

UNIVERSIDAD POLITÉCNICA DE MADRID

E.T.S. DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

PROYECTO FIN DE MÁSTER

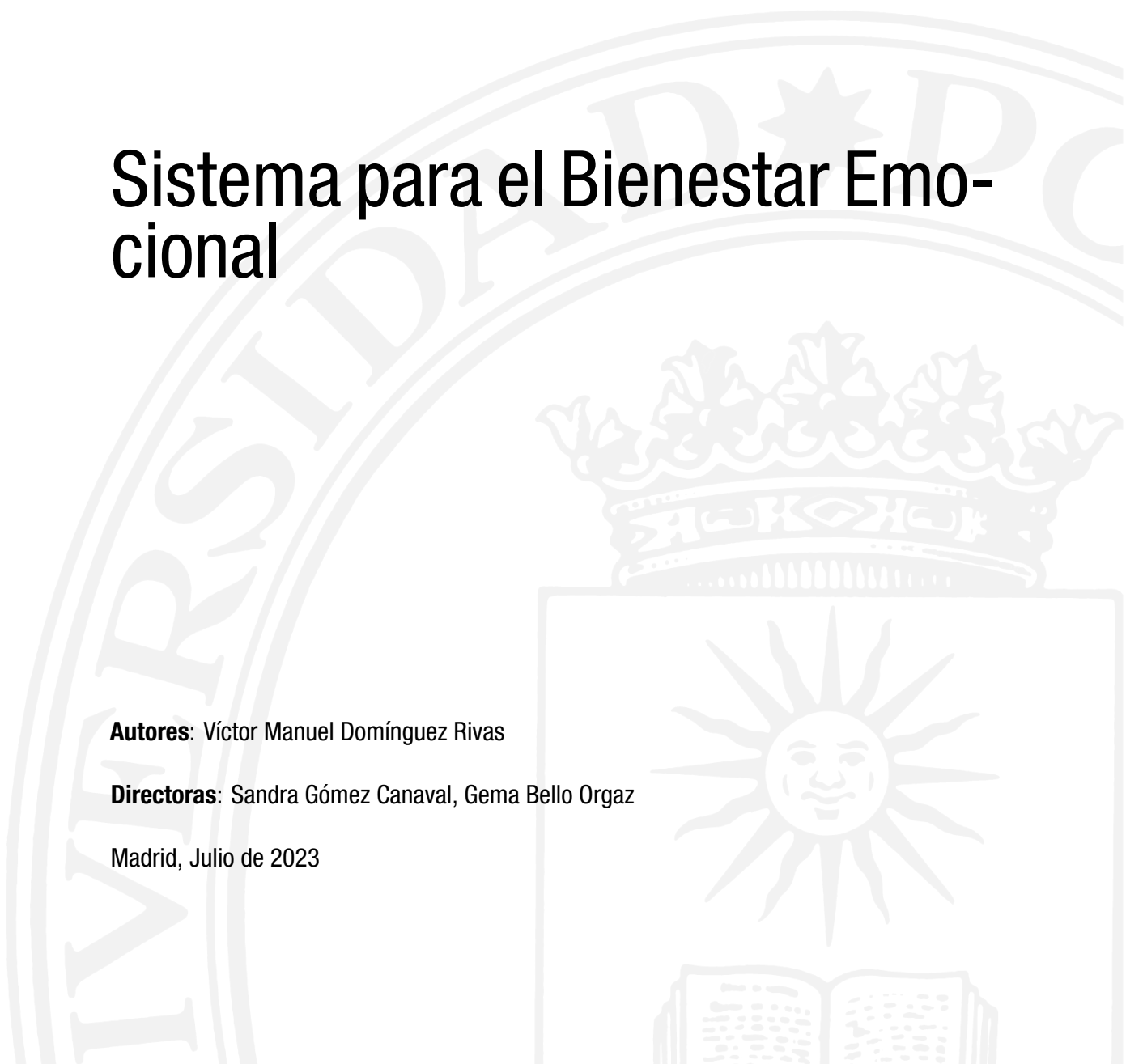
**MÁSTER UNIVERSITARIO EN SOFTWARE DE SISTEMAS DISTRIBUIDOS Y
EMPOTRADOS**

Sistema para el Bienestar Emocional

Autores: Víctor Manuel Domínguez Rivas

Directoras: Sandra Gómez Canaval, Gema Bello Orgaz

Madrid, Julio de 2023



Víctor Manuel Domínguez Rivas

Sistema para el Bienestar Emocional

Proyecto Fin de Máster, 26 de junio de 2023

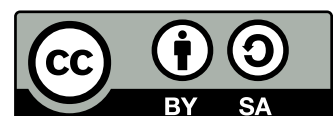
Directoras: Sandra Gómez Canaval, Gema Bello Orgaz

E.T.S. de Ingeniería de Sistemas Informáticos

Campus Sur UPM, Carretera de Valencia (A-3), km. 7

28031, Madrid, España

Esta obra está bajo una licencia **Creative Commons «Atribución-CompartirIgual 4.0 Internacional»**.



Resumen

Palabras clave: Salud Mental, Android, Wearables, Jetpack Compose, Health Connect

Abstract

Keywords: Mental Health, Android, Wearables, Jetpack Compose, Health Connect

Agradecimientos

Este proyecto es el final de un camino de siete años en la escuela, el broche de una etapa tan importante. De alguna manera estos casi once meses han sido la síntesis de un largo viaje, tanto a nivel técnico como humano. Y como en todo viaje que se precie, lo mejor no es necesariamente el destino, sino llegar hasta allí.

No habría sido posible llegar hasta aquí sin la ayuda de mis amigos, por su apoyo en los momentos más delicados y sobre todo por hacer disfrutar el camino partiendo desde la más absoluta nada. Nunca nos hemos conformado con nada, siempre hemos empujado más allá y eso lo ha hecho todo mucho más especial y profundo. Todos han tenido su contribución y la conocen perfectamente, pero por hacer esta memoria finita; quiero acordarme especialmente de Juan Luis y Rocío por haber estado juntos en todo momento, habernos complementado en tantísimos niveles y siendo poco menos que una familia.

A mi familia por haber apoyado en el día a día desde la distancia, a pesar de vivir una etapa tan complicada. En los momentos donde todo cambia, en los que tienes que decidir sobre tantas consecuencias; es donde aparece la calidad humana de las personas.

Obviamente esto tampoco sería posible sin mis tutoras. A Sandra por todo su apoyo, tanto profesional como especialmente personal. A pesar de ser un año cuanto menos complicado para todos ha estado ahí, intentando sacar lo mejor de cada momento y situación. Por supuesto también a Gema por su enorme labor a nivel técnico, ayudándome a decidir sobre tantas y tantas cosas, aportando consejos sobre cómo mejorar... y sobre todo por su faceta personal, por estar ahí y entender situaciones delicadas y también por aportar la ilusión en el día a día. Sin pasión no hay vida.

Por supuesto a todas las personas que han colaborado con su granito de arena. A Cristina y Miriam por su increíble labor en los consejos y cuestionarios. de la aplicación. También a LevelUp por ayudarme con la identidad gráfica del proyecto; sin ellos el proyecto no tendría alma.

También quiero agradecer a todas las personas que se dedican a la investigación y a la ciencia. Una sociedad que no cuida a sus científicos e investigadores está condenada a la decadencia, ya que sin ellos no sería posible el progreso.

Asimismo quiero agradecer humildemente el trabajo de personas como Aaron Swartz y Alexandra Elbakyan, e iniciativas como la Open Access por facilitar el acceso a la ciencia [1]. El acceso libre y gratuito al conocimiento científico es vital para el desarrollo de cualquier sociedad democrática. Una sociedad que lo promueve es una más formada y menos manipulable; algo vital en nuestros tiempos.

