto de singletons del sistema.
- **Paquete dialogs:** Contiene todos los Custom dialogs del proyecto, como el selector de días de la semana (que no es nativo en Android, y por tanto ha tenido que ser diseñado y codificado).
- **Paquete domainmodel:** unifica las capas de dominio y modelo de la aplicación. Contiene por tanto:
 - las fuentes de datos (Parse, Chat, Preferencias, Localización...)
 - las entidades o POJOs que actúan como clases básicas de datos
 - las clases de dominio, nombradas como casos de uso
 - las clases que actúan como ejecutoras de los casos de uso, facilitando la ejecución síncrona y evitando el colapso del hilo principal (UiThread)

4.2.4 Interfaz de usuario

El cliente es donde se concentra la interacción con el usuario, siendo el componente principal de la aplicación. En ella el usuario podrá consultar actividades deportivas existentes mediante búsquedas geolocalizadas, consultar sus características y decidir si unirse a ellas. Si opta por registrarse y autenticarse, el usuario podrá suscribirse a actividades y establecer conversaciones con los usuarios existentes. Dispondrá además de herramientas adicionales como filtros de actividades, consulta y edición de perfil o consulta de perfiles de otros usuarios.

El eje de navegación de la aplicación será un panel lateral deslizable conocido en Android como Drawer. A través de él el usuario podrá acceder a las distintas secciones de la aplicación. Cuando realice el registro y autenticación dispondrá tanto en el drawer como en las diferentes pantallas nuevas opciones que le permitirán acceder a las funcionalidades adicionales previamente descritas.

4.2.4.1 Material Design - Guías de diseño de Google

La interfaz del módulo del proyecto que comprende la aplicación Android se ha desarrollado siguiendo los principios expuestos en la guía de diseño de Google “Material Design” [16]. Esta guía comprende una serie de objetivos:

- Crear un lenguaje visual que sintetice los principios clásicos del buen diseño con la innovación y las posibilidades que ofrecen las nuevas tecnologías.
- Desarrollar un sistema común que unifique la experiencia de usuario independientemente de la plataforma o del tamaño de la pantalla.

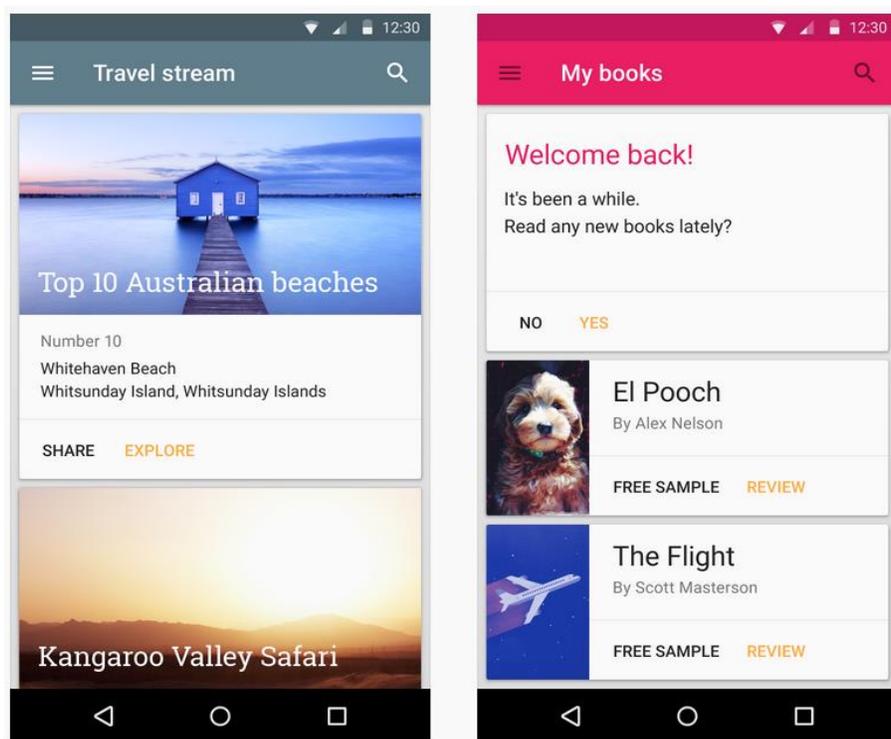


Ilustración 17: Material Design Guidelines: Ejemplo

En la aplicación actual se ha optado por respetar el diseño material especialmente en el uso de componentes (widgets). Entre ellos destacan:

- Uso de tarjetas / cards para mostrar la información

- Uso de tonalidad de color unificada para todas las vistas (tonalidad verdosa en el caso concreto de este proyecto)
- Utilización de botones circulares en la esquina inferior derecha
- Utilización de vista deslizable lateral para la navegación (conocida como navigation drawer)
- Uso de Toolbar, Barra superior que muestra el encabezado de la vista y mantiene los patrones de color básicos de la aplicación.
- Utilización de páginas deslizable para mostrar información cuando el tamaño de la pantalla no es suficiente (conocido como viewPager)
- Utilización de otros componentes Material de entrada de datos (checkboxes, inputs, etc).

4.2.4.2 Vistas de la aplicación:

4.2.4.2.1 Vista principal - Mapa de actividades cercanas:

Tanto si el usuario se encuentra autenticado como si no, la primera pantalla que observará el usuario mostrará un mapa a pantalla completa con la ubicación del usuario y las actividades cercanas. Si no se ha conseguido obtener su localización el mapa no navegará hasta la misma y no mostrará actividades, con lo que se observará un mapa del mundo:

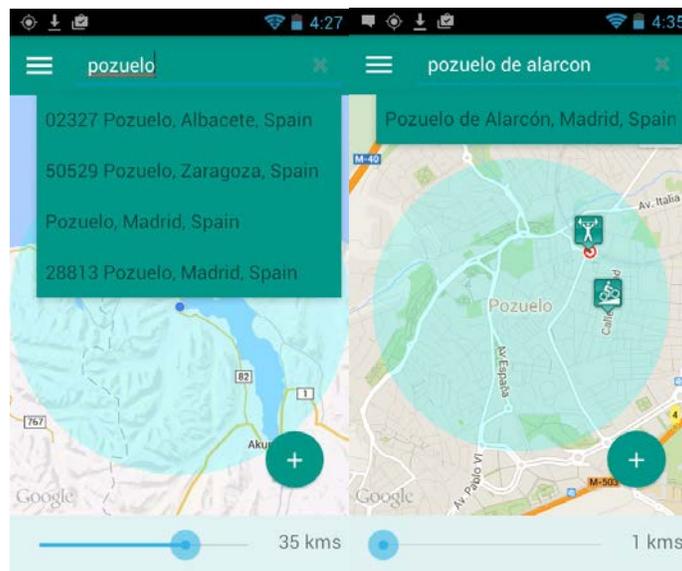


Ilustración 18: Capturas de vista principal

4.2.4.2.2 Vista filtro

En esta vista el usuario tanto si se encuentra autenticado como si no será capaz de establecer filtros de búsqueda de actividades en base a sus preferencias: Horario, nivel de profesionalidad de la actividad, duración, tipo de deporte, etc. Los filtros aplicados en esta vista serán comunicados de inmediato en la vista principal, que automáticamente realizará una nueva búsqueda con los filtros establecidos.

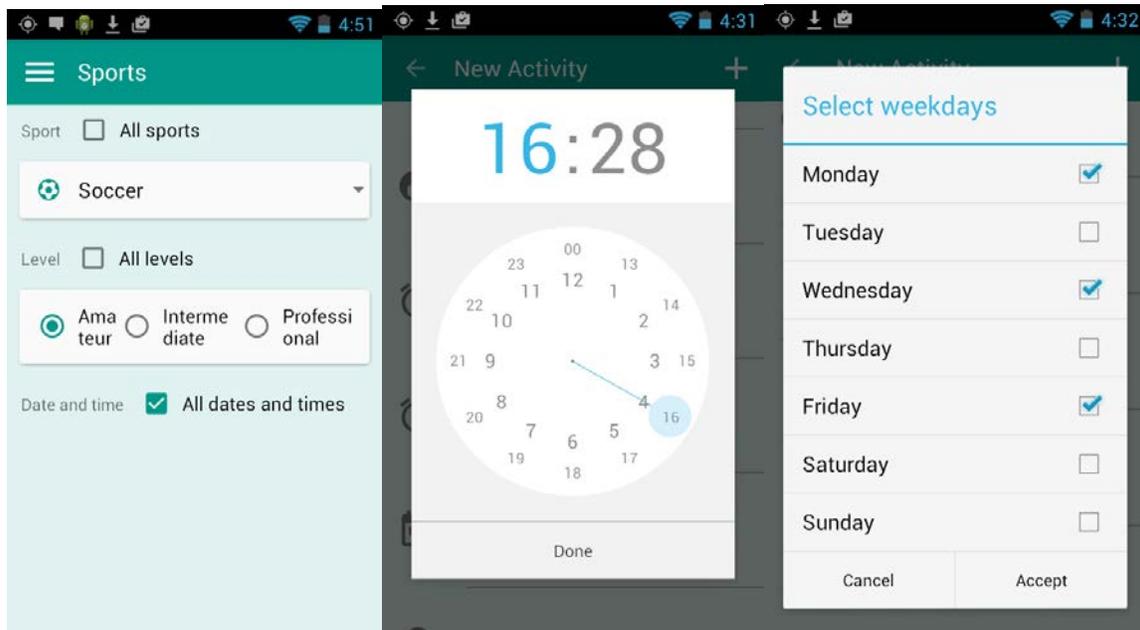


Ilustración 19: Capturas de vista filtro

4.2.4.2.3 Vista Login / Registro

En esta vista el usuario podrá crear una cuenta si no dispone de ella, así como loguearse para tener acceso a mayor funcionalidad en la aplicación.

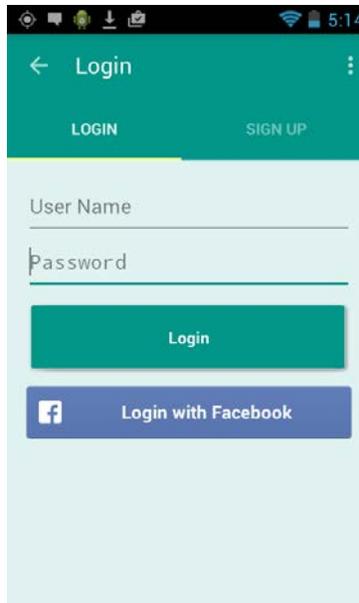


Ilustración 20: Capturas de vista Login / Registro

4.2.4.2.4 Vista actividad

Esta vista es accedida al seleccionar una actividad desde cualquiera de las vías disponibles: Seleccionando una actividad en el mapa de la vista principal, accediendo al listado de actividades suscritas, accediendo al listado de actividades creadas por el usuario en cuestión o por otros usuarios, etc. Se mostrarán las características de la actividad en detalle y el usuario podrá suscribirse si hay espacio disponible, teniendo además acceso al chat.

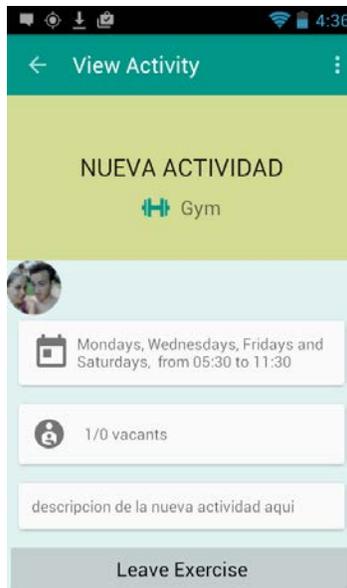


Ilustración 21: Captura de vista actividad

4.2.4.2.5 Vista crear actividad

En esta vista, a la cual se accede a través del FAB³ localizado en la vista principal de los usuarios autenticados, se podrá crear una nueva actividad que se indexará en los servidores. Pueden establecerse los diferentes parámetros que distinguen la actividad: Geolocalización, título y descripción, horario y demás características

³ Floating Action Button, botón circular situado permanentemente en la esquina inferior derecha de la pantalla, según lo establecido en las guías de diseño Material de Google.

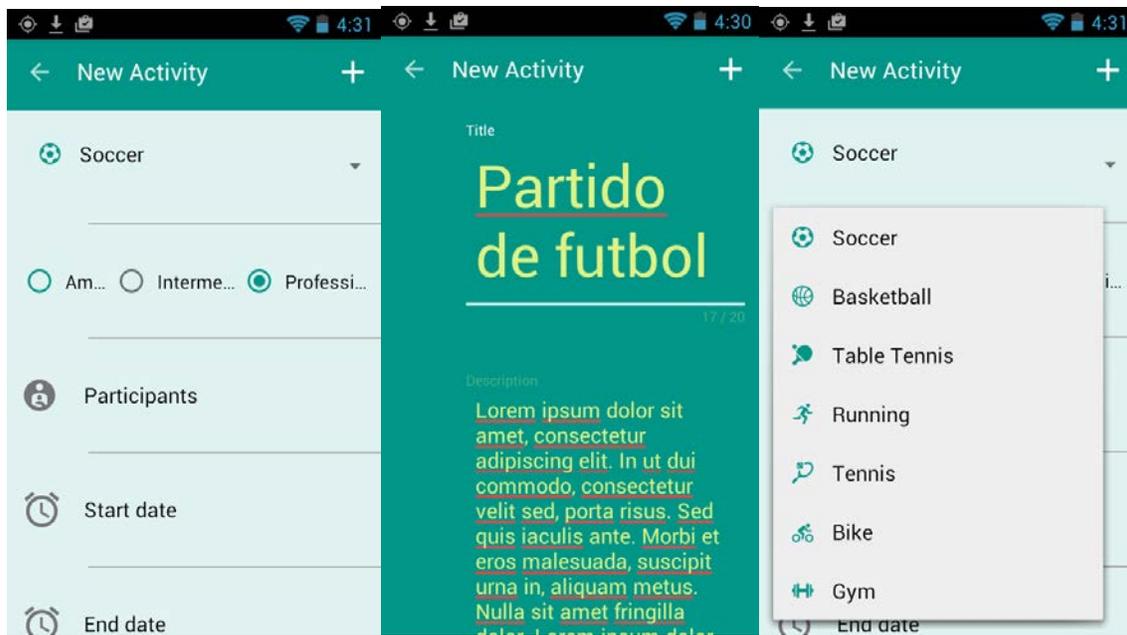


Ilustración 22: Capturas de vista crear actividad

5 ENTORNO DE OPERACIÓN

5.1 ENTORNO DE TRABAJO EN CLIENTE: ALTERNATIVAS Y SELECCIÓN

Para el desarrollo de aplicaciones Android se dispone de dos frameworks o entornos de trabajo soportados oficialmente por Google: ADT, basado en Eclipse, y Android Studio, basado en IntelliJ.

Se ha optado para llevar a cabo este proyecto por Android Studio. Esta plataforma plantea una serie de ventajas que hacen que su competidor haya sido descartado para este proyecto:

- **Gestión de dependencias de una forma cómoda y potente basada en Gradle - Maven:** Sin duda la principal razón por la que esta plataforma se posiciona como principal framework de desarrollo.
- Actualizaciones constantes, novedades y facilidades incorporadas al entorno de manera habitual. Por el contrario, ADT ya comienza a catalogarse como desactualizado [17].
- Sencillez a la hora de **reusar código y recursos**
- Sencillez en **configuración, extensión y personalización** del proceso de desarrollo.
- **Facilita la distribución del código** y por tanto trabajar en equipos (En el caso de este proyecto se ha trabajado en dos máquinas diferentes).
- **Simplificación del proceso de producción:** El proceso de versionado de la aplicación es más accesible, lo que incluye diferentes tareas: distribución para distintos dispositivos y plataformas, versiones gratuitas y de pago...
- **Simplificación del proceso de testeo:** Los testeos unitarios basados en JUnit y orientados a Android están incluidos como parte del entorno de trabajo. De manera sencilla y rápida pueden desarrollarse y ejecutarse test en Android Studio.

