



**Universitat de les
Illes Balears**

TalkApp: Desarrollo de una aplicación de chat basada en chatRooms

MSc Candidate
Lamiaa Boucraa

A MSc thesis submitted to *Departament de Ciències Matemàtiques i Informàtica* of the University of Balearic Islands in accordance with the requirements for the degree of **Màster Universitari Enginyeria Informàtica (MINF)**

Author

Lamiaa Boucraa

MSc Supervisor

Cristina Manresa Yee

MINF Director

Antonia Mas Pichaco

01/04/2020

TalkApp: Desarrollo de una aplicación de chat basada en chatRooms

Lamiaa Boucraa

Tutor: Cristina Manresa Yee

Trabajo de fin de máster universitario Ingeniería Informática (MINF)

Universitat de les Illes Balears

07122 Palma de Mallorca

<lamiaa.boucraa@gmail.com>

Resumen

El objetivo de este trabajo, realizado en colaboración con el equipo de INeDITHOS, es describir el proceso seguido para el desarrollo de un prototipo de mensajería instantánea para cubrir la necesidad de comunicación del entorno social (docentes, familiares y sanitarios) de los afectados por las enfermedades raras, que en España superan los 3 millones de personas.

El desarrollo de esta aplicación se ha centrado en respetar la privacidad de los integrantes ofreciendo una solución tecnológica cuyo objetivo principal es facilitar la intervención inmediata de los profesionales, pero sin privarlos de su derecho a la privacidad. Por ello, se ha recurrido a la visualización mínima de los datos personales y a la restricción de la visibilidad entre los usuarios mediante chatrooms.

Abstract

The objective of this work, carried out in collaboration with the INeDITHOS team, is to describe the process followed to develop a prototype of instant messaging to cover the needs for communication of the social environment (teachers, family and healthcare) of those affected by rare diseases, which in Spain exceed 3 million people.

The development of this application has focused on respecting the privacy of the members, offering a technological solution whose main objective is to facilitate the immediate intervention of professionals, but without depriving them of their right to privacy. Therefore, we show the minimum visualization of personal data and the restriction of visibility among users through chatrooms.

Palabras clave: ChatRoom, Privacidad, Mensajería instantánea, Enfermedad rara, EERR, Visibilidad restringida

1. Introducción

Según la Organización Mundial de la Salud (OMS), existen cerca de 7.000 enfermedades minoritarias (EM) distintas, que afectan a quienes las padecen en sus capacidades físicas, en sus habilidades mentales y en sus cualidades sensoriales y de comportamiento. Se estima que en España existen más de 3 millones de personas con enfermedades poco frecuentes. Estas personas no solo tienen que hacer frente a los procedimientos médicos, sino también al impacto personal, escolar y social que les impone esta situación [16].

Saenz, Martín, García y Orelles [28] narran con las voces de los propios implicados, las relaciones entre los tres pilares principales en sus vidas: los familiares, los docentes y los sanitarios y evidencian que la naturaleza peculiar de los problemas que presentan requiere, en muchas ocasiones, de una intervención inmediata de especialistas sanitarios en las EERR para resolver las dudas o responder a las necesidades de las familias o docentes. Esta necesidad de coordinación se considera fundamental para favorecer la inclusión social de los menores y para la mejora de su bienestar.

El proyecto INeDITHOS [15], en el cual se enmarca este trabajo, es un proyecto del departamento de Pedagogía Aplicada y Psicología de la Educación de la Universidad de las Islas Baleares, cuya principal finalidad es trabajar para la mejora de la calidad de vida de niños y jóvenes con enfermedades raras. En el proyecto se desarrollan tres líneas de intervención: apoyo psicopedagógico a los afectados y sus familias, investigación para dar respuesta a las necesidades que se van detectando en este ámbito y formación del alumnado universitario que colabora en el proyecto utilizando la metodología de Aprendizaje-Servicio.

Este proyecto aborda el primer objetivo (apoyo psicopedagógico a los afectados y sus familias) ofreciendo una solución tecnológica a la necesidad de comunicación de los diferentes integrantes del entorno social (docentes, familiares y sanitarios) de los afectados por las enfermedades raras con el objetivo de contribuir en la mejora de su bienestar, pero sin privar al resto de personas implicadas de sus derechos de privacidad.

Esta solución tecnológica consiste en una aplicación de chat que permite la comunicación entre los docentes, sanitarios y

familiares de las personas con las EERR. Las características principales de esta aplicación y que la diferencian del resto de soluciones tecnológicas disponibles en el mercado son:

- El hecho de basarse en chatrooms: una forma de restringir la visibilidad y la conexión a los contactos deseados ocultando el resto.
- El respeto a la privacidad: no se tiene que facilitar ningún dato sensible (como es el teléfono, email o contacto de redes sociales) al resto de usuarios.

En la primera sección de este documento se describe la idea principal de la plataforma implementada, luego se definen las tecnologías utilizadas y los motivos de su elección, después se explica la arquitectura diseñada del sistema y de la base de datos, seguidamente se explican los detalles de desarrollo de la solución tecnológica adoptada y antes de concluir, se presentan los resultados de las pruebas realizadas con el prototipo.

2. Visión sobre la aplicación

En esta sección se describe la idea principal de la plataforma para cubrir las necesidades de coordinación y comunicación planteadas.

La plataforma de comunicación está diseñada e implementada acorde a las necesidades planteadas por el equipo de INeDITHOS que se basan en una larga trayectoria del proyecto en el campo (desde el 2003). La plataforma aborda la comunicación entre los diferentes perfiles implicados: docentes, familias y sanitarios, siendo esta coordinación más necesaria cuanto más compleja es la situación de la persona afectada por la ER. El tratamiento y la respuesta educativa a las enfermedades raras en cualquiera de los sistemas son complejos y habrá que abordarla desde su misma complejidad y muchas veces desde el desconocimiento, desconcierto o incertidumbre, por lo que no sólo la coordinación sino también las actitudes de los profesionales son fundamentales en su abordaje [28].

Sin embargo, esta actitud y coordinación no tiene que privarles de unos derechos básicos como la privacidad y la comodidad, ya que cada implicado tiene unas necesidades diferentes en la comunicación:

- **Familias de los niños/jóvenes** afectados por las enfermedades que necesitan apoyo por parte de los sanitarios sobretodo, pero también dan soporte a los docentes.
- **Docentes:** son los profesores de los alumnos con enfermedades raras que tienen que estar en contacto con la familia para poder hacerles preguntas sobre cada duda que les surja, ya que este tipo de enfermedades, como bien indica su nombre, es raro y no suele haber un patrón a seguir y cada caso tiene unas características propias que las diferencian de las demás.
- **Sanitarios:** Son profesionales que han tratado con las enfermedades raras, saben sus características especiales y pueden guiar tanto a las familias como a los docentes en su día a día. En muchas ocasiones, especialmente en caso de urgencias médicas, presentarse a un centro de salud,

no le soluciona el problema al afectado por la ER debido a la falta de formación en este tipo de enfermedades.

2.1. Concepto de la aplicación

Dadas las casuísticas especiales de comunicación en este escenario, utilizar una aplicación de las existentes en el mercado era inviable porque la privacidad es elemental en esta comunicación: dada la naturaleza del trabajo del personal sanitario, por ejemplo, facilitar datos sensibles como es el caso del número de teléfono, email de contacto o contacto de redes sociales le priva de un elemento fundamental, la privacidad.

