

Análisis, diseño e implementación de una aplicación para dispositivos móviles multiplataforma para la comunicación entre centros educativos y padres de alumnos



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática



ALUMNO

JOEL DAVID DELGADO PERDOMO

TUTOR

DR. ALEXIS QUESADA ARENCIBIA

GRADO EN INGENIERÍA INFORMÁTICA (INGENIERÍA DEL SOFTWARE)
Julio 2017 - Las Palmas de Gran canaria

Trabajo de Fin de Grado en Ingeniería Informática (intensificación en Ingeniería del Software) de la Universidad de Las Palmas de Gran Canaria presentado por el alumno:

Joel David Delgado Perdomo

Título del proyecto

Análisis, diseño e implementación de una aplicación para dispositivos móviles multiplataforma para la comunicación entre centros educativos y padres de alumnos

Tutor

Dr. Alexis Quesada Arencibia

Agradecimientos

A mi tutor Alexis Quesada Arencibia, por apoyarnos y recibirnos con los brazos desde el primer día, haciendo lo imposible por atendernos.

A mi compañero y amigo Adrián Louro Alonso, por haberme acompañado en este proyecto y haberme ayudado tanto durante la carrera.

Al grupo Monóxido Ferroso, sin vosotros no estaría aquí.

A mi familia, novia y amigos, por apoyarme siempre desde el primer crédito hasta el último.

Resumen

En la actualidad la tecnología está presente en todas partes. Pero existiendo alternativas tecnológicas aún seguimos haciendo uso de herramientas tan tradicionales, como el papel.

Su fabricación conlleva una serie de desventajas hacia el medioambiente, que puede ser paliado haciendo uso de alternativas.

Uno de los usos más populares consiste en el papeleo escolar. Millones de hojas son utilizadas cada año para realizar encuestas, autorizaciones, o circulares escolares, con uso único y desechadas después de cumplir su objetivo, la mayoría de las veces sin reciclarse.

Como alternativa, proponemos un sistema que posibilita a centros y padres para comunicarse de manera rápida, sencilla, directa y económica, permitiendo además que los padres puedan las firmar autorizaciones de sus hijos.

Abstract

Nowadays, technology is present everywhere. But with technologic alternatives we still use some of the most traditional tools, like paper.

Its making carry a series of disadvantages to the enviroment, that can be mitigated using alternatives.

One of the most populars uses consists in scholar papelwork. Millions of paper sheets are used every year to make scholar polls, authorizations, or circulars, with a single-time use and they are discarded after being used, the most of them without recycling.

As an alternative, we purpose a system that enables the communication between centres and parents in a fast, simple, direct and economic way, allowing as well parents to sign authorizations for their children.

Índice

Introducción	1
Estructura del documento	2
1. Estado actual y objetivos iniciales	4
1.1 Estado actual	4
1.1.1 Alternativas Existentes	4
1.1.2 Conclusiones.....	5
1.2 Objetivos	6
2. Competencias específicas cubiertas.....	7
2.1 Comunes a la Ingeniería Informática	7
CII01.....	7
CII03.....	7
CII08.....	7
CII012.....	8
CII013.....	8
CII016.....	8
CII017.....	8
CII018.....	9
2.2 Trabajo Fin de Grado.....	9
TFG01	9
2.3 Ingeniería del Software (IS)	9
IS01	9
IS02.....	10
IS03	10
IS04.....	10
2.4 Tecnologías de Información (TI).....	11
TI03.....	11
TI06.....	11
3. Aportaciones	12
3.1 Entorno socioeconómico.....	12
3.2 Personal.....	13
4. Normativa y legislación	14
4.1 Licencias de Software	14
4.1.1 GNU GPL.....	14
4.1.2 Licencia Apache	15

4.1.3 Licencia PHP	15
4.1.4 Licencia MIT.....	16
4.1.5 Licencia para educación de JetBrains.....	16
4.1.6 Licencia de Justinmind	17
4.1.7 Licencia para educación de Office 365.....	17
4.2 Seguridad de los Datos.....	18
5. Metodología de trabajo y planificación del proyecto	20
Metodología de trabajo	20
Ventajas.....	21
Inconvenientes.....	22
Conclusiones.....	23
5.2 Planificación del proyecto	23
5.2.1 Planificación inicial	23
5.2.2 Ajustes de la planificación	25
6. Tecnologías y herramientas utilizadas	26
6.1 Tecnologías.....	26
6.2 Herramientas.....	27
6.3 Servidores.....	28
7. Análisis.....	29
7.1 Actores	29
7.1.1 Administrador	29
7.1.2 Padre/Madre	30
7.2 Requisitos	30
7.2.1 Casos de uso.....	31
7.2.2 Especificación de casos de uso.....	34
8. Diseño.....	38
8.1 Diseño de la arquitectura del sistema.....	38
8.1.1 Frontend.....	40
8.1.2 Backend	40
8.2 Diseño de la Base de Datos	41
8.2.1 Contenido de las tablas de la Base de Datos.....	43
8.2.2 Almacenamiento de archivos adjuntos.....	44
8.3 Diseño Arquitectónico.....	45
8.4 Diseño de la interfaz de usuario.....	45
Principio 1.....	46
Principio 2.....	46

Principio 3.....	46
Principio 4.....	47
Principio 5.....	47
Principio 6.....	48
Principio 7.....	48
Principio 8.....	48
Principio 9.....	49
9. Desarrollo.....	50
9.1 Ionic.....	50
9.2 Estructura de la aplicación.....	51
9.3 API Restful.....	53
9.3.1 Friends of Symfony.....	53
9.3.2 Configuración de la API.....	53
9.3.3 Enrutamiento de la API.....	55
10. Pruebas.....	57
10.1 Pruebas de usabilidad.....	57
10.2 Pruebas de interacción.....	58
11. Resultados, conclusiones y trabajo futuro.....	59
11.1 Resultados y conclusiones.....	59
11.2 Trabajo futuro.....	60
12. Bibliografía.....	62
Anexo I: Manual de usuario.....	65
Introducción.....	65
Inicio de sesión – Registro.....	65
Circulares.....	70
Autorizaciones.....	73
Encuestas.....	76
Mi Perfil.....	77
Mis datos.....	78
Mis hijos.....	80
Mis centros.....	82
Anexo II: Manual de instalación.....	83
API.....	83
Aplicación móvil.....	83
Anexo III: Script para la base de datos.....	85

Índice de ilustraciones

Ilustración 1: Modelo de construcción de prototipos.....	21
Ilustración 2: Diagrama de casos de uso (Resumen).....	31
Ilustración 3: Diagrama de casos de uso (Gestionar circulares)	31
Ilustración 4: Diagrama de casos de uso (Gestionar autorizaciones)	32
Ilustración 5: Diagrama de casos de uso (Gestionar encuestas).....	32
Ilustración 6: Diagrama de casos de uso (Gestionar cuenta).....	33
Ilustración 7: Instalación típica de una arquitectura multi-nivel	39
Ilustración 8: Diagrama de despliegue.....	40
Ilustración 9: Diseño de la Base de Datos	42
Ilustración 10: Diagrama del patrón Modelo-Vista-Controlador	45
Ilustración 11: Estructura de archivos de la aplicación.....	51
Ilustración 12: Estructura del directorio www	52
Ilustración 13: Inicio de sesión.....	65
Ilustración 14: Validación por SMS.....	66
Ilustración 15: Establecer código de seguridad.....	67
Ilustración 16: Registro	68
Ilustración 17: Selección de centros.....	69
Ilustración 18: Circulares.....	70
Ilustración 19: Búsqueda de mensajes.....	71
Ilustración 20: Contenido de circular	72
Ilustración 21: Autorizaciones.....	73
Ilustración 22: Contenido de autorización	74
Ilustración 23: Solicitud del código de seguridad.....	75
Ilustración 24: Contenido de encuesta	76
Ilustración 25: Mi perfil	77
Ilustración 26: Mis datos	78
Ilustración 27: Cambio de nombre.....	79
Ilustración 28: Mis hijos	80
Ilustración 29: Confirmación al desasociar hijo/a.....	81
Ilustración 30: Mis centros.....	82

Índice de tablas

Tabla 1: Resumen de casos de uso.....	34
Tabla 2: Especificaciones de casos de uso (Responder autorizaciones)	35
Tabla 3: Especificaciones de casos de uso (Responder encuestas).....	35
Tabla 4: Especificaciones de casos de uso (Cambiar código de seguridad)	36
Tabla 5: Especificaciones de casos de uso (Añadir centro/s)	37
Tabla 6: Contenido de las tablas de la Base de Datos	44

Introducción

El malgasto de los recursos del planeta en general nos parece un problema fundamental en nuestro sistema sociocultural. Por ello, hemos decidido realizar un proyecto que pueda colaborar a remediar este hecho en la medida de lo posible y de lo que nuestros conocimientos en la informática y nuestra capacidad de buscar soluciones nos lo permitan. Hemos decido centrarnos en una actividad que, si bien cotidiana, tiene como resultado un consumo de papel que creemos innecesario con las tecnologías disponibles hoy en día.

Por ello, hemos decidido realizar una propuesta para los centros educativos para evitar el método actual y poco amigable con el medio ambiente de comunicación con los padres. La finalidad de nuestra propuesta es poner fin al uso del papel para enviar mensajes de carácter general y/o autorizaciones a los padres por parte de los centros educativos.

Por consiguiente, hemos desarrollado una solución que permite a los centros enviar distintos tipos de mensajes con uno o varios administradores encargados de gestionar dicha mensajería a los padres de sus alumnos (autorizaciones de actividades, circulares de carácter general y encuestas) y que estos últimos puedan visualizarlas en sus dispositivos móviles e incluso responderlas desde el mismo, evitando costes y ofreciendo una alternativa más actualizada y respetuosa con el ecosistema.

En este Trabajo de Fin de Grado en concreto nos centraremos en la aplicación móvil dedicada a los padres, en la que podrán visualizar el contenido de los tres tipos de mensajes anteriormente mencionados y responder en consecuencia a aquellos en los que se solicite una respuesta, más específicamente, se permitirá autorizar o desautorizar a los hijos/as a asistir a actividades propuestas por el centro, así como responder a encuestas solicitadas por el centro para mejorar su comunicación con el mismo.

La aplicación destinada a ser gestionada por el centro será desarrollada por Adrián Louro Alonso en el Trabajo de Fin de Grado titulado: *“Análisis, Diseño e Implementación de un Backend para la Comunicación entre Centros Educativos y Padres de Alumnos”*, en la que se permitirá a uno o a varios administradores enviar autorizaciones, encuestas y circulares a los padres, además de permitir visualizar los resultados que se envían desde la aplicación cliente a desarrollar en este Trabajo de Fin de Grado.

Como se previó necesario, hemos además diseñado y desarrollado un servicio web *RESTful* que, haciendo uso de una capa de acceso a datos perteneciente al *backend* diseñado en el Trabajo de Fin de Grado encargado del mismo, permita a nuestra aplicación móvil acceder a la base de datos de la aplicación.

Estructura del documento

Esta memoria está separada por capítulos, que procederemos a describir brevemente en las siguientes líneas:

1. Estado actual y objetivos iniciales

Establecemos el punto de partida del proyecto, detallando la situación actual del problema, competidores de nuestra propuesta y los objetivos que nos hemos propuesto alcanzar con el proyecto.

2. Competencias específicas cubiertas

Enumeramos y justificamos cómo se han alcanzado las competencias cubiertas durante la realización del proyecto.

3. Aportaciones

Detallamos cuáles son las aportaciones que tiene nuestro proyecto en el apartado socioeconómico y cuál ha sido el impacto personal que ha tenido en nosotros.

4. Normativa y legislación

Explicamos nuestro análisis de la normativa y legislación vigente y qué medidas hemos tomado en el proyecto para adoptarlas, se incluyen también información sobre las licencias utilizadas de los diferentes productos software utilizados durante el desarrollo del proyecto.

5. Metodología de trabajo y planificación del proyecto

Especificamos cuál de las diferentes metodologías de trabajo hemos escogido y justificamos nuestra elección además de mostrar cuál ha sido la planificación del proyecto.

6. Tecnologías y herramientas utilizadas

Introducimos cuáles han sido las tecnologías escogidas para trabajar en el proyecto, así como las herramientas utilizadas para ello.

7. Análisis

Señalamos y detallamos el análisis realizado del proyecto, así como la identificación de requisitos funcionales y la esquematización de los mismos mediante diagramas de casos de uso.

8. Diseño

Definimos el diseño y la arquitectura bajo los cuáles se ha construido el sistema. Además, estos se justifican debidamente y se detalla su implementación y funcionamiento. Añadimos también el diseño utilizado para nuestra base de datos.

9. Desarrollo

Visualizamos de manera sencilla cómo ha sido el desarrollo en el proyecto, justificando la elección del *framework* escogido, la estructura de archivos de la aplicación móvil, y como se ha configurado y utilizado el paquete escogido para desarrollar nuestra API Restful.

10. Pruebas

Describimos las pruebas que hemos realizado para validar el funcionamiento de la aplicación, separadas por pruebas de usabilidad y pruebas de integración.

11. Resultados, conclusiones y trabajo futuro

Concluimos con el desarrollo de la memoria, señalando cuales han sido nuestras reflexiones, resultados finales y los posibles cambios y mejoras que se podrían introducir en el sistema en el futuro.

12. Bibliografía

Enunciamos todas las referencias a las que se ha hecho alusión durante la memoria, sus autores y sus fuentes respectivas.

13. Anexos

- I. Explicamos de manera concisa y sencilla cómo hacer uso de la aplicación móvil.
- II. Explicamos cómo instalar el sistema de la aplicación móvil y la API Restful para poder utilizarla en escritorio.
- III. Adjuntamos un *script* de sentencias SQL que dan como resultado la base de datos utilizada durante el transcurso del proyecto.

1. Estado actual y objetivos iniciales

1.1 Estado actual

1.1.1 Alternativas Existentes

Si queremos que nuestra aplicación tenga éxito, primeramente, deberemos asegurarnos de que problema queremos resolver. Una vez decidido este punto, debemos comprobar que competidores posibles tenemos en nuestro entorno, y como satisfacen los mismos las necesidades del usuario y cómo solventan el problema. Si conseguimos ofrecer alternativas a dichas soluciones, o mejores soluciones en general, podremos mejorar la probabilidad de éxito de nuestra aplicación incluso antes de comenzar el desarrollo de la misma.

Teniendo en cuenta esta premisa, hemos decidido realizar un análisis del mercado actual para ver que soluciones existen al problema de comunicación entre Centros y Padres haciendo uso de las nuevas tecnologías. A continuación, procederemos a enumerar dichas soluciones, así como comentar que carencias encontramos en las mismas y que no existen en nuestro primer concepto de aplicación.

- **[1] miColegioApp:**
 - Para el centro: permite el envío de circulares y adjuntar documentos a los mismos.
 - Para los padres: confirmar citas, firmar autorizaciones.
 - No permite: realización de encuestas y auto-importación de datos.
- **[2] TokApp:**
 - Para el centro: importación de datos, permite el envío de mensajes entre profesores y alumnos.
 - Para los padres: contactar con los profesores, con los alumnos y con el centro sin restricciones.
 - No permite: Firmar autorizaciones, enviar mensajes de un tipo específico de contenido, restringir la comunicación entre padres/profesores/centro.
- **[3] ClickEdu:**
 - Para el centro: contabilidad, gestión de recursos del centro, gestión de boletines de notas, actividades extraescolares, realización de exámenes, entrega de trabajos.
 - Para los padres: Avisos, Emails:
 - No permite: Firmar autorizaciones, responder encuestas, auto-importación de datos.

- **[4] Remind:**
 - Para el centro: creación de grupos de alumnos/ profesores/padres en forma de chats, envío de circulares.
 - Para los padres: comunicación vía chat con cualquier profesor, alumno o padre.
 - No permite: firmar autorizaciones, responder encuestas, enviar circulares a todos los usuarios, auto-importación de datos.

- **[5] ClassDojo:**
 - Para el centro: permite al profesorado crear una red social con sus alumnos, valorar sus comentarios, enviar mensajes a sus alumnos en específico.
 - Para los padres: Permite visualizar fotografías tomadas en clase o en actividades por parte de los padres, actualizaciones por parte del profesor y comunicación privada con el profesorado.
 - No permite: restringir la comunicación padre/profesorados, envío de circulares, envío de encuestas, adjuntar documentos, enviar encuestas.

1.1.2 Conclusiones

Como hemos podido observar, en general tenemos dos grupos de aplicaciones:

- Las primeras están destinadas a la gestión completa del centro: boletines de notas, calificaciones, gestión de salarios, contabilidad, horarios, realización de exámenes, entregas de trabajos, etc. Por ello, el apartado de la mensajería flaquea y es de carácter general, no suelen permitir diferenciar entre mensajes de carácter informativo de aquellos que requieren de una retroalimentación por parte de los padres.
- Las segundas están destinadas a la mensajería, pero a una mensajería de carácter individual o de grupos pequeños, teniendo los chats como abanderados. Este tipo de aplicaciones es efectivo para tener comunicación en tiempo real y volcar contenido multimedia, pero no lo es tanto para el envío más general de información o documentación, como son las circulares, las encuestas o autorizaciones.

Además de lo mencionado, ambos tipos de aplicación suelen carecer de la auto-importación de datos, algo que consideramos esencial, debido a lo tedioso que puede ser realizar la importación de cientos de alumnos y sus respectivos padres de manera manual.

Respecto a la mensajería, nuestra aplicación se centra más en la información general y de carácter oficial, y permite suplir la mensajería individual que las alternativas ofrecen, sin llegar a proporcionar un libre albedrío a la hora de comunicarse con el centro o el profesorado: toda la comunicación se realiza desde el centro y es este el que decide si desea una respuesta y de qué tipo específico la desea.

Por último, ambos grupos cuentan con aplicaciones pensadas para ser utilizadas por padres acostumbrado a las interacciones más modernas pero que pueden ser no tan intuitivas, y nuestra solución brinda la facilidad de uso por su claridad y sencillez, intentando ser lo más intuitiva y clara posible.

1.2 Objetivos

El objetivo principal de nuestra solución es ofrecer una solución sencilla, amigable con el medioambiente y viable incluso para aquellos centros con infraestructuras mínimas y personal inexperto.

En el caso específico de este Trabajo de Fin de Grado, los objetivos son claros:

- Ofrecer una interfaz de usuario sencilla, clara e intuitiva.
- Proporcionar siempre al usuario transparencia sobre que acciones realiza en la aplicación.
- Garantizar la seguridad del usuario, la privacidad de sus datos y que centros pueden verlos y sobre cuando realizamos acciones que sustituyen a la firma tradicional.
- Asegurarnos de que la información de los centros y la retroalimentación que estos requieren llegue a los padres y no se extravíe como podría suceder con métodos más tradicionales.

Para ello, desde la aplicación móvil, la interfaz está diseñada en un modelo de pestañas, con iconos claros y que representan las diferentes secciones de la aplicación, se informa de manera activa y retroactiva al usuario cuando accede al contenido de los mensajes y de cuando envía respuestas a los mismos y los padres sólo deberán introducir su número de teléfono para iniciar sesión y su nombre y apellidos si nos encontráramos en el registro. La asignación de padres/alumnos se realizaría por parte del centro.

2. Competencias específicas cubiertas

2.1 Comunes a la Ingeniería Informática

CII01

Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.

Durante el proyecto hemos tenido que diseñar y desarrollar la aplicación móvil para los padres, así como evaluar que funciona en todas las condiciones, que cumple los requisitos de seguridad y de calidad establecidos para los mismos, así como asegurar que sólo los centros que un padre seleccione tendrán acceso a su información, para cumplir con la normativa y legislación vigente de protección de datos.

CII03

Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software.

Debido al carácter colaborativo del proyecto, ha sido de vital importancia establecer una buena comunicación entre los dos autores de las dos partes que componen el mismo. Por ello, sobre todo en las fases de análisis y diseño, colaboramos íntimamente con el autor del Trabajo de Fin de Grado encargado del *backend*, para así poder evitar en gran medida posibles problemas que pudieran surgir posteriores al desarrollo de integración de ambas aplicaciones.

CII08

Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

Debido al carácter original del proyecto, hemos tenido que ejecutar todas y cada una de las fases de un proyecto: desde el análisis, pasando por el diseño, hasta el desarrollo. Como el prototipo final escogido de nuestra aplicación requería ser multiplataforma, hemos estudiado las distintas opciones de tecnologías disponibles, escogiendo finalmente *ionic* como *framework* del desarrollo. Para poder comunicarnos con el *backend* debíamos también desarrollar un servicio web *RESTful* en el proyecto del mismo, basado en *Symfony*. Por ello decidimos utilizar el paquete *Friends Of Symfony* que nos permite una configuración sencilla y una implementación mecánica de los métodos requeridos en el servicio.

CIIO12

Conocimiento y aplicación de las características, funcionalidades y estructura de las bases de datos, que permitan su adecuado uso, y el diseño y el análisis e implementación de aplicaciones basadas en ellos.

Debido a las dimensiones del proyecto, nos hemos visto obligados a realizar un análisis exhaustivo y un diseño preciso de la base de datos del sistema, y haciendo uso de un *Object-Relational Mapping* hemos conseguido poder generar objetos a partir de los datos. Con ello hemos sacado partido de las ventajas que nos proporciona la programación orientada a objetos aprendidas durante el grado.

CIIO13

Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los Sistemas de información, incluidos los basados en web.

Dado que el proyecto completo consta de dos aplicaciones, como ya se ha mencionado previamente, se necesita desarrollar una API que nos permita comunicar ambas aplicaciones sobre el mismo sistema de base de datos.

CIIO16

Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

Realizando una comparación entre los distintos ciclos de vida y los principios de los mismos, concluimos que dada nuestra manera de trabajar la mejor opción a escoger sería el desarrollo basado en prototipos, permitiéndonos generar en cada iteración un prototipo funcional que nos permitiera validar los requisitos del usuario, así como comprobar la integración de nuestra aplicación y la del compañero que trabajaba en el *backend* del proyecto.

CIIO17

Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.

Por el formato móvil de nuestra aplicación, resultó muy importante identificar los elementos a destacar dentro de nuestra interfaz, simplificándola lo mayor posible para conseguir una destacable facilidad de uso y sencillez. Se han utilizado como base para ello los contenidos aprendidos en la asignatura de Diseño de Interfaces de Usuario, así como otros adquiridos durante nuestra formación.