5.2 CONFIGURACIÓN DEL ENTORNO DE TRABAJO EN CLIENTE.

El primer paso es descargar e instalar el entorno desde el sitio oficial. Ya que el desarrollo en Android se lleva a cabo en Java, es necesario disponer de las herramientas de compilación incluidas en el JDK/JRE

El paquete de descarga incluye además de la herramienta de desarrollo:

- **El último SDK Android disponible.** En el proyecto actual, se utilizará como target la API 21, correspondiente a la versión de android 5.0.1
- **Gestor de dispositivos virtuales AVD:** herramienta para manejar emuladores Android. Por diversas razones que posteriormente se desarrollarán, se ha optado por una herramienta alternativa a la ofrecida por Google para la emulación de dispositivos.
- **SDK Manager:** Herramienta para la descarga de paquetes adicionales requeridos.

5.2.1 Configuración de dependencias adicionales del SDK de Android

Se instalarán los siguientes paquetes necesarios para el desarrollo del proyecto:

- **Android Support repository y Support library:** Estos paquetes permiten el uso de las características de las versiones android más modernas en dispositivos basados en versiones antiguas del Sistema operativo.
- **Google Play Services:** paquete necesario para el uso de los servicios de Google integrados nativamente en Android. Entre estos servicios se incluye el acceso a “Google Maps API for Android”, que abarca todas las características relacionadas con los mapas de Google. (Mostrar mapas e información en ellos, por ejemplo).

5.2.2 Configuración del proyecto

Tras completar los datos relacionados con el nombrado de aplicación y paquetes, habrá de seleccionarse el SDK mínimo requerido por los dispositivos que vayan a ejecutar la aplicación. Se han tomado las siguientes decisiones de diseño para el desarrollo del proyecto.

- La API mínima requerida en los dispositivos para instalar la aplicación será 14 (correspondiente a Android 4.0). Mediante dependencias a librerías de soporte se podrá acceder a componentes no disponibles en ésta API.
- La API objetivo utilizada será 21 (correspondiente a Android 5.0)

La razón de estas elecciones es sencilla: Con versiones inferiores el aumento de dispositivos en el mercado capaces de ejecutar la aplicación no compensa la reducción de herramientas de desarrollo disponibles en la API, mientras que con versiones superiores el coste de oportunidad en términos de usuarios no es

aceptable. Además, el uso del paquete support instalado previamente supone una gran aproximación de funcionalidades a las APIs más recientes.

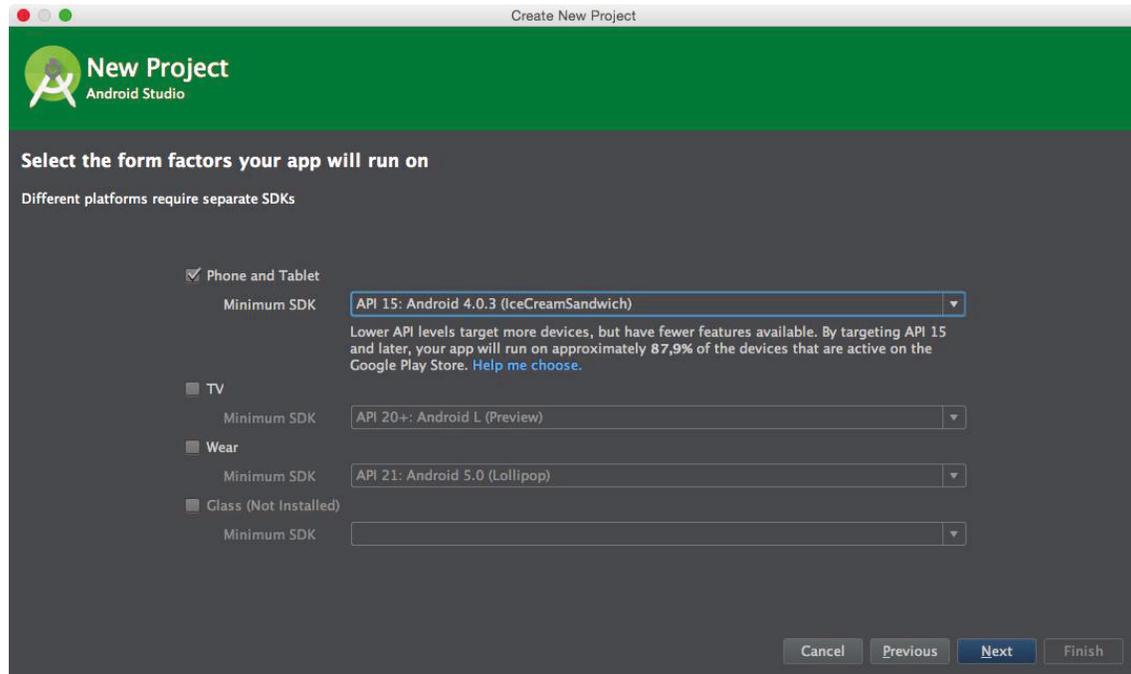
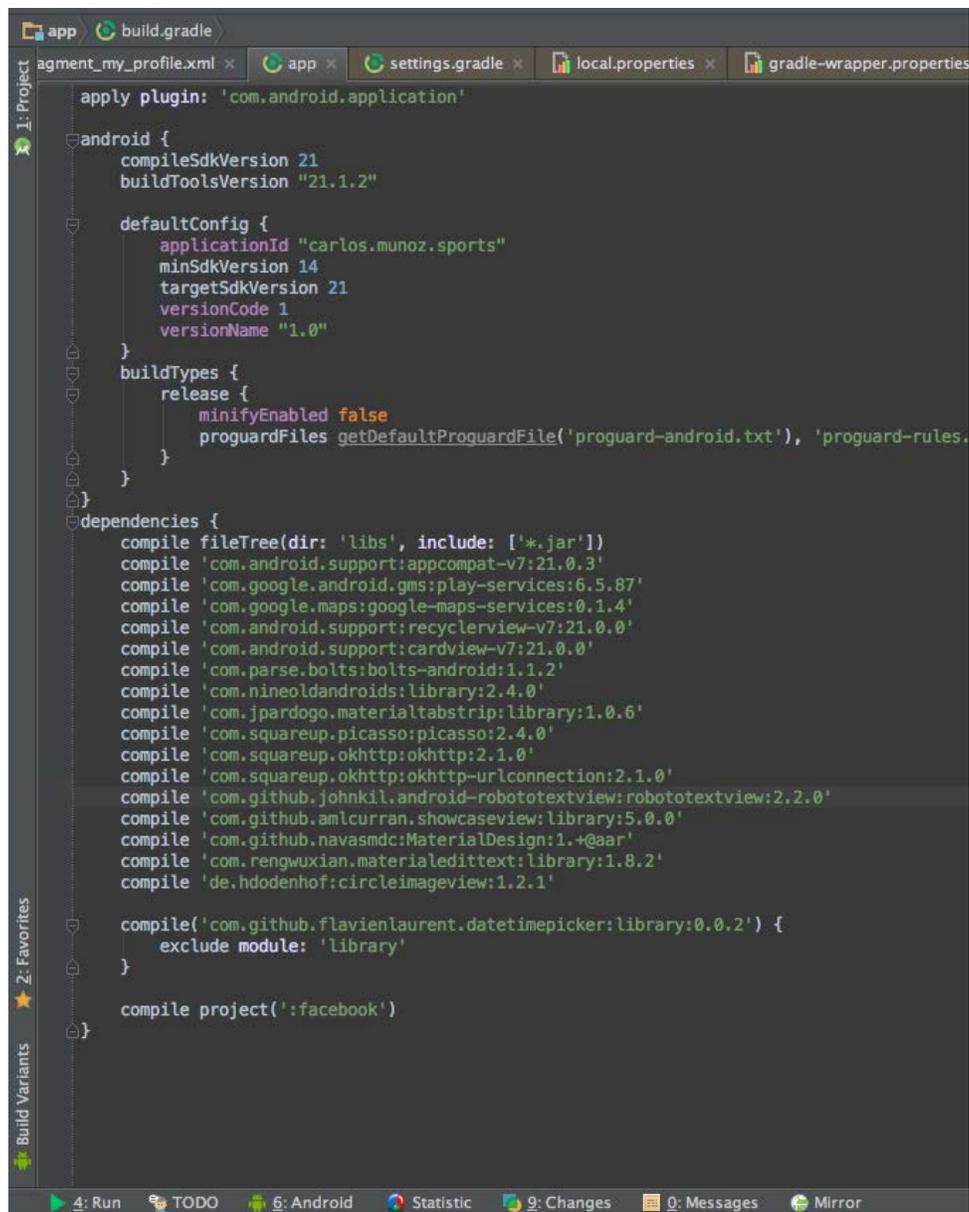


Ilustración 23: Creación de nuevo proyecto en AS

5.2.3 Gradle e Inclusión de dependencias

Como anteriormente se ha comentado, una de las ventajas de Android Studio es que basa la estructuración del proyecto en la herramienta de automatización Gradle. Con ello el proceso de compilación se simplifica enormemente.

La configuración de la herramienta se manejará desde el descriptor **build.gradle**. Es necesario especificar en este archivo información relacionada con el SDK de compilación o versionado, pero este apartado se va a centrar en la inclusión de dependencias.



```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "carlos.munoz.sports"
        minSdkVersion 14
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
    compile 'com.google.android.gms:play-services:6.5.87'
    compile 'com.google.maps.google-maps-services:0.1.4'
    compile 'com.android.support:recyclerview-v7:21.0.0'
    compile 'com.android.support:cardview-v7:21.0.0'
    compile 'com.parse.bolts:bolts-android:1.1.2'
    compile 'com.nineoldandroids:library:2.4.0'
    compile 'com.jpardogo.materialtabstrip:library:1.0.6'
    compile 'com.squareup.picasso:picasso:2.4.0'
    compile 'com.squareup.okhttp:okhttp:2.1.0'
    compile 'com.squareup.okhttp:okhttp-urlconnection:2.1.0'
    compile 'com.github.johnkil.android-robototextview:robototextview:2.2.0'
    compile 'com.github.amlcurran.showcaseview:library:5.0.0'
    compile 'com.github.navasmdc:MaterialDesign:1.+@aar'
    compile 'com.rengwuxian.materialEditText:library:1.8.2'
    compile 'de.hdodenhof:circleimageview:1.2.1'

    compile('com.github.flavienlaurent.datetimepicker:library:0.0.2') {
        exclude module: 'library'
    }

    compile project(':facebook')
```

Ilustración 24: Fichero de compilación Gradle del proyecto

Como puede observarse en la anterior ilustración que referencia la estructura de dependencias existente al finalizar el desarrollo, se han inyectado dependencias mediante 3 vías. A continuación se especificarán las diferencias de cada vía y la funcionalidad de cada dependencia:

Actualmente es la vía más popular de distribución de dependencias. Los desarrolladores crean librerías y módulos y los incluyen en repositorios Maven públicos. El más utilizado es **MavenCentral**.

En el fichero build.gradle del proyecto se especifica a qué repositorios tendrá acceso Gradle a la hora de realizar la inclusión de dependencias, mientras que en el fichero build.gradle del módulo que incluye dependencias se especifica la uri de la librería, provista habitualmente en el sitio web en que se mantiene y desarrolla la dependencia en cuestión. Se especifica en el descriptor mediante la sentencia *compile 'uri del proyecto'*. En siguientes apartados se describen las librerías incluidas mediante este método.

5.2.4 Inclusión de módulos

Definido por la sentencia `compile project(:modulo)`, inyecta en el módulo principal módulos que hacen funciones de librería. En este proyecto:

Facebook: a especificación de esta herramienta no incluye la librería en MavenCentral, con lo que no hay más remedio que incluir el proyecto completo como submódulo y actualizarlo manualmente. Como ventaja de ello puede consultarse y modificarse el código al formar parte de la estructura del proyecto

5.2.5 Inclusión de ficheros .jar

Definido por la primera línea del apartado de dependencias. Incluye en tiempo de compilación todas las librerías que se encuentran en el directorio libs, en este caso:

Parse: la especificación de esta herramienta no incluye la librería en MavenCentral, con lo que no hay más remedio que incluir el jar y actualizarlo manualmente.

5.2.6 Descripción de librerías utilizadas

Librerías de soporte y de servicios: se referencian localmente a las descargas que se comentaron en el apartado Instalación de paquetes adicionales del SDK de Android

RecyclerView y CardView: han de referenciarse adicionalmente aunque pertenecen al paquete de soporte.

- **CardView:** permite mostrar vistas en forma de Card conforme al patrón de diseño Material Design de Android [18].

- **RecyclerView:** Es una nueva forma de mostrar cualquier vista basada en Adapters de manera similar, y que Google pretende establecer como el sucesor de ListViews y GridViews. Se ha optado por su uso en este proyecto por experiencia previa en su uso.

Bolts: Dependencia necesaria para Parse, es una herramienta para la ejecución de tareas en hilos alternativa a la nativa en Android (AsyncTask). Además de su uso por parte de Parse, se ha utilizado para diversas tareas independientes del hilo UI, como es la búsqueda de direcciones

NineOldAndroids: subdependencia de algunas de las librerías utilizadas. Se trata de una herramienta de soporte para utilizar transiciones y animaciones en dispositivos con versiones previas a 3.0. No se hace uso específico en el proyecto de ella.

MaterialTabStrip: De nuevo una librería para adaptar el diseño del proyecto a la especificación Material Design. En este caso se utiliza para estilizar los ViewPagers (las vistas estructuradas en pestañas)

Picasso: Una de las librerías de terceros más populares y útiles existentes en Android. Simplifica al extremo la carga asíncrona de imágenes tanto locales como remotas. Se ha utilizado en todos los casos de carga de imágenes en éste proyecto

OkHttp: Dependencia de Picasso y también utilizada en el proyecto, facilita la ejecución de peticiones HTTP.

ShowCaseView: Librería útil para realizar tutoriales de primer uso de una aplicación, focaliza la atención del usuario en un punto de la pantalla permitiendo describir a modo didáctico el uso de ese foco.

Material Design Library: una de las librerías más utilizadas en el proyecto. Provee de widgets y componentes Android rediseñados al estilo Material Design.

Circle Image View: Como su nombre indica, se trata de un componente para mostrar vistas de imágenes circulares. Se ha utilizado en combinación con Picasso. Las imágenes circulares son uno de los elementos recomendados por la especificación Material Design.

Tras este análisis es fácilmente observable la cantidad de librerías dedicadas a adaptar el diseño a Material Design. Si bien este diseño es una especificación de Google, la librería de soporte provista por la compañía aún es escasa en funcionalidad, por ello el uso de librerías de terceros para adaptarse a su diseño.

5.2.7 Sistema de control de versiones

Se ha optado para controlar el desarrollo de la aplicación móvil por utilizar un **VCS**. El sistema de control elegido es **Git**. Además se ha decidido puesto que la aplicación se va a desarrollar en diferentes computadoras, por almacenar el proyecto en un repositorio git en la nube alojado en **Bitbucket**. El proceso seguido para inicializar el proyecto en el VCS ha sido el siguiente:

- Creación en el portal de Bitbucket de un repositorio, al que se ha denominado “sports”. Como resultado obtenemos vínculos HTTPS y SSH al repositorio.
- Enlace con el proyecto Android inicial creado localmente a través de terminal y con ayuda del comando git.
- Subida del proyecto Android inicial al repositorio.

```
Carlos@SOBREMESA ~
$ cd sports/

Carlos@SOBREMESA ~/sports
$ git init
Initialized empty Git repository in c:/Users/Carlos/sports/.git/

Carlos@SOBREMESA ~/sports (master)
$ git remote add origin https://carlospiles@bitbucket.org/carlospiles/sports.git

Carlos@SOBREMESA ~/sports (master)
$ git push origin master
```

Ilustración 25: inicialización del repositorio Git (1)

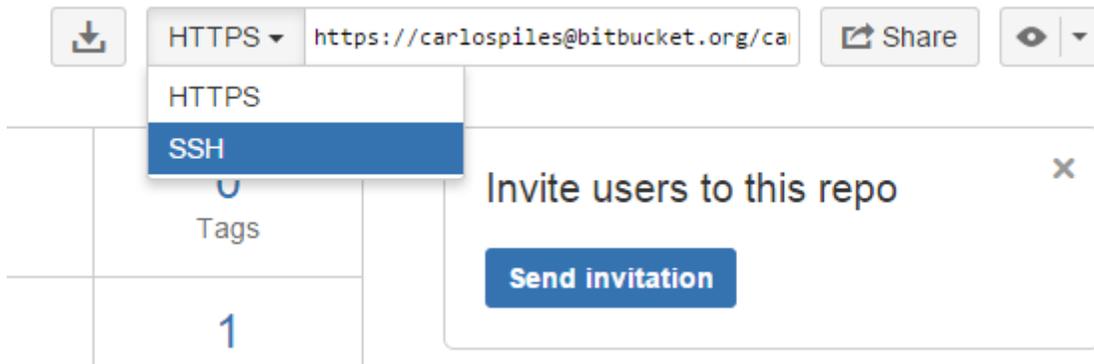


Ilustración 26: inicialización del repositorio Git (2)

Tras realizar esta configuración ya disponemos de un sistema de control de versiones funcional y sincronizado. Para mantener dicha sincronización se han utilizado, entre otros, los siguientes comandos Git:

- **git commit** para hacer efectivos los cambios realizados en el SCV. mediante el parámetro -m se adjunta un mensaje a los cambios realizados
- **git add** para añadir ficheros nuevos al SCV para que éste mantenga el versionado de los mismos.
- **git push** para sincronizar los nuevos cambios realizados localmente en el repositorio
- **git pull** para descargar nuevos cambios existentes en el servidor y disponer de ellos en el proyecto local.

