

UNIVERSIDAD POLITÉCNICA DE MADRID

**E.T.S. DE INGENIERÍA DE SISTEMAS INFORMÁTICOS**

PROYECTO FIN DE MÁSTER

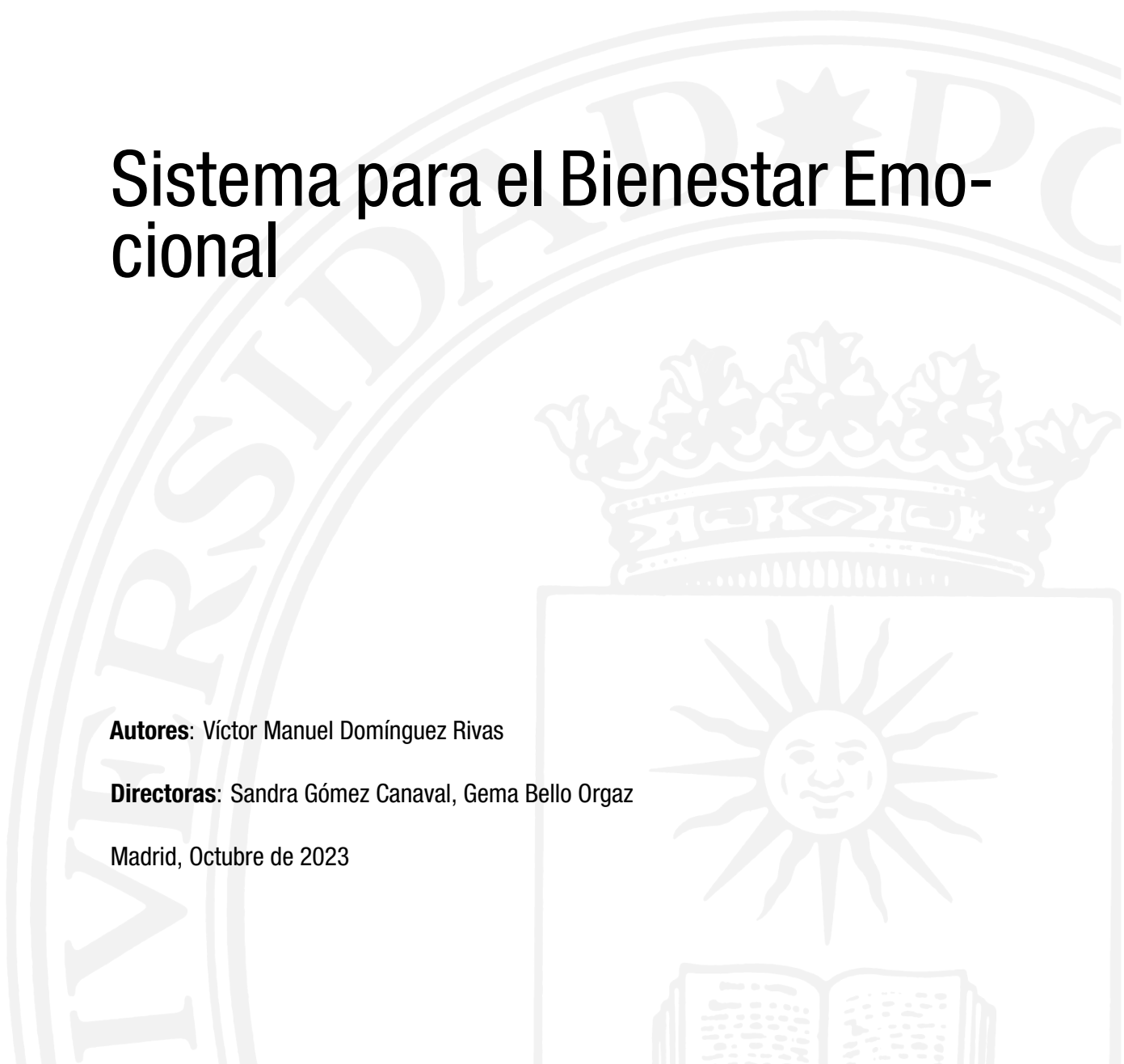
**MÁSTER UNIVERSITARIO EN SOFTWARE DE SISTEMAS DISTRIBUIDOS Y  
EMPOTRADOS**

# Sistema para el Bienestar Emocional

**Autores:** Víctor Manuel Domínguez Rivas

**Directoras:** Sandra Gómez Canaval, Gema Bello Orgaz

Madrid, Octubre de 2023



Víctor Manuel Domínguez Rivas

*Sistema para el Bienestar Emocional*

Proyecto Fin de Máster, 12 de julio de 2023

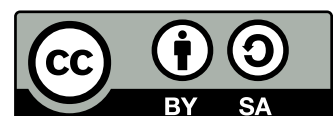
**Directoras:** Sandra Gómez Canaval, Gema Bello Orgaz

**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	iv
Índice de cuadros	v
<b>1 Introducción</b>	<b>1</b>
1.1 Contexto . . . . .	1
1.2 Motivación . . . . .	1
1.3 Justificación . . . . .	1
1.4 Objetivos . . . . .	1
1.5 Estructura del documento . . . . .	3
<b>2 Marco teórico y contexto tecnológico</b>	<b>4</b>
2.1 Marco Teórico . . . . .	4
2.2 Contexto tecnológico . . . . .	4
<b>3 Estado del arte</b>	<b>29</b>
3.1 Análisis de la situación actual . . . . .	29
<b>4 Metodología y materiales</b>	<b>30</b>
4.1 Metodologías de desarrollo del <i>software</i> de Sistemas . . . . .	30
<b>5 Análisis del sistema propuesto</b>	<b>31</b>
5.1 Análisis de Stakeholders . . . . .	31
5.2 Requisitos de usuario . . . . .	31
5.3 Requisitos funcionales . . . . .	31
5.4 Requisitos no funcionales . . . . .	31

<b>6</b>	<b>Diseño del sistema propuesto</b>	<b>32</b>
6.1	Descripción del caso de estudio . . . . .	32
6.2	Diseño del sistema . . . . .	32
<b>7</b>	<b>Desarrollo del sistema</b>	<b>33</b>
7.1	Project Blastoff . . . . .	33
7.2	Setup del proyecto . . . . .	44
7.3	Implementación de la aplicación móvil . . . . .	44
7.4	Implementación de la API del servidor . . . . .	44
<b>8</b>	<b>Pruebas del sistema</b>	<b>45</b>
8.1	Aplicación móvil . . . . .	45
8.2	API del servidor . . . . .	45
<b>9</b>	<b>Resultados</b>	<b>46</b>
9.1	Preparación del entorno . . . . .	46
9.2	Resultados del caso de estudio . . . . .	46
<b>10</b>	<b>Impacto social y medioambiental</b>	<b>47</b>
10.1	Contexto medioambiental . . . . .	47
10.2	Aspectos éticos, sociales y económicos . . . . .	47
<b>11</b>	<b>Presupuesto</b>	<b>48</b>
<b>12</b>	<b>Conclusiones</b>	<b>49</b>
12.1	Conclusiones técnicas . . . . .	49
12.2	Conocimientos adquiridos . . . . .	49
12.3	Reflexión final . . . . .	49
<b>13</b>	<b>Líneas futuras</b>	<b>50</b>
<b>14</b>	<b>Temporal</b>	<b>52</b>
14.1	Setup del proyecto . . . . .	52
14.2	Sprint #0 . . . . .	52
	<b>Bibliografía</b>	<b>53</b>
<b>A</b>	<b>Cuestionarios para el seguimiento diario</b>	<b>60</b>
A.1	Inicio del día . . . . .	60
A.2	Final del día . . . . .	61



<b>B</b>	<b>Recomendaciones</b>	<b>64</b>
B.1	Ansiedad . . . . .	64
B.2	Depresión . . . . .	66
B.3	Riesgo de suicidio . . . . .	67
B.4	Soledad . . . . .	68

# Índice de figuras

---

2.1	Extracto de los documentos alojados en Teams . . . . .	5
2.2	Extracto del registro de horas . . . . .	6
2.3	Logo actual de Android. . . . .	7
2.4	Capas de Android. . . . .	8
2.5	HTC Dream en funcionamiento. . . . .	9
2.6	Estadísticas acumulativas de las versiones de Android . . . . .	10
2.7	Logo de Kotlin. . . . .	11
2.8	Ejemplo de uso de la herramienta <i>Material Design Builder</i> . . . . .	13
2.9	Categorías de pantalla según anchura . . . . .	14
2.10	Algunos elementos gráficos de Material Design 3 . . . . .	14
2.11	Logo de Salud Conectada. . . . .	15
2.12	Arquitectura básica de Salud Conectada. . . . .	16
2.13	Visualización de datos de distancia. . . . .	17
2.14	Visualización del acceso a los datos. . . . .	18
2.15	Granularidad de los permisos. . . . .	18
2.16	Vista previa de una animación personalizada . . . . .	24
2.17	Ejemplo de gráfica de líneas con Vico . . . . .	25
2.18	Ejemplo de gráfica de barras con Vico . . . . .	25
2.19	Logo de Python. . . . .	26
2.20	Logo de MongoDB. . . . .	27
7.1	Componentes básicos del sistema . . . . .	38
7.2	Pila con la arquitectura del sistema . . . . .	39
7.3	Dimensionamiento aproximado del proyecto . . . . .	41

# Índice de cuadros

---

7.1	Lista de NOes del proyecto . . . . .	37
7.2	Riesgos del proyecto . . . . .	40
7.3	Cuestiones y prioridades del proyecto . . . . .	42

# 1.

# Introducción

---

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

Isaac Newton

El proyecto desarrollado consiste, a grandes rasgos, en sistema que permita al usuario conocer su estado mental, en particular de tres variables: estrés, depresión y soledad. Para

Un párrafo o dos como los de Clara

## 1.1 Contexto

Aquí vendemos la motito del tema en cuestión

## 1.2 Motivación

## 1.3 Justificación

Por qué es importante para el mundo (más a nivel político, stats)

## 1.4 Objetivos

### Objetivo general

El objetivo de todo el proyecto

## Objetivos específicos

Los objetivos como tal

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 PFM 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 PFM 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 PFM 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

## 1.5 Estructura del documento

La

## 2. Marco teórico y contexto tecnológico

---

*Una vez que algo es una pasión, hay motivación.*

Michael Schumacher

### 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 7.

referenciar seccion correspondiente

## Gestión del proyecto

### Notion

Notion es una plataforma online parcialmente gratuita, utilizada para el seguimiento de proyectos y de equipos. Esta herramienta 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 y documentación del proyecto, como se expondrá en la sección de Metodología.