Índice general

Índice general	i
Índice de figuras	iii
Índice de cuadros	iv
1 Introducción	1
1.1 Contexto	1
1.2 Motivación	1
1.3 Objetivos	1
1.4 Justificación	2
1.5 Estructura del documento	2
2 Marco teórico y contexto tecnológico	3
2.1 Marco Teórico	3
2.2 Contexto tecnológico	3
3 Estado de la cuestión	9
3.1 Análisis de la situación actual	9
4 Metodología y materiales	10
4.1 Metodologías de desarrollo del <i>software</i> de Sistemas	10
5 Diseño del sistema propuesto	11
5.1 Descripción del caso de estudio	11
5.2 Diseño del sistema	11
6 Desarrollo del sistema	12
6.1 Project Blastoff	12

7	Resultados	22
7.1	Preparación del entorno	22
7.2	Resultados del caso de estudio	22
8	Impacto social y medioambiental	23
8.1	Contexto medioambiental	23
8.2	Aspectos éticos, sociales y económicos	23
9	Presupuesto	24
10	Conclusiones	25
11	Líneas futuras	26
12	Temporal	27
12.1	Setup del proyecto	27
12.2	Sprint #0	28
12.3	Tecnologías	32
12.4	Desarrollo	33
	Bibliografía	34

Índice de figuras

2.1	Logo actual de Android.	4
2.2	Capas de Android.	5
2.3	HTC Dream en funcionamiento.	6
2.4	Estadísticas acumulativas de las versiones de Android	7
6.1	Componentes básicos del sistema	16
6.2	Pila con la arquitectura del sistema	17
6.3	Dimensionamiento aproximado del proyecto	19
12.1	Extracto del tablero Kanban del proyecto	29
12.2	Extracto del diagrama Gantt del proyecto	29
12.3	Extracto de la base de datos de referencias	30
12.4	Extracto de la base de datos de apuntes del proyecto	30
12.5	Extracto de los documentos alojados en Teams	30

Índice de cuadros

6.1	Lista de NOes del proyecto	15
6.2	Riesgos del proyecto	18
6.3	Cuestiones y prioridades del proyecto	20

1.

Introducción

Si he logrado ver más lejos ha sido porque he subido a hombros de gigantes.

Isaac Newton

1.1 Contexto

1.2 Motivación

1.3 Objetivos

Objetivo general

Objetivos específicos

El objetivo de un [Proyecto Fin de Grado \(PFG\)](#), [Proyecto Fin de Máster \(PFM\)](#) y [Tesis Doctoral \(TD\)](#) es una de las piezas clave a plantear, y a su vez una de las más complicadas. Se considera la **finalidad** del proyecto en cuestión a realizar y suele encajar dentro de una de las siguientes categorías:

- **Contraste** o validación de una hipótesis. Este es típico de [TDs](#), aunque algunos [PFMs](#) y (muy raramente) [PFGs](#) pueden caer dentro de esta categoría.
- **Desarrollo** o diseño de algo (e.g. Software, hardware, sistema, edificio). Suele ser el más común en la rama de la ingeniería, tanto [PFMs](#) como [PFGs](#).
- **Estudio** de un tema que deduce o descubre nuevo conocimiento. Éste suele ser más común en las ramas de las ciencias puras y humanidades, tanto [PFMs](#) como [PFGs](#).

Decimos que es una pieza clave porque sirve como primer indicador de la consecución del proyecto. Si nos planteamos un objetivo, en las conclusiones podemos indicar si se ha cumplido o no el objetivo planteado. Por eso es necesario que el objetivo esté bien definido, porque si se acepta como objetivo válido en un proyecto, y éste se concluye como cumplido, el proyecto habrá sido ejecutado correctamente.

Ahora bien, ¿cómo determinamos que el objetivo se ha cumplido? pues intentando definirlo para que se pueda cumplir, es decir, intentando que sea:

- **Acotado en el tiempo**, así es más fácil establecer un marco temporal para su realización y programar temporalmente las partes de las que se compone.
- **Medible**, para saber cómo de lejos estamos de llegar a un resultado aceptable.
- **Específico**, de manera que esté bien acotado y sea difícil embarcarse en tareas que no nos acerquen a su consecución.
- **Alcanzable**, porque si no lo es, por mucha intención y esfuerzo que le pongamos no se va a terminar.
- **Relevante**, porque si, en un **PFG** para Ingeniería del Software, desarrollamos un producto mecánico para sexar pollos, pues por muy importante que sea, poco tiene que ver con lo que se ha estudiado durante todos estos años.

Y sí, para acordarnos de cuáles son estas características podemos usar el acrónimo

1.4 Justificación

1.5 Estructura del documento

2. Marco teórico y contexto tecnológico

La ciencia amigo, está compuesta de errores; pero son errores que es útil cometer ya que nos acercan poco a poco hacia la verdad.

Julio Verne

2.1 Marco Teórico

2.2 Contexto tecnológico

En esta sección se describirán las tecnologías y herramientas más relevantes que se utilizarán a lo largo del desarrollo proyecto. El uso de las mismas se describirá en la sección [6](#).

referenciar
seccion
corres-
pondien-
te

Gestión del proyecto

Notion

Microsoft Teams

Zotero

L^AT_EXy Overleaf

Ingeniería del Software

Inyección de dependencias

Integración continua

Control de versiones

Aplicación móvil

Android

A grandes rasgos, Android es un Sistema Operativo orientado a dispositivos móviles basado en el núcleo Linux, diseñado para ser independiente de la arquitectura hardware de dichos dispositivos. Si bien originalmente fue planteado para teléfonos móviles, con el avance de la industria ha adoptado un enfoque más amplio y es compatible con más dispositivos: tabletas, relojes inteligentes, televisores, pantallas de automóviles... aunque, excepto en el caso de las tabletas, se trata de versiones basadas en Android con su propia idiosincrasia.



Figura 2.1: Logo actual de Android. Imagen extraída de [2]

Normalmente cuando nos referimos a Android, no nos referimos únicamente al sistema operativo, sino a la plataforma creada entorno al mismo; como haremos a lo largo de este proyecto. Dicha plataforma o *framework* consta de numerosas capas, siendo el sistema operativo una parte de ellas. El sistema operativo como tal es denominado AOSP o *Android Open Source Project*, siendo su código fuente público. Cualquier persona puede acceder a él, descargarlo y modificarlo [3].

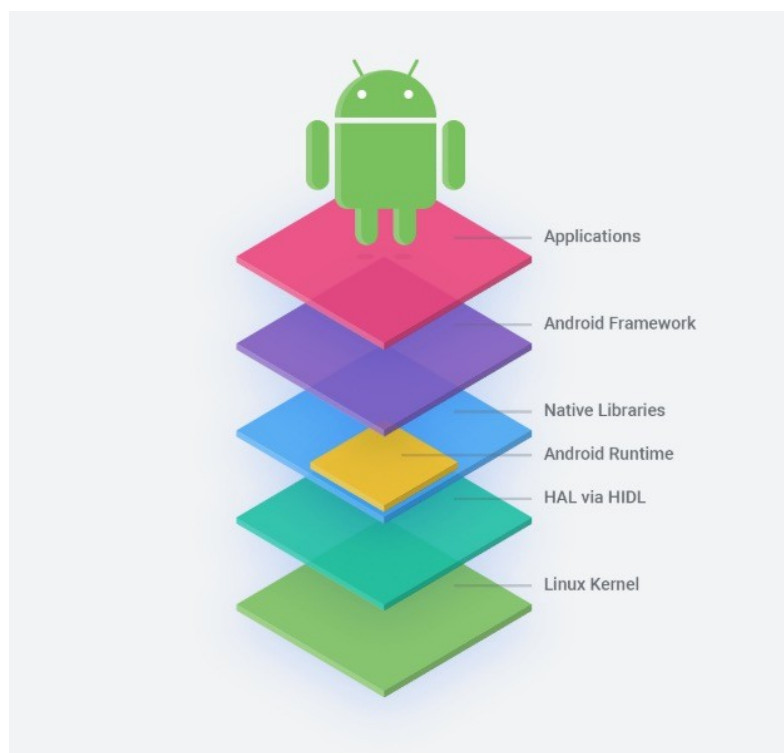


Figura 2.2: Capas de Android. Imagen extraída de [4]

No obstante, en la inmensa mayoría de los teléfonos móviles el sistema operativo es complementado con, entre otros, los GMS (*Google Mobile Services*, o servicios de Google), los cuales solo están disponibles bajo licencia; otorgada a los fabricantes que cumplen con una serie de requisitos. Los GMS se utilizan para tareas como la gestión de notificaciones, servicios de geolocalización... además de para acceder a las herramientas de Google, como la tienda de aplicaciones Play Store. Los fabricantes también pueden personalizar y añadir funciones al sistema operativo, lo que explica que dos terminales con la misma versión puedan verse tan diferentes entre sí.