CII018

Conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional.

Hemos realizado una búsqueda concisa en referencia a aquellos apartados de la legislación actual que se aplican en los distintos apartados de nuestra aplicación, para cumplir de manera adecuada con la normativa y regulación vigentes.

2.2 Trabajo Fin de Grado

TFG01

Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas.

Teniendo en cuenta que es un proyecto complejo, y además original, hemos tenido que pasar por todas las fases de un desarrollo de proyecto completo que nos ha permitido emplear muchos de los conocimientos adquiridos en el grado.

2.3 Ingeniería del Software (IS)

IS01

Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la ingeniería del software.

Utilizando una metodología de software hemos conseguido simplificar la organización de nuestro proyecto, así como realizar un seguimiento y validación de que el mismo satisfacía las necesidades del usuario. Además, haciendo uso de distintas prácticas de la ingeniería del software hemos conseguido generar un sistema fiable, eficiente y robusto, sin perder la flexibilidad de por medio.

IS02

Capacidad para valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas necesidades, reconciliando objetivos en conflicto mediante la búsqueda de compromisos aceptables dentro de las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones.

En las primeras etapas del desarrollo realizamos un amplio estudio de las necesidades de nuestros usuarios, mediante diagramas de casos de uso y tablas de especificaciones de los mismos. Sin embargo, una vez establecidas, realizamos un barrido posterior por los mismos, descartando algunas funcionalidades y priorizando otras, debido a los márgenes de tiempo en las entregas a los que nos enfrentábamos.

IS03

Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.

Haciendo uso de un sistema de control de versiones aprendido en Herramientas Profesionales de Desarrollo Software, más concretamente Git, hemos podido trabajar los dos alumnos, cada uno en su respectiva aplicación de manera independiente, sin interrumpir el trabajo del otro a pesar de tener una parte común en el *backend*.

IS04

Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

Nos hemos encontrado con diversos problemas durante el desarrollo de la aplicación, y para ello hemos tenido que ser capaces de analizarlos, entenderlos, depurar el código causante de dichos problemas y por último encontrar una solución no sólo válida, si no robusta y de calidad. Además, cada vez que se solucionaba uno de estos problemas fue debidamente documentado en el sistema de control de versiones para tener referencias en posibles futuros problemas relacionados.

2.4 Tecnologías de Información (TI)

TI03

Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas.

La elección de una metodología de desarrollo basada en prototipos nos ayuda a centrarnos en el usuario final y a validar cada una de las funcionalidades que se van implementando antes de llegar al producto final. También, el uso de iteraciones nos ha ayudado enormemente a planificar y organizar el desarrollo.

TI06

Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil.

Reincidiendo en lo anteriormente mencionado, nuestra aplicación es un cliente de la desarrollada por Adrián Louro Alonso centrada en el *backend*, por lo que además en este trabajo de fin de grado hemos realizado una implementación de un servicio web *RESTful* que nos sirva de enlace entre ambos para poder gestionar todo el sistema en la misma base de datos.

3. Aportaciones

3.1 Entorno socioeconómico

Este proyecto se centra en tener dos impactos principales: el primero, en el medioambiente y el malgasto del papel y el segundo, mejorar la comunicación de Centros/Padres de manera sencilla y efectiva.

Evitar el malgasto del papel haciendo uso de nuestra aplicación tiene varias repercusiones, entre ellas dejamos una selección de las que consideramos importantes para el medio ambiente:

- Contribuimos a evitar el efecto invernadero.
- Reducimos la contaminación de la tierra, el agua y el aire derivada de los residuos en la producción de papel.
- Reducimos la tala de árboles derivada de la fabricación de papel. Y por tanto también se ve reducida la probabilidad de incendios producidos por la misma, así como el desgaste y erosión de terrenos por la inexistencia de bosques, sin olvidar las sequías en las que pueden derivar los factores anteriormente mencionados.
- Contribuimos a la protección y conservación de hábitats y las especies animales y vegetales que habitan en las zonas de tala.
- Contribuimos a reducir el progreso del cambio climático.

También supone numerosas ventajas para el centro educativo, de entre las cuáles destacamos:

- Reducir los gastos generales: no sólo de la impresión, también de los elementos adheridos a ella – recambios y mantenimiento de impresoras, el propio papel, la tinta- así como gastos de almacenamiento y envío.
- Ahorro de espacio: el espacio físico que se ahorra al no tener que recoger en un mismo sitio las distintas autorizaciones/encuestas/circulares impresas en papel y organizadas correctamente.
- Eficiencia: evitamos que los administrativos encargados de revisar las respuestas a autorizaciones/encuestas tengan que realizar un proceso iterativo de revisión de todas y cada una de las mismas, pasando a una visualización global de los resultados de manera directa y sencilla.
- Mayor seguridad: las copias de seguridad de la base de datos nos añaden un extra de seguridad que, utilizando papel, sólo sería alcanzable mediante copias físicas de los documentos.
- Flexibilidad en las jornadas laborales: no se requiere que los empleados de la administración del centro estén en presencia física en el mismo para realizar

alguna de sus tareas, desde la plataforma web pueden acceder al sistema y realizarlas independientemente de donde se encuentren.

3.2 Personal

En el ámbito personal, hemos aprendido muchísimo. Hemos podido aprender a realizar un trabajo original, pasando por cada una de sus etapas, análisis, diseño y desarrollo. Dichas etapas nos han brindado oportunidades de poner en práctica mucho de los contenidos aprendidos durante el grado y, en especial, en la especialidad de Ingeniería del Software.

En el análisis, hemos aprendido a utilizar herramientas útiles como son los diagramas de casos de uso y sus especificaciones para identificar y validar correctamente los requisitos de usuario, así como experimentar de primera mano lo que es realizar un análisis en colaboración de otro desarrollador, Adrián Louro Alonso, encargado del *backend* del sistema, mejorando sustancialmente el resultado final.

En el apartado de diseño, hemos podido aplicar los diferentes patrones de diseño y arquitecturas del software aprendidas, realizando un diseño coherente para dos aplicaciones que deben trabajar en conjunción la una con la otra. Sin duda, el haber contado con un compañero ayudó a que este apartado tuviera más puntos de vista y mejoras de lo que podría haber sido en el caso de realizar un diseño en solitario.

Y a nivel de desarrollo, realizado de manera individual, la experiencia ha sido muy grata y enriquecedora. Tener un primer contacto con nuevas tecnologías desconocidas hasta ahora tales como: *Symfony 3*, *PHP 7.1* y *Ajax* para la parte de la API que sirve de enlace entre ambas aplicaciones como *Ionic*, *AngularJS* y *Javascript* para la parte de la aplicación cliente del sistema, nos ha ayudado a experimentar como es adaptarnos a una nueva tecnología, analizar su estructura, buscar detenidamente en la documentación y poner a prueba los conocimientos que requerimos.

Como extra, el hecho de haber tenido una parte de integración de las dos aplicaciones nos ha ayudado mutuamente a identificar problemas que no habríamos considerado, no sólo a nivel de integración, si no a nivel del funcionamiento individual de cada una de las aplicaciones asociadas a los trabajos de fin de grado respectivos.

4. Normativa y legislación

En esta sección se tratarán los aspectos relacionados con la normativa vigente y las licencias de los productos software utilizadas durante el desarrollo de la aplicación que comprende este Trabajo de Fin de Grado.

4.1 Licencias de Software

Una licencia de software es un contrato entre el autor del software y el usuario/consumidor del mismo en el que se estipulan los términos y condiciones bajo los que el creador del software autoriza su uso y explotación bajo un tipo de licencia determinado.

Bajo estas líneas se indican cuáles son las licencias software respectivas para cada uno de los productos Software que hemos utilizado durante nuestro desarrollo:

4.1.1 GNU GPL

[6] La Licencia Pública General de GNU (GNU GPL) es la licencia de derecho de autor más ampliamente usada en el mundo del software libre y código abierto, y garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software.

Su propósito es doble: declarar que el software cubierto por esta licencia es libre, y protegerlo (mediante una práctica conocida como *copyleft*) de intentos de apropiación que restrinjan esas libertades a nuevos usuarios cada vez que la obra es distribuida, modificada o ampliada.

Esta licencia es utilizada por los siguientes productos software utilizados durante el desarrollo:

- **[7] XAMPP:** es una compilación de software libre, gratuita y se puede compartir con libertad bajo los términos de la Licencia Pública General de GNU. Pero sólo la compilación de XAMPP está publicada bajo la GPL. Para cada uno de los productos que abarca este software se recomienda comprobar su licencia y qué permiten o no permiten. En el caso de uso comercial en específico, se recomienda sobre todo revisar la licencia de MySQL. Desde el punto de vista comercial XAMPP también es gratuito.
- **[8] MySQL:** para aquellos desarrolladores de aplicaciones *Free Open Source Software (FOSS)* bajo la GPL que quieran combinar y distribuir dichas aplicaciones con el software de MySQL, MySQL de Oracle bajo GPL es la mejor opción. Para desarrolladores y distribuidores de software *open source* bajo una licencia *FOSS* que no sea la GPL, Oracle pone en disponibilidad sus Librerías de Cliente de

MySQL bajo GPL, con una excepción que permite el uso de dichas librerías bajo ciertas condiciones sin implicar que la totalidad del trabajo esté sujeto a la GPL. Por último, para realizar un uso comercial de MySQL sin estar sujeto de ninguna manera a la GPL, los desarrolladores deben llegar a un acuerdo comercial con Oracle.

- **[9] StarUML:** la versión 1 de este software, y su desarrollo están sujetos a la GPL salvando las siguientes dos excepciones:
 - Permitir enlazar varias librerías y componentes comerciales.
 - Permitir enlazar módulos *plug-in* propietarios.

De la versión en adelante, este producto software ya no está sujeto a la GPL. Por lo tanto, su uso requiere la adquisición de una licencia y si su fin es lucrativo, de un acuerdo comercial.

4.1.2 Licencia Apache

[10] La licencia Apache es una licencia de software libre permisiva creada por la *Apache Software Foundation (ASF)*. La licencia Apache requiere la conservación del aviso de derecho de autor y el descargo de responsabilidad, pero no es una licencia copyleft, ya que no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas.

Dicha licencia se utiliza en el Servidor Apache configurado usando XAMPP y en el entorno de desarrollo de aplicaciones móviles Apache Cordova, requerido por Ionic.

4.1.3 Licencia PHP

[11] La licencia PHP es la licencia bajo la que se publica el lenguaje de programación PHP. De acuerdo a la *Free Software Foundation* es una licencia de software libre no copyleft y una licencia de código abierto según la *Open Source Initiative*. Debido a la restricción en el uso del término “PHP”, no es compatible con la licencia GPL.

La licencia PHP está diseñada para incentivar la distribución del código fuente. Se permite la redistribución del contenido licenciado en forma de código fuente o binaria siempre y cuando se cumplan los siguientes requisitos:

1. Se incluya la declaración de los derechos de autor de la licencia PHP;
2. La palabra "PHP" no se use en el título de las obras derivadas;
3. Se incluya el siguiente anuncio bajo cualquier forma en la que se redistribuya el código: *“This product includes PHP software, freely available from <http://www.php.net/software/>”*

Esta licencia ha sido utilizada por el producto software PHP 7.

4.1.4 Licencia MIT

[12] La licencia MIT es una Licencia de software libre permisiva lo que significa que impone muy pocas limitaciones en la reutilización y por tanto posee una excelente compatibilidad de licencia. La licencia MIT permite reutilizar software dentro de software propietario. Por otro lado, la licencia MIT es compatible con muchas licencias *copyleft*, como la GNU GPL (software con licencia MIT puede integrarse en software con licencia GPL, pero no al contrario). El texto de la licencia no tiene copyright, lo que permite su modificación.

Esta licencia permite reutilizar el software así licenciado tanto para ser software libre como para ser software no libre, permitiendo no liberar los cambios realizados al programa original.

También permite licenciar dichos cambios con licencia BSD, GPL u otra cualquiera que sea compatible (es decir, que cumpla las cláusulas de distribución).

Con esta licencia se tiene software libre. Ejemplos en los que podría interesar su aplicación serían las licencias duales, si se pretende difundir un estándar mediante una implementación de referencia, o si simplemente se pretende que el producto sea libre sin mayores consideraciones.

La licencia MIT es utilizada por los siguientes productos software usados en el desarrollo: Symfony, Doctrine, JQuery, Ionic, NodeJS, MomentJS, Angular Moment, Ionic Filter Bar y Atom.

4.1.5 Licencia para educación de JetBrains

[13] La licencia de JetBrains para educación tiene un formato de suscripción anual y es gratuita para los miembros de la comunidad universitaria. No obstante, comprende una serie de restricciones en lo que respecta a su uso, por lo que no se permite:

- Alquilar, reproducir, modificar, adaptar, crear trabajos derivados, vender, sublicenciar o transferir los productos, o proveer dicho producto asociado a nuestra cuenta a terceros.
- El uso de ingeniería inversa, descompilar, desensamblar, modificar, traducir los productos o cualquier intento de descubrir el código fuente de los productos.
- Eliminar u ocultar cualquier aviso de propiedad o de otro tipo contenidos en los productos.
- Usar los productos para propósitos comerciales.

Dicha licencia se ha utilizado en el IDE PhpStorm.

4.1.6 Licencia de Justinmind

[14] Justinmind considera las siguientes declaraciones en lo que respecta al copyright y a los derechos de propiedad:

1. Justinmind o sus proveedores son los propietarios de los derechos de propiedad intelectual de cualquier y todos los componentes protegibles del Servicio, incluyendo por tanto al nombre del Servicio, trabajo artístico y elementos de la interfaz de usuario final contenidos en el Servicio. No se permite copiar, modificar, adaptar, reproducir, distribuir, realizar ingeniería inversa, descompilar, o desensamblar cualquier aspecto del Servicio de los cuáles Justinmind o sus proveedores sean propietarios.
2. Justinmind no reclama los derechos de propiedad intelectual del Contenido que sea subido o proveído al Servicio. Sin embargo, si se usa el Servicio para enviar contenido, se está de acuerdo en que otros puedan ver y compartir el Contenido.

Esta licencia va ligada al producto Justinmind Prototyper.

4.1.7 Licencia para educación de Office 365

[15] La licencia para educación de Office 365 destinada a la Universidad enumera las siguientes restricciones en lo que respecta su uso:

1. Debe ser un “Usuario educacional cualificado” para suscribirse y utilizar el servicio y el software de edición “University”.
2. Suscripción No para reventa. Las tarjetas de suscripción No para reventa se distribuyen con propósitos limitados. No puede vender las tarjetas de suscripción identificadas como “NPR” o “No para reventa”.
3. Programas de terceros. El software puede incluir programas de terceros que Microsoft, no los terceros, le licencia a usted conforme a este contrato. Las notificaciones, si las hay, para el programa de terceros se incluyen para su información solamente.
4. Microsoft le concede una licencia para copiar, distribuir, realizar y mostrar elementos multimedia (imágenes, imágenes prediseñadas, animaciones, sonidos, música, clips de vídeo, plantillas y otro contenido) incluidos con el servicio y/o el software en proyectos y documentos; sin embargo, no puede: (i) vender, licenciar ni distribuir copias de elementos multimedia por sí mismos o como un producto, si el valor principal del producto son los elementos multimedia; (ii) conceder a sus clientes el derecho de volver a licenciar o distribuir los elementos multimedia; (iii) licenciar o distribuir para propósitos comerciales elementos multimedia que incluyan la manifestación de personas, gobiernos, logotipos, marcas o emblemas identificables, o utilizar estos tipos de imágenes de maneras que pudieran implicar una aprobación o una asociación

con su producto, entidad o actividad; o (iv) crear trabajos obscenos o escandalosos mediante los elementos multimedia. Otros elementos multimedia, a los que se puede acceder en otros sitios web a través de las características de Office, se rigen por los términos de esos sitios web.

Se ha utilizado dicha licencia en el producto software Microsoft Word 2016.

4.2 Seguridad de los Datos

[16] La Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal, (LOPD), es una ley orgánica española que tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y privacidad personal y familiar.

Su objetivo principal es regular el tratamiento de los datos y ficheros, de carácter personal, independientemente del soporte en el cual sean tratados, los derechos de los ciudadanos sobre ellos y las obligaciones de aquellos que los crean o tratan. Esta ley afecta a todos los datos que hacen referencia a personas físicas registradas sobre cualquier soporte, informático o no.

Para cumplir con esta ley, se debe garantizar que la protección de datos personales es un elemento presente en todos aquellos apartados relacionados de nuestra aplicación, en especial en lo que respecta a información almacenada, esto es: en la base de datos.

En el caso específico de nuestra aplicación, para cada persona física se recogen los siguientes datos:

- Nombre y Apellidos.
- Número de teléfono personal.

Por lo tanto, figuran en el denominado **[17] Nivel básico** por lo que se deben garantizar los siguientes requisitos y se han adoptado las siguientes medidas:

- Dicha información no es accesible por cualquier persona, y sólo aquellos habilitados para ello tendrán acceso a ella.
 - Para ello, hemos garantizado que dicha información sólo será visible por los propietarios de la misma, así como únicamente por los Administradores designados por el centro para hacer uso del sistema.
 - El acceso a la base de datos está protegido por contraseña, garantizando que solo personas integrantes del equipo de desarrollo de la empresa puedan acceder a ella.
 - La información referente a los padres será únicamente visible para aquellos centros que los padres hayan configurado para ceder su información durante la configuración inicial de su cuenta.

- Esta información tiene que ser necesaria además de contar con fines determinados explícitos y explícitos, garantizando que no se usarán para otros fines que no sean los acordados. Nunca se exigirá al usuario de la aplicación datos innecesarios o superfluos conforme a lo requerido para el correcto funcionamiento de la aplicación.
- Se garantiza que la toma de decisiones en las autorizaciones por parte de los padres para actividades de los centros está protegida por contraseña, siendo únicamente los padres los conocedores de la misma, en forma de un código de seguridad de 4 dígitos.
- El sistema garantiza a que dichos datos sólo serán albergados en su base de datos durante el tiempo necesario para cumplimentar la funcionalidad que se requiere por parte de los centros. En caso de no ser necesarios en cierto punto, serán eliminados de la base de datos del sistema, previa solicitud por parte del centro o de un padre/madre.

5. Metodología de trabajo y planificación del proyecto

Metodología de trabajo

En lo que a la metodología de trabajo respecta hemos escogido basarnos en el **[18] modelo de ciclo de vida del software basado en prototipos** pues es la mejor que se adapta a nuestra forma de trabajo, teniendo en cuenta que el proyecto tiene dos aplicaciones independientes nos permitía poder construir nuestras aplicaciones de manera incremental e ir verificando en cada etapa que se alcanzaban los objetivos y se validaban los requisitos considerados en el análisis junto a la ayuda de nuestro tutor.

Para explicar mejor en que consiste este ciclo de vida, primeramente, definiremos que es un prototipo:

Los prototipos son una representación limitada de un producto, que permite a ambas partes interesadas en el desarrollo (cliente y desarrollador) probarlo en situaciones reales o explorar su uso, creando así un proceso de diseño de iteración que genera calidad.

Teniendo claro lo que es un prototipo procedemos a incidir en la metodología de desarrollo basada en prototipos. Este tipo de metodología pertenece a los modelos de desarrollo evolutivo, por tanto, se establece una primera implementación del proyecto, y se itera sobre este consecutivamente hasta llegar al producto final. Destacan por estar diseñados para ajustarse al cambio durante el desarrollo del proyecto, por lo que resulta ideal en un trabajo original como este.

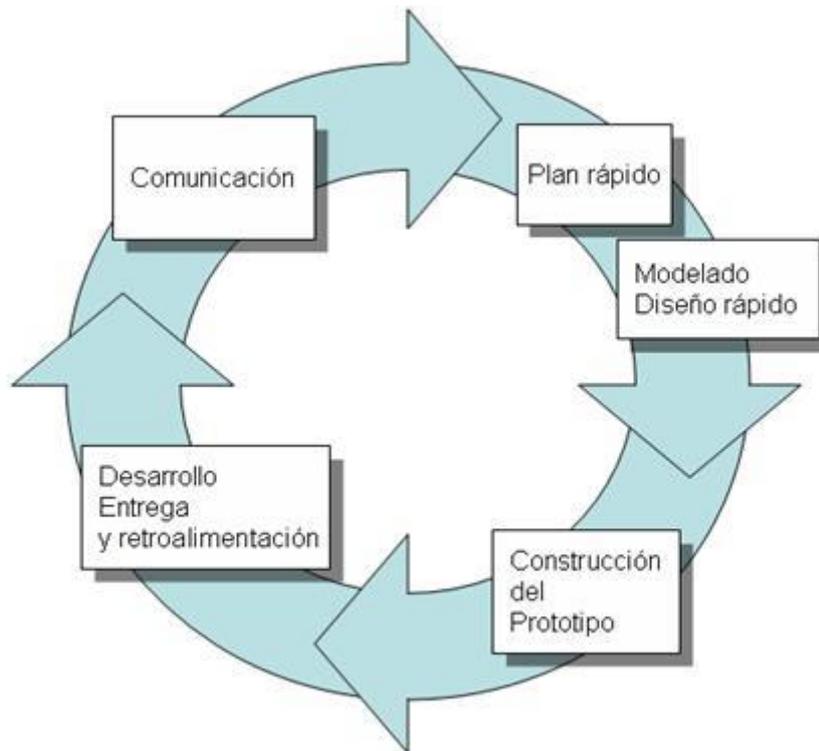


Ilustración 1: Modelo de construcción de prototipos

Tal y como se puede visualizar en la *Ilustración 1* este modelo sigue una estructura iterativa que consta de las siguientes etapas:

- Plan rápido, Modelado y Diseño rápido: esta etapa se centra en la representación de aquellos aspectos visibles para el cliente.
- Construcción del prototipo: se construye un prototipo funcional o no, en el que el cliente valida si dichas funcionalidades son las deseadas.
- Desarrollo, Entrega y retroalimentación: se desarrollan las funcionalidades y se entregan al cliente, consiguiendo una retroalimentación de su parte al respecto acerca de si los requisitos han sido validados correctamente o no.
- Comunicación: El cliente comunica cambios que desea realizar en la implementación o posibles mejoras antes de cerrar la iteración.

