



***Proyecto Fin de Carrera***

**Cliente móvil de integración de información  
industrial**

**(Industrial data integration Mobile Client)**

Para acceder al Título de

**INGENIERO EN INFORMÁTICA**

**Autor: Francisco García Higuera**

**Octubre - 2012**

**INGENIERÍA EN INFORMÁTICA**

**CALIFICACIÓN DEL PROYECTO FIN DE CARRERA**

**Realizado por: García Higuera, Francisco**

**Director del PFC: Sierra Sánchez, Miguel**

**Ponente: Vallejo Alonso, Fernando**

**Título:** “Cliente móvil de integración de información industrial ”

**Title:** “ Industrial data integration mobile client”

**Presentado a examen el día:**

para acceder al Título de

**INGENIERO EN INFORMÁTICA**

Composición del Tribunal:

Presidente (Apellidos, Nombre): González Harbour, Michael

Secretario (Apellidos, Nombre): Martínez Fernández, María del Carmen

Vocal (Apellidos, Nombre): Menéndez de Llano Rozas, Rafael

Vocal (Apellidos, Nombre): Sánchez Barreiro, Pablo

Vocal (Apellidos, Nombre): Sanz Gil, Roberto

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: Vocal

Fdo.: Vocal

Fdo.: Vocal

Fdo.: El Director del PFC

# ÍNDICE

1	Introducción .....	1
1.1	Introducción .....	1
1.2	Objetivo del proyecto.....	3
1.3	Ambiente de la aplicación .....	3
1.4	Estructura del documento.....	4
2	Descripción y planificación.....	5
2.1	Ámbito Funcional .....	5
2.2	Descripción de la aplicación .....	5
2.2.1	Sección Indicadores.....	6
2.2.2	Sección Alarmas .....	6
2.2.3	Sección Sinópticos.....	6
2.2.4	Sección Análisis .....	7
2.2.4.1	Barra de Herramientas .....	8
2.2.4.2	Pantalla Filtro .....	8
2.2.4.3	Pantalla Tiempo Real.....	9
2.2.4.4	Pantalla Datos Históricos.....	9
2.2.4.4.1	Pantalla de configuración datos históricos.....	9
2.2.4.5	Pantalla gráfica .....	9
2.2.4.5.1	Pantalla de configuración gráficas .....	10
2.2.4.6	Pantalla de Agrupaciones temporales .....	10
2.3	Metodología de desarrollo.....	10
2.4	Patrón de diseño MVC.....	12
2.5	Requisitos de alto nivel de la aplicación .....	13
2.6	Sumario .....	16
3	Herramientas y tecnologías utilizadas .....	17
3.1	Herramientas utilizadas durante el proceso de desarrollo.....	17
3.2	Tecnología utilizada.....	18
3.2.1	Sencha Touch .....	18
3.2.2	Servicios Web .....	19
3.3	Sumario .....	19
4	Desarrollo de la aplicación .....	21
4.1	Iteración 1: Desarrollo de pantalla principal y filtrado de datos .....	21
4.1.1	Desarrollo de la iteración .....	23

4.2	Iteración 2: Desarrollo de Consulta de históricos y Tiempo Real .....	24
4.2.1	Desarrollo de la iteración .....	28
4.3	Iteración 3: Desarrollo de gráficas de tendencia .....	30
4.3.1	Desarrollo de la iteración .....	32
4.4	Iteración 4: Agregar puntos a agrupaciones temporales e interoperar con ellos .....	33
4.4.1	Desarrollo de la iteración .....	37
4.5	Iteración 5: Desarrollo de Indicadores y Alarmas .....	37
4.5.1	Desarrollo de la iteración .....	40
4.6	Iteración 6: Desarrollo de gráficos Sinópticos.....	41
4.6.1	Desarrollo de la iteración .....	41
4.7	Sumario .....	41
5	Evaluación y pruebas.....	43
5.1	Sumario .....	43
6	Conclusiones y trabajos futuros.....	45
6.1	Conclusiones.....	45
6.2	Trabajos futuros .....	46
7	Bibliografía .....	47

## INDICE DE FIGURAS

Figura 1 - Esquema IDbox.....	2
Figura 2 - Gráfico Sinóptico .....	7
Figura 3 - Botón añadir puntos agrupación.....	8
Figura 4 - Metodología incremental.....	11
Figura 5 - Diagrama desarrollo evolutivo .....	12
Figura 6 - Patrón MVC .....	12
Figura 7 - Aplicación web .....	18
Figura 8 - Captura pantalla principal .....	21
Figura 9 - Captura pantalla consulta .....	22
Figura 10 - Pantalla de Filtro .....	23
Figura 11 - Botones Tiempo Real e Históricos.....	24
Figura 12 - Pantalla Tiempo Real.....	25
Figura 13 - Pantalla Históricos.....	25
Figura 14 - Pantalla configuración tiempo real .....	26
Figura 15 - Combo-box intervalos de tiempo y resolución .....	27
Figura 16 - Combo-box resolución .....	27
Figura 17 - Mensaje emergente “seleccione puntos” .....	28
Figura 18 - Gráfica en tiempo real.....	28
Figura 19 - Botón gráficas de tendencia.....	30
Figura 20 - Pantalla gráficas de tendencia .....	31
Figura 21 - Aumento de resolución en gráficas de tendencia.....	31
Figura 22 - Pantalla configuración gráficas de tendencia .....	32
Figura 23 - Botón Agregar a agrupación temporal.....	33
Figura 24 - Mensaje emergente Agregar puntos agrupación temporal.....	34
Figura 25 - Botón Agrupación temporal.....	35
Figura 26 - Botón eliminar de agrupación temporal.....	36
Figura 27 - Mensaje emergente eliminar agrupación temporal .....	36
Figura 28 - Mensaje emergente no hay datos agrupación temporal.....	37
Figura 29 - Pantalla de indicadores .....	38
Figura 30 - Pantalla EstadoPlanta.....	38
Figura 31 - Gráfica de tendencia indicadores.....	39
Figura 32 - Alarmas del sistema .....	39
Figura 33 - Gráfico de alarmas .....	40
Figura 34 - Gráfico sinóptico .....	41

## **AGRADECIMIENTOS**

A través de estas líneas quiero expresar mi agradecimiento a varias personas por su apoyo, tanto en el transcurso de la carrera como en el desarrollo de este proyecto.