Commits

All branches ▾ Find commits

Author	Commit	Message	Date
carlospiles	f40f3ad	fixed chat name/content, suscribe/unsuscribe on join/leave exercixe	2015-02-04
carlospiles	c0d2720	lots fo changes	2015-01-31
carlospiles	72f943e	receiving messages correctly	2015-01-28
carlospiles	7e58388	created chatService	2015-01-28
carlospiles	7121159	continued profile page	2015-01-24
carlospiles	df05c16	customized infowindows	2015-01-24
carlospiles	aa1cdad	added horizontal user image list, started profile, minor changes	2015-01-24
carlospiles	dca9fde	fixed facebook, style changes	2015-01-23
carlospiles	16c20fe	Login Activity Paged, added Material Design Library	2015-01-03
carlospiles	250da6e	moved to material pagerslidingtabstrip	2014-12-31
carlospiles	c62605b	facebook interaction finished, refactored MVP at HomeActivity	2014-12-30
carlospiles	838edce	added facebook missing files	2014-12-28

Ilustración 27: Histórico de commits en Git

6 PLANIFICACIÓN Y PRESUPUESTO

6.1 PLANIFICACIÓN

Este apartado detalla la planificación estimada para el desarrollo del proyecto. Se comenzará describiendo las distintas macro tareas del proyecto que posteriormente se estudiarán desglosadas a través de un diagrama de Gantt

6.1.1 Análisis

Durante la fase de análisis y como ya se ha analizado a lo largo de este estudio, se definirá el problema a solventar. Se concluirá cual es la funcionalidad del sistema, pero no en las vías que se seguirán para alcanzarla. Al ser un proceso iterativo e incremental, ante cambios en las consecutivas fases de desarrollo será necesario volver a la fase de análisis y adaptarse a estos nuevos cambios.

6.1.2 Diseño

En esta fase y una vez conocido y estudiado el problema a resolver se estudiarán las vías para su resolución. Se mostrarán las alternativas contempladas para su consecución desde el punto de vista tecnológico y se detallará la selección final. En este apartado además se estudia la arquitectura del sistema desde un punto de vista global, así como detallado por componentes. Al ser un proceso iterativo e incremental, ante cambios en las consecutivas fases de desarrollo será necesario volver a la fase de diseño y adaptarse a estos nuevos cambios.

6.1.3 Implementación y Pruebas

En esta fase y una vez seleccionadas las vías y herramientas para la resolución del problema planteado, se procede a su puesta en marcha y ejecución, usando para ello las herramientas tecnológicas previamente elegidas. Paralelamente a esta fase se encuentra la fase de pruebas, de tal manera que la implementación es totalmente dependiente de las mismas. Cada tarea de implementación llevará asociados una serie de tests tanto unitarios como de integración que habrá de superar la funcionalidad a implementar. De esta manera, los procesos de cambio y refactorización futuros contarán con un colchón de seguridad de testing y pruebas que afianzará la fiabilidad del desarrollo.

6.1.4 Gantt

Una vez descritas las diferentes fases del proyecto y su interrelación, en este apartado se detalla cada una de las mismas aplicada al proyecto actual. A través del diagrama de Gantt se muestra visualmente cada una de las fases y sus subtareas en detalle.

En la ilustración siguiente puede observarse una panorámica reducida del diagrama de Gantt mostrando únicamente las fases principales y su duración:



Ilustración 28: Gantt (reducido)

Puede observarse que las fases se solapan unas con otras, lo que se debe a dos razones fundamentales:

- El proyecto se puede subdividir en 3 subproyectos relativamente independientes, el desarrollo del cliente, servidor A y servidor B. Las tareas de diferentes subproyectos pueden solaparse puesto que no existen dependencias
- Dado el proceso iterativo e incremental que sigue la planificación del proceso, existirán cambios en ciertas fases que retroalimenten el proceso de fases anteriores.

En el anexo IV puede observarse el diagrama de Gantt completo con cada una de las tareas de cada subproyecto.

6.2 PRESUPUESTO

El análisis presupuestario del producto y de la empresa se ejecuta en detalle en el Anexo I: Análisis económico financiero.

7 CONCLUSIONES Y TRABAJOS FUTUROS

A lo largo de este apartado se sintetiza el proceso llevado a cabo a lo largo de este proyecto. A partir de ello se revelaran las conclusiones obtenidas a partir de los resultados de dicho proceso y se analizarán los posibles trabajos futuros a realizar.

7.1 TRABAJOS FUTUROS

Aunque como se ha visto, la implementación a realizar cubre todas las necesidades del usuario detectadas en el proceso de análisis y representadas a través de los requisitos, existen ciertas funcionalidades y mejoras que sin duda mejorarán la experiencia de usuario:

- **[Cliente] Almacenamiento local (Cacheado) de datos:** Actualmente se requiere conexión a internet para ejecutar la práctica totalidad de los casos de uso existentes.
- **[Servidor B] Incorporación de notificaciones push y reducción de uso de sockets:** El uso de sockets excesivo tiene un impacto negativo en la batería de los dispositivos en cliente, con lo que es de gran interés utilizar las notificaciones push para notificar al cliente de los nuevos mensajes de chat que recibe en lugar de mantener un socket abierto por conversación.
- **[Servidor A] Sustitución de MBaaS por servidor independiente:** Aunque su funcionamiento es correcto, el precio del escalado de estas plataformas si la aplicación aumenta en gran medida su base de usuarios se eleva enormemente. Además la creación de un servidor independiente programado por el equipo concede gran flexibilidad a la hora de incorporar funcionalidades complejas que un MBaaS no pueda proveer al no encontrarse en su API las herramientas necesarias para hacerlo.
- **[Todos] Mejoras en medidas de seguridad:** En el caso de cliente, hacer un estudio de seguridad que incluya revisión de logs mostrados, configuración de Proguard, cifrado de datos en comunicaciones con ambos servidores y cifrado de datos de almacenamiento local. En el caso de servidores, cifrado de datos de usuarios almacenados localmente.

7.2 CONCLUSIONES:

En este apartado se describen las conclusiones alcanzadas y se estudia el grado de cumplimiento de los objetivos planteados al inicio de este proyecto.

- **Revelar el contexto y situación de las variables de entorno de la empresa:** Se considera que este objetivo se ha cumplido satisfactoriamente. Se han detectado las variables que forman parte del entorno de la empresa, tanto desde el punto de vista tecnológico como del empresarial, analizándose el estado en que se encuentra cada una de ellas.
- **Estudiar alternativas de diseño y obtener una conclusión óptima:** Se han analizado distintas vías para la consecución del proyecto en términos de diseño, detallándose las ventajas e inconvenientes de cada una de ellas. Se ha seleccionado a partir de esta información una arquitectura de diseño eficiente, con lo que se considera este objetivo cumplido.
- **Describir el proceso de construcción y proporcionar un prototipo del cliente:** A lo largo del proyecto se ha detallado la implementación llevada a cabo para el diseño seleccionado. Se ha representado a través de ilustraciones un prototipo del cliente que muestra sus diferentes vistas, así como el funcionamiento de los servidores que lo complementan. Este objetivo se considera completo.
- **Servir como guía para similares desarrollos en el futuro:** Se considera que este objetivo ha sido cumplido, pudiendo servir este documento en fuente de información útil a consultar a la hora de tomar decisiones en futuros proyectos desarrollados por terceros.

8 MARCO LEGAL, FORMA JURÍDICA Y FISCALIDAD.

8.1 FORMA JURÍDICA

Para iniciar nuestra actividad empresarial existen diferentes opciones legales que determinan el tipo de empresa a crear. Para la constitución de esta empresa habrá de optar por una de las opciones disponibles, entre las que se encuentran el registro como sociedad mercantil o como emprendedor individual [19]. Se ha optado por la primera pues es la que mejor se ajustará a la actividad y mejores ratios beneficio / coste aporta.

Una vez seleccionada la forma de sociedad mercantil se plantean diferentes alternativas. A tener en cuenta en la opción final se encuentran las características del negocio a llevar a cabo, los riesgos que se está dispuesto a aceptar y el capital invertido. Con estas variables finalmente nos decantaremos por la forma legal de Sociedad de Responsabilidad Limitada (en adelante S.L.)

Esta forma empresarial requiere un capital mínimo de 3000 €, que se ajusta a los recursos reducidos disponibles en un principio. La otra opción disponible, Sociedad Anónima, requiere un capital mínimo de 60.000€. Por otro lado las obligaciones contraídas con la forma legal S.L protegen los bienes personales de los accionistas. La sociedad jurídica S.L es la forma más utilizada en España, siendo la más adecuada para pequeñas y medianas empresas.

8.2 PROCEDIMIENTO LEGAL

Dado el gran número de procedimientos legales a realizar y como ya se ha comentado en anteriores apartados, se delegarán los procesos legales a una empresa de consultoría externa. Concretamente y por su experiencia con miembros del organigrama de la empresa se optará por Asesoría Aravaca, empresa afincada en Madrid que dispone de un plan de gestión de todo el procedimiento legal fiscal a llevar a cabo en la creación de una empresa. El procedimiento englobará las siguientes tareas:

- **Registro de la empresa en el Registro mercantil central:** podrá llevarse a cabo vía telemática. Una vez llevado a cabo este proceso la empresa gozará de Denominación Social.
- **Consecución de la escritura pública,** firmada ante notario: La escritura pública habrá de contener:
 - Denominación social de la empresa (derivada del registro mercantil)

- **Objetivos de la sociedad:**
 - Actividades a realizar en la empresa
 - Fecha de cierre del año fiscal de la empresa
 - Localización física del negocio
 - Patrimonio neto de los propietarios
 - Plan de organización y estructura del negocio
- **Registro en la Agencia tributaria** (ver apartado “Fiscalidad”)
- **Registro de la empresa en el Registro mercantil provincial:** Como ya se ha comentado, la sede legal de la empresa se establecerá en Madrid, de tal manera que será necesario registrar la empresa en el Registro Mercantil de Madrid.

Además de este procedimiento principal han de realizarse otros procedimientos paralelos que permitan el desarrollo de nuestra actividad empresarial en comunión con el marco legal y fiscal vigente:

- **Registro en la Tesorería general de la Seguridad Social**
- Solicitud y **adquisición de licencias del Ayuntamiento de Madrid**
- **Registro de datos personales de los usuarios en la Agencia Española de protección de datos**, acogiéndose a la normativa establecida en la Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal [20]

8.3 FISCALIDAD

El establecimiento de una SL entraña una serie de obligaciones fiscales a tener en cuenta por la empresa y los encargados de su iniciación:

- **Disposición de un capital mínimo de 3000€** para la creación de una SL, referido en el Real Decreto Legislativo 1/2010, de 2 de julio, por el que se aprueba el texto refundido de la Ley de Sociedades de Capital.
- Complimentación del modelo 600 y sus apéndices y presentación para su certificación por la Consejería de Hacienda. Esta certificación es el paso previo al obligado **registro de la empresa en la Agencia Tributaria**.
- Complimentación de los modelos 036 y 037 para la **inscripción de la empresa en la Declaración Censal**, que identifica los términos fiscales de la empresa.

- Adquisición del **Código de Identificación Fiscal**.
- Cumplimentación del modelo 200, que supone el **registro y pago del impuesto sobre Actividades económicas (IAE)**, que habrá de pagarse además anualmente a partir del inicio de la actividad empresarial. De acuerdo con el Real Decreto-Ley 4/2013, de 22 de febrero 2012, de medidas de apoyo al emprendedor y de estímulo del crecimiento y de la creación de empleo,
 - Si la empresa obtiene beneficios menores a 300.000 € el impuesto alcanzará el 15% de los beneficios obtenidos en el año.
 - Si la empresa obtiene beneficios mayores a 300.000 € el impuesto alcanzará el 20% de los beneficios obtenidos en el año.

En cuanto al registro en la seguridad social, si la empresa emplea a un trabajador menor de 30 años durante al menos 18 meses estará exento de la tasa de Seguridad social, de acuerdo con lo definido en el BOE.

Por el hecho de emplear trabajadores, la empresa se encuentra sujeta a la Ley 31/1995, de 8 de noviembre, de prevención de Riesgos Laborales (Boletín Oficial del Estado, 1995).

Finalmente, En cuanto a las tasas una vez iniciada la actividad empresarial, habrá de tenerse en cuenta:

- El pago del IVA
- La declaración anual de Operaciones con terceros
- El impuesto de Transmisiones Patrimoniales

8.4 CUADRO RESUMEN

A continuación una tabla resumen incluye los principales deberes y obligaciones fiscales a tener en cuenta:

RESUMEN DE OBLIGACIONES FISCALES

AUTÓNOMO	SOCIEDADES CIVILES Y COMUNIDADES DE BIENES	SOCIEDADES MERCANTILES
ANTES DE INICIAR LA ACTIVIDAD		
DC	DC	DC
IAE	IAE	IAE
	ITP	ITP
UNA VEZ INICIADA LA ACTIVIDAD		
IRPF	IRPF	IS
IVA	IVA	IVA
DAOT	DAOT	DAOT
FINALIZADA LA ACTIVIDAD – BAJA		
DC	DC	DC
	IAE	IAE

DC: Declaración Censal | IAE: Impuesto de Actividades Económicas | ITP: Impuesto de Transmisiones Patrimoniales | IRPF: Impuesto sobre la Renta de las Personas Físicas | IS: Impuesto de Sociedades | IVA: Impuesto sobre el Valor Añadido | DAOT: Declaración anual de Operaciones con terceros

Tabla 62: Resumen de obligaciones fiscales

8.5 PATENTES

Es fundamental para la protección de las ideas y marcas asociadas a nuestros productos y con el fin de evitar posibles copias, plagios y demás inconvenientes legales el registro de WeSweat en la Oficina Española de Patentes y Marcas. El coste final de esta inscripción es de 120€.

8.6 PROTECCIÓN DE DATOS

En cuanto lo que refiere a los datos de los usuarios, dado que se tratará con datos personales y como se ha mencionado anteriormente la LOPD obliga a la empresa a registrar su actividad en la Agencia Española de Protección de Datos. Es de recibo mencionar que este proyecto toma como fuente de datos previa aceptación de permisos por parte del usuario datos personales de Facebook, de los cuales almacena nombre y apellidos. Además, las conversaciones que tienen lugar en la aplicación se almacenan temporalmente en el servidor con el fin de proveerlas a dispositivos sin conexión a internet cuando la recuperen. Estos datos gozan de alta sensibilidad en términos de privacidad y por tanto obligan a esta empresa a registrarse en la Agencia Española de protección de datos.

9 GLOSARIO

CRUD: Acrónimo que hace referencia a las operaciones Crear, Obtener, Actualizar y Borrar (Create, Read, Update and Delete), funciones básicas habitualmente encontradas en la capa de persistencia de un sistema software.

Notificaciones Push: Notificaciones basadas en tecnología Push, protocolo de comunicaciones en que el servidor es el responsable de iniciar el intercambio de mensajes.

Sistema de control de versiones (VCS): Sistemas que facilitan la administración de las distintas versiones de un producto desarrollado, así como las posibles especializaciones realizadas (por ejemplo, para algún cliente específico)

Platform as a service (PaaS): Encapsulación de herramientas que abstrae al programador de las concreciones de la plataforma. En este caso la única preocupación es la construcción de la aplicación, ya que la infraestructura viene provista por la plataforma.