La solución planteada se basa en la misma idea de las aplicaciones de chat ya existentes, pero con un nuevo concepto añadido: Chatrooms. Un chatroom es una forma de restringir la visibilidad y la conexión a los contactos deseados ocultando al resto, basándose en la 'sala' de chat asignada a cada uno de los usuarios de manera que solo se puedan ver y contactar entre sí las personas pertinentes a la misma 'sala de chat'. Esta gestión de chatrooms se realiza siguiendo la estructura de la figura 1:

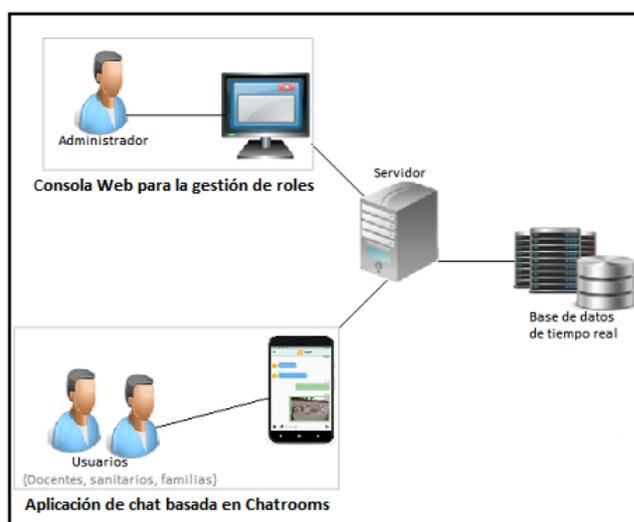


Figura 1: Estructura de la plataforma de comunicación

La solución tecnológica planteada se divide en dos secciones principales:

- **Una consola de administración:** mediante dicha consola, el administrador asigna una 'sala de chat' a los usuarios registrados de manera que todos los usuarios que tengan un interés común, puedan ver y conectarse entre sí.
- **Una aplicación de chat:** esta aplicación permite el intercambio de mensajes, imágenes y ficheros pdf entre los contactos que pertenecen al mismo chatroom. Por defecto, a todos los usuarios registrados en el sistema se les asigna una 'sala de chat' (llamada DEF) que les permite conectar con todos los usuarios de la misma 'sala' (comportándose como otras aplicaciones de mensajería instantánea) mientras el administrador no lo haya cambia-

do. Además, para la identificación de los usuarios solo se permite visualizar su nombre, apellido, foto y rol dentro de la aplicación (docente, sanitario o familia de los afectados por la ER) sin necesidad de exponer otra información privada.

3. Marco tecnológico

Esta sección trata el aspecto tecnológico de la aplicación y aborda las tecnologías que han sido elegidas para llevar a cabo su implementación y el motivo de cada elección.

3.1. Dart

Dart es un lenguaje de programación de código abierto, estructurado, flexible y orientado a objetos, basado en herencia simple y soporte de interfaces [19]. Gilad Bracha, uno de los creadores de este lenguaje junto con Lars Bak y Kasper Lund, considera que el punto más fuerte de este lenguaje es el tipado opcional de datos [18], lo cual significa que si el desarrollador lo desea, puede agregar tipos a las variables.

Este lenguaje se ha vuelto muy popular día tras día porque ha combinado las ventajas de dos de los lenguajes más utilizados: el dinamismo de Javascript con la potencia y estructura de lenguajes como C# o Java, para ofrecer una alternativa mucho más completa y con un rendimiento mucho mayor para el desarrollo de proyectos de gran envergadura [19].

Entre sus principales características encontramos:

- **La facilidad de aprendizaje:** Dart destaca con una sintaxis sencilla que recuerda a Javascript y C, pero con un potencial mucho mayor permitiendo desarrollar tareas complejas en un tiempo corto [19].
- **Portabilidad y flexibilidad:** En Dart se utiliza la misma filosofía que Java "*Write Once, run everywhere*". Dispone de una máquina virtual propia: DartVM [30], que le permite ejecutarse en cualquier sistema operativo.
- **Multiplataforma:** Dart se puede usar para programar tanto el servidor como el cliente desde cualquier aplicación [30].

3.2. Flutter

Flutter es un framework de código abierto y gratuito de Google para la creación de aplicaciones compiladas de forma nativa para dispositivos móviles (tanto con iOS como con Android) y Web desde una única base de código [6].

Este framework apareció como uno de los tres primeros preferidos por los desarrolladores en la encuesta anual publicada por StackOverflow [10]. Esta popularidad se atribuye a las ventajas que se enumeran a continuación:

- **Simplicidad:** Flutter se basa en el lenguaje de programación Dart, cuya principal característica es la sencillez de sintaxis en tareas complejas, además de basarse en una

arquitectura de widgets [32], que son soluciones preconstruidas y tienen asociadas un layout simple y se colocan normalmente en una estructura de árbol intuitiva.

- **Productividad:** gracias al '*Hot reload*', que permite hacer cambios en el código y ver el resultado en el emulador en milisegundos mientras la app sigue en marcha [32].
- **Alto rendimiento:** las apps de Flutter funcionan a 60 fps (*frames/segundo*) [7], además de que los widgets incorporan todas las diferencias críticas de cada plataforma, como es el scrolling, navegación, iconos y fuentes ofreciendo una percepción de rendimiento alto tanto en iOS como en Android.
- **Buena documentación:** Google proporciona una documentación detallada y completa, con ejemplos bien explicados para los casos básicos, y que hacen que el desarrollador se familiarice muy rápidamente con el framework.

3.3. Firebase

Firebase es una suite de Google en la nube que dispone de diferentes herramientas orientadas al desarrollo de aplicaciones Web y móviles [3]. Se destaca por:

- su adaptabilidad, ya que se integra fácilmente tanto para plataformas web como para aplicaciones móviles y es compatible con grandes plataformas como IOS, Android, aplicaciones web, Unity y C++.
- el hecho de tener agrupado una variedad de servicios en el mismo sitio [2], la convierte en una herramienta ideal para un prototipo.
- su escalabilidad: es una plataforma Cloud muy orientada a proyectos en crecimiento continuo.

A pesar de todas las ventajas que ofrece esta plataforma cloud, existe una desventaja que obliga a hacer un estudio de todos los posibles escenarios antes de optar por ella, sobretodo si se trata de un proyecto a gran escala ya que Google aplica unas tarifas según los servicios consumidos [4].

La figura 2 representa los servicios ofrecidos por Google Firebase [2]. En este proyecto, se han utilizado los servicios que se detallan a continuación:

- **Realtime Database:** Es una base de datos NoSQL alojada en la nube que permite la sincronización de datos de todos los clientes en tiempo real.
- **Authentication:** Es un servicio que gestiona las sesiones de la aplicación de forma simple y segura. Soporta la autenticación con plataformas de terceros (como Google, Facebook, Twitter, Github, etc.) o también mediante la creación de una cuenta con usuario y contraseña que se almacena en Firebase [12]. Esta última opción es la que se utiliza en TalkApp.
- **Cloud Storage:** Es un servicio para el almacenamiento de ficheros en la nube de forma segura [12]. El SDK de almacenamiento permite la subida y descarga de fiche-

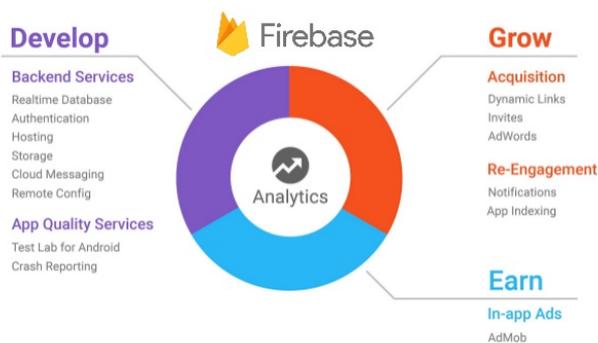


Figura 2: Servicios de Google Firebase

ros de cualquier extensión (audio, vídeo, imágenes, etc.) sin importar la calidad de la red deteniendo y reanudando automáticamente la transferencia a medida que la aplicación pierde y recupera la conectividad.

- **Cloud Functions:** Es un servicio que permite crear funciones escritas en JavaScript, en un entorno Node.js, que son invocadas por eventos de Firebase [12]. Se basa en el concepto de la computación en la nube FaaS (Functions as a Service) que permite dividir el Backend en microservicios que serían las diferentes funciones y cada una de ellas se encarga de una tarea concreta y responde a un evento [1].
- **Cloud Messaging:** Permite gestionar el envío de mensajes y notificaciones programadas acorde a diferentes parámetros de forma gratuita. Estos mensajes se pueden enviar a uno o varios dispositivos y también se pueden agrupar por temas o segmentos de usuarios [12].

3.4. Javascript

Javascript es un lenguaje de programación interpretado orientado a objetos, popular [11], basado en el estándar ECMAScript. Ha sido utilizado mayoritariamente en la parte del cliente, integrado en los navegadores Web para la creación de sitios web interactivos. Gracias a proyectos como Node.js, Electron y React Native, este lenguaje de scripts se ha extendido a entornos de servidores, programas de escritorio y aplicaciones móviles [25].

