GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

PLATAFORMA MÓVIL PARA EL APRENDIZAJE DE IDIOMAS BASADA EN TÉCNICAS DE MACHINE LEARNING

Autor: Alberto Colino Ruipérez

Director: David Contreras Bárcena

Madrid

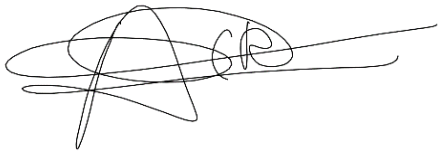
Junio de 2020

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

PLATAFORMA MÓVIL PARA EL APRENDIZAJE DE IDIOMAS BASADO EN TÉCNICAS DE MACHINE LEARNING

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2019/2020 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Alberto Colino Ruipérez Fecha: 11/06/ 2020



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: David Contreras Bárcena Fecha: 12/ 06 / 2020

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

PLATAFORMA MÓVIL PARA EL APRENDIZAJE DE IDIOMAS BASADA EN TÉCNICAS DE MACHINE LEARNING

Autor: Alberto Colino Ruipérez

Director: David Contreras Bárcena

Madrid

Junio de 2020

Agradecimientos

Quiero agradecerle este trabajo especialmente a mi director David, el cual ha hecho un esfuerzo extraordinario para guiarme a lo largo de este proceso y a Martin, director del centro de estudios extranjeros, que fue la persona que nos encargó el proyecto.

A mis compañeros, por vuestra gran ayuda en esos momentos duros y por haber podido trabajar día tras día con vosotros para sacar esta carrera adelante.

Así mismo, quiero mostrar un sincero agradecimiento a mi familia, sin los cuales no podría estar aquí.

Finalmente agradecer a toda la gente que ha estado ahí durante estos 4 años de la carrera. A todos, gracias.

PLATAFORMA MÓVIL PARA EL APRENDIZAJE DE IDIOMAS BASADA EN TÉCNICAS DE MACHINE LEARNING

Autor: Colino Ruipérez, Alberto
Director: Contreras Bárcena, David
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Este proyecto pretende desarrollar una aplicación móvil que implemente una red social para el aprendizaje de idiomas tanto para alumnos incoming y como para alumnos de la propia Universidad. El objetivo es que los usuarios tengan todas las facilidades para encontrar a otros alumnos con los que tener un intercambio de idiomas, pudiendo ver los usuarios que más se adaptan a sus características(analizando entre otros factores sus gustos con técnicas de machine learning) o pudiéndose comunicar con un chat.

Palabras clave: Apache Cordova, Aplicaciones híbridas, Android, IOS, Machine Learning, NLP.

1. Introducción

Vivimos en una sociedad en la que debido a la globalización es fundamental saber muchos idiomas, a pesar de que a veces es difícil aprenderlos porque no se puede viajar al extranjero para practicarlo. Esto unido a la gran cantidad de alumnos extranjeros que vienen a estudiar todos los años a nuestra universidad, ha generado la idea de desarrollar una aplicación móvil, para poder ver qué alumnos tantos extranjeros como nacionales están disponibles y poder tener un encuentro con ellos para poder realizar un intercambio de idiomas y mejorar nuestros idiomas.

2. Definición del proyecto

Este proyecto se ha desarrollado para que todas los alumnos y profesores de la Universidad puedan usarla, y les permita mejorar en el aprendizaje de idiomas. El objetivo principal es desarrollar una aplicación donde todos los estudiantes puedan ver todas las personas registradas, información de su perfil como su idioma nativo, los idiomas que quieren aprender, sus gustos y aficiones o las horas a las que están disponibles, con el objetivo de poder concertar un encuentro para tener dicho intercambio de idiomas. Otros objetivos son:

- Implementar un chat para poder hablar con otros alumnos, pudiendo traducir los mensajes que se mandan al idioma nativo de la persona con la que chatear, y también poder mandar fotos.
- Ver en directo qué usuarios están disponibles.
- Poder establecer recordatorios locales o eventos en el calendario para poder acordarse de cuándo se establece el encuentro.
- Ofrecer por parte del sistema al usuario sugerencias de contactos
- Usar técnicas de NLP y machine learning para clasificar a los usuarios en función de sus gustos.

- Poder enviar peticiones a otros usuarios mediante una notificación para tener un intercambio de idiomas.

3. Descripción del sistema

Se ha utilizado la plataforma apache Cordova para empaquetar una aplicación puramente web (usando HTML, JavaScript, css y php) para generar una aplicación Android e IOS. Este tipo de apps se llamas aplicaciones híbridas, y tienen una gran variedad de ventajas frente a las nativas, entre las que destaca su sencillez a la hora de desarrollarlas.[1]

El funcionamiento básico de la app consiste en una serie de archivos HTML que contendrán toda la interfaz visual de la app. A través de funciones Ajax se realizarán peticiones asíncronas a través de HTTPS al servidor. Dichas peticiones serán procesadas por los archivos php, los cuales permiten acceder a una base de datos MySQL, la cual guardará toda la información de los usuarios y demás información relevante para la app. También el servidor gestionará el acceso a la distintas APIS[2] usadas en al app. Para poder acceder a las funcionalidades del móvil (como la cámara, las notificaciones, el calendario, etc.) provistas por los plugins de Cordova se usarán funciones de JavaScript. A continuación, se muestra un esquema de los distintos archivos que componen la app y la funcionalidad que ejecuta cada uno.

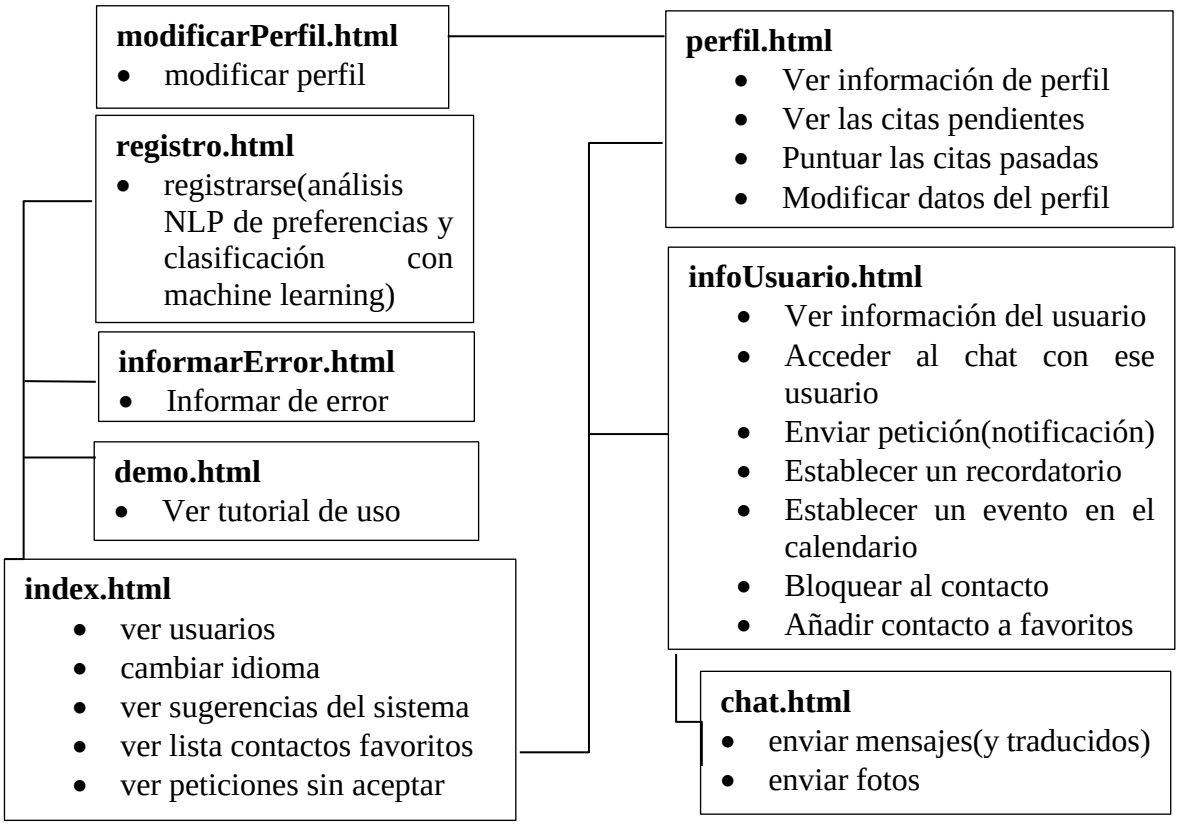


Ilustración 1 Componentes de la aplicación

4. Resultados

El resultado final de este trabajo es la creación de una app móvil capaz de mostrar en directo todos los usuarios que están registrados en ese momento, e información relevante sobre ellos (como su nivel de idioma, sus gustos o sus horas disponibles) para poder elegir con quién tener un intercambio de idiomas. También funciona perfectamente el mandar notificaciones a otros usuarios, tanto cuando se les quiere mandar una petición para poder tener un encuentro o cuando se les manda un mensaje de texto. La app también es capaz de gestionar todas las peticiones que tenemos, para aceptarlas, modificarlas o puntuarlas una vez que se ha realizado el encuentro. Las técnicas de machine learning y NLP permite clasificar de manera satisfactoria la descripción de los gustos de los usuarios para que el sistema ofrezca sugerencias en función de las preferencias o gustos de los otros usuarios Por otro lado, tanto los recordatorios locales como los eventos en el calendario que te permite crear la app harán que recuerdes cuándo habías quedado con alguien para tener un intercambio de idiomas.

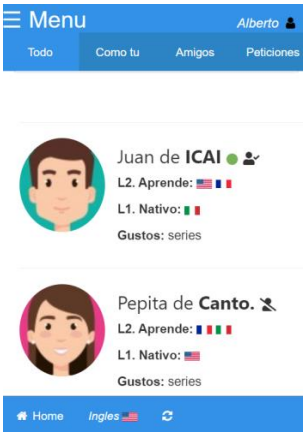


Ilustración 2 Apariencia principal de la aplicación

Ilustración 4 Vista del chat

Ilustración 3 Cómo mandar una petición(notificación)

5. Conclusiones

La aplicación realizada cumple todos los requisitos funcionales planteados al principio del desarrollo. En un futuro, esta aplicación se podría incluir en los servidores de Comillas, para proveerla de una mayor robustez y despliegue. También se podría introducir dentro de la aplicación oficial de la Universidad para que pudiese estar mucho más accesible, y evitar tenerse que volver a registrar para poder usarla.

6. Referencias

[1] A. Cordova, «Apache Cordova Wikipedia,» 25 octubre 2019. [En línea]. Available: https://es.wikipedia.org/wiki/Apache_Cordova. [Último acceso: 18 enero 2020].

[2] RedHat, «Qué son las API y para qué sirven,» [En línea]. Available: https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces. [Último acceso: 06 Junio 2020].

MOBILE PLATFORM FOR THE LEARNING OF LANGUAGES BASED IN TECHNIQUES OF MACHINE LEARNIG

Author: Colino Ruipérez, Alberto
Supervisor: Contreras Bárcena, David
Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

This project aims to develop a mobile application that implement a social network for language learning for both income students and students of the university itself. The main objective is that users have all the facilities to find other students with which to have a language exchange, being able to see the users that best adapt to their characteristics (analyzing among other factors their tastes with machine learning techniques) or being able to communicate with a chat.

Keywords: Apache Cordova, hybrid applications, Android, IOS, Machine Learning, NLP.

1. Introducción

We live in a society where, due to globalization, it is essential to know many languages, even though sometimes it is difficult to learn them because you cannot travel abroad to practice them. This, along with the large number of foreign students who come to study at our university every year, has led to the idea of developing a mobile application, to see that both foreign and national students are available and to be able to meet them in order to carry out a language exchange and improve our languages.

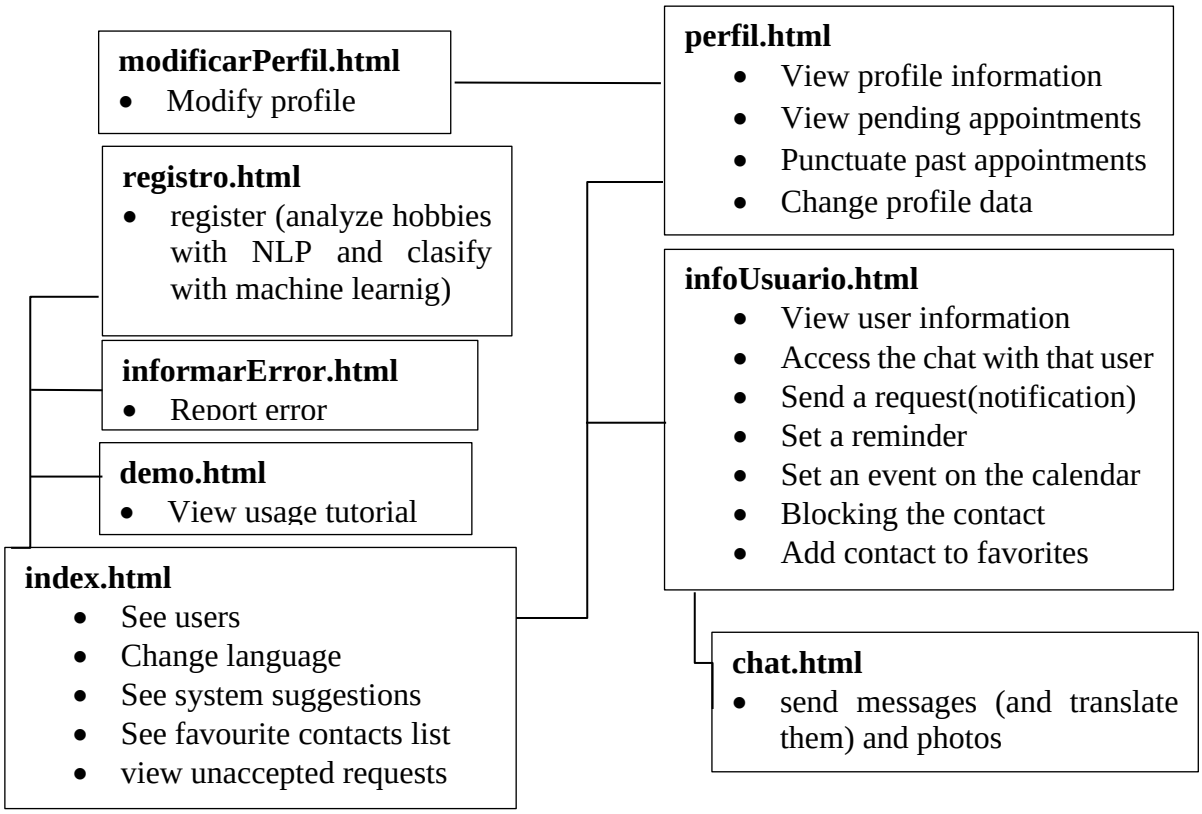
2. Project definition

This project has been developed so that all students and teachers at the University can use it, and it allows them to improve their language learning. The main objective is to develop an application where all the students can see all the registered people, information of their profile such as their native language, the languages they want to learn, their tastes and hobbies or the hours they are available, with the aim of being able to arrange a meeting to have that language exchange. Other objectives are:

- Implement a chat to be able to talk to other students, being able to translate the messages that are sent to the native language of the person with whom you are chatting, and also to be able to send photos.
- See which users are available
- Be able to set local reminders or events on the calendar so that you can remember when the meeting is set.
- To offer the user contact suggestions through the system
- To be able to send requests to other users through a notification to have a language exchange.

3. Description of the system

The Apache Cordova platform has been used to package a purely web application (using HTML, JavaScript, css and php) to generate an Android and IOS application.
The basic operation of the app consists of a series of HTML files that will contain all the visual interface of the app. Through Ajax functions asynchronous requests will be made through HTTPS to the server. These requests will be processed by php files, which allow access to a MySQL database, which will store all user information and other relevant information for the app. Also, the server will manage the access to the different APIS used in the app. In order to access the mobile features (such as camera, notifications, calendar, etc.) provided by the Cordova plugins, JavaScript functions will be used. Below is an outline of the different files that make up the app and the functionality that each one executes.



4. Results

The final result of this work is the creation of a mobile app capable of showing live all the users who are registered at that moment, and relevant information about them (such as their language level or their available hours) so that they can choose with whom to have a language exchange. It also works perfectly to send notifications to other users, either when you want to send them a request to have a meeting or when you send them a text message. The app is also able to manage all the requests we have, to accept them, modify them or rate them once the meeting has taken place.

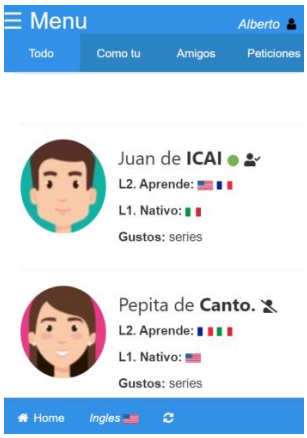


Ilustración 6 Home Page of the app

Ilustración 7 Chat of the app

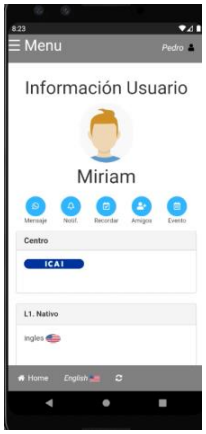


Ilustración 5 How to send a request(Notification)

5. Conclusion

This application meets all the functional requirements set at the beginning of the development. In the future, this application could be included in the servers of Comillas, to provide it with greater robustness and deployment. It could also be introduced within the official application of the University so that it could be much more accessible and avoid having to register again.

6. References

[1] A. Cordova, «Apache Cordova Wikipedia,» 25 Octuble 2019. [En línea]. Available: https://es.wikipedia.org/wiki/Apache_Cordova. [Último acceso: 18 Enero 2020].

[2] RedHat, «Qué son las API y para qué sirven,» [En línea]. Available: https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces. [Último acceso: 06 Junio 2020].

Índice de la memoria

Índice de ilustraciones

Índice de tablas

Capítulo 1. INTRODUCCIÓN

Este proyecto va a consistir en desarrollar una aplicación móvil que implementará una red social para favorecer los intercambios bilingües entre alumnos de Comillas de sus diferentes centros. La aplicación se desarrollará con una plataforma llamada Apache Cordova, la cual permite empaquetar una aplicación puramente web (HTML, CSS y JavaScript) y generar una aplicación para Android o para IOS.

Este tipo de aplicaciones se llaman aplicaciones hibridas, que, a diferencia de las aplicaciones nativas no necesitan un lenguaje distinto para cada plataforma en la que se implemente la aplicación (kotlin en el caso de Android y Swift en el caso de IOS).

La aplicación estará tanto en inglés como en español para que sea más fácil para los alumnos de intercambio poder usar la aplicación. Todos los alumnos podrán ver en tiempo real todos los alumnos que están dados de alta en la aplicación y saber si están usando o no en ese momento la aplicación o si están disponibles para poder tener en ese momento un intercambio de idiomas.

Se ofrecerá también la posibilidad de chatear con los usuarios de la aplicación a través de un chat de mensajes, y poder mandar notificaciones a otros contactos ofreciéndoles tener un intercambio de idiomas. El sistema también ofrecerá al usuario encuentros en función de sus características, la carrera y el centro donde estudia, sus preferencias y gustos y por supuesto por su idioma. Todo con el objetivo de ayudar al usuario a encontrar la persona que más se adecúe a sus características para poder tener un rato de conversación en el idioma que quiere aprender.

1.1 MOTIVACIÓN DEL PROYECTO

Este proyecto se propuso desde el departamento de Idiomas de Comillas como una idea para facilitar los intercambios bilingües entre alumnos extranjeros que vienen a estudiar a Comillas y los que están ya aquí. Haciendo uso del tipo de aplicación más usada actualmente como son las redes sociales (como WhatsApp, Facebook) se pretende facilitar los intercambios de idiomas en la Universidad.

Hoy en día el móvil se ha convertido en el elemento que más usamos en nuestro día, y en concreto las aplicaciones que tiene instaladas. Se estima que de media las personas usamos unas 4 horas el móvil al día, y en concreto todas al aplicaciones de redes sociales, como WhatsApp, Instagram o Twitter. Nos gusta estar continuamente conectados y enterados de lo que pasa a nuestro alrededor o lo que están haciendo nuestros amigos y familiares en cada momento. Por ello, el desarrollo de esta app, que recoge todas las características principales de las aplicaciones más usadas hoy en día.

Por un lado, se pretende favorecer la integración de los alumnos incoming, permitiéndoles dar una plataforma para conocer a gente nueva de la universidad con el objetivo de poder quedar con ellos para poder practicar el español u otro idioma que quieran aprender, y enseñar su idioma nativo a la otra persona. Por otro lado, dar la oportunidad a los estudiantes de origen de la Universidad de poder practicar los idiomas que quieran aprender y poder estar más en contacto con la gente que viene de fuera a estudiar.

Por tanto, debido a la gran cantidad de alumnos incoming se espera que la app sea ampliamente usada por ellos, y que se pueda también extender su uso al resto de alumnos de la universidad.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

2.1 PLATAFORMA DE DESARROLLO (APACHE CORDOVA)

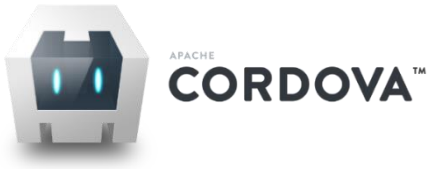


Ilustración 8 Logo de Apache Cordova

Se va a usar la plataforma Apache Cordova que permite encapsular aplicaciones web formadas por HTML, JavaScript, CSS en una aplicación Android o IOS.

"Las aplicaciones resultantes son híbridas, lo que significa que no son ni una aplicación móvil nativa (porque toda la representación gráfica se realiza vía vistas de Web en vez del framework nativo) ni puramente basadas en web (porque no son solo aplicaciones web, sino que están empaquetadas como aplicaciones para su distribución y tienen acceso a las APIs nativas del dispositivo)". [1]

Entre las grandes ventajas de las aplicaciones hibridas encontramos que son capaces de ser visualizadas en cualquier tipo de teléfono móvil, independientemente de su sistema operativo, y que suponen un menor coste en desarrollo y mantenimiento, ya que el código que tienes que escribir para cada una de las plataformas en las que está desplegada la app es el mismo.

Entre sus desventajas están que tienen funciones limitadas, debido a que no tienen acceso a todos los recursos del móvil, ya que son necesarios los plugins para poder acceder a dichas funciones especiales. [2]

Se ha decidido usar esta plataforma y no otras (como Ionic) por varias razones. Por un lado, porque existe gran cantidad de plugin ya creados para Cordova, y además te permite generar una aplicación Android o IOS sin tener que aprender otros idiomas de programación, cosa que habríamos tenido que hacer si lo hubiésemos desarrollado de forma nativa o con Ionic(esta está basada en HTML pero las etiquetas cambian).

Por último, decir que es una plataforma desarrollada por Apache Software Foundation gratuita, de código abierto y de libre distribución, disponible para Windows y Mac.

2.2 FRAMEWORK Y SIMULADORES (ANDROID STUDIO Y XCODE)

Estos framework de desarrollo de aplicaciones móviles tanto para Android como para IOS nos permitirán hacer uso de sus simuladores de móvil para poder probar nuestras aplicaciones en un dispositivo nativo tanto de Android como de IOS. Son gratuitas, aunque Xcode a diferencia de Android Studio, solo está disponible para IOS.

2.3 *SERVIDOR WEB*

Ilustración 9 Logo de WampServer

Como servidor web de desarrollo usaremos **WampServer**, que tiene incluido Apache, MySQL para implementar la base de datos y PHP como lenguaje de backend. Aunque

más adelante pretendemos que la aplicación resida dentro de **los servidores de la universidad,** para estos primeros meses del proyecto usaremos este servidor que tendré instalado en mi portátil.

2.4 LENGUAJES DE PROGRAMACIÓN WEB

Ilustración 10 Logos lenguajes programación web

Haremos uso de los principales lenguajes de programación web como pueden ser **HTML5**, **CSS** para las hojas de estilo y JavaScript, con sus múltiples versiones más modernas como pueden ser **JQuery** con **AJAX**.

Como lenguaje de backend usaremos **PHP**, ampliamente utilizado en el ámbito de la Web. La versión usada será la 7.3.5, siendo esta la última en el momento del comienzo del proyecto. La razón de usar este lenguaje y no Java es que en un futuro se pretende que la aplicación esté integrada dentro de los servidores de Comillas y estos el lenguaje con el que mejor encuentra compatibilidad es con PHP.

Para el acceso a la base de datos **MySQL** usaremos **SQL**, haciendo la conexión a través de PHP. Como interfaz usaremos phpMyAdmin, que nos provee de un interfaz sencillo y completo para poder gestionar las bases de datos que vayamos creando.

Paralelamente se hará uso de **Bootstrap**, que es una biblioteca multiplataforma o framework que contiene plantillas de diseño con infinidad de recursos gráficos y extensiones de JavaScript.

2.5 PLUGINS

2.5.1 INTRODUCCIÓN

Ilustración 11 Logo de Plugins de Apache Cordova

Los plugins son una funcionalidad fundamental de Apache Cordova, los cuales nos proveen de una interfaz para poder acceder a características nativas de los SO específicos (como Android e IOS) desde nuestro código JavaScript.

