



Desarrollo de una APP para gestión de turnos de conducción para profesores

Memoria de Proyecto Final de Máster

Máster en Aplicaciones Multimedia

Autor: Diego Cordoba Aguirre

Consultor: Sergio Schvarstein Liuboschetz

PRA: Laura Porta Simó

10/junio/2018

Créditos/Copyright



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada

[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Desarrollo de una app para gestión de turnos de conducción para profesores</i>
Nombre del autor:	<i>Diego Cordoba Aguirre</i>
Nombre del consultor/a:	<i>Sergio Schvarstein Liuboschetz</i>
Nombre del PARA:	<i>Laura Porta Simó</i>
Fecha de entrega (mm/aaaa):	<i>06/2018</i>
Titulación::	<i>Master en Aplicaciones Multimedia</i>
Área del Trabajo Final:	<i>Trabajo fin de master</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Coche, turno, horario, profesores, app hibrida, phonegap, jquery mobile.</i>
Resumen del Trabajo:	
<p>Este proyecto ha tenido como objetivo dar solución a la creación y gestión de las llamadas "ruedas de coche" que se organizan todos los años en los institutos al inicio del curso escolar. En todos los institutos existen grupos de profesores que comparten el coche para ir al trabajo y, dado que las horas de entrada y salida no coinciden en todo los casos, se organiza una "rueda de coches", es decir, un cuadrante de que día se tiene que llevar el coche cada uno, a quien tiene que recoger y donde. Esto se complica cuanto mas profesores intervienen en la rueda o mas dispares sean los horarios y los lugares de partida.</p> <p>Para desarrollar la propuesta se ha seguido la metodología SCRUM, se ha usado JQuery Mobile como principal framework de desarrollo y Phonegap para empaquetar el código y generar una app híbrida.</p> <p>Como resultado se ha obtenido una app híbrida que permite a los profesores registrar su horario en el movil y compartirlo con el resto de usuarios para crear la rueda de forma automática.</p> <p>Este proyecto me ha permitido aprender el desarrollo de aplicaciones híbridas, así como diseñar aplicaciones web de una sola página como JQM o la gestión de datos con base de datos en un servidor RestFul.</p>	
Abstract :	
<p>The aim of this project was to provide a solution to the creation and management of the so-called "car wheels" that are organized every year in the institutes at the beginning of the school year. In all the institutes there are groups of teachers who share the car to go to work and, given that the hours of entry and exit do not coincide in all cases, a "car wheel" is organized, that is, a quadrant of what day you have to take the car each one, whom you have to pick up and where. This becomes more complicated the more professors are involved in the wheel or the more disparate are the schedules and the places of departure.</p>	

In order to develop the proposal, the SCRUM methodology has been followed, JQuery Mobile has been used as the main development framework and Phonegap has been used to package the code and generate a hybrid app.

As a result, a hybrid app has been obtained that allows teachers to register their time on the phone and share it with the other users to create the wheel automatically.

This project has allowed me to learn the development of hybrid applications, as well as design single-page web applications such as JQM or database data management in a RestFul server.

Cita

“No fracasé, sólo descubrí 999 maneras de cómo no hacer una bombilla” Thomas A. Edison

Abstract

The aim of this project was to provide a solution to the creation and management of the so-called "car wheels" that are organized every year in the institutes at the beginning of the school year. In all the institutes there are groups of teachers who share the car to go to work and, given that the hours of entry and exit do not coincide in all cases, a "car wheel" is organized, that is, a quadrant of what day you have to take the car each one, whom you have to pick up and where. This becomes more complicated the more professors are involved in the wheel or the more disparate are the schedules and the places of departure.

In order to develop the proposal, the SCRUM methodology has been followed, JQuery Mobile has been used as the main development framework and Phonegap has been used to package the code and generate a hybrid app.

As a result, a hybrid app has been obtained that allows teachers to register their time on the phone and share it with the other users to create the wheel automatically.

This project has allowed me to learn the development of hybrid applications, as well as design single-page web applications such as JQM or database data management in a RestFul server.

Resumen

Este proyecto ha tenido como objetivo dar solución a la creación y gestión de las llamadas **"ruedas de coche"** que se organizan todos los años en los institutos al inicio del curso escolar. En todos los institutos existen grupos de profesores que comparten el coche para ir al trabajo y, dado que las horas de entrada y salida no coinciden en todo los casos, se organiza una "rueda de coches", es decir, un cuadrante de que día se tiene que llevar el coche cada uno, a quien tiene que recoger y donde. Esto se complica cuanto mas profesores intervienen en la rueda o mas dispares sean los horarios y los lugares de partida.

Para desarrollar la propuesta se ha seguido la metodología SCRUM, se ha usado JQuery Mobile como principal framework de desarrollo y Phonegap para empaquetar el código y generar una app híbrida.

Como resultado se ha obtenido una app híbrida que permite a los profesores registrar su horario en el movil y compartirlo con el resto de usuarios para crear la rueda de forma automática.

Este proyecto me ha permitido aprender el desarrollo de aplicaciones híbridas, así como diseñar aplicaciones web de una sola página como JQM o la gestión de datos con base de datos en un servidor RestFul.

Palabras clave

Keywords del trabajo separadas por comas. Por ejemplo para éste documento podrían ser Modelo, Pauta, Memoria, Trabajo de Final de Grado/Máster

Índice

Capítulo 1: Introducción	11
1. Introducción	11
2. Descripción	12
3. Objetivos generales	14
3.1. User story map	14
3.2 Objetivos principales	14
3.3 Alcance	15
4. Metodología y proceso de trabajo	16
5. Planificación	18
6. Presupuesto	19
7. Estructura del resto del documento	20
Capítulo 2: Estado del arte	21
Capítulo 3: Diseño	25
1. Arquitectura general de la aplicación	25
2. Arquitectura de la información y diagramas de navegación	27
3. Diseño gráfico e interfaces	30
3.1 Estilos	30
3.2 Wireframes	31
Capítulo 4: Implementación	35
Capítulo 5: Demostración	43
Capítulo 6: Conclusiones y líneas de futuro	46
1. Conclusiones	46
2. Líneas de futuro	47
Bibliografía	48
1. Anexos	49
Anexo A: Entregables del proyecto	49
Anexo B: Diseños y diagramas	49

Figuras y tablas

Lista de imágenes, tablas, gráficos, diagramas, etc., numeradas, con títulos y las páginas en las que aparecen.

Índice de figuras

Figura 1: Ejemplo de la situación de partida	12
Figura 2: Calendario en Calendario de Turnos, MiPlanilla y Saturnos.....	21
Figura 3: Turno nuevo en CuadraTurno, secuencias en Calendario de Turnos	22
Figura 4: Menú lateral en MiPlanilla, Saturnos y menu de cabecera en CuadraTurnos.	23
Figura 5: Estadísticas en MiPlanilla y Calendario de Turnos.....	24
Figura 6: Relación entre los componentes software	25
Figura 7: Esquema de funcionamiento	26
Figura 8: Diagrama de navegación.....	27
Figura 9: Rueda en formato JSON	28
Figura 10: Recorrido de los datos	29
Figura 11: Themroller.....	30
Figura 12: Logotipo.....	30
Figura 13: Wireframes: crear una rueda	31
Figura 14: Wireframes: unirse a una rueda.....	32
Figura 15: Wireframes: visualizaciones	33
Figura 16: Wireframes: menú.....	34
Figura 17: Esquema funcionamiento phonegap/cordova	36
Figura 18: Calculo de la rueda	42
Figura 19: Pantallas 1	43
Figura 20: Pantallas 2	44
Figura 21: Pantallas3	45

Índice de tablas

Tabla 1: Planificación.....	18
Tabla 2: Presupuesto.....	19

Capítulo 1: Introducción

1. Introducción

En todos los institutos existen grupos de profesores que comparten el coche para ir al trabajo y dado que las horas de entrada y salida no coinciden en todo los casos, se organiza una “rueda de coches”, es decir, un cuadrante de que día se tiene que llevar el coche cada uno, a quien tiene que recoger y donde.

Según mi experiencia personal, en todos los centros se manufactura una rueda de coches usando algún paquete ofimático y después se distribuye a los interesados. Esta tarea suele delegarse en una sola persona, que es marcada como responsable de la rueda. Durante el curso, las modificaciones de la rueda se comunican por medio de aplicaciones de mensajería o de viva voz.

La motivación de este proyecto es la de facilitar la gestión de la rueda centralizando todas las tareas en una sola aplicación. El proyecto aborda dos tareas principales:

- Generación del cuadrante.
- Gestión del cuadrante y comunicación entre los integrantes de la misma.

Para desarrollar la propuesta se utilizará las siguientes tecnologías:

- JQuery Mobile como principal framework de desarrollo.
- Phonegap para empaquetar el código y generar una APP híbrida.

Esta aplicación no tendrá relevancia para el público general, pero sí para el grupo en el que se focaliza, ya que realiza una tarea específica relacionada con la profesión que todavía no está cubierta por ninguna aplicación. Claramente, la APP tendrá un breve periodo de uso intensivo al inicio de curso y un periodo de simple consulta o notificación puntual a lo largo del curso.

2. Descripción

En el apartado D de la **orden 02/07/2012 de Castilla-La Mancha [1]** se concretan las normas para componer el horario del profesorado. En líneas generales son veintiuna horas lectivas y cinco horas complementarias de obligada permanencia en el centro. Al dividirse los días en 6 sesiones diarias, que serían 30 semanales, la normativa da lugar a cuatro “huecos” en el horario que normalmente se ubican al inicio o al final de varios días. Esto da lugar a horarios dispares entre los profesores que comparten coche para ir a trabajar.

En la siguiente imagen podemos ver un ejemplo de la situación que se está abordando. A la izquierda se muestra la coincidencia de dos horarios diferentes y a la derecha un cuadrante hecho con un paquete ofimático (el cuadrante se incluye en el anexo B para una mejor visualización

	L	M	X	J	V
Horario de María					
1	Lec.	Lec.	Lec.	Com.	Lec.
2	Lec.	Com.	Lec.	Lec.	Lec.
3	Com.	Lec.	Lec.	Com.	Lec.
4	Lec.	Lec.	Lec.	Lec.	Lec.
5	Lec.	Lec.		Lec.	Lec.
6	Lec.	Lec.		Lec.	Lec.

	L	M	X	J	V
Horario de Jesús					
1	Lec.	Lec.	Lec.	Com.	Lec.
2	Lec.	Com.	Lec.	Lec.	Lec.
3	Com.	Lec.	Lec.	Com.	Lec.
4	Lec.	Lec.	Lec.	Lec.	Lec.
5	Lec.	Lec.	Lec.		Lec.
6	Lec.	Lec.	Lec.		Lec.

HORAS	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
1ª 8:30 a 9:25	JESÚS JOSE LUIS MARIO (CF) PRADO LUIS MANUEL LUISFER IRENE INÉS	MARILUZ BASILIO MARIO (CF) JESÚS ANACRIS LUIS MANUEL JAVIER LUISFER INÉS IRENE JOSE ALBERTO AGUSTÍN MERCHE	BASILIO JESÚS ANACRIS MARILUZ JOSE ALBERTO AGUSTÍN LUISFER MARIO (HOT) JAVIER IRENE INÉS	MARILUZ LUIS MANUEL ANACRIS AGUSTÍN JOSE ALBERTO INÉS LUISFER	LUIS MANUEL PRADO BASILIO ANACRIS MARILUZ MARIO (CF) JOSE ALBERTO AGUSTÍN INÉS IRENE
2ª 9:25 a 10:20	MERCHE JAVIER JOSE ALBERTO BASILIO (HOT)	JOSE LUIS	MERCHE (CF) JOSE LUIS	MARIO (CF) JOSE LUIS BASILIO IRENE MERCHE JAVIER	JOSE LUIS JESÚS LUISFER MERCHE JAVIER
3ª 10:20 a 11:15	AGUSTÍN (CF) MARILUZ ANACRIS	PRADO	LUIS MANUEL PRADO	JESÚS PRADO	
4ª 11:15 a 11:45					
4ª 11:45 a 12:40		MARILUZ (**) AGUSTÍN (**)	BASILIO JOSE ALBERTO	AGUSTÍN JOSE ALBERTO	ANACRIS MARILUZ MARIO (CF) JOSE ALBERTO LUIS MANUEL
5ª 12:40 a 13:35	LUISFER IRENE JOSE ALBERTO BASILIO (HOT) JESÚS MARIO JOSE LUIS	IRENE LUISFER	ANACRIS MARILUZ JAVIER IRENE LUISFER INÉS		BASILIO (HOT) IRENE MERCHE
6ª 13:35 a 14:30	PRADO LUIS MANUEL MARILUZ ANACRIS MERCHE JAVIER INÉS AGUSTÍN	PRADO LUIS MANUEL BASILIO JESÚS JOSE LUIS (**) ANACRIS (**) JAVIER JOSE ALBERTO MERCHE (**) INÉS MARIO (**)	LUIS MANUEL PRADO JOSE LUIS JESÚS MERCHE AGUSTÍN MARIO (HOT)	MARILUZ LUIS MANUEL PRADO BASILIO JESÚS JOSE LUIS ANACRIS IRENE LUISFER MARIO (HOT) MERCHE INÉS JAVIER	JOSE LUIS PRADO JESÚS LUISFER AGUSTÍN INÉS JAVIER

Figura 1: Ejemplo de la situación de partida

Para facilitar esta tarea se pretende obtener al final el siguiente producto:

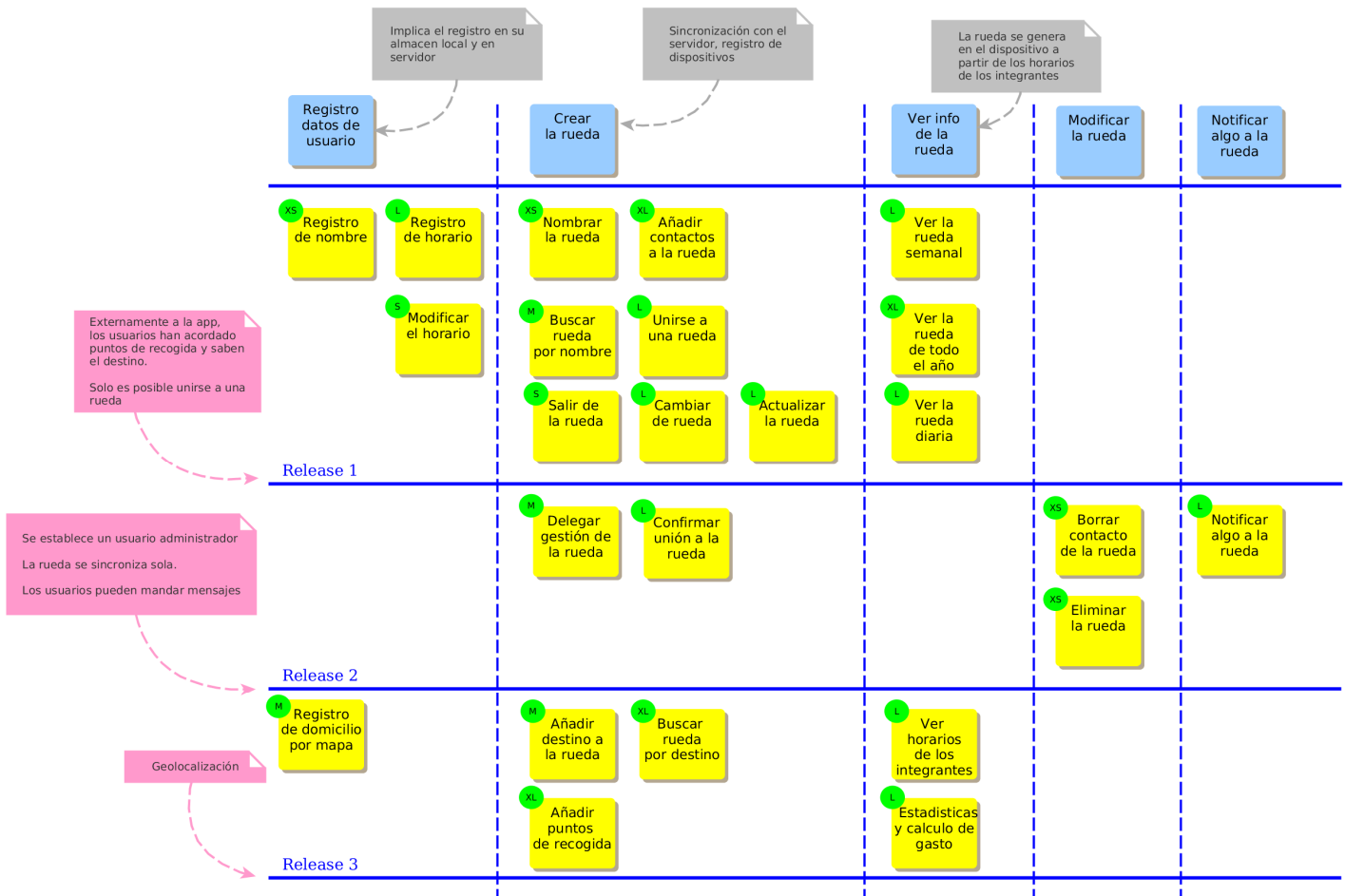
APP donde los profesores puedan registrar su horario en el móvil, compartirlo con otros profesores y generar de forma automática un cuadrante de conducción teniendo en cuenta los siguientes factores:

- Todos los profesores tienen diferentes horarios, que pueden coincidir en diferentes días.
- Las ruedas de coches pueden incluir más de cinco profesores, con lo que puede haber más de un coche cada día.
- Los pasajeros no tienen porque ir y venir en el mismo coche.
- En una misma rueda puede haber diferentes puntos de recogida y los integrantes de los coches son segregados según el punto de recogida.
- Pueden darse situaciones imprevistas o fortuitas durante el curso que obliguen a modificar la rueda.

3. Objetivos generales

3.1. User story map

Para determinar los objetivos y alcance de este proyecto se realizó un primer análisis de la aplicación por medio del desarrollo de un User Story Map siguiendo las directrices de **Jeff Patton**[2] (el mapa se incluye en el ANEXO B para una mejor visualización)



3.2 Objetivos principales

Una vez desarrollado el diagrama anterior se determinó que el proyecto solo desarrollaría el primer release.

Se plantean los siguientes objetivos de la aplicación:

- Registrar el horario del profesor en el móvil y en un servidor de forma automatizada.
- Compartir el horario entre profesores.
- Automatizar la generación de la rueda a partir de los horarios y lugares de partida compartidos entre los usuarios.
- Notificar a los usuarios las actualizaciones de la rueda.

Objetivos del autor:

- Aplicar una metodología ágil de desarrollo de software.
- Usar una tecnología para producción de app híbridas que permita un desarrollo rápido y con prestaciones cercanas al desarrollo de app nativas.
- Profundizar en el estudio del desarrollo de aplicaciones multimedia con tecnologías web y el uso de base de datos NoSQL en servidor.

3.3 Alcance

Teniendo en cuenta el tiempo establecido para el desarrollo de este proyecto, para hacer posible el alcance los objetivos anteriores se establecen las siguientes limitaciones:

- La app solo podrá almacenar una única rueda.
- Se limitará el desarrollo a los horarios de los profesores de secundaria.
- Solo tendrá en cuenta los horarios y no los lugares de partida.
- Se diseñará el interfaz gráfico con componentes estándar o prediseñados del framework de desarrollo siempre que sea posible.
- La actualización de la rueda será disparada por el usuario.

4. Metodología y proceso de trabajo

El proceso de trabajo se desarrollo en las siguientes fases:

1. Estudio del arte

Después del *estudio del arte*, que más adelante se expondrá, se hizo patente que la mejor opción era desarrollar un producto nuevo partiendo de cero, pues no existía ningún producto anterior del que se pudiera partir.

2. Guías de desarrollo

Se impusieron como guías de desarrollo las siguientes:

- Simplicidad de las interfaces. Uso de componentes preconstruidos del framework en medida de lo posible.
- Recoger la mínima información necesaria para llevar a cabo la tarea.
- Caminos de navegación con la menor cantidad de bucles posibles.

3. Determinación de la metodología

Para llevar a cabo el desarrollo se ha utilizado la **metodología ágil**, siguiendo las líneas generales de **SRUM** [3]. Se definieron como sprints cada una de las PECs de este proyecto de forma que siempre que se entregara la aplicación fuera un producto funcional, aunque limitado.

4. User story map

El primer paso del desarrollo fuera la determinación de las historia de usuario y la estimación de las tareas, lo cual se hizo por medio de un **user story map** que se anexa a esta documentación, siguiendo las directrices expuestas por **Jeff Patton** [2]. La estimación de las tareas es hizo con la **técnica de las camisetas** [4].

Una vez determinada la estimación se determinó que funcionalidades tendría el producto al final, que son las que se incluyen en el primer release del user story map. Este se ha ido modificando según ha ido avanzando el proyecto.

5. Planificación

Con la estimación terminada se determinó una planificación del desarrollo, que se fue modificando para resolver las eventualidades y que se refleja en el apartado siguiente. Esta planificación dividió el desarrollo en tres hitos importantes, donde cada hito tenía como producto una APP operativa, aunque limitada en prestaciones.

6. Caminos de navegación

Se desarrolló el diagrama de navegación que se incluye en el ANEXO B, que sirvió para reflexionar sobre una primera aproximación a la solución. Este diagrama se ha ido modificando desde su primera versión según ha avanzado el desarrollo de app.

7. Esquema de funcionamiento.

El esquema de funcionamiento (incluido en el ANEXO B) sirvió para reflexionar como se tendrían que relacionar los diferentes dispositivos con el servidor, cuál sería la secuencia para compartir los datos y como se generaría la rueda.

Gracias a este esquema se llegó a la conclusión de que no era necesario guardar las rotaciones generadas para cada usuario en el servidor, sino que sería más sencillo generarlas a partir del horario cada vez que se encienda la aplicación. De esta forma reducimos la cantidad de información que debemos almacenar en el servidor y también la complejidad. Hay que tener en cuenta que las rotaciones son personalizadas para cada profesor y para todo el año.

8. Breve estudio sobre Phonegap/Cordova y JQuery Mobile

Una vez determinado que es lo se quería hacer, se estudiaron las tecnologías que se utilizarían, para determinar cuál era la forma adecuada de emplearlas. Todo el desarrollo del proyecto le ha servido al autor a profundizar en estas tecnologías.

9. Montaje del entorno de desarrollo

Se instalaron todos los paquetes necesario para utilizar Phonegap/Cordova, se desacargaron las librerías necesarias para desarrollar el código con JQuery Mobile y se sincronizó todo para poder depurar usando Android Studio y construir la app.

10. Wireframes y color

Se desarrollaron varios Wireframes de las principales pantallas que tendría la aplicación. Una vez hecho esto se utilizó la aplicación themeroller para determinar los colores que se usarían.

11. Implementación.

Como se ha dicho la implementación siguió tres fases, en cada una de las cuales se entregó un producto funcional, pero limitado.

12. Memoria

Finalizado el desarrollo se realizó esta memoria.

5. Planificación

A continuación se muestra la planificación que se ha seguido en el desarrollo del proyecto. (Uno de los consejos que se da **Rodriguez**[5] es utilizar cojines de tiempo de un 10-20% del total, de tal forma que se ha añadido en cada hito un cojín de dedicación que permita corregir desfases en la planificación.)

Actividad	Inicio	Fin	Días	Horas
PEC 3. Generar la rueda				
Registrar el usuario en el dispositivo	27/03/18	29/03/18	3	6
Registrar el horario en el dispositivo	30/03/18	03/04/18	5	10
Nombrar la rueda	04/04/18	04/04/18	1	2
Añadir contactos a la rueda (solo local)	05/04/18	09/04/18	5	10
Generación de la rueda	10/04/18	14/04/18	5	10
Visualización de la rueda diaria	15/04/18	19/04/18	5	10
Cojín de dedicación	20/04/18	23/04/18	4	8
	27/03/18	23/04/18	28	46
PEC 4. Añadir contactos a la rueda (con servidor)				
Uso de restdb.io como BD RestFull	24/04/18	25/04/18	2	4
Gestión de la rueda en el servidor	26/04/18	30/04/18	5	10
Recuperar la rueda del servidor e integrarla	1/05/18	2/05/18	2	4
Buscar y unirse a una rueda	3/05/18	5/05/18	3	6
Cambiar rueda	6/05/18	6/05/18	1	2
Actualizar rueda	7/05/18	8/05/18	2	4
Abandonar la rueda	9/05/18	9/05/18	1	2
Rueda semanal	10/05/18	17/05/18	8	16
Cojín de dedicación	18/05/18	21/05/18	4	8
	24/04/18	21/04/18	28	46
PEC 5. Finalización y memoria				
Ver la rueda de todo el año	22/05/18	27/05/18	6	12
Desarrollo de la memoria	28/05/18	02/06/18	6	12
Preparación presentación	03/06/18	08/06/18	6	12
Cojín de dedicación	09/06/18	11/06/18	3	6
	22/05/18	11/06/18	21	42
			76	152

Tabla 1: Planificación

6. Presupuesto

A continuación se muestra el presupuesto de coste de desarrollo de la aplicación hasta su estado actual.

Partida	Concepto	Horas	Precio /hora (*)	Total
Análisis y diseño	User story map	10	25 €	250 €
	Estudio de las tecnologías necesarias	5	25 €	125 €
	Arbol de navegación	2	25 €	50 €
	Wireframes	10	25 €	250 €
	Colores del tema	10	25 €	250 €
				925 €
Primera entrega <i>La APP almacena usuarios y genera la rueda. Sin servidor</i>	Registrar el usuario en el dispositivo	18	25 €	450 €
	Añadir contactos a la rueda (solo local)	10	25 €	250 €
	Generación de la rueda	10	25 €	250 €
	Visualización de la rueda diaria	20	25 €	500 €
				1450 €
Segunda entrega <i>La APP almacena los datos en un servidor externo</i>	Gestión de la rueda en el servidor	30	25 €	750 €
	Buscar y unirse a una rueda	6	25 €	150 €
	Cambiar, actualizar, abandonar la rueda	16	25 €	400 €
	Visualización de la rueda semanal	24	25 €	600 €
				1900 €
Última entrega	Control de errores de usuario	10	25 €	250 €
	Visualización de la rueda anual	25	25 €	625 €
				875 €
			Total	5150 €

* Se ha tenido en cuenta la experiencia personal sobre el mercado actual para determinar el precio por hora.

Tabla 2: Presupuesto

7. Estructura del resto del documento

El resto del documento contiene los siguientes capítulos:

- Capítulo 2: Estado del arte. Donde se expone cual es la situación actual en el mercado en relación con el problema que nos ocupa.
- Capítulo 3: Diseño. Se expone la arquitectura general de la aplicación, las estructuras de datos usadas y el diseño de las interfaces.
- Capítulo 4: Implementación. En el que se resaltan las decisiones relativas a la implementación más importantes, como las tecnologías usadas, problemas y algoritmos.
- Capítulo 5: Demostración. En la que se pueden ver las principales pantallas del producto final.
- Capítulo 6: Conclusión y líneas de futuro. Se dan las impresiones finales y se habla sobre posibles líneas de actuación para mejorar la aplicación.

Capítulo 2: Estado del arte

En la actualidad no existe ninguna APP que se centre en solucionar el problema objeto de este proyecto, lo más parecido a lo que se quiere conseguir son APPs que nos permiten crear cuadrantes de trabajo, existiendo muchas APPs de este tipo en GooglePlay. Para este estudio se han seleccionado las primeras de las listas en cuanto a valoración, siendo las siguientes algunas de ellas:

- *SaTurnos*
- *CuadraTurnos*
- *MiPlanilla*
- *Calendario de turnos*

En el siguiente análisis se incluye no solo la descripción de como estas APPs resuelven determinados problemas, sino como estas inspiran la solución que se dará en nuestra AP.