meter aqui la ref

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

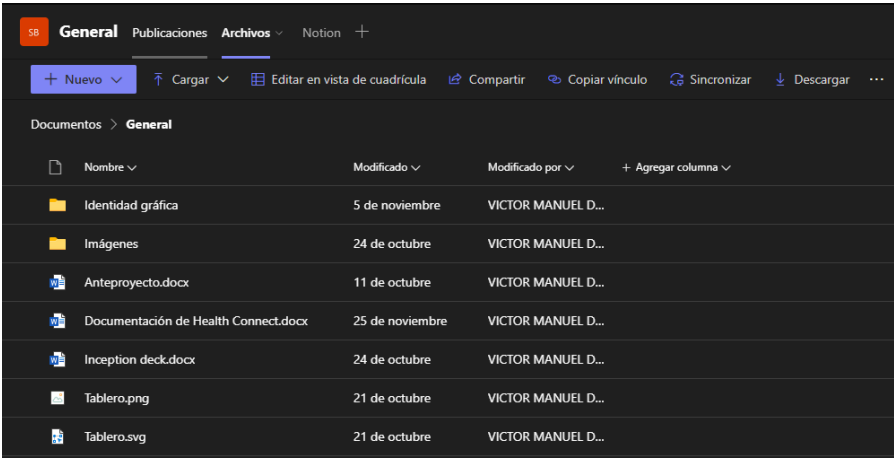


Figura 2.1: Extracto de los documentos alojados en Teams

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



	A	B	C	D	E
1	Leyenda	Fecha	Nombre del empleado	Hora de entrada	Hora de salida
2	Rojo = editado	09/04/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/04/2022 01:38 PM	09/04/2022 08:01 PM
3	Púrpura = agregado	09/04/2022	VICTOR MANUEL DOMINGUEZ RIVAS		
4	Azul = eliminado	09/05/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/05/2022 12:56 PM	09/05/2022 02:50 PM
5		09/05/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/05/2022 04:35 PM	09/05/2022 06:43 PM
6		09/06/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/06/2022 04:01 PM	09/06/2022 05:16 PM
7		09/07/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/07/2022 12:18 PM	09/07/2022 02:08 PM
8		09/07/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/07/2022 04:54 PM	09/07/2022 06:59 PM
9		09/09/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/09/2022 11:58 AM	09/09/2022 01:28 PM
10		09/12/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/12/2022 10:44 AM	09/12/2022 02:33 PM
11		09/14/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/14/2022 01:07 PM	09/14/2022 02:17 PM
12		09/14/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/14/2022 05:10 PM	09/14/2022 06:05 PM
13		09/15/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/15/2022 08:45 PM	09/15/2022 09:52 PM
14		09/16/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/16/2022 01:30 PM	09/16/2022 02:46 PM
15		09/16/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/16/2022 03:55 PM	09/16/2022 05:45 PM
16		09/16/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/16/2022 11:50 PM	09/17/2022 12:31 AM
17		09/17/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/17/2022 07:50 PM	09/17/2022 10:52 PM
18		09/18/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/18/2022 12:13 AM	09/18/2022 01:19 AM
19		09/20/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/20/2022 11:34 AM	09/20/2022 07:24 PM
20		09/21/2022	VICTOR MANUEL DOMINGUEZ RIVAS	09/21/2022 11:59 AM	09/21/2022 01:27 PM

Figura 2.2: Extracto del registro de horas

## Zotero

Zotero es una plataforma open-source (bajo la licencia 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, GitHub y GitHub Actions

Git es un extremadamente popular y utilizado sistema de control de versiones distribuido libre y gratuito, creado por Linus Torvalds, creador de Linux. Es utilizado para guardar diferentes versiones de un conjunto de archivos, conocido como repositorio; pudiéndose recuperar en cualquier momento cualquier versión del mismo, a la vez que se guarda un registro de cuándo, quién y qué cambios se han realizado entre versiones [2].

Asimismo, Git, como otros sistemas de control de versiones, dispone de herramientas para la integración de cambios de varios usuarios, algo muy útil en el desarrollo de software. Por otra parte, al ser distribuido, se dispone una copia de todo el repositorio en cada máquina, por lo que no dependemos de un nodo central para poder trabajar; si bien se define normalmente un nodo remoto que hace las veces de *punto de encuentro* entre los distintos nodos.

Por otra parte, GitHub es una plataforma web, propiedad de Microsoft, donde los usuarios pueden alojar repositorios Git tanto públicos como privados; facilitando los procesos de compartir código y de colaboración en proyectos de otros programadores. GitHub no es la única plataforma para este proceso: Gitlab y Bitbucket son algunas de las alternativas [3].

El potencial de GitHub no acaba ahí: gracias a su extensión Acciones (o Actions en inglés) permite que en los repositorio se puedan automatizar y ejecutar flujos de trabajo, lo que abre las puertas a la ejecución de flujos de integración y despliegue (o entrega) continuo, conocidos como CI/CD.

En este proyecto se ha elegido Git para la gestión del código software como el del presente documento, mientras que el uso de GitHub se debe tanto para almacenar una copia de seguridad del código adicional, para la mejora de la visibilidad del proyecto y por último para poder configurar y ejecutar flujos de CI/CD para nuestro software.

## GitHub Actions

## 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 2.3: Logo actual de Android. Imagen extraída de [4]

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 [5].

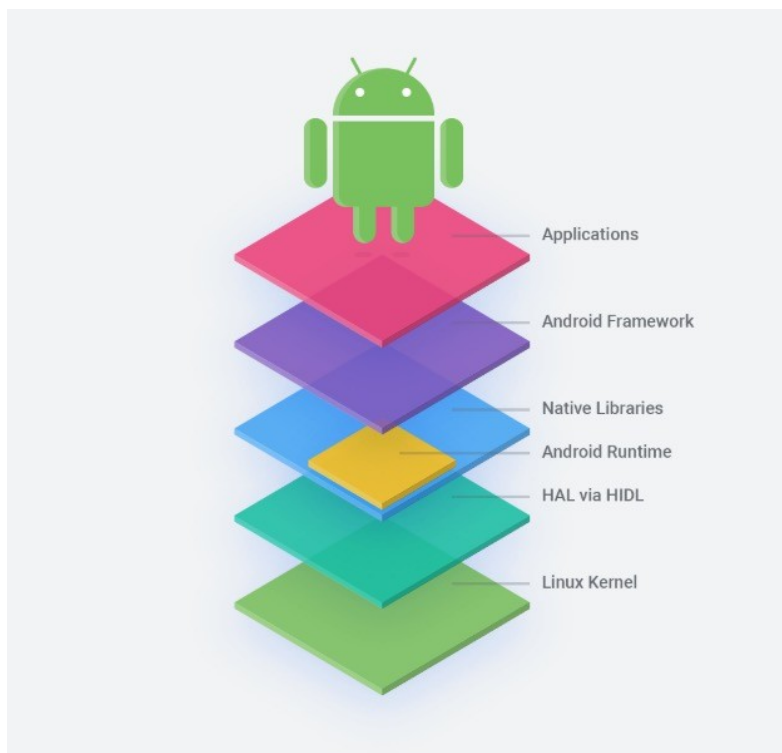


Figura 2.4: Capas de Android. Imagen extraída de [6]

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 [7] [8].

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



Figura 2.5: HTC Dream en funcionamiento. Imagen extraída de [9]

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

Figura 2.6: Estadísticas acumulativas de las versiones de Android. Imagen extraída de [10]

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 [11].

## 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++ [12]. No obstante, al ser Java un lenguaje interpretado sobre una máquina virtual (JVM o *Java Virtual Machine*) se abrió la puerta para utilizar otros lenguajes que utilizasen la JVM. En la conferencia de Google I/O de 2017 fue anunciado el soporte oficial y completo de Kotlin dentro de Android.



Figura 2.7: Logo de Kotlin. Imagen extraída de [13]

Kotlin es un lenguaje de programación desarrollado por JetBrains<sup>1</sup> y publicada su primera versión estable el 15 de febrero de 2016. El objetivo de este lenguaje fue tan sencillo como ambicioso: crear un lenguaje conciso (permitiendo reducir la cantidad de código *boilerplate*), con soporte de nuevas funcionalidades; pero sin renunciar a la rapidez de compilación de Java ni al todo el código escrito en él [14].

Sus principales características son las siguientes [15] [16]:

- Interoperable al 100% con Java, lo que facilita la reutilización de código ya existentes. Es interoperable en ambos sentidos.
- Permite escribir código más seguro, ya que en el diseño del lenguaje se solucionaron problemas crónicos de Java como las *Null Pointer Exception*. Según datos internos de Google, las aplicaciones escritas en Kotlin tienen un 20% de probabilidades menos de fallar.
- Soporte nativo y estructurado para la programación concurrente y asíncrona mediante construcciones como las *corrutinas* y los *flujos*.
- Desarrollo multiplataforma, no solo para Android: aplicaciones web, *backend* e iOS.
- Permite desarrollos en varios paradigmas: orientada a objetos, funcional, imperativa...

Asimismo, al convertirse en la I/O de 2019 en el lenguaje de referencia para el desarrollo de Android [17], los desarrollos de librerías y herramientas relacionadas con Android están escritas en este lenguaje, aprovechando al máximo sus nuevas características. Por tanto, si bien es interoperable con Java, está recomendado que los nuevos desarrollos lo utilicen [18], como se ha realizado en este proyecto.

## Jetpack Compose

Jetpack Compose es un 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 [19].

---

<sup>1</sup>JetBrains es una empresa muy reconocida dentro de la industria por crear una serie de entornos de desarrollo muy populares, como PhpStorm, CLion o IntelliJ IDEA. Sobre este último está construido Android Studio, el entorno de desarrollo oficial dentro de Android.

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.

Este enfoque declarativo nos permite describir cómo queremos que sea nuestra interfaz gráfica. Asimismo, las interfaces que construimos con este sistema pueden estar interconectadas a un estado que definamos; describiendo cómo será nuestra interfaz para cada posible estado. Cuando ese estado cambie, nuestra interfaz gráfica cambiará automáticamente para mostrar el nuevo estado, simplificando enormemente el desarrollo y reduciendo el código necesario [20].

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 exhaustivamente. 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/Kotlin de la aplicación se accediera a dichos elementos y se le aplicaran manualmente modificaciones y transformaciones [21].

Además, en Jetpack Compose, a diferencia del sistema anterior, los componentes gráficos están desacoplados del sistema operativo; por lo que no dependemos de la versión del terminal para mostrar correctamente nuestra interfaz gráfica. Eso ocurría anteriormente y como ya vimos, la fragmentación en Android es un problema endémico, lo que complicaba bastante el desarrollo. Asimismo, es compatible con los componentes XML del sistema anterior, lo que facilita la migración de los proyectos antiguos a este nuevo paradigma.

No obstante, como ya vimos en el apartado anterior, está diseñado para ser utilizado desde Kotlin, por lo que en la práctica obliga a usar dicho lenguaje; lo que en algunos casos puede resultar en un pico de dificultad hasta que se domina el lenguaje.

Por último, esclarecer que Jetpack Compose es parte de un conjunto de librerías más grande conocido como Android Jetpack, el cual es promovido por Google para mejorar el desarrollo dentro de Android [22] [23]. Por otra parte, estas librerías no tienen ninguna relación entre sí; por lo que en la literatura especializada se abusa de la notación. Al resto de las librerías de Jetpack se les conoce por su nombre, sin el prefijo Jetpack; siendo la única que lo incorpora Compose. En este documento seguiremos ese enfoque para mejorar la legibilidad.

## Material Design 3

Material Design 3 (también conocido como *Material You*) es la tercera iteración del conjunto de principios y directrices de diseño de Google, como respuesta a la creciente ubicuidad de Android: móviles con pantallas plegables, *smartwatch*, televisores [24]... Su primera implementación estable para Jetpack Compose fue lanzada el 24 de octubre de 2022 [25].

Al ser utilizado por Google para la creación de elementos gráficos tanto en sus aplicaciones como en el sistema operativo, es la guía de diseño de facto dentro del ecosistema Android.