Los plugins están compuestos por 3 partes fundamentales: Un código JavaScript, un módulo de Cordova para traducirlo a código y varias partes nativas, una específica para cada una de las plataformas en las que vamos a desarrollar la aplicación: [3]

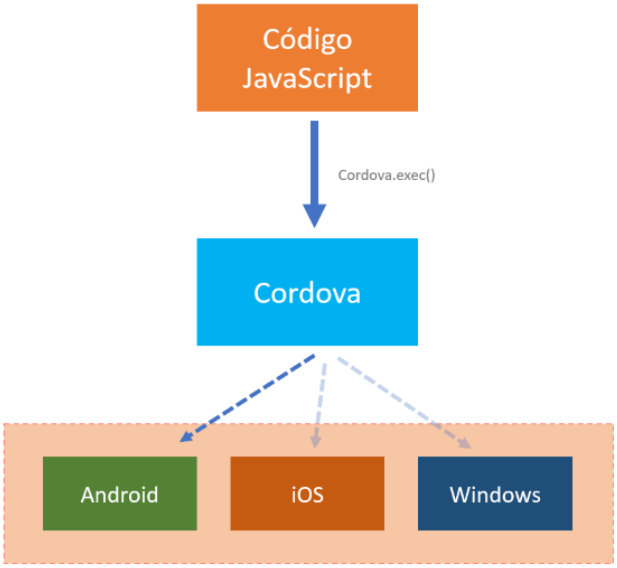


Ilustración 12 Componentes de un plugin

Otro apunte importante es que, debido a la peculiaridad de esta herramienta, toda la aplicación se desarrolla como una aplicación web y luego se usa la herramienta Cordova para obtener la aplicación móvil. Pero los plugin solo funcionan en la versión móvil, ya que estos acceden a una funcionalidad específica del sistema operativo o del hardware del móvil (la cámara, las fotos, etc.). Por tanto, si la aplicación se ve en el navegador, todas estas funcionalidades no podrán ejecutarse.

2.6 APIS

Una API nos provee de una interfaz para poder integrar software y accede a ciertos recursos web ofrecidos por otras páginas.

Todas las APIs usadas en esta aplicación son de Google, el cual tienen a la disposición del público una enorme variedad de APIs para poder acceder a diversos recursos, como pueden ser las notificaciones Push, procesamiento de lenguaje natural, traducción de textos con el traductor de Google.

Para poder acceder a la API debemos seguir 3 pasos:

1. Darnos de alta en el uso de la API desde la consola de Google.
2. Obtener unas claves para poder acceder a dicha API.
3. Implementar en nuestro código php la llamada a la API.

2.6.1 CURL

Ilustración 13 Logo de Curl

Para poder hacer el 3 paso, vamos a hacer uso de cURL, el cual es : “Curl es un proyecto de software consistente en una biblioteca (libcurl) y un intérprete de comandos (curl) orientado

a la transferencia de archivos" [4] Este soporta gran cantidad de protocolos, aunque a nosotros solo nos va a servir el HTTPS. El principal objetivo de cURL es automatizar transferencias de archivos o secuencias de operaciones no supervisadas, es decir, acciones que no a hacer el usuario sino el sistema.

Por ejemplo, cuando el usuario desee traducir un mensaje a otro idioma, cURL se va a encargar de hacer la llamada a la API de Google, establecer la conexión, enviarle el texto y recibir la traducción. Para ello se configura una url que tiene la dirección del servidor al que se le va a hacer la consulta, junto con una serie de parámetros, como por ejemplo el texto que se desea traducir. Curl recibirá la respuesta y así nosotros podremos mostrarla.

2.6.2 COMPOSER

Ilustración 14 Logo de Composer

Composer consiste en un sistema gestor de paquetes y dependencias para PHP. Descarga e instala de forma automática las librerías necesarias en un proyecto determinado de este lenguaje [5]. Es tan sencillo como elegir el paquete que queremos instalar, coger el comando que debemos usar y pegarlo y ejecutarlo en un cmd localizado en la carpeta donde queremos descargarlo. Todos los archivos se nos descargarán en una carpeta que se creará él llamada "vendor". Para poder usar los archivos del paquete desde otro archivo php simplemente hay que introducir al principio el comando **include()** y el comando **use** para poder usar toda la información de los archivos que queremos exportar.

La página principal donde están la mayoría de los paquetes de php es https://packagist.org/, en la cual elegimos el paquete que queremos de descargar y obtenemos el comando que debemos usar. Un ejemplo de comando sería :

```
composer require nlp-tools/nlp-tools
```

2.6.3 GOOGLE CLOUD PLATFORM

En esta página de Google podemos darnos de alta en las distintas APIS que ofrece al público Google, junto a una gran variedad de información como el uso que le damos a cada API, el coste que nos han cargado en nuestra cuenta por su uso o la lista de todas las API en las que nos podemos dar de alta(en el apartado de Habilitar APIS y servicios). Para poder usar dicha plataforma debemos estar registrados con una cuenta de Google, y a continuación crear un proyecto(en este caso notifa7a) en el que vamos a activar las APIS. En la siguiente imagen podemos ver cómo estamos dados de alta en la API de "Cloud Natural Language" y en la de "Cloud Translation ", y que las solicitudes que hemos realizado a ambas han sido bastante bajas en los últimos meses, ya que simplemente las hemos usado para comprobar que funcionaban bien.

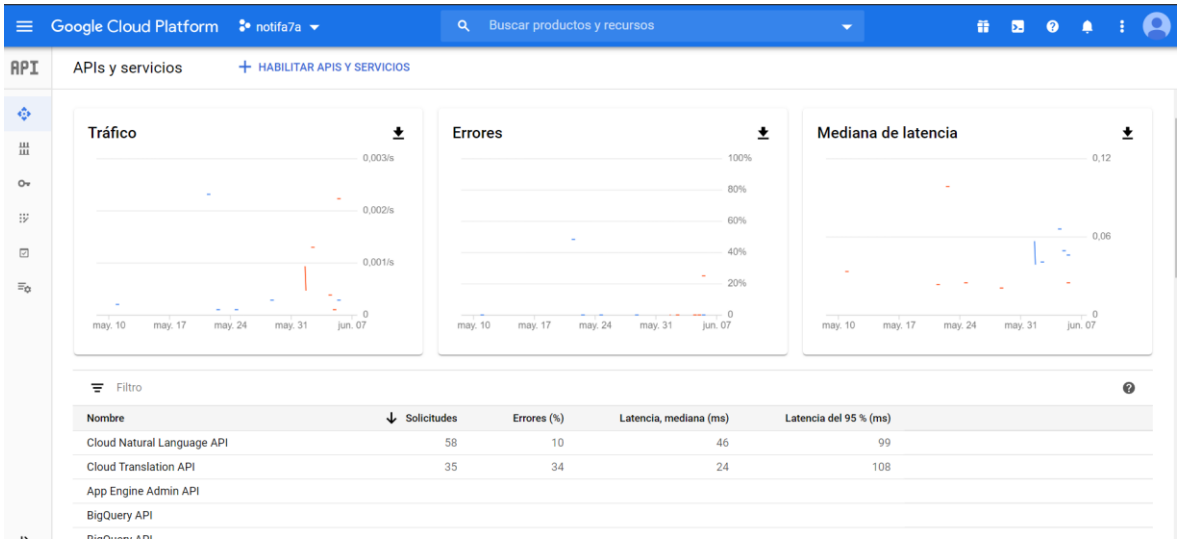


Ilustración 15 Google Cloud Platform

Capítulo 3. ESTADO DE LA CUESTIÓN

3.1 TANDEM

Disponible para iOS y Android, esta aplicación tiene actualmente más de 5 millones de personas que la usan, y está presente en unos 100 países del mundo, pudiendo elegir entre una gran cantidad de idiomas (en torno a unos 150), está centrado en una forma de aprender basada en el intercambio de idiomas en el que influyen no solo los aspectos meramente lingüísticos, sino también el conocimiento del entorno social y cultural de las personas que llevan a cabo la conversación. Esto en la práctica se traduce en que al principio la aplicación te pide que escribas tus gustos y aficiones y una frase que te describa, datos que luego influirán a la hora de ofrecerte sugerencias la aplicación.

Con respecto a la forma de funcionar de la aplicación esta consta de la posibilidad de iniciar un chat con una persona para intercambiar mensajes en otro idioma. Para poder empezar a mandar fotos o hacer llamadas o videollamadas se necesita que la persona haya enviado 1 mensaje al usuario con el que quiere comunicarse. También se puede seguir a los usuarios, de forma parecida a como se hace en Instagram o en Twitter. Cómo peculiaridad esta aplicación también ofrece una serie de juegos para practicar los idiomas y así nos resulte más ameno aprender, aunque dicha funcionalidad solo está disponible en IOS. También tiene una versión de pago.

Además, se ofrece la opción de “Aprender con profesionales” en la que puedes entrar en contacto con profesores expertos para que te den clase. El precio de las clases está en torno a los 15 euros por hora y la duración de las clases varía entre los 20 y 60 minutos

Una diferencia importante con HelloTalk es que una vez que te has registrado te pone un mensaje de que al haber lista de espera en varios países tu solicitud puede tardar hasta 7 días en procesarse, no pudiendo usar la aplicación de forma instantánea después de haberte dado de alta en ella. [6] [7]

3.2 HELLOTALK

HelloTalk es una aplicación que está disponible en IOS y Android y que está presente en 150 países, pudiendo elegir más de 100 idiomas para practicar.

La aplicación cuenta con dos ejes principales. Por un lado, tiene un chat para que puedes conectarte con hablantes nativos y puedas chatear con ellos de forma totalmente gratuita. Cualquiera de los mensajes que se envían te da la opción de poder traducirlos a tu lenguaje nativo, comprobar que están bien escritos en el idioma que estas chateando o ver como se pronuncia correctamente esa palabra. No solo puedes chatear con una persona de forma individual, sino que puedes unirte a grupo como lo harías en otras aplicaciones como WhatsApp para poder hablar con varias personas a la vez. Aparte de mandar mensajes también se pueden mandar emoticonos, imágenes, videos, audios, tu ubicación.

Con el chat puedes concertar una cita para tener un intercambio lingüístico cara a cara según tus preferencias de ciudad, distancia, idioma …

Otra opción muy interesante que ofrece son los llamados “Momentos”, en los que tú puedes compartir cualquier tipo de publicación como por ejemplo una noticia para que la gente pueda verla, comentarla y darte feedbak, pero en el idioma que elijas.

La aplicación ofrece una versión VIP a la que te puedes apuntar pagando una cuota al mes. Tienes la opción de pagar 3.19 euros al mes, o 79.99 si lo quieres para toda la vida. Esta versión de pago ofrece un mejor maching a la hora de sugerirte contactos, poder aprender 3

idiomas a la vez, aumentar el envío de mensajes que se pueden mandar a extraños entre otras muchas opciones. [8] [6]

Ilustración 16 Ejemplo del chat de la app HelloTalk

3.3 SPEAKY

Speaky es una aplicación que implementa una red social para aprender idiomas. Está formada por más de 1 millón de usuarios de más 180 países en los que se hablan unos 110 idiomas diferentes. La aplicación está presente tanto para Android como para IOS, y también se puede usar la versión web.

La herramienta principal de esta aplicación es su chat de texto, que implementa un servicio de traducción para las palabras o frases que no se entiendas. La aplicación pone mucho énfasis en diferenciar a los usuarios nativos, y no nativos, ya que esto es la primera clasificación que nos encontramos de los usuarios. También a la hora de elegir persona para conversar se pueden aplicar filtros en función del idioma que queramos practicar, del género o de la edad de la persona. A diferencia de las otras aplicaciones vistas anteriormente, Speaky no tiene la opción de hacer llamadas o videollamadas. Por último, también está la opción de agregar amigos, y poder tenerlos en un apartado especial para poder acceder a ellos de forma más rápida. La aplicación tiene un apartado especial donde se marcan los logros que has conseguido, como por ejemplo haber conseguido traer varios amigos a la aplicación. [9] [6]

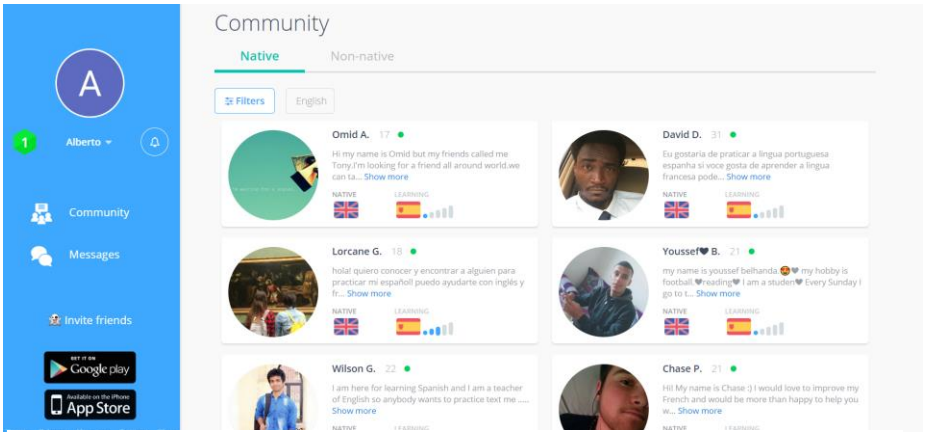


Ilustración 17 Vista de Speaky

3.4 HI UTANDEM

Esta aplicación está disponible para Android, y puedes elegir de entre más de 100 idiomas distintos y la app está disponible en todos los países del mundo. Esta se centra más en los intercambios de idiomas presenciales entre personas que viven en una misma ciudad, quedando primero a través de un chat. Se ofrece la opción de bares, en las que te sugieren los mejores bares para hacer tus intercambios, y también se ofertan academias donde puedas

aprender de forma más específica y con profesionales un idioma. Por último, también tienes la capacidad de añadir eventos a tu calendario personal para no perderte las citas que has acordado con alguien, ya sea en un bar o en otro sitio.

Se pueden añadir a tu perfil tus intereses y aficiones para poder quedar con personas que compartan tus mismos gustos. No es muy buena alternativa ya que tanto el login a la aplicación no funcionan como la página web de la aplicación no te permite acceder a ella. [6]

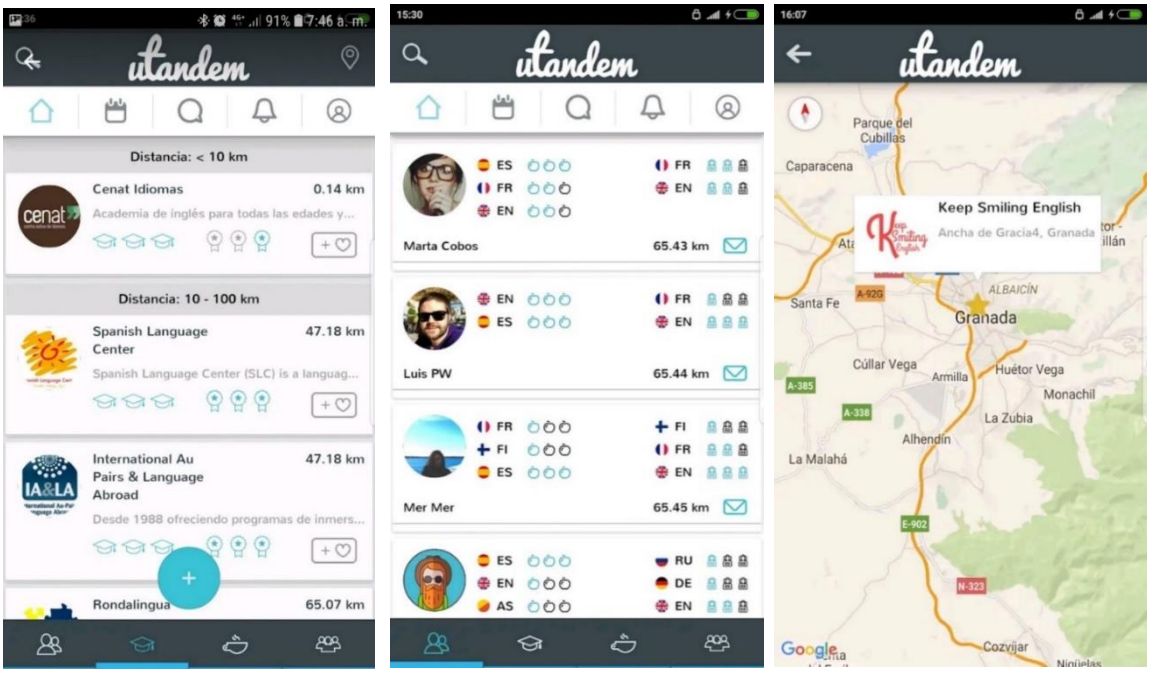


Ilustración 18 Apariencia de la app Hi uTandem

3.5 TABLA COMPARATIVA

	Android	IOS	Facebook o Web	Número descargas	Número idiomas
Speaky	Sí	Sí	Sí	+ 1 M	110
HelloTalk	Sí	Sí	No	+15 M	100
Tandem	Sí	Sí	No	+5 M	100
Hi uTandem	Sí	No	Sí	-	-

Tabla 1 Tabla comparativa aplicaciones

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

Debido a que actualmente no hay ningún servicio de la Universidad que ofrezca la posibilidad de realizar intercambios de idiomas entre los alumnos incoming y los propios alumnos de la Universidad, el desarrollo de esta aplicación supone una herramienta muy útil para conseguir ofrecer este servicio que hasta ahora no se podía ofrecer por parte de la Universidad.

Aunque este servicio no nos parezca a priori algo fundamental, si lo analizamos con más detalle, nos damos cuenta de que por un lado se ayuda mucho a los alumnos incoming a estudiar el español, hecho que les ayudará tremendamente en el desarrollo de su día a día en nuestro país, y también les ayudará a entender mejor las clases, y a relacionarse mejor con el resto de los alumnos de la Universidad. Pero debido que esta aplicación está disponible para todos los alumnos, no solo para los extranjeros, también permite favorecer el contacto entre alumnos de distintos centros de la Universidad, ya que estos también pueden ayudarnos a mejorar nuestros idiomas teniendo un encuentro para conversar en los respectivos idiomas que se quieren aprender.

Por otro lado, los idiomas son uno de los conocimientos que más se valora en un profesional hoy en día, ya que vivimos en un mundo completamente globalizado en la que la mayoría de los trabajos en ingeniería incluyen equipos de trabajo internacionales. Esto quiere decir que si no sabes desenvolverte bien en el inglés o en otros idiomas vas a tener un serio problema en el trabajo. Esta aplicación pretende proponer una herramienta muy útil para aprender idiomas, que es practicarlo cara a cara con un nativo en ese idioma.

4.2 OBJETIVOS

Como primer paso para plantear los objetivos del proyecto se celebró una reunión con Martin Beagle, director del departamento de Idiomas de Comillas, que es la persona que tuvo la idea del proyecto, para que nos contase cuáles eran sus principales ideas sobre el proyecto. Posteriormente, se desarrolló un estudio que consistió en desarrollar los casos de usos y un análisis de requisitos del proyecto. De todo ello salieron los siguientes objetivos:

1. **Consultar usuarios:** Gestionar en tiempo real la disponibilidad de los usuarios, pudiendo ver todos los usuarios dados de alta en la aplicación, y poder acceder a su información.
2. **Interactuar con otros usuarios:**
    a. **Consultar su perfil:** Ver en más detalle la información de ese usuario, sus preferencias o aficiones, las horas a las que está disponible para tener un intercambio de idiomas, los idiomas y el nivel que tienen en cada uno de ellos, etc.
    b. **Concertar una cita.**
    c. **Mandarles una notificación:** Se puede mandar una notificación en tiempo real y poder establecer notificaciones locales en el propio móvil para recordarnos si tenemos alguna cita establecida.
    d. **Iniciar un chat con ellos:** El sistema dispondrá de un chat para que los usuarios puedan chatear entre ellos.
3. **Perfil propio:** Poder editar en todo momento tu perfil, para modificar los idiomas que aprendes, tus preferencias, gestionar las citas pendientes…
4. **Sugerencias del sistema:** El sistema ofrecerá encuentros en función de las preferencias y características propias de cada usuario (el centro en el que esta, el horario disponible que tiene, los idiomas que quiere aprender, sus hobbies).
5. **Analisis de los gustos de los usuarios con machine learning:** Después de que los usuarios se registre el sistema analizara con técnicas de análisis de lenguaje natural el texto que haya introducido en el que explicara sus gustos, aficiones o hobbies. Después se usará machine learning para ir entrenando y testeando un modelo que permita predecir

la palabra de referencia de ese texto. Por ejemplo si el usuario introduce "Me encanta ir a correr todos los días", el modelo predecirá que la palabra es correr.

6. **Generar estadísticas uso:** Hacer un estudio estadístico posterior sobre el uso que se hace de la aplicación por parte de los distintos grupos de alumnos que la usan.

4.3 METODOLOGÍA

El proyecto va a seguir una metodología agile, siendo esta una de las más famosas y más ampliamente usadas actualmente en el desarrollo software.

La metodología agile está basada en el desarrollo iterativo e incremental en el que los requisitos y soluciones van evolucionando con el tiempo según las necesidades cambiantes del proyecto. Esta metodología consta de un conjunto de sprint(iteraciones), o pequeños objetivos en los que se divide el proyecto. Esto lo que favorece es la agilidad (como su propio nombre indica) mientras se desarrolla el proyecto. Después de cumplir el sprint este puede ser mejorado por posteriores iteraciones. Por tanto, esto último facilita que todo lo desarrollado sea revisado varias veces dotándole de mayor calidad.

Dentro del desarrollo del software vamos a tener una serie de sprint, que se corresponderán con los distintos objetivos que tenemos marcados. Tenemos el sprint Notificaciones, chat, mostrar usuarios en tiempo real, disponibilidad de los usuarios, sugerencias del sistema, desarrollar la interfaz visual principal, concertar cita, alojar la aplicación en los servidores de la Universidad …

4.3.1 CRONOGRAMA

En la siguiente ilustración se plasma el diagrama de GANTT, que es una gráfica que nos permite ver cuánto tiempo tenemos pensado dedicarle a cada una de las tareas que vamos a realizar en nuestro proyecto. Podemos observar cómo en la gráfica hay tareas que se superponen, debiéndose esto a que muchas veces a la vez que se desarrolla una tarea se revisa y mejora la anterior.

Hay que aclarar que el tiempo dedicado es una estimación y muchas veces se ha puesto más tiempo del que luego realmente se va a usar.

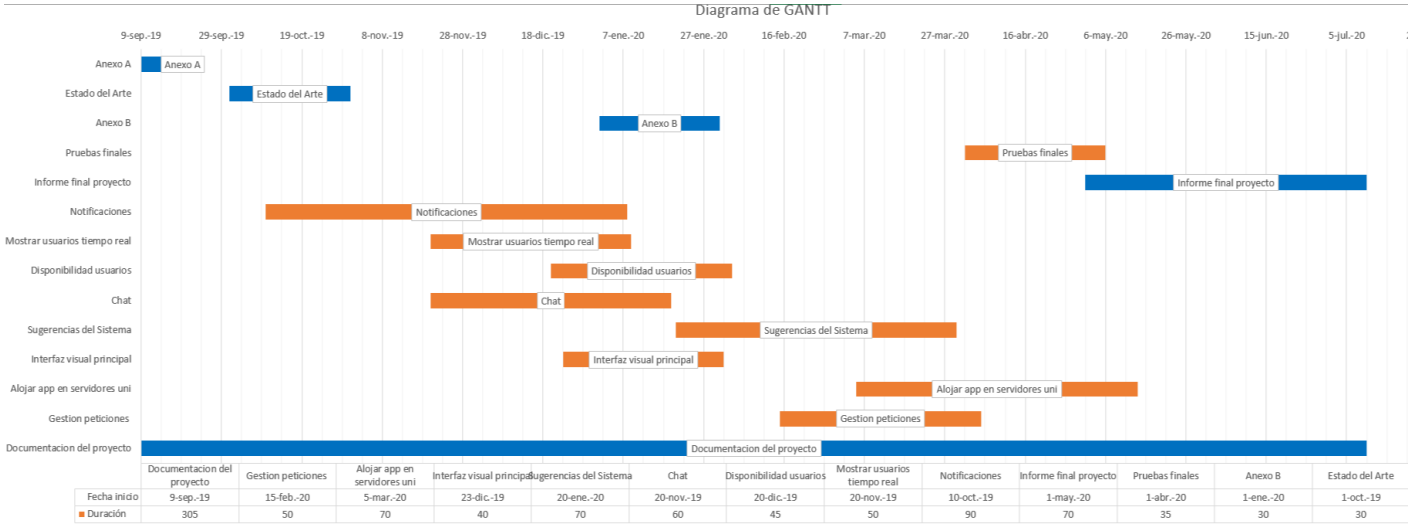


Ilustración 19 Diagrama de Gantt

4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

En este proyecto se ha usado una serie de herramientas de software libre y totalmente gratuitas, con lo cual se ha podido disminuir de manera significativa el coste del proyecto. Habría que tener en cuenta de que en el proyecto se han hecho uso de una serie de APIS de Google, las cuales al principio son gratuitas, y a medida que se va aumentando el uso de esta ya sí que empiezan a cobrar por el uso. Por tanto, durante el desarrollo del proyecto ha sido gratuito su uso, pero cuando se despliega la aplicación para su uso comercial sí que ya habría que tener en cuenta el dinero que cobrarían por ello.