Por dicho funcionamiento, el modelo basado en la construcción de prototipos disfruta de las siguientes **[19] ventajas e inconvenientes:**

Ventajas

- Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.

- En nuestro caso particular, nuestro tutor comparte la autoría de la idea original del proyecto y, por tanto, conoce en detalle cuáles son los objetivos generales de la aplicación para móvil, inclusive conociendo e ayudándonos a identificar los requisitos para la misma, lo cual es una gran ventaja.
- También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.
 - Dada nuestra falta de experiencia en desarrollar aplicaciones móviles y que, en general, todo desarrollo tiene margen de mejora, resulta ideal poder validar que cada implementación está yendo en la dirección adecuada para validar los requisitos del producto conforme se va realizando.
- Se puede reutilizar el código.
 - Hemos podido aprovechar durante el desarrollo código implementado durante los prototipos parcialmente funcionales, ahorrándonos trabajo y sobre todo tiempo de desarrollo para cumplir con las entregas.

Inconvenientes

- El usuario tiende a crearse unas expectativas cuando ve el prototipo de cara al sistema final. A causa de la intención de crear un prototipo de forma rápida, se suelen desatender aspectos importantes, tales como la calidad y el mantenimiento a largo plazo, lo que obliga en la mayor parte de los casos a reconstruirlo una vez que el prototipo ha cumplido su función. Es frecuente que el usuario se muestre reacio a ello y pida que sobre ese prototipo se construya el sistema final, lo que lo convertiría en un prototipo evolutivo, pero partiendo de un estado poco recomendado.
 - En este caso, dado que las validaciones han sido realizadas por el tutor, sustituyendo al usuario, y que el mismo conoce las características y el proceso de esta metodología no hemos tenido ninguno de estos problemas.
- En aras de desarrollar rápidamente el prototipo, el desarrollador suele tomar algunas decisiones de implementación poco convenientes (por ejemplo, elegir un lenguaje de programación incorrecto porque proporcione un desarrollo más rápido). Con el paso del tiempo, el desarrollador puede olvidarse de la razón que le llevó a tomar tales decisiones, con lo que se corre el riesgo de que dichas elecciones pasen a formar parte del sistema final.
 - Gracias a nuestra experiencia y las buenas prácticas aprendidas durante el grado hemos podido evitar en mayor medida la mala toma de

decisiones al realizar el prototipo y escoger las tecnologías para el desarrollo.

Conclusiones

A pesar de que tal vez surjan problemas, la construcción de prototipos puede ser un paradigma efectivo para la ingeniería del software. La clave es definir las reglas del juego desde el principio; es decir, el cliente y el desarrollador se deben poner de acuerdo en:

- Que el prototipo se construya y sirva como un mecanismo para la definición de requisitos.
- Que el prototipo se descarte, al menos en parte.
- Que después se desarrolle el software real con un enfoque hacia la calidad.

Pudiendo tener estas condiciones claras y siguiendo las indicaciones del tutor en el papel de cliente, al final esta metodología nos ha aportado muchas ventajas y muy pocas desventajas reales, claro que todo ello es debido a la naturaleza de nuestro cliente por lo que dichos resultados no deben desvirtualizarse y generalizarse, pues la metodología basada en prototipos aún contempla ciertos riesgos.

5.2 Planificación del proyecto

5.2.1 Planificación inicial

Para realizar una planificación correcta del proyecto, primeramente, realizamos un estudio del estado del arte, para poder identificar cuáles eran los usuarios principales de nuestra aplicación, así como que funcionalidades requerirían los mismos dentro de nuestro sistema. A partir de esta información pudimos identificar mejor cuáles serían los actores de nuestro sistema y que requisitos funcionales precisaba nuestro sistema. Para representar esta información y tenerla a nuestra disposición de manera clara y sencilla, construimos una serie de diagramas de casos de uso, con sus respectivas tablas de especificación, donde se detalla al completo la funcionalidad a implementar.

Realizada dichas representaciones procedimos a validarlas con nuestro tutor, realizando los ajustes que consideramos oportunos en las mismas. Una vez actualizadas, pasamos a realizar una primera representación gráfica de la interfaz de nuestro sistema, haciendo uso del producto software Justinmind Prototyper. Dicha interfaz gráfica nos ayudó muchísimo a darnos especificar mejor las funcionalidades que habíamos definido, así como a identificar algunas que nos habíamos planteado. Sumándose a esto, nos permitió visualizar cómo interaccionarían los usuarios con sus aplicaciones y sus respectivas funcionalidades.

Para continuar, teníamos que definir bien cuál iba a ser la arquitectura de nuestro sistema, hecho esto, deberíamos realizar un análisis de las tecnologías disponibles para el desarrollo del sistema, ver sus distintas ventajas e inconvenientes para nuestra aplicación, y realizar una elección en base a cuál de ellas se adaptaba mejor a nuestras necesidades.

Al final nos acabamos decantando por las tecnologías de Ionic y AngularJS, declarándonos así abanderados de las aplicaciones móviles híbridas multiplataforma. Una vez escogidas dichas tecnologías, comenzó un periodo de aprendizaje de dichas tecnologías, siguiendo las documentaciones correspondientes y realizando pequeñas aplicaciones superfluas que nos permitieran aprender lo más posible acerca de las mismas, así como ganar agilidad desarrollando con ellas.

Una vez ganada cierta destreza con dichas tecnologías comenzó el desarrollo de la aplicación móvil y su servicio *RESTful* correspondiente. Siguiendo las directivas del modelo basado en prototipos, detallado en el apartado anterior de esta memoria, el proyecto fue dividido en una serie de iteraciones y se fueron realizando conforme al ciclo de este modelo. Dichas iteraciones consistieron en lo siguiente:

- 1º. Se realizó una instalación de las herramientas necesarias para el desarrollo, así como del proyecto del *backend*. Realizamos una primera implementación del inicio de sesión y registro en el sistema, así como de los métodos necesarios para ello en la API REST.
- 2º. Se desarrollaron las vistas de las distintas pestañas principales de la aplicación. Se implementaron las visualizaciones del listado y el contenido de Circulares, Autorizaciones y Encuestas, así como de la vista de Mi Perfil.
- 3º. Realizamos la implementación del envío de respuestas a las autorizaciones y encuestas desde la vista del contenido de las mismas, así como de los métodos del servicio web requeridos para su correcto funcionamiento. Además, se añadió el código de seguridad como método de autenticación.
- 4º. Se implementaron las funcionalidades respectivas a la pestaña de Mi Perfil, tales como la visualización de Mis Datos, Mis Hijos y la selección de Centros en el apartado Mi Centro. También se corrigió el flujo del inicio de sesión/registro para que incluyera un establecimiento del código de seguridad y de una selección de centros. Por ende, se añadieron también los métodos del servicio necesarios para el funcionamiento de dichas vistas.
- 5º. Por último, se realizó una refactorización del código en donde fue posible, ahorrando mucho código previamente escrito al comienzo del desarrollo, sustituible por directivas, así como una mejor organización de las rutas de la API para seguir las buenas prácticas propuestas por las APIs REST pragmáticas. Y para finalizar se realizó un control de errores y validación de formularios en aquellas partes de la aplicación en las que era necesario.

5.2.2 Ajustes de la planificación

La aplicación ideada originalmente contemplaba una implementación de un sistema de tutorías entre los padres de los alumnos y el profesorado del centro. Por ende, el sistema a desarrollar era mucho más complejo, dicho sistema implicaría añadir a las funcionalidades ya existentes las que están bajo estas líneas:

- Administradores del centro:
 - Registrar profesores en el centro.
 - Asociar profesores a un curso.
 - Asociar un tutor a un curso.
 - Establecer asignaturas en el centro.
 - Asociar asignaturas a un profesor.
- Profesores:
 - Establecer un horario de tutorías.
 - Solicitar tutorías a los padres.
 - Ver calendario de tutorías.
 - Confirmar tutorías.
 - Cancelar tutorías.
- Padres:
 - Solicitar tutorías a los profesores de sus hijos.
 - Confirmar asistencia a una tutoría.
 - Cancelar tutorías.

Conforme avanzábamos en el desarrollo, llegamos a la conclusión de que dichas implementaciones requerirían de un tiempo y adaptación a las nuevas tecnologías de las que no disponíamos. Por ello, consultando con Adrián Louro Alonso y nuestro tutor concluimos que, para cumplir con las iteraciones planificadas y los plazos de entrega con la mejor calidad posible, lo mejor sería descartar estas ideas en la implementación de este trabajo de fin de grado.

6. Tecnologías y herramientas utilizadas

En este apartado detallaremos cuáles fueron las herramientas y tecnologías escogidas para el desarrollo de la aplicación dividiéndose en tres categorías: Tecnologías, Herramientas y Servidores.

6.1 Tecnologías

- **[20] Ionic:** es un *framework* gratuito y de código abierto para desarrollar aplicaciones híbridas multiplataforma que utiliza las tecnologías de Cordova, HTML5 y CSS como base y hace uso de AngularJS para gestionar las aplicaciones. El desarrollo se ha realizado en su versión 1 dado que utiliza angular 1 como base, y por ello su diseño está basado en el patrón Modelo-Vista-Controlador, con el que nos hemos familiarizado durante el grado. También incorpora componentes de interfaz de usuario sencillos y personalizables, y el resultado final es muy similar a una aplicación nativa, salvando el hecho de que la aplicación sólo se desarrolla una vez para distintas plataformas.
- **[21] Cordova:** es un entorno de desarrollo de aplicaciones móviles que permite a los programadores construir aplicaciones para dispositivos móviles haciendo uso de CSS3, HTML5 y Javascript en lugar de lidiar con las APIs específicas de cada plataforma, así como encapsular CSS, HTML y código Javascript dependiendo de la plataforma del dispositivo.
- **[22] AngularJS:** es un *framework* de JavaScript y de código abierto mantenido por Google, que se utiliza para crear aplicaciones web. Permite crear aplicaciones web siguiendo el patrón de Modelo-Vista-Controlador de manera mucho más sencilla.
- **[23] Symfony 3:** es un *framework* para construir aplicaciones web con PHP siguiendo el patrón Modelo-Vista-Controlador que separa la lógica del negocio, la lógica del servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.
- **[24] PHP 7:** es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Usándolo, el código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

- **[25] Doctrine:** es un mapeador de objetos-relacional (ORM) escrito en PHP que proporciona una capa de persistencia para objetos PHP. Es una capa de abstracción que se sitúa justo encima de un SGBD (sistema de gestión de bases de datos).
- **[26] JavaScript:** es un lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.
- **Ajax:** es una técnica de desarrollo web para crear aplicaciones interactivas. 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. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.
- **HTML:** hace referencia al lenguaje de marcado para la elaboración de páginas web, es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web.
- **CSS:** es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de las páginas web, e interfaces de usuario escritas en HTML o XHTML.
- **[27] MySQL:** es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo sobre todo para entornos de desarrollo web.
- **[28] Git:** es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

6.2 Herramientas

- **[29] PhpStorm:** es un IDE de programación desarrollado por JetBrains. Es uno de los entornos de programación más completos de la actualidad, permite editar código no sólo del lenguaje de programación PHP como lo indica su nombre, sino que también incluye la posibilidad de editar código HTML y JavaScript. Cuenta además con un depurador, autocompletado y refactorización para PHP y JavaScript.

- **[30] Atom:** es un editor de código de fuente abierta para macOS, Linux, y Windows con soporte para complementos escritos en Node.js, incrustando Git Control, desarrollado por GitHub. Atom es una aplicación de escritorio construida utilizando tecnologías web. La mayor parte de los paquetes tienen licencias de software libre y es construido y mantenido por su comunidad.
- **[31] MySQL Workbench:** es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL.
- **[32] Google Chrome:** es un navegador web desarrollado por Google y compilado con base en varios componentes e infraestructuras de desarrollo de aplicaciones de código abierto, como el motor de renderizado Blink. Está disponible gratuitamente bajo condiciones específicas del software privativo o cerrado. Este explorador cuenta con herramientas para desarrolladores web muy útiles que permiten depurar e incluso realizar cambios en el código en tiempo real.
- **[33] Justinmind Prototyper:** es una herramienta de creación de prototipos de alta fidelidad para aplicaciones web y móviles. Ofrece capacidades que suelen estar presentes en herramientas de diagramación como colocar componentes con arrastrar y soltar, redimensionar, formatear y la exportación / importación de complementos.
- **[34] GitHub:** es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git.

6.3 Servidores

- **[35] XAMPP:** es un paquete de instalación independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.
- **[36] HTTP Apache:** es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft Windows, macOS y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. También presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido.

7. Análisis

En esta sección de la memoria nos centraremos en la etapa de Análisis de nuestra aplicación.

7.1 Actores

Primeramente, hablaremos de los actores identificados durante el análisis, separando cada uno y explicando que papel juegan en nuestra aplicación.

7.1.1 Administrador

El administrador es la persona responsable de gestionar el centro y de realizar las distintas gestiones posibles en el menú de mensajería del centro. Por ende, las funciones que podrá realizar se dividen en dos grupos:

- Gestión del alumnado y cursos del centro: estas funcionalidades son necesarias para poder llevar a cabo aquellas que realmente alcanzan el objetivo final del sistema. Entre ellas se encuentran:
 - Importar automáticamente los datos del centro.
 - Creación de cursos.
 - Registrar alumnos.
 - Asociar y desasociar alumnos de un curso.
 - Asociar y desasociar padres de un alumno.
- Gestión de la mensajería: estas funcionalidades llevan a cabo el objetivo principal del sistema. Permiten enviar distintos tipos de mensajes a los padres de alumnos, tales como:
 - Circulares generales que contienen información relevante sobre algún aspecto específico del centro.
 - Autorizaciones de actividades para el alumnado, que permiten a los padres firmarlas desde su teléfono móvil. También se permite visualizar un listado de aquellos alumnos que hayan sido autorizados y aquellos que no para una determinada actividad.
 - Encuestas para que los padres formen parte de las tomas de decisiones que el centro considere oportunas. Además, se podrán visualizar los resultados de dichas encuestas.

7.1.2 Padre/Madre

En este rol se engloba a los padres/madres/tutores legales del alumnado. Dichos actores podrán realizar las siguientes acciones:

- Ver el contenido de los distintos tipos de mensajes que el centro les ha enviado (circulares generales, autorizaciones de actividades y encuestas).
- Filtrar dichos mensajes por su asunto.
- Firmar las autorizaciones, haciendo uso de un código de seguridad establecido por el padre/madre/tutor legal.
- Responder a las encuestas enviadas por el centro.
- Establecer qué centros pueden tener acceso a su información personal.
- Ver y desasociar a sus hijos previamente asociados por el centro.

Como se puede deducir, este rol tiene un papel principal en nuestra aplicación, ya que serán el usuario final de la aplicación móvil. Debe incluirse, por ende, una fase de registro en el sistema por parte de los usuarios pertenecientes a este rol, de lo contrario la otra alternativa es que los registre el centro, pudiéndose dar el caso de que los padres no tuvieran la aplicación instalada y el centro pensara de manera errática que los mensajes están llegando a sus destinatarios cuando no lo estarían haciendo.

7.2 Requisitos

Para realizar una representación clara y sencilla de los requisitos funcionales del sistema hemos decidido hacer uso de una de las herramientas vistas durante el grado, **[37] UML**.

La razón de haber escogido UML es que, además de la experiencia que previamente hemos adquirido con este lenguaje de modelado, es el más utilizado en la actualidad en lo que respecta a sistemas software. Y una de sus grandes ventajas es que está reconocido como estándar por la ISO (Organización Internacional de Normalización), lo que permite que independientemente del idioma o del país, las ideas estarán plasmadas de la misma manera y serán reconocidas por cualquier otra persona que esté familiarizada con UML.

El objetivo de este Trabajo de Fin de Grado solo abarca el desarrollo de la aplicación móvil utilizada por los padres, por ello, creemos adecuado centrarnos únicamente en los casos de uso de dichos actores.

Los casos de uso relacionados con la administración del centro estarán explicados en detalle en el Trabajo de Fin de Grado titulado: *“Análisis, Diseño e Implementación de un Backend para la Comunicación entre Centros Educativos y Padres de Alumnos”*.

7.2.1 Casos de uso

En esta subsección, mostraremos el diagrama de casos de uso realizado para tener una representación clara y concisa de los requisitos funcionales identificados en la fase de análisis, dichos diagramas están separados en diferentes imágenes para facilitar su lectura en un documento de este tipo.

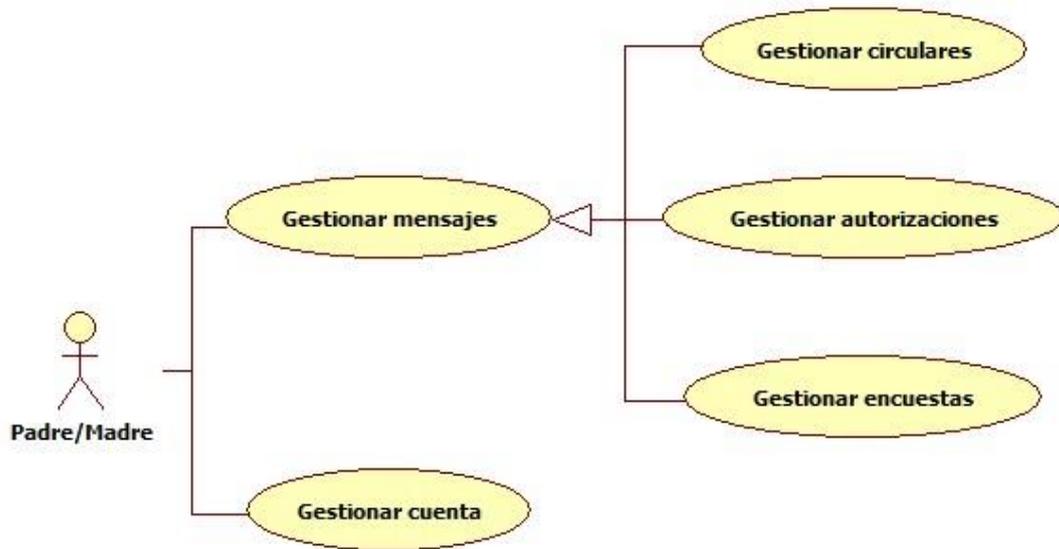


Ilustración 2: Diagrama de casos de uso (Resumen)

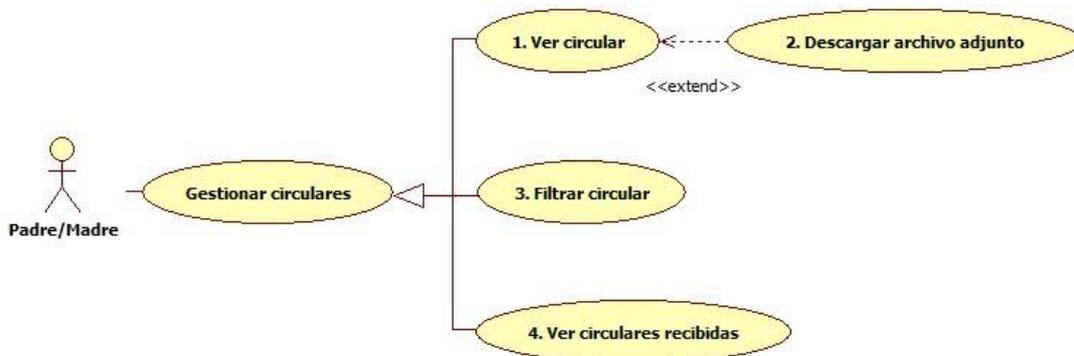


Ilustración 3: Diagrama de casos de uso (Gestionar circulares)

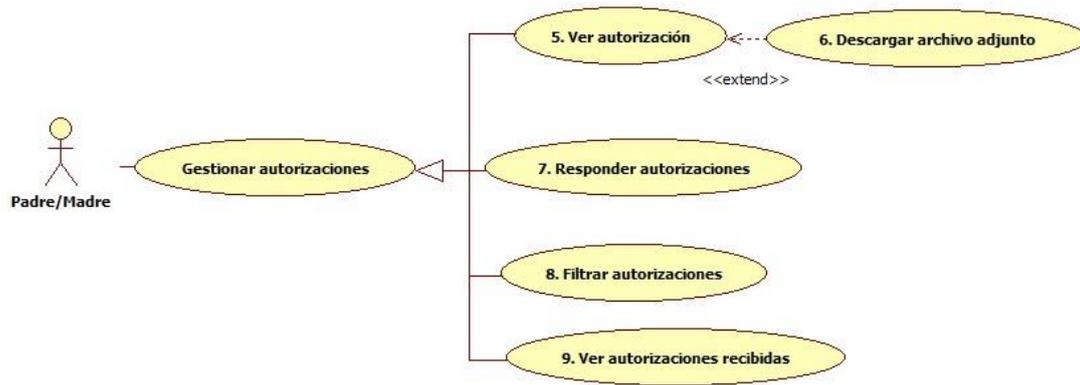


Ilustración 4: Diagrama de casos de uso (Gestionar autorizaciones)

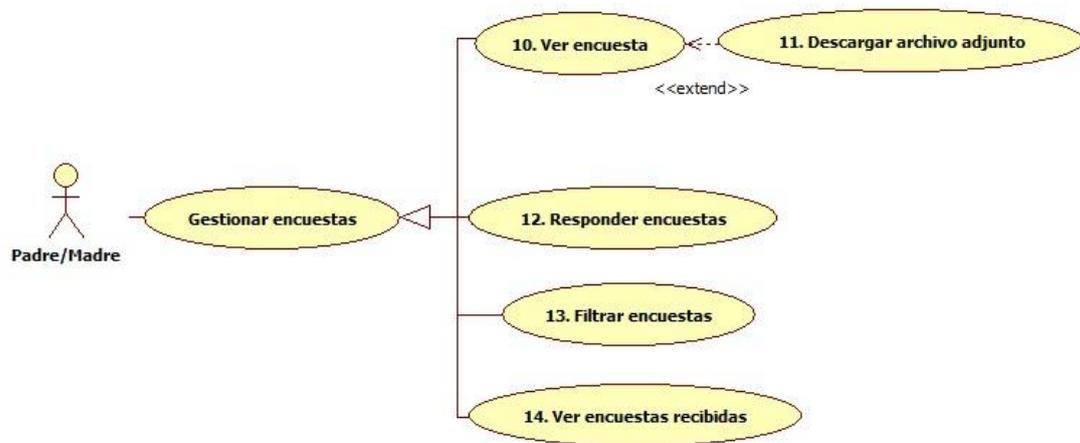


Ilustración 5: Diagrama de casos de uso (Gestionar encuestas)

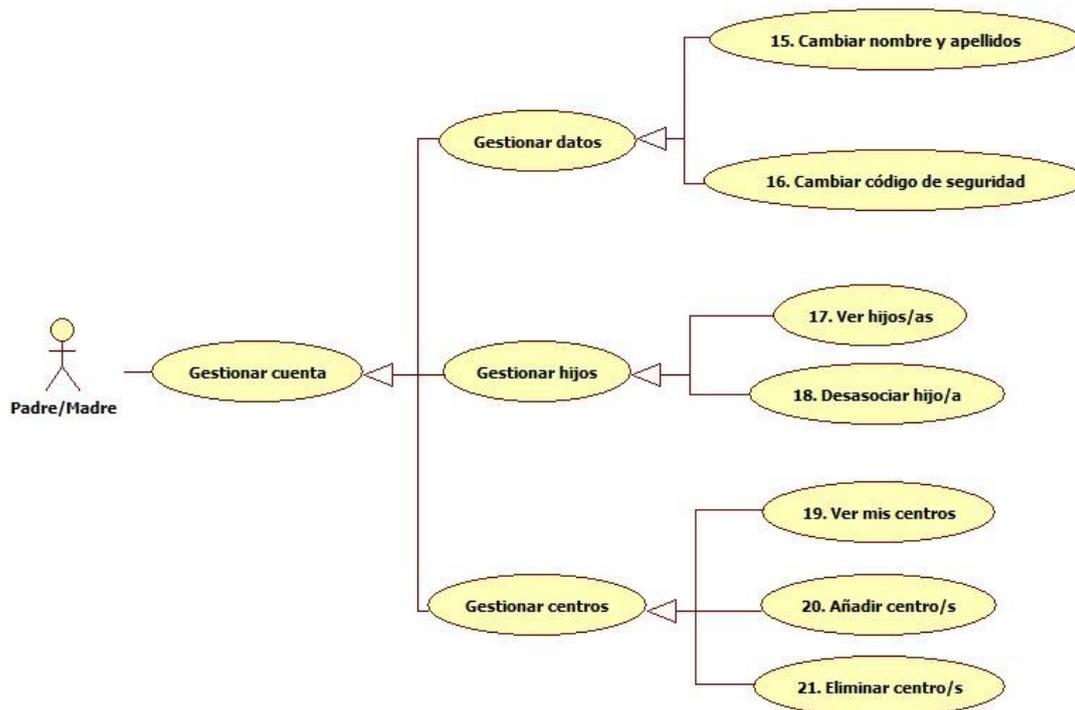


Ilustración 6: Diagrama de casos de uso (Gestionar cuenta)

Bajo estas líneas, especificaremos en la *Tabla 1* cada uno de los casos de uso representados anteriormente en el diagrama, acompañados de una breve descripción de los mismos.