Sus principales características son las siguientes [26]:

- Centrado en la personalización de la interfaz gráfica. Los diseñadores definirán tres colores principales, los cuales serán utilizados para los elementos gráficos de forma totalmente transparente al programador. A partir de dichos colores, se elaborará mediante la herramienta *Material Design Builder* [27] una paleta de colores con variantes de los mismos, diseñada para cumplir estándares de accesibilidad<sup>2</sup>, garantizando el nivel de contraste correcto.

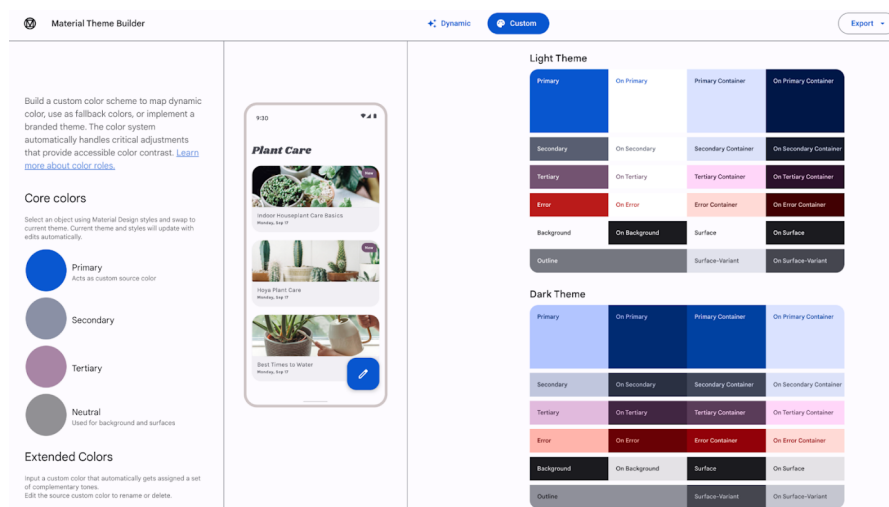


Figura 2.8: Ejemplo de uso de la herramienta *Material Design Builder*. Imagen extraída de [25]

Además, si el dispositivo dispone de Android 12 o superior, pueden tomarse dichos colores desde el fondo de pantalla del usuario, incrementando exponencialmente la personalización; si bien se permite establecer colores *fijos* para ciertos contenidos.

- Soporte nativo para categorizar el tamaño de la pantalla del dispositivo, tanto en altura como en anchura.

<sup>2</sup>Describir dichos estándares está fuera del alcance del proyecto, pero es un proceso basado en proporciones de luminancia



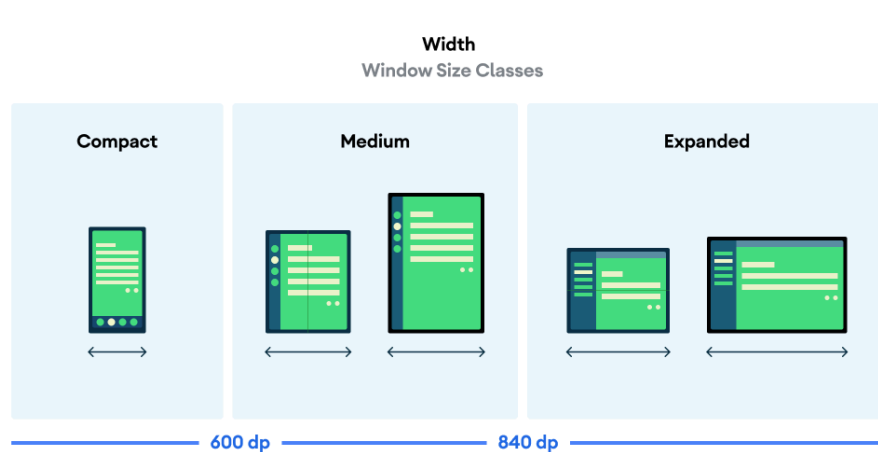


Figura 2.9: Categorías de pantalla según anchura. Imagen extraída de [25]

- Sistema de fuentes basado en estilos principales para cada tipo de contenido: desde titulares hasta etiquetas, pasando por títulos, cuerpos de texto...
- Soporte nativo para animaciones, las cuales ya son utilizadas en los componentes gráficos nativos, como los *switch*.
- Evolución de muchos elementos gráficos, como las tarjetas, botones, selectores de fechas...

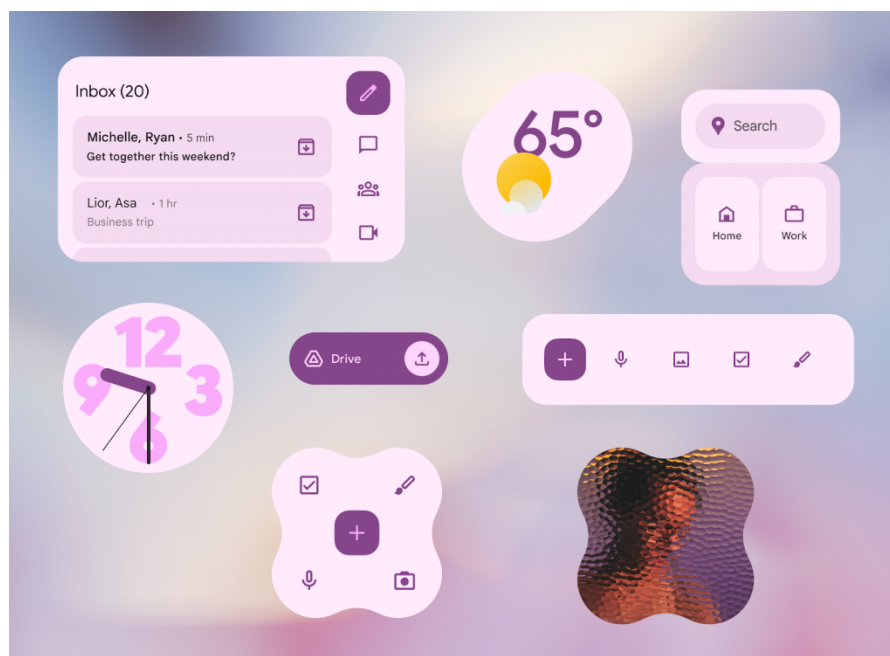


Figura 2.10: Algunos elementos gráficos de Material Design 3. Imagen extraída de [28]

- Sistema de formas o *redondeo* multinivel para modernizar nuestros elementos y hacerlos más distinguibles entre sí.
- Mejora el concepto conocido como *elevación*, basándose en colores y no en sombras. Este elemento permite superponer elementos y transmitir gráfica y visualmente la importancia de cada uno de ellos.
- Soporte nativo para tema claro y oscuro, ya que en los últimos años los temas oscuros se han hecho cada vez más populares en las aplicaciones y no estaba presente previamente de forma nativa.

En pocas palabras, en esta versión se han dedicado a modernizar el lenguaje de diseño, haciéndolo más atractivo y completo; alejándose de lo puramente funcional, mejorando la accesibilidad y explorando el mundo de la personalización para el usuario.

## Salud Conectada

En la conferencia I/O (sí, otra vez) de 2022 se anunció *Health Connect* (o Salud Conectada en castellano), una aplicación creada por Google (junto con Samsung [29]) que aglutinará todos los datos relacionados con salud dentro del ecosistema Android. La aplicación (en beta desde noviembre de 2022) es compatible con Android 9 o superior, mientras que está previsto que para Android 14 venga incorporada aplicación de fábrica o preinstalada<sup>3</sup> [30].



Figura 2.11: Logo de Salud Conectada. Imagen extraída de [31]

Esta herramienta viene a solucionar una problemática importante y es la ausencia del reaprovechamiento de los datos relacionados con la salud. Hasta ese momento, la posición casi unánime dentro la industria se basaba en aislar los datos recoletados por sus dispositivos *wearables* o aplicación dentro de su ecosistema [32] [33].

<sup>3</sup>Recordar aquí que esta aplicación es parte de los servicios de Google y no del sistema operativo propiamente dicho o AOSP.

La única manera utilizar dichos datos en otras aplicaciones estaba restringida a sistemas propietarios como Google Fit, cuyos datos se almacenaban en la nube con los problemas de privacidad asociados, o proyectos *open source* como Gadget Bridge [34]; que accedían a datos mediante ingeniería inversa.

A fecha de mayo de 2023, se estima que más de 100 aplicaciones han integrado Salud Conectada, incluyendo las aplicaciones de las pulseras Fitbit y Samsung, Peloton, Oura [35]...

Centrándonos en Salud Conectada, se trata tanto de una plataforma como una API para los desarrolladores. La plataforma puede registrar datos como la actividad física, el sueño, la nutrición o incluso el ciclo menstrual<sup>4</sup>, siendo un intermediario entre las aplicaciones que generan o escriben dichos datos y las que quieren acceder a esos datos.

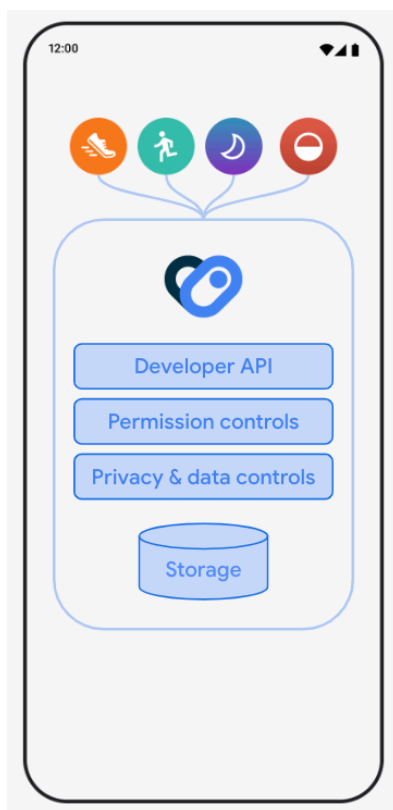


Figura 2.12: Arquitectura básica de Salud Conectada. Imagen extraída de [29]

Esta API estandariza el acceso a todas las fuentes de datos, independientemente de su procedencia; lo que simplifica enormemente tanto la lectura como escritura de los datos. Además, el sistema permite consultar datos agregados, pudiendo ser una agregación acumulativa (como el total de

<sup>4</sup>La lista completa de los datos que puede registrar Salud Conectada se encuentra disponible en [36]

distancia caminada en un intervalo de tiempo) o estadística (las pulsaciones mínimas, máximas o promedio en un intervalo de tiempo).

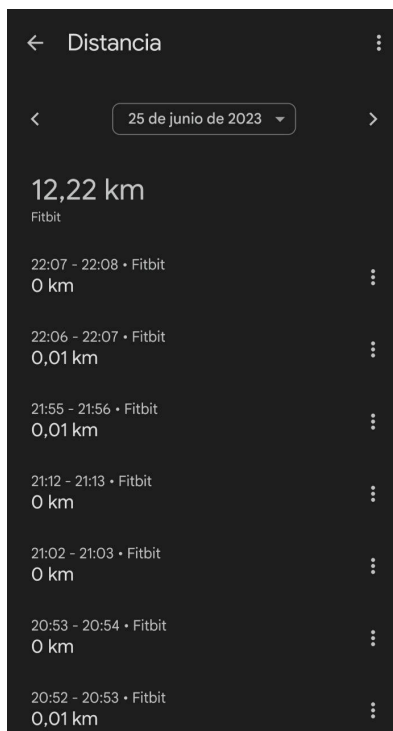


Figura 2.13: Visualización de datos de distancia. Elaboración propia.