4.4.1 PRECIO GOOGLE TRANSLATE

En la siguiente imagen de la tabla de los precios de Google Translate se pone lo que cobrarían por el uso de las traducciones de texto provistas por dicho servicio: [10]

Precios mensuales

API Translation: edición básica

Función	Puede optar al uso gratuito	Precio
Detección de idioma	✓	20 USD por millón de caracteres*
Traducción de texto (modelos generales de NMT)	✓	20 USD por millón de caracteres*
Traducción de texto (modelos generales de PBMT)	✓	20 USD por millón de caracteres*

API Translation: edición avanzada

Función	Puede optar al uso gratuito	Precio
Detección de idioma	✓	20 USD por millón de caracteres*
Traducción de texto (modelos generales de NMT)	✓	20 USD por millón de caracteres*
Traducción de texto (modelos generales de PBMT)	✓	20 USD por millón de caracteres*
Traducción de texto (modelos de AutoML)	✓ **	Los precios se basan en el volumen mensual (consulta la página de precios de AutoML)

* El precio se calcula por cada carácter que se envíe a la API para procesarse, incluidos los espacios en blanco. Las consultas vacías se cobran como un carácter. Google cobra por carácter, incluso si estos están compuestos de varios bytes; cada carácter corresponde a un punto de código. Por ejemplo, al traducir al inglés "こんにちは", se facturan 5 caracteres.

** A partir del 1 de noviembre del 2019, se aplicarán 40 USD de uso gratis al mes a las operaciones de predicción de la traducción de modelos de AutoML por cuenta facturada, lo que equivale a 500.000 caracteres al mes.

Ilustración 20 Precios Google Translate

4.4.2 PRECIOS FIREBASE MESSAGING

Con respecto al coste por el uso de Firebase Messaging, si consultamos su página podemos ver cómo es totalmente gratuito independientemente del uso que le demos: [11]

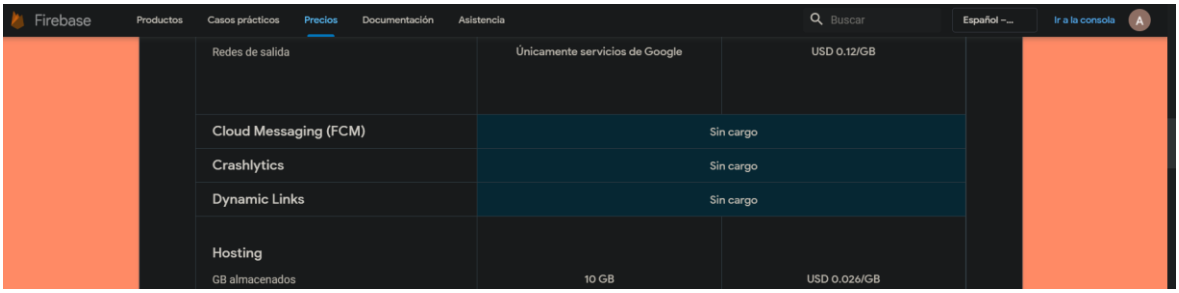


Ilustración 21 Precios Firebase Cloud Messaging

4.4.3 PRECIOS NLP GOOGLE

Respecto al uso de la API de cloud natural languague de Google, su precio queda estipulado por el uso que hagas de ella. Cuantas más consultas hagas, más se cobra al mes. Como el uso que hacemos es bastante pequeño, no nos suponen ningún coste [12]

Precios mensuales

Función	De 0 a 5,000	De 5,001 a 1,000,000	De 1,000,001 a 5,000,000	De 5,000,001 a 20,000,000
Análisis de entidades	Gratis	$1.00	$0.50	$0.25
Análisis de opiniones	Gratis	$1.00	$0.50	$0.25
Análisis sintáctico	Gratis	$0.50	$0.25	$0.125
Análisis de opiniones por entidad	Gratis	$2.00	$1.00	$0.50

Función	De 0 a 30,000	De 30,001 a 250,000	De 250,001 a 5,000,000	Más de 5,000,000
Clasificación de contenido	Gratis	$2.00	$0.50	$0.10

Ilustración 22 Precios API NLP Google

4.4.4 COSTE TOTAL DEL PROYECTO

El análisis del coste total del proyecto, es decir, el dinero que nos va a costar llevar a cabo el proyectos siempre es una de las partes más importantes y cruciales del trabajo. Debido a que la mayoría de las herramientas de software usadas son totalmente gratuitas(o en el caso de las APIs su reducido uso también hace que sean gratuitas) los únicos costes del proyecto han sido el ordenador usado para acometer el proyecto y el coste por hora trabajada.

Con respecto al coste por hora trabajada, hay que tener en cuenta que este proyecto se ha desarrollado de septiembre a junio, es decir 10 meses con 40 semanas. Si consideramos que de media se ha trabajo unas 12’5 horas semanales, se ha trabajado en total 500 horas en el desarrollo completo. Asignando un coste de 9 € por hora trabajada, nos da un coste final de 4.500 €

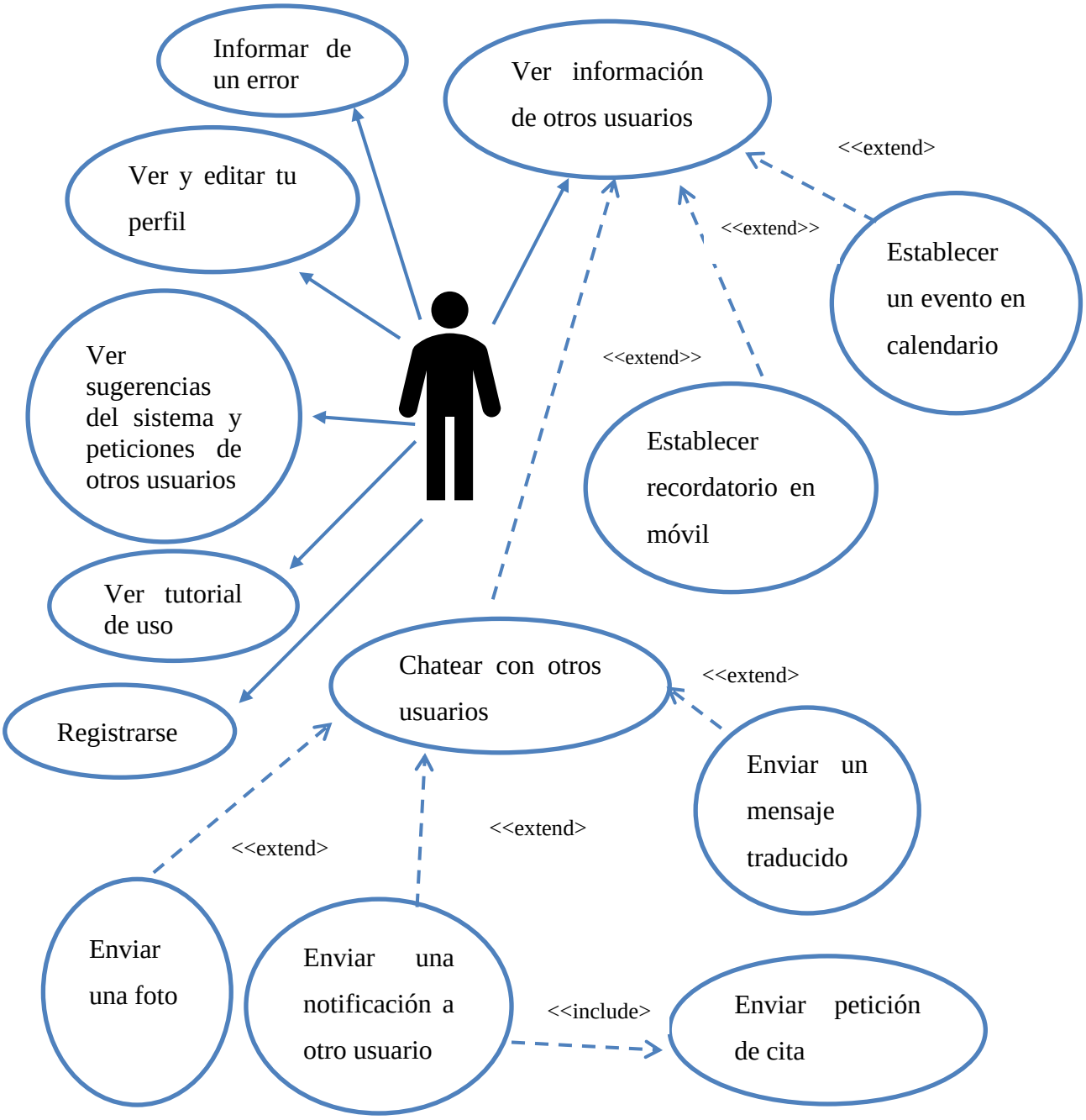
Concepto	Explicación	Coste monetario
Coste desarrollador	Coste total para pagar al desarrollador del software	4.500 €
Total		4.500 €

Tabla 2 Coste total del proyecto

Capítulo 5. SISTEMA

5.1 ANALISIS

5.3.1.1 CASOS DE USOS DE LA APLICACIÓN



5.3.1.2 HISTORIAS DE USUARIO

Historia: Registrarse.

Como : Usuario de la app.

Quiero : Registrarme, introduciendo los datos que me va pidiendo el formulario de registro.

Para : Poder usar todas las funcionalidades de la app, como enviar peticiones, recibir sugerencias del sistema, etc.

Historia: Obtener estadísticas de uso.

Como : Administrador de la app.

Quiero : Poder obtener un Excel con toda la información de uso de la app.

Para : Poder analizar posteriormente que tipo de perfiles usan la app, quién la usa más, cuál es la funcionalidad más usada, etc.

Historia: Iniciar un chat.

Como : Usuario.

Quiero : Poder comenzar un chat de texto con otro usuario.

Para : Poder mandarle mensajes de texto, tanto en mi idioma como en el suyo(haciendo uso del traductor incorporado).

Historia: Concertar un encuentro.

Como : Usuario.

Quiero : Poder concertar un encuentro con otro usuario a una hora y un día determinado.

Para : Poder quedar con él para tener un encuentro de idioma y que la app me pueda avisar de cuando he quedado.

Historia: Consultar las personas, entre ellas las disponibles

Como : Usuario.

Quiero : poder ver qué personas están disponibles justo en ese momento para tener un encuentro.

Para : poder mandarles una notificación o un mensaje para ofrecerles tener un intercambio de idiomas.

Historia: Consultar mi perfil.

Como : Usuario.

Quiero : poder ver mi información de perfil y poder editar mi perfil.

Para : asegurarme de que la información que se muestra sobre mí en el app está actualizada y pueda ver fácilmente las citas que tengo concertadas para tener un intercambio de idiomas.

Historia: Consultar información de otros usuarios.

Como : Usuario.

Quiero : Poder tener una idea más destallada del perfil del otro usuario(sus gustos, idiomas que aprende, horas disponibles, etc.)

Para : Poder elegir a la personas que más se adapten a mis características.

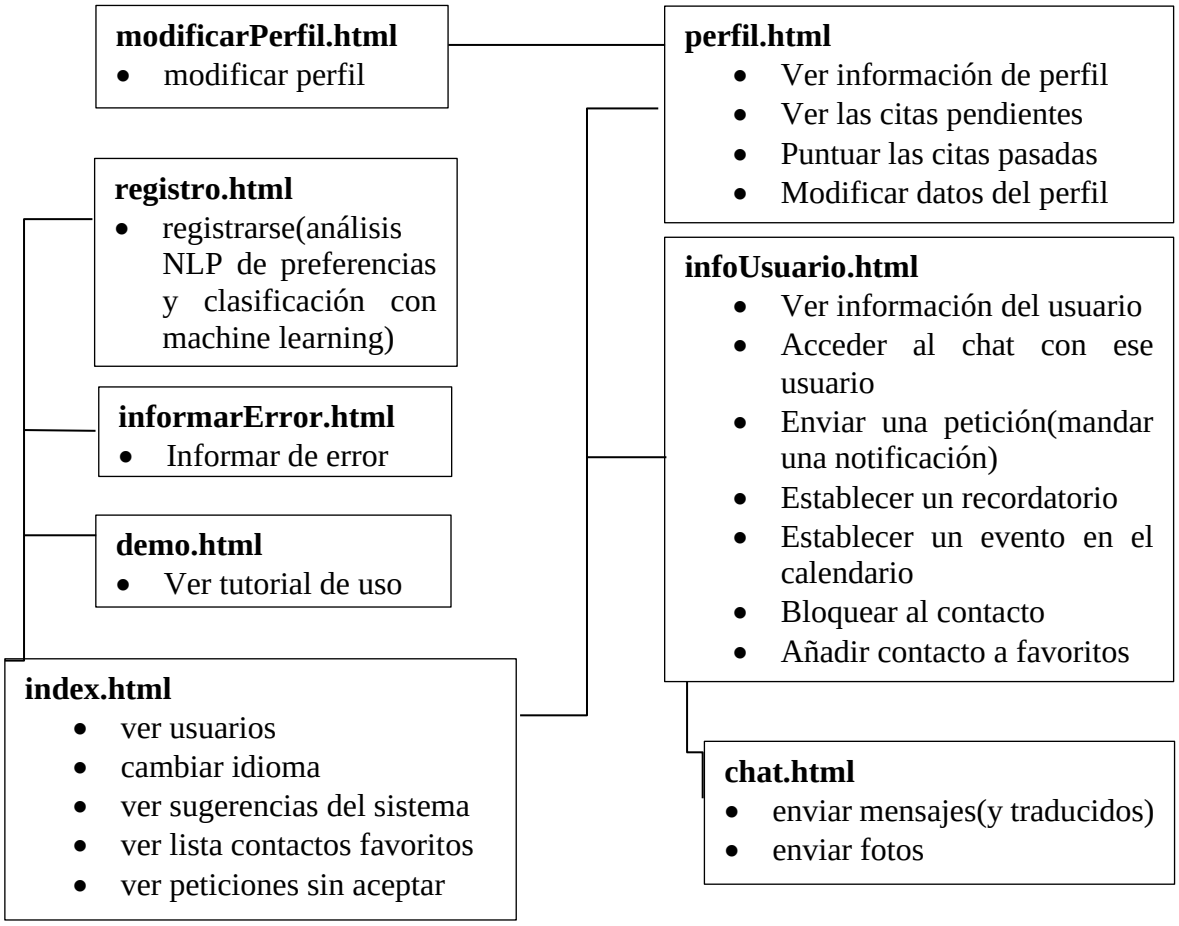
Historia: Establecer un recordatorio de un encuentro

Como : Usuario

Quiero : Poder establecer un recordatorio en mi móvil o un evento en mi móvil de un encuentro que voy a tener.

Para : Que no se me olvide cuándo he concertado un encuentro con alguien.

5.3.1.3 COMPONENTES DEL SISTEMA SEGÚN SU FUNCIONALIDAD



5.2 DISEÑO

5.2.1 DISEÑO DE LA LÓGICA DEL SERVIDOR

Por el lado del servidor vamos a usar como lenguaje PHP el cual es: "PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML." [13] Es un lenguaje que se ejecuta en el lado del servidor, es decir, que nos permite interactuar con el servidor para por ejemplo acceder a la base de datos, acceder a los elementos de los formularios que viajan en el POST o el GET de HTTPS, etc.

A continuación, se muestra un esquema básico del flujo de datos a través de la app, en el que se ve cómo el usuario hace una petición al servidor a través del navegador web(que en nuestro caso puede ser la web o la app del móvil). Desde el servidor se establece una conexión con la base de datos para acceder a la información de disco: [14]

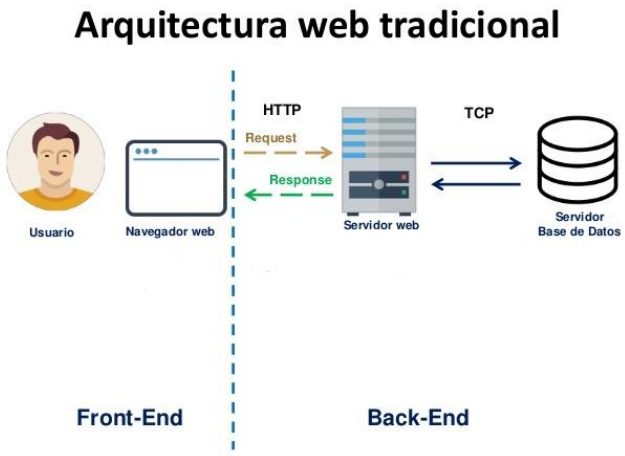


Ilustración 23 Arquitectura web de la aplicación

5.2.1.1 APIS

5.2.1.1.1 API Google Translate

Al conectarnos a la api de Google traductor, podemos mandar mensajes para que sean traducidos. Esta funcionalidad es muy útil ya que se espera que la aplicación sea usada por personas que hablan idiomas diversos. Aun así, esta funcionalidad solo será usada a la hora de mandar mensajes traducidos, y no para traducir la app en sí, ya que como explicaré más adelante, como solo estará en inglés y en español, la traducción se puede realizar con una sencilla función JavaScript. En el siguiente código se muestra la implementación a la llamada a la API haciendo uso de CURL: [15]

```
1. <?php
2.     $apiKey = 'AIzaSyAWYLkqLgnBflzBl8PuvU-sKkjjF0AXQAs';
3.     $text = $_POST["texto"];
4.     $idioma = $_POST["idioma"];
5.     $idiomaOrigen = $_POST["idiomaOrigen"];
6.     $url = 'https://www.googleapis.com/language/translate/v2?key=' .
   $apiKey . '&q=' . rawurlencode($text) . '&source=&target=' . $idioma;


7.     $handle = curl_init($url);
8.     curl_setopt($handle, CURLOPT_RETURNTRANSFER, true);
9.     $response = curl_exec($handle);
10.    $responseDecoded = json_decode($response, true);
11.    curl_close($handle);
12.    echo 'Original: ' . $text . '. <br>Traducido: ' .
   $responseDecoded['data']['translations'][0]['translatedText'];
13. ?>
```

Podemos observar cómo es un código bastante sencillo en el que primero especificamos en la clave de la API (que la sacamos de la consola de Google, donde podemos configurar todas nuestras suscripciones a las distintas APIs), el texto que queremos traducir y el idioma de origen y destino a los que queremos traducir. En la línea 6 configuramos la url con la que vamos a hacer la llamada a la API, donde ponemos la key, que es clave de la api, la “q” es el texto que deseamos traducir, el source es el idioma del texto que queremos traducir y el

target es el idioma al que queremos traducir el texto. Si no especificamos nada en el source, como es el caso, intentará detectar automáticamente el idioma. En la línea 10, en el "$responseDecoded" irá la respuesta de la API, aunque viene en forma de array, por tanto, la traducción que realmente nos interesa está en:

```
"$responseDecoded['data']['translations'][0]['translatedText'];"
```

5.2.1.1.2 API Google cloud natural language

Según la definición de la página oficial de Google sobre NLP:" Natural Language utiliza el aprendizaje automático para mostrar la estructura y el significado de los textos. Puedes extraer información sobre personas, lugares o eventos, así como comprender mejor las opiniones en las redes sociales y las conversaciones de los clientes. Esta herramienta te permite analizar textos e integrarlos en tu almacenamiento de documentos de Cloud Storage." [16]

Entre las cosas que podemos hacer es analizar el sentimiento de un texto, es decir magnificar la subjetividad del texto. Por ejemplo, si introducimos la frase: "Este texto es malísimo", nos dirá que tiene una valoración negativa. También podemos analizar cada una de las palabras de la frase que introducimos, para poder saber cuáles son sustantivos, adjetivos, o verbos. También es capaz de identificar entidades en el texto tales como direcciones (como ciudades o calles) o nombres propios.

Antes de nada, nos debemos dar de alta en la API para poder acceder a la clave que necesitamos para hacer las llamadas a la API. Esto lo hacemos a través de la página Google Cloud Platform, creándonos un proyecto previamente. Después debemos descargar con "Composer" (explicado en el apartado 5.1.1.2 Composer) la librería de Google Cloud NLP para php con el siguiente comando :

```
composer require google/cloud-language
```

A continuación, se muestra una versión reducida(ya que el original tiene más sentencias case) del código necesario para implementar la llamada a la API [17] y extraer del texto que introducimos los sustantivo y los verbos en infinitivo, que será al final la información que guardemos del campo de hobbies/preferencias que introduzca el usuario a la hora de registrarse. Vemos como el archivo recibe dos parámetros a través del POST, el texto a analizar y el tipo de análisis. A continuación, se mete en un switch case en la que en función de la variable análisis se meterá en un case y otro.

```
1.  <?php
2.      require __DIR__ . '/vendor/autoload.php';
3.      use Google\Cloud\Language\LanguageClient;
4.      $projectId = '689660883871-
    m6ind7hfnqhejsgg3l25uabv4eceqjaj.apps.googleusercontent.com';
5.      $language = new LanguageClient([
6.          'projectId' => $projectId
7.      ]);
8.      $text = $_POST['texto'];
9.      $analisis = $_POST["analisis"];
10.
11.     switch ($analisis)
12.     {
13.         case 'verbosInfinitivoYsustantivos':
14.             $annotation = $language->analyzeSyntax($text);
15.             $tokens = $annotation->tokensByTag('VERB');
16.             $tokens2 = $annotation->tokensByTag('NOUN');
17.             foreach ($tokens as $token) {
18.                 if($token['partOfSpeech']['mood'] == "MOOD_UNKNOWN"  &&
                        $token['partOfSpeech']['number'] == "NUMBER_UNKNOWN" &&
                        $token['partOfSpeech']['person'] == "PERSON_UNKNOWN" )
19.                  {
20.                     echo $token['text']['content'] . " ,  ";
21.                  }
22.             }
23.             foreach ($tokens2 as $token2)
24.                 echo $token2['text']['content'] . " ,  ";
25.         break;
26.
27.         case 'sentimientos':
28.             $annotation = $language->analyzeSentiment($text);
29.             $sentiment = $annotation->sentiment();
30.             echo $sentiment['score'] . ', ' . $sentiment['magnitude'];
31.         break;
32.     }
```

5.2.1.2 Bases de datos de la aplicación

Con el objetivo de persistir la información que van generando los usuarios cuando usan la app, dicha información se irá guardando en bases de datos y tablas para que cuando el usuario vuelva a usar la aplicación el sistema guarde la información de su perfil.

La base de datos que se va a usar es del tipo MySQL, y se va a usar phpAdmin, una interfaz muy sencilla de usar y que simplifica mucho la visualización, creación y edición de los datos en las distintas tablas que nos vamos creando. El puerto usado para conectarse a la base de datos es el puerto 3306.

El sistema tiene dos bases de datos principales, la “primerabbdd” y la “chat1”. La “fcm-push” se ha usado simplemente en la fase de desarrollo del proyecto, pero no tiene ya ninguna utilidad real.

5.2.1.2.1 DAO

Antes de ponerse a hablar de las distintas bases de datos y tablas usadas por la aplicación, hay que hablar de cómo se establece la conexión y se realizan las distintas consultas a las BBDD.

Según la definición de Wikipedia, la DAO es : “Componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo.” [18] Este es un patrón de diseño, una forma concreta de desarrollar el software que se encarga del acceso a las bases de datos. Consiste en intentar centralizar en un mismo código todo el código que se encarga de acceder y hacer las consultas en las distintas tablas.

Por ello, se ha implantado dicha DAO desarrollando un código php donde se encuentran casi todas las llamadas a la base de datos. Este código consiste en un switch case, donde cada case es una consulta distinta. Para hacer una consulta, se llama a través del método POST a este código, y uno de los atributos del POST tiene como id “consulabbdd”, y ahí ira el tipo de consulta que se quiere hacer, lo que marcará en que case se meterá.

5.2.1.2.2 Sesión

También es importante remarcar el papel que ha tenido la sesión en este proyecto. La sesión web es una estructura de datos en las que se pueden ir guardando elementos a medida que se va usando la página web por parte de un usuario. Dicha sesión se guarda hasta que cerramos el navegador. En php existe la variable $_SESSION, que es una de estructura de datos en los que los elementos se guardan en parejas de clave valor. Es decir, si queremos guardar el valor del nombre del usuario para que siempre pueda estar accesible cuando está usando la página web, guardaríamos: $_SESION[“nombre”] = “Alberto”.

En nuestro caso es aún más útil e importante, ya que la Apache Cordova, que es la herramienta que estamos usando para luego convertir nuestra aplicación web en una aplicación móvil no permite mostrar una página que sea mezcla de HTML y php, sino que tiene que ser puramente HTML, y luego con Ajax (JavaScript) cargar de forma asíncrona lo que necesitemos del servidor (a través de los archivos php). El problema está en que si pasamos de una página HTML a otra enviando un formulario en la que sus datos viajan a través de HTTPS con el método POST o GET, para acceder a ellos desde la página HTML de destino necesitamos código php para acceder a ello, el cual no podemos ejecutar en la página HTML ya que apache Cordova no nos permite. La solución a esto es usar la sesión de php. Para ello, cuando se van a mandar datos en un formulario de un HTML a otro, primero se guardan estos datos en la SESSION, y una vez que ya están guardados en ella, desde el HTML de destino se cargan dichos datos haciendo uso de Ajax. Debido a que esta situación se puede dar varias veces, no hay problema ya que los datos se pueden sobrescribir en la sesión.