En primer lugar me gustaría agradecer a mi familia y a mi novia el apoyo constante y la motivación necesaria que me ofrecieron para poder seguir adelante en los momentos más difíciles que sucedieron a lo largo de la carrera.

También quiero recordar en estas líneas a mis amigos más cercanos, a los que conozco desde la infancia y con los que he podido compartir grandes experiencias durante estos años, deseando que esta amistad dure para siempre.

Por otra parte, también me gustaría acordarme de mis compañeros de facultad, con algunos de los cuales me une una gran amistad. Hemos compartido risas y sufrimientos a lo largo de estos años, pero el viaje ha merecido la pena.

Por último ya para finalizar, quiero agradecer su ayuda y apoyo en el desarrollo de este proyecto a Fernando Vallejo, mi tutor durante este proyecto y también a Miguel Sierra y todo el equipo de CIC Consulting Informático, por prestarse a resolver cualquier duda que me surgiera en el desarrollo de este trabajo.

A todos ellos, muchas gracias.

## **RESUMEN**

En CIC Consulting Informático disponen de una aplicación denominada IDbox. Dicha aplicación es un sistema de información de planta, que permite a los usuarios obtener datos en tiempo real de señales (como por ejemplo temperatura, presión, etc.), consultar históricos, visualizar graficas de tendencia, entre otras funcionalidades.

La aplicación esta orientada al entorno industrial y ya se dispone de un cliente web y un cliente Excel. Mediante el desarrollo de este proyecto, se dispondrá también de un cliente móvil, que permitirá aumentar la portabilidad de la aplicación ofreciendo a los usuarios la posibilidad de acceder a la información del sistema en prácticamente cualquier lugar en el que se encuentren.

Debido a la multitud de plataformas móviles que se encuentran en el mercado actualmente, la aplicación móvil será una aplicación web, lo que permite acceder a ella desde los principales smartphones del mercado.

### **Palabras clave**

Aplicación Móvil, entorno industrial, Sistemas de información de planta, portabilidad

## **ABSTRACT**

In Computer Consulting CIC have an application called IDbox. This application is a plant information system that allows users to obtain real-time data signals (such as temperature, pressure, etc.) consulting historical data, seeing trend graphs, among other features.

The application is oriented to industrial environment; currently it has a web client and a Excel client. Through the development of this project, it will have a mobile client too, which will increase application portability by offering users the ability to access information in almost any place where you are.

Because of the amount of mobile platforms that are set on the market nowadays, the mobile application is a web application, which allows access it from the main smartphones.

## **Keywords**

Mobile application, industrial environment, plant information system, portable

# 1 Introducción

## 1.1 Introducción

**IDbox**[IDbox] es un sistema que pone a disposición de los usuarios, la información procedente de un conjunto de fuentes de naturaleza heterogénea (PLCs, sensores, registradores, SCADAs, ERPs, ESBs, ficheros, etc.), de una forma totalmente integrada y homogénea.

El sistema es capaz de recoger, integrar, procesar y visualizar grandes volúmenes de información, como por ejemplo señales de una planta industrial, red eléctrica, petroquímica, gas... tanto centralizados como dispersos geográficamente.

Para la gestión de la información, el sistema **IDbox** dispone de unos agentes de datos que encapsulan el acceso y proporcionan una interfaz estándar para su acceso, permitiendo conectarse a cualquier base de datos. El sistema permite almacenar los datos guardando todos los valores recogidos o mediante diezmado de valores, añadiendo como característica adicional, el marcado temporal de valores para mejorar el rendimiento de las consultas. Para suministrar la información almacenada, el sistema cuenta con varios proveedores de datos, eligiendo el origen más óptimo para cada consulta, permitiendo devolver datos históricos de un año en apenas un segundo.

El sistema **IDbox** proporciona a través de sus clientes, herramientas de análisis desarrolladas con la última tecnología Web (Ext Net MVC, Ajax, y SVG), lo que permite incrementar la productividad del usuario disminuyendo el impacto de instalaciones en los sistemas y la gestión de mantenimiento del programa cliente. Los diferentes clientes del sistema IDbox ofrecen al usuario gráficas de tendencia de históricos y tiempo real, facilitan la exportación de datos a herramientas ofimáticas, visualiza eventos, notificaciones, alarmas... independiente de la plataforma tecnológica y del navegador a utilizar.

El objetivo final es permitir mediante herramientas analíticas y gráficas, la realización de cálculos y análisis de información (en tiempo real o histórica) y de este modo, favorecer la creación de un escenario de ayuda, tanto en la toma de decisiones, como en la propia operación.

Hoy en día no se concibe ninguna operación industrial sin una correcta supervisión y monitorización. En este sentido existe en el mercado multitud de sistemas denominados SCADA[Rod2012], que se encargan de la supervisión de diferentes sistemas.

**IDbox** integra toda la información necesaria para la operación y supervisión y la pone a disposición de todas las personas de la organización y sistemas externos, permitiendo realizar un análisis efectivo, tanto en situaciones normales de operación, como ante disparos u otras alteraciones en la producción.

**IDbox** es un sistema configurable capaz de conectar cualquier dispositivo y almacenar en cualquier base de datos (Mongo DB por defecto, SQLServer, Oracle, MySQL, etc.). Su arquitectura de información lo convierte en un sistema de consulta de históricos muy rápido y eficiente. Dispone de múltiples clientes (web, mobile, Excel y Kinect).