Por otra parte, está diseñada con la privacidad en mente: los datos se almacenan localmente, mientras que el acceso a los mismos está fuertemente granularizado: en la que el usuario puede decidir qué aplicaciones tienen acceso (tanto lectura como escritura) a cada tipo de registro [37].

Asimismo, las aplicaciones solo pueden leer datos con una antigüedad de hasta 30 días previos a su instalación. [38], registrándose además todas las lecturas y escrituras en el sistema, las cuales puede visualizar el usuario.

En definitiva, si bien está aún en desarrollo, esta plataforma nos permite abstraernos del hardware de recolección de datos biométricos, reduciendo la complejidad de nuestro desarrollo; y además permite al usuario un mayor control sobre sus datos sensibles.

## Room

Uno de los servicios que provee Android es SQLite, un sistema de gestión de base de datos relacionales de bajo nivel. Este sistema es usado por la aplicación que lo utiliza, ahorrando configuración y complejidad [39].

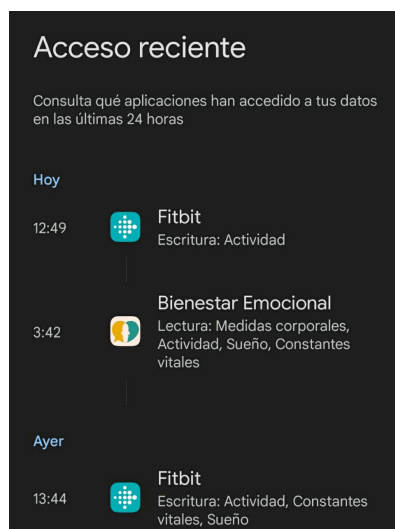


Figura 2.14: Visualización del acceso a los datos. Elaboración propia.

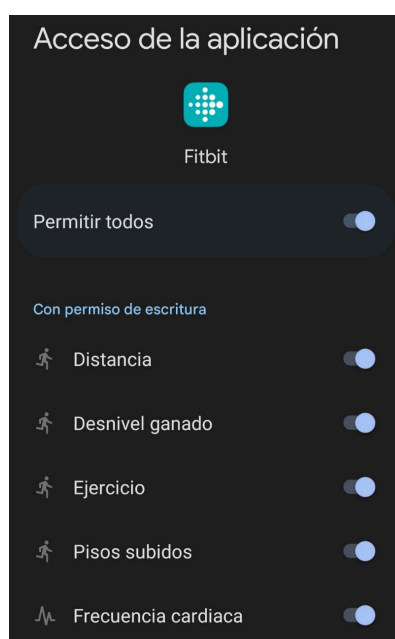


Figura 2.15: Granularidad de los permisos. Elaboración propia.

No obstante para muchas aplicaciones es un modelo demasiado rígido. Para simplificar el uso de SQLite Google creó en 2017 esta librería, la cual hace de intermediario entre el propio SQLite y nuestra aplicación [40]. Técnicamente hablando se trata de una capa de abstracción sobre dicho sistema gestor, la cual permite reducir la complejidad de los usos comunes de la base de datos; sin perder el acceso a SQLite. Asimismo, esta librería brinda otras ventajas, como la verificación de las consultas SQL en tiempo de compilación [41].

Los elementos básicos de esta plataforma son los siguientes:

- *Data Access Objects*, también conocido como DAO. En este elemento (implementado como una interfaz con la anotación `@Dao`) creamos nuestras operaciones sobre la base de datos: consultas, inserciones...
- Entidades, las cuales se corresponden con una clase en Kotlin (anotada como `@Entity`) y normalmente con una tabla de base de datos. Con esta estructura creamos las tablas de la base de datos, mientras que en nuestra aplicación disponemos de objetos que mapean dichos registros. Los atributos de dicha clase serán los campos de la tabla en cuestión, sobre los cuales podemos utilizar anotaciones para establecer claves, índices, etc
- Clase base de datos, donde especificamos las entidades a utilizar y cómo se instancia dicha base de datos (aquí podemos ajustar si estará cifrada o no por ejemplo). Se corresponde con una clase abstracta que hereda de *RoomDatabase* y que usa ciertas anotaciones para que se genere automáticamente el código *boilerplate*.
- Una función que construya la instancia de dicha base de datos con la configuración deseada (nombre del fichero de la base de datos, cifrado..). Para dicha construcción de la instancia Room implementa el patrón factoría.

## Dagger Hilt

Dagger Hilt es una biblioteca que permite la automatización de la inyección de dependencias dentro de nuestra aplicación; la cual es la opción promovida por Google para esta tarea. El nombre compuesto no es casual, ya que hace referencia a que en realidad son dos librerías: Dagger realiza el trabajo de la inyección de dependencias, mientras que Hilt es la integración estándar de Dagger en Android [42], simplificando el proceso y facilitando la curva de aprendizaje de Dagger [43].

Para poder utilizar esta librería, es obligatorio marcar un punto de arranque en la aplicación para que se inicialice el sistema de dependencias. Asimismo, se debe indicar que Dagger Hilt debe inyectar dependencias en cada caso que corresponda. Por otra parte, se necesitan definir uno o más módulos, en los cuales se indica a Dagger Hilt cómo construir las dependencias, el alcance de las mismas y otras características.

Esto nos permite que algunas dependencias sean construidas con una única instancia para todo el proyecto, mientras que otras puedan ser instanciadas en múltiples ocasiones, o con un ciclo de vida asociado al de algunos componentes de Android (como los ViewModel).

En resumen se trata de una tecnología auxiliar que nos permite realizar una inyección de dependencias eficaz sin demasiada complejidad de uso.

## SQLCipher

SQLCipher es una capa de encriptación sobre SQLite, la cual proporciona un cifrado de nuestra base de datos con el algoritmo AES-256. Para realizar ese cifrado, debemos generar aleatoriamente una contraseña para cada dispositivo. Para este procedimiento SQLCipher soporta el uso de dos herramientas de Android: *KeyGenerator* como generador de claves y *KeyStore* como sistema de almacenamiento seguro.

Para la generación de dicha contraseña, acudimos a *KeyGenerator* y le indicamos las características de la contraseña, como para qué algoritmo se usará o el modo de cifrado . que usará la clave.<sup>5</sup> En este caso, generamos una clave para AES-256. El modo de cifrado es libre, utilizando en este caso el sistema GCM gracias a su seguridad, su relativamente bajo coste computacional y su facilidad para ser paralelizado [45].

Por otra parte, necesitamos almacenar de manera segura dicha contraseña. *KeyStore* se encarga de dificultar la extracción de información, garantizando que una aplicación solo puede acceder a su propia información [46].

Tras ello, para implementar dicho cifrado en nuestra base de datos con Room, es necesario indicar que en el proceso de creación de la base de datos se utilice la factoría de esta capa, con la contraseña generada anteriormente. Con la implementación de este cifrado, nuestra base de datos podrá ser leída únicamente por nuestra aplicación.

Una descripción detallada del funcionamiento de esta capa puede encontrarse en [47].

## Compose Destination

Debido a las limitaciones de Jetpack Compose, entre otras circunstancias, el paso de parámetros entre ventanas está implementado con un procedimiento bastante rígido y poco potente comparado con el disponible en la programación de Android *tradicional*. En particular, se realiza de muy similar a como se realiza en la navegación web [48]; siendo necesario definir rutas simbólicas, a las cuales se le añaden parámetros. Dichos argumentos están limitados a unos pocos tipos [49], lo

---

<sup>5</sup>Una lista exhaustiva de los modos disponibles se puede encontrar en [44]

que complica aún más el proceso. Además, el paso de dichos argumentos no es seguro respecto a tipos.

Para solventar toda esta ristra de desventajas, se utiliza la librería *Compose Destination*. Esta se encarga de proporcionarnos un paso de argumentos seguro respecto a tipos, de la misma manera como pasamos parámetros a funciones comunes. Para cubrir ese hueco, utiliza el *procesamiento de símbolos de Kotlin* (KSP o *Kotlin Symbol Processing*) para generar código automáticamente que implemente el paso de argumentos de Jetpack Compose a la vez que nos proporciona una interfaz simple y segura mediante anotaciones.

## Work Manager

Work Manager es el componente oficial de Android para la planificación de subtareas dentro de nuestra aplicación. Esta librería está diseñada para simplificar el uso del amalgama de librerías para esta funcionalidad, ya que, según la versión del sistema, debían utilizarse unas u otras. Work Manager establece una única interfaz para esta funcionalidad, encargándose internamente de utilizar la librería correcta a bajo nivel [50].

Este componente admite dos tipos de tareas: puntuales y periódicas; las cuales se pueden planificar o cancelar. Asimismo, además de la planificación temporal, permite el establecimiento de algunas restricciones, como que el dispositivo esté cargándose o disponga de conexión a internet, ejecutándose la tarea en cuestión cuando se cumplan al mismo las directrices temporales y las adicionales (si las hay) [51].

Asimismo, nos permite establecer una política de reintentos si nuestras tareas no se ejecutan correctamente, por lo que es la solución recomendada para cuando necesitamos realizar cierta funcionalidad de manera fiable.

Los problemas que da los comentamos de pasada referenciando a stackoverflow o ya al final? Por mi solo al final.

## Retrofit

Retrofit es una librería muy popular en el ecosistema Android, ya que nos proporciona una capa de alto nivel para el consumo de API. En particular, nos permite interactuar con dichos recursos de una manera estructurada y orientado a objetos (o a entidades). Además proporciona soporte nativo para el uso de corrutinas, por lo que nos permite realizar llamadas remotas de forma concurrente,



sin bloquear el hilo principal de nuestra aplicación.

Esta biblioteca trabaja junto con otros componentes software conocidos como serializadores, los cuales se encargan de convertir nuestros objetos Kotlin a formato json, los cuales son necesarios para interactuar con los recursos en red. Retrofit soporta una lista de ellos, pero en este proyecto utilizaremos el incorporado por defecto: gson, creado y mantenido por Google.

Sus elementos básicos son los siguientes [52] <sup>6</sup>:

- Entidades, las cuales mapean el formato json necesario (ya sea para enviar o recibir datos) con una clase en Kotlin.
- Una interfaz donde se asocian las llamadas a la API (ruta y tipo) que utilizaremos junto a la cabecera de la función que implementará Retrofit automáticamente. Aquí le indicaremos, si procede, los parámetros de la llamada y el resultado de la misma, los cuales serán normalmente entidades que habremos definido previamente.
- Una función que construya una instancia de la interfaz anterior, donde se le indica la url base de la API en cuestión y configuración adicional, como el serializador a utilizar.

## DataStore

Datastore es una solución oficial de almacenamiento de datos para Android que busca corregir ciertos defectos del procedimiento anterior (*SharedPreferences*) [53] [54]. Este mecanismo permite guardar en un fichero el dispositivo pares claves-valor o objetos que implementen el *protocol buffer* [55].

Asimismo, está diseñado para almacenar cantidades pequeñas de datos (usualmente configuraciones, variables que se necesita que sean persistentes...), tomando Room el testigo para cantidades mayores [56]. Datastore nos proporciona un enfoque transaccional y consistente al almacenar la información, mientras que apuesta fuertemente por la asincronía, haciendo uso extensivo de las técnicas asíncronas de Kotlin, como las corrutinas y los flujos.

En este proyecto utilizaremos el subcomponente orientado a pares clave valor, denominado *Preference DataStore*.