Para usar la sesión en php, lo primero que debemos hacer es ejecutar la función sesión_start(), lo que permite que podamos usarla. A continuación, para escribir en ella, simplemente escribimos $_SESSION[“clave”] =valor, y cuando queramos acceder a ese valor simplemente necesitamos conocer su clave. En la sesión se puede guardar multitud de variables diferentes, desde int, string hasta arrays.

5.2.1.2.3 Primeraddbb

En esta base de datos se van a guardar 3 tablas, la tabla1 que es donde se guarda toda la información relativa a los usuarios, la tabla errores, en la cual se guardan los distintos errores que vayan reportando los usuarios cuando usan la app y por último la tablaimagenes, en la cual se guardan las fotos que van enviándose los usuarios cuando usan el chat.

5.3.1.2.3.1 tabla1

Como ya he dicho en esta tabla cada fila va a representar a un usuario, y cada columna un atributo concreto de un usuario. Las columnas presentes en esta tabla son las siguientes:

1. **id**: De tipo int (5). Es el identificador único del usuario dentro del usuario, aunque a niveles prácticos funciona como una contraseña para poder validar al usuario cuando accede al sistema. Es la primary key de la tabla junto a la N, que es el nombre.
2. **N:** De tipo varchar (50). Se refiere al nombre del usuario.
3. **uni**: De tipo varchar (50). Es el nombre del centro al que pertenece el usuario (puede ser ICAI, ICADE, Cantoblanco, Ciempozuelos).
4. **idioma**: De tipo varchar (70). Se va a guardar una lista de los idiomas separadas por comas que quiere aprender el usuario unidos al nivel que tiene en esos idiomas. Un ejemplo podría ser: frances1, ingles3. Esto significa que el usuario está aprendiendo francés y tiene nivel 1(A1) e inglés, con nivel 3(un B1). Los niveles de idiomas son 6, y son A1, A2, B1, B2, C1 y C2.
5. **nativo**: De tipo varchar (30). Es el idioma nativo que tienen, que puede ser inglés, francés, español, italiano…
6. **hora**: De tipo varchar (70). Es una lista de las horas a las que el usuario está disponible para tener un intercambio de idiomas. Un ejemplo de su contenido podría ser: jueves 16:07, martes 14:00. Esto significa que el usuario puede todos los jueves a las 16:07 y los martes a las 2 de la tarde.
7. **tokens**: De tipo varchar (300). Es la cadena alfanumérica que se genera por parte de Google para poder mandar al usuario una notificación push (evidentemente identifica

unívocamente a cada usuario). Como se puede observar es un número de muchos dígitos, en torno a 160 dígitos.

Un ejemplo podría ser:

cpToHqK0m54:APA91bFLqzyJxwhyBXe7sjd7Wi_M9_f2arSgeiRNDJWlcMj4cZU4MZQ2gbWFpHJiVM4mxWMnzDGR20oC1bOSclWXm88_6WYSgBMF4ZH8fWqFcyIbciYxfelGL2AZrG7UBdmpyWgVpoYI

8. **preferencias**: De tipo varchar (1000). Es un string que representa las preferencias que tienen los usuarios, como por ejemplo el fútbol, los deportes, las series ,etc.
9. **activo**: De tipo int (11). Es un número, o cero o uno, que representa si el usuario está activo o no en ese momento. Esto permite que los usuarios sepan qué usuarios están usando en ese momento la app, ya que aparece junto a ellos un círculo verde que les muestra como online.
10. **disponible**: De tipo varchar (30). Solo puede tomar los valores cero o uno, y representa si el usuario justo en ese momento está disponible para tener un intercambio de idiomas. Pasado un tiempo (de manera predeterminada una hora), el usuario automáticamente pasa a no estar disponible.
11. **peticionesenviadas**: De tipo varchar (300). Es una lista que representa todas las peticiones hechas por ese usuario a otros usuarios para poder tener un intercambio de idiomas.

    Un ejemplo podría ser: 2#jueves 19:00?07-05-2020, 2#miercoles 14:07?06-05-2020 Como se puede ver las diferentes peticiones están separadas por comas y cada una de ellas está compuesta por 3 elementos: identificador del usuario al que le enviamos la petición # hora a la que queremos tener el encuentro? fecha del encuentro.
12. **peticionesrecibidas**: De tipo varchar (300). Es la misma que las peticiones enviadas solo que estas son las que recibe un usuario.
13. **citas**: De tipo varchar (300). Son las peticiones recibidas por un usuario que han sido aceptadas. Siguen la misma estructura que las peticiones enviadas y recibidas: id usuario # hora? fecha

14. **navegador**: De tipo varchar (30). Solo puede valer cero o uno, y representa si ese usuario está usando la app en el navegador (valor uno) o en su versión de móvil (valor cero).
15. **bloqueado**: De tipo varchar (30). Es una lista en la que se guardan los id de los usuarios que has bloqueado, y que por tanto no te podrán enviar notificaciones ni mensajes .
16. **fecharegistro**: De tipo varchar (30). En este campo se guarda el string que representa la fecha en la que se registró por primera vez el usuario. Esto le permite al sistema que usuarios son los últimos que se han registrado, y por tanto pondrá en la lista de usuario una etiqueta que los marque como nuevos usuarios.
17. **puntuacioneshechas**: De tipo varchar (600). Después de haber tenido un encuentro, cada uno de los usuarios puede puntuar dicho encuentro, con una nota de 1 a 5. En ese campo se guardan todas las puntuaciones hechas por un usuario. Un ejemplo podría ser: 2#5,2#4. Como se puede observar, se guardan dos elementos, por un lado, el id del usuario al que puntuamos y después la nota que hemos puesto.
18. **amigos:** De tipo varchar (600). En este campo se guarda una lista (los elementos separados por comas) en la que estarán los id de los usuarios que has guardado como amigos.
19. **carrera**: De tipo varchar (30). En este campo se guarda la carrera que está haciendo ese alumno.

5.3.1.2.3.2 tablaerrores

En esta tabla cada fila representa un error encontrado por un usuario. Cada fila está compuesta por 3 campos, el usuario (el identificador y nombre del usuario que ha reportado el error), el error (una cadena que representa una descripción del error), y por último la fecha en las que se reporta el error.

5.3.1.2.3.3 tablaimagenes

En esta tabla se va a guardar las imágenes en formato blob que van enviado los usuarios en el chat que pueden iniciar con otros usuarios. Cada fila representa una imagen, y tiene 3 columnas: El nombre de la imagen, la imagen en sí (en formato BLOB) y por último el tiempo, que es un campo que no se usa.

5.2.1.2.4 fcm-push

Esta base de datos está formada por una sola tabla, la “token”, en la que en cada fila se va a guardar un token, y que tiene 3 columnas: la id(es un identificador del token, que básicamente es un contador que va sumando uno cada vez que se añade un nuevo token), la columna token, donde se guarda el token, y por último la columna created_data, en la cual se guarda la fecha en la que se ha registrado el token.

Esta tabla no tiene ninguna utilidad práctica para la app, simplemente sirvió para la fase de desarrollo del sistema como una forma de establecer un registro de todos los tokens que se iban registrando.

5.2.1.2.5 Chat1

En esta BBDD se van a ir generando tablas donde se van a guardar los mensajes enviados entre dos usuarios. El nombre de la tabla va a contener el identificador de los dos usuarios que participan en ese chat.

Cada tabla va a contener 4 columnas, y cada fila a representar un mensaje enviado. La columna id es un identificador de cada mensaje, que es simplemente un contador que va sumando uno en cada mensaje nuevo. Este campo va a ser la clave primaria de la tabla. El campo nombre es el nombre del usuario que ha mandado el mensaje. La columna mensaje es un string que representa el mensaje que se ha mandado, y por último la columna fecha, donde se registra la fecha y la hora a la que se ha mandado ese mensaje. De manera predeterminada lo que se va a insertar en este campo es el valor "CURRENT_TIMESTAMP".

A la hora de insertar las imágenes, en la tabla lo que se insertará es una cadena que siempre será igual (alber487!##?) unido al nombre de la imagen. De esta manera, cada vez que se lea la tabla para volcar los mensajes, y el sistema vea que un mensaje empieza por esa cadena en concreto ("alber487!##?") sabrá que se trata de una imagen. Por tanto, cogerá el nombre de la imagen y buscará esa imagen en la tabla imágenes. En el HTML se pondrá un link a una página donde se mostrará la imagen.

Otra peculiaridad de esta base de datos es que cada tabla se va a ir generando de manera automática sin que el usuario lo sepa. Es decir, cada vez que el usuario inicia por primera vez un chat con un nuevo usuario, se va a crear una tabla nueva donde se irán guardando los mensajes intercambiados entre ambos. Sin embargo, el otro usuario cuando se meta por primera vez volcará los mensajes de la tabla ya creada, y no de una nueva, ya que el sistema comprobará cada vez que se meta si ya había una tabla que represente el chat entre esos dos usuarios.

5.2.1.3 Ajax

Según el artículo de Wikipedia Ajax es:" acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano." [19]

Aunque Ajax es una tecnología que se ejecuta del lado del cliente, es el módulo que nos permite comunicarnos con el servidor, hacerle peticiones, mandarle datos y recibir su respuesta, y por tanto he decidido meterla en el apartado del diseño de la lógica del servidor. Un ejemplo de cómo podría ser una sencilla función Ajax es el siguiente:

```
1. $.ajax(
2. {
3.             type: 'POST',
4.             url: "https://lenguajes.ddns.net:443/phpConsultaSESSION.php",
5.             data :{"consultaSesion" : "nombreImagenBD", "tipo" :"2" },
6.             success: function(result3)
7.             {
8.               alert(result3);
9.              },
10. });
```

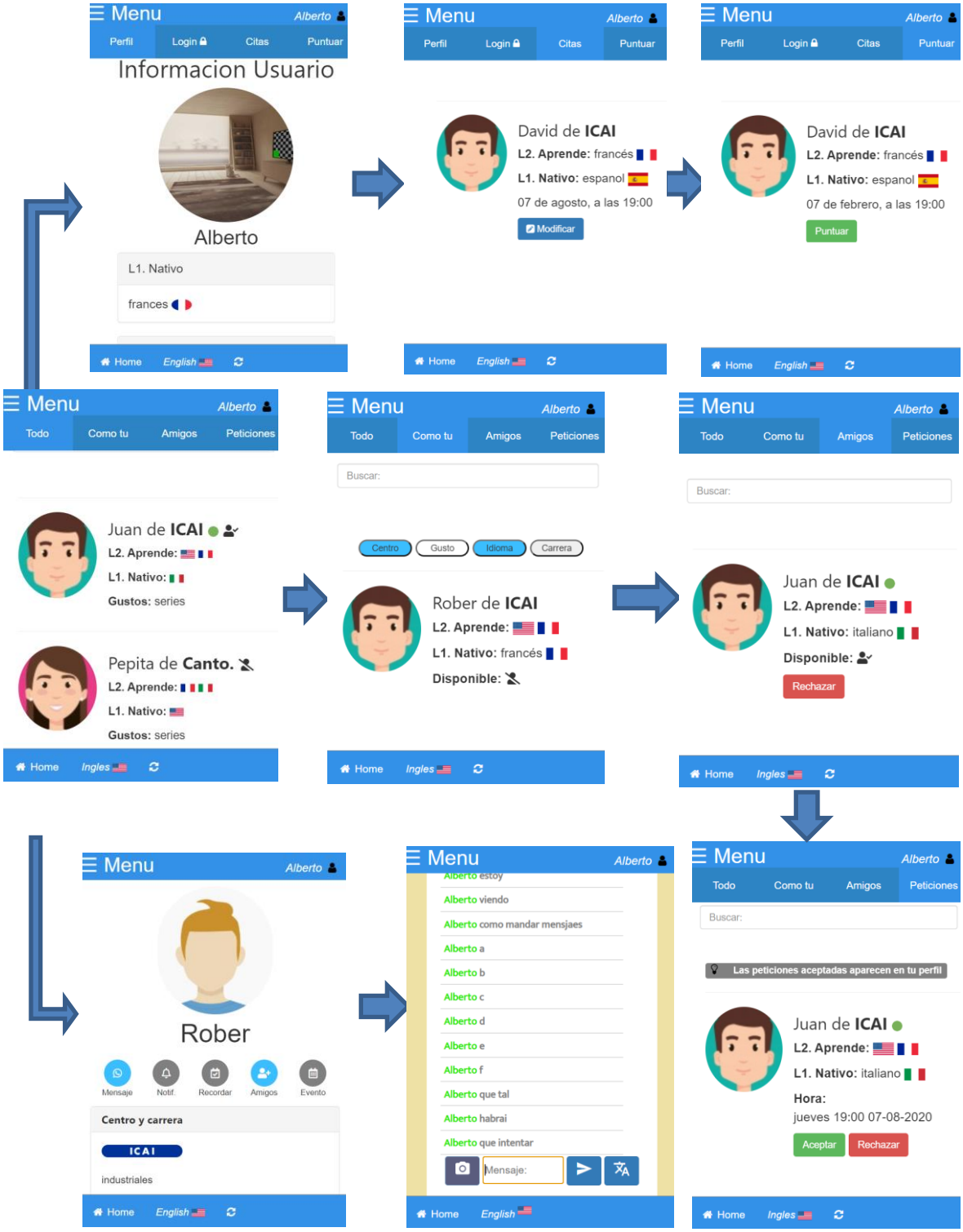
Podemos ver como se deben rellenar una serie de campos, ente los que destacan el type(si se usa POST o GET), la url a la que vamos hacer la consulta, data, que es donde van a ir los datos que le vamos a pasar en la consulta(puestos en forma de clave valor) y por ultimo tenemos la función success que se ejecuta cuando la petición se ha concluido con éxito y se recibe la respuesta del servidor en la variable "result3".

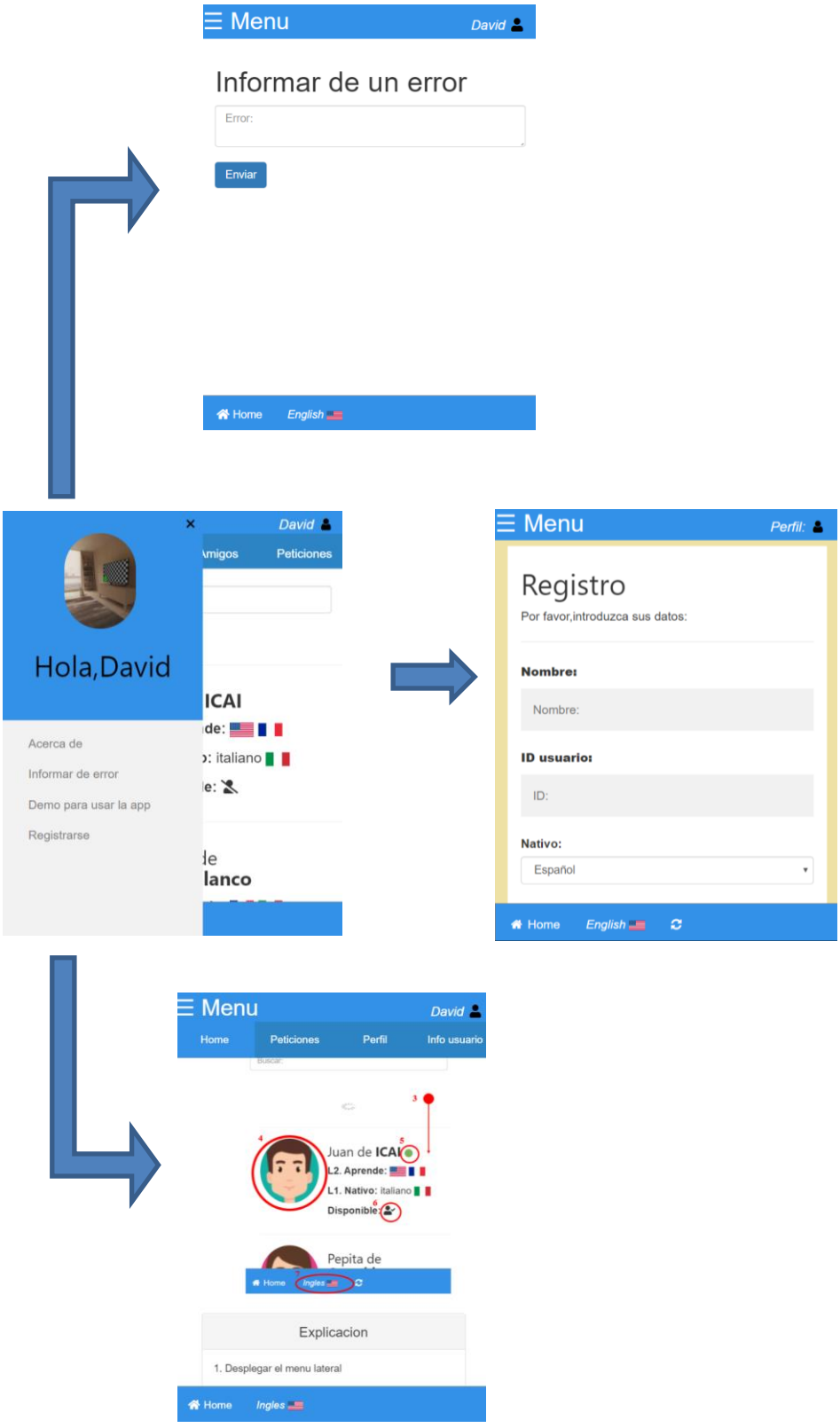
Para permitir que se cargue información a través de Ajax de una página cuyo certificado estaba auto firmado hubo que hacer una pequeña modificación en uno de los archivos de configuración de la aplicación en la plataforma IOS, concretamente en el archivo AppDelegate.m, que está en la carpeta tuProyecto/platforms/ios/HelloWorld/Classes/AppDelegate.m:

```
@implementation NSURLRequest(DataController)
+ (BOOL)allowsAnyHTTPSCertificateForHost:(NSString *)host{ return YES; }
@end
```

5.2.2 DISEÑO DEL INTERFAZ VISUAL

5.2.2.1 Diagrama de Navegación




Menu
David
Informar de un error
Error:
Enviar
Home
English
Hola,David
Acerca de
Informar de error
Demo para usar la app
Registrarse
Amigos
Peticiones
ICAI
Menu
Perfil:
Registro
Por favor,introduzca sus datos:
Nombre:
Nombre:
ID usuario:
ID:
Nativo:
Español
Home
English
Menu
David
Home
Peticiones
Perfil
Info usuario
Juan de ICAI
L2. Aprende:
L1. Nativo: italiano
Disponible
Pepita de
Home
Ingles
Explicacion
1. Desplegar el menu lateral
Home
Ingles

5.2.2.2 Swipper.js

Se va a usar una pequeña librería desarrollada por un tercero para poder deslizar distintas pantallas con un movimiento fluido horizontal. [20]

Para ello solo hay que inicializar swiper en nuestro código HTML, e implementar una serie de funciones que controlaban el movimiento como por ejemplo una función que te avisaba cuando se desplazaba de uno a otro, usada para recargar la página en ese momento.

Otra opción que se configuró es que el tamaño de la página se adaptase a lo que hubiese dentro, de manera que, si en una pestaña no había nada, no hubiese scroll en vertical.

A continuación, se puede ver el código js necesario para inicializar el swipper, y las funciones usadas para actualizar el contenido. Por ejemplo, la función slideChangeTransitionEnd es la que se lanza cuando se acaba la transición entre ventanas.

Desde cualquier punto de nuestro código podemos llamar al objeto swipper, como podemos ver en la línea 10, en la cual después de obtener la variable swipper, llamamos al método slideTo() que nos permite movernos a la ventana que queramos, y el método update(), que actualiza el swipper. De la línea 14 adelante aparece el código que inicializa la variable swipper. Es importante destacar dos líneas, la de autoHeight: true, que permite que el tamaño en vertical de las distintas ventanas se rescale en función del tamaño que necesite para mostrar la información que contiene. Por último, debbuger: true, que nos permite acceder a las distintas funciones que se van lanzado a medida que vamos usando el swipper.

```
<script>
    var myPlugin = {
      name: 'debugger',
      params: {    debugger: false,    },
      on: {  slideChangeTransitionEnd: function () {       },
      },
    };
    function clickear(index)
    {
      var mySwiper = document.querySelector('.swiper-container').swiper;
      mySwiper.slideTo(index);
      mySwiper.update();
    }
    Swiper.use(myPlugin);
    var swiper = new Swiper('.swiper-container', {
      autoHeight: true,
      pagination: {el: '.swiper-pagination', clickable: true,    },
      navigation: {nextEl: '.swiper-button-next',prevEl: '.swiper-button-prev',
      },
      debugger: true,
    });
  </script>
```

5.2.2.3 Pull to reload.js

Con el fin de poder actualizar la página de manera similar a la que se hace en otras aplicaciones, que es arrastrando la página hacia abajo para actualizar la información que se vuelca en la página. Cuando se desplaza hacia abajo se muestra un gif de la típica rueda que gira representando que se está actualizando. Para implementarlo, solo ha había que inicializarlo con una pequeña función de js desde el código HTML, y llamar al principio del código a un archivo JavaScript donde se encuentra todo el código fuente. En él se podían cambiar varios parámetros, pero solo se introdujo el gif y el tiempo que está mostrándose. [21]

5.2.2.4 W3.CSS Modal

Es una combinación de código HTML, css, y JavaScript que nos permite ver una ventana flotante que no ocupa toda la pantalla cuando pulsamos un botón. El código está sacado de la página w3school. [22]

Consiste básicamente en un div HTML que de manera predeterminada está oculto (esto se consigue con css), pero que cuando pulsamos un botón, usando JavaScript cambiamos su propiedad "display" para que se haga visible. De manera análoga, si pulsamos la x de arriba a la izquierda o en cualquier otro punto fuera de la ventana flotante esta se cierra.

5.2.2.5 Snackvar

El snackvar es un pequeño mensaje popup que aparece abajo del todo de la pantalla, y consiste en una combinación de código HTML, css y JavaScript. Está sacado de la página de w3school [23]. Es usado para avisar al usuario cuando se ha registrado correctamente, o cuando ha añadido a un amigo a su lista de contactos favoritos. Aparece durante unos dos segundos y luego desaparece.

Un ejemplo de cómo se vería tanto la ventaja flotante como el snackbar sería:

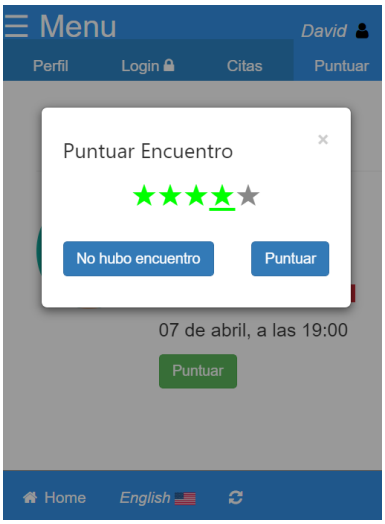


Ilustración 24 Ejemplo ventana flotante 5.4.4 W3.CSS MODAL

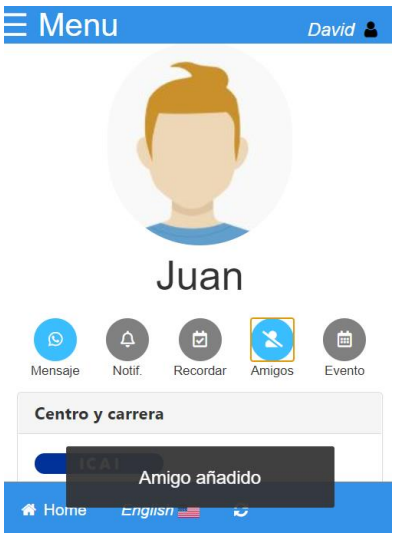


Ilustración 25 Ejemplo de snackbar

5.2.2.6 Sidevar

El sidevar es menú que se encuentra oculto a la izquierda y que se desplaza a la derecha cuando se pulsa un botón, superponiéndose a la ventana principal. Es muy útil porque nos permite guardar en la una serie de links a otras partes de la app sin tener que sobrecargar con más información la pestaña principal.

El código lo he sacado de la página w3school [24], y consiste en dos funciones JavaScript, un div donde se encuentran todos los links y código CSS para dar estilo al menú. Una de las funciones JS es para abrir y otra para cerrar el sidevar cuando se pulsa un botón. Esto se consigue editando su atributo style.display, pasando de block para que se muestre, a none para que se oculte. Con el siguiente código css se consigue ese efecto de desplazamiento cuando abrimos el menú:

```
#main {
  transition: margin-left .5s;
  padding: 20px;
}
```

En el sidevar aparecerá la foto de perfil del usuario (si la tuviese), y 3 links: Registrarse (aunque este desaparecerá una vez que el usuario se haya registrado), Demo para usar la app(donde se pondrán un pequeño tutorial con imágenes para aprender a usar la app) e Informar de un error(donde el usuario podrá reportar cualquier error que encuentre).