En definitiva **IDbox** permite al usuario centralizar todas las señales de su sistema en un único lugar, pudiendo acceder a sus datos en todo momento, obteniendo estadísticas y gráficos de tendencia, valores en tiempo real, etc. En la Figura 1 podemos ver un esquema del funcionamiento de la aplicación IDbox, pasando por todas las etapas desde el origen de los datos hasta que los datos se consultan a través de un PC, móvil, etc.

En este Proyecto se va a desarrollar el cliente mobile para **IDbox**, que permite a los usuarios disponer de gran parte de la funcionalidad que ofrece la versión web de manera ágil y rápida desde su teléfono móvil, sin la necesidad de depender de un PC, portátil o similar. Gracias a esta aplicación los empleados de planta pueden, por ejemplo, acceder a la información del sistema desde cualquier lugar mejorando la productividad y eficiencia del trabajo realizado. También permite obtener un control total del funcionamiento de la planta, ya que dispone de unas secciones de indicadores y alarmas que permiten comprobar en cualquier momento que todo funciona de manera óptima, funcionalidad que actualmente no está implementada en la versión web de **IDbox**.



Figura 1 - Esquema IDbox

### 1.2 Objetivo del proyecto

El objetivo de este proyecto es conseguir exportar la mayor funcionalidad posible desde la aplicación web a la aplicación móvil. Debido a las limitaciones de los dispositivos móviles algunas de estas funcionalidades no podrán ser exportadas, como por ejemplo exportar búsquedas a Excel. Por lo tanto, un primer objetivo será analizar las distintas funcionalidades que proporciona **IDbox** desde su aplicación web y determinar cuáles son exportables a un dispositivo móvil.

Gracias a la aplicación móvil los usuarios tendrán la posibilidad de consultar datos, tanto en tiempo real como históricos, consultar gráficos de tendencia, comprobar si se ha producido alguna alarma en el sistema, etc. Y todo esto podrán realizarlo con su teléfono móvil desde cualquier lugar en el que se encuentren, siempre que disponga de cobertura para poder comunicarse con los servicios web.

Por lo tanto los objetivos de este proyecto son:

- Desarrollar una aplicación web, ya que permitirá abarcar un mayor número de dispositivos móviles al ser accesible desde más de un sistema operativo diferente.
- Desarrollar una sección de análisis que permita a los usuarios realizar consultas de datos en tiempo real, datos históricos, graficas de tendencia... entre otras funcionalidades.
- Desarrollar una sección de indicadores que permita a los usuarios obtener una visión rápida y visual del estado actual del sistema.
- Desarrollar una sección de alarmas que recoja todas las alarmas que se han producido en el sistema en un periodo determinado de tiempo.
- Desarrollar una sección de gráficos sinópticos que permitirá visualizar datos en tiempo real de una manera más visual y atractiva para el usuario.

### 1.3 Ambiente de la aplicación

La aplicación está destinada a ser utilizada a través de smartphones, ya que para tablets la aplicación web funciona de forma razonable y dispone de mayor funcionalidad que la versión móvil.

Dado que para el desarrollo de la aplicación se utilizará el framework JavaScript Sencha Touch[Kum2011], las plataformas a las que va enfocada esta aplicación son principalmente iOS[iOS] de Apple y Android[Android] de Google. En un futuro se espera que Sencha Touch permita soportar Windows Phone, ya que en la actualidad no lo hace debido a su operador nativo Internet Explorer Mobile. Sencha Touch[CB2012] solo es soportado por navegadores basados en WebKit, como es el caso de las plataformas iOS y Android que actualmente utilizan Safari como navegador nativo.

## **1.4 Estructura del documento**

En el siguiente capítulo de este documento se realizará una descripción de la aplicación y cómo ha sido planificada. Posteriormente en el capítulo 3 se describirá la tecnología utilizada en el desarrollo de la aplicación, para continuar en el capítulo 4 con la descripción detallada de su desarrollo. En el capítulo 5 se describe el proceso de evaluación y las pruebas realizadas durante su puesta a punto, para terminar, en el capítulo 6 se mostrarán las conclusiones extraídas del desarrollo del proyecto así como las posibles mejoras futuras de la aplicación.

## 2 Descripción y planificación

### 2.1 Ámbito Funcional

La aplicación está destinada al entorno industrial, siendo actualmente las centrales nucleares su principal nicho de mercado. También abarca otros sectores como pueden ser energías renovables, industria química y petrolífera, SmartCity, SmartGrid, etc.

La aplicación permite al usuario obtener información de las señales de su sistema en tiempo real, datos históricos, etc. A partir de los datos históricos obtenidos, el usuario tiene la posibilidad de realizar gráficos de tendencia, pudiendo ampliar la resolución con la que se obtienen los datos hasta el rango de milisegundos.

Otra funcionalidad importante para el entorno industrial donde se va a implantar la herramienta es la sección de **Indicadores** y la sección de **Alarmas**. Estas dos secciones permitirán al usuario o usuarios de la aplicación obtener de manera rápida y ágil información del estado del sistema en ese momento, pudiendo comprobar también las alarmas que se han producido, clasificadas por su importancia, etc.

### 2.2 Descripción de la aplicación

La aplicación permitirá al usuario realizar una serie de funcionalidades que serán detalladas a continuación. La aplicación se diseñará acorde a las funcionalidades que ofrece la herramienta **IDbox**, añadiendo para esta versión móvil las secciones de **Indicadores** y **Alarmas** que actualmente no están disponibles en la versión web y que se quieren implementar en esta versión para enriquecer el producto.

En la pantalla principal, que es la pantalla que arrancará cuando se inicie la aplicación, aparecerán los cuatro botones principales (**Análisis**, **Indicadores**, **Alarmas** y **Sinópticos**), además de un enlace en la parte superior derecha de la pantalla que permite acceder a la versión web (diseñada para acceder desde ordenadores portátiles, sobremesa, tablets, etc.). Esta opción es muy importante ya que permite al usuario acceder desde el móvil a la versión web de IDbox, que ofrece una mayor funcionalidad que la versión móvil. En caso de que el usuario necesite alguna funcionalidad de la que no dispone la versión móvil, a través de ese enlace podrá acceder a la versión web y realizar la operación. Al acceder a la dirección web de la aplicación IDbox, el servidor redirigirá a la versión web o versión móvil en función del tipo de dispositivo con el que se acceda.