---

<sup>6</sup>Es posible que te recuerde a los componentes de Room y no es casualidad, ya que ambas están diseñadas de una manera similar, utilizando los mismos patrones de diseño software

## Recursos y Lingver

La localización básica de una aplicación en Android comienza por traducir el contenido de la interfaz gráfica. Para realizarlo nos apoyaremos en el concepto de recurso, que se puede definir como cualquier elemento estático que necesitare la aplicación que no sea código: imágenes, textos, sonidos, etc [57]. Una aplicación puede proveer varios conjuntos de recursos, cada uno adaptado a una configuración diferente; mientras que el sistema escogerá el conjunto que más adecuado. Si bien aquí nos centramos en la localización, el sistema de recursos es más potente; ya que permite, por ejemplo, disponer de configuraciones diferentes según la densidad de resolución de la pantalla [58].

Una práctica recomendada es escribir los textos de la aplicación en un fichero XML. Este archivo es creado automáticamente por Android, denominado *strings.xml*, dentro de la carpeta *res/values*. Cada elemento de ese fichero tiene asociado un nombre o clave y un valor, y desde el código puede invocarse a ese recurso por la clave correspondiente.

Por tanto, la localización básica consiste en traducir los valores de ese archivo a otro idioma. Para que el sistema pueda reconocer que es una variante del fichero *strings.xml*, se le guarda en *res/values-X*, siendo X el código del idioma (es para español, en para inglés..). No obstante, dentro de esas carpetas *values* podemos crear más recursos de texto. Esto es útil si queremos evitar un monolito con todos los textos de la aplicación.

Al mantener las claves, este procedimiento es transparente tanto para el programador como para el usuario, ya que el sistema lo resuelve en tiempo de ejecución. No obstante, este sistema tiene una limitación relativamente importante, y es que al resolverlo automáticamente según la configuración en ejecución nos impide sobreescibir esa configuración para nuestra aplicación. Para la inmensa mayoría de situaciones no supone ningún problema, pero nos impide tener la aplicación en un idioma distinto al sistema operativo.

Para resolver este caso límite hay una librería, conocida como Lingver [59]. A grandes rasgos se encarga de guardar el idioma que queremos en nuestra aplicación y de sobreescibir la configuración del idioma dentro de la app, lidiando con las particularidades de las distintas versiones de Android; ya que el sistema no fue diseñado originalmente para incorporar esta funcionalidad.

## Lottie

Lottie es una biblioteca creada por Airbnb que busca facilitar la creación y uso de animaciones en entornos multiplataforma (web, iOS, Android..) [60]. A diferencia de archivos como los GIF, esta

librería permite que las animaciones puedan escalarse sin perder calidad (como en los ficheros SVG), haciendo hincapié en el rendimiento.

Concretamente, la librería renderiza en tiempo real animaciones de Adobe After Effects, las cuales son exportadas como ficheros JSON que serializan dicha animación [61], a través de una extensión de código abierto conocida como Bodymovin. Por tanto, estas animaciones pueden ser personalizadas completamente, a la vez que se reducen las necesidades de cómputo para mostrarlas.

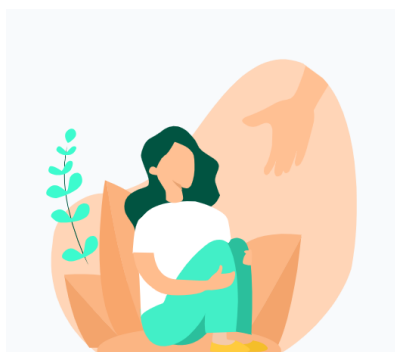


Figura 2.16: Vista previa de una animación personalizada

En este proyecto abordaremos únicamente la personalización de animaciones gratuitas aportadas por la comunidad y su presentación en la aplicación; y no la creación de nuevas. Para conseguir dichas animaciones se ha utilizado el portal LottieFiles.

## Vico

Vico es una biblioteca open-source para la realización de gráficos, creada por Patryk Goworowski. Su particularidad reside en su compatibilidad tanto con el sistema de vistas tradicional como con Jetpack Compose, algo único en el ecosistema de Android [62].

Si bien no es tan potente como otras librerías como *MPAndroid*, sus posibilidades son más que suficientes para este proyecto, aligerando el desarrollo. Además su soporte nativo de Jetpack Compose reduce la complejidad del desarrollo e introduce algunas mejoras, como una interconexión directa con los colores que usará Jetpack Compose para los elementos gráficos, mejorando la calidad visual de la aplicación.

Vico nos permite crear únicamente gráficos de barras y de líneas. Su humilde enfoque, si bien puede condicionar su uso, permite que contenga numerosos añadidos completamente personalizables, tales como leyendas, marcadores (tanto fijos como temporales), *líneas umbral*, personalización programática de los ejes...

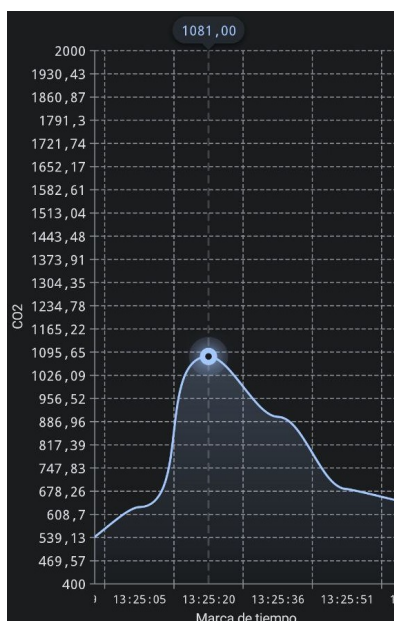


Figura 2.17: Ejemplo de gráfica de líneas con Vico

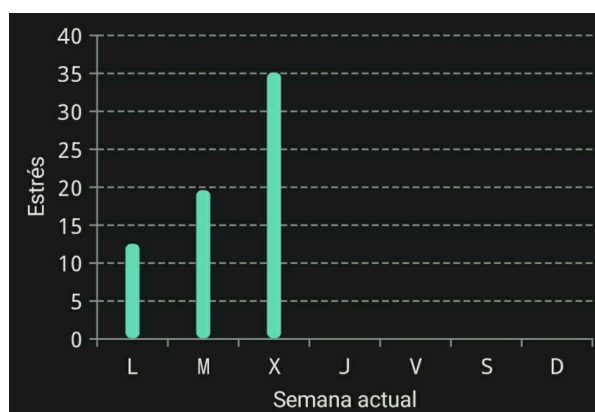


Figura 2.18: Ejemplo de gráfica de barras con Vico

Asimismo, su mantenimiento y mejora es notable y constante. Nuevas versiones son lanzadas cada pocas semanas [63], mientras que el soporte ofrecido en su repositorio es rápido y detallado [64]. Además, dispone de numerosa documentación para aprenderla rápidamente, si bien no cubre los casos de uso más avanzados.

Por último, dispone de una aplicación demo que permite ver rápidamente las posibilidades que ofrece la librería.

## Servidor

### Python

Python es un lenguaje de programación interpretado creado por Guido Van Rossum en 1991, con un marcado enfoque de construir código de la manera más clara y legible posible. Es uno de los lenguajes más populares de la escena, gracias a su facilidad de uso y a su enorme comunidad de desarrolladores. Estos últimos han creado un enorme ecosistema de librerías que facilita la creación de nuevos proyectos, especialmente para prototipos.



Figura 2.19: Logo de Python. Imagen extraída de [65]

Actualmente es utilizado para todo tipo de tareas, tales como desarrollo web, aplicaciones científicas... pasando por inteligencia artificial o análisis de datos; casi siempre apoyándose en potentes librerías que mejoran notablemente el propio lenguaje.

Sus principales características son las siguientes:

- Soporte mutiparadigma: como Kotlin, se puede utilizar programación imperativa, orientada a objetos o funcional en cualquier momento.
- Gran biblioteca estándar que facilita muchas operaciones mundanas, como la lectura de archivos JSON.
- Soporte multiplataforma gracias a la compatibilidad de su intérprete con todos los sistemas operativos principales. Llega hasta tal punto que un subconjunto del lenguaje (MicroPython) es compatible con algunos microcontroladores.
- Sistema de tipos fuertemente tipado y dinámico, lo que permite que una variable pueda cambiar de tipo fácilmente.
- Facilidad de instalación de librerías gracias a la herramienta pip.

## Flask

Flask es un popular entorno ligero o *microframework* de Python utilizado para construir aplicaciones web, tanto como páginas como API. Diseñado para ser ligero y fácil de usar, Flask proporciona las herramientas necesarias para crear dichas aplicaciones web de una manera minimalista, rápida y eficiente, en sintonía con la propia filosofía del lenguaje.

A diferencia de otros entornos más completos como Django, Flask se enfoca en brindar solo lo esencial, lo que permite crear prototipos lo más fácilmente posible, si bien puede quedarse corto para aplicaciones relativamente complejas [66]. Su sistema de enrutamiento HTTP es sencillo y claro, mientras que posee extensiones para autenticación de usuarios, manejo de base de datos, etc.

Gracias a esta simplicidad se usará en este proyecto para prototipar la parte del servidor. Asimismo, debido a su enfoque modular y a su sistema de plantillas podría expandirse su funcionalidad con más elementos web sin ninguna adaptación del código ya existente.

## MongoDB

MongoDB es un sistema gestor de base de datos no relacional orientado a documentos, diseñado para manejar grandes volúmenes de datos de manera eficiente y escalable. A diferencia de las bases de datos relacionales tradicionales, no hay registros sino documentos, unos ficheros con una extensión muy similar a JSON: BSON (JSON Binarios) [67].



Figura 2.20: Logo de MongoDB. Imagen extraída de [68]

Este enfoque no relacional le permite ofrecer un modelo de datos flexible, sin un esquema. Esto se traduce en una mayor libertad para almacenar datos y cambiar el modelo de los mismos, ya que para cambiar cualquier atributo no es necesario reconstruir el modelo; muy útil para prototipos o modelos de datos dinámicos.

Por otra parte, el modelo de consultas ofrece posibilidades similares a las de SQL, por lo que no supone una desventaja. También implementa conceptos presentes en las base de datos relacionales,

como los índices. Además, al disponer de una gran comunidad detrás, existen numerosos módulos que lo permiten integrar con lenguajes de programación, como pymongo para Python.

Asimismo, posee otras dos grandes fortalezas: un núcleo distribuido, lo que permite una alta disponibilidad y escalabilidad; y una licencia de uso gratuito desde octubre de 2018, bajo la licencia AGPL [69].