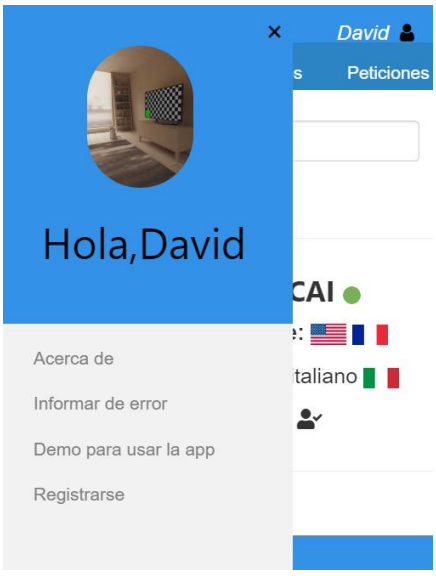


Ilustración 26 Sidevar

5.2.2.7 Full Page Tabs

Siguiendo una plantilla ya diseñada por la página w3school, se ha hecho uso de los Page Tabs [25] , los cuales consisten en una serie de pestañas que se encuentran fijas en la parte superior de la página, y que cuando se va pulsando cada una vamos accediendo a una subpágina distinta, pero sin salirnos de la misma página HTML. Esto se consigue con una mezcla de código HTLM, CSS y JS. Con HTLM creamos una serie de botones que serán las pestañas fijas, y luego creamos una serie de divs que cada uno será cada una de las subpáginas(todos pertenecerán a la misma clase para poderles asignar el mismo estilo con CSS). Por último, un sencilla función JavaScript que se ejecuta cada vez que pulsamos uno de los botones. Esta lo que hace es mostrar la subpágina relacionada con el botón que hemos pulsado, y ocultar el resto, cambiando el color del resto de las pestañas para diferenciarla de la que está en ese momento abierta.

A continuación, se muestra un ejemplo de las pestañas(Todo, Como tú, Amigos, Peticiones) y de cómo se van abriendo distintas subpáginas sin salir de la principal

Ilustración 28 Ejemplo Tab links

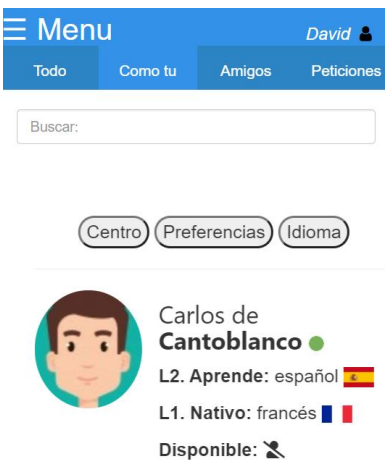


Ilustración 27 Ejemplo 2 tab links

5.2.2.8 Formulario de registro

A la hora de diseñar el formulario de registro se ha seguido una plantilla ya diseñada por la página w3school [26], a la que se han hecho pequeñas modificaciones y ampliaciones para meter más campos de registro de los que tenía.

Entre los campos podemos destacar :

- **El nombre del usuario.**
- **El id del usuario:** En un principio se pensó en este campo como un indicador interno de cada al usuario que no era elegidos por ellos, sino que era asignado por el sistema como una manera de identificar unívocamente a cada usuario. Cuando se supo que no iba a ser posible meter la aplicación en los servidores de comillas, y por tanto los usuarios sí que iban a tener que registrarse, su función pasó a ser más bien la de una contraseña, que como cada usuario debe tener una distinta también es un identificador único de cada usuario.
- **El idioma nativo** del usuario.
- **El idioma que el usuario está aprendiendo y su nivel** (A1, A2, B1, B2, C1 o C2).
- **Un segundo idioma que el usuario esté aprendiendo y su nivel**. De manera predeterminada su valor será ninguno.
- **Las horas disponibles** que tiene el usuario para poder tener un intercambio de idiomas. Se pueden elegir uno o más días pulsando en el botón del día de la semana correspondiente, y eligiendo justo debajo la hora a la que se puede.
- **Centro** donde ese usuario estudia (puede ser ICAI, ICADE, Cantoblanco o Ciempozuelos).
- La **carrera** que ese usuario estudia.
- **La foto de perfil**. Esta funcionalidad es específica de la versión móvil y por tanto en la versión web se ve de gris y sin posibilidad de pulsar en ella.
- **Las preferencias del usuario.** Consiste en una caja de texto donde el usuario puede introducir un texto en el que explique sus preferencias.

5.3 MACHINE LEARNING

Machine learning o aprendizaje automático son un conjunto de técnicas es una rama de la inteligencia artificial con el propósito de desarrollar e implantar técnicas que hagan que los ordenadores puedan "aprender". Esta es una de las tecnologías con más fama actualmente, debido a la gran variedad de aplicaciones que se le pueden dar.

Por tanto, este proceso consiste en primer lugar en introducir una serie de datos de entrada. Después se aplica un algoritmo, entre los que destacan los de aprendizaje supervisado, en el que se le da una serie de entradas y salidas y este debe combinarlas y pode hacer predicciones, y los no supervisados, en el que solo se le dan las entradas y no las categorías en las que los debe clasificar. Por último el modelo, después de haberlo entrenado, hace la predicción.

5.3.1 EMPAREJAMIENTO POR GUSTOS DE USUARIOS

Esto se va a usar para entrenar a la aplicación para que pueda clasificar los usuarios en función de los gustos o preferencias que introduzcan cuando se registran o cuando editan su perfil. De esta manera el sistema extraerá una o varias palabra que resuma toda la frase o frases que meta el usuario, y de esta manera pueda ofrecerle sugerencias en función de sus gustos o aficiones. Para ello primero el sistema debe "tokeneizar" el texto introducido, que consiste en eliminar espacios en blanco y meter cada palabra en una celda de un array para que sea más fácil trabajar con ello, para luego esperar a la predicción que haga el modelo de dicha entrada.

En resumen, esto permite dotar de mayor "inteligencia" a la aplicación, ahorrar memoria ya que ahora en vez de tener que guardar una frase entera solo habrá que guardar una o un par de palabras en la base de datos , y en conclusión dar un mejor servicio al usuario.

5.3.2 PHP NLP TOOLS [27]

Como dice en la página oficial, NlpTools es un grupo de clases php para principiantes o semiavanzados trabajos en procesamiento natural de lenguajes. Nosotros vamos a utilizarlo para entrenar un modelo que clasifique el texto introducido en un tema concreto. Por ejemplo, si introducimos "a mí me encanta jugar al futbol todas las tardes" el modelo lo clasificará como "fútbol", y esa será la información que guardemos de las preferencias/hobbies de ese usuario. Para entrenar el modelo nos generamos un txt en la que en cada fila tenemos una frase y la palabra que representa la categoría de esa frase(por ejemplo fútbol, hockey, etc), separada por una coma:

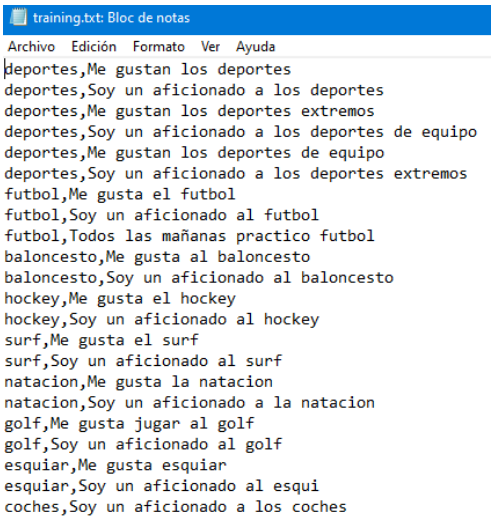


Ilustración 29 Dataset de training

Después me he generado otro txt con las frases para testear que el sistema lo hacía bien.

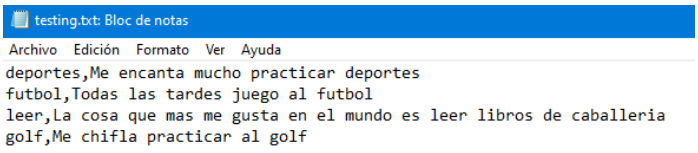


Ilustración 30 Dataset de testing

A continuación se puede ver las predicciones que ha hecho el sistema de cada uno(a la derecha es la predicción y a la izquierda era el valor). Se puede observar cómo ha acertado las 4:

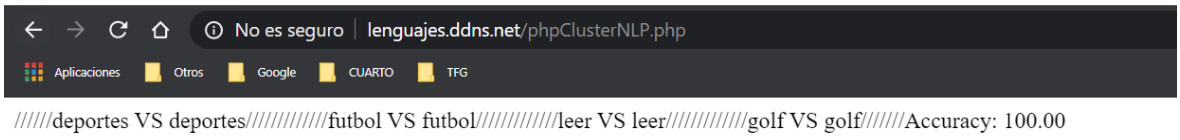


Ilustración 31 Resultado de Clasificación

A continuación, se puede ver una versión reducida del código usado para entrenar y testear el modelo que clasifica: [28]

```
1.  <?php
2.      $tset = new TrainingSet();
3.      $tok = new WhitespaceTokenizer();
4.      $ff = new DataAsFeatures();
5.      foreach ($training2 as $d)
6.      {
7.          //print_r($d);
8.          $tset->addDocument(
9.              $d[0], // class
10.             new TokensDocument( $tok->tokenize($d[1])
11.             )
12.         );
13.     }
14.     $model = new FeatureBasedNB();
15.     $model->train($ff,$tset);
16.     $cls = new MultinomialNBClassifier($ff,$model);
17.     $correct = 0;
18.     foreach ($testing2 as $d)
19.     {
20.         $prediction = $cls->classify(
21.             $grupos,
22.             new TokensDocument(
23.                 $tok->tokenize($d[1]) // The document
24.             )
25.         );
26.         echo "//////" . $prediction. " VS " .$d[0] . "///////";
27.         if($prediction==$d[0]){
28.             $correct ++;
29.         }
30.     }
```

En el código anterior es importante decir que en $training2 iba una matriz de “strings” en la que en cada fila había dos elementos, el tema y el texto vinculado a este.En la variable grupos están guardadas todas las claves sin repetir sacadas del dataset de trainging(futbol, deportes, leer, etc) . De la línea 5 a la 15 tokeneizamos los elementos guardados el matriz que contiene el dataset de los elementos que vamos a usar para entrenar al modelo. Con la línea 14 nos creamos el modelo y en la 15 lo entrenamos. En la 16 nos creamos un

"MultinomialNBClassifier", el cual nos va a permitirnos clasificar la información introducida con un modelo Multinomial Naive Bayes. Por último, de la línea 20 a la 32 testeamos el modelo con un dataset distinto, y vamos sumando en $correct las predicciones que ha conseguido hacer bien.

Con respecto al método kmeans, que nos permite agrupar nuestra información, generando una serie de particiones llamadas clusters, la filosofia seguida es parecida. En la línea 1 del siguiente código nos creamos un "TrainingSet" donde vamos a guardar la información que queremos agrupar. Vemos cómo nos creamos dos tipos de modelos kmeans, uno usando distancias euclídeas y otro usando la distancia del coseno y el ángulo medio. En ambos casos le decimos que nos agrupe la información en 3 grupos. Optamos finalmente por usar el segundo tipo de Kmenas, el de la distancia con el coseno y el ángulo medio. En la variable arrayClavesIngles están guardadas los temas de cada una de las frases del dataset de testeo, las cuales las mostramos para ver como las ha agrupado en los cluster

```
1.  $tset2 = new TrainingSet(); // will hold the training documents
2.  foreach ($trainingIngles as $d){
3.      $tset2->addDocument($d[0],

            new TokensDocument($tok->tokenize($d[1])));

4.  }
5.  $tset3 = new TrainingSet(); // will hold the training documents
6.  foreach ($trainingIngles as $d){
7.      $tset3->addDocument("",new TokensDocument($tok->tokenize($d[0])));
8.  }
9.  $clust = new KMeans(
10.     3,
11.     new Euclidean(),
12.     new EuclideanCF()
13. );
14. $clust2 = new KMeans(
15.     3,
16.     new CosineSimilarity(), // the distance metric
17.     new MeanAngle(), // how we will be creating centroids
18. );
19. var_dump($arrayClavesIngles);
```

```
20.     var_dump($clust2->cluster($tset3, new DataAsFeatures()) );
```

Este es el resultado que obtenemos, en el elemento 0 del array nos dice que elementos van en cada cluster. Vemos como en un cluster nos mete a sport, soccer, golf, cook, sport (los elementos 0,1,4,6,7), en otro a basketball(2) y en el último cluster a los dos read(3 y 5). Se ha equivocado en cook y basketball, ya que cook debería ir en uno aparte y basketball junto al cluster de los deportes

```
C:\wamp64\www\camara\www\phpClusterNLP.php:212:
array (size=8)
  0 => string 'sports' (length=6)
  1 => string 'soccer' (length=6)
  2 => string 'basketball' (length=10)
  3 => string 'read' (length=4)
  4 => string 'golf' (length=4)
  5 => string 'read' (length=4)
  6 => string 'cook' (length=4)
  7 => string 'sports' (length=6)

C:\wamp64\www\camara\www\phpClusterNLP.php:213:
array (size=3)
  0 =>
    array (size=3)
      0 =>
        array (size=5)
          0 => int 0
          1 => int 1
          2 => int 4
          3 => int 6
          4 => int 7
      1 =>
        array (size=1)
          0 => int 2
      2 =>
        array (size=2)
          0 => int 3
          1 => int 5
  1 =>
    array (size=3)
      0 =>
        array (size=4)
          'sports' => float 0.4
          'soccer' => float 0.2
          'golf' => float 0.2
          'cook' => float 0.2
      1 =>
        array (size=1)
          'basketball' => float 1
      2 =>
        array (size=1)
          'read' => float 1
  2 =>
    array (size=8)
      0 =>
        array (size=3)
          0 => float 0.24407105398155
          1 => float 1
          2 => float 1
```

Ilustración 32 Resultado del aplicar Kmean

Al final se ha optado por usar el método de clasificación y no de agrupamiento con kmeans ya que es mucho más preciso a la hora de predecir el tema de cada una de la frases introducidas.

5.4 FUNCIONALIDADES

5.4.1 CHAT

Hemos implementado un sencillo chat de mensajes para los usuarios. Su funcionamiento se basa en ir almacenando los mensajes que se intercambian en una tabla de una base de datos. Para mandar el mensaje, se puede o pulsar el botón de mandar el mensaje(lo más cómodo cuando se usa la versión móvil) o el botón enter (lo más cómodo cuando se encuentra en la versión del navegador).

Mediante una función JavaScript (haciendo uso de Ajax) se irá volcando cada segundo la información de esta tabla en el chat, de manera que si se envía el mensaje el receptor lo recibirá casi al instante.

Cada vez que el usuario se mete en el chat para hablar con alguien, lo primero que se hará al acceder al chat es ejecutar una función de JavaScript que está compuesta por varias funciones Ajax que realizan las siguientes funciones:

- Calcular el nombre de la tabla que se va a usar para ir guardando los mensajes de este chat. Para ello, se va a llamar al archivo phpComprobarBDchat, el cual calcula dos nombres de tablas posibles, o el tabla_id1_id2 o el tabla_id2_id1. A continuación comprueba si alguna de esas tablas ya existe. Si existe, guarda en la sesión el nombre de la tabla y si no, crea la tabla en la base de datos y guarda el nombre de la tabla en la sesión.
- Una vez que sabe el nombre de la tabla, se vuelcan por primera vez todos los mensajes que estaban en la tabla.
- Comprueba si se encuentra en el navegador o en el móvil, y para ello hace una consulta a la “tabla1”, la cual es la tabla que guarda toda la información de los usuario, y la cual tiene un campo que indica si ese usuario está usando la app en el navegador o en el móvil. Esto sirve para bloquear o no el botón de la cámara, ya que si se encuentra en el navegador el usuario no puede usar la función de la cámara.

- Por último, se comprueba la fecha de envío de todas las imágenes que se han mandado en ese chat. Si alguna hace más de 7 días que se mandó, se accede a la taba donde se guardan las imágenes y se elimina.

También se dará la opción al usuario de mandar una foto, instalando previamente el plugin que nos permite tener acceso a la cámara del móvil.

El sistema dará a elegir al usuario la posibilidad o de hacer la foto con la cámara del móvil o sacándola de la galería. Después de realizar la foto, esta se guardará en base64 en la base de datos de las imágenes, y en la tabla donde se guardan los mensajes se guardará el nombre de la imagen para poder acceder a ella en un futuro el usuario. El usuario lo que vera será un mensaje que pondrá: *Pincha para ver la imagen*, para que cuando pinche le lleve a una página donde le muestra a tamaño completo la imagen. Si el usuario se encuentra usando la app en la versión del navegador, le aparecerá como bloqueado el botón para poder mandar fotos. Con respecto al diseño visual del HTML, una parte de él se ha inspirado en un diseño sacado de internet [29] .A continuación se muestra una captura de cómo se vería el chat:

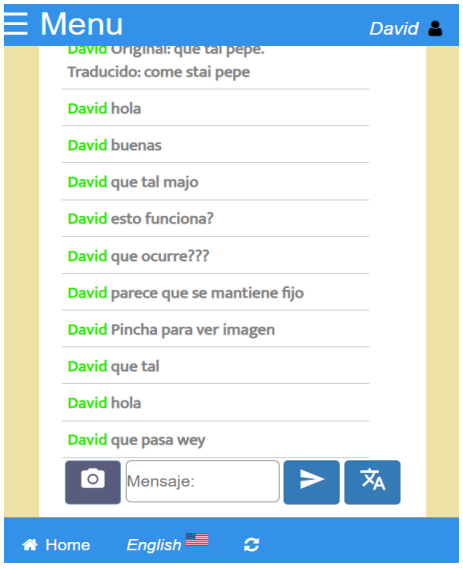
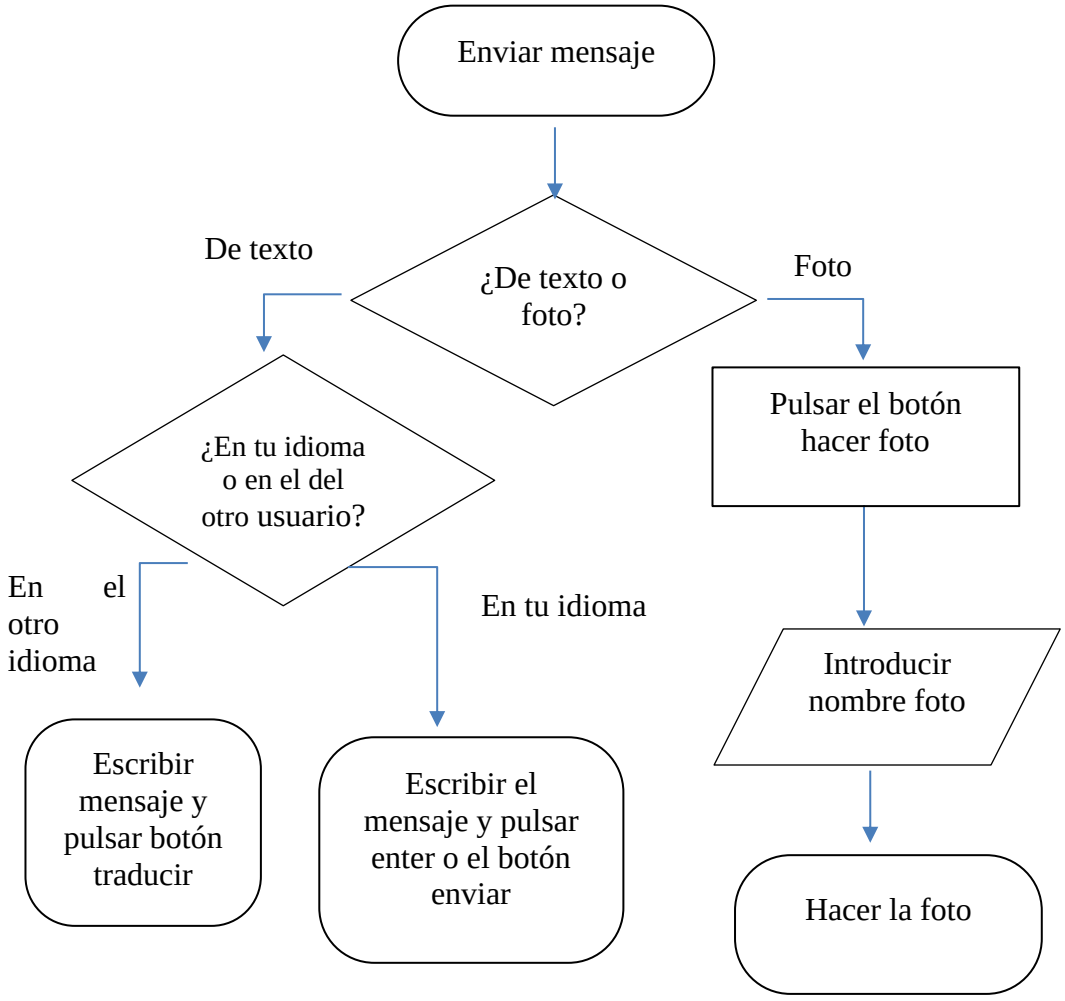


Ilustración 33 Chat de la aplicación

A continuación, se muestra un diagrama de flujo que muestra los pasos que hay que dar para mandar un mensaje.



5.4.2 INTERNACIONALIZACIÓN: CAMBIAR EL IDIOMA DE LA APLICACIÓN

Se ofrece al usuario de poder usar la aplicación en inglés o en español, y además puede cambiarlo en cualquier momento ya que el botón para cambiarlo se encuentra en la barra inferior siempre presente.

Para traducir el idioma, se usa una sencilla función de JavaScript que va cogiendo el texto de las distintas etiquetas presentes en la pantalla y las sustituye por su traducción en inglés.

Debido a que en cada página HTML habrá una serie de etiquetas con texto distintos entre unas y otras, cada página HTML deberá tener una función JavaScript distinta, aunque para no ocupar mucho espacio en el código HTML estas funciones irán aparte en un archivo .js que ira vinculado al HTML.

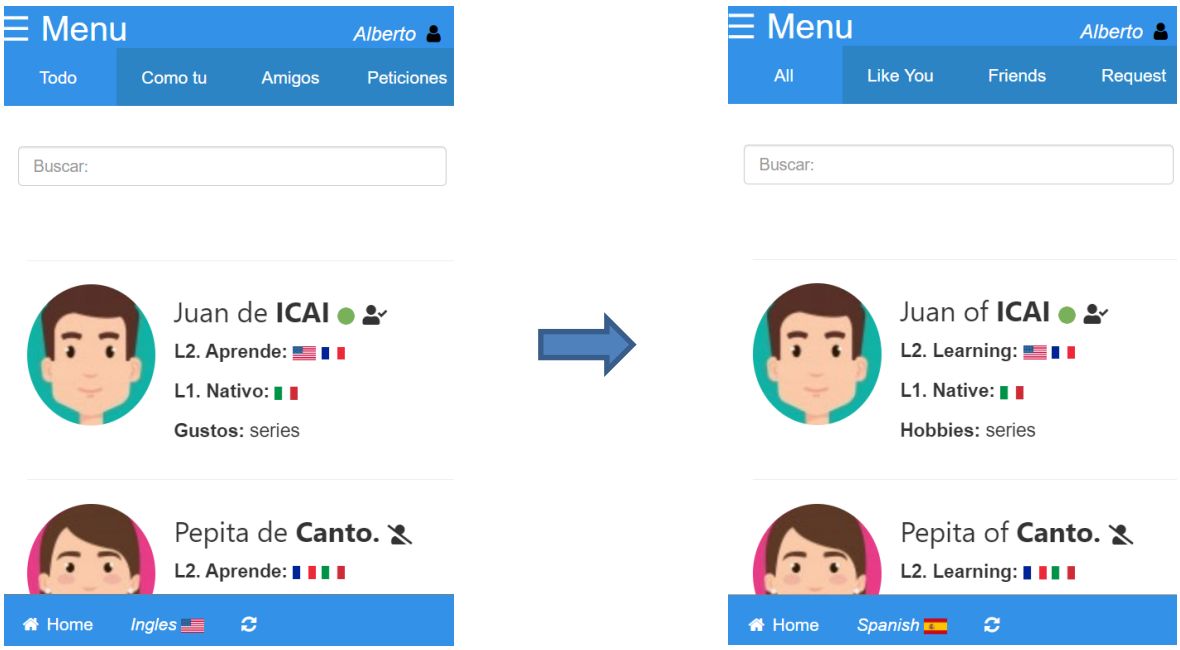


Ilustración 34 Cambiar idioma de la app

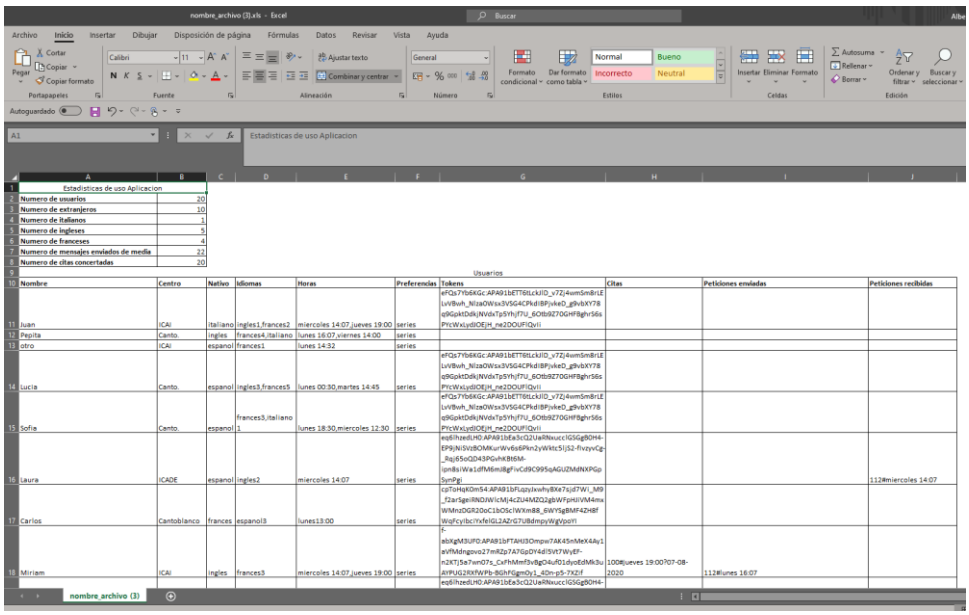
5.4.3 GENERAR EXCEL CON TODA LA INFORMACIÓN DE LA APLICACIÓN

Esta funcionalidad solo estará disponible para el administrador o administradores de la aplicación. Con ella, se podrá descargar desde el navegador un Excel con toda la información de uso relevante de la aplicación. Entre los datos generados están la cantidad de usuarios que tiene, el número de extranjeros y no extranjeros que la usan, la media de mensajes enviados por cada usuario, toda la información de perfil de cada usuario, etc.