ID	Actor	Descripción
1	Padre/Madre	Se muestra el contenido de la circular seleccionada.
2	Padre/Madre	Se descarga el documento adjunto a la circular que se está visualizando.
3	Padre/Madre	Se muestra el listado de circulares que coinciden con el filtro introducido.
4	Padre/Madre	Se muestra un listado de todas las circulares destinadas al usuario.
5	Padre/Madre	Se muestra el contenido de la autorización seleccionada.
6	Padre/Madre	Se descarga el documento adjunto a la autorización que se está visualizando.
7	Padre/Madre	Se autoriza o desautoriza al hijo/hija de la autorización mostrada a realizar la actividad propuesta por el centro.
8	Padre/Madre	Se muestra el listado de autorizaciones que coinciden con el filtro introducido.
9	Padre/Madre	Se muestra un listado de todas las autorizaciones destinadas al usuario.
10	Padre/Madre	Se muestra el contenido de la encuesta seleccionada.
11	Padre/Madre	Se descarga el documento adjunto a la encuesta que se está visualizando.

12	Padre/Madre	Se responde a la encuesta propuesta por el centro.
13	Padre/Madre	Se muestra el listado de encuestas que coinciden con el filtro introducido.
14	Padre/Madre	Se muestra un listado de todas las encuestas destinadas al usuario.
15	Padre/Madre	Se cambia el Nombre y Apellidos actuales del usuario por el introducido.
16	Padre/Madre	Se cambia el código de seguridad actual del usuario por el introducido.
17	Padre/Madre	Se muestra un listado de los hijos/as asociados al usuario.
18	Padre/Madre	Se desasocia el hijo/ha seleccionado del usuario.
19	Padre/Madre	Se muestra un listado de los centros asociados al usuario.
20	Padre/Madre	Se añade el centro seleccionado a la lista de centros permitidos del usuario.
21	Padre/Madre	Se elimina el centro seleccionado de la lista de centros permitidos del usuario.

Tabla 1: Resumen de casos de uso

7.2.2 Especificación de casos de uso

Bajo estas líneas adjuntamos una serie de casos de uso, los cuáles consideramos los más relevantes de la aplicación, especificados siguiendo un modelo tabulado:

CASO DE USO	7	RESPONDER AUTORIZACIONES	
Descripción	El padre/madre autoriza o desautoriza a su hijo/a para asistir a una actividad.		
Actores	Padre/Madre.		
Precondiciones	El padre/madre debe haber iniciado sesión en el sistema.		
Flujo normal	Paso	Acción	
	1	Se accede a una de las autorizaciones de actividades.	
	2	Se autoriza la actividad.	
	3	Se introduce el código de seguridad del usuario.	
	4	Se responde a la autorización.	
Postcondiciones	Se modifica la lista de autorizados para dicha actividad, añadiendo al hijo/a del padre/madre correspondiente.		
Variaciones	Paso	Acción	

	2 A	Se desautoriza la actividad.	
Extensiones	Paso	Condición	Caso de Uso
Excepciones	3	El código de seguridad introducido no es correcto.	
Observaciones			

Tabla 2: Especificaciones de casos de uso (Responder autorizaciones)

CASO DE USO	12	RESPONDER ENCUESTAS	
Descripción	El padre/madre responde a una encuesta enviada por el centro para un tema determinado.		
Actores	Padre/Madre.		
Precondiciones	El padre/madre debe haber iniciado sesión en el sistema.		
Flujo normal	Paso	Acción	
	1	Se accede a una de las encuestas.	
	2	Se selecciona la respuesta/s deseada entre las mostradas.	
	3	Se introduce el código de seguridad del usuario.	
	4	Se responde a la encuesta.	
Postcondiciones	Se registra el voto/s del padre/madre en la encuesta.		
Variaciones	Paso	Acción	
Extensiones	Paso	Condición	Caso de Uso
Excepciones	4	El código de seguridad introducido no es correcto.	
Observaciones			

Tabla 3: Especificaciones de casos de uso (Responder encuestas)

CASO DE USO	16	CAMBIAR CÓDIGO DE SEGURIDAD	
Descripción	El padre/madre cambia el código de seguridad que estableció al registrarse/iniciar sesión.		
Actores	Padre/Madre.		
Precondiciones	El padre/madre debe haber iniciado sesión en el sistema.		
Flujo normal	Paso	Acción	
	1	Se accede a la ventana de "Mis datos".	
	2	Se introduce el código actual en el formulario del código de seguridad.	
	3	Se introduce el nuevo código de seguridad.	
	4	Se introduce la repetición del código de seguridad.	
	5	Se realiza el cambio del código de seguridad.	
Postcondiciones	Se establece el nuevo código y se limpia el formulario.		
Variaciones	Paso	Acción	
Extensiones	Paso	Condición	Caso de Uso
Excepciones	5	El código de seguridad actual no es correcto.	
	5	El nuevo código de seguridad no coincide.	
Observaciones			

Tabla 4: Especificaciones de casos de uso (Cambiar código de seguridad)

CASO DE USO	20	AÑADIR CENTRO/S	
Descripción	El padre/madre añade un centro/s a su lista de centros permitidos que pueden consultar su información personal.		
Actores	Padre/Madre.		
Precondiciones	El padre/madre debe haber iniciado sesión en el sistema.		
Flujo normal	Paso	Acción	

	1	Se accede a la ventana de “Mis Centros”.	
	2	Se seleccionan aquellos centros del listado que se quieren añadir.	
	3	Se envían los centros nuevos y se añaden a la lista ya existente.	
Postcondiciones	Se registra el voto del padre/madre en la encuesta.		
Variaciones	Paso	Acción	
Extensiones	Paso	Condición	Caso de Uso
Excepciones			
Observaciones			

Tabla 5: Especificaciones de casos de uso (Añadir centro/s)

8. Diseño

En este capítulo vamos a describir el diseño al que hemos llegado a posteriori de realizar el análisis, comenzando por el diseño de la arquitectura del sistema.

8.1 Diseño de la arquitectura del sistema

La arquitectura de sistema escogida para el proyecto ha sido la arquitectura de tipo multi-nivel que procederemos a definir a continuación.

[38] La arquitectura multi-nivel o por capas: en dicha arquitectura a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).

El más utilizado actualmente es el diseño en tres niveles (o en tres capas):

- Capa de presentación: la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.
- Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
- Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

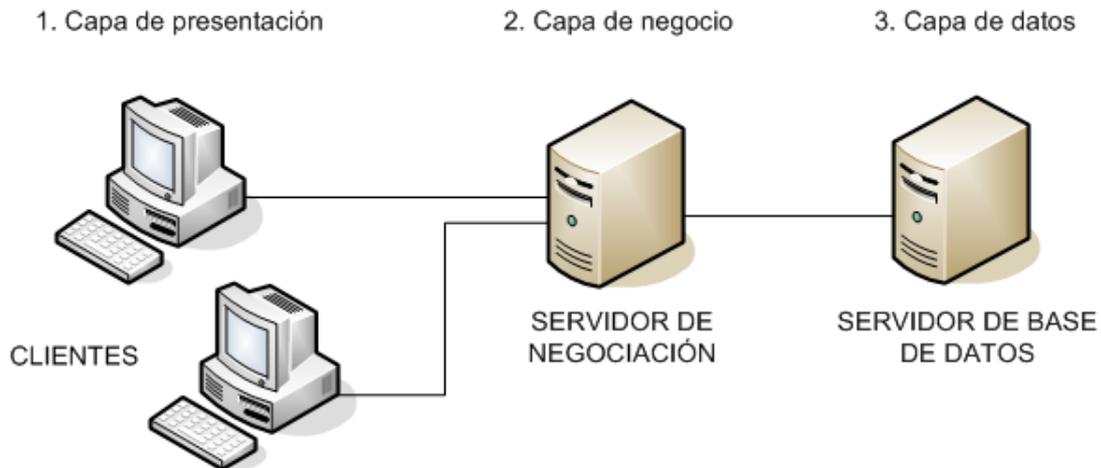


Ilustración 7: Instalación típica de una arquitectura multi-nivel

Sobre estas líneas, en la *Ilustración 7*, podemos apreciar que, la capa de presentación reside en el lado de los clientes, la de negocio suele incluirse en un servidor de negociación y por último la capa de datos en el servidor que alberga la base de datos.

Hemos decidido escoger esta arquitectura porque nos proporciona las siguientes ventajas:

- **Centralización del control:** los accesos, recursos e integridad de los datos son controlados por el servidor, de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de actualizar dichos datos u otros recursos.
- **Escalabilidad:** se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento.
- **Fácil mantenimiento:** al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- **Simplifica la seguridad:** no es necesario aplicar políticas de seguridad complejas en las aplicaciones del cliente, ya que la lógica de negocio y los datos están alojados en un único servidor. Además, Existen tecnologías, suficientemente desarrolladas, diseñadas para esta arquitectura que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de uso.

En nuestro caso particular, las aplicaciones del cliente sólo tienen la capa de presentación y una capa de red que permite conectarse al servidor, y el *backend* de la aplicación web y la API RESTful residen en un servidor web, que sustituye a las capas de negocio, de acceso a datos y de datos. Dicha arquitectura puede observarse en detalle en la *Ilustración 8* que acompaña a estas líneas.

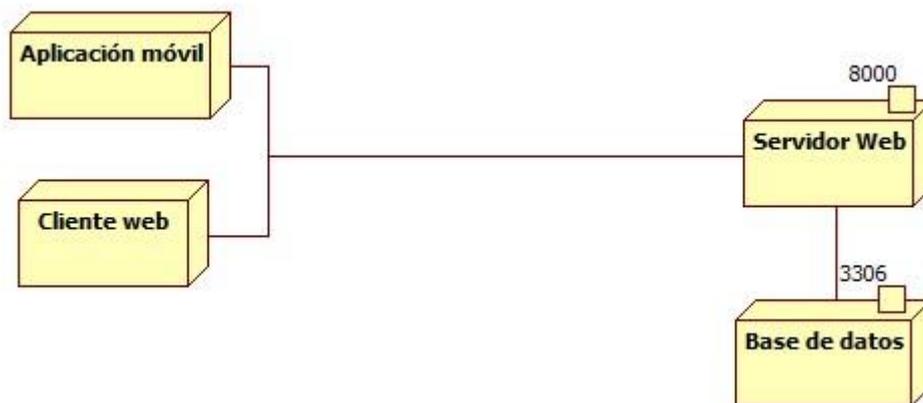


Ilustración 8: Diagrama de despliegue

8.1.1 Frontend

En el *frontend* del sistema contamos con dos aplicaciones distintas:

- **Aplicación móvil:** en la cual, el actor principal es el padre/madre del alumno. Su funcionalidad conlleva permitir ver los mensajes enviados por la administración de los centros de sus hijos, así como responder a autorizaciones y encuestas.
- **Aplicación web:** en ella, el actor principal es el administrador del centro. Su funcionalidad contempla permitir enviar mensajes a los padres de los alumnos, y poder visualizar qué alumnos han sido autorizados para una determinada autorización por sus padres y ver los resultados de una encuesta realizada a los padres de los alumnos.

El uso de la aplicación móvil se restringirá a cualquier dispositivo móvil que los padres tengan, dado el carácter multiplataforma de nuestra aplicación. Por otro lado, el uso de la aplicación web estará restringido a un equipo de la administración que el centro designe para el uso de nuestro sistema.

8.1.2 Backend

Como hemos descrito y se podía observar en la *Ilustración 8* el *backend* de nuestro sistema se encuentra alojado en un servidor, que ejecuta un servidor web Apache que incorpora tanto la API RESTful que necesita nuestra aplicación móvil para funcionar, como el *backend* de la aplicación web.

Tanto el *backend* de la aplicación móvil como el de la aplicación web están desarrollados e implementados utilizando las mismas tecnologías (Symfony3, PHP7 y Doctrine) de tal manera que ambas utilicen la misma capa de acceso a datos y evitemos un sistema repetitivo.

8.2 Diseño de la Base de Datos

Como se mencionó en el capítulo destinado al a especificación de herramientas y tecnologías, hemos utilizado el sistema de gestión de bases de datos relacional MySQL.

La estructura de dicha base de datos puede observarse en la *Ilustración 9* adjunta en la siguiente página.

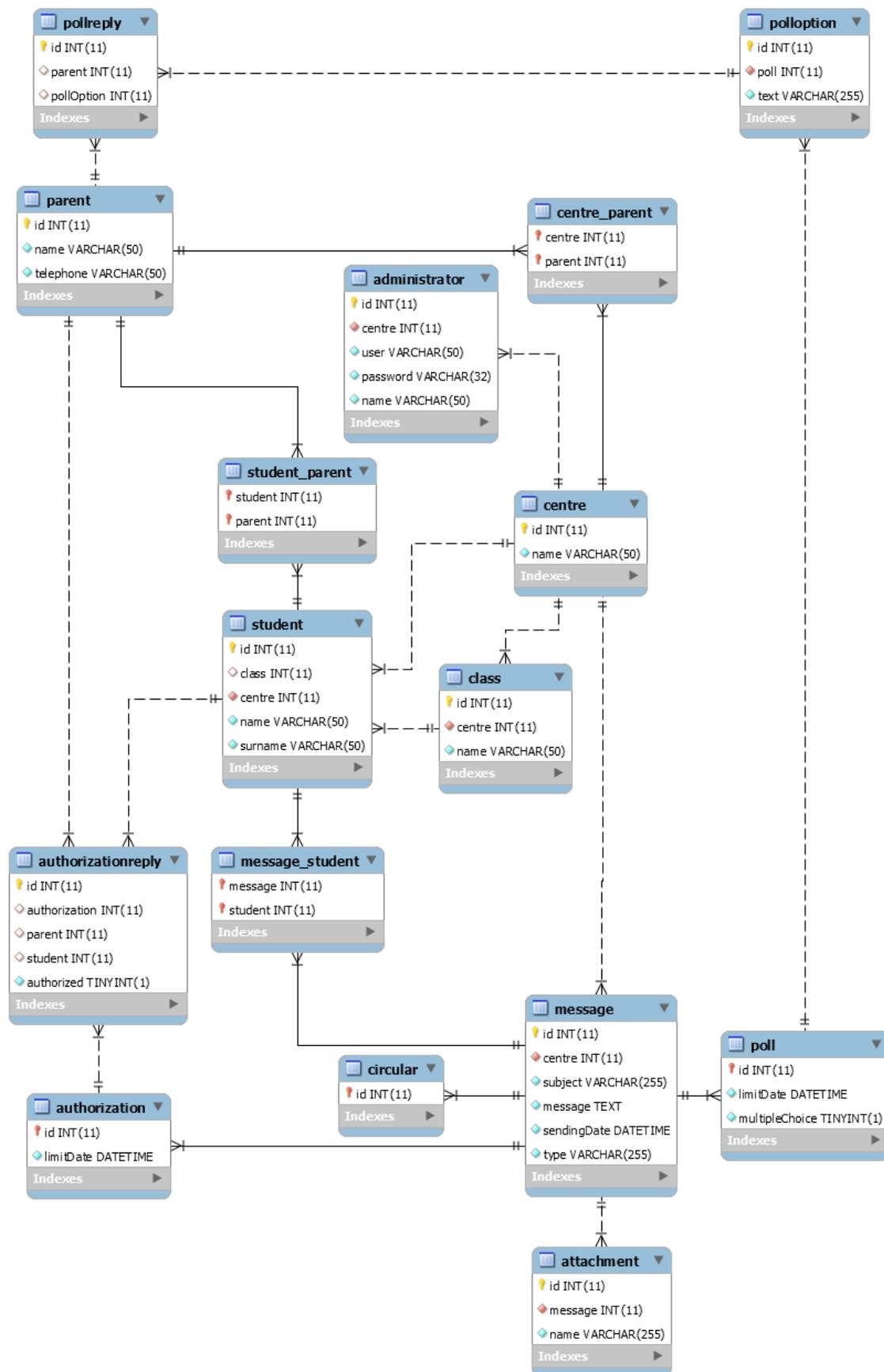


Ilustración 9: Diseño de la Base de Datos

Durante el diseño de la misma, decidimos que las estructuras de las tablas *circular*, *poll* y *authorization* deberían implementarse bajo el uso de la herencia, para evitar repetir campos sin un propósito específico. De esta manera, las tres tablas mencionadas, heredan de la tabla *message* la cual contiene aquellos campos comunes a todos los tipos de mensajes. Esta separación también nos permite hacer clara la distinción entre las relaciones necesarias entre varias tablas:

- Una *pollOption* sólo puede estar relacionada con una *poll*.
- Una *authorizationReply* sólo puede estar relacionada con una *authorization*.
- Un *attachment* puede estar relacionado con cualquier tipo de *message*.
- Un *student* puede estar relacionado con cualquier tipo de *message*.

Actualmente el sistema sólo permite adjuntar un archivo por mensaje, no obstante, hemos preferido realizar una tabla de *attachment* enlazada a *message* por si en un futuro se quisiera implementar permitir enviar más de un archivo adjunto, sin requerir una modificación severa de la Base de Datos.

8.2.1 Contenido de las tablas de la Base de Datos

Ahora pasaremos a describir brevemente el contenido de las tablas siguiendo la *Tabla 6*.

Tabla	Contenido
<i>administrator</i>	Administradores que usarán la aplicación web.
<i>authorizaton</i>	Autorizaciones enviadas por los centros.
<i>authorizationReply</i>	Respuestas enviadas por los padres a las a las autorizaciones enviadas por los centros.
<i>attachment</i>	Nombre de los archivos adjuntos.
<i>centre</i>	Centros educativos registrados en el sistema.
<i>centre_parent</i>	Relaciona a los padres con los centros a los que permiten ver su información personal.
<i>circular</i>	Circulares enviadas por los centros.
<i>class</i>	Cursos pertenecientes a los centros.
<i>message</i>	Mensajes enviados (campos comunes) por los centros.
<i>message_student</i>	Relaciona a los mensajes con alumnos.
<i>parent</i>	Padres registrados en el sistema que usarán la aplicación móvil.
<i>poll</i>	Encuestas enviadas por los centros.
<i>pollOption</i>	Opciones asociadas a cada encuesta.

<i>pollReply</i>	Respuestas enviadas por los padres a las encuestas enviadas por los centros.
<i>student</i>	Estudiantes registrados en cada centro.
<i>student_parent</i>	Relaciona a los padres con los alumnos.

Tabla 6: Contenido de las tablas de la Base de Datos

8.2.2 Almacenamiento de archivos adjuntos

Para realizar el almacenamiento de archivos, en lugar de guardarlos directamente en la base de datos, hemos tomado la decisión de guardarlos directamente en el sistema de archivos del servidor. **[39] Las razones por las que hemos llegado a dicha decisión** son estas:

- Guardar los archivos directamente en la base de datos implica realizar un código más complejo para guardar los mismos, dado que hay que indicar los distintos tipos de datos y se requiere realizar una capa de software distinta para cada tipo de dato que se desee adjuntar.
- Por la razón anterior, resulta más complicado acceder a dichos archivos desde la aplicación móvil y la aplicación web.
- El nivel de conocimiento requerido para mantener una base de datos es directamente proporcional a su tamaño y complejidad, ya que, a más complejidad y más tamaño, el uso de migraciones y copias de seguridad se convierte en una tarea mucho más difícil.
- No podríamos aprovechar las posibilidades de utilizar el almacenamiento en la nube en un futuro. Usando un sistema de archivos sería relativamente sencillo dar el salto a dichas opciones.