Backend vs Frontend: el front-end es la parte del software que interactúa con el o los usuarios y el backend es la parte que procesa la entrada desde el front-end.

Mobile Backend as a Service: Encapsulación de herramientas que abstrae al programador del desarrollo del backend de comunicaciones de una aplicación móvil.

SDK: conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto.

10 BIBLIOGRAFÍA

- [1] ATUS, “Sports and Exercise: BLS Spotlight on Statistics,” vol. 2015. 2009.
- [2] C. y D. Ministerio de Educación, *Anuario de estadísticas deportivas*. Spain: Gobierno de España, 2014.
- [3] Statista, “Number of free and paid mobile app store downloads worldwide from 2011 to 2017,” vol. 2015. 2015.
- [4] Qualcomm, “Behind the sixth sense of smartphones: the Snapdragon processor sensor engine,” vol. 2015. Qualcomm, 2014.
- [5] P. R. Center, “Cell Phone and Smartphone Ownership Demographics,” vol. 2015. Pew Research Center’s Internet & American Life Project, 2014.
- [6] M. Wilcox, “Backend-as-a-Service Shootout,” vol. 2015. Developer Economics, 2013.
- [7] IDC, “IDC: Smartphone OS Market Share,” vol. 2015. www.idc.com, 2015.
- [8] PhoneArena, “Android’s Google Play beats App Store with over 1 million apps, now officially largest,” vol. 2015. Phone Arena, 2013.
- [9] S. P. John, “Android Architecture,” vol. 2015. EazyTutz, 2015.
- [10] T. Louis, “How Much Do Average Apps Make?,” *Forbes*, Forbes, 20-May-2015.
- [11] A. Stellman and J. Greene, *Applied Software Project Management*. United States: O’Reilly Media, Inc, USA, 2005.
- [12] P. V. Gómez, “Effective Android UI,” vol. 2015. GitHub, 2013.
- [13] I. Square, “Dagger,” vol. 2015. 2013.
- [14] S. M, “A useful stack on android #1, architecture,” vol. 2015. 2014.
- [15] I. Square, “Otto, An enhanced event bus with emphasis on Android support,” vol. 2015. 2013.
- [16] Google, “Material design - Google design guidelines,” vol. 2015. Google design guidelines, 2014.
- [17] W. J. Francis, “Android Studio eclipses Eclipse on Google’s developer tools page,” vol. 2015. TechRepublic, 2014.

- [18] Google, "Cards - Components - Google design guidelines," vol. 2015. Google design guidelines, 2014.
- [19] D. DGIPYME, "Financiación para emprendedores y PYME," vol. 2015. 2015.
- [20] Jefatura del Estado, "Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal." 1999.
- [21] Parse, "Parse - Pricing Plans," vol. 2015. 2015.
- [22] Google, "Get Started with Publishing | Android Developers," vol. 2015. 2015.
- [23] A. I. Apple Inc., "iOS Developer Program - Apple Developer," vol. 2015. 2015.
- [24] S. A. Ross, R. W. Westerfield, and J. Jaffe, *Corporate Finance with S&P card*, 9th ed. New York: McGraw-Hill Higher Education, 2010.
- [25] M. de Industria, "Emprendedores | Financiación | ENISA. Impulso financiero a la pyme," vol. 2015. 2015.
- [26] A. Damodaran, "Betas by Sector, 2015," vol. 2015. 2015.
- [27] F. A. Name, "Mobile Application Life Cycle Characterization via Apple App Store Rank," no. 2009, 2014.

11 ANEXO I: ANÁLISIS ECONÓMICO - FINANCIERO

En los sucesivos apartados se resumirá el plan económico y financiero de la empresa. Se comenzará por detallar el modelo de ingresos estimados, costes e inversión. A continuación se analizarán las pérdidas y ganancias desde un punto de vista anual y se estimará el flujo de caja de la empresa. Después de ello seremos capaces de llevar a cabo un estudio de valoración de la empresa y de esbozar un balance contable, así como de analizar indicadores y valores financieros de relevancia para la empresa.

11.1 INGRESOS, COSTES E INVERSIÓN

11.1.1 Ingresos

Los beneficios de nuestra empresa procederán de dos vías, en función del producto ofertado:

- **Producto 1: WeSweat Standard:** se obtendrán ingresos económicos a través del modelo de publicidad in-app a través de las plataformas de *advertising* ya comentadas en este plan de negocio.
- **Producto 2: WeSweat Pro:** en lo que respecta a la versión Pro, los ingresos provendrán de los modelos de suscripción disponibles.

Estos dos productos son los únicos que tendremos en cuenta en el modelo, si bien se pretende ofertar nuevos productos en un futuro⁴

Conocidas las fuentes de ingreso es necesario modelizar el número de usuarios y su patrón de crecimiento y decrecimiento. Es de vital importancia para la exactitud del modelo tener en cuenta el ciclo de vida medio de un producto de este tipo. Conociendo el ciclo de vida medio para una aplicación categorizada como “deportes” (VER ANEXO III), es posible esbozar una función por partes que simule el crecimiento y madurez de nuestro producto.

⁴ Para simplificar se han unificado los subproductos disponibles en las 3 plataformas (Android, IOS, Web) en uno solo. En adelante todos los valores establecidos serán valores promedio de las plataformas.

Si se diera un crecimiento constante en un modelo idealizado, una función que describiese dicho crecimiento sería la siguiente:

$$f(t_n) = f(t_{n-1}) * (k + 1)$$

- $f(t_n)$ = número de descargas de la aplicación en el periodo (mes) n.
- $f(t_{n-1})$ número de descargas de la aplicación en el periodo (mes) anterior a n.
- k : constante de crecimiento

Dado este modelo idealizado, el número de descargas de nuestro producto sería constantemente creciente, y por tanto no aplicable al mundo real. Por tanto, se pretenderá predecir y modelar la variación del factor de crecimiento en consonancia con el modelo de ciclo de vida analizado en el ANEXO III:

$$k(n) = \begin{cases} k_{ini}, & n < inf1 \\ k_{cre}, & inf1 < n < inf2 \\ k_{n-1} * (1 - d), & n > inf2 \end{cases}$$

- Donde
 - k_{ini} : factor de crecimiento constante en la fase de introducción de la aplicación
 - k_{cre} : factor de crecimiento constante en la fase de crecimiento de la aplicación. Será necesariamente mayor que k_{ini}
 - $inf1$: punto de inflexión 1: nacimiento comercial de la aplicación
 - $inf2$: punto de inflexión 2: auge de la aplicación
 - d : factor de reducción del crecimiento en la fase de madurez de la aplicación

Una vez determinada la evolución prevista del crecimiento y para su correcta relación con los ingresos es necesario establecer unos valores iniciales a aplicar a la función anterior.

WeSweat Standard	
Número inicial de usuarios	500
Precio inicial	0,12 €
Tasa de crecimiento (fase inicio)	10%

Tasa de crecimiento (fase crecimiento)	35%
Tasa de decrecimiento	22%
Punto de inflexión 1	2
Punto de inflexión 2	17

Ilustración 29: Valores iniciales WeSweat Standard

WeSweat Pro	
Número inicial de usuarios	35
Precio inicial	4,24 €
ratio usuarios standard / pro	7%

Ilustración 30: Valores iniciales WeSweat Pro

De estos datos es necesario aclarar:

- La lógica dicta que la versión gratuita no tiene precio, si bien se ha tomado como precio los ingresos medios por publicidad relativos al número de descargas, dato del que se deriva una aproximación del valor ingresos por descarga [10].
- Para establecer el precio de la versión de pago se ha utilizado una media ponderada de los precios de los 3 planes de pago disponibles.
- El número de usuarios de la versión de pago sigue indirectamente la expresión de crecimiento descrita, si bien se calcula como un ratio en función del número de usuarios gratuitos. Se ha elegido un ratio de 7% [3], lo que significa que por cada 100 descargas gratuitas se prevén 7 adquisiciones del plan de pago

A continuación se muestra gráficamente la evolución estimada del número de descargas/adquisiciones de ambos productos durante los primeros cinco años⁵:

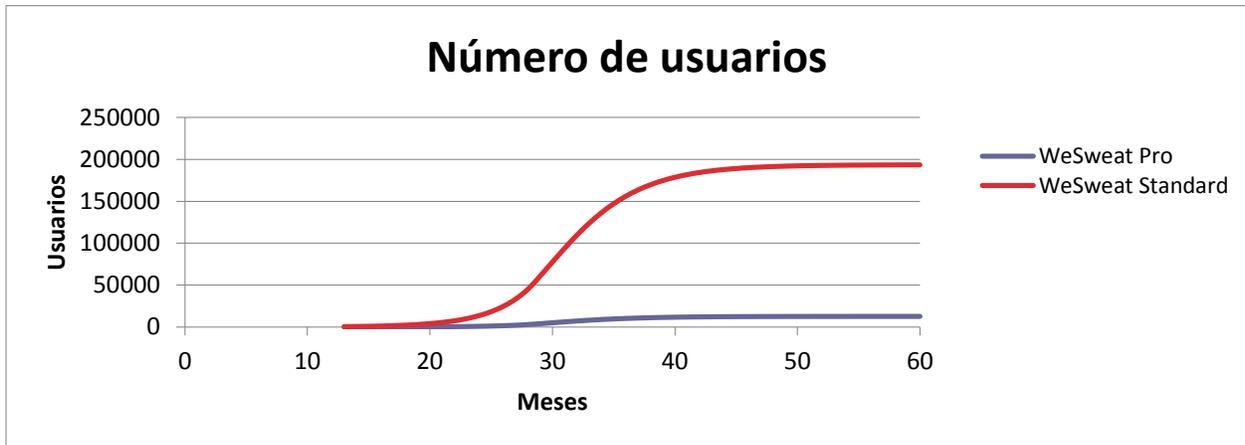


Ilustración 31: Evolución del número de usuarios

Se puede apreciar en el gráfico como el crecimiento se dispara o estanca en función de la etapa de vida en que se encuentra el producto. En la fase de introducción se produce un crecimiento lento causado por el desconocimiento propio de las primeras fases de un producto; En el momento que se alcanza el primer punto de inflexión y se alcanza el nacimiento comercial de nuestros productos, la aplicación comienza a llamar la atención de un número mayor de usuarios y a ser conocida, lo que se revela en el aumento exponencial del factor de crecimiento. Finalmente y después de alcanzar el segundo punto de inflexión correspondiente al alcance del auge comercial (momento en que la aplicación alcanza su máximo número de descargas en todo el ciclo de vida), el crecimiento comienza a reducirse paulatinamente. Es de recibo destacar el gran gap entre las adquisiciones del producto gratuito y el de pago.

Este patrón es muy común en aplicaciones móviles, como ya se ha analizado y concluido en este estudio.

⁵ ambos productos se lanzan simultáneamente en enero de 2016.

En el gráfico a continuación se revelan los ingresos de la empresa durante el mismo periodo, los cinco primeros años de los productos. Se muestra además el ingreso total recaudado por la gama de productos disponible.

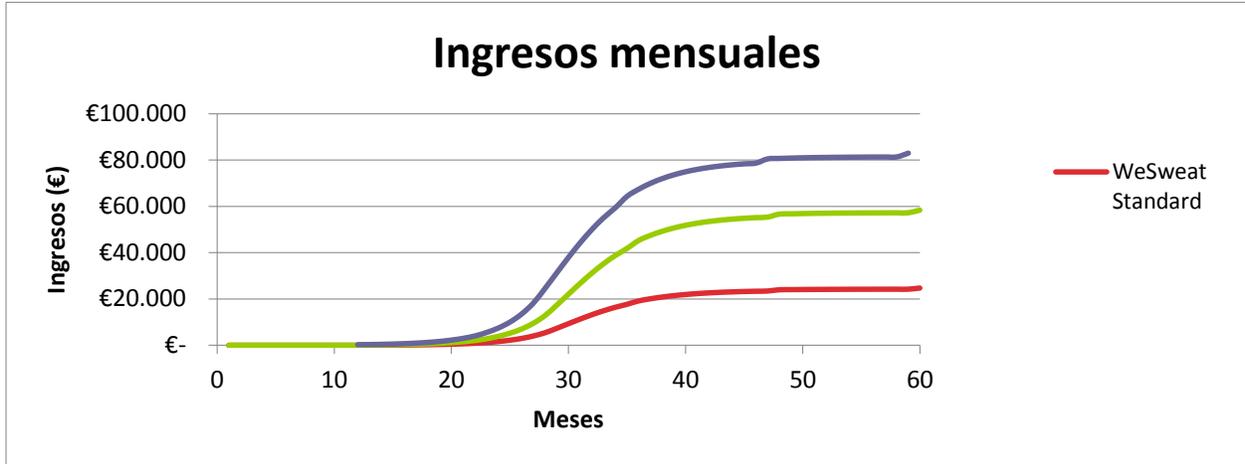


Ilustración 32: Ingresos mensuales

Podemos observar a continuación los datos recogidos en las tablas anteriores relativos a ingresos y descargas en detalle:

	Año 0	Año 1	Año 2	Año 3	Año 4
WeSweat Pro	- €	14.260,38 €	292.562,80 €	635.171,07 €	685.484,76 €
WeSwat Standard	- €	6.040,62 €	123.928,13 €	269.055,26 €	290.367,89 €
Total Ingresos	- €	- €	- €	- €	- €

Tabla 63: Ingresos anuales

11.1.2 Costes

Para el cálculo de costes se han establecido las siguientes categorías:

- **Costes de personal:** engloba costes relativos a salarios de los trabajadores de la empresa que no tienen que ver con el desarrollo de los productos. Esto es, salarios de directivos, trabajadores del departamento comercial, salarios relacionados con soporte, etc. De acuerdo con el plan de incorporaciones descrito en el apartado de recursos humanos, la plantilla irá creciendo proporcionalmente al crecimiento de la empresa, lo que se tendrá en cuenta en la contabilidad de este tipo de costes.
- **Costes de oficina:** Partida que contiene todos los costes relativos al inmueble en que se desarrolla la actividad empresarial. La partida “Alquiler” englobará costes de alquiler, luz y agua, puesto que al alojarse la empresa en un vivero buena parte de los costes habituales de este tipo se encuentran incluidos en el alquiler. No estarían incluidos teléfono e internet en el coste del alquiler.
- **Costes de marketing:** Engloba todos los costes comentados en el plan de marketing. La mayor parte de esta partida es relativa a los costes de promoción, entre los que se encontrarán desglosados:
 - **Publicidad:** Costes relativos a promoción a través de anuncios impresos, anuncios web o en otras aplicaciones móviles, redes de anuncios, *native ads*...
 - **Marketing Directo:** Costes relativos a la creación distribución y envío de comunicaciones directas con el cliente. Destacarán correos electrónicos, así como folletos y otros medios en papel.
 - **Relaciones Públicas:** Costes relativos a creación y patrocinio de eventos.
 - **Promoción de ventas:** Costes relativos a promociones y descuentos.
- **Costes de infraestructura:** Abarca todos los costes de ventas no salariales. En concreto
 - **Hosting y Servidores:** Costes relativos a servicios en la nube de almacenamiento y gestión de datos. Concretamente los datos están basados en el plan de pago de Parse.com [21]
 - **Licencias:** Coste de licencias requeridas para el desarrollo de los productos. Concretamente engloba las licencias de desarrollo en IOS y Android [22] [23].
 - **Costes de soporte:** Salarios del personal de soporte y atención al cliente.
- **Costes de gestoría:** Englobará todos los costes de gestión fiscal y financiera, que como ya se ha comentado serán externalizados a una asesoría. Se encargarán de los procesos contables, financieros relativos tanto al arranque de la empresa y sus productos como de los trámites que surjan a lo largo de la vida de la compañía.

11.1.3 Inversión

La inversión de la empresa se compone de 3 elementos principales:

- **Equipos informáticos:** ordenadores en los empleados de los departamentos de desarrollo y comercial realizarán su labor
- **Smartphone / tabletas:** dispositivos móviles en que se testarán los productos desarrollados
- **Salario de desarrolladores:** Salario de los desarrolladores internos, cuyo trabajo se refiere a la realización de la app y sus mejoras⁶.

Durante el periodo que se analiza en este modelo la empresa adquirirá nuevos equipos informáticos, smartphones y tabletas conforme se deterioren o queden anticuados.