Visualización de calendario

Todas las app muestran un calendario donde se pueden ver de un vistazo los turnos del mes, existiendo dos formas de navegar por los diferentes meses:

- Botones. Es la usada en *MiPlanilla*, donde podemos movernos de izquierda a derecha, cambiando de mes, por medio de los botones colocados arriba.
- Deslizamiento de la pantalla. En *Calendario de Turnos* se cambia de mes deslizando la pantalla de izquierda a derecha. En cambio en *SaTurnos* se puede desplazar la pantalla hacia arriba y van mostrándose los meses siguientes.

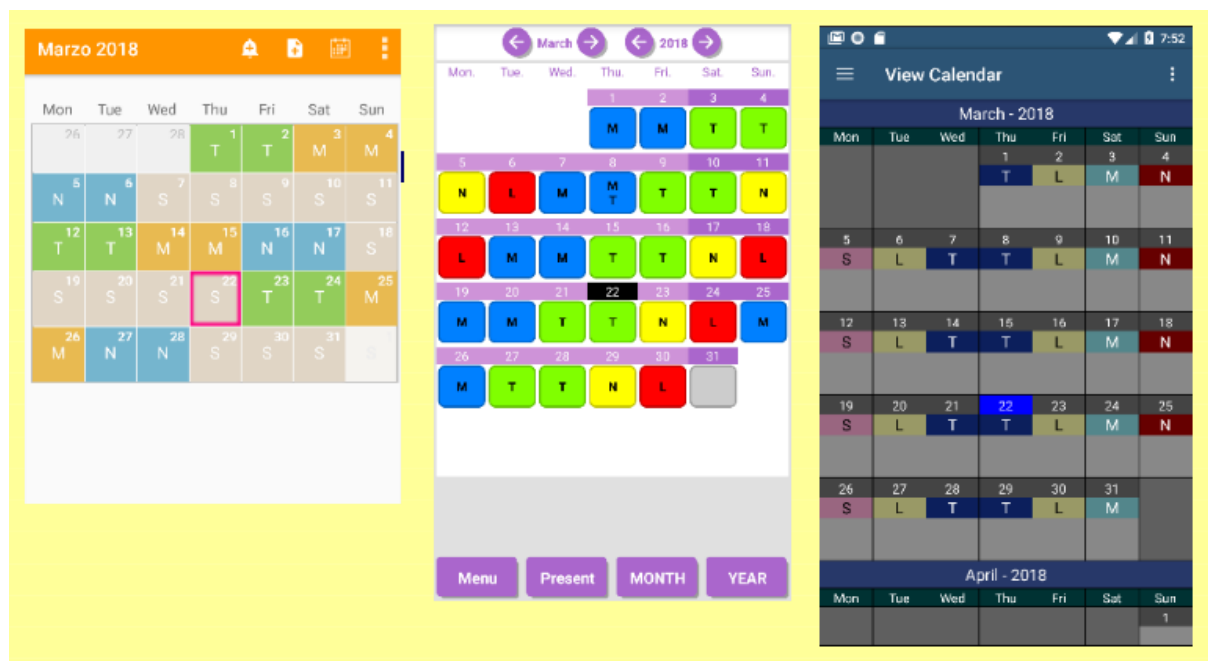


Figura 2: Calendario en Calendario de Turnos, MiPlanilla y Saturnos

Los turnos se simbolizan por medio de la primera letra del turno diario, existiendo la posibilidad en algunos de ellos de tener varios turnos en el mismo día, como podemos ver en el día 8 de marzo en

MiPlanilla. En la app que se pretende desarrollar los recuadros de los turnos irán rellenos con los nombres de los integrantes de cada día de la rueda, por lo que, debido al espacio necesario, la vista preferente será **semanal** en vez de mensual.

Creación de los turnos

Las apps analizadas ofrecen las siguientes posibilidades para crear la secuencia de turnos:

- Crear un tipo de turno y asignarlo día a día. Permite crear turnos que no siguen una secuencia específica. En la imagen inferior podemos ver cual es la pantalla de creación de nuevos tipos de turno en *CuadraTurnos*. Una vez creado es posible aplicárselo a cualquier día del calendario.
- Aplicar una secuencia de turnos. Se crea una plantilla de secuencia o se utiliza una predefinida, que aplica los turnos a varios días de un golpe. En la imagen inferior podemos ver como *Calendario de Turnos* ofrece varias secuencias predefinidas como: T, T, M, M, N, N. Además este programa permite crear nuestra propia secuencia, como se ve en la captura de la derecha. La secuencia se crea concatenando tipo de turnos disponibles en la parte inferior, después se nombra la secuencia y se guarda. Finalmente se elige un día de comienzo para aplicar la secuencia.

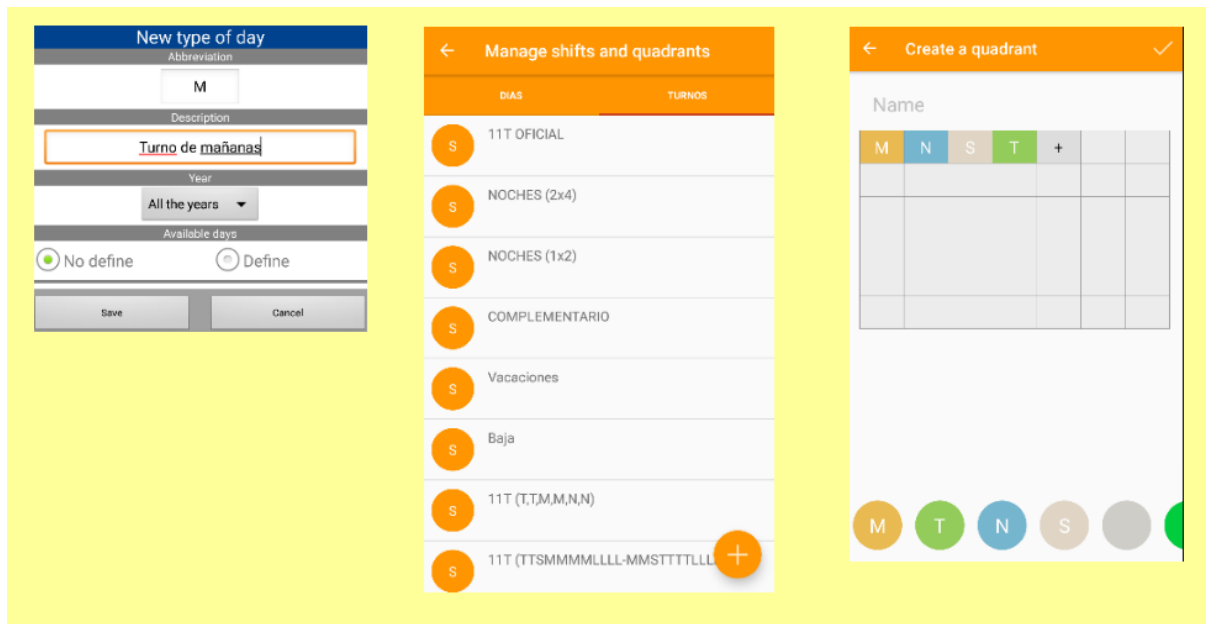


Figura 3: Turno nuevo en CuadraTurno, secuencias en Calendario de Turnos

En nuestro caso la secuencia debe generarse **automáticamente** teniendo en cuenta horarios de diferentes usuarios y puntos de recogida. Liberar al usuario de este trabajo es el objetivo principal de la aplicación.

Menú

La gestión del espacio disponible es un factor crucial en este tipo de dispositivos. Además la colocación del menú y sus opciones pueden a veces suponer el éxito de una APPs, pues el usuario puede desistir de usarla si no consigue encontrar las opciones en una interfaz poco amigable.

En las APPs analizadas se usan dos tipos de menú:

- En el encabezado. Está a la vista y es más accesible. Es la utilizada en *CuadraTurnos* como se ve en la imagen de abajo a la derecha. El problema con este tipo de menú es que disponemos de poco espacio y por lo tanto podemos ofrecer pocas opciones.
- Desplegable lateral. Deja libre el espacio al plegarse y puede contener mas opciones. Es el usado en *MiPlanilla* y *SaTurnos*. Como se puede ver permite mas opciones, nombres mas largos para las mismas e, incluso, secciones.

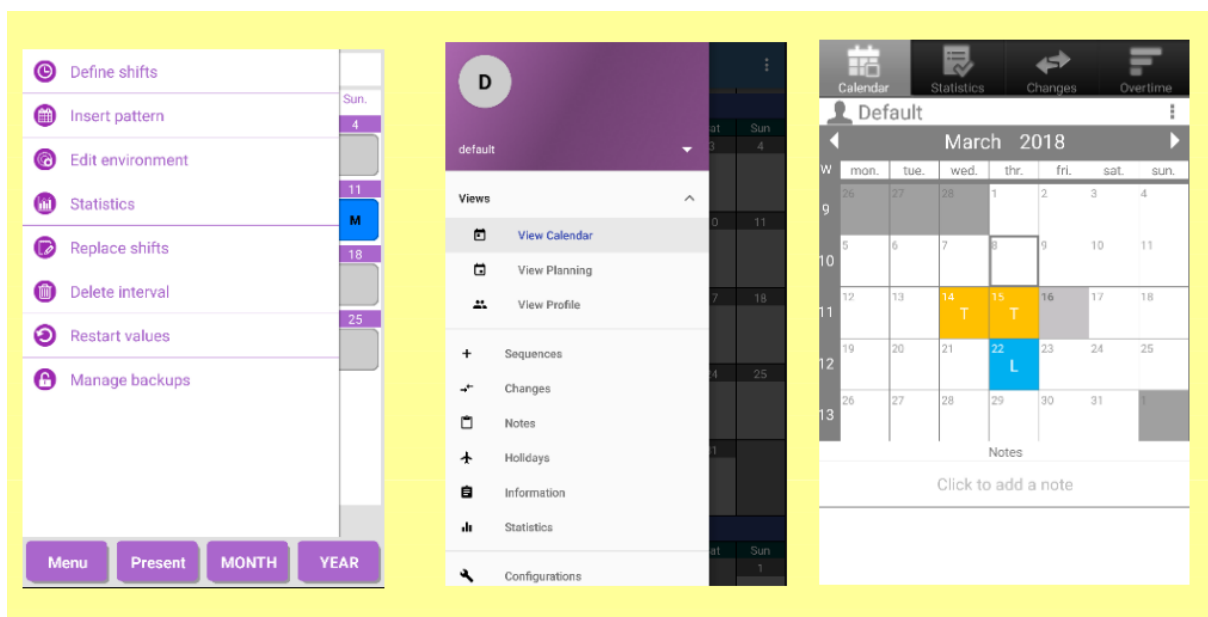


Figura 4: Menú lateral en MiPlanilla, Saturnos y menu de cabecera en CuadraTurnos.

Para nuestra app utilizaremos el menú lateral con un botón en la cabecera que lo haga visible si el usuario lo necesita.

Estadísticas

Casi todas las APPs analizadas incluyen estadísticas sobre el resumen de tipos de jornada que se han hecho, como se puede ver en las imágenes. En el caso de *MiPlanilla* se muestra un diagrama y la estadística de los turnos diurnos, nocturnos, libres, etc.

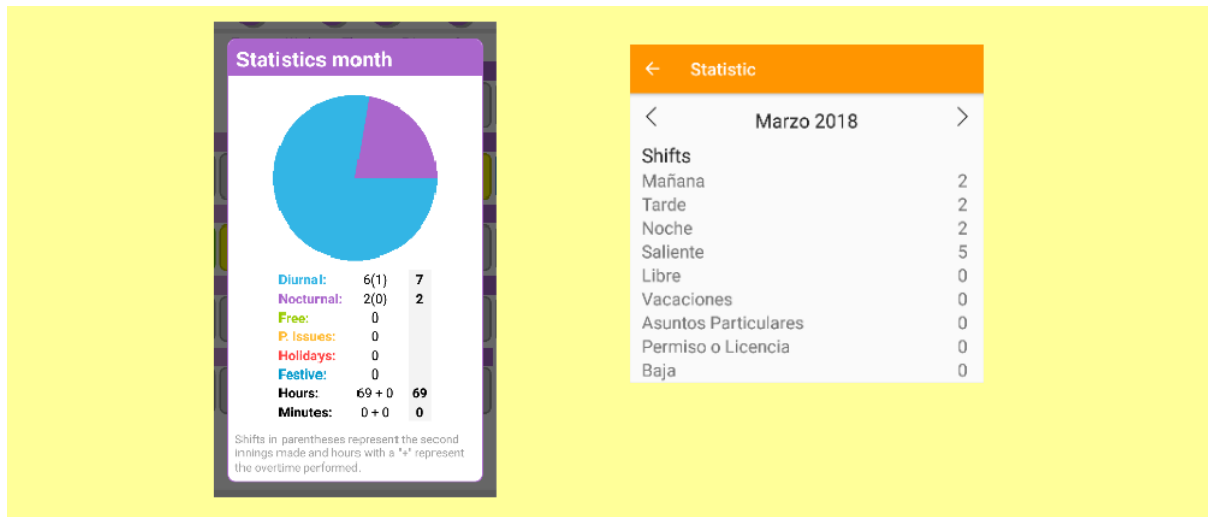


Figura 5: Estadísticas en MiPlanilla y Calendario de Turnos

En el caso que nos ocupa se podrían incluir estadísticas sobre: cantidad de veces que nos ha tocado conducir; kilómetros recorridos hasta ahora y kilómetros que nos quedan hasta acabar el curso; gasolina gastada y ahorrada en comparación con no compartir coche, etc.

Ninguna de las APPs analizadas permite cumplir el objetivo de este proyecto, que es representar los turnos de la rueda de coches.

Conclusión

En comparación con las anteriores, nuestra APP tiene las siguientes características diferenciadoras:

- Va a necesitar compartir información entre los usuarios.
- La generación de la secuencia de turnos es totalmente automática en función de los horarios de los integrantes.
- El fin último de nuestra APP es la coordinación de los usuarios, no solo el registro del turno.