Los cuatro botones de la aplicación permitirán acceder a las diferentes secciones de la aplicación, en las que el usuario podrá:

- Visualizar indicadores de planta, incluyendo gráficas y valores en tiempo real
- Analizar los datos obtenidos en tiempo real o a partir de históricos
- Visualizar alarmas que se producen en el sistema
- Visualizar sinópticos

A continuación serán detalladas las características de cada una de estas secciones.

### 2.2.1 Sección Indicadores

En la sección **Indicadores** el usuario podrá obtener valores en tiempo real de las señales más críticas de su sistema. Esta sección está diseñada para adaptarse a las necesidades de cada cliente, por lo que en este proyecto se presentará la maqueta inicial. La sección indicadores dispondrá de un panel en el que aparecerán una serie de luces, que tendrán el color verde si los datos que reciben están dentro de los parámetros correctos o rojo en caso contrario. Además en esta versión inicial se añadirán varias tablas con diferentes datos a modo de ejemplo para que los clientes interesados puedan hacerse una idea de la funcionalidad que ofrece la herramienta.

Para acceder a la sección **Indicadores**, el usuario deberá seleccionar el botón de la pantalla inicial con idéntico nombre. En la pantalla de **Indicadores**, el usuario podrá visualizar datos sobre el estado del sistema en tiempo real. En un futuro, y en función de las peticiones del cliente, esta sección permitirá al usuario elegir las señales que desea visualizar, obteniendo así un acceso rápido a las señales consideradas más críticas en el sistema.

### 2.2.2 Sección Alarmas

En la sección **Alarmas**, se proporcionará información al usuario acerca del número de alarmas que se han producido en el sistema clasificándolas por tipos (Alta, Crítica, Baja). Estas alarmas se obtendrán de un servicio web que irá reportando las alarmas que se produzcan en el sistema y el tipo de importancia de cada una de ellas. En función de la importancia de cada alarma, éstas se asignaran a uno u otro tipo. Cuando el usuario coloca el dispositivo en posición tumbada o **“landscape”**, se muestra una gráfica con las estadísticas de dichas alarmas.

A través del uso de esta funcionalidad, el usuario puede saber si el número de alarmas y el tipo de las mismas es el adecuado para el correcto funcionamiento del sistema o, en caso contrario, proceder a analizar los problemas que puedan estar causando un mal funcionamiento del mismo.

### 2.2.3 Sección Sinópticos

En esta sección, el usuario podrá visualizar una serie de gráficos sinópticos con valores en tiempo real sobre el estado de las señales de la planta. Un gráfico sinóptico está formado por un gráfico vectorial SVG[Eis2002] en el cual se incluyen una serie de señales de las que se obtienen datos en tiempo real. Se utilizan SVGs ya que estos sinópticos ya habían sido diseñados para la versión web de IDbox y además, al ser gráficos vectoriales, permiten que al realizar zoom sobre dichos gráficos no se pierda calidad en la imagen.

Mediante estos gráficos sinópticos, el usuario puede comprobar el estado del sistema en tiempo real, observando los valores que se van obteniendo en cada una de las señales de los gráficos sinópticos, además de contextualizar cada una de las señales dentro de las secciones de las que dispone la planta.

Por ejemplo, en la Figura 2 se observa un gráfico sinóptico en el que aparece una vasija industrial con diferentes valores que van apareciendo en tiempo real.

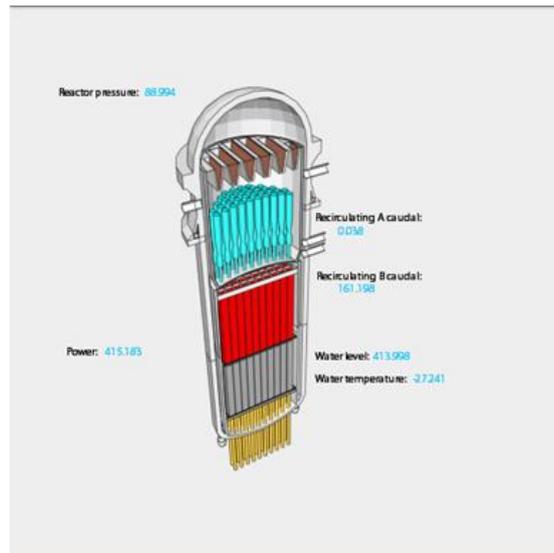


Figura 2 - Gráfico Sinóptico

#### 2.2.4 Sección Análisis

Por último, la sección **Análisis**. Esta sección es la que más funcionalidad tiene y la que ofrece al usuario una mayor cantidad de posibilidades para acceder a la información del sistema.

Cuando el usuario pulsa en el botón **Análisis** de la pantalla principal, accede a la pantalla **Consulta**. En esta pantalla el usuario tiene la posibilidad de buscar pids, que son los puntos o señales del sistema, tanto por id como por descripción además de poder elegir también la agrupación en la que desea realizar la búsqueda. También en esta pantalla, aparece por primera vez la **barra de herramientas** situada en la parte inferior de la misma y que permite acceder a los datos en tiempo real, históricos, gráficas de tendencia y agrupaciones temporales.

Desde esta pantalla se accede a las siguientes opciones:

- Barra de Herramientas
- Filtros
- Tiempo Real
- Datos Históricos
- Gráficas
- Agrupaciones Temporales

### 2.2.4.1 Barra de Herramientas

La **barra de herramientas** dispone de cinco botones que permiten realizar diversas acciones.