Para cada elemento se amortizará linealmente con un periodo distinto. El cálculo de la amortización mensual es sencillo: Dados los precios de cada elemento y su periodo de amortización (en número de meses), una simple división resultará en lo amortizado cada mes:

	Inversión/equipo	periodo de amortización	amortización mensual
Equipos informáticos	650,00 €	36 meses	18,06 €
Smartphone / Tablets	270,00 €	36 meses	7,50 €
Salario desarrolladores	2.400,00 €	60 meses	40,00 €

Tabla 64: amortización mensual

Con ello puede observarse la evolución estimada de inversión y amortización a lo largo del periodo estudiado en este modelo:

Año 0	Año 1	Año 2	Año 3	Año 4
-------	-------	-------	-------	-------

⁶ Los desarrolladores internos, si su trabajo se refiere a la realización de la app y sus mejoras, su salario puede ser capitalizado (en el caso de que se trate de un gasto en desarrollo con viabilidad comercial), catalogándose entonces en el activo fijo tal como determina el plan general contable.

INVERSIÓN	Equipos informáticos	3.900,00 €	1.300,00 €	1.300,00 €	1.300,00 €	3.900,00 €
	Smartphone / Tablets	1.080,00 €	1.080,00 €	1.080,00 €	1.350,00 €	1.890,00 €
	Salario desarrolladores	86.400,0 0 €	86.400,0 0 €	86.400,0 0 €	172.800,0 0 €	259.200,0 0 €
AMORTIZACIÓN						
	Equipos informáticos	1.300,00 €	1.625,00 €	2.058,33 €	1.300,00 €	1.950,00 €
	Smartphone / Tablets	1.300,00 €	1.625,00 €	2.058,33 €	1.300,00 €	1.950,00 €
	Salario desarrolladores	780,00 €	975,00 €	1.235,00 €	1.560,00 €	2.145,00 €

Tabla 65: Inversión y amortización

La mayor parte de la inversión tiene que ver con los salarios de los trabajadores que forman parte del desarrollo del producto. Recordemos que al no haber materia prima y ser el coste de ventas por tanto reducido en comparación con otras empresas, los empleados son nuestra fuerza principal.

11.2 PÉRDIDAS Y GANANCIAS

A lo largo de esta sección se analizarán las pérdidas y ganancias de la empresa.

En primer lugar se dispone de los **ingresos** estimados con carácter anual, que se derivan del apartado Ingresos descrito anteriormente. Para obtener el **margen bruto** de la empresa se sustrae el coste de ventas de la empresa, que como ya se ha comentado, abarcan los siguientes aspectos:

- **Hosting y Servidores:** Costes relativos a servicios en la nube de almacenamiento y gestión de datos. Concretamente los datos están basados en el plan de pago de la plataforma Parse.com
- **Licencias:** Coste de licencias requeridas para el desarrollo de los productos. Concretamente engloba las licencias de desarrollo en IOS y Android.
- **Costes de soporte:** Salarios del personal de soporte y atención al cliente.

Recordemos que el margen bruto tiene en cuenta únicamente los costes relacionados con la producción. Para obtener los beneficios finales será necesario contabilizar el resto de **costes** a realizar. Como se comentó en la sección de costes, se incluirá:

- **Costes de personal:** engloba todos los costes relativos a salarios de los trabajadores de la empresa que no tienen que ver con el desarrollo de los productos. Esto es, salarios no catalogados como coste de ventas: directivos, trabajadores del departamento comercial, salarios relacionados con soporte, etc. De acuerdo con el plan de incorporaciones descrito en el apartado de recursos humanos, la plantilla irá creciendo proporcionalmente al crecimiento de la empresa, lo que se tendrá en cuenta en la contabilidad de este tipo de costes.
- **Costes de oficina:** Partida que contiene todos los costes relativos al inmueble en que se desarrolla la actividad empresarial. La partida "Alquiler" englobará costes de alquiler, luz y agua, puesto que al alojarse la empresa en un vivero buena parte de los costes habituales de este tipo se encuentran incluidos en el alquiler. No estarían incluidos teléfono e internet en el coste del alquiler.
- **Costes de marketing:** gasto en publicidad, marketing directo, relaciones públicas y promoción de ventas.
- **Costes de gestoría,** externalizados a una asesoría independiente.
- **Costes de personal** relacionado con cuestiones que no tengan que ver directamente con el desarrollo del producto.

De la sustracción de estos costes obtendremos el EBITDA (Earnings Before Interest, Taxes, Depreciation and Amortization, es decir, beneficio antes de intereses, impuestos, depreciación y amortización).

Por ultimo habrán de realizarse dos operaciones finales:

- Para obtener el **BAI** (Beneficio Antes de Impuestos) ha de sustraerse los **intereses de la deuda**.
- Para obtenerse el **BN** (Beneficio Neto) ha de sustraerse **los impuestos (25%)**

Con todo ello, en la siguiente tabla podrá verse un resumen de la cuenta de resultados a nivel anual en que se desarrollan todos estos cálculos.

	Año 0	Año 1	Año 2	Año 3	Año 4
CUENTA DE RESULTADOS					
Cuenta de Resultados	Total AÑO 0	Total AÑO 1	Total AÑO 2	Total AÑO 3	Total AÑO 4
INGRESOS	0,00	20.301	416.491	904.226	975.853
COSTE DE VENTAS	124,00	38.499,00	54.140,61	72.816,14	78.988,76
Servidores y Hosting	0	0	11.802	26.253	27.779
Licencias	124	99	99	99	99
Personal de soporte	0	12.000	13.200	14.520	15.972
Key Account	0	26.400	29.040	31.944	35.138
MARGEN BRUTO	-124	-18.198	362.350	831.410	896.864
		-90%	87%	92%	92%
TOTAL GASTOS	8.435	53.904	100.873	325.418	350.089
RRHH no de producción	0	30.000	33.000	182.700	200.970
Director general	0	0	0	38.400	42.240
Director departamento Android	0	0	0	36.000	39.600
Director departamento IOS	0	0	0	36.000	39.600
Director departamento Web	0	0	0	36.000	39.600

Director comercial	0	30.000	33.000	36.300	39.930
Gastos Generales	8.435	23.904	67.873	142.718	149.119
Publicidad	0	5.436	5.436	10.872	10.872
Marketing Directo	0	588	588	1.068	1.068
Promoción de ventas	0	1.899	38.289	81.602	86.346
Relaciones públicas	0	6.000	12.000	36.000	36.000
Total Marketing	0	13.923	56.313	129.542	134.286
Alquiler	3.275	3.603	3.963	4.359	4.795
Teléfono e Internet	360	378	397	417	438
Total Oficina	3.635	3.981	4.360	4.776	5.233
Total Gestoría	4.800	6.000	7.200	8.400	9.600
EBITDA	-8.559	-72.102	261.478	505.992	546.775
		-355%	63%	56%	56%
Resultado financiero	0	3.523	2.818	2.114	1.409
Amortización inmovilizado	0	1.300	1.625	2.058	1.300
BAI	-8.559	-76.924	257.035	501.820	544.066
Acumulado 3 años	-8.559	-85.483	171.551	681.930	1.302.921
tipo impositivo	25%	25%	25%	25%	25%

Impuestos (25%)	0	0	42.888	125.455	136.016
BENEFICIO neto	-8.559	-76.924	214.147	376.365	408.049

Tabla 66: Cuenta de resultados

Finalmente podemos observar en la gráfica siguiente como se comportarán según la estimación los diferentes indicadores referentes a beneficios. El conocimiento de este comportamiento es esencial para comprender el siguiente apartado, el estudio de los flujos de caja.

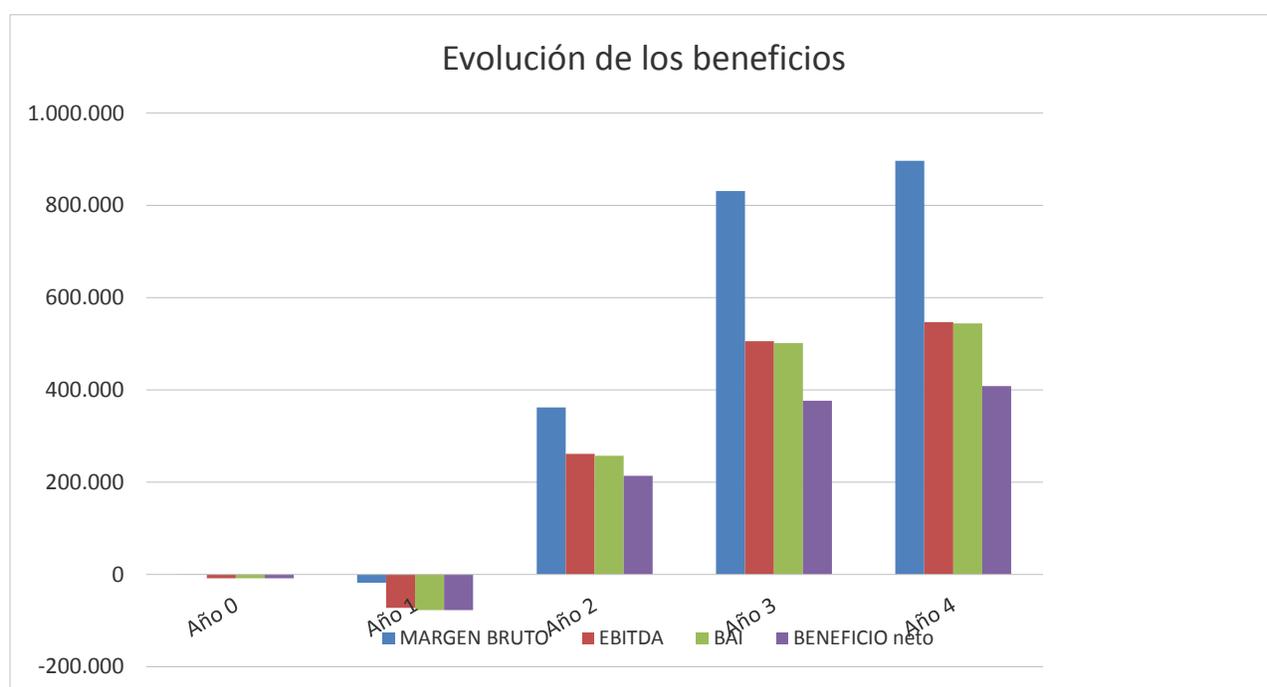


Ilustración 33: Evolución de beneficios

11.3 FLUJOS DE CAJA

En el anterior apartado se muestra una tabla referente a la cuenta de resultados estimados de la empresa. En este punto se analizará el flujo de caja, entendido como entradas y salidas de caja y efectivo a lo largo de un periodo. Es uno de los indicadores más relevantes de la liquidez de la empresa [24], puesto que se

trata de la acumulación neta de activos líquidos, o lo que es lo mismo, dinero “real”. El flujo de caja se calcula de la siguiente manera:

$$\text{Cash Flow} = \text{BN} - \Delta\text{NOF} - \Delta\text{AFN}$$

- *BN*: Beneficio Neto
- ΔNOF : variación de necesidades operativas de fondos, entendidas como los fondos que se necesitan para financiar satisfactoriamente las actividades de la empresa.
- ΔAFN : Incremento de Activo fijo neto: diferencias entre inversión y amortización acumuladas para el periodo dado.

Para calcular las variaciones de NOF, utilizaremos la siguiente expresión:

$$\text{NOF} = \text{Caja mínima} + \text{Clientes} + \text{Stock} - \text{Proveedores}$$

- Caja mínima: Mínimo a mantener en caja para poder operar.
- Stock: En nuestro caso 0, ya que no se cuenta con stock
- Clientes: utilizaremos la siguiente expresión: $\text{Clientes} = \frac{(\text{días de clientes} * \text{volumen de negocio anual})}{365}$
- Proveedores: utilizaremos la siguiente expresión: $\text{Proveedores} = \frac{(\text{días de proveedores} * \text{compras})}{365}$

Se entenderá días de clientes como el número de días transcurridos entre la adquisición del producto por parte del cliente y la recepción del dinero por parte de la empresa. Interesa su reducción en tanto que no se puede disponer de ese dinero hasta efectuarse el cobro. En el caso de los días de proveedores sucede lo contrario, Su maximización en la medida de lo posible beneficiara a la empresa.

En la siguiente tabla se desarrolla el cálculo del cash flow del proyecto a lo largo de 5 periodos anuales y siguiendo el modelo descrito:

Flujo de Caja	Total AÑO 0	Total AÑO 1	Total AÑO 2	Total AÑO 3	Total AÑO 4
(+)Beneficio Neto	-8.559	-76.924	214.147	376.365	408.049

(-)Variación de NOF	9.995	92	31.921	39.320	5.633
caja mínima	10.000	10.000	10.000	10.000	10.000
clientes	0	1.669	34.232	74.320	80.207
días de clientes	30	30	30	30	30
stock	0	0	0	0	0
días de stock	0	0	0	0	0
proveedores	5	1.582	2.225	2.992	3.246
compras	124	38.499	54.141	72.816	78.989
días de proveedores	15	15	15	15	15
Necesidades Operativas de Fondos (NOF)	9.995	10.086	42.007	81.328	86.961
(+)Variación de IVA	-1.034,99	-631	3.455	1.635	-505
IVA en caja	-1.034,99	-1.666,12	1.788,49	3.423,15	2.918,21
días de iva	18,00	18,00	18,00	18,00	18,00
Saldo IVA	-20.987	-33.785	36.267	69.414	59.175
IVA soportado	20.987,20	38.048,40	51.196,56	120.473,72	145.754,21
IVA repercutido	0,00	4.263,21	87.463,09	189.887,53	204.929,05
(-)Variación activo fijo	88.000	84.555	83.428	171.290	258.945
inversión	91.380	88.780	88.780	175.450	264.990
inversión acumulada	91.380	180.160	268.940	444.390	709.380
amortización	3.380	4.225	5.352	4.160	6.045
amortización acumulada	3.380	7.605	12.957	17.117	23.162
Activo fijo neto	88.000	172.555	255.983	427.273	686.218

CF OPERATIVO O DE PROYECTO	-107.589	-162.202	102.252	167.389	142.966
CF acumulado	-107.589	-269.791	-167.539	-149	142.817

Tabla 67: Flujo de caja del proyecto

Conocidas las NOF y las inversiones a realizar durante estos periodos, es necesario determinar una estructura de Equity y Deuda que permitan financiarlas. Los accionistas cubrirán los beneficios netos negativos de los primeros periodos y la devolución de la deuda no cubierta por el beneficio. En cuanto a la deuda, se ha optado por acogerse al plan de financiación Emprendedores [25]. Han de cumplirse los siguientes requisitos para su consecución:

- Ser PYME conforme a la definición de la Unión Europea, tener personalidad jurídica propia independiente de la de sus socios o partícipes, y estar constituida, como máximo, en los 24 meses anteriores a la solicitud.
- Actividad principal y domicilio social en el territorio nacional.
- Modelo de negocio innovador/novedoso o con claras ventajas competitivas.
- No estar enmarcada en el sector inmobiliario o financiero.
- Cofinanciación de las necesidades financieras asociadas al proyecto empresarial.
- Nivel de fondos propios, como mínimo, igual a la cuantía del préstamo solicitado
- Estructura financiera equilibrada.
- Profesionalidad en la gestión.
- Viabilidad técnica y económica del proyecto empresarial.
- Cuentas depositadas en Registro Mercantil o Registro Público que proceda, del último ejercicio cerrado.