Esta funcionalidad se encontrará en el perfil del administrador, en la pestaña perfil, debajo del todo, con el título de "obtener estadísticas" y solo está disponible cuando se use la aplicación desde el navegador.

La implementación de esta funcionalidad se basa en el uso de un sencillo código php en la que se establecen una serie de cabeceras para generar el Excel, y luego se imprime una sencilla tabla HTML para que emule la tabla en Excel.

Ilustración 35 Obtener estadísticas de uso de la app



Estadisticas de uso Aplicacion	
Numero de usuarios	20
Numero de extranjeros	10
Numero de italianos	1
Numero de ingleses	5
Numero de franceses	4
Numero de mensajes enviados de media	22
Numero de citas concertadas	20

Usuarios

Nombre	Centro	Nativo	Idiomas	Horas	Preferencias	Tokens	Citas	Peticiones enviadas	Peticiones recibidas
Juan	ICAI	italiano	ingles1,frances2	miercoles 14:07,jueves 19:00	series	eFQs7Yb6kGc:APA91bETT6tLckJlD_v7Zj4wm5mBrLE LvVBwh_NIzaOWsx3VSG4CPkdiBPjvkeD_g8vbXY78 q9GpktDdkjNV6kTp5YhjfTU_6OtbZ70GHF8ghrS6s PYcWxLydJOEjH_ne2OOUFIQvII			
Pepita	Canto.	ingles	frances4,italiano	lunes 16:07,viernes 14:00	series				
otro	ICAI	espanol	frances1	lunes 14:32	series				
Lucia	Canto.	espanol	ingles3,frances5	lunes 00:30,martes 14:45	series	eFQs7Yb6kGc:APA91bETT6tLckJlD_v7Zj4wm5mBrLE LvVBwh_NIzaOWsx3VSG4CPkdiBPjvkeD_g8vbXY78 q9GpktDdkjNV6kTp5YhjfTU_6OtbZ70GHF8ghrS6s PYcWxLydJOEjH_ne2OOUFIQvII			
Sofia	Canto.	espanol	frances3,italiano 1	lunes 18:30,miercoles 12:30	series	eFQs7Yb6kGc:APA91bETT6tLckJlD_v7Zj4wm5mBrLE LvVBwh_NIzaOWsx3VSG4CPkdiBPjvkeD_g8vbXY78 q9GpktDdkjNV6kTp5YhjfTU_6OtbZ70GHF8ghrS6s PYcWxLydJOEjH_ne2OOUFIQvII			
Laura	ICADE	espanol	ingles2	miercoles 14:07	series	eqSIhzedLH0:APA91bEa3cQ2UaRNxuccIGSGgBOH4- EP9jNiSVzBOMKurWv6s6PkN2yWktc5lj52-fIvzyvCg- _Raj65oQD43PGvhKBt6M- qmBsIWx1dfM6mJ8gFraCd9C995qAGUZMdNXPGp SynPgI			112#miercoles 14:07
Carlos	Cantoblanco	frances	espanol3	lunes13:00	series	cpToHqKOm54:APA91bFLqqylxwhyBXe7ajd7WI_M9 _fZarSgeIRNDJWiclMj4cZU4M2QgbWFpmKiVM4ma WMnsDGR2OoC1bO5c1WXm88_6WY5gBMf42H8F WqFcyibcIYxfeIGL2AZrG7UBdmpyWgVpoYI			
Miriam	ICAI	ingles	frances3	miercoles 14:07,jueves 19:00	series	f- a9JKgM3UFO:APA91bFTAHI3Ompxi7AK45mMeX4Ay1 aVfMdngovo27mRZp7A7GpDY4dI5VtTWyEF- n2KTj5a7wnO7s_CxFhMmf3vBgO4ufO1dyoEdMk3u AYPUG2IXfWPb-BGhFGgmOy1_4Dn-p5-7XZif eqSIhzedLH0:APA91bEa3cQ2UaRNxuccIGSGgBOH4-	100#jueves 19:00?07-08-2020	112#lunes 16:07	

Ilustración 36 Excel con la información de toda la aplicación

En la anterior captura no aparece ningún usuario real, son todos inventados por mí para testear la aplicación, por tanto, no se está mostrando información real de usuarios.

En el siguiente párrafo aparece una versión reducida del código usado para generar el Excel. Las partes más importantes para destacar son la línea 3 y 4, donde se configuran los header del archivo que se va a generar, en este caso un archivo Excel. A continuación se lee la base de datos donde se guarda la información de los usuarios, para ir extrayendo la información de cada usuario que luego se va a mostrar en la tabla HTML que viene justo debajo.

```
1.  <?php
2.          session_start();
3.          header('Content-type: application/vnd.ms-excel;charset=iso-8859-15');
4.          header('Content-Disposition: attachment; filename=nombre_archivo.xls');
5.          $id4 = $_SESSION['id'];
6.          $con = mysqli_connect('localhost','root','','primerabbdd');
7.          if (!$con) {    die('Could not connect: ' . mysqli_error($con));   }
8.          $consulta = "SELECT * FROM tabla1 ORDER BY id ASC";
9.          while($row = mysqli_fetch_array($result)){  ...    }
10. ?>
11. <table border="1" cellpadding="2" cellspacing="0" width="100%">
12.     <caption>Estadisticas de uso Aplicacion</caption>
13.     <tr> <td> <b>Numero de usuarios</b></td> <td> <?= $numUsuarios ?></td>
14.     </tr>
15.     <tr>
16.        <td> <b>Numero de extranjeros  </b></td>
17.         <td> <?= $numExtranjeros ?></td>
18.     </tr>
```

5.4.4 VER INFORMACIÓN USUARIOS

Cuando pinchamos sobre un usuario nos lleva a una página donde nos aparece la información de perfil de ese usuario, en la que podemos ver su nombre y demás información. Entre ella podemos encontrar:

- El centro al que pertenece y la carrera que estudia.
- Su idioma nativo.
- Los idiomas que aprende y el nivel que tiene en ellos(del A1 al C1, 6 niveles).
- Sus preferencias, que pueden ser ver series, el fútbol, etc.
- Las horas a las que está disponible para tener un intercambio de idiomas.
- Botón del chat para iniciar el chat con ese contacto.
- Botón de añadir ese contacto a tu lista de contactos favoritos.
- Botón de añadir un recordatorio local en el móvil.
- Botón para añadir un evento en tu calendario.
- Botón al final de la página para bloquear a ese contacto.

En la lista de usuarios de la ventana principal podemos ver qué usuarios están usando en ese momento la aplicación. Esto es gracias a una funcionalidad de Apache Cordova que nos permite saber qué dispositivos están activos en ese momento y cuáles no. Los que estén online les parece un pequeño circulo verde junto a su contacto dándonos a entender que están online. Debido a que esta funcionalidad depende de usar una función JavaScript específica de Cordova, solo está disponible cuando usamos la aplicación en su versión móvil. Las funciones que se ejecutan cuando se sale de la app y cuando se vuelve a entrar se llaman “onPause()” y “onResume()”.

En el siguiente código podemos ver como cuando se ejecuta onPause, ejecutamos una petición con Ajax para cambiar el atributo "activo" de ese usuario, y se haría de manera análoga con la funcionResumen(), en la que se pone activo =1

```
1.  onPause: function()
2.  {
3.   $.ajax({
4.    type: "get",
5.    url: "https://lenguajes.ddns.net:443/phpActualizarDispositivoEnLinea.php",
6.    data: 'activo=0',
7.    success: function() {     }
8.          });
9.      },
10. },
```

En las siguientes capturas se puede observar cómo se vería la información de un usuario:

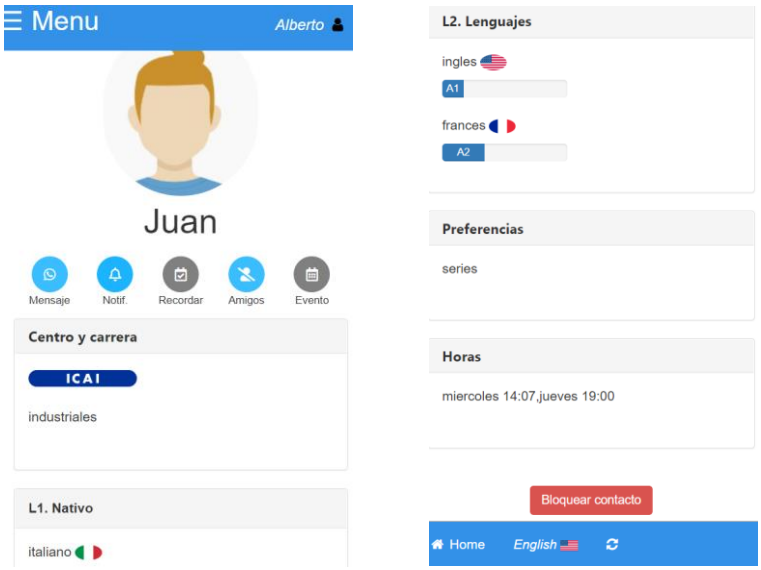
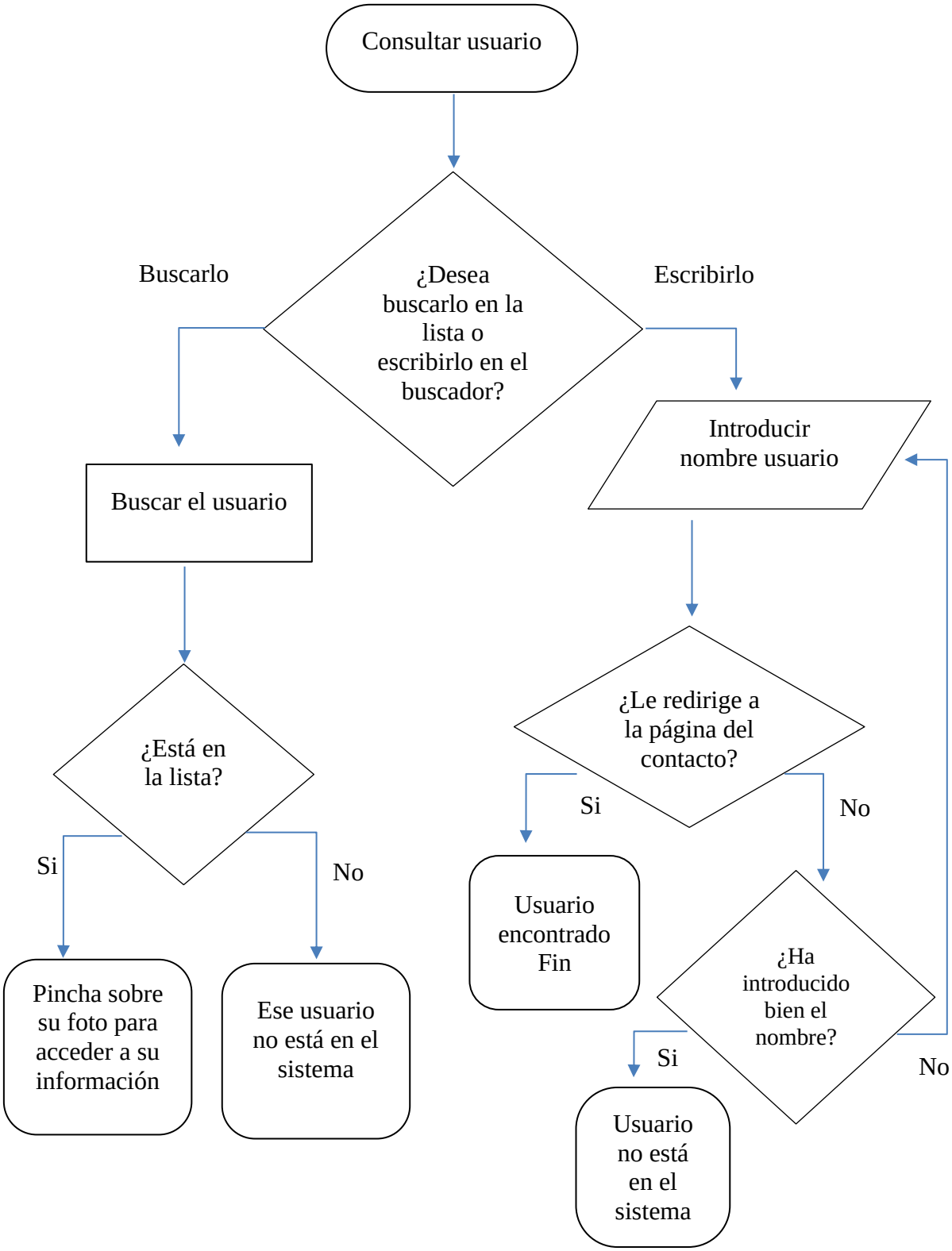


Ilustración 37 información de usuario

5.4.4.1 Diagrama de flujo de consulta usuario



5.4.5 FILTRAR USUARIOS POR CARACTERÍSTICAS

Dentro de la página principal aparece una pestaña llamada Como tu en la que podemos ver los usuarios pero filtrándolos en función de una serie de campos(Centro donde estudia el alumno, sus gustos, el idioma o la carrera). Con respecto al idioma, si pulsamos sobre él nos mostrará los usuarios que son nativos en el idioma que estamos aprendiendo. Con respeto a los filtros, podemos pulsar cualquiera combinación de los botones, dando lugar a 16 combinaciones distintas. Es decir, podemos pulsar centro y carrera y nos mostrará todos los alumnos que estudien en tu mismo centro y tu misma carrera. O si quieres ver qué usuarios son los que más se te parecen, puedes pulsar los 4 filtros y ver aquellos que estudien lo mismo, tengan tus mismos gustos y sean nativos en el idioma que estudias.

5.4.6 VER AMIGOS

En la tercera pestaña de la pantalla principal (Amigos), podemos ver una lista de los contactos que hemos añadido a favoritos, para tenerlos más accesibles que el resto de los usuarios. Al lado de cada contacto favorito aparece un botón de "Eliminar" para eliminar a ese contacto. En las siguientes capturas se puede ver un ejemplo de cómo se vería los usuarios por filtro y a la derecha la lista de contactos favoritos.

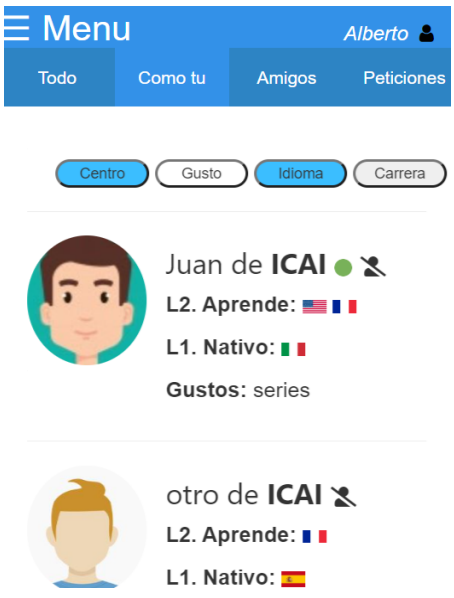


Ilustración 39 Usuarios filtrados

Ilustración 38 Vista de los contactos favoritos

5.4.7 ENVIAR UNA PETICIÓN A UN USUARIO

Desde la página del perfil del usuario se puede enviar una petición para tener un encuentro con ese usuario. Por tanto si pulsamos sobre el botón "Notif." nos aparece una ventana flotante donde nos da a elegir un mensaje de texto para mandarle, junto con la fecha y hora disponible para quedar con esa persona. Esta información es extraída de las horas disponibles de ese usuario, poniendo la fecha que corresponde a esta semana o a la siguiente, dándoselo a elegir al usuario. Después de pulsar aceptar, se mandará una notificación al otro usuario(haciendo uso de las notificaciones push provistas por el servicio FCM).

Después de recibir la notificación(tanto si está abierta o cerrada la app, o encendido o apagado el móvil), la petición le aparecerá al otro usuario en el apartado de peticiones de la página principal de la app. El usuario podrá aceptarla o rechazarla. Si la acepta, la petición pasara a ser una cita y aparece en el apartado citas del perfil . En las siguientes capturas se puede ver cómo sería la apariencia de enviar una petición y de recibirla por parte del usuario:

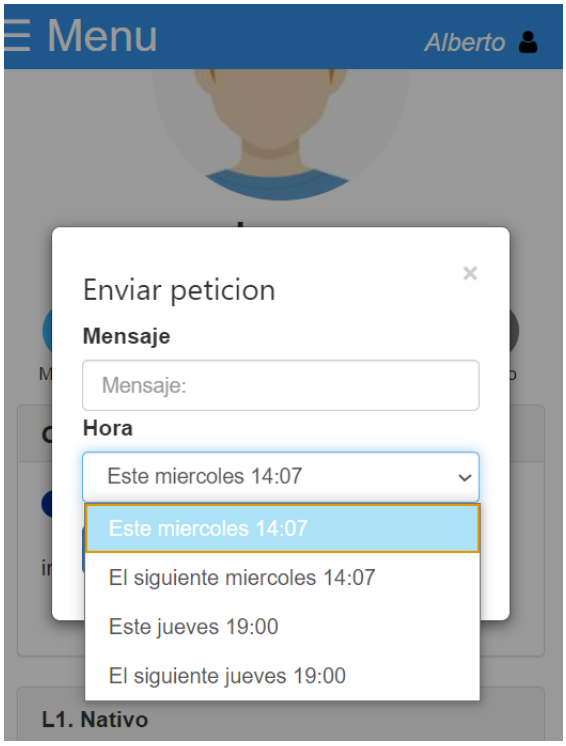


Ilustración 41 Envió de una petición

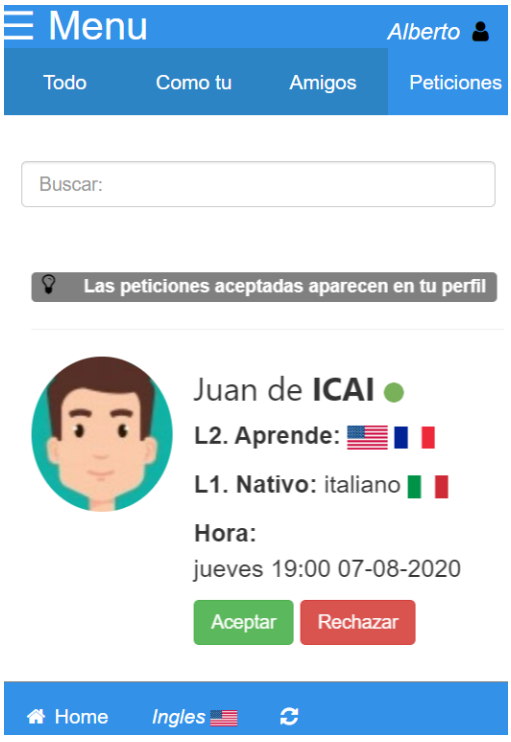


Ilustración 40 Vista de la peticiones recibidas

5.4.8 PUNTUAR UN ENCUENTRO

Dentro del conjunto de peticiones aceptadas, están aquellas cuya fecha ya ha pasado y otras en las que no. El sistema detectará cuales de esas citas su fecha ya ha pasado, y se las mostrará al usuario en el apartado de Puntuar en su perfil, para que pueda puntuar el encuentro. Se ofrecerá una puntuación de uno a cinco, y un botón de "No encuentro", por si al final a pesar de que tenía concertada esa cita al final no se realizó. Dicha puntuación se guardará en la base de datos junto al identificador del usuario al que ha puntuado. Los usuarios no podrán ver la puntuación que le ha puesto el otro alumno. En principio dicha puntuación solo servirá para aumentar la información que obtenemos cuando nos descargamos el Excel con las estadísticas de uso de la app. A continuación se puede ver cómo se vería la ventana flotante en la que se puede puntuar un encuentro.

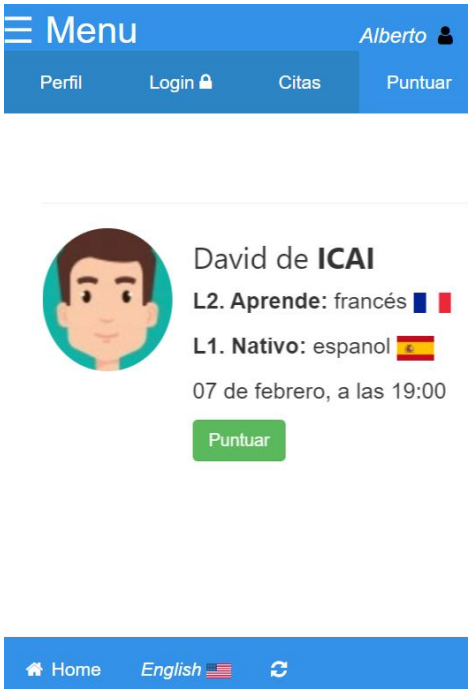


Ilustración 43 Vista de los encuentros que se pueden puntuar

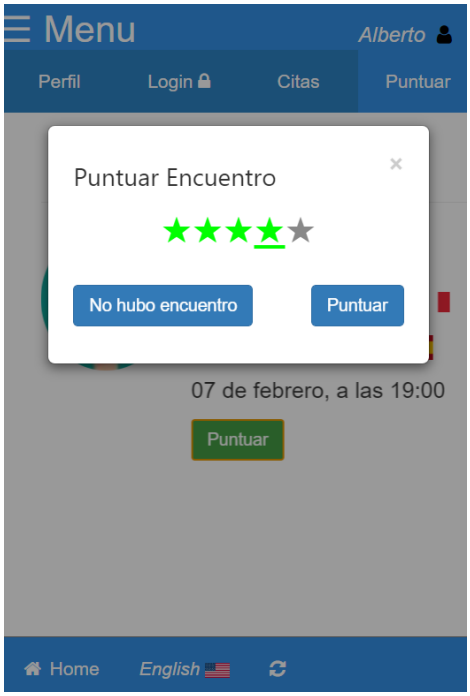


Ilustración 42 Vista de puntuar un encuentro

5.4.9 FUNCIONALIDADES EN NAVEGADOR Y MÓVIL

Debido a que esta aplicación se puede usar en el navegador y en el móvil, hay ciertas funcionalidades que proveen los plugins de apache Cordova a la versión móvil que no están soportados en la versión del navegador. Por tanto, se usa una función JavaScript que se ejecuta cada vez que se accede a la página principal de la aplicación que calcula si la app se usa en el móvil o no. Para ello, se usa un try catch para ejecutar una línea de código que solo se puede ejecutar si se encuentra en la versión móvil (la de "modelo=decive.model"). Por tanto, si está en el navegador, esa línea de código dará error y se irá al catch, donde la variable navegadora se hará que valga uno. Si no da error, la variable navegadora seguirá valiendo 0. Por último se usa una función Ajax para guardar en la columna navegador de la fila de la tabla1 correspondiente a ese usuario el valor de la variable navegador (previamente también se ha guardado en la sesión con el id "navegador").

```
<script>
setTimeout(function(){
        var navegador="0";
        var modelo;
        try {
          modelo = device.model;
        }
        catch(e){
          navegador="1";
        }
        if(navegador=="0") {
        }
        $.ajax(
          {
              type: 'POST',
              url: "https://lenguajes.ddns.net:443/phpConsultaSESSION.php",
              data: {"consultaSesion" : "navegador","navegadorV" : navegador},
              success: function(result)
              {
                $.ajax(
                {
                    type: 'POST',
                    url: "https://lenguajes.ddns.net:443/phpDAO.php",
                    data: {"consultaBBDD" : "guardarNavegador"},
                    success: function(result)   {      },
                });
              },
          });
      }, 3500);
    </script>
```

Una vez que se sabe en qué tipo de plataforma está el usuario, esto nos permite ocultar o mostrar ciertos botones que nos permiten ejecutar ciertas funcionalidades de la aplicación, como pueden ser el mandar notificaciones, establecer un recordatorio, etc. A continuación, muestro una lista de qué plataformas soporta cada una de las funcionalidades de la aplicación:

Funcionalidad	**Soportada en navegador**	**Soportado en móvil**
Mandar mensajes a otros usuarios	Sí	Sí
Mandar fotos a otros usuarios	No	Sí
Mandar notificación cuando mandas una petición	No	Sí (solo en Android)
Enviar un mensaje traducido	Sí	Sí
Actualizar estado online usuario	No	Sí
Cambiar tu foto de perfil	No	Sí
Establecer un recordatorio	No	Sí
Establecer un evento en el móvil	No	Sí
Añadir un usuario a tus contactos favoritos	Sí	Sí

Tabla 3 Funcionalidades soportadas en función de la plataforma

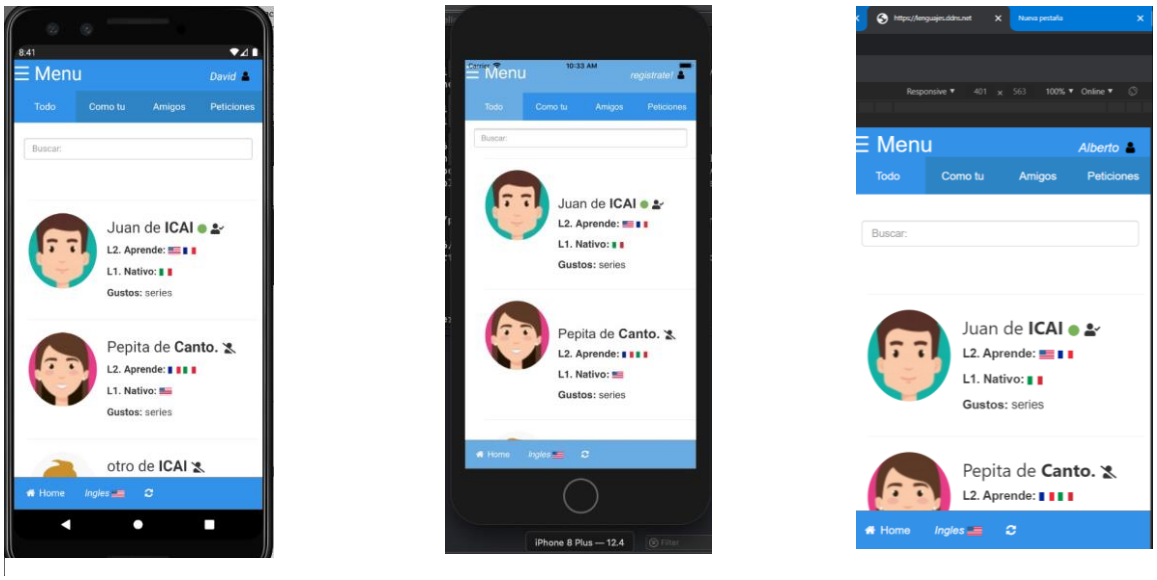


Ilustración 44 Aplicación corriendo en Android ,IOS y en el navegador

5.5 PLUGINS

5.5.1 NOTIFICACIONES

Para poder mandar notificaciones entre usuarios vamos a usar el servicio de Google llamado Firebase Cloud Messaging. Este servicio implementa las notificaciones push, que a diferencia de las notificaciones locales estas son mandadas por el servidor. Entre las múltiples ventajas de usar este servicio es que el dispositivo que recibe la notificación no debe estar corriendo la aplicación ni encendido para recibirla. Todo este apartado lo podríamos haber metido también en el sección de las APIs, pero resulta que para que este servicio funcione necesita tener instalado un plugin, y esa es la razón de que haya metido este apartado en el capítulo de los plugins y no en el de las APIS. Para identificar cada uno de los usuarios cuando te das de alta por primera vez se genera automáticamente un token, que es un código alfanumérico bastante largo que sirve para identificar cada uno de los usuarios. Para obtener este token la primera vez, instalamos el plugin: [30]

"cordova-plugin-fcm-with-dependecy-updated"

Con este plugin lo único que conseguimos es el token, y este lo guardamos en una base de datos donde guardamos la información de cada usuario.