# 3.

## Estado del arte

---

*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

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

---

*Si no buscas metas más altas y te esfuerzas continuamente no alcanzarás el siguiente nivel.*

Dr. Toshiya Joshima

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

añadir tableros etc

hablar de los repositorios git y github

Metodología del desarrollo seleccionada

## 5. Análisis del sistema propuesto

---

*Cada día me gusta levantarme porque hay otro reto.*

Roger Penske

### 5.1 Análisis de Stakeholders

### 5.2 Requisitos de usuario

### 5.3 Requisitos funcionales

### 5.4 Requisitos no funcionales

## 6. Diseño del sistema propuesto

---

*Si tienes tanto miedo al fracaso, nunca tendrás éxito. Tienes que arriesgarte.*

Mario Andretti

### 6.1 Descripción del caso de estudio

### 6.2 Diseño del sistema

# 7.

# Desarrollo del sistema

*¿Quién dijo miedo habiendo hospitales?*  
Sabiduría popular de la ETSISI

## 7.1 Project Blastoff

descripcion del blastoff

### Inception Deck

El *Inception Deck* (también conocido como *Agile Inception*) [70] [71] 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 [72] [73] [74] [75] 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

nos falta la imagen cuando esté disponible

## NOT list

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

• Realización profesional de la gestión de proyecto siguiendo metodologías ágiles.	
No resuelto	

Cuadro 7.1: 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

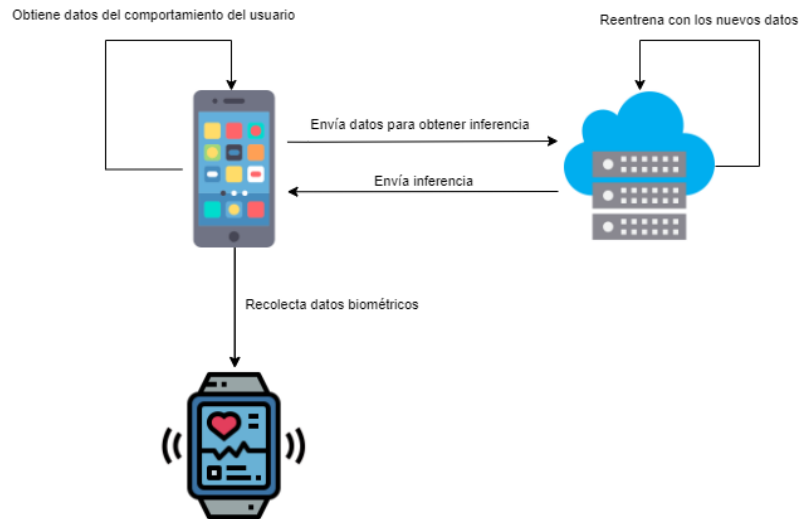


Figura 7.1: Componentes básicos del sistema



Figura 7.2: 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"> <li>• Realización de pruebas exhaustivas tanto del <i>backend</i> como de la interfaz gráfica de usuario.</li> <li>• Participación de voluntarios que aporten datos al estudio con los que mejorar la precisión y evitar sesgos.</li> <li>• Elaboración de unos términos y condiciones legales acerca de los datos obtenidos en el proyecto.</li> <li>• Accesibilidad correcta de la aplicación.</li> <li>• Elaboración de un diseño coherente a nivel UX/UI de la aplicación.</li> <li>• Comunicación y difusión eficaz del proyecto en la escuela, en la defensa de este TFM y en redes sociales.</li> <li>• Continuidad del proyecto después del presente TFM para, entre otros, publicar una aplicación en iOS.</li> <li>• Definición clara de la arquitectura y de los procesos de mantenimiento y ampliación de la aplicación.</li> <li>• Mantenimiento futuro de la aplicación con las nuevas versiones de Android.</li> <li>• Potencia de cómputo suficiente para entrenar el modelo de Inteligencia Artificial</li> </ul>	<ul style="list-style-type: none"> <li>• Problemas de compatibilidad entre las versiones de Android.</li> <li>• Muestra de estudiantes variada y contrastada participando en el proyecto.</li> <li>• Soportes de los fabricantes de <i>wearables</i> (en especial Samsung) a Health Connect.</li> <li>• Funcionamiento de las pulseras Xiaomi en aplicaciones no oficiales.</li> <li>• Disponibilidad de los datos del móvil y de los <i>wearables</i>.</li> <li>• Fiabilidad suficiente de los datos obtenidos por el móvil y los <i>wearables</i>.</li> <li>• Incumplimiento de la planificación por circunstancias ajenas al proyecto.</li> <li>• Retrasos en la entrega del hardware necesario para el sistema.</li> <li>• Ausencia de fallos al publicar la aplicación en la Google Play Store.</li> <li>• Precisión insuficiente del modelo de Inteligencia Artificial para ser utilizado en un entorno real.</li> </ul>

Cuadro 7.2: Riesgos del proyecto

## Size it up

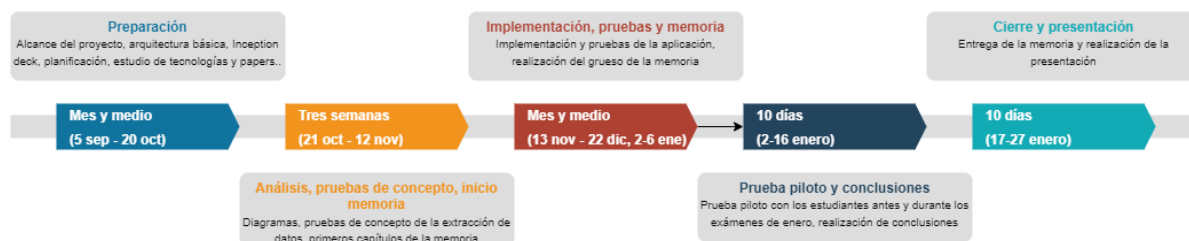


Figura 7.3: Dimensionamiento aproximado del proyecto

## 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 7.3: 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€.

## 7.2 Setup del proyecto

## 7.3 Implementación de la aplicación móvil

## 7.4 Implementación de la API del servidor

Subida de datos de usuario

Subida de los datos de los cuestionarios

Datos de la comunidad

## 8.

# Pruebas del sistema

---

*Las palabras tienen poder, pero sólo si la gente las escucha. Cuando no lo hacen, las acciones hablan lo suficientemente alto como para que cualquiera las escuche.*

Ryosuke Takahashi

### 8.1 Aplicación móvil

### 8.2 API del servidor

Pruebas unitarias

Pruebas de integración



# 9.

# Resultados

---

*¡Los números Mason! ¿Qué significan?*

Jason Hudson

## 9.1 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

## 9.2 Resultados del caso de estudio

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

# 10. Impacto social y medioambiental

---

*Solo se vive una sola vez, pero si hiciste las cosas bien una vez es suficiente.*

Cita atribuida a Mae West

Hablar aquí de la privacidad de los datos antes de health connect (usa, gpd)

Estamos en la cresta de la ola al venir todo para Android 14

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

## 10.1 Contexto medioambiental

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

## 10.2 Aspectos éticos, sociales y económicos

11.

# Presupuesto

---

*Lo perfecto es enemigo de lo bueno.*

Cita atribuida a Voltaire

# 12.

# Conclusiones

---

*El éxito es realmente la libertad de hacer lo que quieras.*

Cita atribuida a Magnus Walker

12.1 Conclusiones técnicas

12.2 Conocimientos adquiridos

12.3 Reflexión final

# 13.

## Líneas futuras

---

*La victoria más bella es siempre la próxima.*

Enzo Ferrari

Comprobar si hemos hablado del sistema como prototipo

Comprobar si alguna de estas la hemos metido al final

Comprobar si se habló de hacer la prueba piloto antes y si es así referenciarlo aquí

Al ser el sistema de este proyecto un prototipo, existen numerosas vías para ampliarlo y refinarlo. Algunas de dichas mejoras son:

1. Cifrado de las comunicaciones entre la app y el servidor mediante un certificado SSL en el propio servidor.
2. Establecimiento de una política de privacidad detallada. Este punto es imprescindible para poder publicar la aplicación (en tiendas virtuales como Play Store) y recoger datos con legalidad.
3. Desplegar la aplicación en el ámbito universitario. Realizándolo se podría obtener una retroalimentación más amplia y profunda de la aplicación, y además se podrían recolectar datos para entrenar modelos que predigan cada una de las variables de Salud Mental.
4. Comprobar la recolección de datos en otras pulseras, tanto de Fitbit como de otros fabricantes.
5. Adaptar la aplicación para soportar Android 14. Como hemos comentado en la sección 2.2, Android 14 vendrá preinstalado, y Google ha decidido introducir cambios notables, pasan-

do Salud Conectada de ser un APK a un *framework* [76], por lo que se necesita ajustar la aplicación para cubrir este caso.

6. Elaboración de un conjunto de pruebas exhaustivo para la aplicación móvil, incluyendo reportes de *coverage* del código.
7. Establecer que al instalar la aplicación se realice un cuestionario diario. Según la hora del día se elegiría si realizar el de mañana o el de noche, pero en cualquier momento del día uno de esos dos.
8. Dar más presencia a los cuestionarios puntuales, ya sea con gráficas, notificaciones...
9. Estudiar la imputación de datos nulos con otras técnicas, tales como: regresión, *Last Observation Carried Forward*, *Next Observation Carried Backward*... [77]
10. Realizar agregación de datos para ciertas variables, como las pulsaciones, para aligerar el volumen de datos.
11. Diseñar e implementar una interfaz gráfica centrada en las pantallas *expandidas*<sup>1</sup>.
12. Localizar la aplicación a más idiomas.
13. Disponer de al menos dos pautas de recomendación a los usuarios para cada variable y estado.
14. Segmentar la comunidad de usuarios para mejorar la representatividad del conjunto.

---

<sup>1</sup>Pantallas que disponen de más de 840dp de ancho o 900dp de alto

# 14.

# Temporal

---

## 14.1 Setup del proyecto

## 14.2 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.

### $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ y Overleaf

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  como medio de creación de la presente memoria y Overleaf como editor de dicho lenguaje.

# 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] Atlassian, *Software de control de versiones: descripción general*, es. dirección: <https://bitbucket.org/product/es/version-control-software> (visitado 06-07-2023).
- [3] *Git vs GitHub – ¿Qué es el Control de Versiones y Cómo Funciona?*, es, feb. de 2021. dirección: <https://www.freecodecamp.org/espanol/news/git-vs-github-what-is-version-control-and-how-does-it-work/> (visitado 06-07-2023).
- [4] 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).
- [5] 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).
- [6] 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).
- [7] 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).



- [8] 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).
- [9] M. Oryl, *T-Mobile G1 Launch Event*, sep. de 2008. dirección: [https://commons.wikimedia.org/wiki/File:T-Mobile\\_G1\\_launch\\_event\\_2.jpg](https://commons.wikimedia.org/wiki/File:T-Mobile_G1_launch_event_2.jpg) (visitado 26-06-2023).
- [10] E. Belinski, *Android API Levels*. dirección: <https://apilevels.com/> (visitado 26-06-2023).
- [11] E. Press, *Así se reparten Android e iOS el mercado global de sistemas operativos móviles*, (SCHE=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).
- [12] 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).
- [13] *Kotlin brand assets / Kotlin*, en-US. dirección: <https://kotlinlang.org/docs/kotlin-brand-assets.html> (visitado 27-06-2023).
- [14] R. Rao K, *The history of Kotlin - Kotlin for Enterprise Applications using Java EE [Book]*, en, ISBN: 9781788997270. dirección: <https://www.oreilly.com/library/view/kotlin-for-enterprise/9781788997270/ea4ec584-db64-4026-89a8-2086301eb9c5.xhtml> (visitado 26-06-2023).
- [15] *Kotlin for Android / Kotlin*, en-US. dirección: <https://kotlinlang.org/docs/android-overview.html> (visitado 26-06-2023).
- [16] *Enfoque de prioridad de Kotlin en Android / Android Developers*, es-419. dirección: <https://developer.android.com/kotlin/first?hl=es-419> (visitado 26-06-2023).
- [17] M. Braun, *Celebrating 5 years of Kotlin on Android*, en, ago. de 2022. dirección: <https://android-developers.googleblog.com/2022/08/celebrating-5-years-of-kotlin-on-android.html> (visitado 26-06-2023).
- [18] F. Lardinois, *Kotlin is now Google's preferred language for Android app development*, en-US, mayo de 2019. dirección: <https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development/> (visitado 26-06-2023).