La siguiente tabla muestra la entrada de fondos propios de los accionistas, así como de deuda

TOTAL	AÑO 0	AÑO 1	AÑO 2	AÑO 3	AÑO 4
-------	-------	-------	-------	-------	-------

Aumento Deuda	89.930	- €	89.930,33 €	- €	- €	- €
Aumento Fondos propios	-179.861	89.930,33 €	89.930,33 €	- €	- €	- €

Tabla 68: Entrada de Deuda y FP

En la siguiente tabla además puede observarse en detalle la deuda contraída, su devolución e intereses

		ARRANQUE	AÑO 0	AÑO1	AÑO2	AÑO3	AÑO4
		89.930					
N(Deuda viva)		-	-	89.930	71.944	53.958	35.972
Devolución principal		-	-	-	17.986	17.986	17.986
Intereses (sobre D viva)	4%	-	-	3.523	2.818	2.114	1.409
Plazo devolución	5	-	-				
ΔDEUDA		-	-	89.930	- 17.986	- 17.986	- 17.986

Tabla 69: Deuda

Con estos datos queda obtener el flujo de caja del accionista, que resulta de sustraer el aumento de deuda a los flujos de caja operativos:

	AÑO 0	AÑO 1	AÑO 2	AÑO 3	AÑO 4
--	-------	-------	-------	-------	-------

(+) incremento de deuda	0	89930	-17986	-17986	-17986
Flujo de caja accionista	-107589	-72272	84266	149403	124980
Flujo accionista acumulado	-107589	-179861	-95594	53809	178789

Tabla 70: Flujos de caja del accionista

Es necesario aclarar en este punto que el flujo del accionista no supone necesariamente el pago de dividendos cuando sus valores son positivos. Su cálculo es parte del estudio de la rentabilidad, en tanto que nos permite ver a qué volumen asciende ésta en el supuesto caso de que el reparto de dividendos se produjese. Como se observará más adelante en el balance, los flujos de caja se retienen.

11.4 VALORACIÓN DE LA EMPRESA

A lo largo de este apartado se pretenderá valorar la empresa a través de uno de los métodos más populares y utilizados, el descuento de flujos de caja del accionista. Para ello seguiremos los siguientes pasos:

1. Estimación de los flujos de caja libres futuros durante “n” años. (realizado en el apartado anterior)
2. Elección de una tasa de descuento que refleje el riesgo de la inversión.
 - a. Obtención de una beta desapalancada (sin tener en cuenta la deuda) que refleje el riesgo de nuestra empresa, mediante dos vías:
 - i. beta desapalancada media del sector en que operamos (demasiado genérica)
 - ii. beta desapalancada cualitativa en función de los riesgos ponderados de nuestra empresa.
 - b. Apalancamiento de la beta para la estructura de capital (Deuda / Equity) de nuestra empresa
 - c. Obtención de la tasa de descuento a partir de la beta apalancada (CAPM)
3. Realizar el descuento de los flujos de caja y del valor residual.
4. La suma del descuento de los flujos y el valor residual será el valor de la empresa.

Para obtener una tasa de descuento realista a la hora de valorar una empresa es mandatorio obtener el coeficiente beta de la misma. Este valor es un indicador del riesgo y volatilidad del activo al que representa, en este caso nuestra empresa.

Uno de los métodos más habituales para obtener una beta aproximada de la empresa es estudiar las betas medias del sector en el que operan. Concretamente se toman las betas desapalancadas (unlevered), esto es, los coeficientes beta sin tener en cuenta el riesgo introducido por la deuda.

En nuestro caso se han seleccionado dos sectores que se asimilan al nuestro y se ha obtenido la beta media de los mismos [26]:

Sector 1: Software (Entertainment)	1,10
Sector 2: Software (System & Appl)	1,12
Promedio beta desapalancada	1,11

Tabla 71: Betas de sectores similares

Éste método si bien hace las veces de punto de partida no obtiene un valor muy preciso. Para comprobarlo no hay más que estudiar las empresas que componen cada sector. Encontraremos grandes firmas como Microsoft, Apple o Google, cuyos volúmenes son muy lejanos a los previstos para nuestra empresa. Por tanto se ha decidido obtener otra beta desapalancada a través de una metodología cuantitativa más precisa. Para ello se ha seguido el siguiente proceso:

- Aislar y determinar las fuentes de riesgo de la empresa.
- Obtener el peso que tiene cada fuente en el total de riesgo de la empresa.
- Determinar una medida de riesgo para cada fuente, de 1 (bajo) a 5 (alto).

Siguiendo este procedimiento, la tabla siguiente muestra las fuentes de riesgo, su peso y su medida de riesgo para la empresa, así como la beta desapalancada obtenida:

Negocio, sector y producto	3	10%
Apalancamiento operativo	2	15%
Directivos	3	20%
Exposición a otros riesgos	2	5%
Mercado relevante	3	5%
Flujos	2	15%
Liquidez de la inversión	2	10%
Acceso a fuentes de fondos	2	10%
Socios	1	5%
Estrategia	2	5%
BETA UNLEVERED	1,15	

Tabla 72: Beta cualitativa

Finalmente la beta obtenida no dista mucho de la media del sector (1,15 frente a 1,201).

El siguiente paso será apalancar el coeficiente beta utilizando la estructura Deuda / Equity de nuestra empresa. Esta estructura de capital varía cada periodo, de manera que la beta apalancada será diferente en el tiempo.

Para realizar este cálculo, partimos de la afirmación de que la beta del activo de una empresa endeudada (β_L) es igual a la media ponderada de las betas de los recursos propios (β_e) y de la deuda. Considerando que la deuda tiene un riesgo despreciable calculamos la beta apalancada a través de la siguiente expresión:

$$\beta_L = \beta_U * \frac{E + D}{E}$$

Una vez obtenida la beta apalancada aplicamos el modelo Capital Asset Pricing Model (en adelante CAPM) para obtener la tasa de descuento a la que se descontarán los flujos de caja del accionista:

$$r_e = r_f + \beta_L(r_m - r_f)$$

- r_e : tasa de descuento
- β_L : beta apalancada obtenida anteriormente
- r_f : tasa libre de riesgo. Se utilizará en este estudio el bono español a 10 años
- r_m : prima de riesgo de mercado: IBEX 35 a largo plazo.

Con todos estos datos introducidos en el modelo CAPM, la tabla anterior muestra la tasa a la que descontar los flujos de caja a lo largo del periodo estudiado. Además se muestra finalmente el flujo de caja ya descontado para cada tasa de descuento:

	AÑO 0	AÑO 1	AÑO 2	AÑO 3	AÑO 4
Beta desapalancada	1,15	1,55	1,55	1,55	1,55
Beta apalancada	1,15	3,03	1,91	1,67	1,60

Tasa de descuento por años	7,48%	16,86%	11,29%	10,09%	9,74%
-----------------------------------	-------	--------	--------	--------	-------

Tabla 73: betas y tasas de descuento

El paso final sería el cálculo del VAN y TIR. Para obtener el VAN únicamente hay que tomar el flujo de caja descontado acumulado (sumando todos los flujos de caja descontados). La TIR se calcula igualando a 0 el VAN y obteniendo la tasa de descuento. La siguiente tabla muestra ambos valores con y sin valor terminal:

Valor descontado anual	-100101,35
VAN (sin TV)	88.373
TIR (sin TV)	29,53%
VAN (con TV)	550.436
TIR (con TV)	62%

Tabla 74: VAN y TIR

Ambos indicadores sugieren que la **inversión en esta empresa es rentable.**

11.5 BALANCE

La siguiente tabla muestra el balance contable previsto para los primeros 5 años:

	ARRANQUE	AÑO 0	AÑO1	AÑO2	AÑO3	AÑO4
ACTIVO	179.860,67 €	171.306,72 €	186.099,93 €	387.006,59 €	751.204,14 €	1.142.262,97 €
Caja	179.860,67 €	82.271,74 €	10.000,00 €	94.266,36 €	243.669,67 €	368.649,69 €
Caja Extra	179.860,67 €	72.271,74 €	- €	84.266,36 €	233.669,67 €	358.649,69 €
Caja mínima		10.000,00 €	10.000,00 €	10.000,00 €	10.000,00 €	10.000,00 €
Cientes		- €	1.668,58 €	32,1 74.319,97 € 3 €	80.207,07 €	
Stock		- €	- €	- €	- €	- €
Hacienda pública deudora		1.034,99 €	1.876,36 €	2.524,76 €	5.941,17 €	7.187,88 €
Inmovilizado		88.000,00 €	172.555,00 €	255.983,33 €	427.273,33 €	686.218,33 €
PASIVO + NETO	179.860,67 €	171.306,72 €	186.099,93 €	387.006,59 €	751.204,14 €	1.142.262,97 €
PASIVO	- €	5,10 €	91.722,72 €	78.482,47 €	66.314,96 €	49.324,34 €
Proveedores		5,10 €	1.582,15 €	2.224,96 €	2.992,44 €	3.246,11 €
Hacienda Pública acreedora		- €	210,24 €	4.313,25 €	9.364,32 €	10.106,09 €
Deudas a l/p con entidades de crédito		- €	89.930,33 €	71.944,27 €	53.958,20 €	35.972,13 €
NETO	89.930,33 €	171.301,63 €	94.377,21 €	308.524,11 €	684.889,18 €	1.092.938,63 €

Capital	89.930,33 €	179.860,67 €	171.301,63 €	94.377,21 €	308.524,11 €	684.889,18 €
Resultado del ejercicio		-	-	214.146,90 €	376.365,07 €	408.049,45 €
		8.559,04 €	76.924,42 €			

Nótese que se ha representado un periodo inicial (arranque), en el que se representa el capital aportado por los socios inicialmente. Se anota como tal en el activo y como caja (efectivo) en el activo. El resto de los años las anotaciones en el balance son las siguientes:

- **ACTIVO**
 - Clientes y Stock: estudiados en el apartado 6.3: *Flujos de caja*
 - Caja mínima: establecida en 10000€, representa el efectivo mínimo a disponer para asegurar la consecución de las actividades de la empresa.
- **PASIVO**
 - Proveedores: estudiados en el apartado 6.3: *Flujos de caja*

Puede observarse que el contenido en caja que no es caja mínima se ha denominado caja extra. Su estructura sigue una evolución similar a la del resultado del ejercicio: Durante los 2 primeros periodos en que los resultados de la actividad de la empresa son negativos, la caja extra va recudiéndose. En el momento en que la empresa comienza a generar beneficios esta cifra comienza a aumentar.

11.6 SENSIBILIDAD

En este apartado se estudiará cómo se comportan los principales indicadores de la viabilidad del proyecto ante ciertos cambios.

Para ello se realizarán simulaciones haciendo uso del software Oracle Crystal Ball. El funcionamiento de este software es sencillo:

- Selección de parámetros de suposición: Se seleccionan los valores que el software variará. En nuestro caso se ha seleccionado una distribución triangular donde se establecen los valores mínimo, más probable y máximo. Los valores mínimo y máximo tienen probabilidad mínima, mientras que según se acercan al valor más probable aumentan su probabilidad.

- Selección de variables de previsión: Se seleccionan los valores de estudio, aquellos de los que se desea ver su variación ante cambios en los parámetros de suposición

Las siguientes ilustraciones muestran los parámetros de suposición seleccionados y su distribución. Se ha decidido variar los parámetros relativos a ingresos: Numero inicial de usuarios, las tasas de crecimiento en ambas fases y la tasa de decrecimiento:

Tabla 75: Suposiciones

Para estas variaciones, estudiaremos posibles cambios en indicadores de viabilidad de la empresa: TIR y VAN, con y sin terminal value. Las siguientes ilustraciones muestran los resultados:

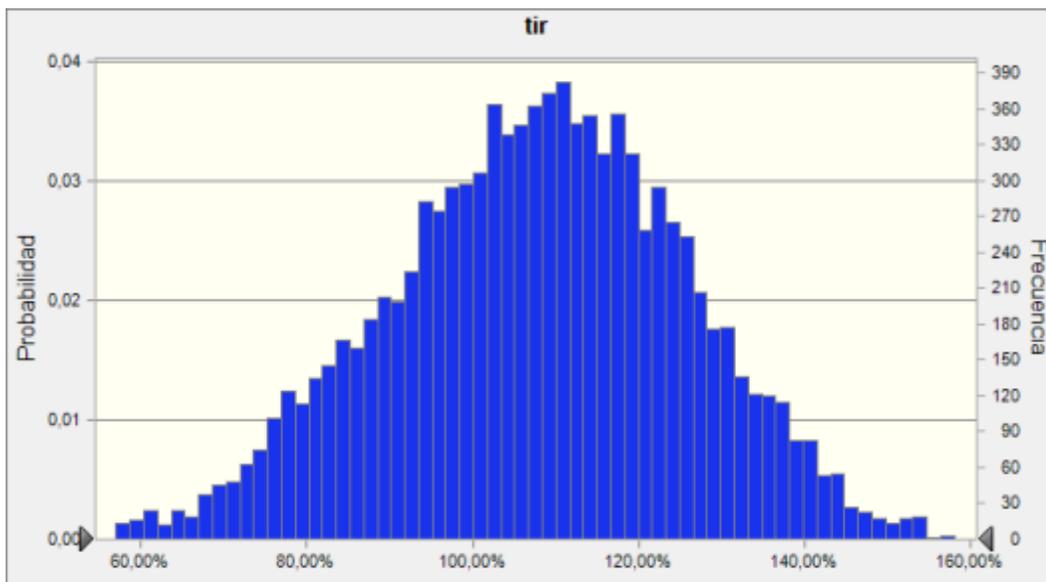


Ilustración 34: Previsión de TIR sin TV

En el caso del TIR, observamos que para la variación de los 4 parámetros que afectan a la evolución de los ingresos su valor puede moverse entre 60 y 170% aproximadamente. Además, alcanza su probabilidad máxima en torno al 100-110%. En todo momento son valores muy altos indicadores de rentabilidad.

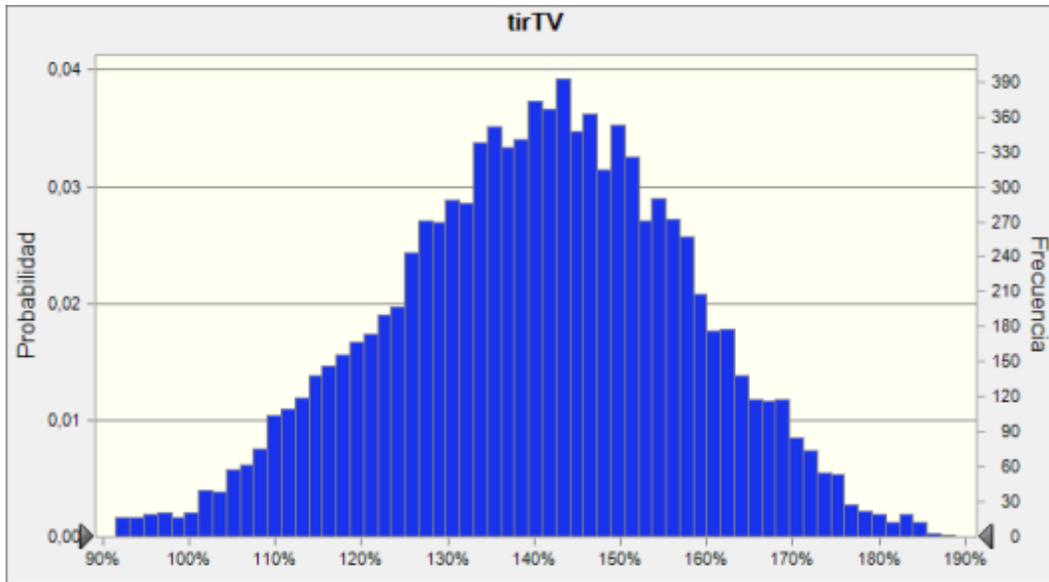


Ilustración 35: Previsión de TIR con TV

Incluyendo el terminal value los valores son algo mayores, de nuevo hablamos de números muy positivos que indican posible éxito de la empresa.