En este proyecto, Javascript se ha utilizado para la implementación de los microservicios encargados de la gestión de las notificaciones en la parte del servidor.

3.5. Node.js

Node.js es un entorno de tiempo de ejecución multiplataforma, de código abierto, basado en Javascript y diseñado para generar aplicaciones web de forma optimizada. Usa un modelo de E/S no bloqueante dirigido por un bucle de eventos y un único hilo ofreciendo una alta escalabilidad y un rendimiento importante en aplicaciones de tiempo real [27]: En comparación con las técnicas tradicionales de servicio web donde cada

conexión (que crea una solicitud) genera un nuevo hilo, ocupando la RAM del sistema, Node.js se basa en el tratamiento de las conexiones de forma unificada a partir de un único hilo complementado con un bucle de eventos (Event Loop) de tipo asíncrono, lo que le permite soportar decenas de miles de conexiones al mismo tiempo. Igualmente, viene con un ecosistema de paquetes llamado NPM que permite instalar con un comando todas las dependencias de un proyecto a través del archivo 'package.json'.

En este proyecto, se ha utilizado node para el hosting y el despliegue del proyecto utilizando la librería firebase-tools, además de la creación de la estructura de ficheros para las funciones encargadas de las notificaciones.

3.6. Git

Es un sistema de control de versiones distribuido (DVCS), de código abierto que permite guardar el historial de cambios de la aplicación pudiendo recuperar cualquier cambio en todas las etapas del desarrollo [8]. Se basa en un sistema de trabajo con ramas permitiendo abrir diferentes ramas de desarrollo a partir de la rama principal, una característica muy útil en proyectos desarrollados por varias personas o que tratan varios aspectos independientes que se pueden separar durante el desarrollo de manera que se pueden unificar al acabar.

Este proyecto se ha desarrollado de manera individual pero el uso de un controlador de versiones ha sido imprescindible para el control de los cambios en todas las etapas del proyecto, además de poder trabajar en el proyecto desde entornos diferentes (ordenador de casa, de la oficina, portátil) y de tenerlo todo en la nube para evitar incidentes.

4. Arquitectura

En esta sección, se explica la estructura del sistema utilizada para la plataforma desarrollada, además de explicar la arquitectura de base de datos y la tecnología usada para su implementación.

4.1. Estructura del sistema

La plataforma de comunicación diseñada para este proyecto se compone de varios elementos. Estos elementos junto con el flujo están representados en la figura 3. A continuación, se explican los diferentes componentes del sistema:

- **Backoffice:** Es una consola web gestionada por un administrador que se encarga del registro de los usuarios nuevos y de asignarles chatRooms por temática. En este caso, cada chatroom podría tener como temática un tipo de ER en concreto y juntar todas las personas que tratan (sanitarios), educan (docentes) o tienen relación (familiares) con los casos afectados por la ER, o bien podría tener como temática uno de los casos afectados por la ER. La decisión de la temática de los chatrooms se hace a criterio del administrador dependiendo del objetivo. En

este prototipo, no se ha implementado la página web para llevar a cabo esta gestión y se está gestionando directamente mediante la consola de firebase.

- **Aplicación móvil:** Es una aplicación de mensajería instantánea que está diseñada para ser utilizada por los usuarios potenciales de este proyecto y que son: los sanitarios, los docentes y los familiares de los jóvenes con enfermedades raras. Los detalles de la aplicación se detallarán en las siguientes secciones.
- **Base de datos:** La base de datos, como es lógico, es compartido entre el Backoffice y la app (esta sincronización se representa mediante las flechas rojas en la figura 3. Cualquier usuario registrado a través del backoffice, se verá en la aplicación en el chatroom que se le ha asignado. En la siguiente sección se explica la arquitectura de la base de datos usada.

4.2. Arquitectura de la base de datos

4.2.1. Uso de bases de datos NoSQL

La naturaleza del proyecto y la tipología de datos que se van a manejar ha llevado a utilizar una base de datos Nosql "Not just SQL". Las principales ventajas de este tipo de base de datos comparado con los sistemas de almacenamiento tradicionales [26] se describen a continuación:

- **El manejo de gran cantidades de datos y la escalabilidad horizontal:** la app se basa en el envío de mensajes en grandes cantidades y un sistema NoSQL permite crecer fácilmente utilizando el concepto de la escalabilidad horizontal. Este modelo implica tener varios servidores conocidos como **nodos** trabajando en una red de servidores, conocida como **Cluster** (mirar la figura 4).
- **La libertad de esquema:** Esta primera versión de la aplicación es una POC (prueba de concepto), el hecho de usar NoSQL nos ofrece mucha flexibilidad para hacer cambios en el modelo, cambiar sus características y crecer sin preocuparnos por el esquema de la base de datos.
- **La eliminación de los cuellos de botella y el uso eficiente de recursos:** A diferencia de los sistemas SQL que necesitan transcribir cada sentencia para poder ser ejecutada y utilizan sentencias complejas de varios niveles, NoSQL recurre a la desnormalización para realizar consultas simples de un único nivel o realizar JOIN mediante el Software, en la capa de aplicación. A pesar de todas las ventajas que tienen los sistemas NoSQL para este tipo de proyectos, la principal desventaja es la falta de estandarización, aunque grandes plataformas optan por ellas.

4.2.2. Modelo de base de datos

En este proyecto, la tecnología utilizada para la base de datos NoSQL es 'Firebase Realtime Database' que es una base de datos basada en documentos [20] alojada en la nube

que permite almacenar y sincronizar datos entre los usuarios en tiempo real [5]. En ausencia de un estándar de modelado de datos, existen varios artículos ([24], [31], [14], [21], [22], [23]) que presentan varias propuestas de modelado de bases de datos NoSQL. Para este artículo, nos basamos en el modelado presentado en el artículo [24] por ser el más intuitivo. La libertad de esquema en las bases de datos NoSQL hace que la mayoría de las restricciones se gestionen a nivel de código, de hecho, el diagrama de la figura 5 no es más que una representación básica cuyos detalles se explican a continuación basándose en la figura 3.

Los principales componentes de un chat son los usuarios, las conversaciones y los mensajes. Cada uno se representa mediante un documento ¹. La relación [M..F] entre usuario y conversación [24] significa que un usuario puede tener muchas conversaciones (M) pero una conversación pertenece a pocos usuarios (en este caso, se utiliza [M..F] porque solo se permiten conversaciones bidireccionales en esta versión²). A su vez, una conversación (1) contiene una infinidad de mensajes (S) [1..S], Esos mensajes los envía o los recibe un usuario [1..S] (no se han implementado las conversaciones grupales). Varios usuarios están alojados en un chatroom [M..1] y pueden conversar entre ellos. Los detalles de cada documento de los mencionados se presentan a continuación.

Un usuario se registra en el sistema guardando:

- Sus datos personales: nombre y apellido,
- Su rol: Indica si se trata de un familiar de los jóvenes con EM, de un docente o de un sanitario.
- El chatRoom: Si no se indica un chatroom para un usuario, se le asigna uno por defecto 'DEF'. Este prototipo se ha limitado a un único chatroom por cada usuario y por eso se ha tratado como una columna más del documento 'Usuario'.
- El token del dispositivo usado: Este token se obtiene directamente de los datos del dispositivo y se guarda para usarlo luego en el envío de las notificaciones.
- Imagen: Si así lo decide el usuario, puede cambiar su imagen de perfil cuya URL se guarda con sus datos y la imagen se guarda en un sistema de almacenamiento en la nube.

Como se puede ver en la figura 6, **un usuario** se indexa mediante un 'ID', y solo se crean las columnas cuyo valor no es nulo. Por ejemplo, en el caso de la imagen que es un valor opcional, solo se ha creado la columna 'imagenURL' en el caso del primer usuario. Ésta es una gestión automática de la base de datos de tiempo real de Firebase.