- El botón **Home** permite volver a la pantalla principal de la aplicación desde cualquier pantalla de la sección **Análisis** en la que se encuentre el usuario. En el resto de secciones, por ejemplo **Indicadores**, este botón no es necesario ya que nunca se avanza más de dos ventanas, por lo que el retorno una a una no es traumático.
- El botón **Tiempo Real** permite obtener valores en tiempo real de los pids seleccionados. Para ello es necesario seleccionar previamente al menos un pid de la lista. En caso de no seleccionar ningún pid, la aplicación muestra al usuario un aviso por pantalla informándole que debe seleccionar algún pid.
- El botón **Históricos** permite obtener valores históricos de los pids seleccionados. Al igual que para el botón Tiempo Real, es necesario seleccionar previamente al menos un pid de la lista y en caso contrario, la aplicación mostrara un mensaje por pantalla.
- El botón **Gráficas** permite obtener gráficas de tendencia de los pids seleccionados. Al igual que las dos opciones anteriores requiere al menos un pid seleccionado previamente.
- El botón **Agrupación temporal** permite al usuario consultar la agrupación de puntos que haya añadido previamente en la pantalla de **Filtro**. En caso de que no exista ningún punto agregado en una agrupación temporal, este botón mostrará un mensaje al usuario informándole de este hecho.

### 2.2.4.2 Pantalla Filtro

Una vez introducidos los parámetros que deseamos consultar, avanzamos a la pantalla **Filtro** en la cual obtenemos un listado con los pids que cumplen dichos criterios. Para realizar este filtro, podemos elegir en la pantalla **Consulta** obtener valores por id o descripción, además de poder seleccionar la agrupación en la cual se desea realizar la búsqueda. A partir de este listado, el usuario puede:

- Seleccionar uno o varios pids y visualizar su valor en tiempo real pulsando el botón de la barra de herramientas **"Tiempo Real"**.
- Seleccionar uno o varios pids y visualizar su valor histórico
- Seleccionar uno o varios pids y obtener sus gráficas de tendencia
- Seleccionar uno o varios pids y agregarlos a una agrupación temporal. Para ello el usuario debe seleccionar los pids que desea y pulsar el botón **"Añadir pids a agrupación temporal"**, el cual se muestra en la Figura 3.



Figura 3 - Botón añadir puntos agrupación

### **2.2.4.3 Pantalla Tiempo Real**

La pantalla **Tiempo Real** permite al usuario visualizar valores en tiempo real de los pids seleccionados. Dichos pids son seleccionados por el usuario en la pantalla **Filtro**. Además, el usuario dispone de la opción Pausar/Reanudar la ejecución en tiempo real.

Cuando el usuario gira el dispositivo, colocándolo en posición **"landscape"**, la pantalla **Tiempo Real** muestra una gráfica en la que van apareciendo las señales en tiempo real.

### **2.2.4.4 Pantalla Datos Históricos**

En esta pantalla el usuario puede visualizar los datos históricos de los pids seleccionados. Debido a las limitaciones de pantalla de los dispositivos móviles, solo se pueden visualizar los datos históricos de un único pid a la vez.

En caso de que el usuario seleccione más de un pid, los datos históricos mostrados corresponderán al primer pid seleccionado. Para poder visualizar los datos históricos del resto de pids seleccionados, el usuario deberá acceder a la pantalla de **configuración de datos históricos** pulsando el botón Configuración situado en la parte superior derecha de la pantalla. Una vez pulsado, aparecerá una nueva ventana que será detallada a continuación.

#### **2.2.4.4.1 Pantalla de configuración datos históricos**

Esta pantalla de configuración, permite al usuario configurar los datos históricos de varios pids seleccionados, con diferentes tipos de resoluciones. Al acceder a esta pantalla, el usuario puede seleccionar una de las configuraciones definidas: **"Predefinido"** y **"Personalizado"**.

- **Predefinido**, el usuario dispone de la posibilidad de elegir el pid que desea visualizar dentro de los seleccionados previamente, así como la resolución con la que quiere obtener los datos históricos de dicho pid
- **Personalizado**, el usuario puede seleccionar el pid que desea visualizar dentro de los seleccionados previamente, así como la información que quiere visualizar: el intervalo de tiempo que desea analizar, introduciendo la fecha inicial y la fecha final, y la resolución de dichos datos.

### **2.2.4.5 Pantalla gráfica**

En esta pantalla el usuario puede ver gráficas de tendencia a partir de los valores históricos de los pids seleccionados en la pantalla de **Filtro**. También se ofrece la posibilidad de realizar zoom sobre la gráfica, modificándose automáticamente la resolución con la que el sistema retorna los datos, adaptándolos a los nuevos intervalos seleccionados por el usuario.

En esta pantalla también existe la posibilidad de configurar la resolución y los intervalos a aplicar sobre la gráfica. Para ello el usuario debe pulsar en el botón "Configurar gráficas".

#### **2.2.4.5.1 Pantalla de configuración gráficas**

Al acceder a esta pantalla, el usuario puede seleccionar una de las configuraciones definidas: **“Predefinido”** y **“Personalizado”**.

- **Predefinido:** el usuario puede elegir el tipo de resolución que desea
- **Personalizado:** el usuario puede también elegir el tipo de resolución que desea, así como establecer un intervalo temporal eligiendo la fecha de inicio y la fecha final

#### **2.2.4.6 Pantalla de Agrupaciones temporales**

La pantalla de **Agrupaciones Temporales** permite al usuario consultar las agrupaciones temporales creadas previamente. Desde esta pantalla el usuario puede realizar todo tipo de acciones con estos pids, como pueden ser consulta de históricos, consulta datos en tiempo real o gráficas de tendencia.

También el usuario tiene la posibilidad de eliminar pids de la agrupación, seleccionándolos previamente y pulsando a continuación el botón **“Eliminar”**, situado en la esquina superior derecha de la pantalla.