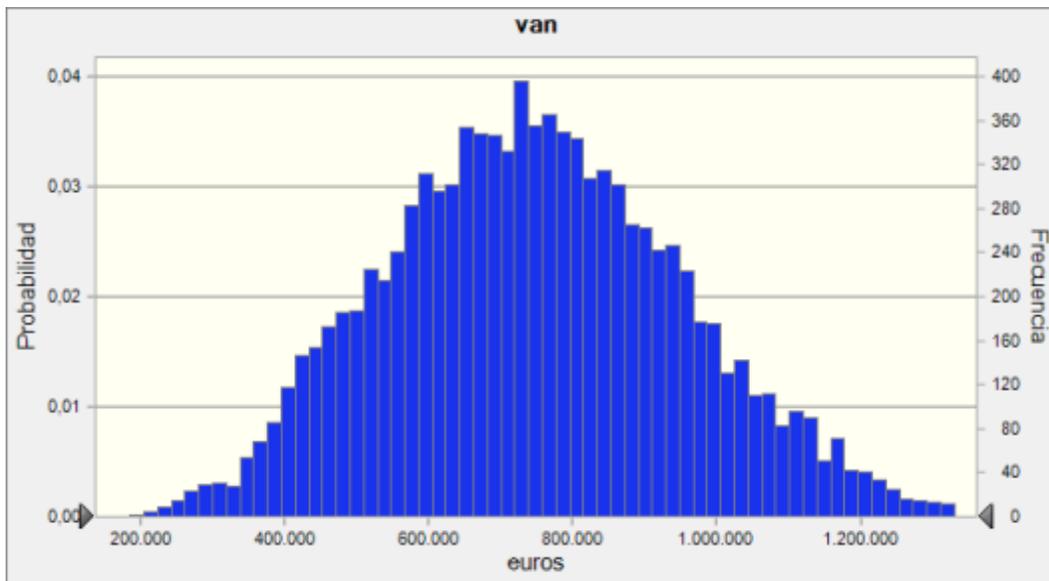


Ilustración 36: Previsión de VAN sin TV

El VAN del proyecto toma valores de entre 200.000 y 1.200.000 € en función de la variación de los parámetros con afectación en los ingresos. La probabilidad máxima se mueve en torno a los 700.000

euros. Al igual que se ha observado en la tir, son valores muy altos y en este caso motivan la creación del proyecto en tanto que son indicadores de su viabilidad

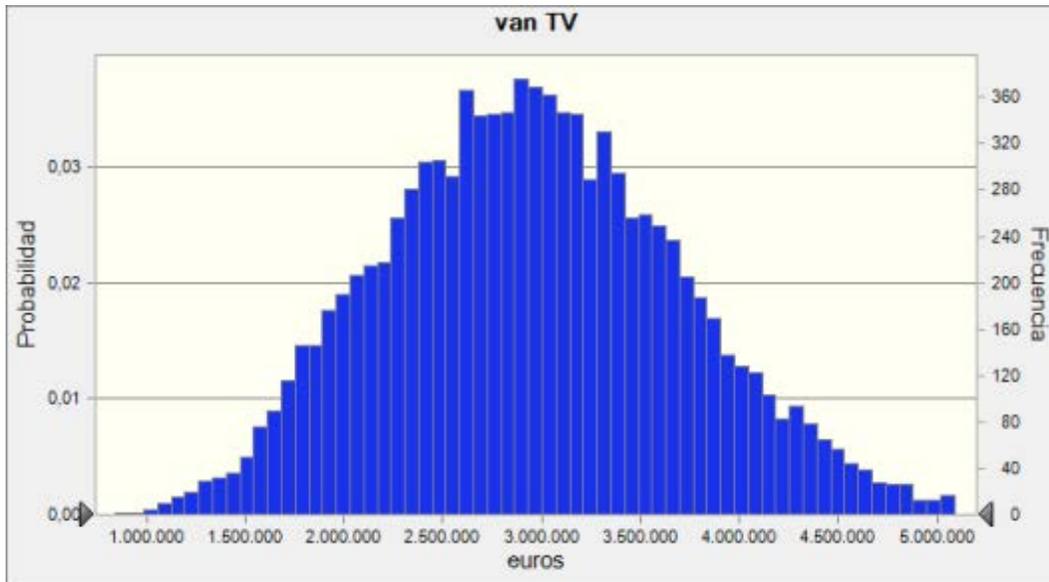


Ilustración 37: Previsión de VAN con TV

Incluyendo el terminal value los valores son algo mayores, de nuevo hablamos de números muy positivos que indican posible éxito de la empresa.

Como ha podido observarse, los valores en todos los casos son positivos para el desarrollo de la empresa, con lo que su rentabilidad goza de gran probabilidad.

Es de especial interés conocer la sensibilidad de estos parámetros ante cambios en cada una de las variables estudiadas, de cara a conocer con previsión aquellos puntos de mayor debilidad para la empresa. El software utilizado nos provee un estudio de estas características, cuyos resultados se muestran en la siguiente ilustración:

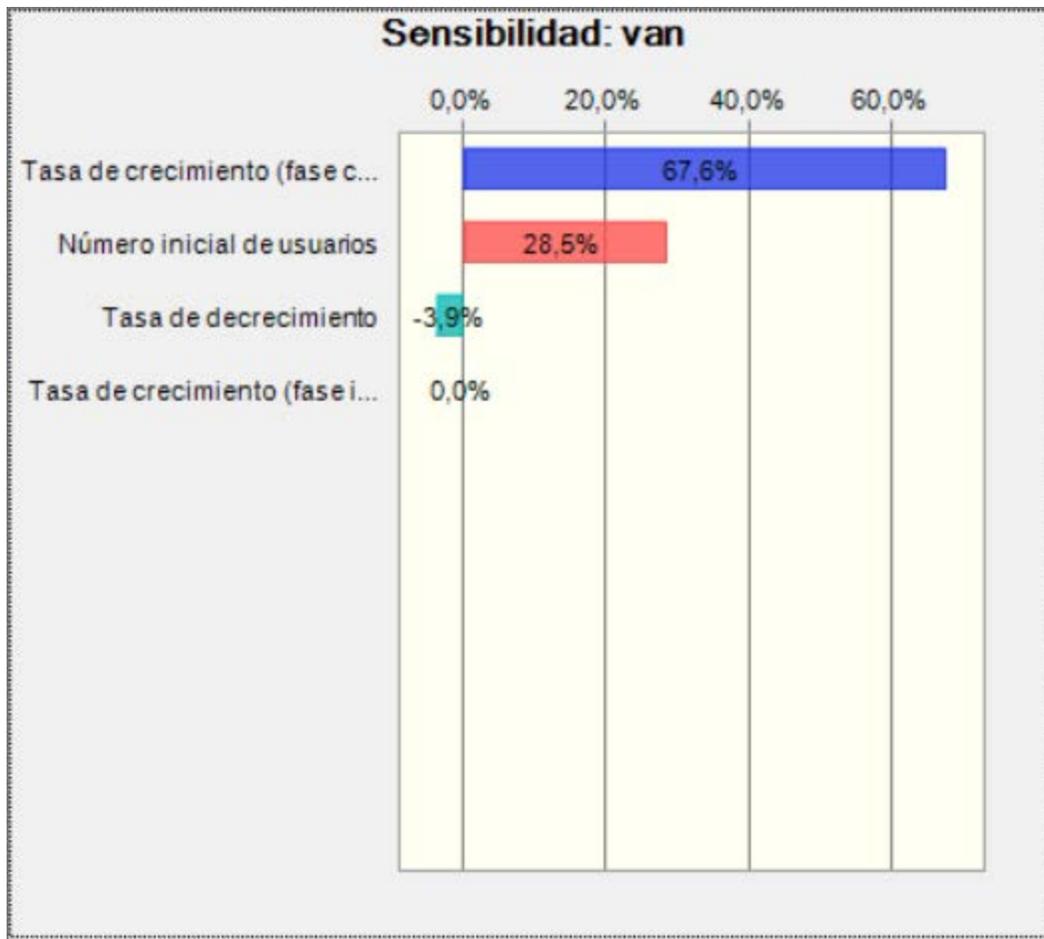


Ilustración 38: Sensibilidad de indicadores de viabilidad

La sensibilidad para los 4 indicadores es muy similar, con lo que mostrar uno de ellos es suficiente. Estos resultados nos revelan que la tasa de crecimiento en la fase de inicio, cuando aún la aplicación no es conocida en exceso, no tiene especial importancia en el desarrollo de la empresa. Es por ello que se pueden tomar medidas como la reducción de gastos en publicidad focalizados en adquisición de producto de pago por ejemplo, interesando más orientar la promoción en alcanzar el primer punto de inflexión.

La tasa de crecimiento en la fase de crecimiento sin embargo afecta en gran medida a los indicadores, de tal manera que ha de realizarse un gran esfuerzo en maximizar los usuarios en esta fase.

11.7 INDICADORES FINANCIEROS

Para aportar información adicional de la empresa y su valor se han calculado indicadores financieros adicionales:

- Apalancamiento financiero: definido como la proporción entre el capital propio (Equity) y el crédito (Deuda), refleja el grado en que la empresa depende de su deuda.
- Ratio de endeudamiento: Relación entre los fondos propios de una empresa y su deuda tanto a corto como a largo plazo.
- ROE: ratio que relaciona el beneficio de la empresa con los recursos necesarios para obtenerlos.

12 ANEXO II: ESTUDIO DE DESCARGA DE APLICACIONES MÓVILES

Fuente	Appbrain						Calculada	Calculada
Category	All apps	Average rating	Apps with >50K downloads	Paid apps	Average price	Low quality apps	% of success apps	% of Low quality apps
Libraries & Demo	542							
	5	3.94	243	297	\$4.70	417	4%	8%
	168							
Music	1	3.90	296	158	\$1.71	118	18%	7%
	549							
Comics	3	3.96	453	1281	\$2.98	789	8%	14%
	534							
Board	7	3.99	548	518	\$1.92	401	10%	7%
	422							
Word	3	4.03	570	450	\$2.19	332	13%	8%
	231							
Medical	43	4.04	631	3996	\$9.97	4716	3%	20%
	796							
Trivia	7	3.97	787	438	\$2.64	870	10%	11%
	635							
Weather	9	4.01	793	690	\$2.24	467	12%	7%
	658							
Casino	9	3.99	962	613	\$4.22	1447	15%	22%

	118							
Adventure	49	4.01	1056	1102	\$2.82	1186	9%	10%
	102							
Business	569	4.12	1064	3312	\$8.03	21778	1%	21%
Educational	159							
I	09	4.02	1088	2569	\$2.14	1852	7%	12%
	568							
Strategy	0	3.93	1099	716	\$2.21	413	19%	7%
Role	436							
Playing	7	3.91	1223	548	\$3.24	378	28%	9%
Transporta	230							
tion	60	3.96	1391	1767	\$4.26	1994	6%	9%
	893							
Family	9	3.82	1399	847	\$1.64	1135	16%	13%
	372							
Sports	13	4.12	1556	4828	\$4.04	4882	4%	13%
	927							
Card	4	3.89	1571	1422	\$1.87	1116	17%	12%
	319							
Shopping	30	4.03	1593	604	\$3.39	3281	5%	10%
Sports	881							
Games	6	3.78	1868	1214	\$1.53	854	21%	10%
	385							
Finance	67	4.03	2002	2492	\$4.88	6749	5%	17%
Health &	448							
Fitness	08	3.92	2104	5816	\$3.28	6686	5%	15%

			879						
Simulation	6	3.66	2130	695	\$1.99	874	24%	10%	
			374						
Social	41	4.12	2151	1732	\$3.76	2636	6%	7%	
Travel &	711			1130					
Local	89	4.02	2175	7	\$3.65	15603	3%	22%	
News &	429								
Magazines	82	4.03	2216	817	\$2.17	5069	5%	12%	
Media &	209								
Video	85	3.92	2252	1959	\$5.49	2065	11%	10%	
			969						
Racing	5	3.82	2452	841	\$1.42	693	25%	7%	
			194						
Action	12	3.95	2594	1355	\$1.93	1617	13%	8%	
Productivit	468								
y	34	4.02	2733	6545	\$3.77	3914	6%	8%	
Communic	381								
ation	96	4.05	2876	2968	\$3.16	2843	8%	7%	
Photograp	218								
hy	65	3.93	3060	2075	\$3.71	2331	14%	11%	
Music &	577								
Audio	79	4.18	3883	6167	\$2.17	4677	7%	8%	
Books &	792			2024					
Reference	20	4.17	4086	8	\$4.15	19770	5%	25%	
			475						
Arcade	23	3.99	4843	5781	\$1.55	2993	10%	6%	

	120			2306				
Education	618	4.06	4970	4	\$3.88	21465	4%	18%
	109							
Lifestyle	536	4.03	5694	9270	\$2.82	20212	5%	18%
	581							
Puzzle	77	3.97	5838	7864	\$1.51	5377	10%	9%
	513							
Casual	88	3.81	7721	5559	\$1.45	5073	15%	10%
	964			3088				
Personalization	87	4.17	8074	0	\$1.71	12295	8%	13%
	956			1298				
Tools	11	3.97	8708	6	\$3.01	6656	9%	7%
	103			1015				
Entertainment	366	3.92	9127	6	\$2.55	11316	9%	11%

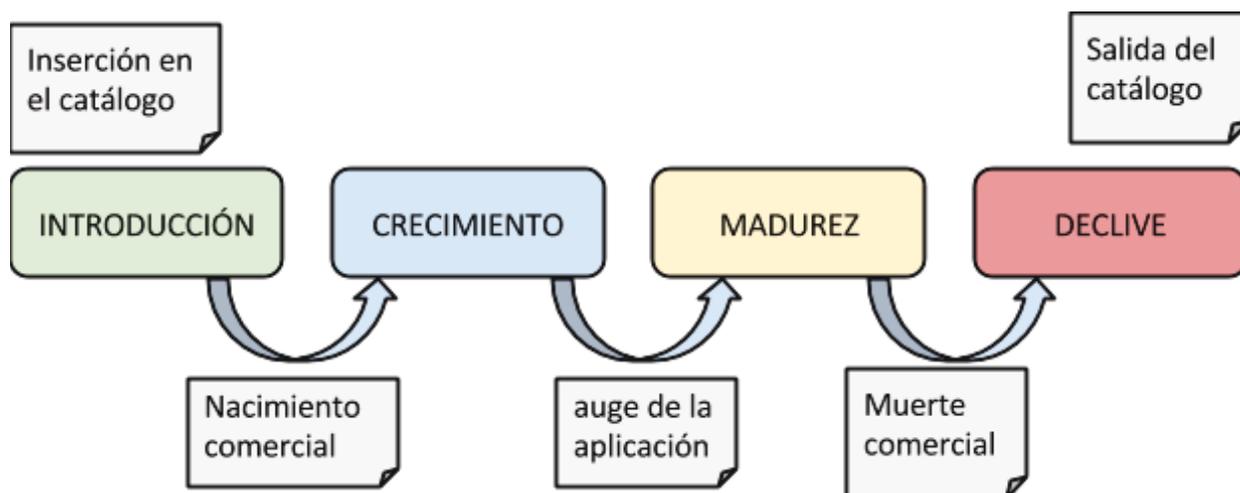
13 ANEXO III: CICLO DE VIDA DE LAS APLICACIONES MÓVILES

13.1 INTRODUCCIÓN

Es de vital importancia para predecir el movimiento futuro de nuestro producto el analizar cómo se comporta el mercado ante un producto del tipo aplicación para Smartphone / Tablet. Con el rápido crecimiento del mercado, la industria ha modelado una gran variedad de funciones que estiman y predicen el ciclo de vida (en adelante PLC, Product Life Cycle) que recorrerá el producto desde su salida a su (posible) declive. En este apartado modelaremos un nuevo ciclo de vida aplicando el modelo clásico de PLC a las aplicaciones móviles. Para su mayor exactitud nos basaremos en las estadísticas públicas de los principales mercados de aplicaciones móviles.

13.2 ETAPAS DEL CICLO DE VIDA

A continuación se muestra el modelo habitual de ciclo de vida de un producto, donde las notas en tono grisáceo referencian los puntos críticos que pueden definir el cambio de estado en el caso concreto de las aplicaciones móviles.



- **Inserción en el catálogo:** Momento en que la aplicación completa su desarrollo y es añadida al catálogo de aplicaciones de la empresa. En concreto definiremos este punto en el momento en que la aplicación se encuentre disponible en la Play Store / App Store

- **Nacimiento comercial:** Momento en que la aplicación comienza a llamar la atención de un número mayor de usuarios y a ser conocida. Se ha optado por situarlo en el momento en que la aplicación en cuestión aparece en el ranking de las 200 aplicaciones más descargadas de la plataforma [27]
- **Auge de la aplicación:** momento en que la aplicación alcanza su máximo número de descargas en todo el ciclo de vida.
- **Muerte comercial:** Suele definirse cuando no se supera una pequeña proporción del máximo número de adquisiciones de un producto en su ciclo de vida. En nuestro caso las adquisiciones se asemejan a las descargas con respecto a un periodo de tiempo, y tomaremos el porcentaje en un 20%.