Como en todas las aplicaciones, **un mensaje** se caracteriza por tener (figura 7):

- un transmisor y un receptor: que se corresponden con la columnan 'from' y 'to' y guardan el identificador del

¹Un documento es el equivalente a una entidad en un modelo ER. Para más detalles, consúltese <https://www.infoq.com/articles/unified-data-modeling-for-relational-and-nosql-databases/>

²en el caso de las bases de datos NoSQL, el modelo cambia según las necesidades del sistema

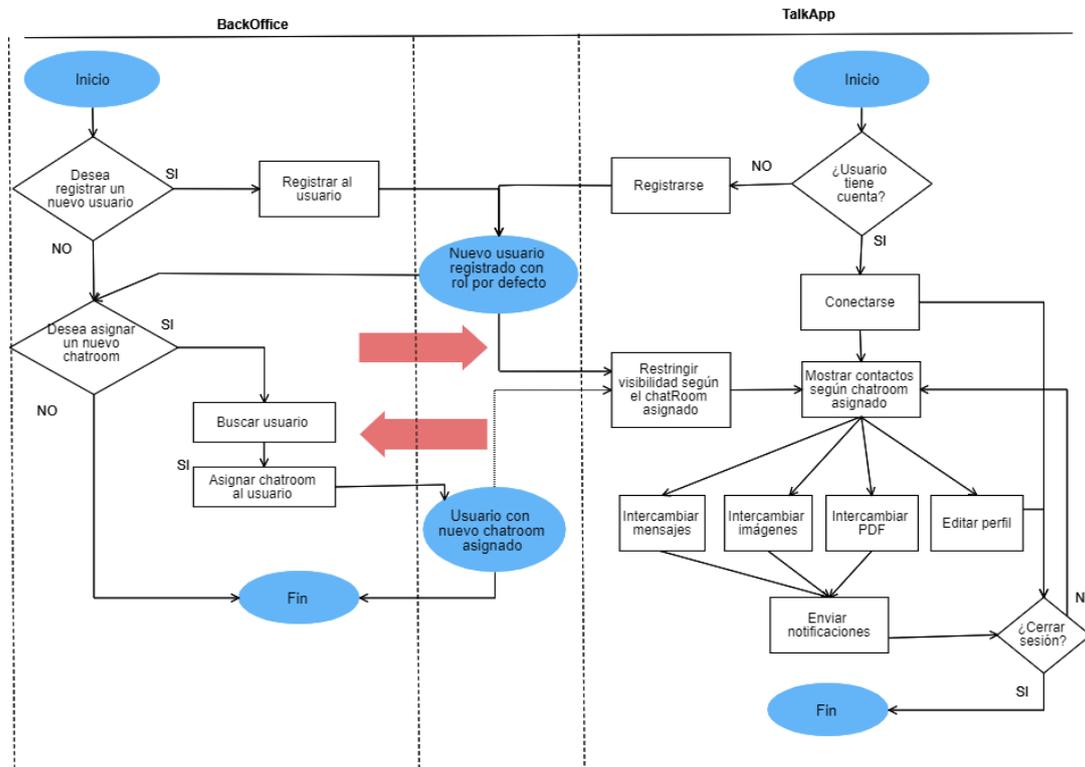


Figura 3: Diagrama que ilustra el flujo de datos del sistema



Figura 4: Escalabilidad horizontal

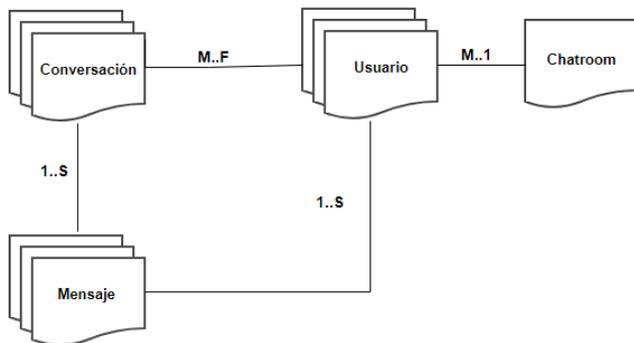


Figura 5: Modelo de la base de datos del sistema representado mediante documentos

```

users
├── dvS6erTvjrUQLCS3XDAzD4Ak4gR2
│   ├── imageUrl: "https://firebasestorage.googleapis.com/v0/b/ine
│   ├── name: "Lamiaa"
│   ├── role: "Familia"
│   ├── room: "DEF"
│   ├── surname: "Boucraa"
│   ├── token: "cJ_88f916Ns:APA91bHP4-ezGUj1g6cd-pyUXafw2N_YxYf
│   ├── uid: "dvS6erTvjrUQLCS3XDAzD4Ak4qf
├── wOG4IJRuvRaezCksP6ePPJLetOR2
│   ├── name: "Laura"
│   ├── role: "Sanitario"
│   ├── room: "FRN"
│   ├── surname: "Sastre"
│   ├── token: "cxP0jZxEQ4w:APA91bF9EJ94_0ufKkRm6YM_UmG1okcW_r
│   └── uid: "wOG4IJRuvRaezCksP6ePPJLetOf

```

Figura 6: Json que representa el documento del usuario en la base de datos

- usuario.
- un texto
- una fecha y hora de envío
- también los enlaces de las imágenes o pdf's enviados en caso de que existan

Las bases de datos NoSQL dejan mucha libertad al desarrollador para la construcción de la estructura. En este proyecto,

```

messages
- 5PycHt4rvcZ6KmPIBIsVZAzndzJ3+FxQrHIEor6MnkuNmgLa6NngclX93+
  - 1584886454796
    - dateString: "1584886454796"
    - from: "FxQrHIEor6MnkuNmgLa6NngclX93"
    - text: "Hola!!"
    - to: "5PycHt4rvcZ6KmPIBIsVZAzndz."
  - 1585429932407
    - dateString: "1585429932407"
    - from: "FxQrHIEor6MnkuNmgLa6NngclX93"
    - imageUrl: "https://firebasestorage.googleapis.com/v0/b/ine"
    - to: "5PycHt4rvcZ6KmPIBIsVZAzndz."
  - 1585431410398
    - dateString: "1585431410398"
    - from: "FxQrHIEor6MnkuNmgLa6NngclX93"
    - fileUrl: "https://firebasestorage.googleapis.com/v0/b/ine"
    - to: "5PycHt4rvcZ6KmPIBIsVZAzndz."

```

Figura 7: Json que representa el documento del mensaje en la base de datos

para que las consultas sean más eficientes, los mensajes se han indexado con los 'ID' de los usuarios transmisor y receptor y luego por la fecha y hora. De esta forma, no se tiene que recorrer el contenido de todos los mensajes para encontrar el transmisor y receptor buscados.

Estos mensajes se guardan dentro de **una conversación** que se actualiza después del envío de cada mensaje guardando:

- las relaciones entre usuarios (solo se guarda en la creación de la conversación),
- el último mensaje enviado
- la fecha y hora del último mensaje en milisegundos.
- el estado del último mensaje: para distinguir un mensaje leído 'N' de otro que no ha sido leído 'X' todavía.

La eficiencia en este tipo de sistemas de tiempo real es mucho más importante que el almacenamiento, por eso se ha optado por la estructura de la figura 8 para representar las conversaciones. Como se puede ver en la figura, la indexación se ha hecho de manera que podamos acceder a todas las conversaciones enviadas a un usuario con una simple consulta para evitar los anidados. Para representar las conversaciones de un usuario, bastaría con que realizar una consulta con su 'ID' para obtener todas las conversaciones. En el detalle de cada conversación (figura 9), se guardan también los datos del emisor por el mismo objetivo.

5. Diseño y Desarrollo

En esta sección, se describe cada uno de los módulos implementados para la aplicación TalkApp (cuyo nombre se ha elegido de manera que transmita su objetivo), las decisiones tomadas en cada caso y las tecnologías usadas para su desarrollo. El diseño se ha hecho en base a los requisitos establecidos y han sido aceptados y validados por el equipo de INEDIT-HOS.

```

conversations
- 5PycHt4rvcZ6KmPIBIsVZAzndzJ3 ID del usuario
  + FxQrHIEor6MnkuNmgLa6NngclX93
  + I8kLya8US3Pa4OMZErrjaYgFKPr1
  + fUGLD3SpGcgTWUzJ5yFpBdpAVRR2
  + gZ7gDf1MsnMkDwk63VIQizQdg2I2
  + ywYRK9kICHCd3km2zFPgjwsOIBw1
  - FxQrHIEor6MnkuNmgLa6NngclX93
    + 5PycHt4rvcZ6KmPIBIsVZAzndzJ3
    + I8kLya8US3Pa4OMZErrjaYgFKPr1

```

Figura 8: Json que representa el documento de la conversación en la base de datos. Los cuadros rojos representan el usuario receptor y los cuadros azules representan el usuario transmisor

```

conversations
- 28v7xtw8PMN0giPGIgjbcE4MRz2
  - GoPuUTC3Hs00f15bhEf9NChFdds2
    - dateString: "1591879158772"
    - imageUrl: "https://firebasestorage.googleapis.com/v0/b/ine"
    - last_message: "pensa que na lamiaa ho necesdita x info"
    - myId: "GoPuUTC3Hs00f15bhEf9NChFdds"
    - name: "Francisca"
    - role: "Docente"
    - room: "DEF"
    - state: "N"
    - surname: "Negre"
    - token: "d3gRW-1WFP4:APA91bE0VoT0eeAFiu4AUUnN6Uau_RDDgs"
    - uid: "GoPuUTC3Hs00f15bhEf9NChFdds"

```

Figura 9: Json que representa el detalle de la conversación en la base de datos.