- [19] A.-C. Bellini y N. Butcher, *Jetpack Compose is now 1.0: announcing Android's modern toolkit for building native UI*, en, jul. de 2021. dirección: <https://android-developers.googleblog.com/2021/07/jetpack-compose-announcement.html> (visitado 26-06-2023).
- [20] A. Leiva, *Qué es Jetpack Compose y cómo crear tu primer proyecto en Android*, es, sep. de 2021. dirección: <https://devexperto.com/jetpack-compose-que-es/> (visitado 26-06-2023).
- [21] *Programación imperativa vs declarativa: Google Jetpack Compose*, es-ES, mayo de 2021. dirección: <https://www2.deloitte.com/es/es/blog/todo-tecnologia/2021/programacion-imperativa-vs-declarativa-google-jetpack-compose.html> (visitado 26-06-2023).
- [22] C. Huamán, *Qué es y qué no es Android Jetpack*, en, jun. de 2018. dirección: <https://medium.com/orbismobile/qu%C3%A9-es-y-qu%C3%A9-no-es-android-jetpack-1248d1f36670> (visitado 05-07-2023).
- [23] *Recursos para desarrolladores de Android Jetpack - Android Developers / Desarrolladores de Android*, es-419. dirección: <https://developer.android.com/jetpack?hl=es-419> (visitado 05-07-2023).
- [24] I. Ramírez, *Qué es Material You y en qué se diferencia de Material Design*, es, Section: sistema-operativo, mar. de 2022. dirección: <https://www.xatakandroid.com/sistema-operativo/que-material-you-que-se-diferencia-material-design> (visitado 26-06-2023).
- [25] G. Singh, M. Design y D. Advocate, *Material Design 3 for Compose is now stable*, en, oct. de 2022. dirección: <https://material.io/blog/material-3-compose-stable> (visitado 26-06-2023).
- [26] *Material Design*, en. dirección: <https://m3.material.io/> (visitado 26-06-2023).
- [27] *Material Design Builder*. dirección: <https://m3.material.io/theme-builder#/dynamic> (visitado 26-06-2023).
- [28] A. Cerda, *Material Design 3: Novedades para el sistema de diseño de Google*, es-CL, mayo de 2022. dirección: <https://blog.ida.cl/disenio/material-design-3-novedades-sistema-disenio-google/> (visitado 26-06-2023).
- [29] C. Wilk, *Introducing Health Connect, a new API for Android app developers to securely access user health data*, en, mayo de 2022. dirección: <https://android-developers.googleblog.com/2022/05/introducing-health-connect.html> (visitado 26-06-2023).

- [30] R. Pandey, *Health Connect could be built right into Android 14*, en, Section: Operating Systems, feb. de 2023. dirección: <https://www.androidpolice.com/health-connect-built-right-into-android-14/> (visitado 26-06-2023).
- [31] *Health Connect / Desarrolladores de Android*, es-419. dirección: <https://developer.android.com/guide/health-and-fitness/health-connect?hl=es-419> (visitado 26-06-2023).
- [32] M. Ramírez, *Android 14 tendrá como eje central tus datos de salud y así arreglar el desajuste actual con tanta app*, es, Section: Actualizaciones Android, dic. de 2022. dirección: [https://www.lespanol.com/elandroidelibre/noticias-y-novedades/actualizaciones-android/20221219/android-central-datos-salud-arreglar-desajuste-actual/727177645\\_0.html](https://www.lespanol.com/elandroidelibre/noticias-y-novedades/actualizaciones-android/20221219/android-central-datos-salud-arreglar-desajuste-actual/727177645_0.html) (visitado 26-06-2023).
- [33] M. Rahman, *Android 14 has built-in support for Google and Samsung's Health Connect platform*, en, Section: Mobile, feb. de 2023. dirección: <https://www.xda-developers.com/android-14-health-connect-built-in/> (visitado 26-06-2023).
- [34] Freeyourgadget, *Gadgetbridge*, es-ES. dirección: <https://codeberg.org/Freeyourgadget/Gadgetbridge> (visitado 26-06-2023).
- [35] A. Malik, *Google's Health Connect platform is coming to Android 14 with new features*, en-US, mayo de 2023. dirección: <https://techcrunch.com/2023/05/10/googles-health-connect-platform-is-coming-to-android-14-with-new-features/> (visitado 26-06-2023).
- [36] *Lista de tipos de datos / Desarrolladores de Android*, es-419. dirección: <https://developer.android.com/guide/health-and-fitness/health-connect/data-and-data-types/data-types?hl=es-419> (visitado 27-06-2023).
- [37] R. Sáez, *Google lanza Health Connect, el sitio donde podrás gestionar todas tus apps de salud y fitness*, es, Section: Aplicaciones, nov. de 2022. dirección: <https://www.lavanguardia.com/tecnologia/aplicaciones/20221117/8608956/google-lanza-health-connect-sitio-podras-gestionar-todas-apps-salud-fitness-pmv.html> (visitado 26-06-2023).
- [38] *Preguntas frecuentes / Desarrolladores de Android*, es-419. dirección: <https://developer.android.com/guide/health-and-fitness/health-connect/frequently-asked-questions?hl=es-419> (visitado 27-06-2023).
- [39] A. P. Recio, *Persistencia de datos en Android con Room*, es, mar. de 2019. dirección: <https://www.adictosaltrabajo.com/2019/03/04/persistencia-de-datos-en-android-con-room/> (visitado 27-06-2023).

- [40] A. Leiva, *Room, la librería de Base de datos de Android*, es, jul. de 2020. dirección: <https://devexperto.com/room-la-libreria-de-base-de-datos-de-android/> (visitado 27-06-2023).
- [41] *Cómo guardar contenido en una base de datos local con Room*, es-419. dirección: <https://developer.android.com/training/data-storage/room?hl=es-419> (visitado 27-06-2023).
- [42] *Inserción de dependencias con Hilt | Desarrolladores de Android*, es-419. dirección: <https://developer.android.com/training/dependency-injection/hilt-android?hl=es-419> (visitado 06-07-2023).
- [43] A. Leiva, *Dagger Hilt: Cómo hacer inyección de dependencias en Android*, es, jun. de 2020. dirección: <https://devexpert.io/dagger-hilt/> (visitado 06-07-2023).
- [44] *KeyProperties*, en. dirección: <https://developer.android.com/reference/android/security/keystore/KeyProperties> (visitado 03-07-2023).
- [45] I. Kariyawasam, *Selecting the Best AES Block Cipher Mode (AES-GCM VS AES-CBC)*, en, mayo de 2021. dirección: <https://isuruka.medium.com/selecting-the-best-aes-block-cipher-mode-aes-gcm-vs-aes-cbc-ee3ebae173c> (visitado 03-07-2023).
- [46] J. Sena, *Using the Android Keystore system to store and retrieve sensitive information*, en, abr. de 2019. dirección: <https://medium.com/@josiassena/using-the-android-keystore-system-to-store-sensitive-information-3a56175a454b> (visitado 03-07-2023).
- [47] C. Ferry, *Protect your Room database with SQLCipher on Android*, en, nov. de 2020. dirección: <https://sonique6784.medium.com/protect-your-room-database-with-sqlcipher-on-android-78e0681be687> (visitado 03-07-2023).
- [48] *Navigating with Compose | Jetpack Compose*, en. dirección: <https://developer.android.com/jetpack/compose/navigation> (visitado 03-07-2023).
- [49] *Pass data between destinations*, en. dirección: <https://developer.android.com/guide/navigation/use-graph/pass-data> (visitado 03-07-2023).
- [50] *WorkManager*, en. dirección: <https://developer.android.com/reference/androidx/work/WorkManager> (visitado 04-07-2023).
- [51] *Arquitectura de apps: Capa de datos - Cómo programar tareas con WorkManager - Android Developers | Desarrolladores de Android*, es-419. dirección: <https://developer.android.com/topic/libraries/architecture/workmanager?hl=es-419> (visitado 04-07-2023).

- [52] *Retrofit*. dirección: <https://square.github.io/retrofit/> (visitado 04-07-2023).
- [53] G. Marosfalvi, *New way of storing data in Android — Jetpack DataStore*, en, ago. de 2021. dirección: <https://medium.com/supercharges-mobile-product-guide/new-way-of-storing-data-in-android-jetpack-datastore-a1073d09393d> (visitado 06-07-2023).
- [54] *Arquitectura de apps: Capa de datos - DataStore - Android Developers | Desarrolladores de Android*, es-419. dirección: <https://developer.android.com/topic/libraries/architecture/datastore?hl=es-419> (visitado 06-07-2023).
- [55] KeepCoding, *Usar DataStore en Android Studio | KeepCoding Bootcamps*, es, Section: Blog, ago. de 2022. dirección: <https://keepcoding.io/blog/using-datastore-in-android-studio/> (visitado 06-07-2023).
- [56] S. Sakunia, *Preference DataStore (The Generic Way)*, en, dic. de 2022. dirección: <https://proandroiddev.com/preference-datastore-the-generic-way-d26b11f1075f> (visitado 06-07-2023).
- [57] *Cómo localizar tu app | Desarrolladores de Android*, es-419. dirección: <https://developer.android.com/guide/topics/resources/localization?hl=es-419> (visitado 06-07-2023).
- [58] *Descripción general de los recursos de las apps | Desarrolladores de Android*, es-419. dirección: <https://developer.android.com/guide/topics/resources/providing-resources?hl=es-419> (visitado 06-07-2023).
- [59] Y. Berezanskyi, *Lingver*, original-date: 2019-10-18T12:02:25Z, jun. de 2023. dirección: <https://github.com/YarikSOffice/lingver> (visitado 06-07-2023).
- [60] E. Rubianes, *Lottie: qué es, para qué sirve y cómo utilizarlo en tu web*, es, Section: Diseño Web, dic. de 2021. dirección: <https://refrescandonegocios.com/lottie/> (visitado 06-07-2023).
- [61] *Lottie*, en-US. dirección: <https://airbnb.design/introducing-lottie/> (visitado 06-07-2023).
- [62] P. Goworowski, *Vico*. dirección: <https://patrykandpatrick.com/vico/wiki/> (visitado 06-07-2023).
- [63] P. Goworowski, *Vico Releases*, original-date: 2021-02-14T16:30:48Z, jul. de 2023. dirección: <https://github.com/patrykandpatrick/vico/releases> (visitado 06-07-2023).
- [64] P. Goworowski, *Vico Repository*, original-date: 2021-02-14T16:30:48Z, jul. de 2023. dirección: <https://github.com/patrykandpatrick/vico> (visitado 06-07-2023).

- [65] *The Python Logo*, en. dirección: <https://www.python.org/community/logos/> (visitado 27-06-2023).
- [66] A. Rodríguez, *Flask: minimalismo para el desarrollo web en Python*, es, ago. de 2014. dirección: <http://hipertextual.com/2014/08/flask-python> (visitado 27-06-2023).
- [67] *JSON And BSON*, en-us. dirección: <https://www.mongodb.com/json-and-bson> (visitado 27-06-2023).
- [68] *MongoDB Brand Resources*, en-us. dirección: <https://www.mongodb.com/brand-resources> (visitado 27-06-2023).
- [69] *¿Qué Es MongoDB?*, es. dirección: <https://www.mongodb.com/es/what-is-mongodb> (visitado 27-06-2023).
- [70] J. Rasmusson, *The Agile Samurai*, English. Dallas: The Pragmatic Bookshelf, 2010, ISBN: 978-1-934356-58-6. (visitado 24-10-2022).
- [71] 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).
- [72] 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.
- [73] 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.
- [74] 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.
- [75] W. Rui, C. Fanglin, C. Zhenyu et al., «StudentLife: Assessing Mental Health, Academic Performance and Behavioral Trends of College Students using Smartphones», *Ubicomp*, 2014.
- [76] *Cómo migrar Health Connect de Android 13 (APK) a Android 14 (framework)* / *Desarrolladores de Android*, es-419. dirección: <https://developer.android.com/guide/health-and-fitness/health-connect-guidelines/migrate/migrate-from-android-t-to-u?hl=es-419> (visitado 29-06-2023).
- [77] U. Gupta, *Null Values Imputation (All Methods)* / *Kaggle*, en. dirección: <https://www.kaggle.com/discussions/general/a> (visitado 29-06-2023).