Capítulo 3: Diseño

1. Arquitectura general de la aplicación

Las siguientes tecnologías y componentes son la base de la aplicación:

- **Phonegap/cordova** envuelve todo el software en un wrapper que permite utilizar una aplicación web en un dispositivo móvil, por medio de un webview, como si de una aplicación nativa se tratara.
- **JQueryMobile** es un framework para construir aplicaciones web en un solo archivo html dividido en secciones, denominadas paginas, que serán las que efectivamente harán la función de páginas web. JQM carga el código en DOM de forma dinámica en función de que página sea necesario visualizar. Además, está enfocada a crear una interfaz responsive por lo que provee gran cantidad de componentes estándar, como sliders, cabeceras, footers, etc. que permiten agilizar el desarrollo.
- **GestorPaginas.js** se ha desarrollado como controlador principal de la aplicación. Decide que pagina cargar en cada momento y hace la gestión de los datos por medio de los objetos que definen los dos componentes que se van a describir a continuación.
- **Profesor.js y Rueda.js** abstraen las necesidades de gestión de información en dos objetos que encapsulan la gestión del almacenamiento interno del dispositivo y en el servidor.

En la siguiente figura podemos ver la relación entre estos componentes:

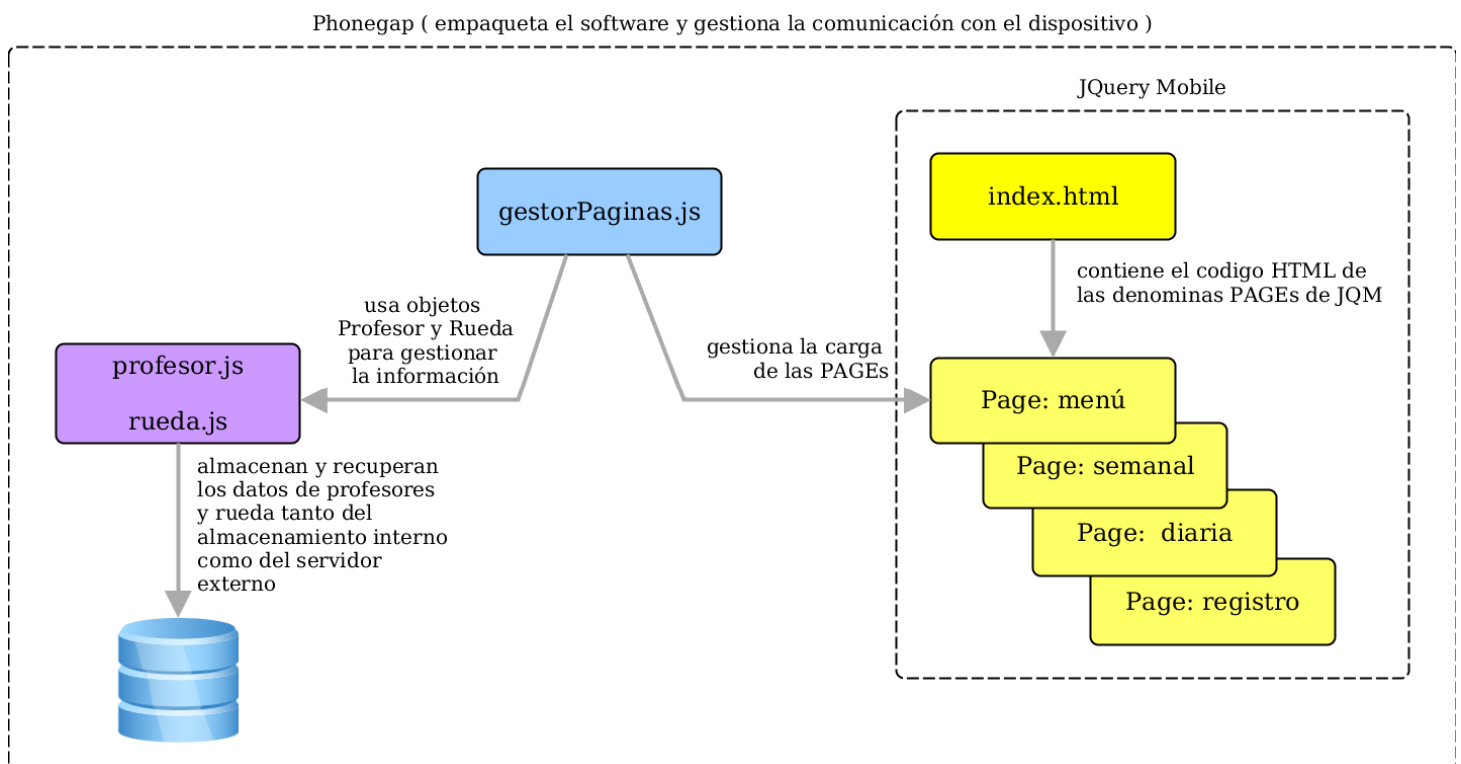


Figura 6: Relación entre los componentes software

En el siguiente diagrama se puede ver el esquema general de funcionamiento de la aplicación. Se incluye en ANEXO B para una mejor visualización.

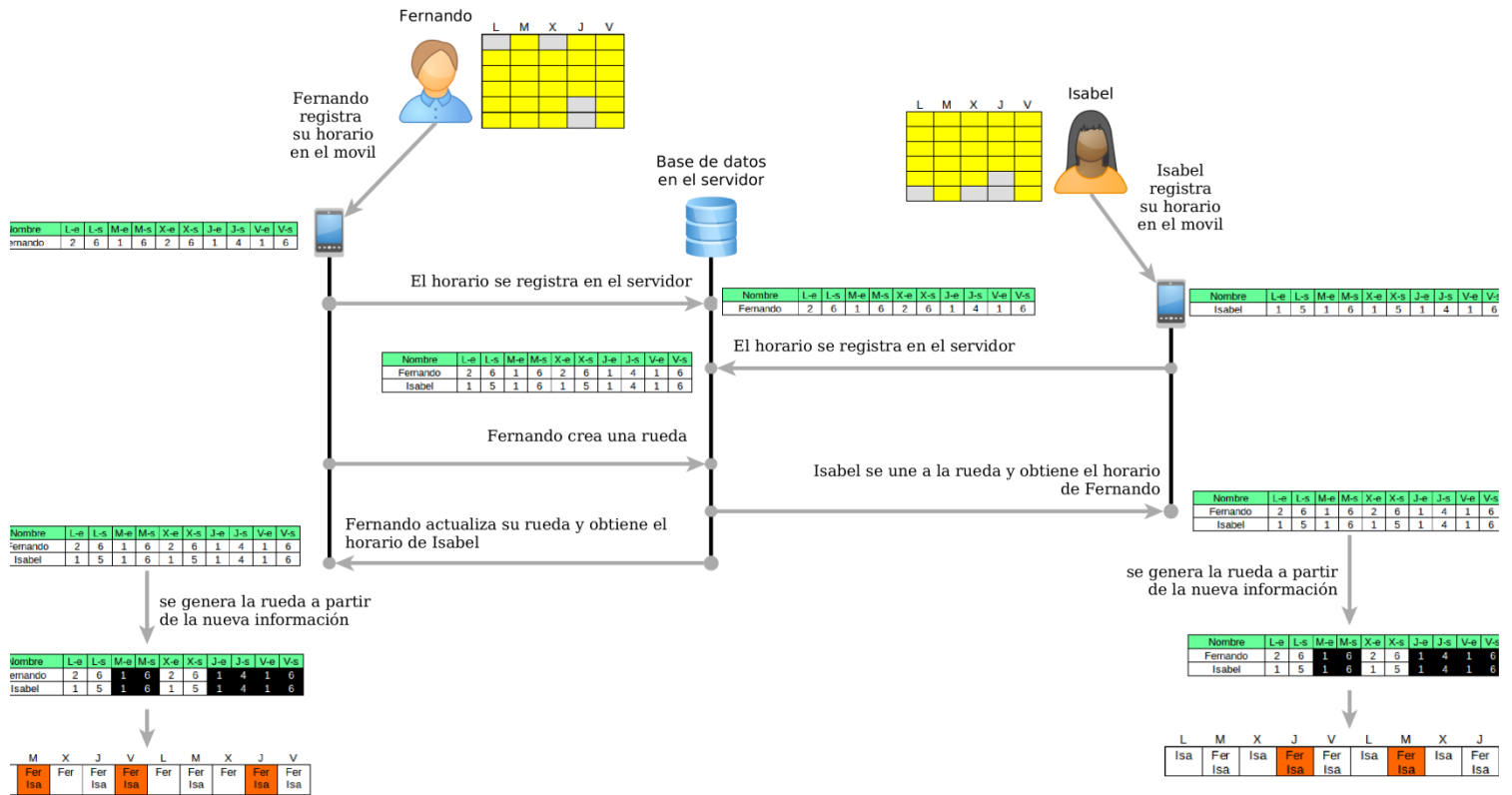


Figura 7: Esquema de funcionamiento

2. Arquitectura de la información y diagramas de navegación

Árbol de navegación

En siguiente diagrama se puede ver el árbol de navegación de las pantallas de la APP.

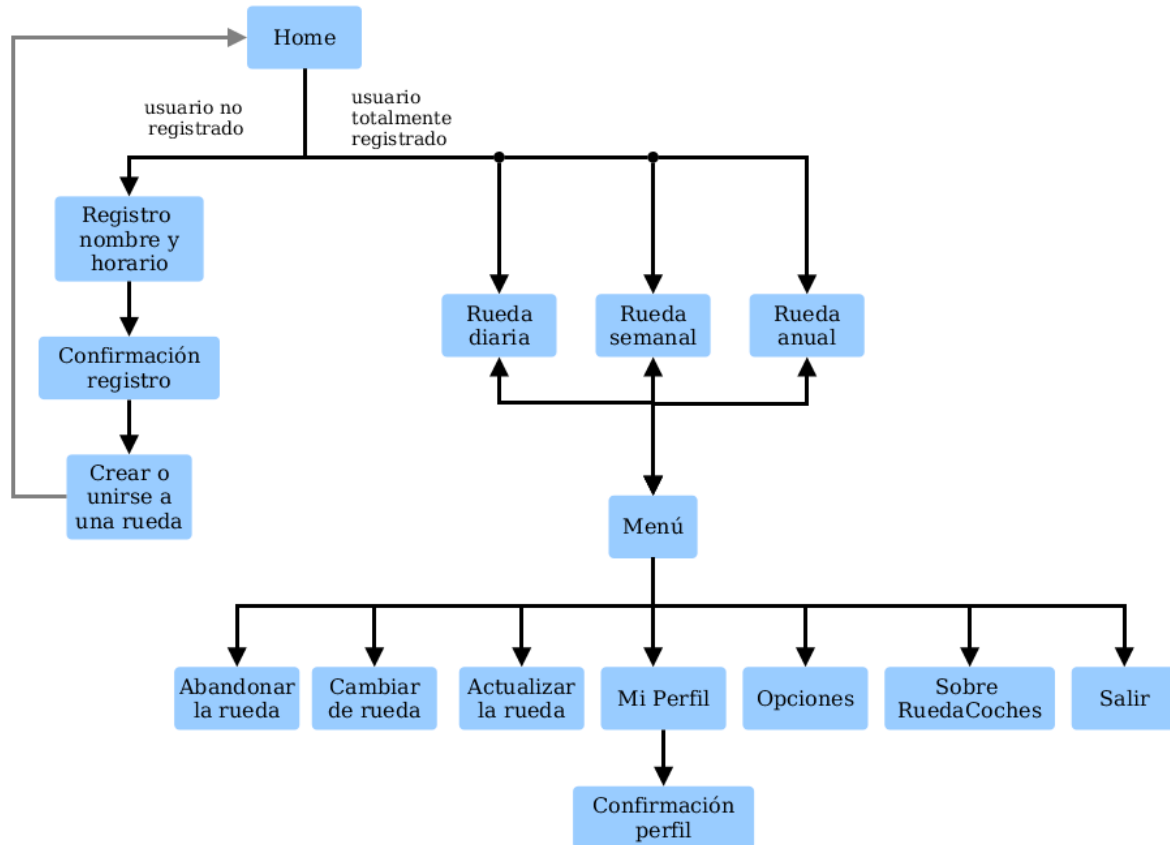


Figura 8: Diagrama de navegación

Almacenamiento

Toda la aplicación se basa en el almacenamiento de una sola estructura de datos que almacena los de los profesores con los siguientes campos:

- Nombre: nombre del profesor
- UUID: identificador del dispositivo
- Lunes_e, martes_e, miércoles_e, jueves_e y viernes_e: horas de entrada de lunes a viernes.
- Lunes_s, martes_s, miércoles_s, jueves_s y viernes_s: horas de salida de lunes a viernes.
- Rueda: nombre de la rueda a la que pertenece este profesor.
- IDDB: identificador del registro en base de datos NoSQL.

Estos campos son almacenados en diferentes espacios:

- Objetos profesor dentro del código de la aplicación.
- Localforage del dispositivo en formato JSON, para su posterior recuperación y creación de un objeto profesor.
- Servidor de BD Restful Resdb.io. La información se comparte por medio del acceso de los diferentes profesores a esta BD para obtener el horario de otros profesores.

La información sobre los horarios de todos los componentes se guarda dentro del dispositivo en un objeto Rueda y en el localforage en formato JSON. Un ejemplo de una rueda puede verse en la siguiente imagen.

```
{
  "titulo": "Puertollano",
  "miembros":
  [
    {
      "nombre": "diego",
      "UUID": 45643737,
      "lunes_e": 1,
      "martes_e": 1,
      "miercoles_e": 1,
      "jueves_e": 2,
      "viernes_e": 1,
      "lunes_s": 6,
      "martes_s": 4,
      "miercoles_s": 6,
      "jueves_s": 6,
      "viernes_s": 6,
      "rueda": "Puertollano",
      "idBD": 89898514634
    },
    {
      "nombre": "Federico",
      "UUID": 231455155,
      "lunes_e": 2,
      "martes_e": 1,
      "miercoles_e": 1,
      "jueves_e": 1,
      "viernes_e": 1,
      "lunes_s": 5,
      "martes_s": 6,
      "miercoles_s": 6,
      "jueves_s": 5,
      "viernes_s": 6,
      "rueda": "Puertollano",
      "idBD": 32141251514634
    }
  ]
}
```

Figura 9: Rueda en formato JSON

Dentro del **ANEXO B** el archivo **recorridoDeLosDatos** resume el viaje que hace la información en el sistema y las diversas transformaciones de formato necesarias para poder almacenarlas en los diferentes contenedores.