Por otra parte, Android fue inicialmente desarrollado por la empresa homónima, si bien fue comprada en 2005 por Google por 50 millones de dólares. La salida del sistema operativo se produciría dos años después, el 5 de noviembre de 2007, si bien el primer terminal que lo utilizaba (HTC Dream, también conocido como T-Mobile G1) fue comercializado el 23 de septiembre de 2008

[5] [6].



Figura 2.3: HTC Dream en funcionamiento. Imagen extraída de [7]

Desde entonces, numerosas versiones de Android han sido lanzadas, siendo la última versión estable Android 13; estableciéndose por parte de Google la *costumbre* de lanzar cada año una nueva versión principal. En cada una de ellas se introducen nuevas características, pero esto no significa que todos los dispositivos puedan actualizar. Los fabricantes no están obligados a actualizar sus terminales, lo que en la práctica supone que las nuevas versiones no son utilizadas masivamente y que los programadores deben de tener en cuenta las versiones antiguas en sus aplicaciones.

Debido a que Google dejó de publicar oficialmente las estadísticas de uso de su sistema operativo, no es posible conocer con plena exactitud dichas cifras. La comunidad se ha encargado de estimar dicha información [8]; relevando que a fecha de mayo de 2023 sólo el 20% de los dispositivos tienen la última versión, mientras que las versiones 12, 11 y 10 están presentes en el 20,8%, 21,1% y 16,6% respectivamente.

Version	SDK / API level	Version code	Codename	Cumulative usage ¹	Year
Android 14 ^{DEV}	Level 34	UPSIDE_DOWN_CAKE	Upside Down Cake	—	TBD
Android 13	Level 33	TIRAMISU	Tiramisu ²	20.0%	2022
	▪ targetsdk will need to be 33+ for new apps and app updates by August 2023.				
Android 12	Level 32 Android 12L	S_V2	Snow Cone ²	40.8%	2021
	Level 31 Android 12	S			
	▪ targetsdk must be 31+ for new apps and app updates.				
Android 11	Level 30	R	Red Velvet Cake ²	61.9%	2020
Android 10	Level 29	Q	Quince Tart ²	78.5%	2019
Android 9	Level 28	P	Pie	86.6%	2018
Android 8	Level 27 Android 8.1	O_MR1	Oreo	91.3%	2017
	Level 26 Android 8.0	O		92.9%	
Android 7	Level 25 Android 7.1	N_MR1	Nougat	94.0%	2016
	Level 24 Android 7.0	N		96.2%	
Android 6	Level 23	M	Marshmallow	97.9%	2015
Android 5	Level 22 Android 5.1	LOLLIPOP_MR1	Lollipop	99.1%	2015
	Level 21 Android 5.0	LOLLIPOP, L		99.3%	2014
	▪ Jetpack Compose requires a minSdk of 21 or higher.				
Android 4	Level 20 Android 4.4W ³	KITKAT_WATCH	KitKat	No data	2013
	Level 19 Android 4.4	KITKAT			
	▪ Google Play services do not support Android versions below API level 19.				

Figura 2.4: Estadísticas acumulativas de las versiones de Android. Imagen extraída de [8]

Por último, a fecha de marzo de 2023, Android dispone de una cuota de mercado del 71% en el segmento de sistemas operativos para dispositivos móviles, teniendo su mayor rival, el sistema operativo iOS (propiedad de Apple) un 28%. Entre ambos acaparan el mercado, con un 99% de cuota de mercado. En cuanto a España, el porcentaje de Android asciende hasta el 77,73% por el 21,81 de iOS [9].

Kotlin

Durante el diseño de Android se estableció que el lenguaje principal para desarrollar aplicaciones sería Java, si bien incorpora soporte para utilizar código C y C++ [10]. No obstante, al ser Java un lenguaje interpretado sobre una máquina virtual (JVM o *Java Virtual Machine*) se abrió la puerta para que se pudieran utilizar otros lenguajes basados que usasen la JVM, como ocurriría eventualmente con Kotlin.

Jetpack Compose

Material Design 3

Salud Conectada

Room

Servidor

Python

Flask

MongoDB

3.

Estado de la cuestión

En este capítulo se introduce una revisión del estado del arte y del estado de la cuestión en lo que respecta al marco teórico del proyecto que se propone en este TFM, así como una descripción sobre los sistemas que existen en el mercado o que han sido reportados en la literatura, los cuales presentan aspectos comunes con la solución propuesta.

3.1 Análisis de la situación actual

En esta sección, se pone en valor y/o contraste la información introducida en las secciones anteriores, de tal forma que pueda realizarse un análisis del entorno en el que se desarrolla este proyecto, así como también se pueda poner de manifiesto la relevancia y la idoneidad del desarrollo propuesto.

Proyectos relacionados

Contribución de la solución propuesta

4. Metodología y materiales

En este capítulo, se introducen algunas de las principales metodologías de desarrollo *software* que actualmente más se utilizan.

4.1 Metodologías de desarrollo del *software* de Sistemas

Metodología del desarrollo seleccionada

5. Diseño del sistema propuesto

5.1 Descripción del caso de estudio

5.2 Diseño del sistema

6.

Desarrollo del sistema

6.1 Project Blastoff

descripci
del
blastoff

Inception Deck

El *Inception Deck* (también conocido como *Agile Inception*) [11] [12] es un conjunto de actividades propio de las metodologías ágiles, las cuales tienen como objetivo establecer inequívocamente los propósitos del proyecto y sus expectativas, a través de la comunicación entre todas las personas involucradas de alguna manera en el proyecto.

Más concretamente, el *Inception Deck* es un conjunto de 10 complejos ejercicios y preguntas, los cuales permiten unificar todas las visiones del producto hacia una sola, asegurando que el equipo avance en sintonía hacia una misma dirección.

Dichas dinámicas son las siguientes:

- ¿Por qué estamos aquí? (*Why are we here?*): describe los motivos principales por los que se realiza este proyecto, y cómo hemos llegado hasta este punto.
- *Elevator Pitch*: consiste en transmitir resumida y atractivamente la idea de nuestro proyecto condensada en unos 30 segundos (lo que podría durar un viaje en ascensor, de ahí el nombre) para captar la atención de otras personas. En este pequeño discurso se muestra el problema a resolver, cómo se pretende solventar y cuál es el valor diferencial que tiene nuestra idea y/o equipo, para alentar al oyente a colaborar con el mismo.
- Diseñar una caja de producto (*Design a Product Box*): obliga a imaginar nuestro proyecto encapsulado en un cartel publicitario, buscando para ello los mensajes e imágenes para promocionar el producto.