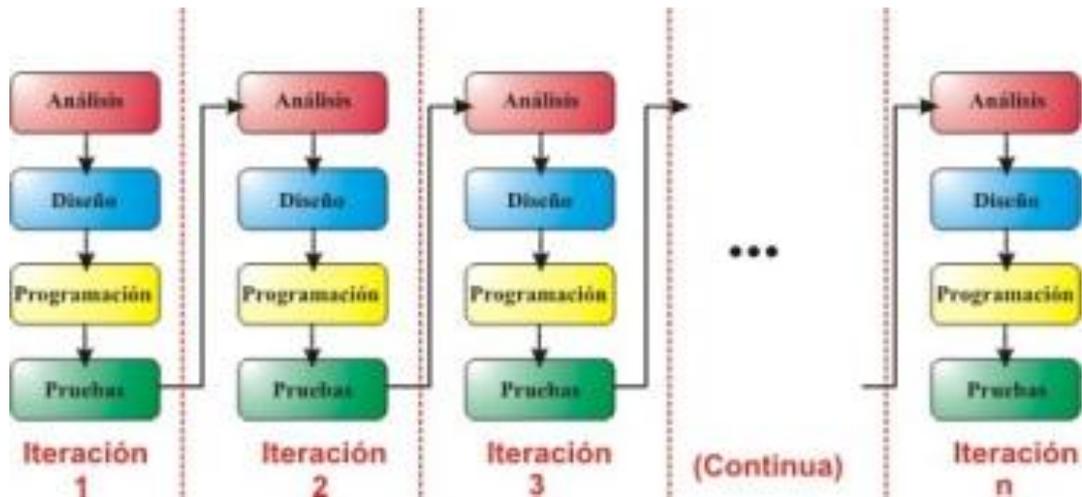
### **2.3 Metodología de desarrollo**

Para el desarrollo de este proyecto se ha utilizado una metodología ágil como es la metodología iterativa incremental[GIEP2012]. Esta metodología permite corregir las debilidades de la metodología tradicional en cascada, como la no necesidad de disponer de unos requisitos completamente definidos al comienzo del desarrollo, permitiendo variaciones a lo largo del mismo. El cliente puede definir a grandes rasgos la funcionalidad que debe ofrecer la aplicación a desarrollar, y en función de esos requerimientos se elaboran una serie de iteraciones, donde en cada una de ellas se desarrollará parte de la funcionalidad de la aplicación software teniendo prioridad aquellas funcionalidades que son más importantes.

El uso de esta metodología permite poder entregar versiones parciales del producto a medida que se va construyendo, permitiendo al cliente poder involucrarse en el desarrollo de forma importante. Cada versión generada se denomina iteración e incorpora los requisitos de funcionalidad y diseño añadidos en anteriores iteraciones. Además, en cada iteración se van agregando nuevas funcionalidades y/o requisitos o mejorando las posibles deficiencias detectadas en anteriores iteraciones.

Durante las primeras iteraciones, se busca obtener un producto con unas funcionalidades básicas que satisfaga los requerimientos principales del sistema software para que el cliente pueda utilizar el producto desde que se finaliza la primera iteración.

En definitiva, el modelo incremental utiliza un desarrollo modular, permitiendo obtener entregas parciales del producto, denominadas incrementos, hasta alcanzar el producto final como se puede observar en la Figura 4.



**Figura 4 - Metodología incremental**

Para el desarrollo de este proyecto se ha elegido utilizar una metodología incremental debido al desconocimiento de que cantidad de funcionalidad de la aplicación web se podría exportar a la aplicación móvil permitiendo que la usabilidad de la aplicación fuese óptima. Debido a esto, lo primero que se realizó fue una primera iteración en la que se incluyeron funcionalidades muy básicas para poder comprobar su funcionamiento en un dispositivo móvil. Una vez comprobado que la aplicación se comportaba de manera óptima con estos requerimientos, se fueron incorporando más funcionalidades en las siguientes iteraciones, hasta llegar a la versión final del producto.

Se trata de un modelo evolutivo, en el cual los requisitos del producto suelen cambiar a lo largo del tiempo por lo que debe permitir ser modificados sin que ello suponga reiniciar el desarrollo de la aplicación. Por ejemplo en la metodología en cascada y cascada realimentado no se permite realizar variaciones una vez comenzado el desarrollo ya que los requisitos se suponen estáticos y definidos desde el comienzo del desarrollo.

En la Figura 5 podemos visualizar un diagrama en el que se muestran las etapas de un desarrollo evolutivo, pudiendo observar como obtenemos versiones intermedias del producto a medida que se van desarrollando iteraciones en función a unas especificaciones definidas y una vez validadas estas versiones se llega a la versión final del producto software.



Figura 5 - Diagrama desarrollo evolutivo

## 2.4 Patrón de diseño MVC

En cuanto al patrón de diseño, para realizar este proyecto se ha elegido el patrón MVC [FFB2004]. Este patrón es utilizado frecuentemente en el desarrollo de aplicaciones web, como la aplicación de este proyecto. El patrón MVC está formado por los componentes **Modelo**, **Vista** y **Controlador**. El modelo hace referencia a los datos con los que la aplicación va a interactuar; la vista se encarga de transformar ese modelo en una interfaz gráfica con la que el usuario interactúa y el controlador se encarga de atender los eventos que se produzcan en la aplicación y responder a ellos.