El mecanismo que se utiliza para el guardado de archivos en el servidor de la base de datos consiste en guardar los mismos en una carpeta del sistema de ficheros, usando como nombre el *id* con el que se insertará en la tabla *attachment*. En dicha tabla se adjunta también en la inserción un campo *name* que contiene el nombre original del fichero, incluida su extensión.

8.3 Diseño Arquitectónico

El *framework* Ionic en su versión 1, está basado en el patrón arquitectónico [40] **Modelo-Vista-Controlador (MVC)**.

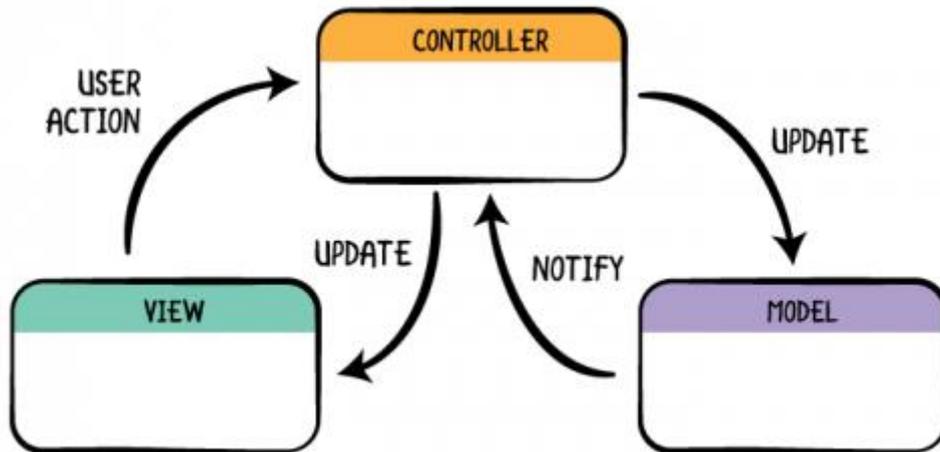


Ilustración 10: Diagrama del patrón Modelo-Vista-Controlador

En la *Ilustración 10* podemos observar cuál es la estructura base de dicho patrón que pasaremos a explicar en detalle a continuación:

- **Modelo:** Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la Vista aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al Modelo a través del Controlador.
- **Vista:** Presenta el Modelo (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho Modelo la información que debe representar como salida.
- **Controlador:** Responde a eventos (usualmente acciones del usuario) e invoca peticiones al Modelo cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su Vista asociada si se solicita un cambio en la forma en que se presenta el Modelo (por ejemplo, desplazamiento por un documento o por los diferentes registros de una base de datos), por tanto, se podría decir que el Controlador hace de intermediario entre la Vista y el Modelo.

8.4 Diseño de la interfaz de usuario

La interfaz de usuario es el puente que une al usuario y el software, y su importancia es mayor de la que solemos pensar. El buen diseño en las mismas garantiza de

sobremanera cierta parte del éxito de una aplicación. Hay muchas aplicaciones excelentes en cuanto a funcionalidades, sin embargo, debido a lo complejo o dificultoso que se vuelve su uso debido a la interfaz, el usuario se siente rechazado por el software y termina por abandonarlo, convirtiéndose en un producto no satisfactorio. Sin embargo, existen en el mercado muchísimas aplicaciones que, sin contar las mejores implementaciones atrapan a millones de usuarios porque les hacen sentir cómodos, que las cosas se realizan sin dificultad, no tienen que buscar entre un millón de opciones y la información se muestra de manera clara y precisa, sin excesos ni adornos innecesarios. Por ello hemos realizado una interfaz de usuario móvil que cumpla con los principios de diseño definidos por **[41] Jeff Johnson** en el libro ***GUI Bloopers 2.0***.

Principio 1

- *Considerar como más importante el usuario y las tareas que debe realizar anteponiéndose a la tecnología.*
- *Comprender al usuario, sus tareas y el entorno de software en las que se desarrolla.*

Para cumplir este principio, en nuestra aplicación hemos tenido en cuenta el tipo de público al que va dirigido, esto que implica muchos de los padres/madres normalmente no están familiarizados con las últimas interfaces de usuario y sólo les interesa poder ver el contenido de los mensajes del centro y responder a ellos de manera sencilla y clara. Por ello se han evitado menús deslizantes, elementos dinámicos, pantallas flotantes y gestos multitáctiles con el fin de evitar acciones erráticas y mantener una interfaz clara.

Principio 2

- *El primer paso en el diseño de una interfaz es desarrollar un modelo conceptual de la interacción y las tareas. Los aspectos de presentación deben abordarse posteriormente.*

Para este principio, previo al diseño final de la interfaz de usuario y a la capa de presentación realizamos ciertos esbozos de cómo sería la interfaz de usuario de los padres, así como prototipos sencillos haciendo uso del producto software Justinmind Prototyper.

Principio 3

- *Centrarse en la visión que tiene el usuario respecto a la tarea.*

- *Esforzarse por la naturalidad, ideas y vocabulario de él.*
- *Las cuestiones internas del programa (de la tecnología en general) deben permanecer internas.*
- *Encontrar un equilibrio entre prestaciones y complejidad.*

Para conseguir cumplimentar con estos puntos, pedimos segundas opiniones a padres y madres conocidos sobre qué interfaz de usuario les parecía más clara o más concisa, y comprobar si nuestro diseño casaba con la idea que ellos tenían de cómo debían realizarse ciertas acciones. Siempre hemos intentado acercarnos al vocabulario formal, con el que padres y madres suelen estar familiarizados y evitar mostrar cualquier tipo de información de los procesos internos de la aplicación. Hemos intentado que el usuario disponga de funcionalidades que puedan resultarle útiles, pero sólo para realizar de manera más sencilla las tareas principales, sin aumentar la complejidad de la aplicación.

Principio 4

- *Diseña pensando en las cosas más comunes y frecuentes, no en las extraordinarias.*
- *Haz que los casos comunes y frecuentes sean muy fáciles de realizar.*

Debida a la naturaleza de la aplicación, hemos centrado nuestros esfuerzos en que las acciones principales no requieran muchos cambios de pantalla para realizarse, ver el contenido de los mensajes no debería estar más allá de dos vistas, la del listado del mensaje y la de su contenido. Y para responder a un mensaje no es necesario salir de la pantalla actual. De esta manera y muchas otras, hemos intentado simplificar las tareas más comunes de la aplicación, haciendo que la navegación sea rápida y eficiente.

Principio 5

- *No le compliques la vida al usuario.*
- *No le obligues a razonar en demasía.*

Se ha conseguido no mostrar pantallas innecesarias al usuario y no mostrar mensajes de error indescifrables, si ocurre algún error durante la ejecución de alguna funcionalidad, especifica claramente el porqué ha ocurrido y que debe hacer el usuario para ponerle solución.

Principio 6

- *Facilita el aprendizaje.*
- *Consistencia: significado de los controles deducido a partir de ellos mismos o pistas muy fáciles, nunca valiéndonos de nuestra memoria (poner el conocimiento de uso de las cosas en el mundo no en nuestro cerebro).*
- *Evita las consecuencias de los errores, el usuario se atreverá a explorar cosas nuevas, ya que no suponen riesgo.*

La configuración inicial de la aplicación aclara cómo se debe seleccionar un centro, qué datos realmente se conocerán del usuario final y el avance por dicha configuración siempre está aclarado haciendo uso de grandes botones que indican la acción que se está realizando en cada etapa. Partiendo de que el uso de los controles en la aplicación está limitado a tocar y realizar un desplazamiento por el contenido, dichos controles son homogéneos en toda la aplicación. Además, ninguno de los errores que se muestran en la aplicación conlleva un efecto negativo en el usuario y en su experiencia.

Principio 7

- *Proporciona información al usuario, no simplemente soltarle datos.*
- *“La pantalla le pertenece al usuario” es una forma de no alterar ese espacio en el que él se mueve. Respeta su territorio y no cambies las cosas bruscamente (inercia de la pantalla).*

Cuando el usuario realiza alguna actividad que debería de tener una retroalimentación, se ha implementado que tenga algún tipo de respuesta gráfica. Por ejemplo, cuando el usuario autoriza o desautoriza una autorización, se actualiza la interfaz (no se cambia bruscamente) y se muestra un mensaje indicando el estado de la autorización, así como el único botón que permite “Desautorizar” en caso de haber autorizado, y “Autorizar” en el caso contrario.

Principio 8

- *Diseña para la interactividad entre la aplicación y el usuario.*
- *Respetar el tiempo del usuario.*

El tiempo del usuario es vital, por ello, se ha diseñado la aplicación para que la navegación por la misma sea rápida, con el menor tiempo de espera y mostrando el menor número de pantallas posible sin repercutir en la sencillez de uso.

Principio 9

- *Prueba los diseños con usuarios reales, es decir: Test y Verificación.*
- *Corrige los fallos.*

Además de las pruebas de la aplicación realizadas con el Tutor para validar los requisitos funcionales, se hicieron pruebas con usuarios potenciales para verificar si el diseño cumplía con su cometido y si todas las funcionalidades se ejecutaban correctamente. Gracias a esto, se descubrieron algunos errores que no se habían identificado durante las pruebas del desarrollo y se corrigieron debidamente.

9. Desarrollo

9.1 Ionic

Ionic es un SDK móvil atractivo, gratuito y de código abierto para desarrollar aplicaciones web nativas y progresivas con facilidad. Las principales que obtenemos de usar Ionic como *framework* principal de desarrollo y por las que nos hemos decantado para hacerlo son:

- El desarrollo principal se realiza en HTML junto con CSS y JS, lenguajes muy extendidos por la comunidad de desarrolladores, con lo que la implantación de esta herramienta en la empresa facilitará el desarrollo de proyectos de la forma más efectiva aun cuando la plantilla de desarrolladores contenga nuevas incorporaciones.
- Si ya contamos con una web app que queremos convertir en aplicación móvil, en la mayoría de los casos habremos hecho uso de JavaScript, por lo que el código es reutilizable.
- Para el caso de aplicaciones híbridas, tendremos con un único proceso de desarrollo e implementación, una app para Android, iOS y web.
- AngularJS: Ionic trabaja perfectamente con AngularJS. Dando lugar a una arquitectura robusta para el desarrollo de apps. Podremos crear apps móviles ricas y robustas, para subir desde la plataforma a tu tienda de aplicaciones a escoger.
- Es fácil de entender: no tenemos que complicarnos en exceso utilizando Ionic, es bastante sencillo de entender. Si ya hemos programado alguna aplicación para iOS o Android, seguro que nos entenderemos bien con el SDK, pues con Ionic también. Permite desarrollar un código una vez y reutilizarlo las veces que quieras ya que desde una única fuente podremos llegar a las plataformas que soporta este *framework* (Android e iOS).
- Pulcro: Ionic es moderno y está diseñado para trabajar con lo más actual, con un diseño limpio y pulcro. Los componentes son atractivos, la tipografía, etc.
- Crea, construye, prueba y compila: con Ionic podremos crear, construir, y compilar aplicaciones en cualquier plataforma, todo con un solo comando. Por eso se considera un potente CLI.
- Funciona rápido: si no disfrutamos de excesiva paciencia, nos gustará Ionic. Está hecho para ser rápido.
- Ionic Creator: una de las ventajas de Ionic, es Ionic Creator. Básicamente nos permite crear las Interfaces sin tener que meter el código de manera tradicional. Podremos crear la parte gráfica fácil sin apenas escribir código para ello.
- Construido y mantenido por diseñadores y desarrolladores.

9.2 Estructura de la aplicación

Las aplicaciones de Ionic están construidas haciendo uso de Cordova. Cordova es un conjunto de HTML/CSS/JavaScript empaquetados que permiten ejecutar el código en dispositivos móviles y en escritorio, y nos provee de una arquitectura de complementos para acceder a funcionalidades nativas más allá de lo que podría hacer un código JavaScript en una aplicación web. Por esto, las aplicaciones de Ionic tienen **[42] la estructura de archivos de Cordova**.

En la *Ilustración 11* que acompaña a estas líneas podemos observar la estructura principal de nuestro proyecto, de la cual procederemos a detallar a continuación el contenido de los directorios más relevantes.

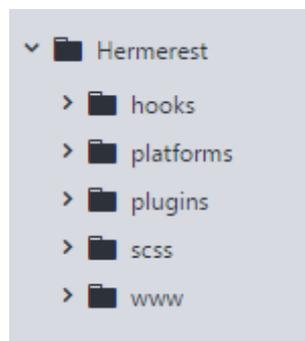


Ilustración 11: Estructura de archivos de la aplicación

- *hooks*: se utiliza para almacenar las acciones personalizadas que queremos que se realicen en nuestra aplicación durante el proceso de desarrollo de Cordova. Puede sernos útil en grandes proyectos que requieran de procesos automatizados de ejecución y modificación de código, pero normalmente no lo usamos.
- *platforms*: contiene nuestros proyectos de iOS y Android. En general, no necesitamos trabajar en esos directorios a no ser que estemos *hacking* nativo personalizado o llevando nuestras aplicaciones a producción.
- *plugins*: es el directorio donde Cordova almacena los plugins añadidos al proyecto haciendo uso del comando:

```
ionic cordova plugin add {plugin}
```

- *scss*: almacena el archivo SASS para nuestra aplicación. El uso de SASS es opcional en Ionic, pero de por sí Ionic está construido con SCSS, por lo que hay muchos estilos por defecto que se pueden cambiar y personalizar rápidamente sin añadir ninguna sobre-escritura a los CSS de la aplicación.
- *www*: es el directorio donde desarrollamos nuestra aplicación.

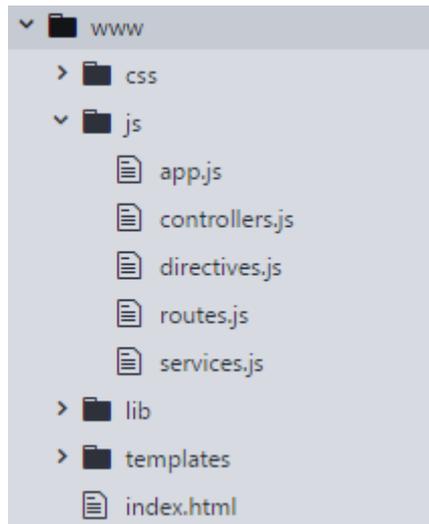


Ilustración 12: Estructura del directorio *www*

En la *Ilustración 12* podemos observar el contenido del directorio *www* del cuál mencionaremos también qué contienen sus subdirectorios.

- *css*: contiene o bien el archivo CSS especificado para nuestra aplicación, o bien contiene el archivo de salida generado por nuestro SCSS, que debemos usar junto al resto de archivos CSS que deseamos utilizar. La carpeta *css* está enlazada a nuestro proyecto bajo la etiqueta `<link>` en el archivo *index.html*.
- *js*: por defecto, en este directorio se incluyen los siguientes archivos enumerados a continuación.
 - *app.js*: contiene nuestros métodos de ejecución y configuración de Angular, en él se definen las variables de entorno, por ejemplo: que tipo de estilo de pestañas usar si de iOS o Android.
 - *controllers.js*: contiene nuestros controladores de Angular para los estados que así lo requieran.
 - *directives.js*: contiene las directivas de Angular personalizadas que deseemos utilizar.
 - *routes.js*: define el enrutamiento de nuestra aplicación, así como los distintos estados que existen en la misma y la vista que requiere mostrar cada uno de ellos si la hubiera.
 - *services.js*: contiene los servicios personalizados de Angular que deseemos usar, como, por ejemplo: una *factory* que nos permite realizar peticiones Ajax haciendo uso del módulo `$http` de manera sencilla.
- *lib*: contiene las librerías de Ionic y otras librerías que hayamos instalado, como aquellas instaladas con *Bower*: por ejemplo, en nuestro caso, *angular-moment* utilizó este instalador de paquetes, por lo que el contenido de la librería yace en este directorio.
- *templates*: en este directorio se albergan los archivos de vistas de nuestra aplicación, enlazados a estados en el archivo de *routes.js*.

9.3 API Restful

Para el desarrollo de nuestra API Restful buscamos algún tipo de herramienta que nos permitiera crear un servicio web de manera sencilla partiendo del proyecto del *backend* que ya contaba con una capa de acceso a datos hecha, explicada en detalle en el trabajo de fin de grado correspondiente al mismo, para así poder ahorrar repetir trabajo innecesario. De entre todas las opciones acabamos decantándonos por **[43] Friends of Symfony**.

9.3.1 Friends of Symfony

Este paquete nos provee de varias herramientas para desarrollar de manera rápida APIs RESTful y aplicaciones con Symfony. Entre sus características destacan:

- Una capa de vista para habilitar la salida y formatear controladores.
- Un **[44] cargador personalizado de rutas** que genera *urls* que siguen los convenios REST.
- Acepta la negociación del formato de la cabecera incluyendo la manipulación de la misma para crear tipos personalizados.
- Decodifica en RESTful el cuerpo de las solicitudes HTTP y de las cabeceras de aceptación.
- Crea excepciones en el controlador para enviar códigos de estado HTTP apropiados.

9.3.2 Configuración de la API

Para la configuración de la API requerimos de la instalación de los paquetes Friends of Symfony y Nelmio-Cors (este último permite que la API sea accesible fuera de la máquina local) y para ello deben ser correctamente configurados en el proyecto del *backend* del sistema.

Primeramente, debemos instalar ambos componentes usando *composer*, desde consola ejecutamos en el directorio del *backend* del sistema:

```
composer require friendsofsymfony/rest-bundle
```

```
composer require nelmio/cors-bundle
```

Una vez instalados los paquetes, debemos editar el archivo localizado en la ruta */Hermerest/App/AppKernel.php* añadiendo las siguientes líneas en la sección de *bundles* para poder registrar los paquetes recién instalados e indicar que ahora forman parte de nuestro proyecto.

```
new FOS\RestBundle\FOSRestBundle(),
```

```
new Nelmio\CorsBundle\NelmioCorsBundle(),
```

A continuación, debemos editar la configuración del proyecto para incorporar la configuración propia de dichos paquetes, añadiendo las líneas que siguen al archivo */Hermerest/App/config/config.yml* al final del mismo:

```
# Nelmio CORS Configuration
```

```
nelmio_cors:
```

```
  defaults:
```

```
    allow_credentials: false
```

```
    allow_origin: ['*']
```

```
    allow_headers: ['*']
```

```
    allow_methods: ['GET', 'POST', 'PUT', 'DELETE', 'OPTIONS']
```

```
    max_age: 3600
```

```
    hosts: []
```

```
    origin_regex: false
```

```
# Friends Of Symfony Configuration
```

```
fos_rest:
```

```
  view:
```

```
    view_response_listener: 'force'
```

```
  formats:
```

```
    json: true
```

```
  format_listener:
```

```
    rules:
```

```
      - { path: ^/api, priorities: [ json ], fallback_format: json,  
        prefer_extension: true }
```

```
      - { path: ^/, priorities: [ html ], fallback_format: html,  
        prefer_extension: true }
```

Debemos también, añadir en la sección de *Frameworks* de este archivo la siguiente línea para activar la serialización:

```
serializer:  
    enabled: true
```

Añadimos ahora, un directorio en el que se almacenarán las rutas configuradas para el paquete de Friends of Symfony. Dicho directorio tendrá la siguiente ruta final: `src/AppBundle/Resources/config`

En ese directorio, crearemos el archivo `api-rest-routing.yml`, que incluirá las rutas del servicio web. Siguiendo una forma del estilo:

```
api_progenitors:  
    type: rest  
    resource: "@AppBundle/Controller/api/ProgenitorsController.php"  
    name_prefix: api_progenitors_
```

Ahora, en el archivo de enrutamiento general del proyecto, `app/config/routing.yml`, debemos añadir la ruta para que el paquete pueda encontrar su lista de enrutamientos:

```
app_bundle_api:  
    type: rest  
    prefix: /api  
    resource: "@AppBundle/Resources/config/api-rest-routing.yml"
```

Por último, permitimos que se acceda al servicio web si autenticarse como administrador. Esto es necesario porque la aplicación web está protegida para forzar la autenticación como administrador. Para ello editamos el archivo `/app/config/security.yml` y en la sección `access_control` añadimos:

```
{ path: ^/api/, roles: IS_AUTHENTICATED_ANONYMOUSLY }
```

Y ya tendríamos los paquetes configurados para hacer uso de la API en nuestro proyecto de *backend*.

9.3.3 Enrutamiento de la API

En el servicio web de la aplicación móvil el enrutamiento hacia los métodos correctos ha sido una parte importante del desarrollo de la API. Haciendo uso del módulo `$http` desde Ionic hemos realizado peticiones Ajax de manera sencilla a dicho servicio web, por lo que nos pareció relevante organizar de manera correcta y homogénea dichos enrutamientos. Por este motivo hemos decidido seguir las prácticas recomendadas para realizar **[45] APIs RESTful pragmáticas** en la construcción de nuestra API, obteniendo los siguientes beneficios:

- Usa estándares web que tienen sentido.
- Es amigable y entendible para el desarrollador, además de explorable con sólo visualizar la ruta de la barra de direcciones.
- Es sencilla, intuitiva y consistente, no sólo fácil de usar, también agradable.
- Provee la suficiente flexibilidad para proveer a cualquier interfaz de usuario.
- Es eficiente, manteniendo el balance con los otros requerimientos.

Para ello hemos tomado las siguientes medidas:

- Hemos separado nuestra API en recursos lógicos usando nombre sustantivos que tienen sentido desde la perspectiva del consumidor de la API.
- Usamos los métodos HTTP para manejar las acciones CRUD (crear, leer, actualizar y eliminar) mapeados de la siguiente manera:
 - GET: obtenemos un recurso
 - POST: creamos un recurso.
 - PUT: actualizamos un recurso.
 - PATCH: actualizamos parcialmente un recurso.
 - DELETE: eliminamos un recurso.
- Para evitar el uso de plurales irregulares, lo más sencillo es utilizar siempre el plural en los nombres de los *endpoint*.