# A.

## Cuestionarios para el seguimiento diario

---

### A.1 Inicio del día

#### Ansiedad

1. Me siento nervioso/a
2. Me siento angustiado/a
3. Me siento activo/a
4. Estoy preocupado/a

La respuesta a cada pregunta es un número entero en la escala de 0 a 10.

#### Depresión

1. Me siento triste
2. Me siento vacío/a
3. Me siento apático/a

La respuesta a cada pregunta es un número entero en la escala de 0 a 10.

## Suicidio

1. Tengo pensamientos de suicidio
2. En los últimos días, ¿has pensado seriamente en suicidarte?
3. ¿Existe alguna posibilidad de que pienses acabar con tu vida hoy o en los próximos días?

Las posibles respuestas a cada pregunta es sí o no.

## Soledad

1. Me siento solo/a
2. Me siento incomprendido/a
3. Me siento exclusivo/a
4. Me siento poco ayudado/a

La respuesta a cada pregunta es un número entero en la escala de 0 a 10.

## A.2 Final del día

### Ansiedad

1. Me he sentido nervioso/a
2. Me he sentido angustiado/a
3. Me he sentido activo/a
4. He estado preocupado/a

La respuesta a cada pregunta es un número entero en la escala de 0 a 10.



## Depresión

1. Me he sentido triste
2. Me he sentido vacío/a
3. Me he sentido apático/a

La respuesta a cada pregunta es un número entero en la escala de 0 a 10.

## Suicidio

1. He tenido pensamientos de suicidio
2. En el día de hoy, ¿has pensado seriamente en suicidarte?
3. ¿Existe alguna posibilidad de que pienses acabar con tu vida hoy o en los próximos días?

Las posibles respuestas a cada pregunta es sí o no.

## Soledad

1. Me he sentido solo/a
2. Me he sentido incomprendido/a
3. Me he sentido exclusivo/a
4. Me he sentido poco ayudado/a

La respuesta a cada pregunta es un número entero en la escala de 0 a 10.

## ¿Cómo has estado hoy?

1. ¿Has experimentado cambios en el apetito?

Posibles respuestas:

- Excesivamente alto
- Adecuado
- Excesivamente bajo

2. ¿Con cuánta energía te has notado?

Posibles respuestas:

- Alta
- Moderada
- Baja

3. ¿Cuál ha sido tu nivel de descanso?

Posibles respuestas:

- Satisfactorio
- Moderado
- Insuficiente

4. Tu nivel de concentración ha sido...

Posibles respuestas:

- Satisfactorio
- Adecuado
- Insuficiente

5. ¿Cuál ha sido tu nivel de libido?

Posibles respuestas:

- Satisfactorio
- Adecuado
- Insuficiente

6. ¿Cómo te has encontrado a nivel de dolor?

Posibles respuestas:

- Sin dolor
- Dolor moderado
- Dolor alto

## B.

# Recomendaciones

---

## B.1 Ansiedad

### Baja

Estupendo, sigue así.

### Moderada

#### Pauta 1

Hemos percibido que estás experimentando niveles moderados de ansiedad o estrés. Por ello, te recomendamos que planifiques un espacio en el día de hoy para hacer ejercicio físico.

El ejercicio físico puede disminuir el estrés por varias razones:

- Liberación de endorfinas: Durante el ejercicio, el cuerpo libera endorfinas, que son hormonas que actúan como analgésicos naturales y generan sensaciones de bienestar.
- Reducción de la hormona del estrés: El ejercicio regular puede disminuir los niveles de cortisol, la hormona del estrés.
- Mejora del sueño: El ejercicio regular puede promover un sueño más profundo y reparador.
- Distracción y enfoque: Participar en actividades físicas puede distraer la mente de las preocupaciones y tensiones diarias. Cuando te concentras en el ejercicio, tu mente se enfoca en la actividad física en lugar de en los problemas, lo que puede ayudar a reducir el estrés y proporcionar un descanso mental.

- Aumento de la confianza y la autoestima: El ejercicio regular puede ayudar a mejorar la confianza y la autoestima. Al establecer metas y lograr objetivos en el ámbito del ejercicio, puedes desarrollar una mayor sensación de logro y fortaleza personal. Esto puede ayudar a reducir el estrés al proporcionar una sensación de control y empoderamiento sobre tu vida.

## Pauta 2

En caso de que no dispongas de mucho tiempo, te proponemos una serie de alternativas.

- Trata de buscar un momento para ti, libre de estímulos estresantes. Puedes salir a dar un pequeño paseo, darte una ducha relajante, poner música y centrarte en escucharla durante unos minutos, practica unos estiramientos corporales... El objetivo es rebajar de forma rápida los niveles de ansiedad para poder retomar las tareas desde un estado emocional más adecuado.
- -Trata de eliminar algunos estímulos que puedan estar aumentando tu ansiedad: apaga el móvil a partir de determinada hora en la noche para tener unas horas libres de notificaciones antes de dormir, ponte unos cascos con música relajante para no escuchar el ruido de alrededor, si tienes pendiente tomar una decisión o discutir algo con alguien, aplázalo durante unas horas o días y permítete postponer los pensamientos al respecto, etc. El objetivo es eliminar los estímulos que están produciendo estrés para poder rebajar los niveles de ansiedad y así enfrentarnos de forma más adecuada a nuestros problemas o dificultades.

## Alta

Te proponemos que realices un ejercicio de respiración abdominal.

El objetivo de esta técnica es regular la respiración y, en consecuencia, disminuir la respuesta de activación fisiológica y la sensación de ansiedad.

Para ello, trata de llevar el aire hasta tu abdomen en cada inspiración para llenar tus pulmones en profundidad. Visualmente, deberías observar cómo tu tripa se hincha al llenarse de aire. Al expulsar el aire durante la espiración, el abdomen debería retornar a su posición habitual. Evita mover el pecho, los hombros o las clavículas, pues esto indica que el aire está llegando únicamente a la parte superior de los pulmones.

Inhala durante la ascensión de la curva y exhala durante el descenso. Trata de no hacerlo de forma demasiado profunda. Puedes repetirte mentalmente una palabra como calma o relax, puedes ima-

ginar que estás en un lugar tranquilo, o centrar tu atención en cómo el aire entra y sale y cómo la tensión se escapa con cada exhalación.

## B.2 Depresión

### Baja

Estupendo, sigue así.

### Moderada

Te proponemos que incluyas en tu día de hoy alguna actividad agradable o placentera.

Es posible que sientas que no tienes ganas o energía para hacerlas o, incluso, que aunque las hagas no lo disfrutarás. No obstante, es importante que entiendas que “las ganas se hacen”.

Esto significa que cuando nuestro estado de ánimo está un poquito bajo, si esperamos a experimentar ganas para hacer las cosas, probablemente nunca las hagamos. Esto a su vez hará que nuestro estado de ánimo disminuya todavía más, y entremos en un círculo vicioso en el que no haremos nada porque no tenemos ganas porque estamos tristes, y como no hacemos nada estaremos aún más tristes.

Para no caer en esta problemática te sugerimos que realices alguna actividad agradable, que no sea muy costosa y que te permita sentirte mejor. Pueden ser actividades que hagas tú solo/a o acompañado/a.

Aquí te dejamos algunas sugerencias.

- Dar un paseo al aire libre.
- Ir a comprar al supermercado y cocinar una receta que te guste.
- Ver una serie o película que te apetezca.
- Quedar con un amigo/a tomar algo o pasear.
- Leer un libro, escuchar música, dibujar...

## Alta

Te sugerimos que busques apoyo en las personas de tu alrededor.

Es muy beneficioso que puedas expresar cómo te estás sintiendo a otras personas. El mero hecho de contarlo supondrá un desahogo emocional que te hará sentirte mejor.

Además, las personas que te escuchen podrán comprender por lo que estás pasando y mostrar su empatía y apoyo. Es posible también que puedan tratar de ayudarte u ofrecerte consejos.

Para que todo vaya bien, trata de elegir a la persona adecuada en el momento adecuado. Busca a una persona que se encuentre bien, que no esté muy estresada u ocupada, que te haya mostrado su afecto en alguna ocasión...

Si necesitas desahogarte con frecuencia, trata de hacerlo con diferentes personas y no focalizarte solo en una, ya que prestar apoyo emocional en ocasiones puede resultar algo cansado.

Por último, ten cuidado de no caer en la queja: comunica tus emociones tratando de buscar soluciones y formas de sentirte mejor, en lugar de anclarte en el problema que ha sucedido.

## B.3 Riesgo de suicidio

### Bajo

Estupendo, sigue así.

### Moderado

Recuerda que, si en algún momento tienes pensamientos relacionados con el suicidio, es importante que pidas ayuda.

### Alto

Por lo que nos has contado, creemos que el riesgo de que puedas hacerte daño o de que acabes con tu vida es alto.

Por favor, acude cuanto antes a un servicio de emergencias o llama a los teléfonos 112 o 024.

Allí encontrarás personas que podrán comprender cómo te sientes y ayudarte a sentirte mejor.

Recuerda que el suicidio es la única opción que no tiene vuelta atrás. Trata de agotar otras posibles soluciones y pide ayuda para conseguirlo.

## B.4 Soledad

### Baja

Estupendo, sigue así.

### Moderada

Te sugerimos que busques apoyo en las personas de tu alrededor.

Es posible que consideres que no tienes a nadie con quién hablar. Sin embargo, si lo intentas, seguro que puedes encontrar personas deseosas de conversar contigo.

Busca a tu alrededor: un vecino con quien hayas tenido contacto, el camarero del bar o el restaurante al que vas en ocasiones, un compañero de trabajo...Puedes tratar de entablar una conversación preguntándoles cómo se encuentran y comentando cosas sobre el ambiente (el tiempo, los precios, los horarios, etc.).

Después, puedes intentar contar alguna anécdota o experiencia personal reciente. Por ejemplo, un programa de TV que te haya gustado, algo curioso que hayas visto recientemente en redes sociales, un plan al que tengas ganas de asistir o que hayas disfrutado si ya lo has hecho, etc.

Poco a poco, podrás progresar en la conversación y hablar más a menudo con estas personas.

### Alta

Te recomendamos que llames a Cruz Roja Te Escucha (900 107 917).

Cruz Roja te escucha es una iniciativa que trata de ofrecer acompañamiento y apoyo a personas que se encuentran en una situación de soledad no deseada.

Podrán facilitarte pautas para sentirte mejor y ofrecerte recursos en tu localidad que puedan servirte de apoyo.

Puedes contactar en el 900 107 917, de lunes a jueves laborales de 10h a 14h y de 16h a 20h (una hora menos en Canarias) y los viernes laborables de 10 a 14h (una hora menos en Canarias). La llamada es gratuita y confidencial.