Para poder configurar el acceso a la API de Google, tenemos que crearnos un proyecto, el cual es gratuito, y después descargarnos un archivo "json" generando por Google después de decirle cuál es el identificador de nuestra aplicación(com.app.helloa21a).

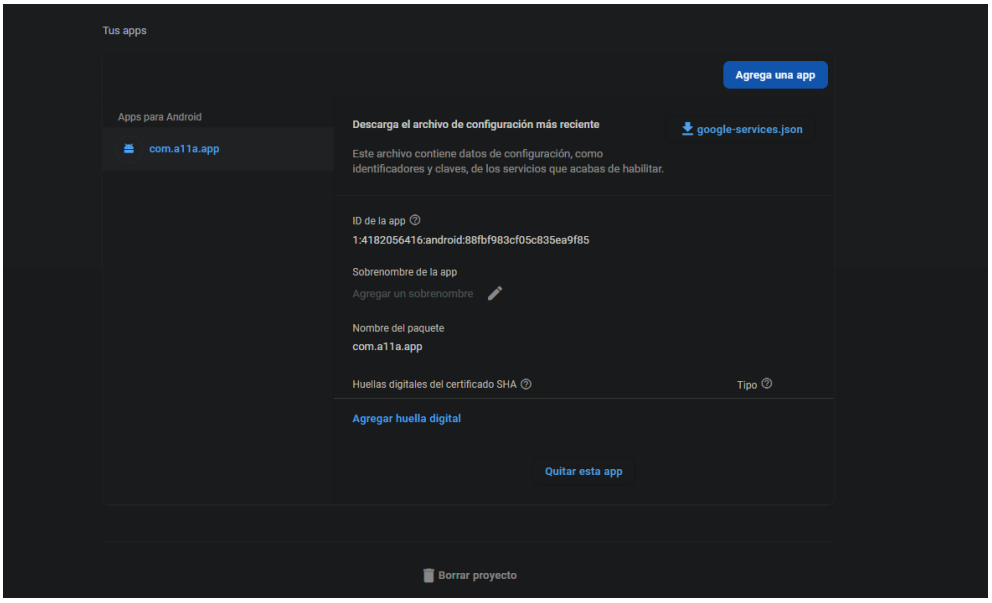


Ilustración 45 Obtener el json para usar FCM

Posteriormente debemos conseguir la clave de la API, la cual se encuentra en los ajustes generales del proyecto:

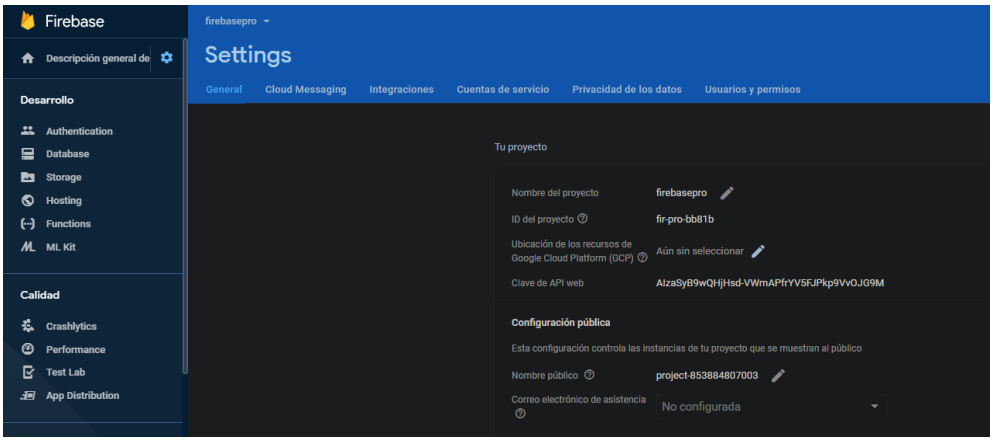


Ilustración 46 Obtener clave API FCM

Además, había que modificar un par de líneas de archivos de configuración de Android generado por Apache Cordova cuando genera la aplicación.

Para conectarlos a la API de Google FCM hacemos uso del código "send.php" [31], pero modificado posteriormente para meterle un par de mejoras. Este código básicamente usa CURL para conectarse a la API de FCM, configura una serie de parámetros, y finalmente mandar la notificación. Entre las modificaciones que hicimos fue conectarse a la base de datos para sacar el token del usuario al que le queremos mandar la notificación, y también recibir el mensaje que le queremos mandar por medio de un parámetro del POST, y adjuntarle ese mensaje en el cuerpo de la notificación.

A partir de la línea 36 podemos ver cómo inicializamos curl, que es una librería que nos va a permitir comunicarnos con el servicio de notificaciones push. Para ello tenemos que establecer una serie de parámetros, como la api_key del servicio de FCM (la cual la sacamos accediendo a la consola de firebase y en los ajustes generales), o el certificado que vamos a usar para mandar los mensajes. De la línea 20 a la 35 configuramos una variable($data_string), que es un json donde configuramos los parámetros de la notificación(el título, el mensaje que lleva dentro, el token del destino al que queremos mandarle la notificación). Luego esta variable la enviaremos por el curl en la línea 42.

En la línea 50 recibimos la respuesta y en la 51 hacemos "*echo"* de esta para mostrar el resultado del envío.

```php
<?php
    $id5 = $_POST['u1'];
    $texto = $_POST['mensaje'];
    $usuarioQueEnvia = $_POST['usuarioQueEnvia'];
    $con = mysqli_connect('localhost','root','','primerabbdd');
    if (!$con) {
        die('Could not connect: ' . mysqli_error($con));
    }
    $consulta = "SELECT * FROM tabla1 WHERE id = {$id5}" ;
    $result = mysqli_query($con,$consulta);
    $row = mysqli_fetch_array($result);
    $i=0;
    $tokenV = $row['tokens'];
    $url = 'https://fcm.googleapis.com/fcm/send';
    $fcm_api_key = 'AIzaSyBMd7bTPQuhaWJhqb2k6zz4szWIgdnOXwc';

    $title = $usuarioQueEnvia;
    $body = "test";
    $topic = "generalTopic";
    $data_string = '{
        "notification": {
            "title": "'.$title.'",
            "body": "'.$texto.'",
            "sound": "default",
            "click_action": "FCM_PLUGIN_ACTIVITY",
            "icon": "fcm_push_icon"
        },
        "data": {
            "param1": "value1",
            "param2": "value2"
        },
      "to": "'.$tokenV.'",
        "priority": "high",
        "restricted_package_name": ""
    }';
    $ch = curl_init();
    $certificate = "C:\wamp64\cacert.pem";
    curl_setopt($ch, CURLOPT_CAINFO, $certificate);
    curl_setopt($ch, CURLOPT_CAPATH, $certificate);
    curl_setopt( $ch, CURLOPT_URL, $url);
    curl_setopt( $ch, CURLOPT_POST, true);
    curl_setopt( $ch, CURLOPT_POSTFIELDS, $data_string);
    curl_setopt( $ch, CURLOPT_FOLLOWLOCATION, 1);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt( $ch, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/json',
        'Authorization: key='.$fcm_api_key
    ));
    curl_setopt( $ch, CURLOPT_RETURNTRANSFER, 1);
    $response = curl_exec( $ch );
    echo json_encode($response);
?>
```

5.5.2 FOTOS

Hemos instalado el plugin oficial de apache Cordova que nos permite acceder a las funcionalidades de la cámara del móvil: [32]

cordova plugin add cordova-plugin-camera

Esto lo usaremos para dos cosas, por un lado, para permitir al usuario cambiar su foto de perfil, o haciéndose una foto o accediendo a la librería de fotos del móvil. Esto lo podemos elegir cuando pulsamos sobre el perfil en editar foto, y nos aparece un mensaje pop- up donde nos da a elegir estas dos opciones. En ambos casos, guardaremos dicha imagen en la base de datos para que cuando el usuario vuelva a entrar en la app el sistema le muestre su foto de perfil.

También se ofrecerá la opción de mandar mensajes por el chat. Estas posteriormente deben ser guardadas en la BBDD para que el usuario las pueda volver a consultar. El plugin una vez que ha hecho una foto nos permite devolver dos cosas: o la url de donde esta guardado el nuestro móvil de manera temporal la foto o de devolvernos un String que representa la foto (en base64). Este es el que usaremos para guardarlo en nuestra BBDD, en formato BLOB. Cuando queramos ver la foto, lo único que tenemos que hacer es acceder a la BBDD, acceder a la foto por su identificador y asignar ese valor guardado en el atributo src de la etiqueta img de html de la siguiente manera:

```
<img id="imagen" src= “" + result width="85%">
```

Este plugin también lo vamos a usar para poder cambiar la foto de nuestro perfil. De manera predeterminada, cuando arrancamos por primera vez la aplicación nos aparece en la foto de perfil una foto de un icono. Una vez que nos registramos podemos cambiar la foto, pulsando en cambiar la foto de perfil. Seguidamente se abrirá la cámara de nuestro móvil para tomarnos una foto y establecerla como foto de perfil.

Por último, he de comentar que el plugin también nos da la opción de poder editar la foto una vez sacada, para si la quisiésemos recortar, girar… antes de subirla definitivamente.

A continuación, muestro el código JavaScript implementado para poder hacer la foto. En la línea 5 aparece la llamada a la cámara para poder hacer la foto. Cuando se haya hecho bien, se lanzará la función onSuccess que llevara como parámetro la imageData, que se la mandaremos a un archivo php a través de Ajax para que la guarde en la base de datos de las imágenes

```
1.  function hacerFoto()
2.  {
3.    var modal = document.getElementById("myModal");
4.    modal.style.display = "none";
5.    navigator.camera.getPicture(onSuccess, onFail, { quality: 50,  destinationType:
    Camera.DestinationType.DATA_URL });
6.  }
7.  function onSuccess(imageData)
8.  {
9.        var imageProfile = document.getElementById('idImagenPerfil');
10.       var nombreImagen = document.getElementById("idNombreImagen").value;
11.       $.ajax(
12.       {
13.         type: 'POST',
14.         url: "https://lenguajes.ddns.net:443/phpDAO.php",
15.         data :{"consultaBBDD" : "insertarImagenChat" ,"dataImagen" :
    imageData,"nombreImagen": nombreImagen},
16.         success: function(result)
17.         {
18.            document.getElementById("divResultado").innerHTML = result;
19.         },
20.        });
21. }
```

5.5.3 CALENDARIO

Se ha instalado un plugin que nos permite acceder a la funcionalidad del calendario del móvil para poder configurar un evento y añadirlo a nuestro calendario: [33]

"https://github.com/EddyVerbruggen/Calendar-PhoneGap-Plugin.git"

El plugin nos provee de muchísimas funcionalidades (creación de calendarios, creación y modificación previa de eventos), pero nosotros solo vamos a usar una, que es la de configurar previamente una serie de parámetros, para luego que el móvil nos abra la aplicación del calendario con la ventana previa de configuración final de un evento. La ventaja de este plugin es que no necesitamos conectarnos con la API de Google Calendar, con las dificultades que eso conlleva, si no que con este plugin nos redirige a la página de creación de un evento, el cual se va a guardar en nuestra cuenta de Google.

5.5.4 NOTIFICACIONES LOCALES

Para implementar la funcionalidad de Recordatorio en nuestro móvil, nos hemos descargado el plugin cordova-plugin-notification, descargado de Gitub: [34]

Como ya he dicho, este plugin nos permite configurar un recordatorio local en nuestro móvil para que nos avise en un determinado momento (a una hora o fecha concreta). El plugin también ofrece diversas posibilidades a la hora de personalizar la notificación que nos aparece en la barra de notificaciones. Por ejemplo, aparte de elegir el mensaje que aparece en la notificación, también podemos elegir qué foto aparece en pequeño, o si la notificación nos ofrece responder con un sí o un no desde la propia notificación, o también poder escribir una respuesta dentro de la propia notificación, muy similar a cómo se puede hacer en WhatsApp.

Otra gran ventaja que nos ofrece este plugin es que la aplicación no tiene que estar abierta, ni en segundo plano para que la notificación del recordatorio se lance, trabajando de esta manera de manera análoga a como lo haría cualquier aplicación de recordatorios.

Cuando el usuario haya quedado con alguien, podrá configurar este recordatorio, apareciéndole un mensaje pop-up donde puede configurar la fecha y el mensaje que quiere que le aparezca. Por poner un ejemplo sencillo de cómo se configuraría una notificación, con el siguiente código configuramos una notificación local para dentro de 1 segundo:

```
let inOneMin = new Date();
inOneMin.setMinutes(inOneMin.getMinutes() + 0);
let id = new Date().getMilliseconds();
let var = { id: id,  title: “a”,  text: "Has quedado ",  at: inOneMin };
cordova.plugins.notification.local.schedule(var);
```

Como podemos observar, con la última línea lo que hacemos es establecer la notificación, y con la variable noteOptions, la configuramos, eligiendo su id, su título, el texto que contendrá la notificación y por último cuando será lanzada(con el atributo at), que en este caso hemos establecido en la variable inOneMin que sea lanzado en 1 segundo después

Capítulo 6. ANÁLISIS DE RESULTADOS

La aplicación ha cumplido todos los requisitos que se habían propuesto, los cuales estaban resumidos en los casos de uso de la aplicación. Debido a la crisis de la pandemia de Covid-19, el objetivo de poder meter la aplicación dentro de los servidores de Comillas para que todos los mundos pueda acceder a la aplicación desde la app oficial no se ha podido realizar, debido a la imposibilidad de ir ahora a instalarlo en los servidores.

Uno de los puntos más difíciles del trabajo fue conseguir que funcionasen las notificaciones push, las cuales como ya he explicado en el capítulo dedicado a ellas, se hizo necesario configurar el uso de un servicio de terceros, en este caso de firebase cloud messaging, de Google. A pesar de la dificultad a la hora de configurar el acceso al servicio de FCM, se consiguió integrar correctamente en la aplicación, pudiendo recibir notificaciones tanto si tienes la app encendida, como apagada, como si tienes el móvil con la pantalla apagada.

En estas imágenes podemos ver como tenemos la aplicación corriendo en dos dispositivos Android distintos, registrados como usuarios distintos(Miriam y Pedro). Miriam le manda una notificación a Pedro, teniendo este la aplicación cerrada.

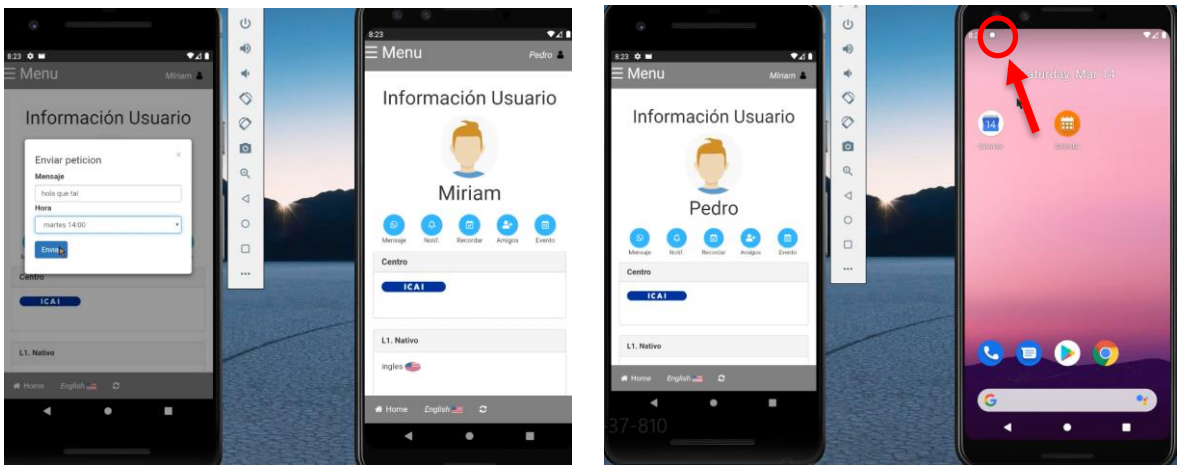


Ilustración 47 Ejemplo de envio de la notificación

Otro desafío fue configurar la página web para poder hacerla pública y accesible para todo el mundo, sin tener que depender de un servicio de hosting, el cual además habría supuesto un incremento en el coste económico del proyecto. Para ello, como explico en el anexo II, hubo que suscribirse a un servicio gratuito de DNS y configurar una serie de parámetros en mi router, como NAT, para poder hacer pública la página web. Además, se configuró https en la página, para proveerla de seguridad. Sin embargo, se optó por auto firmar la clave de manera que la página usa https, pero los navegadores no la reconocen como segura.

El otro gran objetivo de la aplicación era poder ver en tiempo real todos los usuarios, mostrando su información más relevante, como los idiomas de los que son nativos o sus preferencias. También se puede chatear con ellos y mandarles fotos. Tanto cuando navegas por la app viendo a los otros usuarios, como cuando envías mensajes, la actualización de la información y el envío de los mensajes se hacen de manera instantánea, aunque en el caso del actualizar la información de los usuarios se hace refrescando la página desplazándola hacia abajo.

Con respecto al análisis de la descripción de los gustos y aficiones de los usuarios mediante Machine Learning y análisis de lenguaje natural, podemos decir que los dos sirven satisfactoriamente para clasificar el texto introducido. Se han probado diversos tipos de texto para probar que tal clasificaba el modelo, y en la mayoría de los casos el sistema ha clasificado bien el texto introducido(usando además un dataset de testeo distinto al usado para entrenar el modelo).

A la hora de analizar la arquitectura de la aplicación, hay que tener en cuenta que debido a que la aplicación se desarrolla en un entorno web, y luego a través de la herramienta Apache Cordova se pasa a Android o IOS, hay muchas características y funcionalidades que cuando se pasa de uno a otro dejan de funcionar. No solo son las funciones relacionadas con los plugins(como la cámara, el calendario…) sino que a la hora de cargar la información del servidor en el HTML hay muchas características que en el navegador funcionan y luego en el móvil no. Por ejemplo, en el navegador puedes ver un archivo php como si fuese un HTML, pero con la ventaja de ejecutar código servidor como por ejemplo para acceder a la

base de datos, y así solo tener un archivo y no dos. En cambio, en el móvil no te permite ver un archivo php como si fuese un HTML, sino que debes cargar la información del servidor a través del Ajax en el HTML. Por tanto, para usar el mismo código para las dos plataformas(la web, y el móvil) siempre se optó por la versión que funcionaba en los dos, que es la de tener de manera separada los HTML y los php, y usar Ajax para conectar ambos. Esa es la razón de que como resultado, la arquitectura de la aplicación haya quedado con una división clara entre la parte visual(los HTML) y la parte del servidor(los archivos PHP).

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

La aplicación ha cumplido todos los objetivos propuestos al principio del proyecto, tanto en funcionalidad como en implantación en las distintas plataformas en las que se tenía previsto que funcionase la app(Android, IOS y web). El objetivo de integrarla en los servidores de Comillas no se ha podido realizar debido a la imposibilidad de moverse derivada por la crisis del Covid-19.

En un futuro esta aplicación se podría usar también para que los alumnos puedan organizar clases particulares entre ellos y así puedan ayudarse en los estudios, sobre todo en los primeros años de carrera, que son los que suelen costar más y donde el apoyo entre compañeros suele ser más determinante a la hora de ayudarse a entender la materia.

Se podrían desarrollar chat grupales al estilo de WhatsApp, de manera que para poder enviar el mismo mensaje a varios contactos no haya que ir uno por uno, sino que puedas crear un grupo con todos ellos y todos los mensajes que se envíen por dicho grupo los vean todos los participantes.

Con respecto a la parte de Machine Learning, esta te permite ir un paso más allá en el tratamiento de la información que introduce el usuario, permitiendo que cuando más se use la aplicación, más se entrene el modelo, y cada vez sea más preciso a la hora de clasificar los gustos y aficiones de los usuarios, dando mejores sugerencias a estos, y en consecuencia dotando de más “inteligencia” al sistema.

Debido a la pandemia de Covid-19, uno de los objetivos del proyecto que era poder meter la aplicación en los servidores de Comillas no se pudo realizar. Por tanto, como continuación de este trabajo estarían una serie de trabajos:

- Meter la aplicación en los servidores de Comillas, es decir hacer un hosting de la aplicación en dichos servidores, en vez de en el servidor que tengo instalado ahora en mi ordenador.

- Poder introducir la aplicación dentro de la aplicación oficial de Comillas, de manera que los estudiantes no tengan que descargarse dos aplicaciones distintas. De esta manera, los usuarios que quisiesen usar la aplicación se meterían en la app de Comillas, y ahí habría un link para usar la aplicación. Otra ventaja sería que al usar la app ya estarían registrados con sus credenciales de la Universidad, ya que es necesario para poder usar la app oficial de Comillas. De esta manera no tendrían que crear otro usuario y contraseña para esta aplicación, y tampoco se haría necesario tratar dichos datos, ya que del servicio de autenticación ya se encargaría la universidad, el cual es mucho más robusto y seguro que el que yo pueda implantar en mi base de datos de usuarios. Otro de los datos que servirían para mejora la experiencia de uso de la aplicación sería poder acceder a la foto de perfil que tiene guardada la Universidad de cada alumno. De esta manera, en la pantalla principal en vez de ver imágenes de pega(como se ve ahora) de cada alumno se pueda ver su imagen de verdad. Sin embargo, no hay que olvidar que una de las funcionalidades de esta aplicación es permitir que los usuarios puedan cambiar en cualquier momento su imagen de perfil y esta se quede guardada en la base de datos de la universidad para que se pueda seguir viendo cada vez que se mete en la app.

- Otra tecnología que se podría implantar después de alojar la aplicación en los servidores sería configurar socket para recibir los mensajes del chat. Los socket son una combinación de ip y puerto, de manera que se puede asignar un proceso para que siempre este escuchando en ese puerto. Esta configuración mejora sustancialmente el rendimiento, ya que no es necesario estar ejecutando cada segundo una lectura de la base de datos, sino que solo se lee cuando llega un mensaje.

- Si se cumpliese el objetivo del punto anterior, también habría que adaptar el aspecto visual de la app al mismo que el de la aplicación de Comillas, de manera que no dé la sensación a los usuarios que están usando dos aplicación distintas(aunque una esté dentro de la otra) sino que no ha diferencia visual entre una y otra.