10. Pruebas

10.1 Pruebas de usabilidad

Para verificar que nuestra aplicación era satisfactoria y cumplía los objetivos que nos habíamos marcado, realizamos una serie de pruebas del prototipo solicitando a distintos usuarios finales (padres, madres y compañeros del grado con conocimientos en informática) que hicieran uso de nuestra aplicación y íbamos comprobando y solicitando que nos dieran su opinión sobre los siguientes apartados:

- Verificar si la interfaz era clara, sencilla de usar, intuitiva.
- Comprobar si el proceso de inicio de sesión y registro resultaba tedioso o largo de realizar.
- Verificar si los usuarios, por sí mismos y sin indicaciones, eran capaces de navegar sin problemas por la aplicación y realizar las acciones que deseaban.
- Asegurar que cuando se mostraba un mensaje de error, éste era claro y conciso y ayudaba al usuario a subsanarlo por sí mismo.
- Medir y cuantificar si los tiempos de carga resultaban o no molestos para el usuario.
- Preguntar si realmente notaban alguna diferencia respecto a aplicaciones nativas en cuanto a rendimiento, fluidez o diseño.
- Solicitar si recomendarían dicha aplicación a otros padres y colegios.

Una vez realizadas dichas pruebas de usabilidad, vimos que nuestros objetivos habían sido cumplidos por las respuestas de los usuarios:

- La aplicación es sencilla y clara de usar, no tiene demasiados adornos y se indican claramente los apartados.
- El proceso de inicio de sesión es muy similar al de otras aplicaciones de mensajería, por lo que parece estándar, sencillo y rápido.
- No tuvimos que aclarar a los usuarios como navegar por la aplicación, ni mucho menos que acciones podían realizarse o qué significaba algún componente de la interfaz.
- Los mensajes de error son claros y ayudan mucho a darnos cuenta de qué errores se han cometido y hay muy pocos.
- La aplicación es rápida, con un rendimiento muy similar al de otras aplicaciones de mensajería.
- La mayoría pensó que se trataba de una aplicación nativa en el caso de los padres. Los compañeros con conocimientos en informática notaban la diferencia, pero nos comentaban que era casi inapreciable para un usuario estándar.
- Todos los usuarios de prueba recomendarían el uso de la aplicación.

10.2 Pruebas de interacción

Por último, para verificar que nuestra aplicación y nuestra API funcionaban correctamente con el *backend* desarrollado por Adrián Louro Alonso, realizamos una serie de pruebas de interacción siguiendo las siguientes pruebas explicadas a continuación:

- Desplegamos la base de datos y el proyecto del *backend* en uno de nuestros equipos de desarrollo para usarlo como servidor de la aplicación móvil.
- Probamos que, usando la dirección externa de este servidor, en el entorno de pruebas de escritorio permitía que la aplicación móvil funcionaba perfectamente y podía conectarse a la API instalado en un equipo remoto sin problemas.
- Instalamos la aplicación móvil en nuestros respectivos teléfonos, que usan iOS y Android respectivamente, para comprobar que tampoco existía ningún problema de conexión con el servidor externo y que el funcionamiento era adecuado en ambas plataformas.

Por último, procedimos a comprobar que las aplicaciones se comunicaban a tiempo real. Para ello, lanzamos la aplicación web desarrollada por Adrián y realizamos la siguiente batería de pruebas desde la aplicación móvil y la de escritorio:

- Registrando a un padre desde la aplicación móvil y asociándolo a un centro disponible, nos encontrábamos que efectivamente, al acceder con el usuario administrador de ese centro, podíamos enlazar dicho padre a un alumno del centro porque ya aparecía registrado en el sistema.
- Al enviar un mensaje de algún tipo a uno de los hijos asociados al padre registrado, aparecía en la aplicación móvil en la sección correspondiente y se podía visualizar su contenido sin problemas.
- Respondiendo a una autorización o a una encuesta desde la aplicación móvil quedaba reflejado de manera inmediata en la aplicación web, actualizándose el listado de los padres que habían respondido a dicha encuesta o autorización.

11. Resultados, conclusiones y trabajo futuro

11.1 Resultados y conclusiones

En general, consideramos que la realización de este proyecto ha sido una experiencia cuanto menos, enriquecedora. Después de muchos años de esfuerzo por fin nos hemos enfrentado a una situación real, en la que teníamos que elaborar desde cero un trabajo original. Esto, por supuesto ha conllevado muchas horas de trabajo y esfuerzo, no sólo por el desarrollo en sí, si no por todo lo que conlleva.

El pasar por todas las etapas del desarrollo, nos ha ayudado muchísimo a aprender a tener un menor temor a la autocrítica, sin dejar de impulsar nuestra capacidad de toma de decisiones. Este trabajo ha sido analizado, diseñado y ejecutado por nosotros, por lo tanto, teníamos que basarnos en los conocimientos que hemos adquirido durante el grado para poder tomar las decisiones correctas.

Por supuesto, la mayoría del contenido aquí visto no lo hemos visto en la carrera, pero hay mucho que sí, pero eso no es lo importante. Lo importante y que más valoramos de este trabajo es que nos hemos dado cuenta de que durante el grado, en esencia, no se nos proporcionan conocimientos específicos. Se establecen unas bases y se definen buenas prácticas y metodologías de trabajo. Dado el carácter autodidacta que tiene la salida profesional de este grado, creemos que lo más destacable que podemos sacar del grado es justo eso, la capacidad de poder aprender por nosotros mismos, aunque nos cueste repetidas horas de esfuerzo y trabajo, nunca será en vano.

Y hemos podido ver cómo, a pesar de no tener el conocimiento específico de las nuevas tecnologías aprendidas (como Symfony, Doctrine, Ionic, AngularJS), en ningún momento nos hemos sentido del todo perdidos. Siempre hemos tenido alguna idea de que probar o qué cosas podrían funcionar, porque eso es lo que nos ha permitido el desarrollo del grado, ganar todos esos recursos que pueden hacerte pasar de estar estancado en un análisis, un diseño o un desarrollo, a saber, siempre hacia donde seguir.

Creemos fervientemente que el mejor regalo que ha supuesto este trabajo es darnos cuenta de que sin duda, por fin nos sentimos capaces de afrontar cualquier problema en lo que respecta a nuestras competencias, y de que no tenemos ningún reparo en conocer y adaptarnos a nuevas tecnologías. Sin lo cual, dada la naturaleza cambiante de nuestra profesión, estaríamos perdidos en el mundo laboral.

En lo que respecta a los resultados de la aplicación, nos sentimos orgullosos del trabajo realizado. Puede que tenga muchas cosas mejorables, pero tampoco es una aplicación de principiantes. Hemos realizado un buen análisis previo, diseñado una buena arquitectura, desarrollado la aplicación y finalmente, conectar ambas aplicaciones para que funcionaran en conjunto sin problemas. Y no simplemente lo hemos hecho, lo hemos hecho de manera estructurada, organizada y haciendo uso de metodologías, arquitecturas, patrones de diseño y buenas prácticas aprendidas durante el grado. Con

lo que hemos podido ver, como todos los conocimientos adquiridos, se volcaban en este trabajo, dando resultados fructíferos, efectivos y eficientes.

11.2 Trabajo futuro

En lo que respecta al sistema completo, ya adelantamos que había funcionalidades que decidimos no implementar, principalmente por falta de tiempo, pero que a largo plazo podrían ser fácilmente desarrolladas y enriquecerían de sobremanera el sistema. Dichas funcionalidades comprenderían un sistema de tutorías completo, en el que padres de alumnos y el profesorado pudieran comunicarse y establecer citas para comunicarse información los alumnos. Este sistema abarcaba implementar las siguientes características:

- Administradores del centro:
 - Registrar profesores en el centro.
 - Asociar profesores a un curso.
 - Asociar un tutor a un curso.
 - Establecer asignaturas en el centro.
 - Asociar asignaturas a un profesor.
- Profesores:
 - Establecer un horario de tutorías.
 - Solicitar tutorías a los padres.
 - Ver calendario de tutorías.
 - Confirmar tutorías.
 - Cancelar tutorías.
- Padres:
 - Solicitar tutorías a los profesores de sus hijos.
 - Confirmar asistencia a una tutoría.
 - Cancelar tutorías.

Salvando dichas funcionalidades, existen una serie de mejoras en la aplicación móvil que permitirían disfrutar de una experiencia enriquecida:

- Implementar HTTPS para el cifrado del envío de datos personales como el teléfono o el nombre completo.
- Visualizar desde el listado de mensajes de cada categoría si un mensaje ha sido leído o no.
- Dividir la bandeja de entrada de mensajes de circulares en Leídas / No leídas y En vigor/Vencidas para autorizaciones y encuestas.
- Poder destacar un mensaje concreto como importante, poniéndolo al principio de la lista independientemente de la fecha.
- Introducir notificaciones *push* en el proyecto, que permita conocer en tiempo real a los padres cuando ha llegado un nuevo mensaje.

- Permitir que las encuestas tengan múltiples preguntas.
- Refinar la interfaz de usuario, animaciones, diseño.
- Diseñar una interfaz de usuario para específica para tablets.
- Integrar la visualización de los archivos adjuntos en la aplicación, sin que se requiera una descarga, y que desde la visualización se permita abrir el archivo con aplicaciones que admitan la extensión y estén instaladas en el sistema.
- Implementar un sistema de validación por SMS real, dado que requiere el uso de un servicio de pago específico para ello, sólo se implementó a modo de *mockup* en nuestra aplicación.

12. Bibliografía

- [1] creaTáctil S.L. “miColegioApp”, [en línea]. Disponible en: <http://micolegioapp.com/wordpress/que-es-micolegioapp/>
- [2] TokApp Online S.L. “Centros Educativos”, [en línea]. Disponible en: <https://www.tokappschool.com/welcome/centros>
- [3] Clickartedu. “¿Qué es Clickedu?”, [en línea]. Disponible en: <http://www.clickartedu.com/>
- [4] Remind. “Las comunicaciones escolares no deberían ser difíciles”, [en línea]. Disponible en: <https://www.remind.com/es-419/>
- [5] Class Twist Inc. “Aulas más felices”, [en línea]. Disponible en: <https://www.classdojo.com/es-es/>
- [6] Wikipedia. “GNU General Public License”, [en línea]. Disponible en: https://es.wikipedia.org/wiki/GNU_General_Public_License
- [7] Apache Friends. “About the XAMPP Project”, [en línea]. Disponible en: <https://www.apachefriends.org/about.html>
- [8] Oracle Corporation. “Commercial License for OEMs, ISVs and VARs”, [en línea]. Disponible en: <https://www.mysql.com/about/legal/licensing/oem/>
- [9] StarUML. “About StarUML”, [en línea]. Disponible en: <http://staruml.sourceforge.net/v1/license.php>
- [10] Wikipedia. “Apache License”, [en línea]. Disponible en: https://es.wikipedia.org/wiki/Apache_License
- [11] Wikipedia. “Licencia PHP”, [en línea]. Disponible en: https://es.wikipedia.org/wiki/Licencia_PHP
- [12] Wikipedia. “Licencia MIT”, [en línea]. Disponible en: https://es.wikipedia.org/wiki/Licencia_MIT
- [13] JetBrains s.r.o. “Toolbox Subscription License Agreement For Education And Training”, [en línea]. Disponible en: https://www.jetbrains.com/store/license_classroom.html
- [14] Justinmind. “Terms of Use”, [en línea]. Disponible en: <https://www.justinmind.com/terms>
- [15] Microsoft Corporation. “Términos de licencia de Microsoft Office 365”, [en línea]. Disponible en: http://download.microsoft.com/Documents/UseTerms/Office%20365_University_Spanish_e3b1384a-d4f2-415e-9674-67f1997fa630.pdf
- [16] Wikipedia. “Ley Orgánica de Protección de Datos de Carácter Personal (España)”, [en línea]. Disponible en: [https://es.wikipedia.org/wiki/Ley_Org%C3%A1nica_de_Protecci%C3%B3n_de_Datos_de_Car%C3%A1cter_Personal_\(Espa%C3%B1a\)](https://es.wikipedia.org/wiki/Ley_Org%C3%A1nica_de_Protecci%C3%B3n_de_Datos_de_Car%C3%A1cter_Personal_(Espa%C3%B1a))
- [17] A.T.T.I., S.L. “Niveles de seguridad de los ficheros de datos según la LOPD”, [en línea]. Disponible en: http://www.telecomunicacionesalicante.com/informatica/niveles_seguridad_ficheros_datos_en_la_lopd.html

- [18] M.C. Verónica Rodríguez Rodríguez. “Modelos de proceso evolutivo”, [en línea]. Disponible en: <https://sites.google.com/site/is11801/contenido/modelos-de-proceso-evolutivo>
- [19] Wikipedia. “Modelo de prototipos”, [en línea]. Disponible en: https://es.wikipedia.org/wiki/Modelo_de_prototipos
- [20] Drifty Co. “Ionic”, [en línea]. Disponible en: <https://ionicframework.com/>
- [21] The Apache Software Foundation. “Apache Cordova”, [en línea]. Disponible en: <https://cordova.apache.org/>
- [22] Google. “Angular”, [en línea]. Disponible en: <https://angular.io/>
- [23] Fabien Potencier. “Symfony”, [en línea]. Disponible en: <https://symfony.com/>
- [24] The PHP Group. “PHP”, [en línea]. Disponible en: <http://php.net/>
- [25] Doctrine Team. “Welcome to the Doctrine Project”, [en línea]. Disponible en: <http://www.doctrine-project.org/>
- [26] Code Schook a Pluralsight company. “JavaScript”, [en línea]. Disponible en: <https://www.javascript.com/>
- [27] Oracle Corporation. “MySQL”, [en línea]. Disponible en: <https://www.mysql.com/>
- [28] Linus Torvalds “Git”, [en línea]. Disponible en: <https://git-scm.com/>
- [29] JetBrains s.r.o. “PhpStorm Lightning-smart PHP IDE”, [en línea]. Disponible en: <https://www.jetbrains.com/phpstorm/>
- [30] GitHub Inc. “Atom, a hackeable text editor for the 21st Century”, [en línea]. Disponible en: <https://atom.io/>
- [31] Oracle Corporation. “MySQL Workbench”, [en línea]. Disponible en: <https://www.mysql.com/products/workbench/>
- [32] Google. “Un navegador web rápido y gratuito”, [en línea]. Disponible en: <https://www.google.es/chrome/browser/desktop/index.html>
- [33] Justinmind. “All-in one Prototyping Tool for web and mobile apps”, [en línea]. Disponible en: <https://www.justinmind.com/>
- [34] GitHub Inc. “Github”, [en línea]. Disponible en: <https://github.com/>
- [35] Apache Friends. “¿Qué es XAMPP?”, [en línea]. Disponible en: <https://www.apachefriends.org/es/index.html>
- [36] The Apache Software Foundation. “The HTTP Server Project”, [en línea]. Disponible en: <https://httpd.apache.org/>
- [37] Wikipedia. “Lenguaje unificado de modelado”, [en línea]. Disponible en: https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado
- [38] Wikipedia. “Programación por capas”, [en línea]. Disponible en: https://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas
- [39] StackExchange. “Is it a bad practice to store large files (10 MB) in a database?”, [en línea]. Disponible en: <https://softwareengineering.stackexchange.com/questions/150669/is-it-a-bad-practice-to-store-large-files-10-mb-in-a-database>

- [40] Wikipedia. “Modelo-vista-controlador”, [en línea]. Disponible en: <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>
- [41] Juan Méndez Rodríguez. “Reglas Empíricas de Diseño de Interfaces”, [en línea]. Disponible en: [http://telepresencial1617.ulpgc.es/cv/ulpgctp17/pluginfile.php/80359/mod_resource/content/1/Tema I 7.pdf](http://telepresencial1617.ulpgc.es/cv/ulpgctp17/pluginfile.php/80359/mod_resource/content/1/Tema%20I%207.pdf)
- [42] Drifty Co. “Ionic Concepts – App Structure”, [en línea]. Disponible en: <http://ionicframework.com/docs/v1/concepts/structure.html>
- [43] Fabien Potencier. “FOSRestBundle”, <https://github.com/FriendsOfSymfony/FOSRestBundle>
- [44] Fabien Potencier. “Routing (FOS Rest Bundle)”, [en línea]. Disponible en: <http://symfony.com/doc/current/bundles/FOSRestBundle/5-automatic-route-generation-single-restful-controller.html>
- [45] Vinay Sahni. “Best Practices for Designing a Pragmatic RESTful API”, [en línea]. Disponible en: <http://www.vinaysahni.com/best-practices-for-a-pragmatic-restful-api>

Anexo I: Manual de usuario

Introducción

En este apartado procederemos a explicar concisa y precisamente las distintas funcionalidades de la aplicación web desarrollada para los padres/madres del alumnado. Se irán explicando una a una las distintas partes del programa, acompañadas de una ilustración para detallar su funcionamiento.

Inicio de sesión – Registro

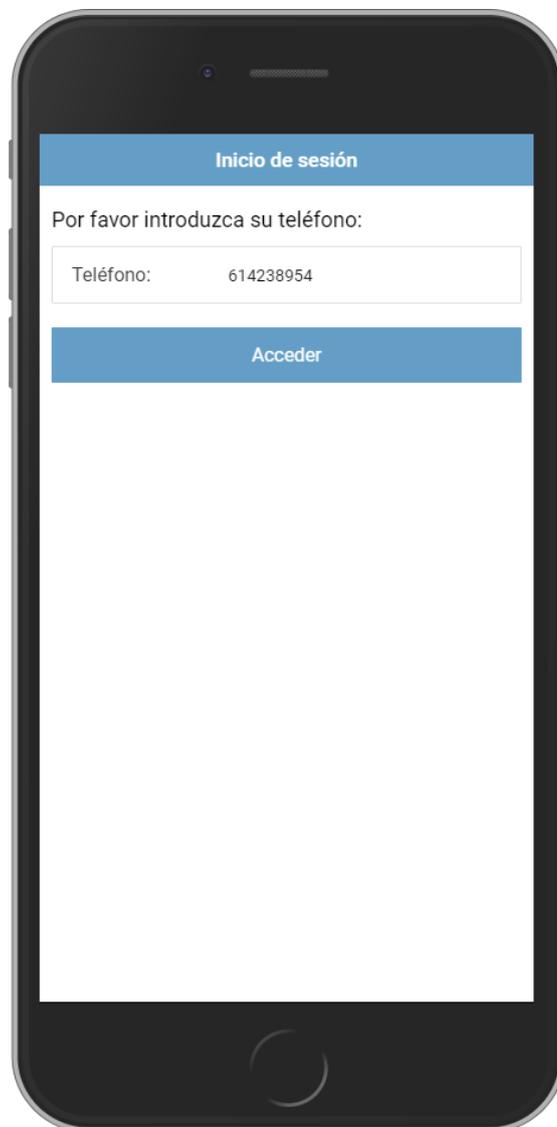


Ilustración 13: Inicio de sesión

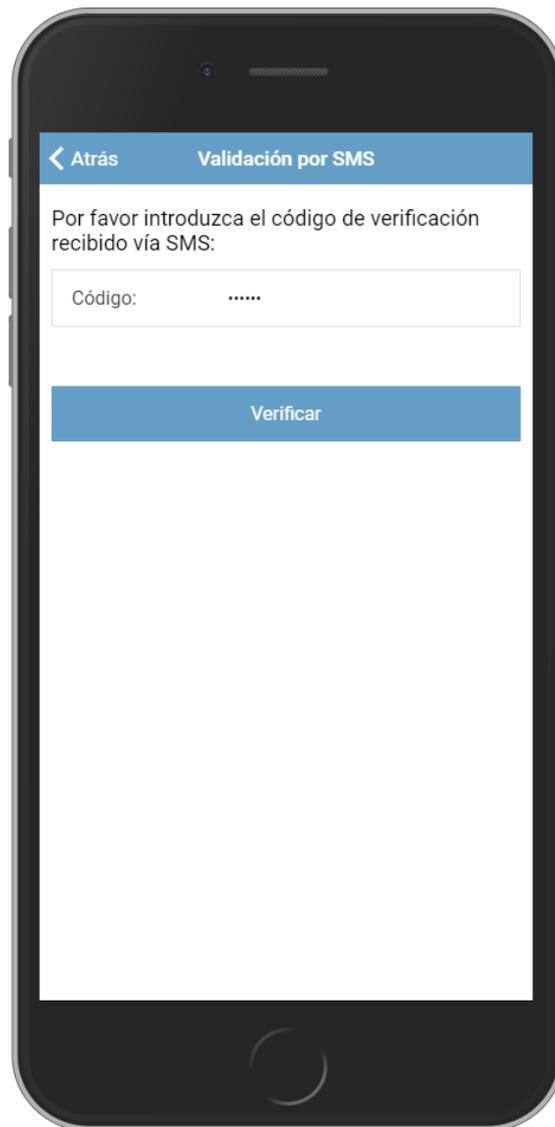


Ilustración 14: Validación por SMS

En la pantalla correspondiente a la *Ilustración 13*, el padre/madre introduce su teléfono móvil siendo únicamente válidos aquellos de 6 dígitos. En caso de encontrar el teléfono en la base de datos del sistema, se procede a un inicio de sesión en el mismo, de lo contrario, se procede al registro del padre/madre.

En la siguiente pantalla, *Ilustración 14*, procedemos a validar el código recibido por SMS. Esta implementación no se realizó por motivos detallados anteriormente, así que solo se muestra a modo de *mock-up*. El código por defecto es "123456".