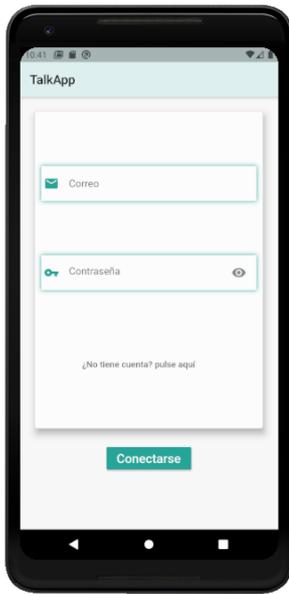


Figura 10: Pantalla de inicio de sesión

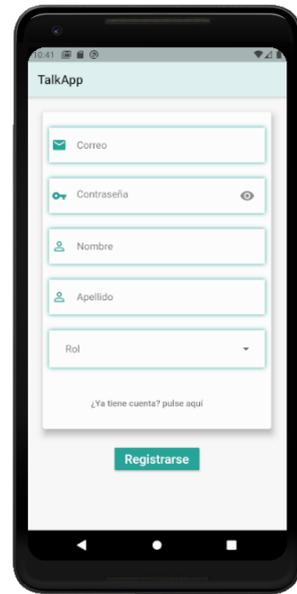


Figura 11: Pantalla de registro

5.1. Módulos

Para la implementación de cada uno de los módulos se ha utilizado el framework Flutter basado en el lenguaje Dart y el diseño se ha inspirado de las aplicaciones de mensajería instantáneas más usadas y ha sido adaptado a los criterios del equipo INeDITHOS que son los que trabajan con los afectados por las EERR y saben sus necesidades. A continuación, se explican los detalles de cada módulo.

5.1.1. Autenticación y registro

Para la autenticación, se conecta al servicio de backend 'Firebase Authentication' a través de la pantalla de la figura 10 que permite al usuario ya registrado conectarse mediante su correo y su contraseña. El contenido de la contraseña se oculta a menos que el usuario indique lo contrario mediante el icono de la derecha del campo.

Si el usuario no está registrado, puede acceder a la pantalla de registro (figura 11) mediante el enlace que sale debajo de los campos de las credenciales. En esta pantalla, además del correo y la contraseña se añaden los campos de nombre, apellido y rol en la aplicación (figura 12) que sirven para identificar a la persona.

Una vez registrado, la información queda grabada en 'Firebase Realtime Database' y el usuario accede a las pantallas de la aplicación que se dividen en dos partes principales:

- la cabecera: que contiene un menú desplegable que permite al usuario cerrar la sesión (figura 13).
- y el cuerpo: que está constituido de una barra de pestañas, entre las cuales se puede mover o bien mediante gestos o directamente pulsando encima de la pestaña y cuyo contenido se explica en las siguientes secciones.



Figura 12: Desplegable de los roles de la aplicación

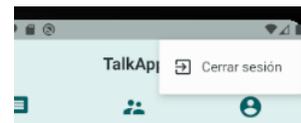


Figura 13: Opción de cierre de sesión en el menú desplegable

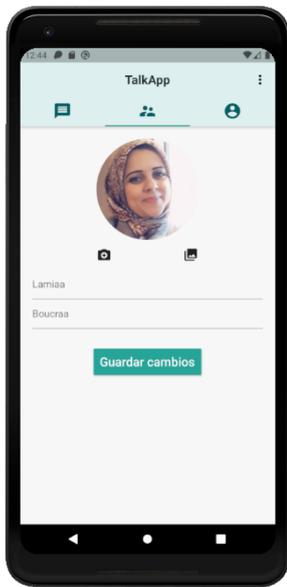


Figura 14: Pantalla del perfil del usuario

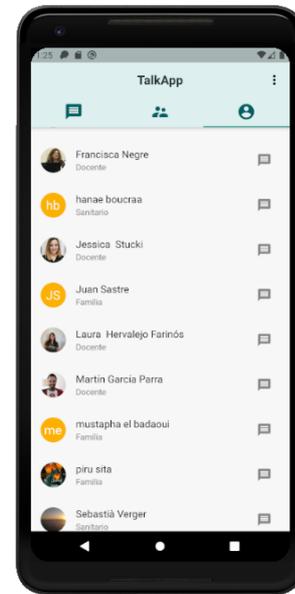


Figura 15: Pantalla de los contactos del usuario

5.1.2. Perfil

La pantalla de perfil (figura 14) permite al usuario consultar o modificar sus datos personales. Está constituida por:

- La foto del usuario: Antes de tener ninguna foto, salen los iniciales del nombre del usuario. Si éste quiere cambiar la foto, lo puede realizar mediante los iconos que salen debajo del espacio reservado para ello. Los iconos permiten hacer una foto al momento o escoger una foto a existente de la galería del móvil.
- El nombre y apellido: Son campos editables que pueden ser modificados en cualquier momento por el usuario.
- Un botón para guardar cambios: Una vez pulsado, en caso de que existan cambios, avisa al usuario al acabar el registro de los datos. En caso de que no haya cambios, también avisa al usuario que no ha habido cambios para ofrecerle feedback.

5.1.3. Contactos

La pantalla de contactos contiene todos los contactos pertinentes al mismo chatroom del usuario ordenados alfabéticamente. Cada fila de la lista de contactos (15), muestra la foto de los usuarios (o las iniciales de su nombre), su nombre completo, su rol en la aplicación y a la derecha un icono que una vez pulsado nos lleva a la pantalla de chat para iniciar una conversación o seguir una ya existente.

5.1.4. Mensajes y conversaciones

La pantalla de conversaciones (figura 16) es la pantalla principal a la cual se accede una vez iniciada la sesión y contiene todas las conversaciones del usuario. Cada una se representa mediante la foto del contacto implicado (o las iniciales de su

nombre), su nombre completo, el último mensaje enviado o recibido, y la fecha u hora de la última interacción. Se ordenan por orden de llegada del último mensaje y se marcan en negrita las que todavía no se han podido leer. Una vez pulsada, la conversación cambia de estado convirtiéndose en 'leída' y se accede a la pantalla de chats.

Estas conversaciones son bidireccionales dentro del mismo chatroom cuya idea es restringir la visibilidad a agrupaciones de personas con un interés común. Sin embargo, cabe la posibilidad de crear chats grupales (es un requisito planteado para la siguiente versión) en los cuales se pueden tratar temas conjuntamente.

5.1.5. Envío de mensajes

En una conversación, el usuario puede intercambiar mensajes, imágenes, y pdf's. La figura 17 muestra la pantalla de mensajes desde el punto de vista de los dos implicados en la comunicación. En la cabecera de la pantalla, se muestra el contacto con el cual se intercambian los mensajes para tenerlo presente en cada momento. El cuerpo de la pantalla contiene todos los mensajes en los cuales se han utilizado colores y posiciones diferentes para distinguir al usuario conectado de su contacto. El pie de la pantalla contiene el área de mensajes que está constituida por:

- Un área de texto: que permite redactar el mensaje.
- Un botón (icono de cámara) para hacer fotos y enviarlas.
- Un botón (icono de galería) para enviar fotos de la galería del terminal.
- Un botón (icono de adjunto) para el envío de ficheros pdf.
- Un botón para el envío del mensaje al contacto.