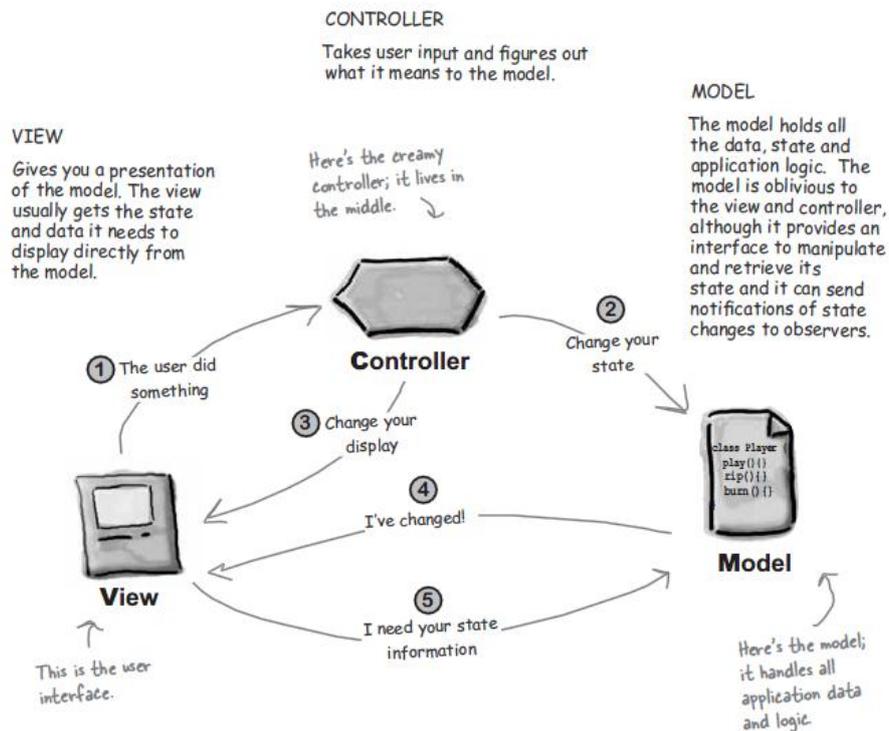


Figura 6 - Patrón MVC

Como se puede apreciar en la Figura 6, el diagrama de flujo de una aplicación que sigue el patrón MVC es el siguiente [FFB2004]:

1. El usuario realiza una acción sobre la interfaz de dicha aplicación. Esta acción provoca que se lance un evento que es capturado por el controlador.
2. El controlador comprueba si el modelo ha cambiado su estado a consecuencia de la acción realizada por el usuario en el paso 1.
3. El controlador comprueba si la acción realizada por el usuario en la vista requiere de alguna modificación en la propia vista. Por ejemplo habilitar o deshabilitar algún botón de la pantalla, etc.
4. El modelo notifica a la vista cualquier cambio que se produzca en su estado, ya sea por una acción realizada por el usuario o por cambios internos que se produzcan.
5. La vista solicita al modelo consultar su estado para mostrar el estado actual del modelo de datos. Esta petición al modelo de datos también puede ser a consecuencia de algún evento manejado por el controlador y realizado en la vista.

### 2.5 Requisitos de alto nivel de la aplicación

Una vez analizado el diseño de la aplicación y la funcionalidad que tendrá la misma, se procede a definir unos requisitos de alto nivel para cada uno de las iteraciones de las que constara el desarrollo del proyecto.

#### Iteración 1

1. La aplicación deberá ser una aplicación web que permita poder acceder a ella desde diferentes tipos de smartphones.
2. La aplicación deberá ser desarrollada utilizando el framework **Sencha Touch**, utilizando para ello el entorno **Visual Studio**.
3. La aplicación deberá ser capaz de exportar la mayor cantidad posible de funcionalidad de la aplicación web ya desarrollada.
4. La aplicación móvil deberá tener un acceso directo en la pantalla principal a la versión web de **IDbox**.
5. La aplicación deberá conectarse a una serie de servicios web de los que obtendrá los datos con los que operar.
6. Cuando se realizan peticiones a un web service, la aplicación debe mostrar mascarar de carga hasta que se obtenga una respuesta de dicho servicio.
7. Cuando el usuario desee realizar un filtrado de datos, podrá elegir filtrar por PID, descripción o agrupación en la que se encuentre dicho punto.
8. Una vez el usuario pulse el botón **"Filtrar"**, la aplicación mostrará una lista con los puntos que cumplen los criterios seleccionados en el filtro.
9. La lista con los puntos filtrados, mostrará el PID que identifica a cada punto y el tipo de dato que es, analógico o digital.

### **Iteración 2**

10. La aplicación permitirá realizar consultas de datos en tiempo real e históricos
11. Para consultar datos en tiempo real, el usuario debe realizar primero un filtrado de datos, pudiendo elegir seleccionar todos los puntos del sistema si así lo requiere, o realizar otros tipos de filtrado como los ya explicados en la iteración 1.
12. Una vez realizado el filtro, el usuario podrá seleccionar uno o varios pids de la lista y pulsando el botón **"Tiempo Real"** obtendrá una vista en la que se mostrarán los puntos elegidos con sus valores en tiempo real.
13. En la pantalla de **Tiempo Real** el usuario dispondrá de un botón que permitirá pausar y reanudar la petición de datos en tiempo real.
14. Una vez realizado el filtro, el usuario podrá seleccionar uno o varios pids de la lista y pulsando el botón **"Datos Históricos"** obtendrá una lista en la que se mostrarán los valores históricos desde esa fecha hasta un año antes.
15. Si el usuario no selecciona previamente ningún punto del filtro, la aplicación mostrará un mensaje emergente informando al usuario de la necesidad de seleccionar al menos un punto antes de realizar consultas de datos en tiempo real e/o históricos.
16. Cuando el usuario se encuentre en la pantalla donde visualiza los datos en tiempo real, podrá girar el dispositivo móvil obteniendo una grafica en tiempo real con los valores de los puntos seleccionados previamente.
17. La pantalla de datos históricos dispondrá de un botón de configuración en el que se permitirá al usuario modificar la resolución con la que se obtienen los valores, así como la posibilidad de elegir el PID del que se quieren visualizar los datos de los seleccionados previamente. Además, esta opción permitirá al usuario elegir el intervalo de fechas en el que quiere obtener los datos históricos del sistema.