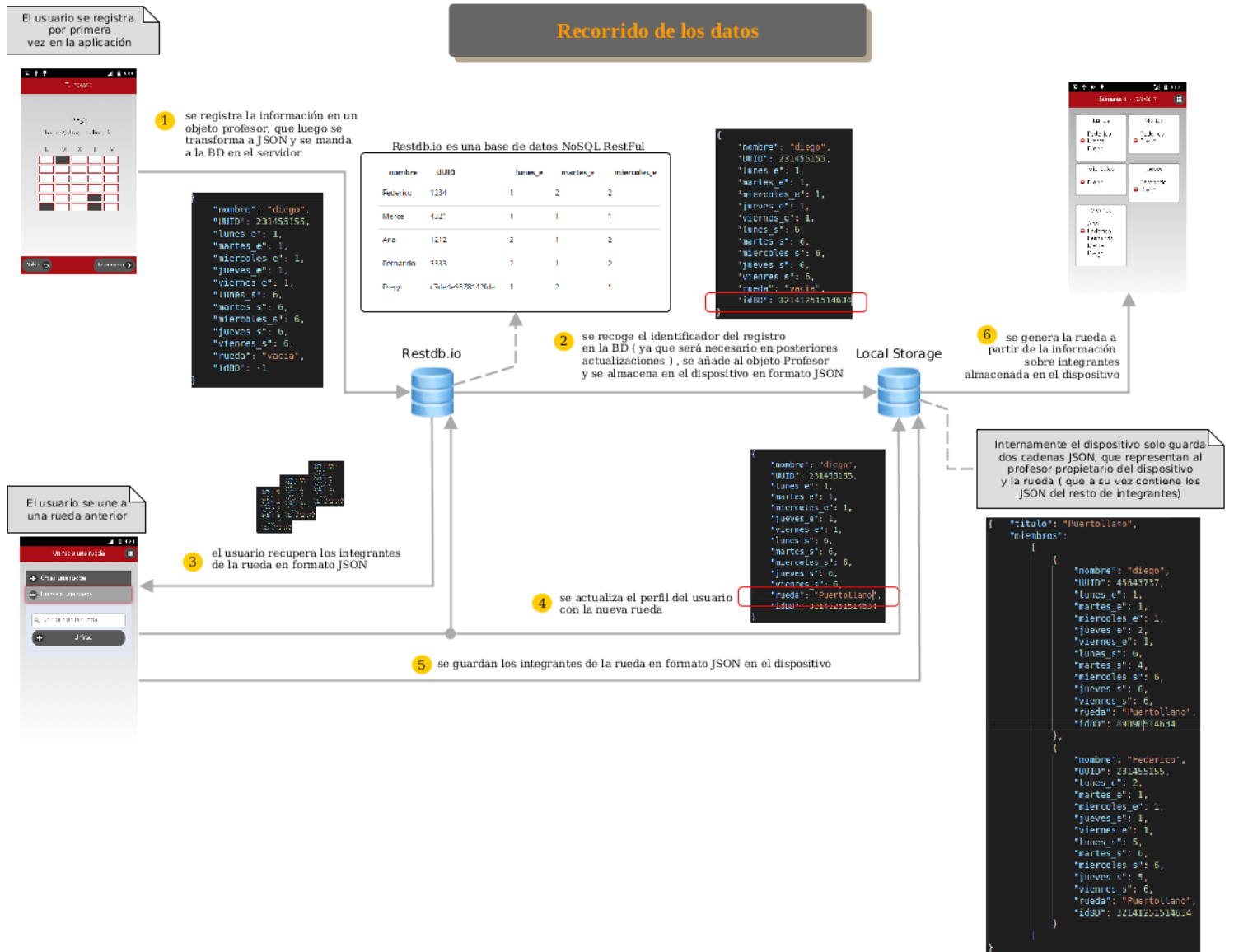


Figura 10: Recorrido de los datos

3. Diseño gráfico e interfaces

3.1 Estilos

Iconos, botones y elementos gráficos

Como ya se ha dicho en apartados anteriores una de las directrices del proyecto ha sido el **usar los elementos gráficos provistos por JQM** siempre que ha sido posible, precisamente para limitar el tiempo de desarrollo evitando tener que diseñar y desarrollar elementos propios.

Paleta de colores

La paleta de colores se ha diseñado usando la pagina web **themeroller.jquerymobile.com**, que nos brinda la oportunidad de ver como querían los colores elegidos aplicados a la interfaz de JQM, para después exportarlo como un tema CSS aplicable a la APP.

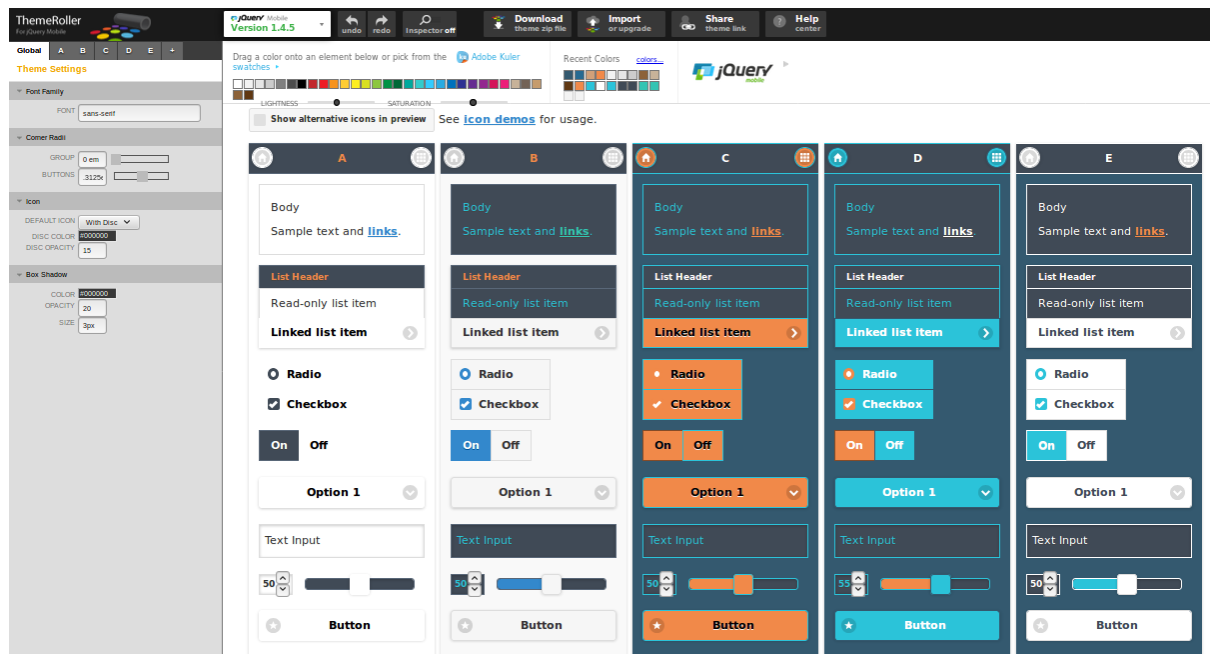
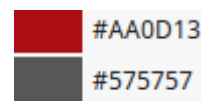


Figura 11: Themeroller

El diseño se basa en dos colores, con pequeñas variaciones:



Logotipo

A continuación se muestra el logotipo desarrollado teniendo en cuenta los colores elegidos y la temática del proyecto.



Figura 12: Logotipo

3.2 Wireframes

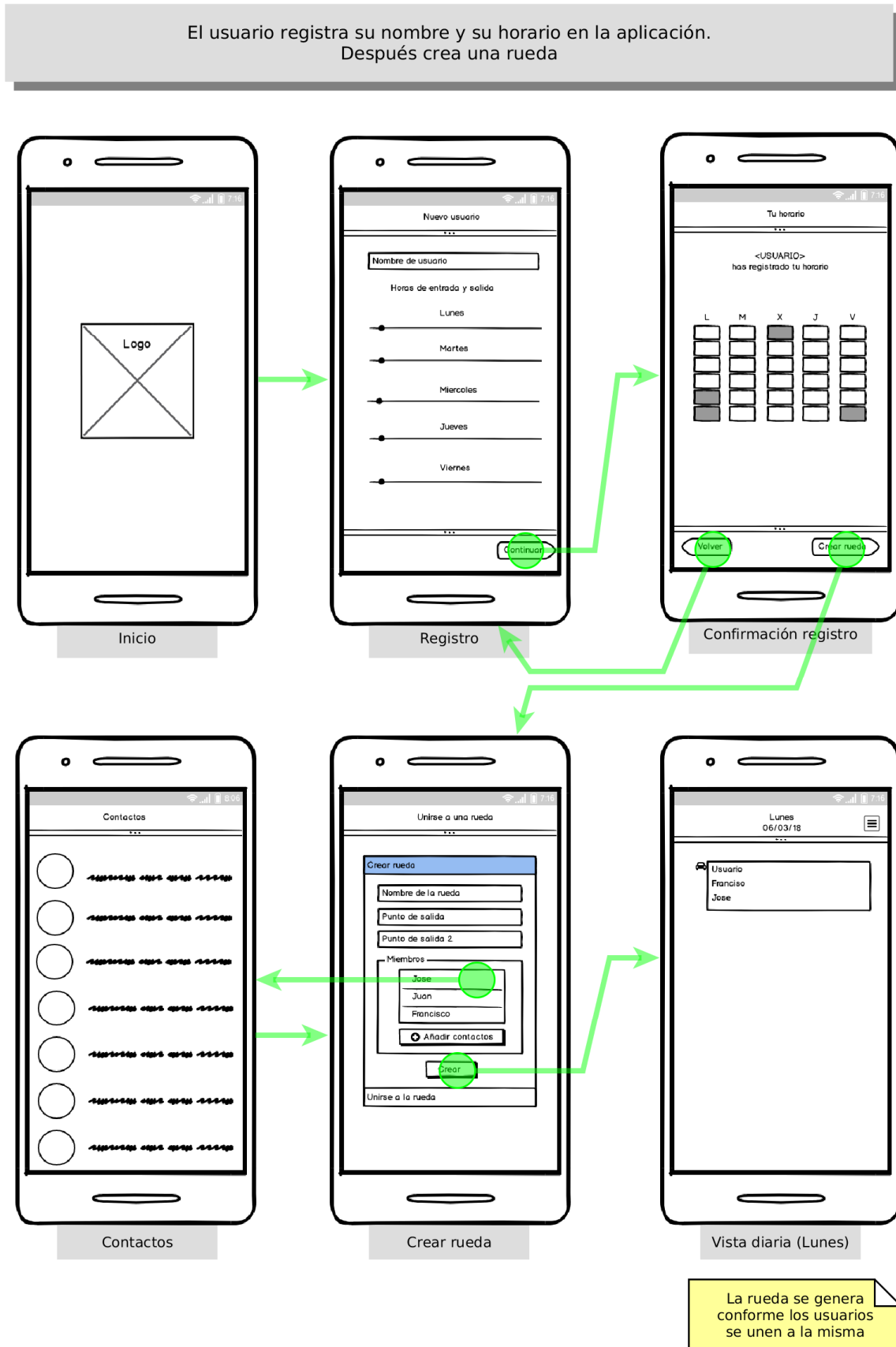


Figura 13: Wireframes: crear una rueda

El usuario se registra su nombre y su horario en la aplicación.
Después se une a una rueda que estaba creada previamente

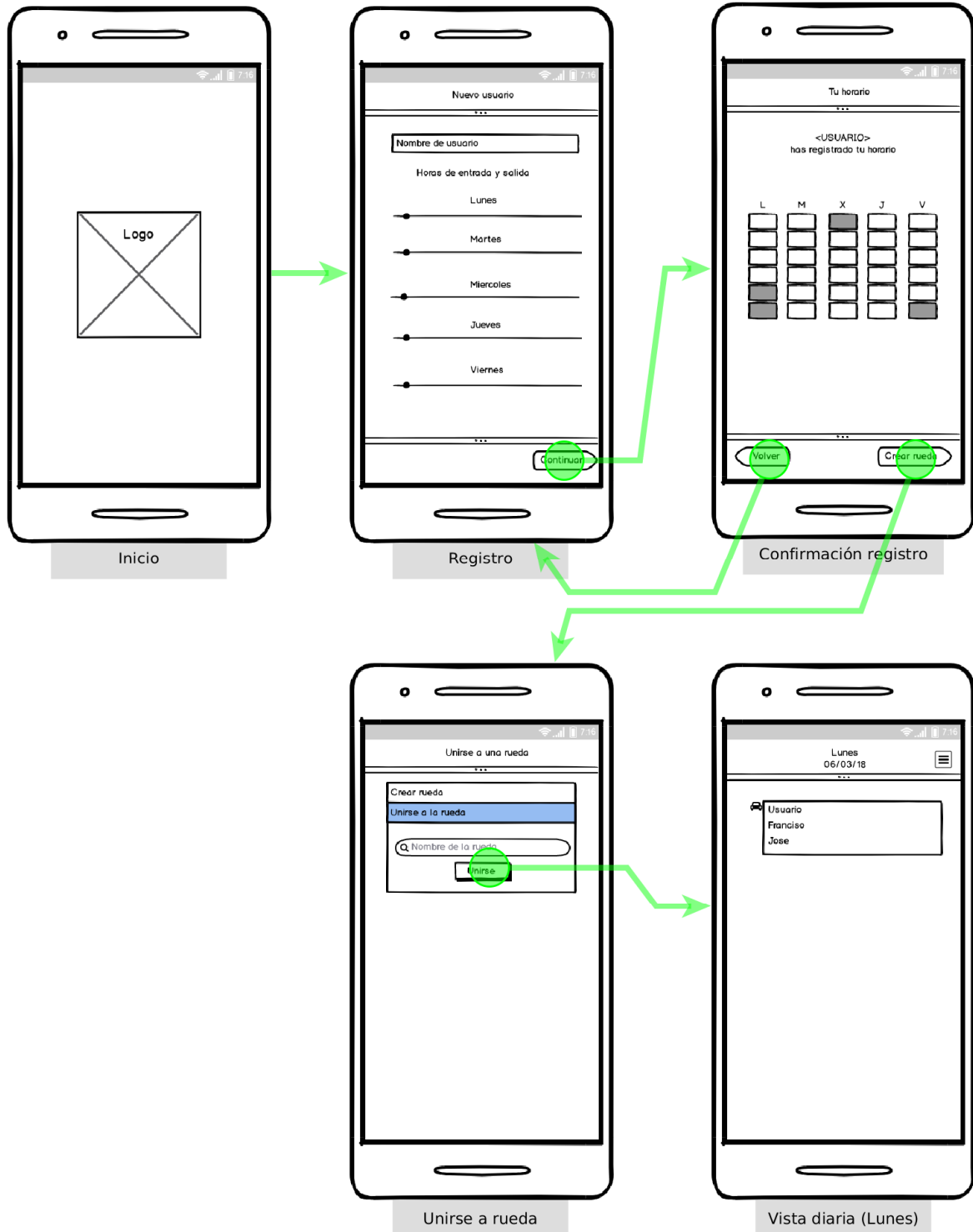


Figura 14: Wireframes: unirse a una rueda

La rueda se puede visualizar en formato diario, semanal o anual.
Nos podemos mover entre los días de la semana en curso mediante deslizamientos