- Crear una lista de NOes (*Create a NOT list*): delimita los límites del proyecto estableciendo lo que no se va a realizar en el mismo.
- Conoce a tus vecinos (*Meet your neighbors*): esclarece las personas o equipos de los cuales depende el éxito de nuestro proyecto.
- Haz ver la solución (*Show the solution*): diseña a alto nivel la arquitectura técnica del producto, de forma legible para todas las partes del proyecto.
- ¿Qué nos quita el sueño? (*Ask what keep us up at night*): identifica los posibles riesgos a los que se enfrentará nuestro proyecto que pueden afectar a su desarrollo y éxito. Se diferencia entre los riesgos en los que podemos influir y en los que no, para decidir medidas para mitigar su posible impacto en el caso de los primeros y concienciar acerca de los segundos.
- Tómale las medidas (*Size it up*): estimación a grandes rasgos del orden de magnitud temporal del proyecto, para controlar las expectativas del mismo.
- Ser claros en qué vamos a dar (*Be clear on what's going to give*): determina las prioridades del proyecto en factores clave como el tiempo, alcance, presupuesto o calidad entre otros, junto a su flexibilidad para redimensionar el proyecto si es necesario.
- ¿Cuánto va a costar? (*Show what it's going to take*): desarrollo un presupuesto aproximado tanto en dinero como en personal y recursos si no se realizó previamente. En caso contrario, se discute la viabilidad el mismo y del proyecto.

A continuación se describe el resultado de dichas actividades para este proyecto.

Why are we here?

La Salud Mental es un área de la salud cuya visibilización aumenta cada día, pero que aún sigue recubierta de estigma y carece de recursos suficientes. Organizaciones como la Organización Mundial de la Salud y la Confederación de Salud Mental de España han realizado informes e infografías donde se presentan datos sobre este problema tan silencioso, grave y desconocido.

En nuestro país, el 6,7% de la población está afectada por la ansiedad, exactamente la misma cifra de personas con depresión. Casi la mitad de los españoles de entre 15 y 29 años (48,9%) considera que ha tenido algún problema de salud mental, mientras que más de la mitad de las personas con trastorno mental que necesitan tratamiento no lo reciben.

A nivel mundial, el 12,5% de todos los problemas de salud está representado por los trastornos mentales, una cifra mayor a la del cáncer y los problemas cardiovasculares. Más de 300 millones

de personas viven en el mundo con depresión, un problema que se ha aumentado en un 18,4% entre 2005 y 2015; mientras que 800.000 personas se suicidan cada año, siendo la segunda causa de muerte en personas de 15 a 29 años.

Con semejantes estadísticas, queda un largo camino por recorrer para que la sociedad considere a la salud mental como un área de salud igual de importante que las demás. Esto se manifiesta enormemente en la falta de atención a sus síntomas, lo que conlleva faltas crónicas de tratamiento entre la población.

Desde la Informática podemos acceder a numerosos datos, tanto de comportamiento de una persona con su móvil, como datos sobre su estado físico con la generalización de dispositivos conocidos como *wearables* (generalmente una pulsera o reloj inteligente equipado con sensores biométricos).

Bajo esta premisa se han publicado estudios [13] [14] [15] [16] que avalan que a partir de dichos datos se estimar por ejemplo si una persona tiene estrés, lo que supone un primer paso para la detección de problemas de salud mental.

Asimismo, los fabricantes de los dispositivos *wearables* ofrecen aplicaciones en las que se pueden consultar los datos recolectados, pero no ofrecen datos acerca de la salud mental, y tampoco son ofrecidos en los smartphones.

Por este motivo estamos aquí: para realizar una aplicación que aunando software y hardware pueda alertar de síntomas de problemas de salud mental, para que dicha persona sea consciente de que su salud mental puede estar deteriorándose y quizás necesite ayuda profesional.

Elevator pitch

Se estima que el 25% de las personas tendrá un trastorno mental a lo largo de su vida, y entre ellas, entre el 35% y el 50% no reciben tratamiento o no es el adecuado. Para la comunidad universitaria presentamos el Sistema para el Bienestar Emocional, una aplicación que permite obtener tu nivel de estrés teniendo en cuenta el uso del móvil y opcionalmente la información de *wearables*.

A diferencia de otras aplicaciones, no nos fijamos únicamente en el dato de un sensor invasivo ni comunicamos datos con terceras empresas, nuestro producto es una aplicación *open source*, por lo que puedes modificarla a tu antojo, que proporciona resultados más elaborados y precisos y elabora recomendaciones para tu situación mental.

Product Box

TODO

NOT list

nos falta la imagen cuando esté disponible

Dentro del alcance	Fuera del alcance
<ul style="list-style-type: none">• Desarrollo de la aplicación de usuario para dispositivos Android en el lenguaje de programación Kotlin.• Lectura de datos biométricos del usuario: ritmo cardíaco y sus variaciones, sueño, actividad física.• Obtención de datos del móvil: sueño, uso de aplicaciones.• Comunicación de resultados a los usuarios mediante notificaciones.• Visualización tanto del estado de bienestar emocional actual como de su evolución.• <i>Layout responsive</i> de la aplicación para adaptarla a todo tipo de dispositivos.• Desarrollo del modo oscuro de la interfaz de la aplicación.• Conexión de la aplicación con el modelo de Inteligencia Artificial.• Aporte de los datos de los usuarios para entrenar el modelo de Inteligencia Artificial.• Validación mediante tests con cuestionarios para corroborar los resultados obtenidos.• Realización profesional de la gestión de proyecto siguiendo metodologías ágiles.	<ul style="list-style-type: none">• Desarrollo de la aplicación para terminales con otro sistema operativo, como iOS.• Retro-aprendizaje del modelo de Inteligencia Artificial para un usuario en particular.• Evolución del modelo de Inteligencia Artificial tras la prueba piloto.• Utilización de mecanismos de <i>edge computing</i> en la aplicación de usuario.• Implementación de cuentas de usuario para mantener los datos entre dispositivos.• Realización de ingeniería inversa a las pulseras Xiaomi desde cero.• Inclusión de otras pulseras que no dispongan del sistema operativo Wear OS.• Uso de la herramienta Google Fit para obtener datos de dispositivos.• Lectura de sensores de estrés y oxígeno en sangre por ser invasivos de cara al usuario.• Utilización de escáneres cerebrales u otros dispositivos similares.
No resuelto	

Cuadro 6.1: Lista de NOes del proyecto

Meet your neighbors

- Estudiantes de la universidad como usuarios del sistema.
- Profesores de la universidad como usuarios y/o promotores del sistema.
- Responsables de la infraestructura de la universidad para poder entrenar el modelo de Inteligencia Artificial.
- Fabricantes tanto de los *wearables* como de los smartphones.
- Desarrolladores del S.O. Android y sus API oficiales.
- Futuros desarrolladores del proyecto, como estudiantes de la escuela para sus proyectos Fin de Titulación; bien a nivel de aplicación o de Inteligencia Artificial.
- Subdirección de Asuntos Económicos de la ETSISI como responsables de la compra de material hardware para este proyecto.
- Consultores internos sobre desarrollo de aplicaciones móviles, ingeniería de software e integración de componentes hardware para transmitir los conocimientos necesarios para continuar el proyecto/resultados del mismo.