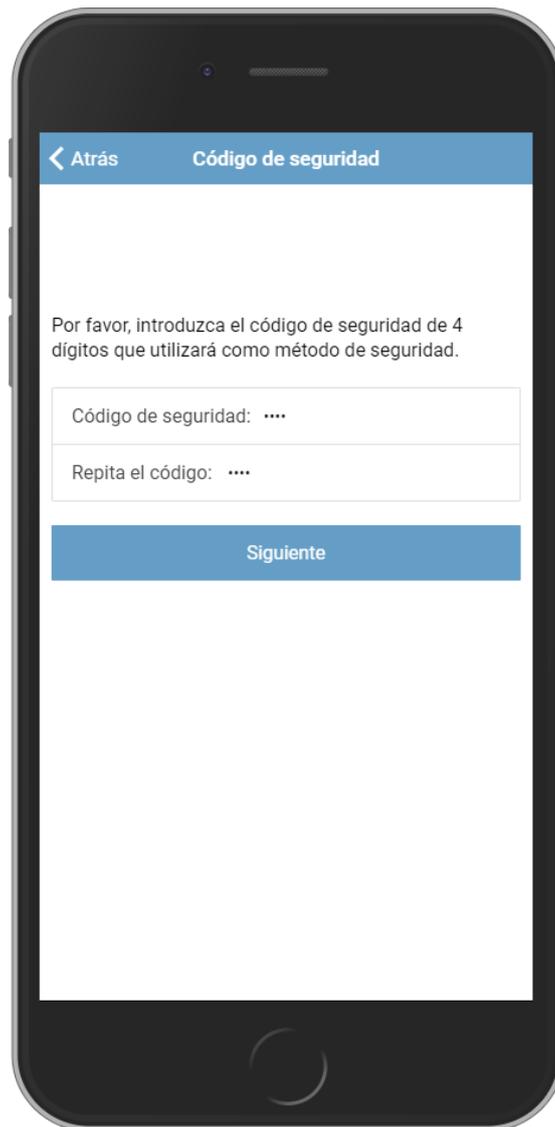


Ilustración 15: Establecer código de seguridad

En la *Ilustración 15* podemos observar que en esta pantalla sólo se requiere establecer un código de seguridad. Para ello, sólo debemos escoger un código de 4 dígitos y repetirlo donde se indica.

En el caso de ser un inicio de sesión, esto es, ya estamos registrados en el sistema, la pantalla mostraría un mensaje de “Finalizar” en el botón mostrado, y nos llevaría directamente a la pantalla de *Circulares*.

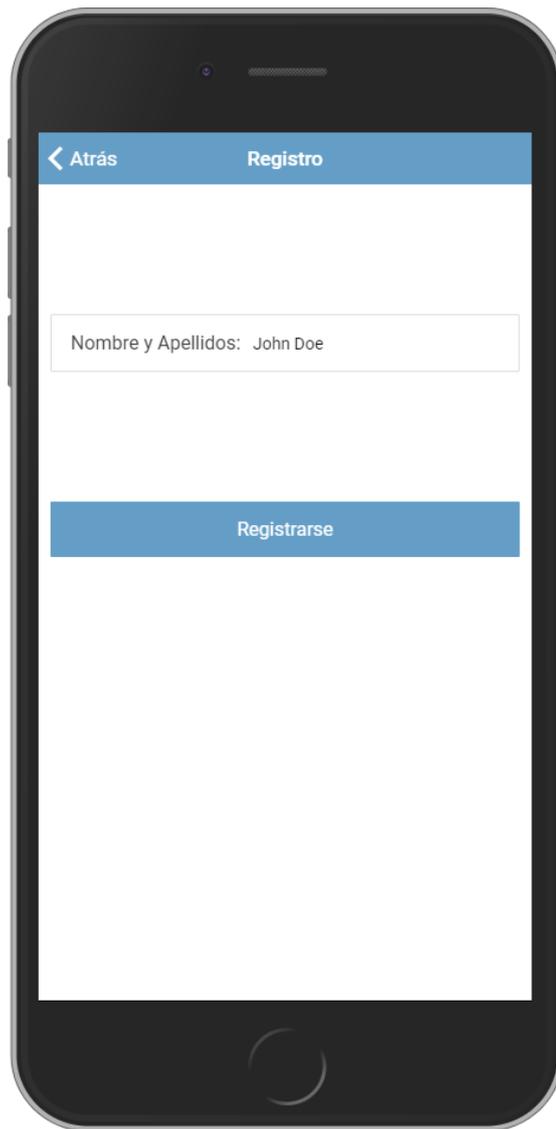


Ilustración 16: Registro

En la *Ilustración 16* podemos observar la pantalla de registro. En ella se solicita al usuario que introduzca su *Nombre y Apellidos* para proceder al registro.

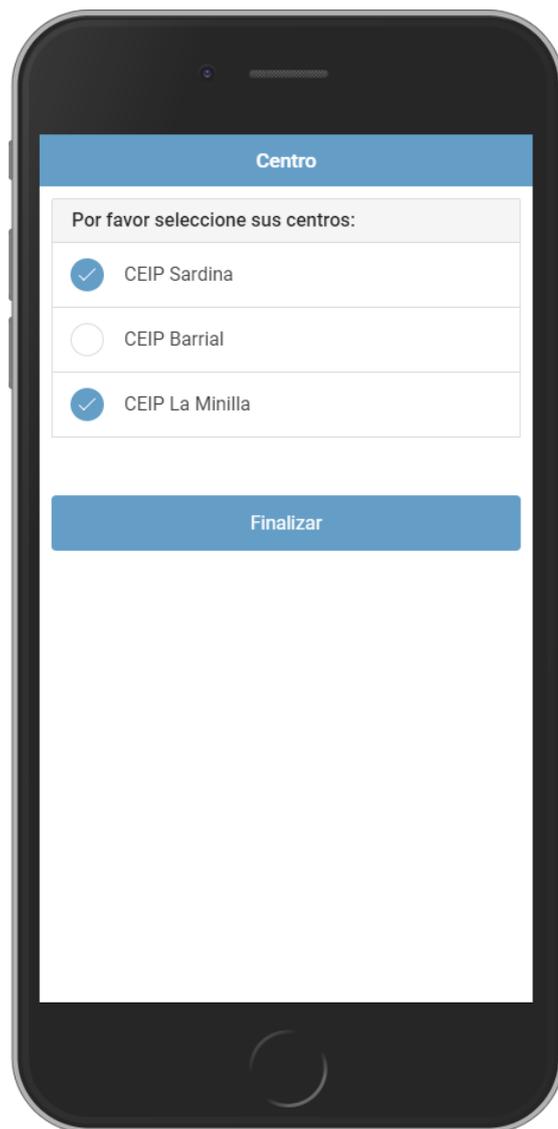


Ilustración 17: Selección de centros

Como último paso del registro, se aprecia en la *Ilustración 17* que se le solicita al usuario que seleccione aquellos centros a los que permitirá conocer su información personal. Una vez seleccionados uno o varios centros, procedemos a finalizar el registro y a pasar a la pantalla observada en la *Ilustración 18*, llamada *Circulares*.

Circulares

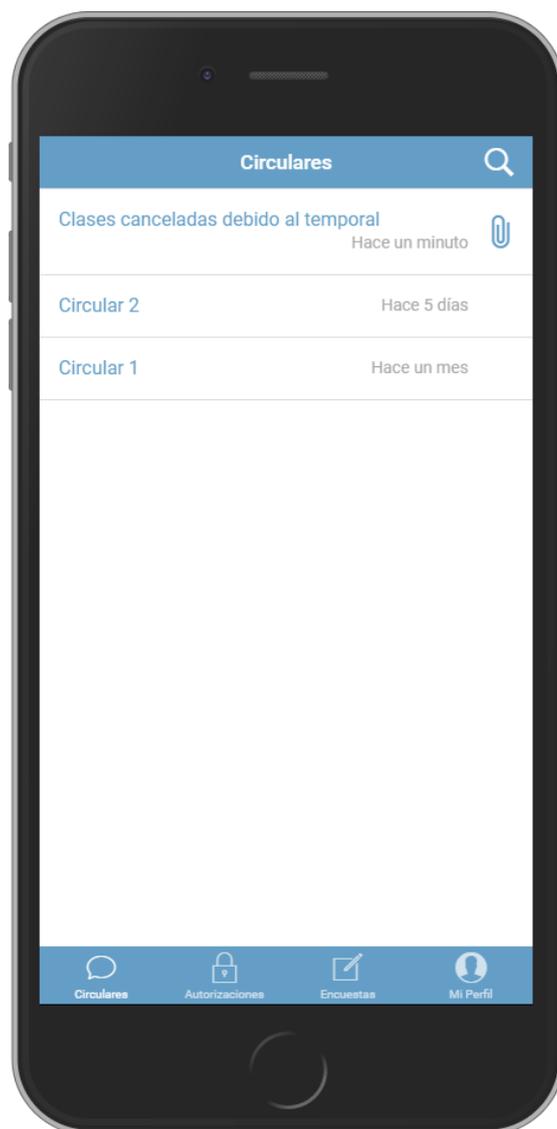


Ilustración 18: Circulares

En *Circulares* se aprecia la interfaz completa de la aplicación. En ella se aprecia que la misma está dividida en cuatro secciones principales: *Circulares*, *Autorizaciones*, *Encuestas* y *Mi Perfil*. Iremos repasando cada una de ellas en este anexo.

En *circulares* se aprecia en cada elemento de la lista su título, detallado en azul, un icono de un clip si dicha circular lleva un archivo adjunto, y un mensaje que nos indica hace cuánto tiempo que se ha enviado el mensaje desde la aplicación web del centro.

Además, en la esquina superior derecha, puede apreciarse un icono de búsqueda, que al activarlo permitirá realizar filtrados entre las circulares, como se puede observar en la *Ilustración 19*. Este componente está presente en todas las secciones, salvando la de *Mi Perfil*.

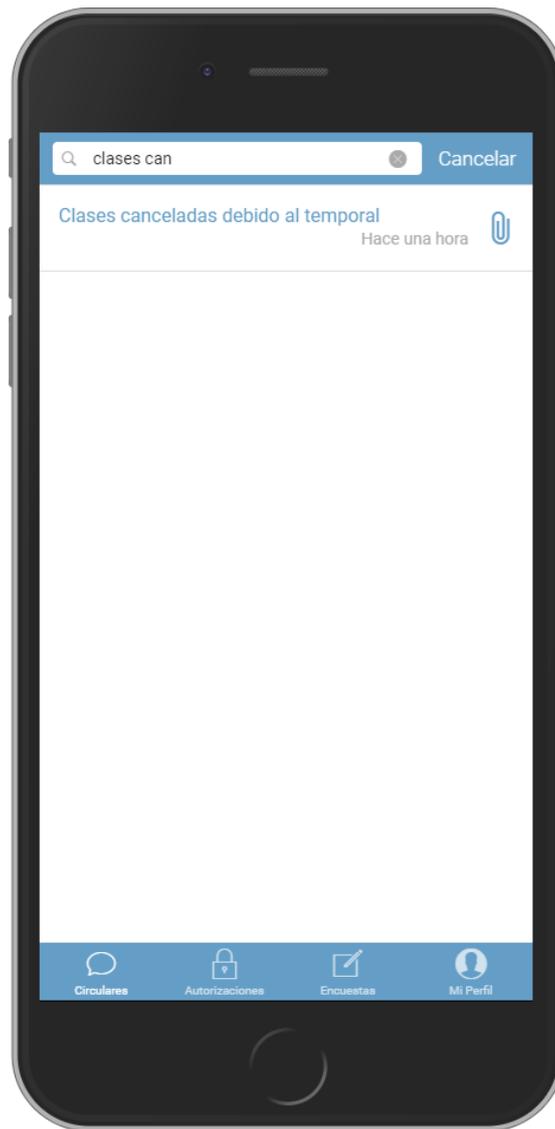


Ilustración 19: Búsqueda de mensajes

Si accedemos a alguno de los elementos de circulares, pasaremos a la pantalla de contenido de las mismas, apreciable en la *Ilustración 20*.

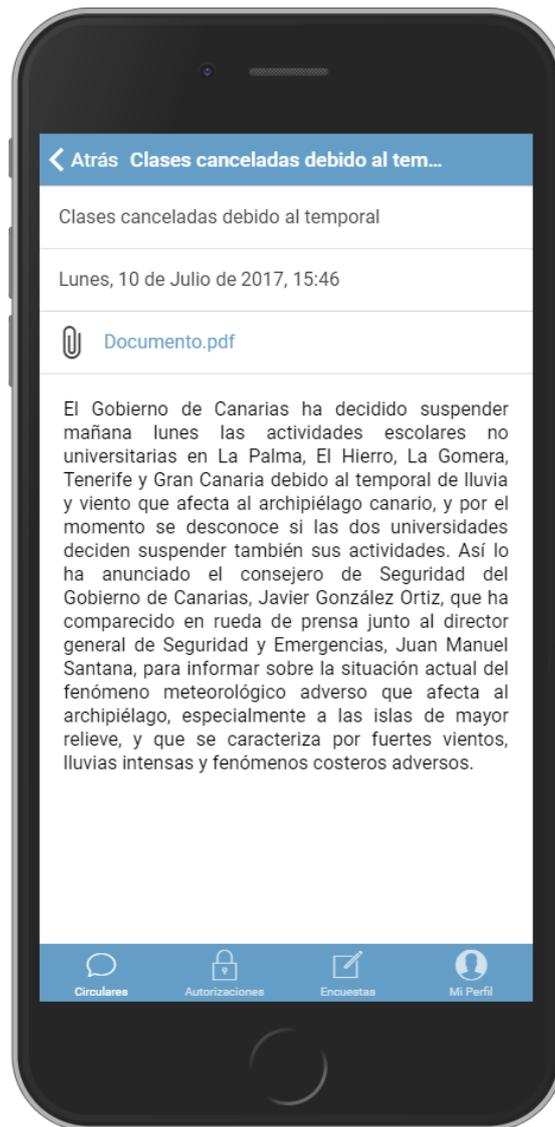


Ilustración 20: Contenido de circular

En ella se pueden apreciar los siguientes elementos: en la esquina superior izquierda, podemos volver a *Circulares* tocando el botón de atrás. En el encabezado de la circular podemos ver el título (a veces acortado por su extensión o resolución de la pantalla), y en su contenido, de nuevo el título por si apareciera acortado, la fecha en la que la circular fue enviada, una sección de archivo adjunto si la circular lo tuviera y por último el contenido. En caso de accionar el botón del archivo adjunto, se nos redirigirá al navegado por defecto del sistema para proceder a su descarga.

Autorizaciones

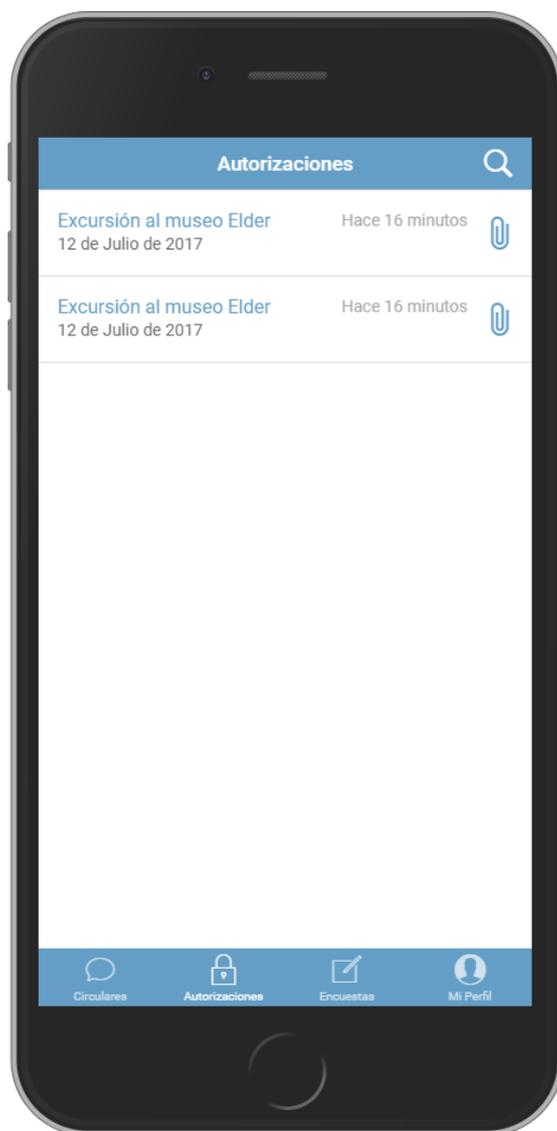


Ilustración 21: Autorizaciones

En la *Ilustración 21* podemos observar que el listado de *Autorizaciones*, así como el de *Encuestas*, añade además un subtítulo, que informa de la fecha límite que dicha autorización para ser respondida. Si accedemos a una de ellas nos encontraremos con la pantalla mostrada en la *Ilustración 22*.

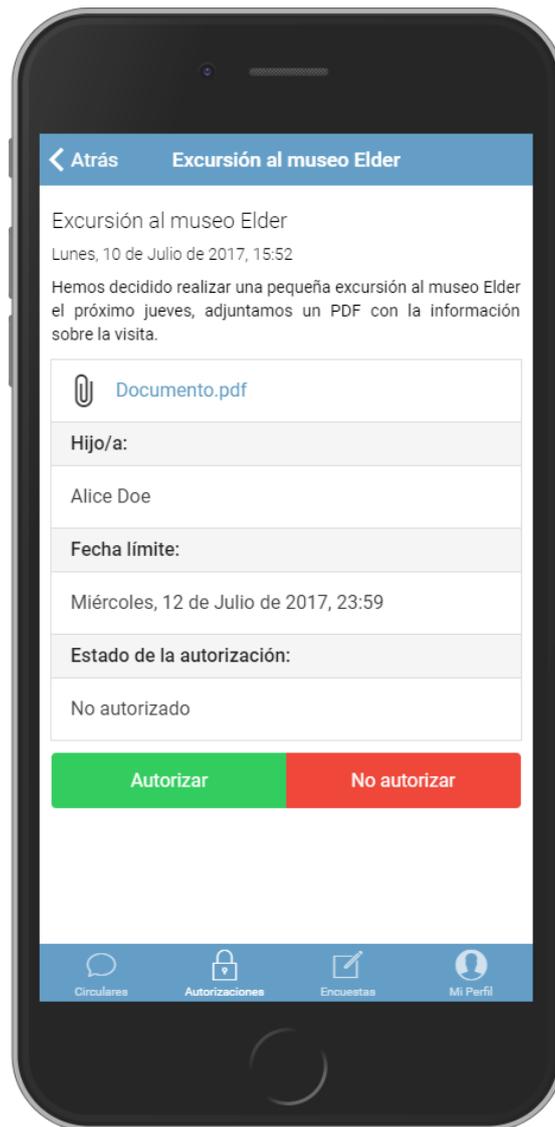


Ilustración 22: Contenido de autorización

Siguiendo el mismo patrón visto en la *Ilustración 20* en el contenido de las autorizaciones aparecen además de los elementos ya vistos la siguiente información: el hijo/a para el que está asociada la autorización, la fecha límite de la autorización y el estado actual de la autorización.

En caso de estar dentro del plazo de la fecha límite, aparecerán además dos botones que permiten responder afirmativamente a la solicitud de la autorización o negativamente. Dichos botones aparecen juntos cuando la autorización no ha sido respondida, en caso contrario, sólo aparecerá el botón que permita realizar un cambio de opinión respecto a la respuesta enviada por primera vez. En caso de accionar cualquiera de los dos botones se accionará el *pop-up* que podemos ver en la *Ilustración 23*.

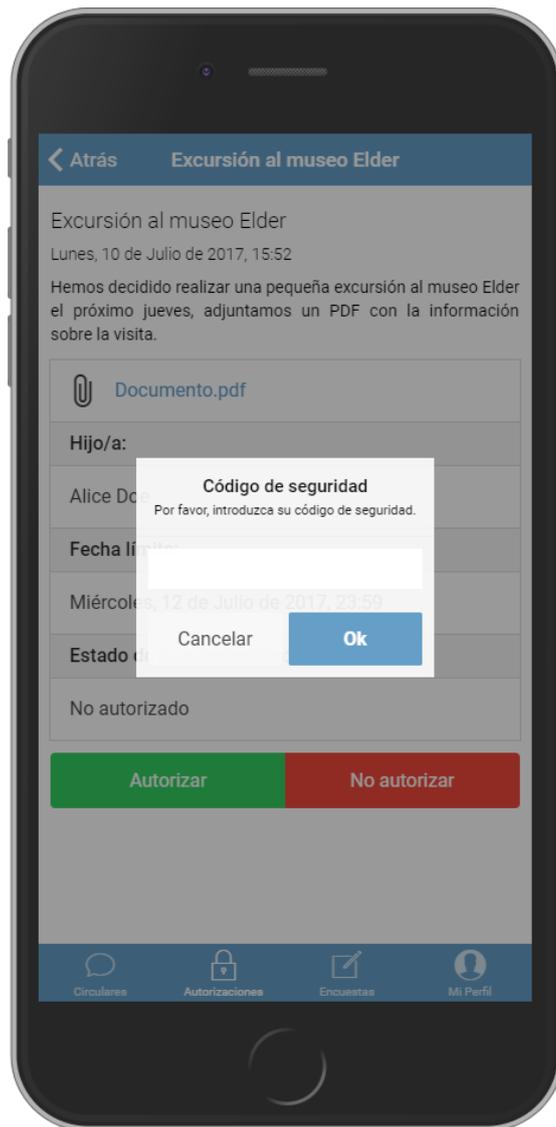


Ilustración 23: Solicitud del código de seguridad

Por cuestiones de seguridad, para poder autorizar o desautorizar a un hijo/hija deberemos introducir nuestro código de seguridad previamente establecido, de no ser correcto, no se permitirá realizar la acción.

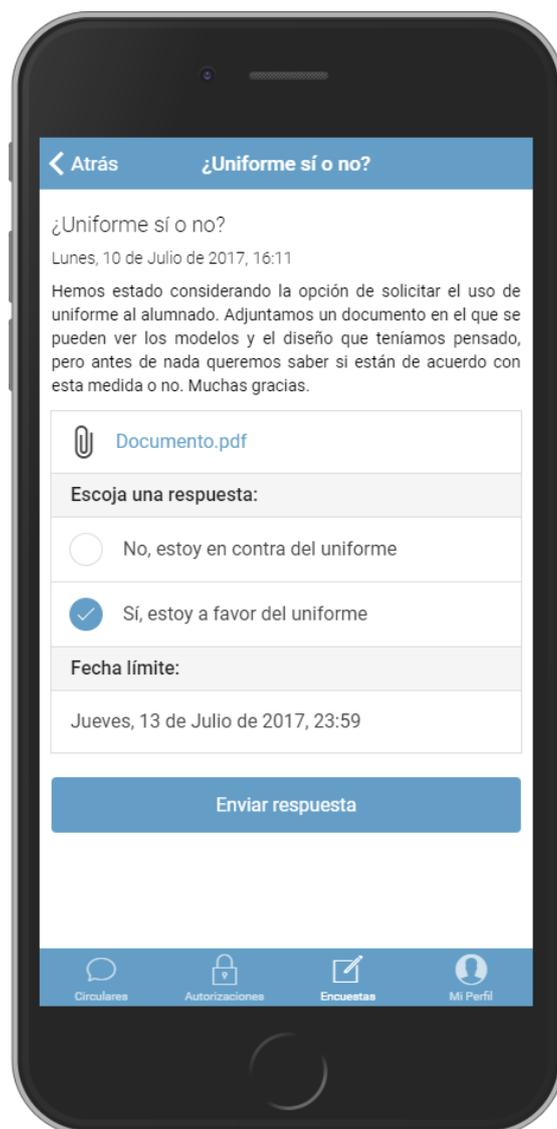


Ilustración 24: Contenido de encuesta

En la vista general de encuestas tenemos una representación idéntica a la *Ilustración 21* por lo que vamos a centrarnos en la vista del contenido de una encuesta. Como podemos apreciar en la *Ilustración 24* sobre estas líneas, se sigue el mismo modelo que en la vista del contenido de *Circulares* añadiendo una nueva sección en la que se muestran las respuestas a escoger, así como la fecha límite para responder la encuesta, como vimos en el contenido de *Autorizaciones*.