13.3 CUANTIFICACIÓN

Se ha tomado para este análisis el conjunto de datos cuyo desglose por categoría es el siguiente [27]:

Category	Number of Apps
NAVIGATION	1807
SOCIALNETWORK	1824
NEWS	1912
FINANCE	1925
PRODUCTIVITY	2385
HEALTH & FITNESS	2617
REFERENCE	2866
PHOTOGRAPHY	3046
MUSIC	3067
MEDICAL	3134
SPORTS	3156
UTILITIES	3274
BUSINESS	3551
TRAVEL	3609
LIFESTYLE	3645
ENTERTAINMENT	4384
EDUCATION	4428
GAME	4688
BOOKS	5058
ALL (cross-category)	5560

En este análisis se basan únicamente en aplicaciones cuyo ciclo de vida ya se ha completado, y cuyos puntos críticos son localizables.

En el mismo estudio y desglosadas las aplicaciones muestreadas por categoría, han cuantificado la duración de cada etapa del ciclo de vida de las aplicaciones.

Category	Ave Intro	Ave Growth	Ave Maturity	Ave Total	Ave Growth + Maturity
EDUCATION	12.785	6.209	21.04	40.034	27.249
ENTERTAINMENT	5.975	7.004	21.334	34.313	28.338
BOOKS	15.861	8.837	23.039	47.737	31.876
LIFESTYLE	7.505	8.892	25.035	41.432	33.927
ALL (ALC)	20.859	9.853	24.124	54.836	33.977
GAME	13.756	10.766	25.847	50.369	36.613
BUSINESS	8.265	9.799	27.817	45.882	37.616
UTILITIES	6.841	10.432	28.248	45.521	38.680
PHOTOGRAPHY	4.368	7.842	32.175	44.385	40.017
MUSIC	4.703	8.874	33.191	46.768	42.065
REFERENCE	10.796	11.411	31.309	53.517	42.720
FINANCE	7.380	10.790	33.494	51.664	44.284
HEALTHCARE AND FITNESS	7.130	7.941	36.562	51.632	44.503
NEWS	9.793	11.908	32.873	54.573	44.781
PRODUCTIVITY	8.175	10.753	35.250	54.178	46.003
TRAVEL	14.002	10.211	38.103	62.316	48.314
MEDICAL	8.741	7.604	41.754	58.100	49.358
SPORTS	7.831	12.214	40.243	60.287	52.457
SOCIAL NETWORK	5.790	7.997	44.586	58.373	52.583
NAVIGATION	7.653	11.735	46.903	66.290	58.638

De esta información se desprenden datos de gran interés:

- El ciclo de vida de una aplicación móvil es mucho más corto que el de otros productos.
- El ciclo de vida de las aplicaciones móviles varía considerablemente en función de la categoría de la aplicación
- La duración media de una aplicación hasta que alcanza la madurez es de 53 días.
- En lo que respecta a las aplicaciones de categoría “deportes”
 - Tienen una etapa de crecimiento muy larga, así como una de madurez de gran duración en comparación con el resto de categorías.
 - El tiempo hasta que comienza su crecimiento es más reducido en comparación con las aplicaciones del resto de categorías.

14 ANEXO IV: DIAGRAMA DE GANTT

A continuación se muestra el diagrama de Gantt completo de estimación para el proyecto llevado a cabo:

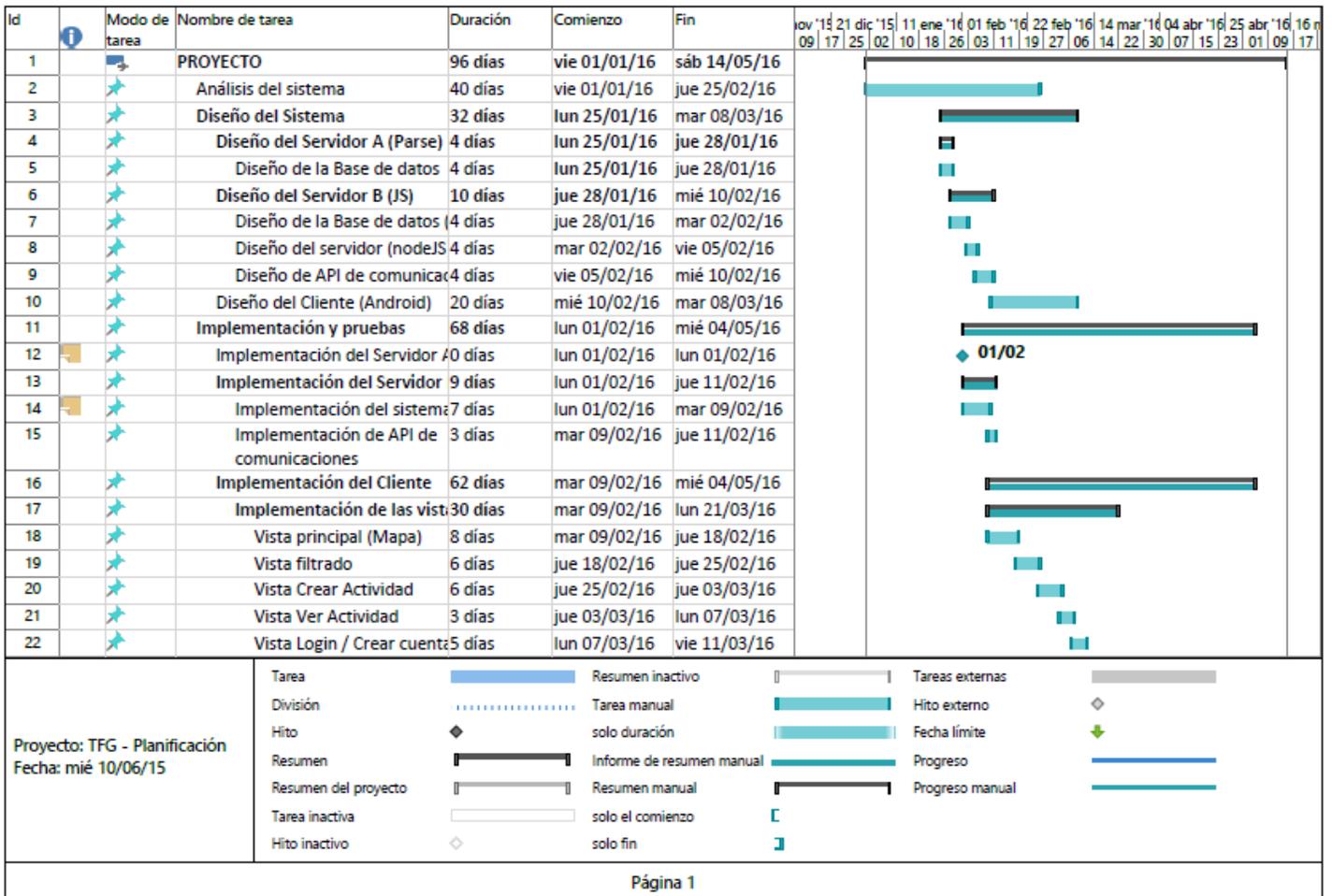


Ilustración 39: Gantt extendido (1)

15 ANEXO V: INTRODUCCIÓN Y CONCLUSIONES EN INGLÉS

Tal y como se requiere en el plan de estudios 2011, a lo largo de este apartado se detalla una sección del proyecto en inglés. Concretamente abarcará el resumen ejecutivo, Introducción y conclusiones. Este anexo no se incluye en el índice con el objetivo de evitar redundancias en sus entradas.

15.1 ABSTRACT

The project developed along this pages corresponds to the development of the main product of WeSweat, SL. The purpose for which this company has been made is to build a range of products that simplifies the creation of sport events through generating sport networks, both individual and collective, ultimately facilitating their organization.

The main incentive for the creation of this project lies in the identification of two existing fundamental problems in de daily life of usual sports. The first problem is the lack of members to achieve the minimum capacity needed to perform a sport session. This lack is usually caused by schedule incompatibilities.

The second problem is the unknowing by people who plays certain sport about if there are people playing that sport, or having similar interests around them. Both situations often end up in the loss of interest about the aforementioned sport, which ultimately may enhance the abandonment.

The large number of athletes not connected in conjunction with the penetration of smart mobile devices and cloud-based services represent a great business opportunity for the development of a mobile application to solve the problems detected by creating a large network of geographic athletes.

Once identified the business opportunity and proposed the idea to exploit it, it will be pretended along this document to describe the detailed development process of the mobile product WeSweat. It will start analyzing the environment around the business; after that it will be described the design and implementation processes corresponding to each architectural component of the product, detailing also everything concerting legal, financial and fiscal topics.

15.2 OBJECTIVES

After studying the environment that influences the activity of this company, in this section we will analyze the objectives to achieve throughout the project.

These objectives will determine the degree of success of the project, concretely through the study of the similarity between actual results in the final stages of the work, and the mentioned objectives to achieve proposed in this section.

- **Reveal the context and situation of the environment variables around the company:** Both from the business and technological point of view and perspective it will be pretended to describe the state of the art of every component with potential influence on the project and its implementation. These variables will cover a wide range of aspects, including from technologies to be used in client and server to suppliers, competitors or potential customers of the product.
- **Study design alternatives and obtaining an optimal conclusion:** It is intended to detect and analyze the different ways of accomplishing the project in terms of design. After analyzing the advantages and disadvantages of each model, one of the main objectives to reach is to select a design architecture that optimizes the operation of each component.
- **Describe building and implementation process and provide a prototype to the customer:** A key objective to achieve will be to reveal details emerged during the implementation of the chosen design. In addition, the ideal goal to achieve here lies in the development of a prototype to be shown to the client. This prototype should show the different views that will compose the final product, as well as the working business logic in communication with the servers.
- **Serve as a guide for further projects and developments in the future:** Through the detailed and documented process of designing and implementing the product, one of our aims is to convert

this document into a source of useful information to be reviewed on future projects developed by third parties.

15.3 MOTIVATION

Even before I started my double degree in Colmenarejo Campus of Universidad Carlos III de Madrid, I have been interested in mobile development, but it was in my second year there when I started developing Android applications.

Since that I've involved into Android development world in multiple ways. I've attended to dozens of Android events, conferences and workshops. I have been working as a collaborator with the Applied Artificial Intelligence Group (GIAA), developing augmented reality applications for Android devices. Currently I'm working as Android developer at a phone carrier company.

On the other hand, I'm a regular basketball player. I have played this sport since I was a child. I am currently part of a team in my hometown, but I also meet my friends to play. Concretely is this latter case the one which has made this idea arises. For us is very common to find new people playing our favorite sport at the court we usually play, and sometimes a question appears: If you love this sport and you live near here, How come that we have not played together before?

So this way, with the guidance of my both tutors, I started this project that has helped me finish in successfully this period of my life.

15.4 DOCUMENT STRUCTURE

This section will reveal the different parts and subsections in which this document is segmented.

By reading by this guide, the reader should find easier to read, understand and find information that could be relevant for him.

15.4.1 Introduction

This document, which describes the complete process followed to perform this project, is composed by ten main sections, which are also complemented by four annexes.

The first section is the introduction, In the introduction has been written with the aim to describe, without going into the technical and procedural details, what consist the project, why it has been chosen, which are the objectives to be achieved, what resources has been used to fulfill it and what information this document contains.

15.4.2 State of the art

This section will describe and explain the context of this project. Concretely the idea of the state of the art is to perceive and detect those variables which may affect the performance of the Company and its products.

As a double degree student it has been pretended to contribute to this section with both a technological and a business point of view. Therefore, in this section we will study a wide range of variables: from similar products found in the market to technologies that have been used at client and server side.

15.4.3 System Analysis

This section will reveal, as accurately as possible, the system specifications and requirements, compiling all the functionalities of the system. It will start by a system definition showing the main problem to resolve across this Project. After that, the requirements collected and synthesized will be shown with table format. We will study the range of possible use cases that could take place across the life of the product, and it will finish with a traceability matrix.

15.4.4 System Design

Subsection “System Design” will focus on the solution found to solve the problem mentioned in the previous subsection, “System Analysis”. Concretely it will define and describe the architecture of the system.

It will be pretended in this subsection to give a general point of view about the architecture just to give a first idea of its components. Once achieved that, it will be possible to elaborate on each component, its behavior and functionality.

It is necessary to mention that one of the objectives of this section is to show different alternatives to achieve a solution, then study their advantages and drawbacks and finally choose an efficient solution.

15.4.5 Operation environment

This section is more technical, and shows up the setup of the framework that will be used to build the product. It will also show the technical details that have come up during de execution of the project.

15.4.6 Planning and estimated Budget

This section will show the estimated planning to build the Project and its tasks. The Gantt diagram technique will be used to show the results.

On the other hand, the estimated Budget subsection will link to one of the annexes, which will refer to the economic plan carried out in the business plan of this project. Because of that this estimated Budget section will be a little more complex, and also more financial and business oriented.

15.4.7 Legal

In this section we will summarize everything related to the legal component of the company and its product. As a company, we will have to perform some bureaucratic procedures in order to start our activity. It will be mentioned also in this section details about patent registering and data protection legal measures.

15.4.8 Conclusions and further work

In the conclusions and further work section will have different targets. Firstly, it should show how close the final results of the project are with the objectives raised at the “Introduction” section. On the other hand, further Works subsections will analyze how this project result product could be improved in the future, describing next steps to take. Finally, it will show up a personal point of view of the project, manifesting what I have learned carrying it out.

15.4.9 Bibliography

This section will show the sources that have been consulted when carrying out this project. Along this document and according to bibliography preferences, each item in this section will be referenced with a number based format.

15.4.10 Annexes

The last part of this document will include some relevant information that is not included in the main structure of the project. Concretely we will find five annexes:

15.4.10.1 Financial and Economic Plan

This annex refers to a study carried out in the business plan. Along this study it can be reviewed the incomes and expenses of the company and its production process, how it is pretended to fund the company, and some accounting issues in detail.

15.4.10.2 App Download Study

This annex refers to a study carried out in the business plan. Along this study it can be reviewed how apps are downloaded from the main mobile device application markets, filtered by different variables, mainly categories.

15.4.10.3 Application life cycle

This annex refers to a study carried out in the business plan. Along this study it can be reviewed how an app lives in the device application market it is stored, since its release to its death. It is pretended in this annex to resemble the general product lifecycle to the specific mobile application lifecycle.

15.4.10.4 Extended Gantt

This annex will show the extended planning study through a Gantt Diagram.

15.5 CONCLUSIONS AND FURTHER WORK

Throughout this section the entire work process executed it will be synthesized. Starting from that summary, it will be pretended to reveal the conclusions derived from the results.

As we commented previously, the final subsection will analyze possible future work to perform in order to improve the product.

15.5.1 Future work

Although as we have seen selected implementation covers all user needs identified in the analysis process and represented through the requirements, there are certain features and enhancements that will undoubtedly improve the user experience:

- **[Guest] Local storage (cache) of data:** Currently user requires Internet connection to run almost all existing use cases. The idea is to implement some local storage, via both cache and database persist methods, to bring the user a great user experience by reducing load times.
- **[B Server] Incorporation of push notifications and reduction of socket use:** Excessive use of sockets has a negative impact on client devices battery. Because of that it is very interesting to use push notifications as a way to notify the client when new chat messages are received, instead of keeping a socket during the entire application life cycle.
- **[Server A] Replacement of MBaaS by independent server:** Although proper operation, the price of scaling an MBaaS platform such as the one implemented in this project if the application greatly increases its user base rises enormously. Besides creating a separate server programmed by the team gives flexibility to incorporate complex features that an MBaaS cannot provide, because their APIs usually do not provide the necessary tools to do so.

- **[All] Security measures improvement:** In the case of client side, make a security analysis that should include reviewing logs, proguard configuration, and data encryption in communication with both servers and data encryption of local storage. In the case of servers, encryption of user data stored locally.

15.5.2 Conclusions:

This section describes the conclusions reached after carrying out this project and the degree of compliance of the goals set at the beginning of it.

- **Reveal the context and situation of the environment variables around the company:** It is considered that this objective has been achieved successfully. Along the “state of the art” section the business environment variables have been detected, both from a technological and business point of view.
- **Study design alternatives and obtaining an optimal conclusion:** It has been analyzed different ways to carry out the project in terms of design, detailing the advantages and drawbacks of each. Starting from that, we have selected the (for us) most efficient architecture design. We consider this aim accomplished.
- **Describe building and implementation process and provide a prototype to the customer:** Throughout this project we have detailed the implementation carried out for the selected design. It has been represented through a prototype some illustrations showing different views composing the final product. This target then is considered complete.
- **Serve as a guide for further projects and developments in the future:** It is considered that this objective has been achieved, this document can be used as a useful source of information to consult when making decisions on future projects developed by third parties.