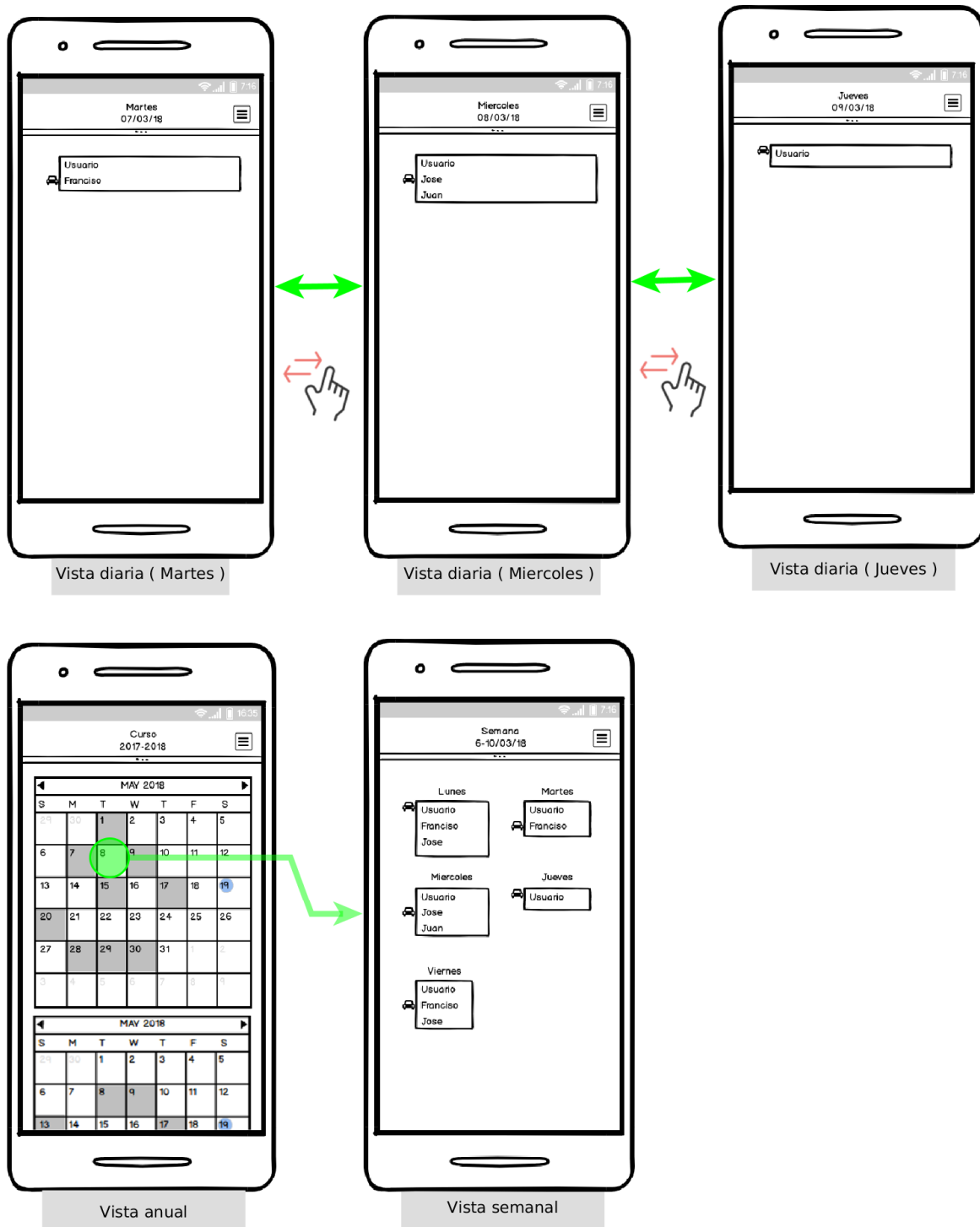
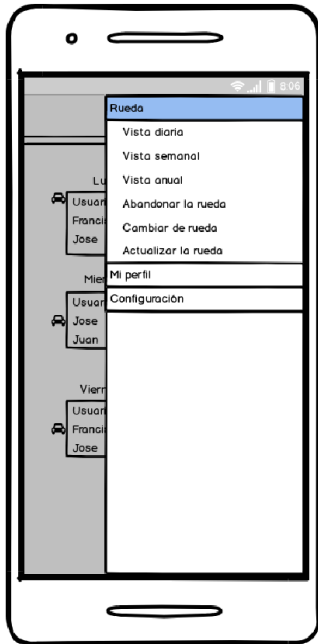
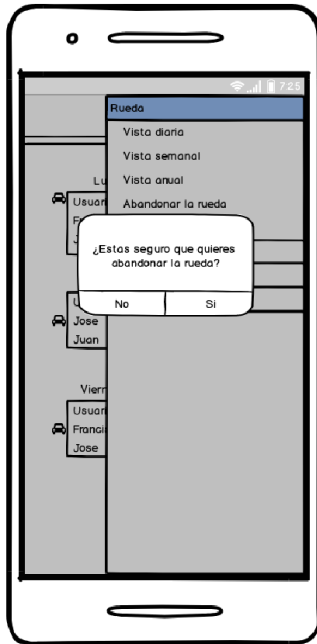


Figura 15: Wireframes: visualizaciones

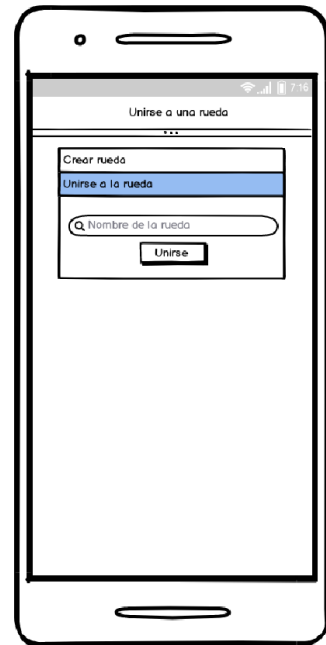
Pantallas del menú



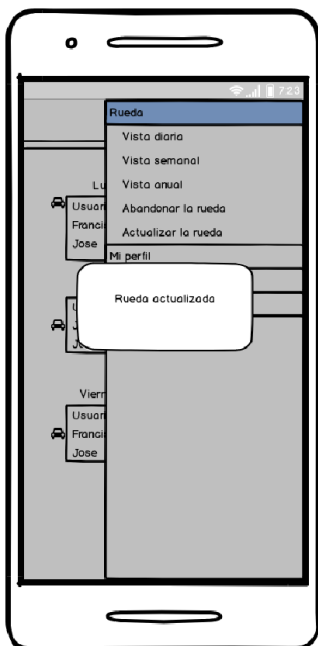
Menú lateral



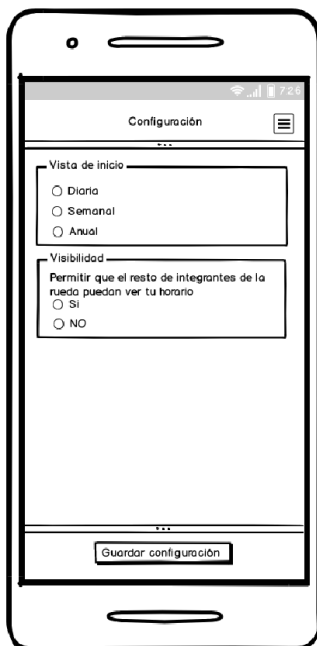
Abandonar la rueda



Cambiar de rueda



Actualización de rueda



Configuración

Figura 16: Wireframes: menú

Capítulo 4: Implementación

En este capítulo se va a hacer un recorrido por las principales consideraciones en cuanto a implementación que se han tomado en el desarrollo de la APP.

Entorno de desarrollo

Se ha utilizado el siguiente software y frameworks:

- **Jquery Mobile** (en adelante JQM). Es el framework de base del desarrollo.
- **Phonegap**. Empaqueta el proyecto web y construye una app con un WebView que corre la aplicación en el dispositivo. Concretamente se está empaquetando para Android, con lo que resulta en un .apk.
- **Firefox**. Se ha utilizado el navegador Firefox para la depuración inicial de la aplicación y para todos aquellos aspectos que no estén íntimamente ligados a un dispositivo móvil.
- **Android Studio**. Se ha integrado el proyecto Phonegap con Android Studio para depurar la aplicación directamente en un dispositivo móvil.
- **Genymotion**. Se ha utilizado este emulador para realizar pruebas paralelas a las realizadas con el emulador de Android Studio y para realizar la documentación.
- **Handlebars.js**. Se ha utilizando el motor de plantillas Handlebars para la generación del contenido dinámico y se prevé que la generalización de su uso para todo el proyecto.
- **Gimp**. Se ha usando este editor gráfico para el diseño de las imágenes.
- **themeroller.jquerymobile.com** El diseño del color se ha llevado a cabo por medio de esta herramienta web.
- **yEd**. Este editor de diagramas se ha utilizado para crear mapas de historia de usuario, arbol de navegación, diagrama de pantallas, etc.
- **balsamiq.com**. Se ha usado para diseñar los wireframes de la aplicación.
- **Restdb.io**. Servidor de base de datos RestFul.

Se han realizado también pruebas en un dispositivo hardware con las siguientes características:

- Marca: ZTE
- Modelo: Blade L3
- Versión Android: 5.0.2

Phonegap/Cordova

Phonegap/cordova[6] es un marco de desarrollo que permite desarrollar APPs híbridas, es decir, aplicaciones desarrolladas con tecnologías web que después se ejecutan como si hubieran sido implementadas en código nativo del dispositivo.

En la siguiente imagen se puede ver un esquema de cuál es la filosofía de este tipo de sistema:

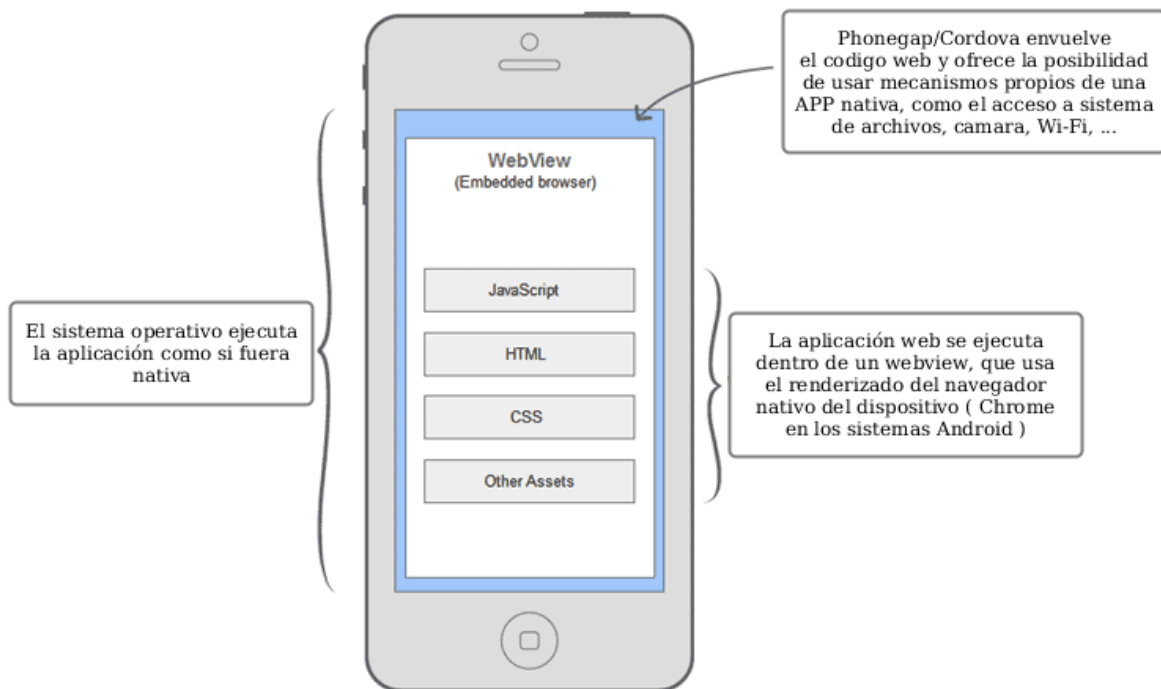


Figura 17: Esquema funcionamiento phonegap/cordova

La aplicación se desarrolla usando tecnologías web (HTML, CSS, javascript...) por lo que el resultado es una página web, que por medio de un WebView es posible ejecutar dentro de un dispositivo móvil como si fuera una aplicación nativa.