Para responder a una encuesta simplemente seleccionamos nuestra opción y accionamos el botón de enviar respuesta. En caso de ser una respuesta única la aplicación sólo permitirá seleccionar una respuesta, y, por el contrario, en caso de ser una respuesta múltiple se permitirán seleccionar una o más.

Mi Perfil

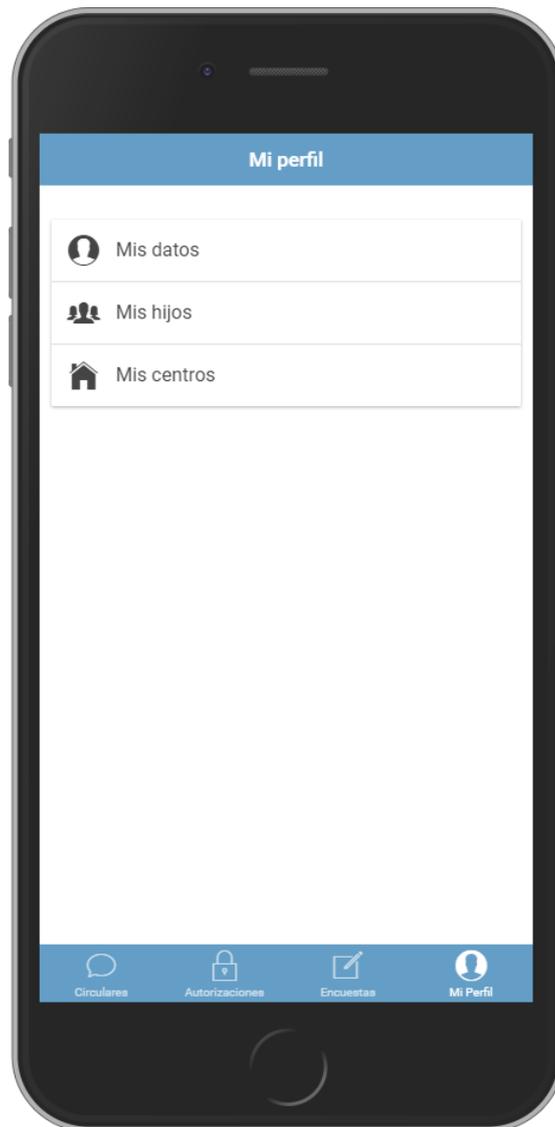


Ilustración 25: Mi perfil

En esta sección se pueden apreciar en la *Ilustración 25* que existen tres apartados diferenciados. Comenzaremos con *Mis datos*.

Mis datos

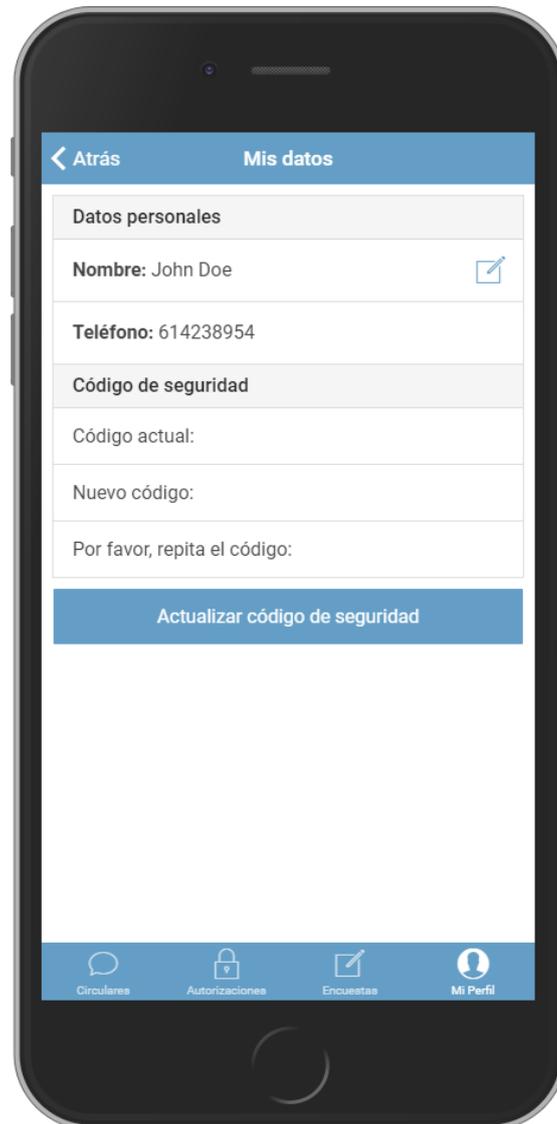


Ilustración 26: Mis datos

En este apartado, como muestra la *Ilustración 26* se nos muestran los datos personales del usuario, así como se nos permite cambiar el código de seguridad. Para realizarlo sólo tenemos que introducir el código actual y verificar dos veces el nuevo código. Además, se puede apreciar que, donde se muestra el nombre completo del usuario se permite editar el mismo con un botón a la derecha. Dicho botón lanzará la pantalla de edición que podemos ver en la *Ilustración 27* en la siguiente página.

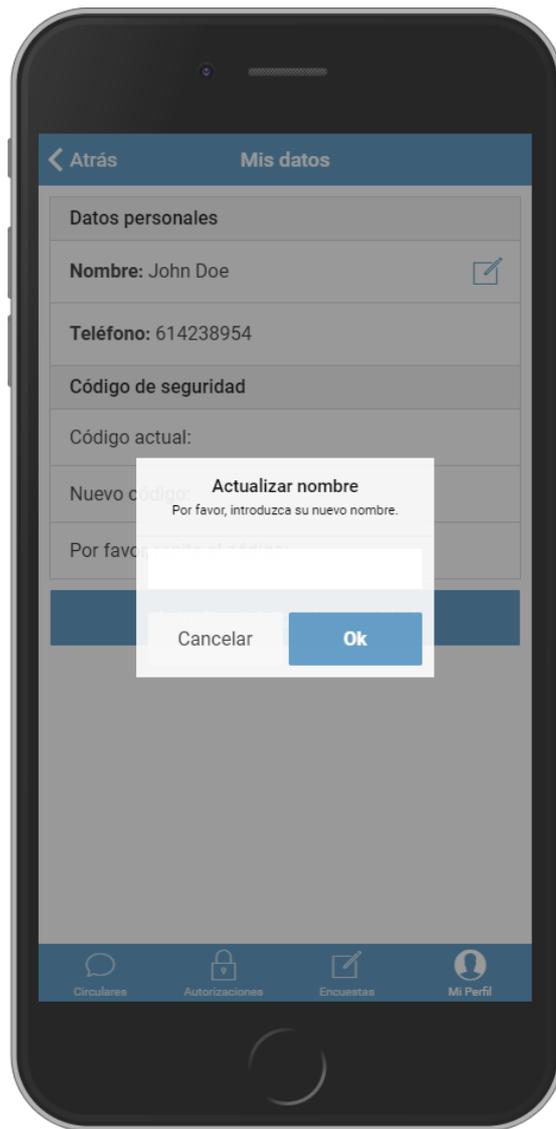


Ilustración 27: Cambio de nombre

En esta parte del apartado, simplemente introduciremos el nuevo nombre deseado y continuaremos tocando el botón “Ok”.

Mis hijos

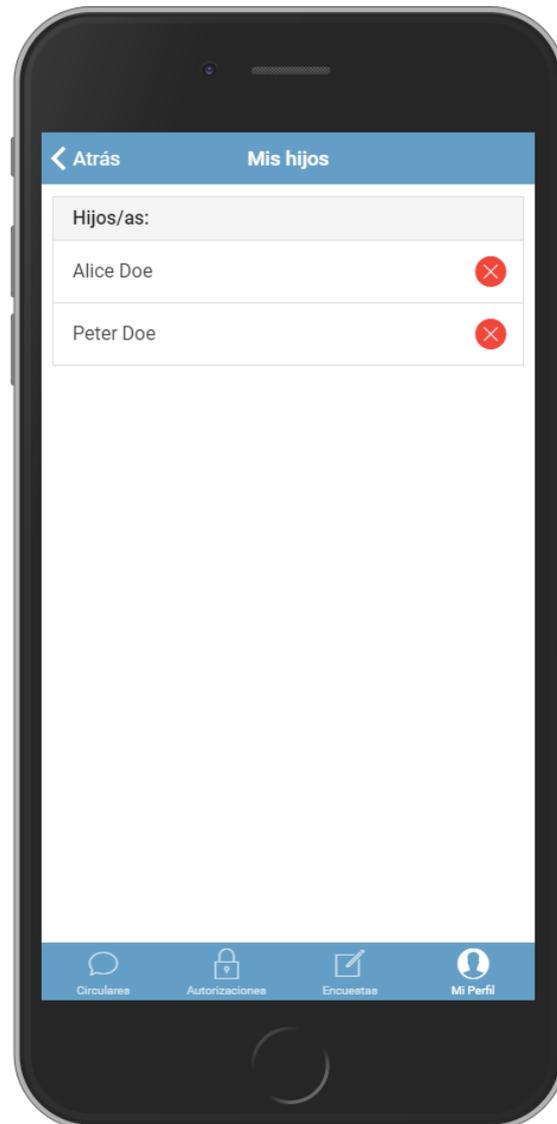


Ilustración 28: Mis hijos

Al acceder a esta sección (*Ilustración 28*) se nos presenta un listado de los hijos/as que el usuario tiene asociados. Tendremos la opción de desasociar un hijo/a realizando un simple toque en el botón de eliminar de su derecha.

Esto lanzará una pequeña pantalla de confirmación, apreciable en la *Ilustración 29*.

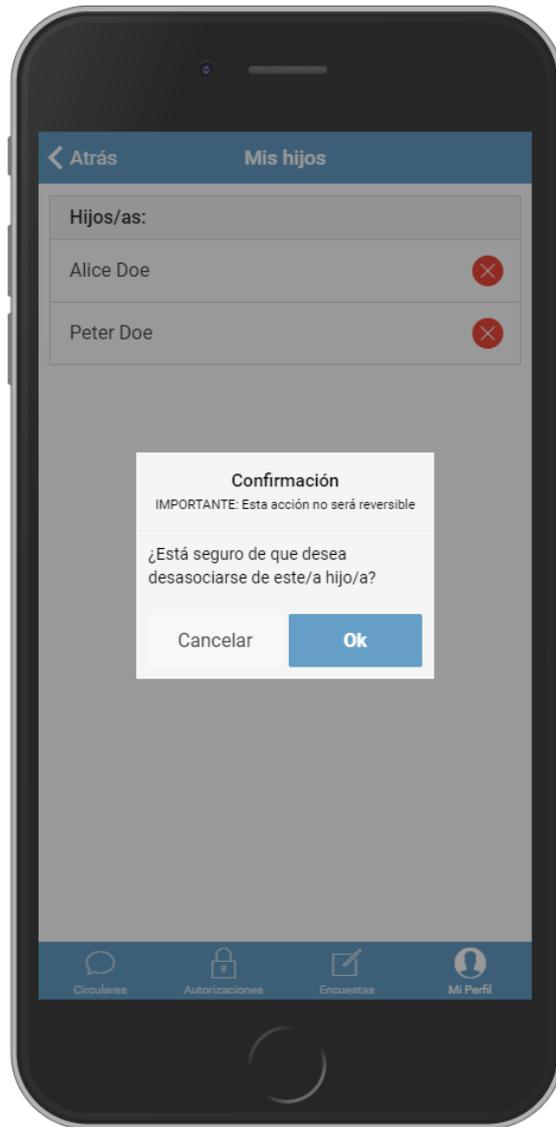


Ilustración 29: Confirmación al desasociar hijo/a

Mis centros

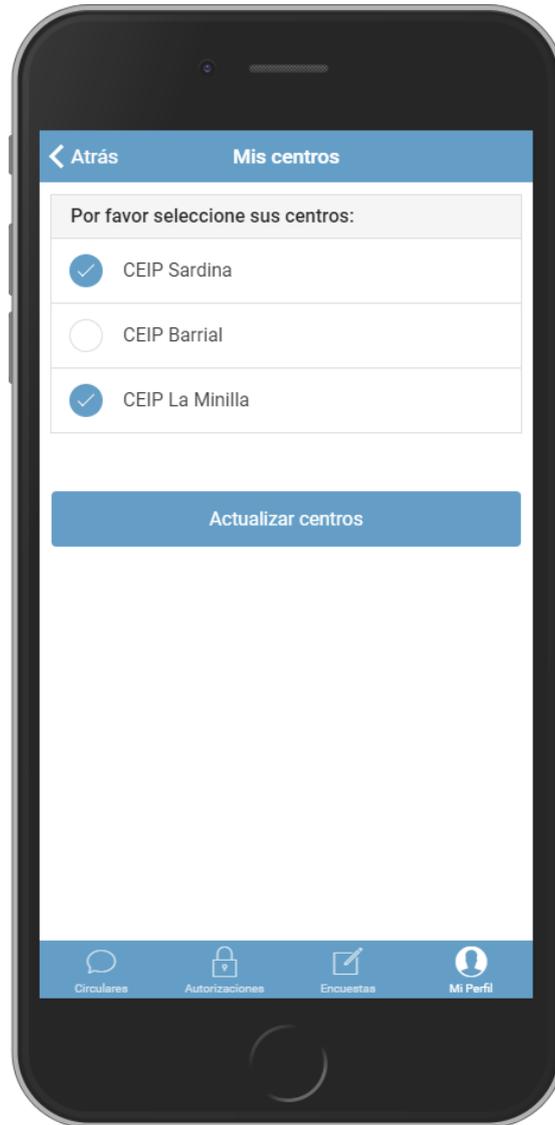


Ilustración 30: Mis centros

De igual manera que en el registro, podremos marcar y desmarcar los centros que deseemos de entre el listado de todos los centros del sistema, permitiendo sólo a los marcados visualizar nuestra información personal, tal y como se ve en la *Ilustración 30*.

Anexo II: Manual de instalación

En este anexo explicaremos de manera rápida como instalar la aplicación móvil en escritorio y la API en el *backend* del sistema. Por ello, se requiere previamente haber instalado el *backend* del sistema, el manual para ello se encuentra en el Anexo II del Trabajo de Fin de Grado titulado: “*Análisis, Diseño e Implementación de un Backend para la Comunicación entre Centros Educativos y Padres de Alumnos*”.

API

Para la instalación de la API Restful sólo debemos seguir los pasos de configuración previos señalados en el apartado 9.3.2 de esta memoria. Una vez hecho, simplemente copiaremos el directorio *api* del directorio abajo señalado y lo copiamos en la misma ruta. La localización del directorio es:

```
/Hermerest/src/AppBundle/Controler/
```

Y sustituimos copiamos el directorio *config* del directorio abajo señalado y lo copiamos en la misma ruta. La localización del directorio es:

```
src/AppBundle/Resources/
```

Aplicación móvil

Para la aplicación móvil requerimos de tener instalados en el sistema los siguientes componentes:

- Node.js
- Cordova
- Ionic

En caso de no tener dichos componentes instalados en el sistema, primeramente, instalaríamos Node.js descargándolo de su página web:

<https://nodejs.org/en/download/>

Una vez descargado e instalado, procedemos a la instalación de Cordova. Abrimos una ventana del terminal (*cmd* en el caso de Windows) y ejecutamos el siguiente comando:

```
npm install -g cordova
```

Una vez finalizada la instalación, continuaremos instalando Ionic ejecutando este comando:

```
npm install -g ionic
```

Una vez instalado todo el software requerido podemos ejecutar nuestra aplicación de ionic, abriendo el directorio donde esté localizada y ejecutando un terminal desde esa ubicación. En el mismo, ejecutaremos el siguiente comando:

```
npm install -g ionic
```

Y si todo funciona correctamente, deberíamos poder ver nuestra aplicación funcionando en el explorador web.

Si al intentar arrancar el servidor de pruebas locales de Ionic nos diera un error solicitando la ruta del *gulp file* simplemente deberemos ejecutar desde el directorio de la aplicación la siguiente secuencia de comandos para solucionarlo:

```
npm update
```

```
npm remove gulp-sass
```

```
npm install gulp-sass --save-dev
```

Anexo III: Script para la base de datos

```
DROP DATABASE IF EXISTS hermerest;
CREATE DATABASE hermerest
DEFAULT CHARACTER SET utf8
DEFAULT COLLATE utf8_general_ci;
CREATE USER IF NOT EXISTS hermerest;
GRANT ALL ON hermerest.* to 'hermerest'@'localhost' IDENTIFIED BY
'hermerest';
CREATE TABLE centre(
id int AUTO_INCREMENT,
name VARCHAR(255) NOT NULL,
PRIMARY KEY(id)
);
CREATE TABLE administrator(
id int AUTO_INCREMENT,
user VARCHAR(255) NOT NULL UNIQUE,
password VARCHAR(32) NOT NULL,
name VARCHAR(255) NOT NULL,
centre int NOT NULL,
PRIMARY KEY(id),
FOREIGN KEY(centre) REFERENCES centre(id) ON DELETE CASCADE
);
CREATE TABLE class(
id int AUTO_INCREMENT,
name VARCHAR(255) NOT NULL UNIQUE,
centre int NOT NULL,
PRIMARY KEY(id),
UNIQUE (name,centre),
FOREIGN KEY(centre) REFERENCES centre(id) ON DELETE CASCADE
);
CREATE TABLE student(
id int AUTO_INCREMENT,
name VARCHAR(255) NOT NULL,
```

```

surname VARCHAR(255) NOT NULL,
class int,
centre int NOT NULL,
PRIMARY KEY(id),
FOREIGN KEY(class) REFERENCES class(id) ON DELETE SET NULL
FOREIGN KEY(centre) REFERENCES centre(id) ON DELETE CASCADE
);
CREATE TABLE parent(
id int AUTO_INCREMENT,
name VARCHAR(255) NOT NULL,
telephone VARCHAR(255) NOT NULL UNIQUE,72
PRIMARY KEY(id)
);
CREATE TABLE student_parent(
student int NOT NULL,
parent int NOT NULL,
PRIMARY KEY(student, parent),
FOREIGN KEY(student) REFERENCES student(id) ON DELETE CASCADE,
FOREIGN KEY(parent) REFERENCES parent(id) ON DELETE CASCADE
);
CREATE TABLE centre_parent(
centre int NOT NULL,
parent int NOT NULL,
PRIMARY KEY(centre, parent),
FOREIGN KEY(centre) REFERENCES centre(id) ON DELETE CASCADE,
FOREIGN KEY(parent) REFERENCES parent(id) ON DELETE CASCADE
);
CREATE TABLE message(
id int AUTO_INCREMENT,
subject VARCHAR(255) NOT NULL,
message TEXT NOT NULL,
sendingDate TIMESTAMP NOT NULL,
centre int NOT NULL,
type VARCHAR(255) NOT NULL,

```

```

PRIMARY KEY(id),
FOREIGN KEY(centre) REFERENCES centre(id) ON DELETE CASCADE
);
CREATE TABLE authorization(
id int,
limitDate TIMESTAMP NOT NULL,
PRIMARY KEY(id),
FOREIGN KEY(id) REFERENCES message(id) ON DELETE CASCADE
);
CREATE TABLE circular(
id int,
PRIMARY KEY(id),
FOREIGN KEY(id) REFERENCES message(id) ON DELETE CASCADE
);
CREATE TABLE poll(
id int,
limitDate TIMESTAMP NOT NULL,
multipleChoice TINYINT(1) NOT NULL;
PRIMARY KEY(id),
FOREIGN KEY(id) REFERENCES message(id) ON DELETE CASCADE
);
CREATE TABLE pollOption(
id int AUTO_INCREMENT,73
text VARCHAR(255) NOT NULL,
poll int NOT NULL,
PRIMARY KEY(id),
FOREIGN KEY(poll) REFERENCES poll(id) ON DELETE CASCADE
);
CREATE TABLE message_student(
message int NOT NULL,
student int NOT NULL,
PRIMARY KEY(message, student),
FOREIGN KEY(student) REFERENCES student(id) ON DELETE CASCADE,
FOREIGN KEY(message) REFERENCES message(id) ON DELETE CASCADE

```

```

);
CREATE TABLE pollReply(
  id int AUTO_INCREMENT,
  parent int,
  pollOption int,
  PRIMARY KEY(id),
  UNIQUE(parent, pollOption),
  FOREIGN KEY(parent) REFERENCES parent(id) ON DELETE CASCADE,
  FOREIGN KEY(pollOption) REFERENCES pollOption(id) ON DELETE
  CASCADE
);
CREATE TABLE authorizationReply(
  id int AUTO_INCREMENT,
  authorization int,
  parent int,
  student int,
  authorized boolean NOT NULL,
  PRIMARY KEY(id),
  UNIQUE(parent, authorization, student),
  FOREIGN KEY(authorization) REFERENCES authorization(id) ON DELETE
  CASCADE
  FOREIGN KEY(parent) REFERENCES parent(id) ON DELETE CASCADE,
  FOREIGN KEY(student) REFERENCES student(id) ON DELETE CASCADE,
);
CREATE TABLE attachment(
  id int AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  message int NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY(message) REFERENCES message(id) ON DELETE CASCADE
);

```