Show the solution

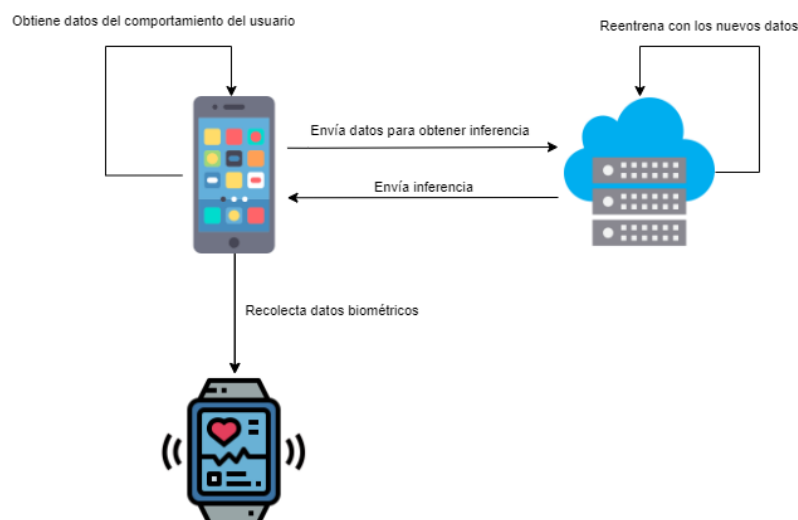


Figura 6.1: Componentes básicos del sistema



Figura 6.2: Pila con la arquitectura del sistema

Up at night

Riesgos en los que podemos influir	Riesgos en los que no podemos influir
<ul style="list-style-type: none"> • Realización de pruebas exhaustivas tanto del <i>backend</i> como de la interfaz gráfica de usuario. • Participación de voluntarios que aporten datos al estudio con los que mejorar la precisión y evitar sesgos. • Elaboración de unos términos y condiciones legales acerca de los datos obtenidos en el proyecto. • Accesibilidad correcta de la aplicación. • Elaboración de un diseño coherente a nivel UX/UI de la aplicación. • Comunicación y difusión eficaz del proyecto en la escuela, en la defensa de este TFM y en redes sociales. • Continuidad del proyecto después del presente TFM para, entre otros, publicar una aplicación en iOS. • Definición clara de la arquitectura y de los procesos de mantenimiento y ampliación de la aplicación. • Mantenimiento futuro de la aplicación con las nuevas versiones de Android. • Potencia de cómputo suficiente para entrenar el modelo de Inteligencia Artificial 	<ul style="list-style-type: none"> • Problemas de compatibilidad entre las versiones de Android. • Muestra de estudiantes variada y contrastada participando en el proyecto. • Soportes de los fabricantes de <i>wearables</i> (en especial Samsung) a Health Connect. • Funcionamiento de las pulseras Xiaomi en aplicaciones no oficiales. • Disponibilidad de los datos del móvil y de los <i>wearables</i>. • Fiabilidad suficiente de los datos obtenidos por el móvil y los <i>wearables</i>. • Incumplimiento de la planificación por circunstancias ajenas al proyecto. • Retrasos en la entrega del hardware necesario para el sistema. • Ausencia de fallos al publicar la aplicación en la Google Play Store. • Precisión insuficiente del modelo de Inteligencia Artificial para ser utilizado en un entorno real.

Cuadro 6.2: Riesgos del proyecto

Size it up

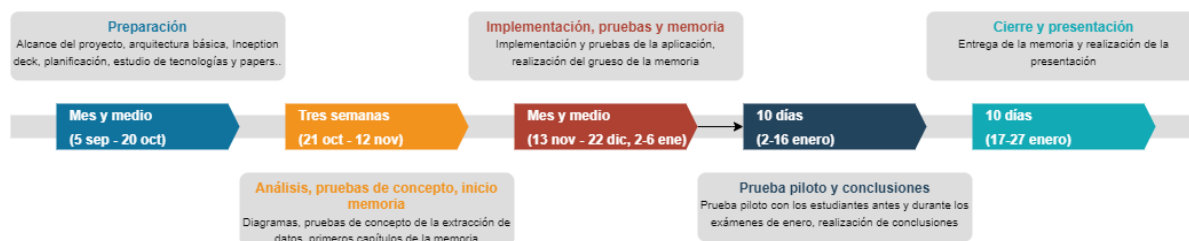


Figura 6.3: Dimensionamiento aproximado del proyecto

What's going to give

Aspecto	Importancia	Implicaciones
Detalle de la documentación	Media	Se desea que la memoria explique detalladamente las cuestiones importantes del proyecto y cubra todo lo realizado en el mismo, pero no es necesario en detalles menores ni se pretende que la memoria sea excesivamente larga.
Alcance de la implementación	Baja - media	Una vez realizada la conexión con las API de Android y con el hardware concedido, se puede establecer flexibilidad con el alcance del prototipo.
Calidad	Media - alta	Se deben realizar pruebas detalladas y obtener métricas de calidad del software implementado, pero se permite cierta flexibilidad en las mismas sobre el software que maneje el hardware del proyecto por su alta dificultad.
Comunicación del proyecto	Alta	Se necesitan usuarios para la prueba piloto y para obtener retroalimentación del proyecto. No obstante, si bien se desea que el proyecto sea continuado, no es estrictamente necesario.

Experiencia de usuario	Media	La aplicación debe ser intuitiva y accesible para el usuario, pero no se necesita que el prototipo tenga un diseño visual (elementos, animaciones, gráficas) muy elaborado y detallado al ser un prototipo.
Fiabilidad	Media	No están previstas pruebas exhaustivas con todos los dispositivos compatibles (versiones del SO, dispositivos hardware), por lo que <i>bugs</i> leves son admisibles.
Presupuesto (hardware)	Muy alta	El presupuesto está cerrado al aprobado, por lo que no se pueden incorporar nuevos componentes.
Rendimiento	Baja - media	La optimización del software queda fuera del proyecto, si bien se debe de garantizar un rendimiento mínimo para no afectar la experiencia de usuario.
Seguridad	Alta	Si bien los datos del móvil no son personales, los de los <i>wearables</i> sí lo son, por lo que deben ser debidamente protegidos tanto en su almacenamiento como en su transmisión con técnicas estándar.
Tiempo	Muy alta	El máster proporciona dos convocatorias fijas en el calendario, y por la situación personal del autor solo puede acudir a la de enero, por lo que el proyecto debe finalizar para esa fecha.

Cuadro 6.3: Cuestiones y prioridades del proyecto

What it's going to take

En el apartado de Tómale las medidas se establecieron las etapas del proyecto, por lo que ya conocemos el tiempo estimado de desarrollo del proyecto: cuatro meses y medio. Asimismo, al ser un Trabajo Final de Máster realizado por una persona se necesita a un único ingeniero para el proyecto.

Durante ese plazo las labores del desarrollador incluyen (pero no se limitan a):

- Análisis y planteamiento del problema a resolver.
- Diseño de un sistema que combine hardware y software.
- Implementación de una aplicación Android con el lenguaje de programación Kotlin.
- Diseño de interfaces de usuario y experiencia de usuario.
- Comunicación de smartphones Android con dispositivos *wearables* mediante Bluetooth.
- Realización de pruebas unitarias, funcionales y de integración.
- Uso de sistemas de control de versiones, como Github.
- Desarrollo de un pipeline CI/CD.
- Aplicación de metodologías ágiles.

No obstante, son labores que puede desarrollar un perfil con formación universitaria en Informática, ya que al ser un proyecto relativamente breve no se necesita de personal especializado en análisis, pruebas, etc.

Por tanto, se supone que dicho perfil es un Android Developer y que su salario es el promedio. Según la conocida web de empleo Glassdoor, dicho salario medio en Madrid es de 33.000 €/año, por lo que su salario prorrateado es de 12.375€. Asimismo, hay que sumar el presupuesto concedido por la universidad, que asciende a 460€.

Por tanto, la estimación del coste total del proyecto es de 12.835€.

7.

Resultados

¡Los números Mason! ¿Qué significan?

Jason Hudson

7.1 Preparación del entorno

En este apartado se enumeran los pasos seguidos para la preparación del entorno que ha sido utilizado en el desarrollo del sistema.

Sistema Operativo

Tecnología 1

Tecnología ...

Tecnología N

SectionImplementación del caso de estudio

7.2 Resultados del caso de estudio

En este apartado se muestran algunos ejemplos de los utilización y resultados que podrían obtenerse.

8.

Impacto social y medioambiental

En este capítulo se recogen los beneficios que la implantación del proyecto desarrollado podría generar tanto a nivel medioambiental como su impacto social.

8.1 Contexto medioambiental

Los Objetivos de Desarrollo Sostenible (ODS)....

8.2 Aspectos éticos, sociales y económicos

9.

Presupuesto

10.

Conclusiones

11.