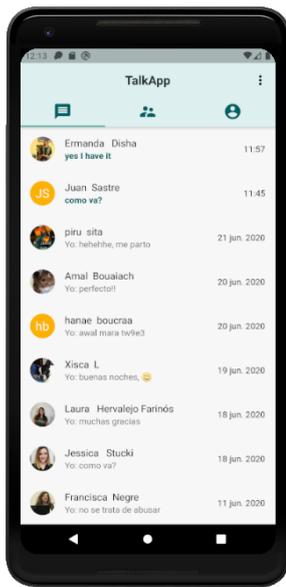


Figura 16: Pantalla de las conversaciones del usuario

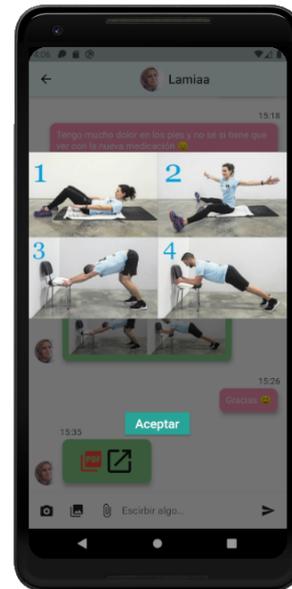


Figura 18: Imagen ampliada

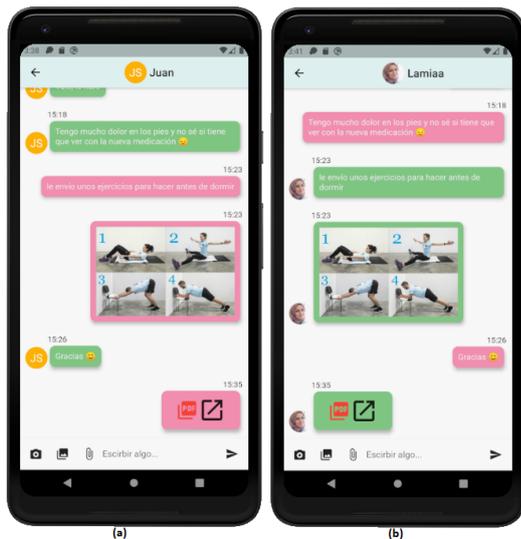


Figura 17: Pantalla de chat

Una imagen enviada puede ser ampliada si se pulsa encima (figura 18). De la misma forma, se puede consultar el contenido del pdf (figura 19).

Para que el acceso a los mensajes sea eficiente, las imágenes y pdf's se han guardado en el repositorio 'Firebase Cloud Storage' y se ha limitado a guardar sus URL's en la base de datos.

5.1.6. Notificaciones

Las notificaciones son uno de los elementos más importantes de una aplicación de chat. Para su implementación, se ha utilizado la tecnología 'Firebase Cloud Messaging' gestionada automáticamente con 'Firebase Cloud Functions' según el



Figura 19: Consulta del contenido del pdf



Figura 20: Ejemplo de una notificación

siguiente proceso (figura 20) :

- Cuando un usuario envía un nuevo mensaje, éste se guarda en la base de datos de tiempo real 'Realtime Database'.
- Un trigger de 'Firebase Cloud Functions' salta al detectar este nuevo envío y ejecuta el microservicio previamente programado para el envío de notificaciones. Este microservicio recibe como entrada el emisor, el destinatario y el token del terminal destino y llama a la plataforma de 'Cloud Messaging' para enviar la notificación push.
- Una vez enviada la notificación push. El destinatario podrá consultar el nuevo mensaje recibido pulsando sobre la notificación.

'Firebase Cloud Messaging' permite agrupar las notificaciones mediante tags. En el caso de TalkApp, el tag de agrupación de mensajes es el receptor de manera que cada vez que se genera un mensaje y por consiguiente una notificación al mismo receptor, se pone en el mismo contenedor y se evita la duplicidad de notificaciones en el terminal del usuario.

La notificación está formado por el logo y el nombre de la aplicación, la hora del envío, el emisor y los primeros 100 caracteres del mensaje. En caso del envío de fotos o imágenes, solo se le notifica al usuario de que se ha realizado en envío para que lo pueda consultar en la aplicación.

6. Pruebas con la aplicación

Este apartado analiza el resultado de las pruebas realizadas con usuarios para evaluar la usabilidad del producto implementado. En este prototipo, se ha limitado a probar la aplicación en el sistema operativo Android, aunque el framework utilizado nos permite construir una versión iOS a partir del mismo código.

6.1. Pruebas y método de evaluación

En la fase final del proyecto, una vez realizados los tests primarios para validar el correcto funcionamiento de la aplicación, se ha generado una versión de prueba en Android (según el proceso del Apéndice B.) que ha sido enviada a varios usuarios para validar su correcto funcionamiento y para detectar posibles errores. La versión creada es compatible con versiones de Android superiores a Android 5.0 (Lollipop - API level 21). Para el análisis y la evaluación de estas pruebas, se

ha diseñado una encuesta de usabilidad (mirar Apéndice A) basada en SUS (System Usability Scale) [13], un cuestionario genérico y corto creado por John Brooke. Diferentes pruebas y tests han demostrado que los resultados obtenidos a partir del SUS suelen ser fiables y acertados [29], razón por la cual es uno de los métodos de medición de usabilidad más utilizados en Experiencia de Usuario. El cuestionario está formado por diez preguntas (mencionadas en el apartado de Preguntas SUS del Apéndice A) con respuesta dentro de una escala de Likert de 5 puntos y ofrece un resultado entre 0 y 100.

Además de las preguntas SUS qué sirven para tener una visión general de la usabilidad del sistema, se han diseñado preguntas directamente relacionadas con las funcionalidades más importantes de la aplicación para tener una opinión más específica sobre cada una de ellas y realizar modificaciones en base a ello.

A continuación, se explican los resultados obtenidos a partir de las pruebas y encuestas realizadas.

6.2. Análisis de los resultados de las encuestas

6.2.1. Datos demográficos

Trece participantes (5 hombres, 8 mujeres) cuyas edades estaban comprendidas entre 24 y 55 años (media: 37.3 años, desviación estándar: 11.5) participaron en la evaluación del sistema.

Todos los participantes eran voluntarios. Seis de ellos eran miembros del equipo de INeDITHOS que dan apoyo a los niños y jóvenes con EERR y conocen de primera mano sus necesidades de comunicación.

6.3. Análisis del cuestionario

Todos los participantes en las pruebas han declarado haber utilizado alguna aplicación de mensajería instantánea para comunicarse, lo que les ha ayudado a utilizar TalkApp sin problemas.

Se han utilizado una variedad de versiones de Android en las pruebas donde la mínima fue 'Lollipop 5.0 - 5.1.1' y la máxima 'Android 10'. Durante las pruebas, se ha agrupado a un conjunto de usuarios en chatrooms separados para probar y evaluar las distintas funcionalidades cuyas valoraciones se presentan a continuación.

La valoración de la usabilidad del sistema a partir del cuestionario SUS (apartado de Preguntas SUS del Apéndice A) por parte de los participantes dio lugar a una media de 85.58 con una desviación estándar de 7.6 obteniendo así una calificación final contenida en el rango [77.91, 93.25]. Este resultado reflejado en la escala de valores SUS (figura 21) significa que la usabilidad del sistema implementado está en el rango aceptable y con una calificación adjetiva entre buena y excelente. Además, la mayoría de los usuarios de la muestra no han encontrado dificultades ni incoherencias en el sistema.

Según el trabajo [29], los resultados del método SUS son fiables, pero para disponer de una fuente de información más

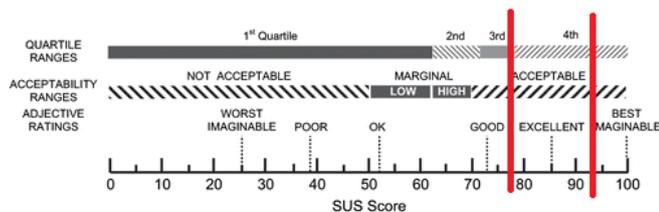


Figura 21: Escala de valores SUS

¿Has podido crear una cuenta rápidamente en la aplicación ?