Al hacer la aplicación híbrida tenemos dos **ventajas** principales:

- Si somos desarrolladores web podemos desarrollar aplicaciones sin tener que aprender nuevos lenguajes.
- Derivado de lo anterior el diseño de las interfaces tiene más libertad porque podemos usar toda la potencia de tecnologías web como CSS o HTML 5.

En contraposición tenemos dos **desventajas**:

- El renderizado de la aplicación depende del navegador del dispositivo.
- La velocidad de ejecución puede decaer si se hace un uso intensivo del dispositivo.

Para el caso que nos ocupa en este proyecto se ha tomado como público objetivo usuarios de Android y por tanto las pruebas se han hecho en este SO, confirmando el correcto renderizado de las

interfaces. Así mismo, se ha testeado la aplicación y establecido como aceptable la velocidad de ejecución de la APP.

JQuery Mobile

Según la página oficial [7]:

"jQuery Mobile is a HTML5-based user interface system designed to make responsive web sites and apps that are accessible on all smartphone, tablet and desktop devices."

Las razones de elección de este framework han sido:

- La mayor parte del código se carga de una vez y después se va modificando el DOM, lo que hace que solo haya un momento de lentitud al principio de la carga.
- Ofrece controles predefinidos, ahorrando así tiempo de desarrollo.
- Es uno de los frameworks más usados, por lo que ha sido ampliamente testeado y existe gran cantidad de documentación.

La APP desarrollada se compone de un archivo principal (index.html) que contiene el código base HTML de todas las pantallas de la aplicación. En JQM estas secciones se identifican poniendo el atributo: **role="page"**. En el siguiente código podemos apreciar la definición de páginas distintas, una para los lunes y otra para los martes.

```
<div data-role="page" class="dia" id="Lunes" data-theme="a" data-next="Martes">
  <div role="main" class="ui-content">
    <div class="ui-body ui-body-a ui-corner-all">
      </div>
    </div>
  </div>
<div data-role="page" class="dia" id="Martes" data-theme="a" data-next="Miercoles">
  <div role="main" class="ui-content">
    <div class="ui-body ui-body-a ui-corner-all">
      </div>
    </div>
  </div>
</div>
```

Siguiendo el **diagrama de navegación** expuesto en el punto 3.2 se han definido tantas páginas como pantallas del diagrama. Todas incluidas en el archivo index.html. Este archivo además hace la carga de todos los archivos javascript y CSS necesarios para iniciar la aplicación.

Incidencias destacadas

- Imposibilidad de dividir el código. El framework promueve que se use un solo archivo para contener todas las páginas de un proyecto. Aunque es posible dividir el proyecto en diversos ficheros multipágina, no se ha encontrado el procedimiento óptimo de carga para que esta estructura funcione bien con la gestión que hace JQM de la carga de páginas.
- La gestión del ciclo de vida de las páginas JQM es esencial a la hora de diseñar el funcionamiento de cualquier aplicación desarrollada con este framework. Algunas de las referencias utilizadas a este respecto son las que se encuentran en las siguientes URLs:

<http://jqm.readthedocs.io/en/jqm-all-1.4.1/archi.html>

<https://jqmtricks.wordpress.com/2014/03/26/jquery-mobile-page-events/>

- Rangesliders. Una de las directrices del proyecto es utilizar elementos predefinidos por el framework JQuery Mobile, los cuales se ajustarán en la medida de lo posible para personalizar su diseño. Esto limita las posibilidades de diseño de la interfaz, pero agiliza el desarrollo de las funcionalidades. Una de las implicaciones de esta decisión ha sido el uso de rangeslider [8] para el registro del horario, que es menos intuitivo para el usuario. En un principio se planeó utilizar una interfaz similar a la de confirmación del horario, pero se descartó dada la limitación de tiempo y la necesidad de programación extra para esta opción. Aún así fue necesario modificar la visualización de este control, para no mostrar los contadores de los sliders, mediante el código CSS que se muestra.

```
.ui-rangeslider .ui-rangeslider-sliders {  
  margin: 0 15px;  
}  
  
.ui-slider-handle {  
  border-radius: 100%;  
}  
  
div.ui-rangeslider-sliders:first-child div.ui-slider-bg {  
  background: transparent !important;  
}
```

Almacenamiento

Como se puede ver dentro del ANEXO B en el archivo **caminoDeLosDatos**, la información toma diferentes formatos a lo largo de su viaje por las diferentes partes del sistema.

Una gran decisión de diseño de la aplicación fue el optar por el almacenamiento en **Local Storage** en vez de usar una base de datos local al dispositivo. El almacenamiento en Local Storage se hace por medio de un mecanismo clave/valor, es decir, se le da un nombre a lo que vamos a almacenar y luego se puede recuperar por ese nombre. Entonces ¿cómo se pueden almacenar datos estructurados? Por medio de cadenas de caracteres que representen esos datos estructurados en formato JSON.

Realmente esta forma de almacenamiento es la más adecuada para la estructura de información que tenemos, pues vamos acceder siempre de un golpe a toda la información. Cuando se genera la rueda, necesitamos todos los datos almacenados de los profesores y **nunca partes de estos**, por tanto tiene menos sentido el usar una base de datos y acarrear con las complejidades de la misma.

En Local Storage siempre se almacenan dos objetos: la rueda y el propietario de la misma, con sus respectivos nombres: rueda_g y propietario_g.

El principal problema hallado con LocalStorage es que es síncrono, pero la app necesita hacer comprobaciones en el servidor al inicio de la misma y almacenarlos de forma local. El problema radica en que es imposible **almacenar información el LocalStorage desde una función asíncrona**, como son las de consulta del servidor.

Por ello se cambió el método de almacenamiento usando **localforage.js**, una librería desarrollada por Firefox para unir la simplicidad de LocalStorage con los beneficios de WebSQL o indexedDB. En el artículo de Morony[10] pueden leerse un breve resumen de los diferentes tipos de almacenamientos disponibles con Phonegap/Cordova.

La aplicación hace las siguientes **comprobaciones al inicio**:

- Comprobar si el usuario existe en el almacenamiento local.
- Comprobar si existe la rueda en el almacenamiento local.
- Si no existe en el almacenamiento local comprobar si existe en el servidor. Esto **evita una pérdida de sincronización** entre el dispositivo y el servidor, en caso del borrado de localforage, como en una reinstalación. La aplicación detecta que el dispositivo se registró anteriormente en el sistema y recupera los datos automáticamente, tanto del propietario como de la rueda.

Identificación de usuarios

En una primera fase se utilizaba como identificador de usuarios el número de teléfono de los profesores, pero después se ha cambiado por el UUID del dispositivo debido a la dificultad (o casi imposibilidad) de obtener el propio número de teléfono desde la APP (tanto es así, que APPs como WhatsApp te piden expresamente el número).

Este factor ha imposibilitado el uso de los contactos para crear la rueda recuperando sus horarios de la base de datos, por lo que la lista de contactos en la creación de la rueda ha quedado relegada a una mera notificación a los usuarios pidiéndoles su anexión a la rueda (esto quedaría implementarlo mediante notificaciones PUSH).

Promesas

Una de las dificultades técnicas encontradas durante el desarrollo de la aplicación ha sido la gestión de los métodos asíncronos de acceso a la base de datos. El principal problema radicaba en que la interfaz se renderizaba antes de haber recuperado los datos. Para solucionarlo se ha incluido la implementación de **promesas** [9] en los métodos asíncronos.

Las promesas permiten seguir con el código de la aplicación, mientras se queda a la espera del valor que se pidió. Cuando la operación haya terminado y esté disponible el valor, este se almacena allí donde se haya especificado.

Por ejemplo, para unirnos a una rueda primero se debe recuperar la rueda del servidor, después se calculan las coincidencias y finalmente se renderiza la rueda semanal. No se puede renderizar hasta que tengamos la rueda, pero al ser un método asíncrono es posible que la renderización se realice antes de tener los datos, por ello se incluye la consulta en una promesa como se muestra a continuación:

```
Rueda.recuperaRuedaServer = function(tituloRueda) {  
  return new Promise(function(resolve, reject) {  
    var settings = {  
      'async': true,  
      'crossDomain': true,  
      'dataType': 'json',  
      'url': 'https://ruedacoches-e561.restdb.io/rest/profesor?q={"rueda": "' + tituloRueda + '"}',  
      'method': 'GET',  
      'headers': {  
        'content-type': 'application/json',  
        'x-apikey': '51515151515151515151515151515151',  
        'cache-control': 'no-cache'  
      }  
    }  
  })  
}
```

Renderizado y Handlebars

Un problema parecido al anterior surge al utilizar el motor de plantillas Handlebars, pues en ciertos momentos es necesario acceder a elementos generados por las plantillas y puede que este acceso se produzca antes de que los elementos hayan terminado de generarse. Por ello, y dado que el uso de handlebars no es crítico para el desarrollo de la APP, se ha suprimido su uso en la generación de la vista semanal y diaria. En sustitución se ha desarrollado un código propio que realiza la misma función.

Algoritmo de calculo de la rueda

Toda la aplicación gira en torno al cálculo de la rueda. La visualización de la rueda se da en tres formatos: diaria, semanal y anual.

Una de las consideraciones que se han tomado en el diseño de la APP, es que se calcula la rueda a partir de los horarios de los integrantes, pero este cálculo no se almacena en memoria permanente, sino que se calcula en cada ejecución de la APP. Esto evita almacenar información que puede calcularse en el momento y hace más sencillo el recalcu de la rueda si surge algún cambio en los integrantes.

El algoritmo toma en consideración lo siguiente:

- El curso tiene 43 semanas completas, contadas desde el primer lunes de septiembre.
- Solo nos interesan los profesores que coincidan con el usuario que esté ejecutando la aplicación, es decir, cada usuario dispondrá de una rueda personalizada aunque sea el mismo conjunto de profesores el que integre la rueda, pues las coincidencias en los horarios serán diferentes.
- En la vista anual se muestra solo cuando nos toca conducir, pero para hacer este cálculo se calcula la rotación de todos los días del año.

En esencia lo que hace el algoritmo es:

- 1) Buscar las coincidencias de cada día de los horarios de los profesores con el propietario.

```
for (var i = 0; i < this.miembros.length; i++) {  
    if (this.propietario.getEntrada(dia) == this.miembros[i].getEntrada(dia) &&  
        this.propietario.getSalida(dia) == this.miembros[i].getSalida(dia)) {  
        coincidencias.push(this.miembros[i]);  
    }  
}
```

- 2) Calcular cual es el primer lunes de septiembre.
- 3) Generar la rotación del turno de cada día para las 43 semanas que componen el curso.

```
Rueda.prototype.conductoresCurso = function() {  
    var calendario = new Array();  
  
    var lunes = this.generarRuedaDia(0);  
    var martes = this.generarRuedaDia(1);  
    var miercoles = this.generarRuedaDia(2);  
    var jueves = this.generarRuedaDia(3);  
    var viernes = this.generarRuedaDia(4);  
  
    for (var i = 0; i < 43 * 5; i += 5) { // El curso tiene 43 semanas de 5 días  
        calendario[i] = lunes[(i / 5) % lunes.length].getUUID();  
        calendario[i + 1] = martes[(i / 5) % martes.length].getUUID();  
        calendario[i + 2] = miercoles[(i / 5) % miercoles.length].getUUID();  
        calendario[i + 3] = jueves[(i / 5) % jueves.length].getUUID();  
        calendario[i + 4] = viernes[(i / 5) % viernes.length].getUUID();  
    }  
  
    return calendario;  
}
```

- 4) Una vez calculadas las rotaciones para todo el curso, la aplicación solo necesita saber el número de la semana dentro de las 43, para recuperar toda la información relativa a la misma.

En la siguiente figura se esquematiza el tratamiento que hace el algoritmo de los horarios de los profesores para calcular la rueda de todo el curso.

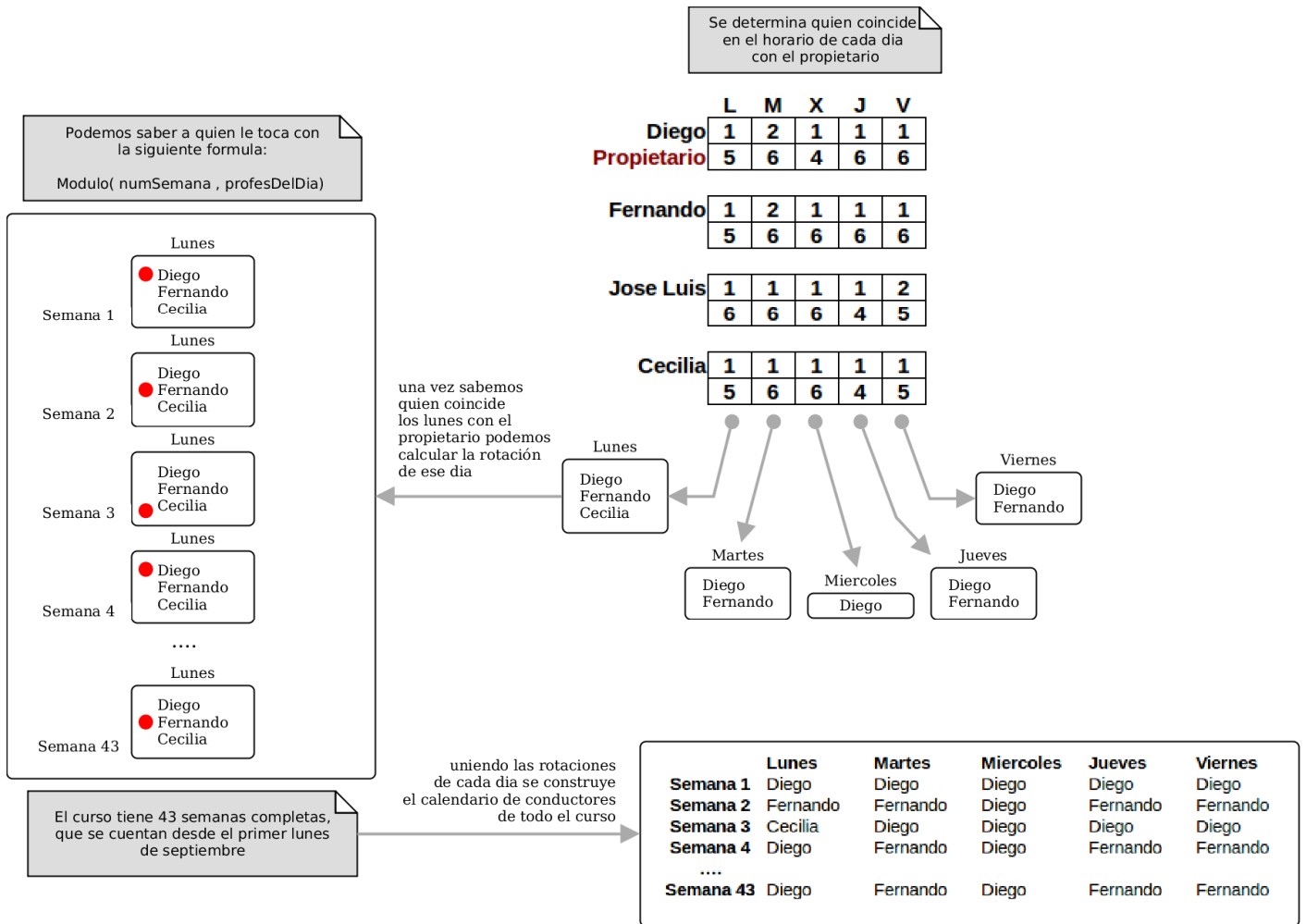


Figura 18: Calculo de la rueda

Capítulo 5: Demostración

En el siguiente apartado se muestran las principales pantallas del resultado final del proyecto. Todas se incluyen en el Anexo B.