Líneas futuras

12.1 Setup del proyecto

CI/CD

CI/CD es un concepto de Ingeniería de Software, relacionado estrechamente con la filosofía *DevOps* y con las metodologías ágiles; en el que se implementan flujos de trabajo que permitan distribuir las aplicaciones a los clientes de forma rápida, continua y fiable, a través de la automatización de ciertas etapas del desarrollo software.

El nombre es una combinación de dos siglas, las cuales tienen diferentes significados: CI hace referencia a *Continuous Integration* (integración continua), mientras que CD puede indicar dos procesos distintos: bien *Continuous Delivery* (entrega continua) o *Continuous Deployment* (despliegue continuo), los cuales en ocasiones se utilizan indistintamente.

Ahondando en dichos conceptos, la integración continua es un proceso por el cual se establece un único repositorio central, el cual contiene tanto el código del producto software como sus pruebas unitarias y de integración. Tras ello se establece una capa de automatización que permite realizar la integración, construcción y pruebas del software en cada actualización del repositorio.

Por otra parte, la entrega y despliegue continuo tienen en común la automatización de las etapas de puesta en marcha del nuevo software. No obstante, la entrega continua aboga por un despliegue manual de los artefactos al no necesitarse o no ser beneficioso cambios frecuentes en el entorno de producción, mientras que el despliegue continuo automatiza dicho despliegue. Por tanto, el uso de un proceso u otro reside en las necesidades del proyecto.

La implementación de una infraestructura CI/CD permite a los desarrolladores entregar cambios de forma rápida al reducir enormemente el tiempo de integración y despliegue, incrementando la

eficiencia y eficacia en la implementación de software.

Asimismo, permite a los miembros del equipo trabajar de forma simultánea en distintas partes del software, sin tener que parar para la integración de todos los cambios y resolver los conflictos que puedan aparecer.

12.2 Sprint #0

Este sprint se desarrolló entre el 5 y el 21 de septiembre, y consistió en la selección y puesta en marcha de las herramientas que se utilizarían para el desarrollo de este proyecto.

Notion

Notion es una plataforma online parcialmente gratuita utilizada para el seguimiento de proyectos y de equipos, la cual es fácilmente personalizable a partir de una serie de elementos predefinidos, entre los que se encuentran bases de datos embebidas (y vistas visuales como las mismas, como tableros Kanban o diagramas Gantt), soporte para imágenes, vídeo y audio, integración con otras plataformas como Slack, Zoom, Twitter...

En esta herramienta se pueden construir páginas, las cuales pueden contener texto formateado con el lenguaje de marcado Markdown o cualquiera de los bloques anteriores, por lo que fue elegida para conformar el centro neurálgico del seguimiento del proyecto.

Se implementaron los siguientes elementos

- Tablero Kanban. Aquí son almacenadas las tareas del proyecto, con atributos sobre su estado, su categoría y sus fechas de desarrollo.
- Diagrama Gantt. En esencia es una vista a la misma base de datos que la del tablero Kanban.
- Base de datos de referencias con todos los link de interés para la implementación del proyecto. Algunos de ellos aparecen en la bibliografía de esta memoria por su relevancia para exponer ciertas cuestiones.
- Base de datos con apuntes de consulta. Aquí se alojan descripciones de conceptos de utilidad para el proyecto. Está enlazada con la base de datos de referencia para acceder rápidamente a los enlaces para ampliar esa información.

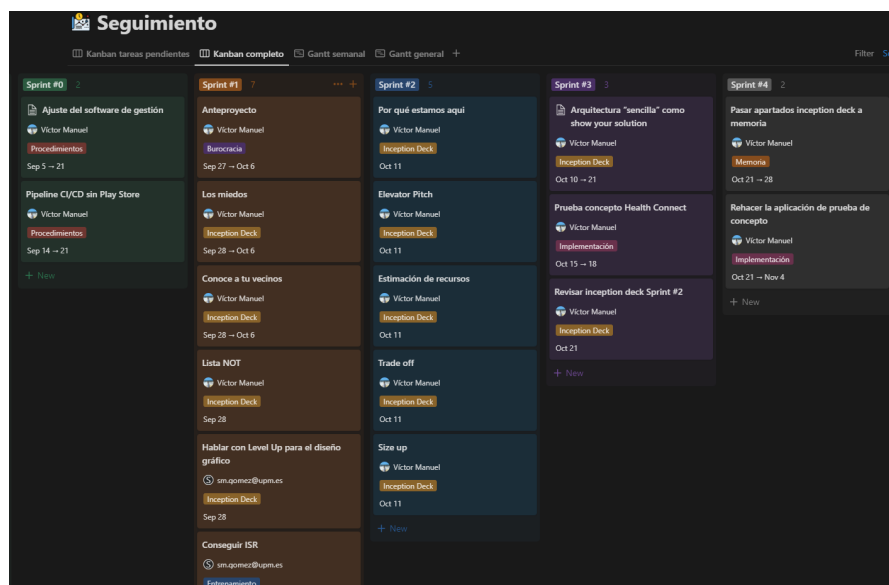


Figura 12.1: Extracto del tablero Kanban del proyecto

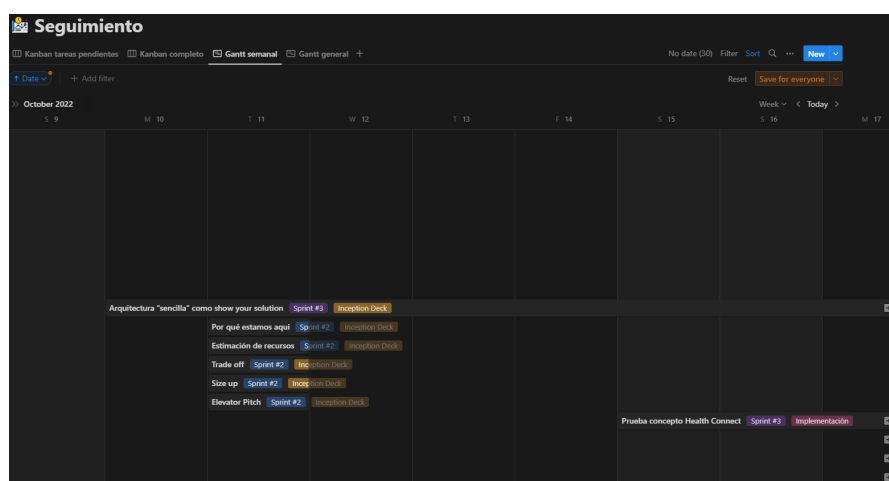


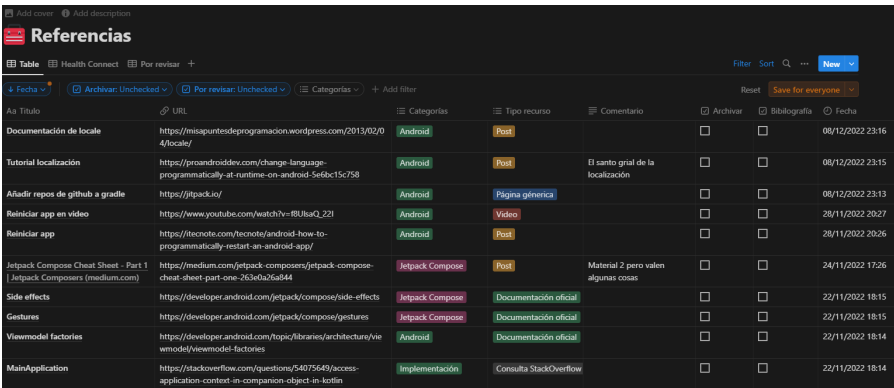
Figura 12.2: Extracto del diagrama Gantt del proyecto

Microsoft Teams

Microsoft Teams es una herramienta de comunicación y colaboración para equipos propiedad de Microsoft. Es principalmente utilizada para alojar canales de comunicación y videollamadas.