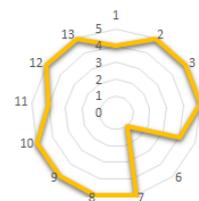


Figura 23: Opiniones de los usuarios sobre la creación de la cuenta en el sistema

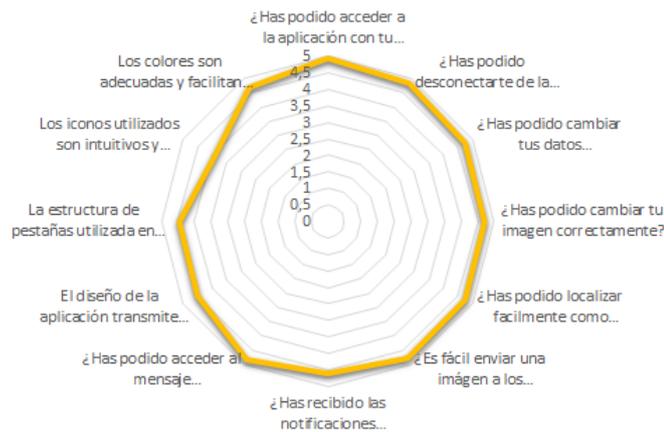


Figura 22: Promedio de las opiniones de los usuarios sobre los diferentes aspectos del sistema

completa, hemos ampliado este cuestionario a preguntas más específicas que se analizan a continuación.

La figura 22 ofrece una visión general (promedio) de las opiniones de los usuarios en las funcionalidades y aspectos del sistema planteados en la encuesta y permite destacar los puntos que requieren mejoras en la aplicación. Observando el gráfico podemos deducir que la valoración de los usuarios a los diferentes aspectos del sistema está contenida en el rango [4,5], lo que significa que la satisfacción es relativamente alta menos en un caso. Varios usuarios encuentran que los iconos no son lo más intuitivos y este punto se va a tener en cuenta en las siguientes versiones del sistema.

Analizando individualmente las preguntas planteadas, se ha observado un patrón diferente al resto en la pregunta correspondiente a la creación de las cuentas en la aplicación que se representa mediante la figura 23. Dado que es una opinión aislada y debido a que la muestra es pequeña (13 usuarios), podría tratarse de un error del terminal o un error en el uso de la aplicación que no podemos analizar por falta de información.

En general, los resultados de la encuesta reflejan un grado de satisfacción relativamente alto. Sin embargo, algunos usuarios han propuesto varias funcionalidades que les gustaría tener en los siguientes evolutivos y que se explican en la siguiente sección.

6.4. Sugerencias de los usuarios y trabajo futuro

6.4.1. Sugerencias de los usuarios

En el cuadro 1, se presentan las sugerencias de mejora más importantes propuestas por los usuarios en el cuestionario junto con algunas observaciones sobre la importancia de éstas en la mejora de la experiencia del usuario del sistema.

6.4.2. Trabajo futuro

A parte de las sugerencias mencionadas anteriormente, un punto muy importante que se puede tener en cuenta en versiones futuras es la ampliación de la aplicación a varios chatrooms, permitiendo a los profesionales cambiar de un chatroom a otro según el caso tratado. Es decir que cada sanitario o docente podría tener una lista de chatrooms en el menú desplegable con el nombre de los casos tratados. De esta manera, sería más sencillo y más ágil tratar a diario con los jóvenes con EERR y sus familias.

Otro punto a tener en cuenta en las versiones futuras es el envío de los ficheros pdf ya que solo se ha podido probar esta funcionalidad en los simuladores por un problema de compatibilidad que impide visualizar el fichero en los terminales.

7. Conclusión

La finalidad de este trabajo es describir el proceso del desarrollo de una aplicación de mensajería instantánea propuesta por el equipo de INeDITHOS y destinada a la comunicación de los familiares de las personas con enfermedades minoritarias y de los profesionales que tratan con ellos: sanitarios y docentes.

El principal objetivo de la aplicación es ofrecer a los implicados un entorno privado en el cual pueden comunicarse de forma profesional sin sentirse intimidados facilitando sus datos personales. En consecuencia, se ha enfocado al uso de la mínima cantidad de datos para el registro, además del uso de chatrooms para restringir la visibilidad dividiendo los usuarios en salas de chats aisladas.

Para el desarrollo de esta aplicación, se han utilizado varias tecnologías como: Flutter destinado a aplicaciones móviles,

| Sugerencia | Observaciones |
|---|---|
| Opción de restablecer las contraseñas | Esencial en el sistema |
| Especificación de las características de las contraseñas | No se han puesto restricciones debido al hecho de ser un prototipo |
| Traspaso y descarga de archivos de varios formatos (no solo las imágenes) | Es una opción ya contemplada en la primera versión y no se ha liberado debido a problemas de compatibilidad |
| Explicación sobre la privacidad de los datos guardados en el sistema | Es esencial para fiabilidad del sistema |
| Envío de audio | Es importante en caso de usuarios con discapacidad |
| Llamadas y videollamadas | Descartada porque no concuerda con el propósito del sistema |
| Citación de mensajes anteriores en las respuestas | Es importante en aplicación de chat |
| Manual de uso de la aplicación | Bastará con las indicaciones en la descripción del 'playstore' |
| Adjudicar colores diferentes según el rol (docente, familia o sanitario) en caso de que no haya foto en el perfil del usuario | Ayudará a los usuarios a identificar a simple vista los roles en cada conversación |
| Crear grupos de conversaciones | Es un requisito previsto para las siguientes versiones. |
| Cifrado de mensajes | En un contexto como el que trata la aplicación, la privacidad y la seguridad son primordiales |
| Uso de emojis | Es un aspecto que proporcionan los teclados propios de varios terminales. |

Cuadro 1: Sugerencias de nuevas funcionalidades reportadas por los usuarios

servicios de Firebase, bases de datos NoSQL...cuya elección se ha hecho en base a las necesidades del sistema y a su facilidad de evolución en el futuro.

Las pruebas de la aplicación no se han podido realizar in situ para obtener un feedback más realista debido a las condiciones vividas en la actualidad por el COVID-19 y se ha limitado a hacer una simulación con varios usuarios voluntarios. Algunos de ellos forman parte del equipo de INeDITHOS (el proyecto en el cual se enmarca esta aplicación) que trata y da apoyo a los niños y jóvenes con EERR.

Finalmente, el póster «Aplicación de Chat con Chatrooms. Comunicación de Docentes, Sanitarios y Familias de Niños con Enfermedades Minoritarias» que explica el proyecto [17], ha sido aceptado para ser presentado en la 5ª Edición del congreso IPT 'Informática para tod@s' (Valencia, 24-25 Abril 2020) [9]. Este congreso no ha tenido lugar en las fechas previstas por las circunstancias actuales a causa del COVID-19 y se ha aplazado al año 2021.

Referencias

- [1] FaaS (Function-as-a-Service). <https://www.ibm.com/cloud/learn/faas>. Accessed: 2020-05-04.
- [2] Firebase: Backend As Service. <https://www.linkedin.com/pulse/firebase-backend-service-rajat-singh/>. Accessed: 2020-05-04.
- [3] Firebase expands to become a unified app platform. <https://firebase.googleblog.com/2016/05/firebase-expands-to-become-unified-app-platform.html>. Accessed: 2020-05-04.
- [4] Firebase: Planes de precios. <https://firebase.google.com/pricing>. Accessed: 2020-05-04.
- [5] Firebase Realtime Database. <https://firebase.google.com/products/realtime-database>. Accessed: 2020-06-01.
- [6] Flutter. <https://flutter.dev/>. Accessed: 2020-05-04.
- [7] Flutter performance profiling. <https://flutter.dev/docs/perf/rendering/ui-performance>. Accessed: 2020-05-04.
- [8] Git. <https://git-scm.com/>. Accessed: 2020-05-04.
- [9] Informática para tod@s. <https://ipt.acm.org/acerca-de/>. Accessed: 2020-06-01.
- [10] Most Loved, Dreaded, and Wanted Other Frameworks, Libraries, and Tools. https://insights.stackoverflow.com/survey/2019#technology_-most-loved-dreaded-and-wanted-other-frameworks. Accessed: 2020-05-04.
- [11] Most Popular Technologies. <https://insights.stackoverflow.com/survey/2019>. Accessed: 2020-05-04.
- [12] Plataforma de desarrollo de aplicacio-