Figura 19: Pantallas 1

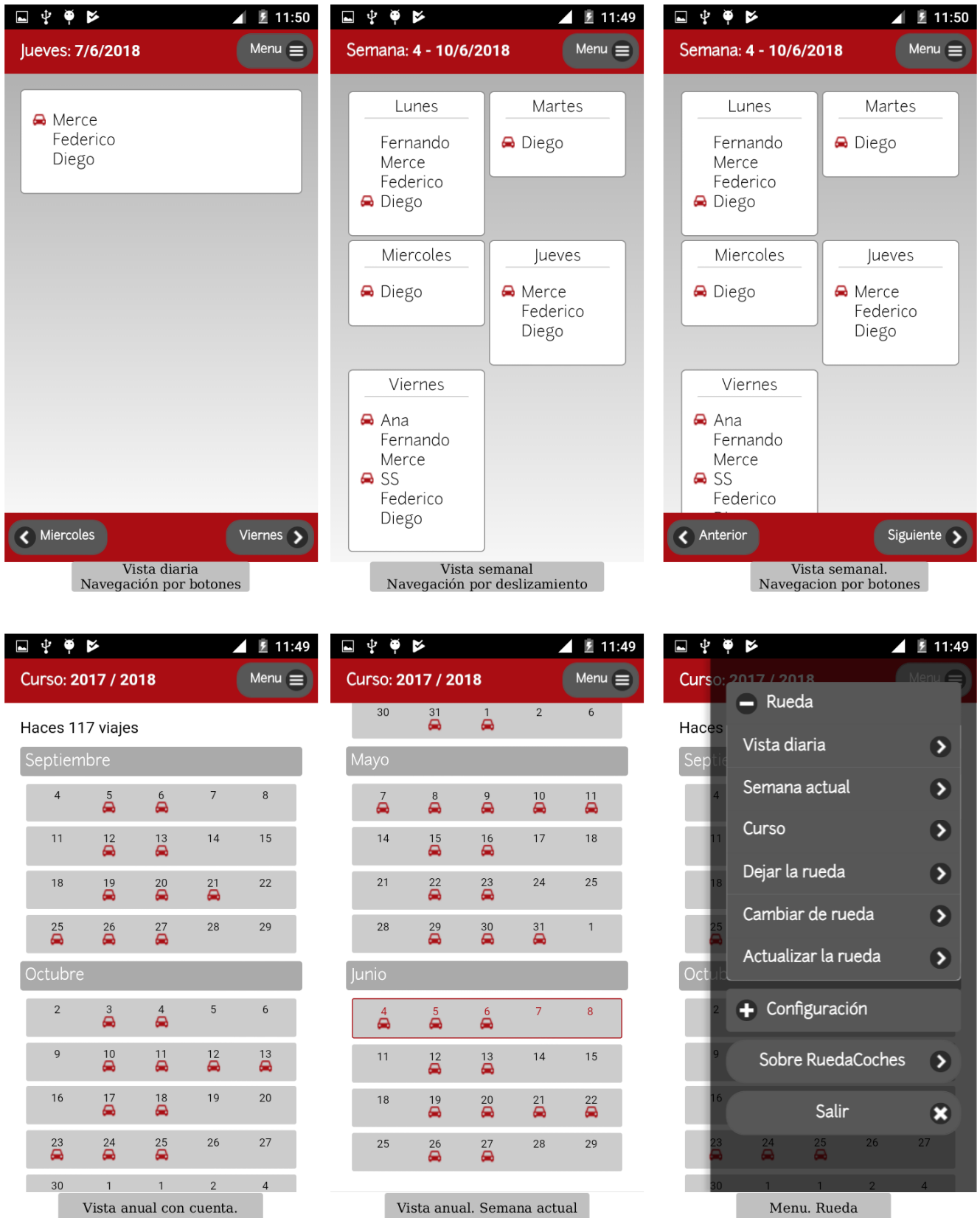


Figura 20: Pantallas 2

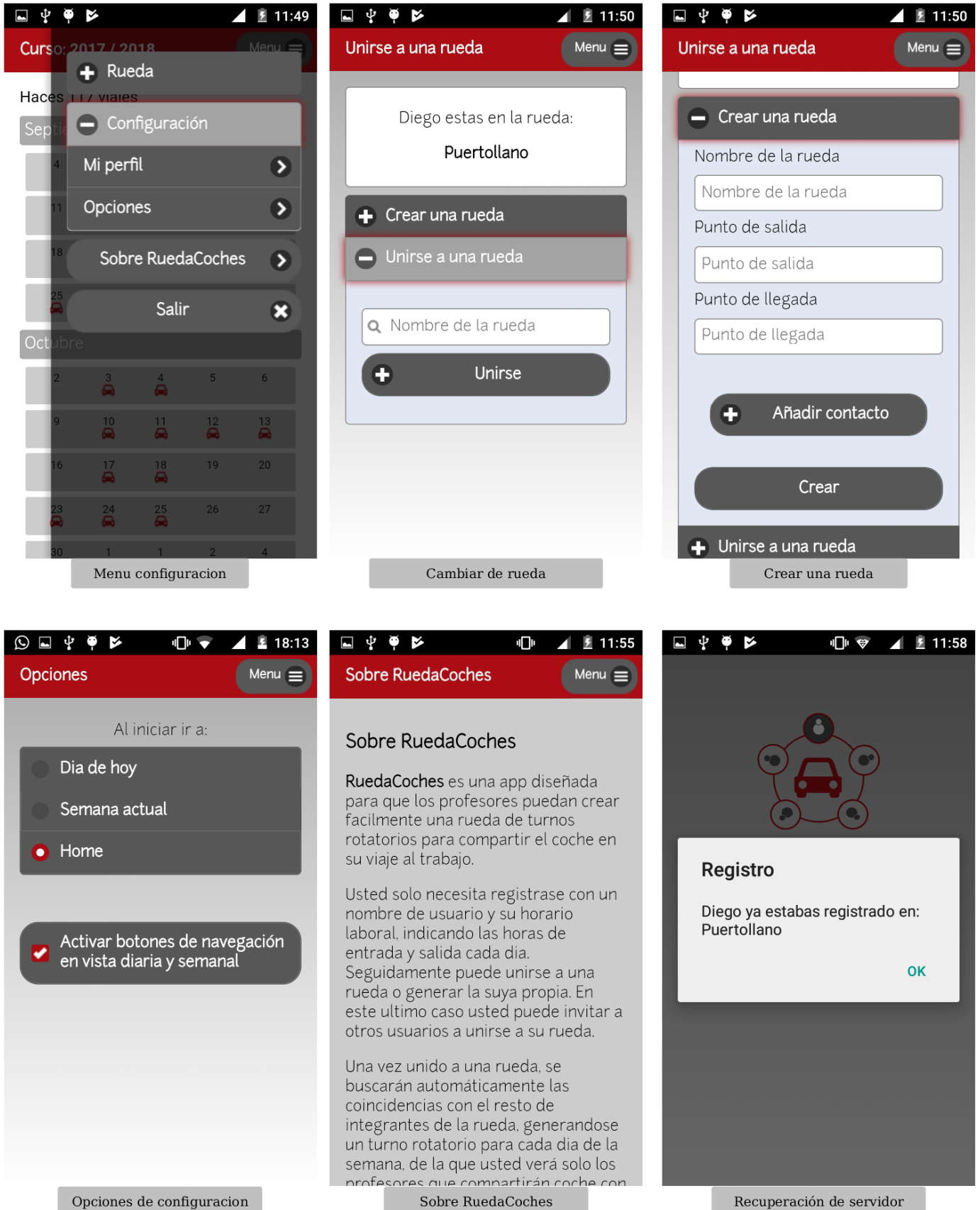


Figura 21: Pantallas3

Capítulo 6: Conclusiones y líneas de futuro

1. Conclusiones

La mayor lección aprendida en el desarrollo de este proyecto ha sido la gestión del tiempo en cuanto a la planificación de un desarrollo software de estas características. Se ha aprendido a secuenciar el desarrollo y a gestionar las incidencias sopesando la relación coste/beneficio de cada decisión.

En cuanto a tecnologías de desarrollo ha sido muy interesante el aprender cómo construir una APP híbrida, los frameworks necesarios y las posibilidades de utilización de las tecnologías web. Se ha profundizado en el conocimiento de la gestión del DOM que hacen frameworks como JQuery. Este proyecto también me ha permitido aprender cómo se desarrollan webs en un solo archivo, al estilo de JQM.

El proyecto me ha permitido aprender mucho y se ha gestionado el tiempo para conseguir un producto mínimo viable. Se han conseguido los objetivos planteados al principio del proyecto, a falta de implementar las notificaciones PUSH para la actualización automática de la rueda, por falta de tiempo.

Durante todo el proyecto, teniendo en cuenta que el equipo de desarrollo es una sola persona, se ha intentado seguir una metodología ágil, al estilo de SCRUM, liberando productos funcionales, pero limitados, en las PEC 3, 4 y 5.

Para ello la planificación ha sido muy importante, pues se esperaba tener una serie de funcionalidades implementadas para cada entrega. Siguiendo la metodología ágil, la planificación se tuvo que modificar en la PEC4 para acomodarse al ritmo de desarrollo y hacer un seguimiento real del mismo. Modificación que consistió en eliminar la automatización de las actualizaciones de la rueda mediante notificaciones PUSH, que, en conjunto, no resta funcionalidades al APP, pues el usuario puede actualizar la rueda mediante un botón.

2. Líneas de futuro

La aplicación ahora mismo cumple su objetivo de una forma rudimentaria, puede usarse, pero también puede enriquecerse con las funcionalidades que se pueden ver en el user story map incluido en el ANEXO B. Algunas de estas funcionalidades y otras mejoras son las siguientes:

- Mejora del algoritmo del cálculo de la rueda, incluyendo distintos puntos de salida.
- Uso de Firebase para implementar:
 - o Actualización automática de la rueda mediante notificaciones cada vez que se produzca un cambio.
 - o Notificaciones de petición de unión a una rueda.
 - o Notificaciones de salida de una rueda.
 - o Notificaciones de borrado de una rueda.
 - o Mandar un mensaje a los integrantes de la rueda
- Estadísticas de uso del coche, ahorro, etc.
- Geolocalización de los puntos de salida y llegada.
- Registro del domicilio del usuario y cálculo del mejor punto de salida.
- Búsqueda de rueda por punto de llegada.
- Visualización de los horarios del resto de integrantes de la rueda.

Bibliografía

1. *Orden de 02/07/2012 por la que se dictan instrucciones que regulan la organización y funcionamiento de los institutos de educación secundaria en la Comunidad Autónoma de Castilla-La Mancha*. Consejería de Educación, Cultura y Deportes de Castilla-La Mancha. 3 de julio de 2012.
2. PATTON, Jeff. [Consulta 13/03/2018] “*Story Mapping Quick Reference*.” <<https://jpattonassociates.com/story-mapping-quick-ref/>>
3. KNIBERG, Henrik. (2007) “*Scrum y XP desde las trincheras*”. C4 Media.
4. SERRANO, Raul. . [Consulta 13/03/2018] “*Estimación basada en tallas*” <<https://www.scribd.com/document/59520129/Estimacion-basada-en-tallas-para-el-calculo-del-esfuerzo-de-las-Historias-de-Usuario>>
5. Rodríguez, J. R. (2013) “*La gestión del proyecto a lo largo del trabajo final*”. Apuntes de la Universitat Oberta de Catalunya.
6. Phonegap. <<http://phonegap.com>>
7. JQM. <<http://jquerymobile.com>>
8. JQM. RangeSlider. <<https://api.jquerymobile.com/rangeslider/>>
9. Promesas Javascript <<https://developers.google.com/web/fundamentals/primers/promises?hl=es>>
10. MORONY, “*A Summary of Local Storage Options for PhoneGap Applications*” <<https://www.joshmorony.com/a-summary-of-local-storage-options-for-phonegap-applications/>>

1. Anexos

Anexo A: Entregables del proyecto

1. Código. Código maestro de la APP.
2. RuedaCoches.apk. Archivo para instalar la APP en Android.

Anexo B: Diseños y diagramas

1. *UserStoryMap*. Mapa de historias de usuario.
2. *EsquemaDeFuncionamiento*. Esquema con la descripción del funcionamiento general de la APP.
3. *Diagrama de navegación*.
4. *Wireframes*. Directorio con los Wireframes de las interfaces de la APP.
5. *Pantallas*. Directorio con las principales pantallas de la APP.
6. *Recorrido de los datos*. Esquema con la transformación de los datos a lo largo de la APP.