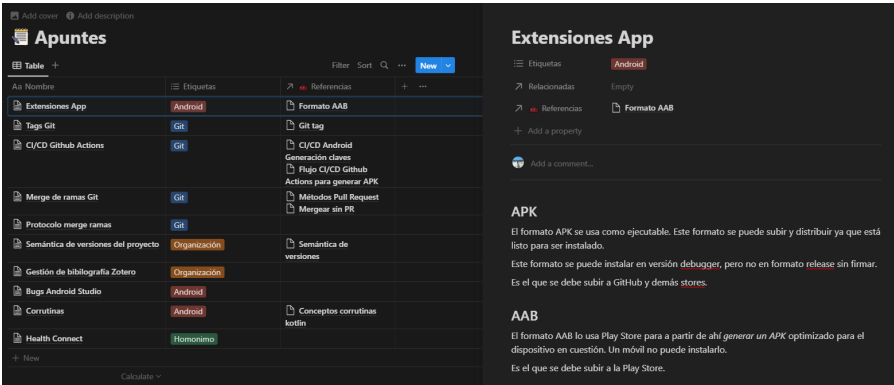
Para este proyecto lo hemos seleccionado por dos de sus funcionalidades: desplegar un sistema de archivos en la nube para cada canal de comunicación y registro de turnos de trabajo.

En cuanto a la primera de ellas, la hemos elegido por delante de otras opciones de alojamiento ya que por ser usuarios de la UPM, disponemos de 1 TB de almacenamiento, por lo que pode-



As	Título	URL	Categorías	Tipo recurso	Comentario	Archivar	Bibliografía	Fecha
	Documentación de locale	https://misapuntedesprogramacion.wordpress.com/2013/02/04/locale/	Android	Post		<input type="checkbox"/>	<input type="checkbox"/>	08/12/2022 23:16
	Tutorial localización	https://proandroiddev.com/change-language-programmatically-at-runtime-on-android-5e6bc15c758	Android	Post	El santo grail de la localización	<input type="checkbox"/>	<input type="checkbox"/>	08/12/2022 23:15
	Añadir repos de github a gradle	https://itpack.io/	Android	Página generica		<input type="checkbox"/>	<input type="checkbox"/>	08/12/2022 23:13
	Reiniciar app en video	https://www.youtube.com/watch?v=8UJkaQ_22I	Android	Video		<input type="checkbox"/>	<input type="checkbox"/>	28/11/2022 20:27
	Reiniciar app	https://tccote.com/tccote/android-how-to-programmatically-restart-an-android-app/	Android	Post		<input type="checkbox"/>	<input type="checkbox"/>	28/11/2022 20:26
	Jetpack Compose Cheat Sheet - Part 1 [Jetpack Compose] (medium.com)	https://medium.com/jetpack-compose/jetpack-compose-cheat-sheet-part-one-263a9a2a844	Jetpack Compose	Post	Material 2 pero valen algunas cosas	<input type="checkbox"/>	<input type="checkbox"/>	24/11/2022 17:36
	Side effects	https://developer.android.com/jetpack/compose/side-effects	Jetpack Compose	Documentación oficial		<input type="checkbox"/>	<input type="checkbox"/>	22/11/2022 18:15
	Gestures	https://developer.android.com/jetpack/compose/gestures	Jetpack Compose	Documentación oficial		<input type="checkbox"/>	<input type="checkbox"/>	22/11/2022 18:15
	Viewmodel factories	https://developer.android.com/topic/libraries/architecture/viewmodel/viewmodel-factories	Android	Documentación oficial		<input type="checkbox"/>	<input type="checkbox"/>	22/11/2022 18:14
	MainApplication	https://stackoverflow.com/questions/54075649/access-application-context-in-companion-object-in-kotlin	Implementación	Consulta StackOverflow		<input type="checkbox"/>	<input type="checkbox"/>	22/11/2022 18:14

Figura 12.3: Extracto de la base de datos de referencias



As	Nombre	Etiquetas	Referencias
	Extensiones App	Android	Formato AAB
	Tags Git	Git	Git tag
	CI/CD Github Actions	Git	CI/CD Android Generación claves Flujo CI/CD Github Actions para generar APK Métodos Pull Request Mergear sin PR
	Merge de ramas Git	Git	
	Protocolo merge ramas	Git	
	Semántica de versiones del proyecto	Organización	Semántica de versiones
	Gestión de bibliografía Zotero	Organización	
	Bugs Android Studio	Android	
	Corrutinas	Android	Conceptos corrutinas kotlin
	Health Connect	Homomimo	

Extensiones App

Etiquetas: Android

Relacionadas: Empty

Referencias: Formato AAB

+ Add a property

+ Add a comment...

APK

El formato APK se usa como ejecutable. Este formato se puede subir y distribuir ya que está listo para ser instalado.

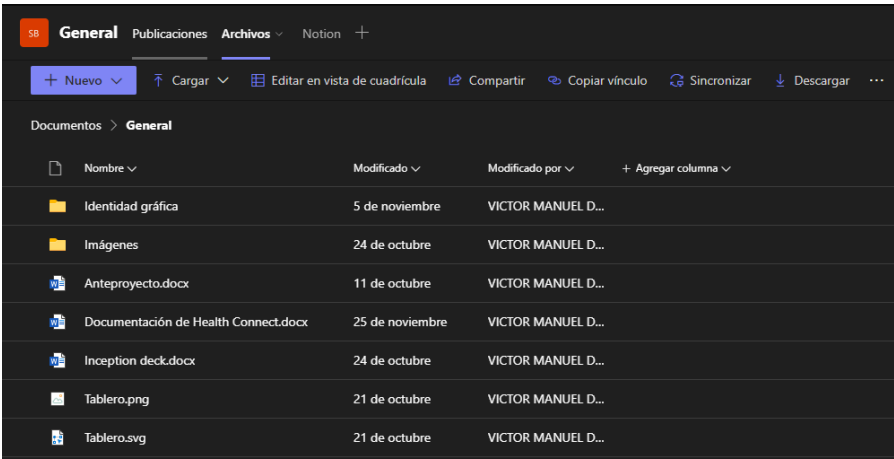
Este formato se puede instalar en versión debugger, pero no en formato release sin firmar. Es el que se debe subir a GitHub y demás stores.

AAB

El formato AAB lo usa Play Store para a partir de ahí generar un APK optimizado para el dispositivo en cuestión. Un módul no puede instalarlo. Es el que se debe subir a la Play Store.

Figura 12.4: Extracto de la base de datos de apuntes del proyecto

mos alojar todo tipo de contenido multimedia prácticamente sin restricciones y de forma fácil y sencilla.



Nombre	Modificado	Modificado por
Identidad gráfica	5 de noviembre	VICTOR MANUEL D...
Imágenes	24 de octubre	VICTOR MANUEL D...
Anteproyecto.docx	11 de octubre	VICTOR MANUEL D...
Documentación de Health Connect.docx	25 de noviembre	VICTOR MANUEL D...
Inception deck.docx	24 de octubre	VICTOR MANUEL D...
Tablero.png	21 de octubre	VICTOR MANUEL D...
Tablero.svg	21 de octubre	VICTOR MANUEL D...

Figura 12.5: Extracto de los documentos alojados en Teams

Asimismo, la funcionalidad de registro de turnos es cómoda de usar y nos genera automáticamente

un excel con las horas registradas por parte de todos los miembros del equipo, ideal si queremos registrar el total de horas empleadas en el proyecto.

Zotero

Zotero es una plataforma open-source (licenciado con AGPL) de gestión de referencias y citas bibliográficas, que permite recopilar y almacenar referencias de libros, artículos, sitios web y otros documentos cómodamente. Una vez registradas, permite exportar dichas citas de acuerdo a diferentes estilos de citación.

En este proyecto ha sido seleccionada por su facilidad de uso, por ser open-source y disponer de una gran cantidad de plugins, como *Zotero Connector*, que nos permite agregar referencias desde el navegador.