Capítulo 8. BIBLIOGRAFÍA

[1] A. Cordova, «Apache Cordova Wikipedia,» 25 Octuble 2019. [En línea]. Available: https://es.wikipedia.org/wiki/Apache_Cordova. [Último acceso: 18 Enero 2020].

[2] vexsoluciones, [En línea]. Available: https://www.vexsoluciones.com/apps-moviles/apps-nativas-vs-hibridas/. [Último acceso: 29 Mayo 2020].

[3] c. MVP, «Qué son los plugins en Apache Cordova y para qué sirven,» [En línea]. Available: https://www.campusmvp.es/recursos/post/Que-son-los-plugins-en-Apache-Cordova-y-para-que-sirven.aspx. [Último acceso: 22 Mayo 2020].

[4] Wikipedia, «cURL,» [En línea]. Available: https://es.wikipedia.org/wiki/CURL. [Último acceso: 05 Mayo 2020].

[5] Wikipedia, «Composer,» [En línea]. Available: https://es.wikipedia.org/wiki/Composer. [Último acceso: 05 Junio 2020].

[6] E. Pais, «Cinco ‘apps’ para hacer intercambio de idiomas,» 29 Junio 2017. [En línea]. Available: https://elpais.com/tecnologia/2017/06/29/actualidad/1498752468_039580.html. [Último acceso: 15 Octubre 2019].

[7] Tandem, «tandem.net,» [En línea]. Available: https://www.tandem.net/es. [Último acceso: 18 Octubre 2019].

[8] HelloTalk, «helloTalk,» [En línea]. Available: https://www.hellotalk.com. [Último acceso: 15 Octubre 2019].

[9] Speaky, «Speaky,» [En línea]. Available: https://www.speaky.com/es/. [Último acceso: 20 Octubre 2019].

[10] Google, «Precio de Cloud Translation,» [En línea]. Available: https://cloud.google.com/translate/pricing?hl=es. [Último acceso: 28 Mayo 2020].

[11] Google, «Planes de precios Firebase,» [En línea]. Available: https://firebase.google.com/pricing?hl=es-419#blaze-calculator. [Último acceso: 28 Mayo 2020].

[12] Google, «Precios Cloud Natural Language,» [En línea]. Available: https://cloud.google.com/natural-language/pricing?hl=es-419. [Último acceso: 29 Mayo 2020].

[13] php, «¿Qué es PHP?,» [En línea]. Available: https://www.php.net/manual/es/intro-whatis.php. [Último acceso: 31 Mayo 2020].

[14] O. Jose, «ARQUITECTURA DE LAS APLICACIONES WEB,» [En línea]. Available: https://images.app.goo.gl/GiryyAYGfgrY9TLz6. [Último acceso: 06 Junio 2020].

[15] J. Barecki, «Using Google Translate API with PHP,» sitepoint, 30 Octubre 2013. [En línea]. Available: https://www.sitepoint.com/using-google-translate-api-php/. [Último acceso: 1 Junio 2020].

[16] Google, «Natural Language,» [En línea]. Available: https://cloud.google.com/natural-language?hl=es. [Último acceso: 1 Junio 2020].

[17] Google, «Natural Language Client Libraries,» [En línea]. Available: https://cloud.google.com/natural-language/docs/reference/libraries. [Último acceso: 12 Junio 2020].

[18] Wikipedia, «Objeto de acceso a datos,» [En línea]. Available: https://es.wikipedia.org/wiki/Objeto_de_acceso_a_datos. [Último acceso: 30 Mayo 2020].

[19] Wikipedia, «AJAX,» [En línea]. Available: https://es.wikipedia.org/wiki/AJAX. [Último acceso: 31 Mayo 2020].

[20] «Swipper.js,» [En línea]. Available: https://swiperjs.com/api/. [Último acceso: 10 02 2020].

[21] ErlendEllingsen, «Pull to reload,» [En línea]. Available: https://www.cssscript.com/pull-refresh-library-mobile-desktop-pull-reload-js/. [Último acceso: 24 Mayo 2020].

[22] w3schools, «W3.CSS Modal,» [En línea]. Available: https://www.w3schools.com/w3css/w3css_modal.asp. [Último acceso: 26 Mayo 2020].

[23] w3schools.com, «How TO - Snackbar / Toast,» [En línea]. Available: https://www.w3schools.com/howto/howto_js_snackbar.asp. [Último acceso: 26 Mayo 2020].

[24] w3school, «How TO - Side Navigation,» [En línea]. Available: https://www.w3schools.com/howto/howto_js_sidenav.asp. [Último acceso: 29 Mayo 2020].

[25] w3schools, «How TO - Full Page Tabs,» [En línea]. Available: https://www.w3schools.com/howto/howto_js_full_page_tabs.asp. [Último acceso: 2 Junio 2020].

[26] w3school, «How TO - Register Form,» [En línea]. Available: https://www.w3schools.com/howto/howto_css_register_form.asp. [Último acceso: 2 Junio 2020].

[27] A. Katharopoulos, «PHP NlpTools,» [En línea]. Available: https://github.com/angeloskath/php-nlp-tools. [Último acceso: 07 Junio 2020].

[28] A. Katharopoulos, «Getting started with NlpTools,» [En línea]. Available: http://php-nlp-tools.com/documentation/index.html. [Último acceso: 07 Junio 2020].

[29] render2web, «render2web.com,» 25 Abril 2017. [En línea]. Available: http://render2web.com/como-crear-un-chat-con-html5-css3-php-mysql-y-ajax/. [Último acceso: 11 Junio 2020].

[30] andrehtissot, «cordova-plugin-fcm-with-dependecy-updated),» [En línea]. Available: (https://www.npmjs.com/package/cordova-plugin-fcm-with-dependecy-updated).. [Último acceso: 12 Junio 2020].

[31] A. Mudianto, «Push Notification Cordova, FCM And PHP,» techgalery.com, 05 December 2018. [En línea]. Available: https://www.techgalery.com/2018/12/push-notification-cordova-fcm-and-php.html. [Último acceso: 29 Mayo 2020].

[32] Apache Cordova, «cordova-plugin-camara,» [En línea]. Available: https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-camera/. [Último acceso: 07 Junio 2020].

[33] EddyVerbruggen, «PhoneGap Calendar plugin,» [En línea]. Available: https://github.com/EddyVerbruggen/Calendar-PhoneGap-Plugin. [Último acceso: 30 Mayo 2020].

[34] S. Katzer, «cordova-plugin-local-notifications,» [En línea]. Available: https://github.com/katzer/cordova-plugin-local-notifications. [Último acceso: 30 Mayo 2020].

[35] J. Carles, «Encontrar servidor con DNS dinamico,» 21 Agosto 2013. [En línea]. Available: https://geekland.eu/encontrar-servidor-con-dns-dinamico/. [Último acceso: 06 Junio 2020].

[36] «¿Cómo habilitar la URL https (localhost) en el servidor WAMP (v2.5)?,» [En línea]. Available: https://www.rephp.com/como-habilitar-la-url-https-localhost-en-el-servidor-wamp-v2-5.html. [Último acceso: 07 Junio 2020].

[37] BERKINALEX, «Como instalar Apache Cordova en Windows 10,» 22 Enero 2019. [En línea]. Available: https://berkinalex.com/como-instalar-apache-cordova-en-windows-10. [Último acceso: 1 Junio 2020].

[38] Wikipedia, «Node.js,» [En línea]. Available: https://es.wikipedia.org/wiki/Node.js. [Último acceso: 29 Mayo 2020].

[39] Java, «Java JDK,» [En línea]. Available: (https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html. [Último acceso: 12 Junio 2020].

[40] «Composer Wikipedia,» 11 Abril 2020. [En línea]. Available: https://es.wikipedia.org/wiki/Composer.

[41] «Procesamiento de lenguaje natural Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Procesamiento_de_lenguajes_naturales. [Último acceso: 11 Abril 2020].

[42] Apache Cordova, «Apache Cordova,» [En línea]. Available: https://cordova.apache.org/. [Último acceso: 06 Junio 2020].

[43] RedHat, «Qué son las API y para qué sirven,» [En línea]. Available: https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces. [Último acceso: 06 Junio 2020].

[44] Google, «Natural Language Client Libraries,» [En línea]. Available: https://cloud.google.com/natural-language/docs/reference/libraries. [Último acceso: 12 Junio 2020].

ANEXO I MANUAL DE INSTALACIÓN

8.1 INSTALACIÓN ENTORNO DE DESARROLLO/ EDITOR DE TEXTO SUBLIME TEXT

El editor de texto elegido será sublime text3. Esto es debido a que presenta un entorno muy sencillo, pero a la vez robusto para escribir código, y además es muy ligero, siendo esta su principal ventaja frente a otros como pueden ser Eclipse o VisualStudio.

8.2 INSTALACIÓN SIMULADORES

8.2.1 ANDROID STUDIO

Para instalar Android Studio nos vamos a la página oficial(https://developer.android.com/studio) y nos descargamos la última versión.

8.2.2 XCODE

Para descargarnos Xcode debemos seguir muchos menos pasos que para Android Studio. Para empezar, nos iremos al Apple Store de nuestro Mac y nos descargaremos la aplicación Xcode, la cual es gratuita. Una vez instalada, la arrancamos y nos da a elegir entre si crear un nuevo proyecto o abrir uno. Siempre elegiremos esta segunda opción ya que siempre tendremos ya creado el proyecto por Apache Cordova. Lo único que debemos configurar es el simulador donde querremos correr nuestra aplicación. Elegimos un iPhone 8 y no el X porque el X pesa mucho más y es mucho más lento al arrancar en comparación con el 8.

8.3 INSTALACIÓN SERVIDOR: WAMP SERVER

Para instalar el servidor WAMP server seguimos los siguientes pasos:

1. Descargar WAMP server desde la página oficial
2. Ejecutar el archivo que se descarga. Después nos irán poniendo una serie de ventanas a las que tendremos que ir dando a aceptar.

8.4 CONFIGURACIÓN PAGINA WEB

8.4.1 VHOST

Una vez configurado nuestro servidor apache en local en nuestro portátil, lo primero que debemos hacer es configurar una VHost, para decirle que esa carpeta donde tenemos el proyecto se llama de una determinada manera.

8.4.2 DOMINIO

Después debemos registrarnos en un servicio DNS, para poder asignarle un nombre a nuestra página web, para evitarnos poner cada vez que entramos la ip publica de nuestro router. Existen muchos servicios online que nos ofrecen darnos de alta en un DNS, pero el elegido será noIP por tres razones. La primera, porque es una de las opciones más elegidas por todos los usuarios, la segunda, porque después deberemos configurar nuestro router para que sepa de esta conversión ip en nombre DNS, y el único servicio reconocido por nuestro router (Movistar Mitra) es el de NoIp. La tercera es porque es un servicio totalmente gratuito. La otra razón de instalar el servicio en nuestro router es que este cambia su ip cada cierto tiempo (lo que se llama ip dinámica). De manera que cuando lo cambia se lo debe notificar a NoIp para actualizar el servidor DNS.

Dentro de la página noIp, nos damos de alta y luego elegimos el nombre que queremos que tenga nuestra aplicación. Por último configuramos qué tipo de router tenemos, y nos descargamos un programa que será el encargado de saber qué ip tiene nuestro router.

8.4.3 CONFIGURACIÓN EN EL ROUTER

8.4.3.1 Dynamic DNS

Debemos acceder a la configuración de nuestro routers para configurar el DNS. Para ello ingresamos en el navegador la dirección 192.168.1.1, que es la ip de nuestro router, y nos abre la configuración remota de nuestro Gateway(primero nos pide una contraseña que viene en la parte de atrás de nuestro router). Después accedemos a configuración avanzada. Una

vez ahí, pinchamos sobre Advanced SetUp, luego en DNS y por último en Dynamic DNS. [35]

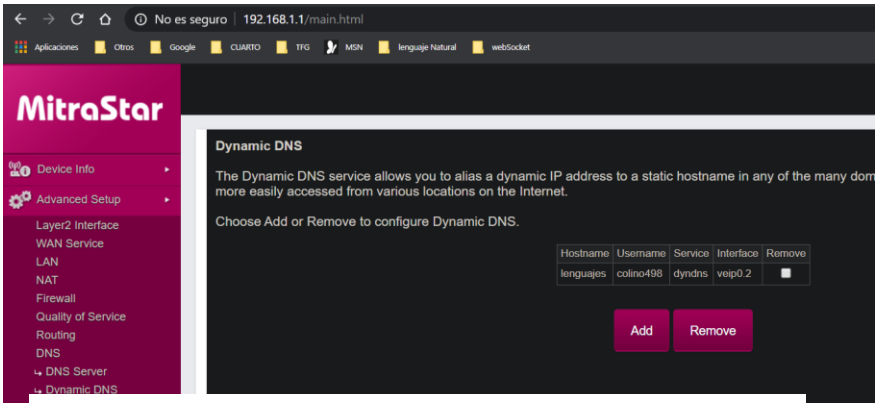


Ilustración 48 Configuración Dynamic DNS

8.4.3.2 NAT

Aparte de configurar el Dynamic DNS, debemos también establecer una configuración más, la relacionada con NAT.NAT es un protocolo que nos permite traducir una ip pública en una ip privada, de manera que cuando venga de internet una petición hacia nuestra página web nuestro router redirija petición a la ip de nuestro PC, que es donde está escuchando nuestro router. En la captura que aparece a continuación podemos ver como hemos añadido 4 reglas de NAT, una para TCP y otra para UDP (aunque esta no la vamos a usar ya que todo el tráfico https corre sobre TCP/IP), y luego dos para el puerto 80(http) y otras dos para el

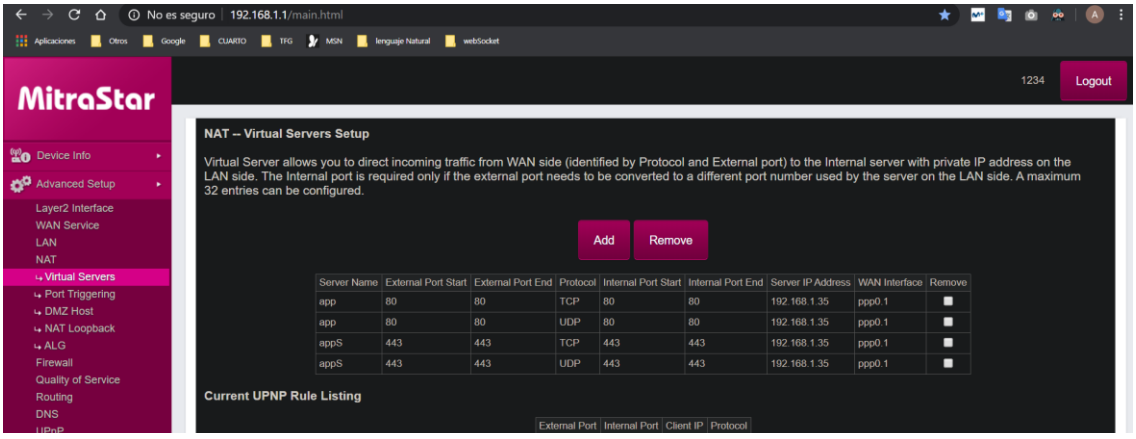


Ilustración 49 Configuración de NAT en el router

puerto 443(https). Esto se debe a que cuando lo configuramos por primera vez solo pusimos http, pero después añadimos un certificado para poder usar el tráfico cifrado de https.

8.4.4 CONFIGURACIÓN DE NUESTRO SERVIDOR

1. Debemos dirigirnos al archivo \wamp\bin\apache\Apache2.2.11\conf\httpd.conf y cambiar la línea #Listen 12.34.56.78:80 por Listen 192.168.1.35:443, que es la ip y el puerto donde va a estar escuchando nuestro servidor

2. En la línea 249 debemos poner el siguiente código;

```
<Directory />
      # AllowOverride none
      # Require all denied
      AllowOverride All
      Order allow,deny
      Allow from all
</Directory>
```

3. Debemos añadir una entrada en el archivo vhost que es la siguiente

```
<VirtualHost *:80>
     ServerAdmin webmaster@camara
     DocumentRoot "C:/wamp64/www/camara/www"
     <Directory "C:/wamp64/www/camara/www">
       AllowOverride All
       Options +Indexes +Includes +FollowSymLinks +MultiViews
       #Require local
       Require all granted
     </Directory>
     ServerName camara
</VirtualHost>
```

Debemos introducir en nuestro archivo host (ubicado en C:\Windows\System32\drivers\etc) lo siguiente

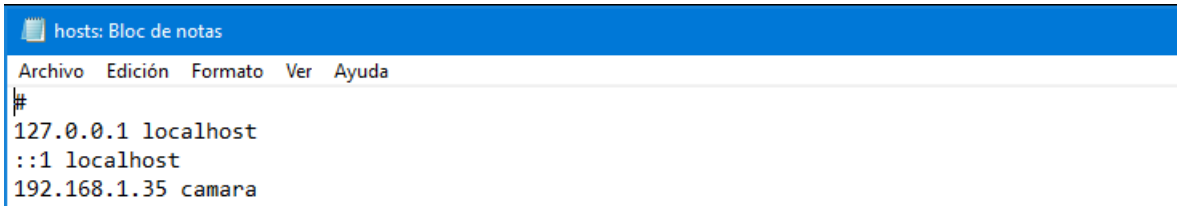


Ilustración 50 Archivo host del ordenador

8.4.4.1 Configuración en el firewall de nuestro PC

Debemos añadir una nueva regla de entrada y de salida en nuestro firewall para que deje pasar todo el tráfico TCP/UDP que pase por el puerto 80 y 443.

8.4.4.2 Configuración https

Para configurar HTTPS debemos por un lado descargarnos un certificado(que en nuestro caso será auto firmado ya que si no tendríamos que pagar dinero por uno oficial) y editar una serie de líneas del archivo https-ssl.conf (ubicado en C:\wamp64\bin\apache\apache2.4.39\conf\extra) con lo siguiente: [36]

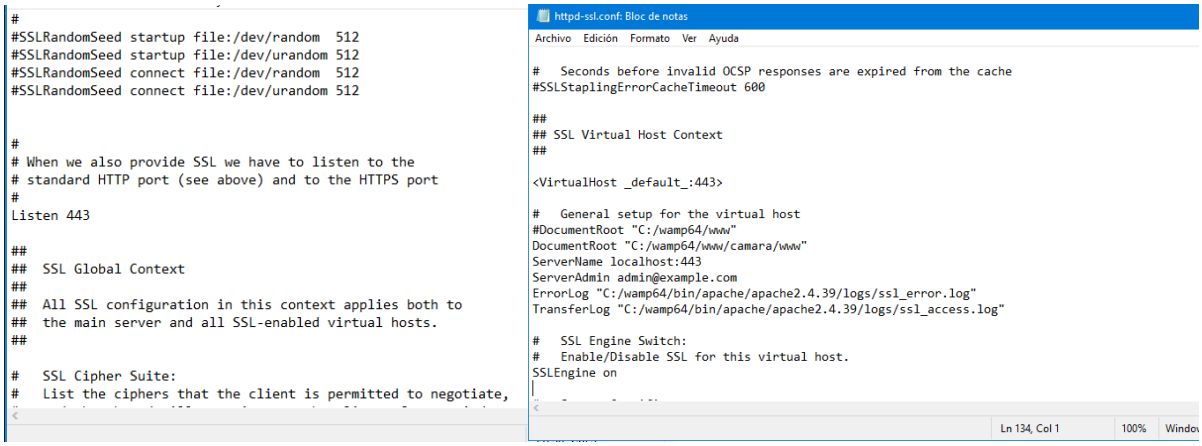


Ilustración 51 Configuración de https

Uno de los pasos del tutorial es descargarse open.sl, una librería que nos permite generarnos una clave pública y privada con la que firmaremos nuestro certificado(por eso es auto firmado y los navegadores lo catalogan como no seguro). Este certificado también es importante ya que es necesario para poder mandar las notificaciones push a través del servicio FCM

8.5 INSTALACIÓN APACHE CORDOVA [37]

Nos descargamos la versión de 64 bits de Node.js desde la página oficial (http://nodejs.org/download/). "Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello)

basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google" [38]

Una vez instalado Node.js, necesitamos instalarnos el JDK de Java a través de su página oficial [39]. A continuación, añadimos a las variables de entorno JAVA_HOME, con la ruta donde hemos instalado Java, al igual que con ANDROID_HOME.

Abrimos la consola de comandos de node.js en modo administrador y ejecutamos el comando: `npm install -g cordova.`

8.6 CREACIÓN DE UN PROYECTO APACHE CORDOVA

Para crear un proyecto de Cordova, primero nos posicionamos en la carpeta donde queremos tenerlo. En nuestro caso, va a ir dentro de la carpeta www de nuestro servidor Wamp, para poder colocar dentro nuestros archivos php. Por tanto, para crear el proyecto ejecutamos este comando:

```
cordova create MiPrimeraApp com.mi.MiprimeraApp MiPrimeraApp
```

Nos metemos dentro del directorio miPrimeraApp y instalamos la plataforma en la que vamos a instalar la app:

```
cd MiPrimeraApp
cordova platform add android
```

Dentro de la carpeta "miPrimeraApp" encontramos una carpeta llamada www. En ella vamos a guardar todos los fuentes de nuestra aplicación. Por último, para construir nuestra app ejecutamos el comando:

```
cordova build Android
```

Por último, ejecutamos este comando para que lance la app en nuestro simulador (para ello debemos haberlo arrancado previamente). Después de ejecutar este comando en la ventana de comando nos aparecerá la ruta donde se ha generado el APK, que es la aplicación que luego podemos instalar en nuestro móvil o subirla a Play Store: `cordova emulate Android`

ANEXO II OBJETIVOS DE DESARROLLO SOSTENIBLE

Objetivo 4: Garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante toda la vida para todos.

Con esta aplicación se favorece el aprendizaje de idiomas, y el acercamiento entre personas de distintos países, ya que el proyecto está enfocado para los estudiantes incoming que vienen a nuestra universidad y que puedan tener una herramienta que les permita el poder tener mayor contacto con la gente de España que estudia en la Universidad. Con ello se pretende solventar uno de los mayores problemas que le surgen a una persona cuando se va a estudiar a otro país: el idioma. Aprendiendo y practicando el idioma del país en el que estudias, te ayudará a integrarte mejor en ese país, y entender mejor las clases que tienes.

Vivimos en una sociedad totalmente involucrada con las redes sociales y las nuevas tecnologías, que nos permiten, entre otras muchas cosas, estar en contacto con la gente que tenemos más lejos. Pero ¿y si usamos los recursos que nos proveen internet, las nuevas tecnologías y las redes sociales para promover encuentros cara a cara con otras personas? Y además con el propósito principal de ayudar a las personas a aprender o perfeccionar nuevos idiomas que están aprendiendo. Muchas veces ponemos como excusa que para aprender un idioma de verdad hay que irse fuera de España a vivir o a pasar un año o un verano a ese país para aprenderlo de verdad, pero esto muchas veces no está al alcance de la mayoría de las personas. Por ello, si aprovechamos a esos extranjeros que ya están estudiando en nuestra Universidad para practicar con ellos su idioma y nosotros enseñarles el nuestro, también podremos echarles una mano para integrarse más, ya que si no sabes hablar bien el idioma del país en el que vives, difícilmente vas a poder relacionarte bien con los demás.

Además, la aplicación al estar enfocada a todos los miembros de la universidad de Comillas es totalmente inclusiva, ya que no discrimina ni impide que alguien la use por cualquier tipo de razón, como por ejemplo la sede donde estudies, la carrera que estudies, o si eres extranjero o vives en España. La pueden usar tanto profesores como alumnos, da igual su edad. Además, es gratuita, por tanto, la cuestión económica tampoco es un problema a la hora de poder usarla. Se podría decir que esto es uno de sus puntos más fuertes, ya que normalmente cuando queremos practicar idiomas nos debemos apuntar a una academia o contratar a una persona nativa con la que tener conversación, y esto suele costar mucho dinero, por lo que sólo está al alcance de las personas que se lo pueden gastar, haciendo que el aprender un idioma se convierta en justo lo contrario a inclusivo.

Estudiar asignaturas de tu carrera en una universidad extranjera, implica una dificultad añadida al estar residiendo en un país extranjero durante meses. Ahora más que nunca, podremos poner al alcance de alumnos extranjeros, una herramienta fundamental para poder sacar el máximo rendimiento a la estancia en nuestro país. Educación y cultura son fundamentales para poder conocer un país distinto al nuestro y hoy en día el estrechar lazos culturales con personas de distintos países enriquece nuestra sociedad. El mundo actual está demandando estudiantes que puedan, una vez acabado sus estudios, realizar su vida laboral en cualquier país del mundo. Profesorado y alumnado estarán conectados a través de esta aplicación y podremos conseguir que el idioma no sea un impedimento para que nuestra universidad ofrezca a alumnos extranjeros un valor añadido en su educación.

El alumno logrará tener mejor curriculum al obtener mejor preparación académica, lo cual le abrirá las puertas a optar a mejores empleos internacionales dada la mayor competitividad que hoy en día existe en el mercado laboral, tanto nacional como internacional, abriéndole por tanto las puertas a más oportunidades de aprender. En resumen, invertir en aprender idiomas te permite prepararte para en un futuro seguir aprendiendo sin que ya el idioma sea un impedimento.