- nes. https://firebase.google.com/?hl=es-419gclid=EAIaIQobChMI942hRwb6QIVVZ3VCh0p7wQgEAAYASAAEgIa7vD_BwEg.
- [13] System Usability Scale (SUS). <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>. Accessed: 2020-06-01.
- [14] A. AIT BRAHIM, R. TIGHILT FERHAT, and G. ZURFLUH. Extraction process of conceptual model from a document-oriented nosql database. In *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–5, 2019.
- [15] Francisca Bennasar and Sebastià Gelabert. Ineditos - un proyecto de pedagogía hospitalaria dedicado a la mejora de la calidad de vida de niños y jóvenes con eerr a partir de la intervención e investigación con voluntariado universitario. 23:107–119, 01 2017.
- [16] Sebastià Verger Gelabert; Francisca Negre Bennásar. *Soluciones educativas para la mejora de la calidad de vida del niño con enfermedad minoritaria desde una intervención innovadora y transdisciplinar*. Programa Estatal d’R+D+i Orientada als Reptes de la Societat: Projectes d’R+D. Ministerio de Economía y Competitividad (MINECO)., 2016-2020.
- [17] L. Boucraa and C. Manresa-Yee. Aplicación de chat con chatrooms. comunicación de docentes, sanitarios y familias de niños con enfermedades minoritarias. In *V Edición Informática para tod@s*, 2020.
- [18] Gilad Bracha (Foreword by Erick Meijer). *The Dart Programming Language*. Pearson Education, Inc. 2015953614, 2016.
- [19] Chris Buckett (Foreword by Seth Ladd). *Dart IN ACTION*. Manning Publications Co., 2013.
- [20] H. Hashem and D. Ranc. Evaluating nosql document oriented data model. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pages 51–56, 2016.
- [21] H. Hashem and D. Ranc. Evaluating nosql document oriented data model. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pages 51–56, 2016.
- [22] H. Hashem and D. Ranc. Evaluating nosql document oriented data model. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pages 51–56, 2016.
- [23] H. Hashem and D. Ranc. Evaluating nosql document oriented data model. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, pages 51–56, 2016.
- [24] A. A. Imam, S. Basri, R. Ahmad, N. Aziz, and M. T. González-Aparicio. New cardinality notations and styles for modeling nosql document-store databases. In *TENCON 2017 - 2017 IEEE Region 10 Conference*, pages 2765–2770, 2017.
- [25] Fernando Luna. *JavaScript - Aprende a programar en el lenguaje de la web*. Ciudad autónoma de Buenos Aires, 2019.
- [26] S. Palanisamy and P. SuvithaVani. A survey on rdbms and nosql databases mysql vs mongodb. In *2020 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–7, 2020.
- [27] Sandro Pasquali. *Mastering Node.js*. Pact Publishing Ltd, November 2013.
- [28] García y Orelles(cood.) Saenz, Martín. *La innovación escolar desde la perspectiva de personas con enfermedades raras en el País Vasco: Historias de vida, prácticas escolares, necesidades del sistema educativo y propuestas de mejora para una escuela y sociedad inclusiva*. Bilbao: FEDER Euskadi G-91018549, 2016.
- [29] JEFF SAURO and JAMES R.LEWIS. *Quantifying the user experience. Practical statistics for user research*. 2012.
- [30] Kathy Walrath and Seth Ladd. *What is Dart?* O’Reilly Media, Inc., 2012.
- [31] Allen Wang. Unified Data Modeling for Relational and NoSQL Databases. <https://www.infoq.com/articles/unified-data-modeling-for-relational-and-nosql-databases>. Accessed: 2020-06-01.
- [32] Eric Windmill. *Flutter in Action*. Manning, January 2020.

Apéndice A. Encuesta de usabilidad

Encuesta sobre la usabilidad de la aplicación TalkApp

TalkApp es una aplicación de mensajería que permite la comunicación entre los docentes, sanitarios y familiares de las personas con Enfermedades Minoritarias.

Fecha: Indique la fecha de cumplimentación del formulario

| | | | |
|-----------|----------------------------|------|------------------|
| Nombre | Escriba aquí su nombre | | |
| Apellidos | Escriba aquí sus apellidos | | |
| Edad | Edad | Sexo | Elija una opción |

| |
|---|
| Qué marca de móvil has usado en la prueba |
| Escriba aquí la marca |

| |
|---|
| Qué marca de Android has usado en la prueba |
| Elija una opción. |

| | |
|---|--|
| ¿Has utilizado anteriormente alguna aplicación de mensajería en el móvil? | <input checked="" type="radio"/> Sí <input type="radio"/> No |
|---|--|

Opinión del uso del sistema

A continuación, valore los siguientes aspectos de la aplicación de mensajería con uno de los siguientes valores y ponga las observaciones oportunas para explicar el problema:

- 1: Totalmente en desacuerdo
- 2: Muy desacuerdo
- 3: Poco desacuerdo
- 4: De acuerdo
- 5: Totalmente de acuerdo

| | Respuesta: Hay que indicar una respuesta de las que salen en el desplegable | Observaciones |
|---|--|---------------|
| Preguntas pantallas de registro y acceso a la aplicación | | |

| | | |
|---|------------------|--|
| ¿Has podido crear una cuenta rápidamente en la aplicación? | Elija una opción | |
| ¿Has podido acceder a la aplicación con tu usuario y contraseña fácilmente? | Elija una opción | |
| ¿Has podido desconectarte de la aplicación correctamente? | Elija una opción | |
| Preguntas sobre la pantalla del perfil | | |
| ¿Has podido cambiar tus datos correctamente? | Elija una opción | |
| ¿Has podido cambiar tu imagen correctamente? | Elija una opción | |
| Preguntas Mensajería | | |
| ¿Has podido localizar fácilmente como enviar un mensaje a una persona entre los contactos? | Elija una opción | |
| ¿Es fácil enviar una imagen a los contactos? | Elija una opción | |
| ¿Has recibido las notificaciones correctamente cuando te han enviado un mensaje? | Elija una opción | |
| ¿Has podido acceder al mensaje directamente desde la notificación? | Elija una opción | |
| Preguntas sobre la aplicación y la estructura | | |
| El diseño de la aplicación transmite su utilidad | Elija una opción | |
| La estructura de pestañas utilizada en la aplicación es intuitiva | Elija una opción | |
| Los iconos utilizados son intuitivos y facilitan el uso | Elija una opción | |
| Los colores son adecuados y facilitan el uso | Elija una opción | |
| Preguntas SUS (Sistema Usability Scale) | | |
| Me gustaría utilizar esta aplicación frecuentemente | Elija una opción | |
| La aplicación es innecesariamente compleja | Elija una opción | |
| La aplicación es fácil de usar | Elija una opción | |
| Necesitaría la ayuda de un técnico para poder utilizar la aplicación | Elija una opción | |
| Encuentro que las funciones de la aplicación están bien integradas | Elija una opción | |
| Hay mucha inconsistencia en esta aplicación | Elija una opción | |
| A cualquier persona le resultaría rápido aprender a usar la aplicación | Elija una opción | |
| La aplicación es muy engorrosa de usar | Elija una opción | |
| Me siento seguro a la hora de facilitar mis datos | Elija una opción | |
| Necesité aprender muchas cosas antes de usar la aplicación | Elija una opción | |
| Otras funcionalidades que añadirías | | |
| Describe las funcionalidades que desearías añadir a esta aplicación para que se tengan en cuenta en versiones futuras | | |



Figura 24: Código QR para descargar la aplicación

Apéndice B. Liberación del primer prototipo

Una vez acabada la fase de desarrollo, se ha construido una versión de la aplicación en Android y se ha generado un en-

lace (<https://tinyurl.com/y98ps53z>) y un código QR (figura 24) que se ha enviado a los usuarios para que puedan realizar la descarga. No se han dado instrucciones previas para realizar la prueba para comprobar que los usuarios hayan podido entender el objetivo de la aplicación y se ha optado por hacer pruebas libres para que la descubran en la primera fase.

Una vez conectados a la aplicación y familiarizados con su funcionamiento, en la segunda fase, se ha pasado la encuesta del Apéndice A tanto a los usuarios como a los expertos de INeDITHOS para que puedan hacer una valoración de las funcionalidades más destacadas, y se ha separado a varios usuarios en chatrooms. Esta separación es transparente ya que el usuario no sabe de la existencia del resto y por eso no se ha incluido en la encuesta de evaluación.