Git y Github

Git es sistema de control de versiones open-source (bajo la licencia GNU GPL v2) diseñado por Linus Torvalds, siendo muy popular entre los equipos de desarrollo software. Si bien Git almacena y gestiona cambios en archivos de cualquier tipo, está pensado para ser más eficiente en los archivos de código fuente.

Por otra parte, Github es una plataforma en línea que ofrece alojamiento y gestión de proyectos que utilicen Git. Para este proyecto, además del alojamiento, utilizamos sus herramientas de CI/CD (GitHub Actions), por la que a través de un fichero .yaml podemos configurar un pipeline de acciones con las que realizar los procesos que queramos automáticamente. Además, pueden establecerse flujos de trabajos por ramas de desarrollo u otras condiciones, tales como formatos de tags, tipos de eventos...

L^AT_EXy Overleaf

L^AT_EX como medio de creación de la presente memoria y Overleaf como editor de dicho lenguaje.

12.3 Tecnologías

Jetpack Compose

Jetpack Compose es el conjunto de herramientas *modernas* de Android para el desarrollo de interfaces gráficas, lanzado en su primera versión estable el 28 de julio de 2021 por Google. Este kit de librerías permite desarrollar en el ecosistema Android de forma nativa interfaces gráficas de manera declarativa como en los sistemas React, Flutter o SwiftUI, siguiendo las tendencias actuales de la industria en el desarrollo de aplicaciones móviles.

Hasta la aparición de Jetpack Compose, el desarrollo interfaces gráficas nativas en Android se realizaba con el enfoque conocido como programación imperativa. En este tipo de desarrollo es necesario especificar paso por paso cómo se va a construir dicha interfaz gráfica. En dicho proceso (conocido en Android como sistema de vistas) se codificaba un fichero XML, en el que se describían todos los elementos gráficos (botones, textos...); para en el código Java (o Kotlin) de la aplicación se accediera a dichos elementos y se le aplicaran modificaciones y transformaciones de manera totalmente manual.

En Jetpack Compose la interfaz está interconectada a un estado, por lo que si dicho estado cambia, nuestra interfaz gráfica cambiará para mostrar el nuevo estado, simplificando enormemente el desarrollo y reduciendo el código necesario. Asimismo, a diferencia del sistema anterior, está desacoplada del sistema operativo, por lo que no dependemos de la versión del terminal para ejecutar correctamente nuestra interfaz gráfica, como si ocurría anteriormente. Por último, Jetpack Compose es compatible con los componentes XML del sistema anterior, lo que facilita la migración de los proyectos antiguos a este nuevo paradigma.

No obstante, este conjunto de herramienta solo es compatible con el lenguaje de programación Kotlin, por lo que en algunos casos puede incrementarse la curva de dificultad para aprender a utilizarlo.

Para crear nuestras interfaces en Jetpack Compose crearemos funciones con la anotación `Composable`, la cual nos permite crear elementos gráficos dentro de su contenido, tales como botones, textos, imágenes..

Para poder ver cómo va quedando nuestra interfaz gráfica tenemos que recurrir a las vistas previas. Estas funciones llevan la anotación `Composable` y una nueva conocida como `Preview`, la cual

permite que se pueda visualizar su contenido de manera independiente a cambio de imponer una serie de restricciones. En esas funciones Preview llamamos a la función Composable que queremos visualizar.

12.4 Desarrollo

Inyección de dependencias (Dagger Hilt)

Cifrado de la base de datos (SQLCipher)

Navegación entre ventanas (Compose Destination)

Planificación de tareas (Work Manager)

Comunicación con el servidor (Retrofit)

Ajustes del usuario (Datastore)

Localización de la app (Strings.xml y Lingliver)

Onboarding (Lottie)

Gráficas (Vico)

Flujos de integración continua (GitHub Actions)

Bibliografía

- [1] A. Salamanca, *La geopolítica de los papers: conocimiento libre contra la millonaria industria de las revistas académicas*, es, mar. de 2023. dirección: <https://elordenmundial.com/sci-hub-revistas-academicas-lucha-guerrillera-industria-millonaria/> (visitado 24-06-2023).
- [2] G. L. VulcanSphere vectorised by CMetalCore and optimised by, *English: The Android logo as of 2019*, ago. de 2019. dirección: [https://commons.wikimedia.org/wiki/File:Android_logo_2019_\(stacked\).svg](https://commons.wikimedia.org/wiki/File:Android_logo_2019_(stacked).svg) (visitado 26-06-2023).
- [3] C. Collado, *Qué es AOSP: así funciona el Android sin Google*, es, mayo de 2022. dirección: <https://www.lavanguardia.com/andro4all/android/que-es-aosp-asi-funciona-el-android-sin-google> (visitado 26-06-2023).
- [4] E. Pérez, *AOSP: así es el Android 'open source' sin Google que queda como opción para Huawei*, es, Section: aplicaciones, mayo de 2019. dirección: <https://www.xataka.com/aplicaciones/aosp-asi-al-android-open-source-google-que-queda-como-opcion-para-huawei> (visitado 26-06-2023).
- [5] R. Adeva, *Android: qué es, versiones, aplicaciones y cómo saber la versión instalada*, es, feb. de 2023. dirección: <https://www.adslzone.net/reportajes/software/que-es-android/> (visitado 26-06-2023).
- [6] J. Marquez, *Así era el HTC Dream, el primer teléfono de la historia con Android (y sí, tenía teclado QWERTY)*, es, Section: móviles, mayo de 2022. dirección: <https://www.xataka.com/moviles/asi-era-htc-dream-primer-telefono-historia-android-tenia-teclado-qwerty> (visitado 26-06-2023).
- [7] M. Oryl, *T-Mobile G1 Launch Event*, sep. de 2008. dirección: https://commons.wikimedia.org/wiki/File:T-Mobile_G1_launch_event_2.jpg (visitado 26-06-2023).
- [8] E. Belinski, *Android API Levels*. dirección: <https://apilevels.com/> (visitado 26-06-2023).

- [9] E. Press, *Así se reparten Android e iOS el mercado global de sistemas operativos móviles*, (SCHEME=ISO639) es, Publisher: Europa Press, abr. de 2023. dirección: <https://www.europapress.es/portaltic/software/noticia-asi-reparten-android-ios-mercado-global-sistemas-operativos-moviles-20230404123515.html> (visitado 26-06-2023).
- [10] A. Developers, *Cómo agregar código C y C++ a un proyecto / Android Studio*, es-419. dirección: <https://developer.android.com/studio/projects/add-native-code?hl=es-419> (visitado 26-06-2023).
- [11] J. Rasmusson, *The Agile Samurai*, English. Dallas: The Pragmatic Bookshelf, 2010, ISBN: 978-1-934356-58-6. (visitado 24-10-2022).
- [12] M. López Mendoza, *Agile Inception: Qué es y cómo ejecutarlo / OpenWebinars*, abr. de 2021. dirección: <https://openwebinars.net/blog/agile-inception-que-es-y-como-ejecutarlo/> (visitado 24-10-2022).
- [13] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger y K. Van Laerhoven, «Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection», *ICMI'18*, oct. de 2018.
- [14] M. Boukhechba, A. R. Daros, K. Fua, P. I. Chow, B. A. Teachman y L. E. Barnes, «DemonicSalmon: Monitoring mental health and social interactions of college students using smartphones», *Smart Health*, 9-10, 2018.
- [15] B. A. Hickey, T. Chalmers, P. Newton et al., «Smart Devices and Wearable Technologies to Detect and Monitor Mental Health Conditions and Stress: A Systematic Review», *Sensors*, 2021.
- [16] W. Rui, C. Fanglin, C. Zhenyu et al., «StudentLife: Assessing Mental Health, Academic Performance and Behavioral Trends of College Students using Smartphones», *Ubicomp*, 2014.