### **Iteración 3**

18. La aplicación permitirá mostrar gráficas de tendencia a partir de datos históricos
19. Para realizar gráficas de tendencia, el usuario debe realizar primero un filtrado de datos, pudiendo elegir seleccionar todos los puntos del sistema si así lo requiere, o realizar otros tipos de filtrado como los ya explicados en la iteración 1.
20. Una vez realizado el filtro, el usuario podrá seleccionar uno o varios pids de la lista y pulsando el botón **"Gráficas"** obtendrá una gráfica en la que se mostrara la tendencia que siguen los datos a lo largo del tiempo con una resolución de 1 año(1 día).
21. Cuando el usuario realice el gesto de **"zoom"** en la gráfica, la resolución con la que se obtienen los datos variará, llegando a resoluciones de milisegundos.
22. La gráfica dispondrá de un botón de **"Reset"** que permitirá reducir de nuevo la resolución hasta 1 año(1 día).
23. La pantalla de gráficas de tendencia dispondrá de un botón de configuración en el que se permitirá al usuario modificar la resolución con la que se obtienen los valores, así como los intervalos de fechas en los cuales se quiere realizar la petición de datos.

#### **Iteración 4**

24. La aplicación permitirá agregar puntos a agrupaciones temporales con el fin de facilitar las operaciones que deseen realizar los usuarios.
25. Si el usuario pulsa el botón "**Agrupación temporal**" antes de haber agregado ningún punto a dicha agrupación, la aplicación mostrara un mensaje emergente informando al usuario que no existe ningún punto en dicha agrupación temporal.
26. Para agregar puntos a agrupaciones temporales el usuario deberá seleccionar previamente uno o varios pids de la lista y pulsar el botón "**Agregar**".
27. Si el usuario no selecciona ningún punto previamente, la aplicación mostrara un mensaje emergente informando al usuario de la necesidad de seleccionar al menos un punto para poder agregarlo a la agrupación temporal.
28. Cuando el usuario selecciona uno o varios pids y pulsa el botón "**Agregar**", el/los punto/s seleccionado/s son agregados a la agrupación temporal, aparecerá un mensaje emergente informando del correcto funcionamiento de la acción.
29. Cuando el usuario selecciona uno o varios pids y pulsa el botón "**Agregar**", el/los punto/s seleccionado/s son agregados a la agrupación temporal y dicha selección es eliminada, permitiendo volver a elegir otros puntos y agregarlos a la agrupación temporal.
30. Si un pid ya ha sido agregado a una agrupación temporal, no podrá ser agregado de nuevo a dicha agrupación.
31. Desde la agrupación temporal, el usuario podrá realizar todas las acciones que requiera con dichos puntos, como son consultar datos en tiempo Real e históricos y graficas de tendencia.
32. Desde la agrupación temporal, el usuario podrá eliminar puntos de la agrupación seleccionándolos previamente y pulsando en el botón "**Eliminar**".
33. Si el usuario no selecciona ningún punto antes de pulsar el botón eliminar, la aplicación mostrara un mensaje emergente informando al usuario de que se requiere al menos seleccionar un punto para poder eliminar.

#### **Iteración 5**

34. La aplicación dispondrá de una sección denominada **Indicadores** en la que aparecerán datos de señales en tiempo real, así como otros tipos de datos elegidos por el cliente.
35. La aplicación dispondrá de una sección denominada **Alarmas** que permitirá a los usuarios comprobar el correcto funcionamiento del sistema a partir de los datos que se obtengan a través de listados con el número de alarmas producidas o bien a través de gráficos de barras.
36. Cuando el usuario se encuentra dentro de la sección de **Alarmas**, puede girar el dispositivo y obtener un gráfico de barras con la cantidad de alarmas producidas y el tiempo de las mismas.

### **Iteración 6**

37. La aplicación dispondrá de una sección denominada **“Sinópticos”** en la que el usuario podrá visualizar gráficos sinópticos con señales en tiempo real.
38. El usuario podrá realizar el gesto táctil **“pinch to zoom”** para realizar un zoom sobre el gráfico y visualizar mejor los valores que en él se muestran.

## **2.6 Sumario**

En este capítulo se ha descrito el ámbito en el que será utilizada la aplicación, así como los aspectos a desarrollar en la misma, haciendo una descripción de todas las secciones y pantallas que forman parte de la aplicación.

También se ha descrito la metodología utilizada así como el patrón de diseño y los requisitos de alto nivel de la aplicación, que marcarán el desarrollo del proyecto.

### 3 Herramientas y tecnologías utilizadas

En el desarrollo de este proyecto se han utilizado principalmente los lenguajes orientados al desarrollo web **JavaScript**[Fla2011], **HTML5**[Firt2012] y **CSS3**[BSS2011].

**JavaScript** es un lenguaje interpretado, orientado a objetos que se ejecuta normalmente en el lado cliente. Se utiliza principalmente en navegadores web, permitiendo utilizar scripts para alterar el contenido de la página **HTML**. Mediante la utilización del lenguaje **JavaScript** el programador puede realizar páginas web en las que aparecen animaciones, eventos como pulsar un botón y otras funcionalidades. Al ser un lenguaje interpretado, **JavaScript** no necesita de un compilado previo a su ejecución. A pesar de su nombre, **JavaScript** no guarda ninguna relación directa con el lenguaje de programación **Java**.

**HTML5** es la última evolución del lenguaje **HTML**, un lenguaje utilizado para el desarrollo de páginas y aplicaciones web. Respecto a anteriores versiones, **HTML5** introduce nuevos elementos y atributos adaptados a los navegadores presentes y futuros.

**CSS3** es también la última evolución del lenguaje **CSS** que es un lenguaje utilizado para definir el diseño de un documento escrito en lenguaje **HTML** o **XML**. Se utiliza para separar el desarrollo de la aplicación en sí, de la parte de diseño y presentación.

En cuanto a las herramientas utilizadas en el desarrollo del proyecto han consistido en el entorno de desarrollo **Microsoft Visual Studio 2010** y los navegadores web **Google Chrome** y **Safari** para probar y depurar la aplicación desde el mismo PC. Una vez comprobada la aplicación en el entorno de escritorio se procede a publicar la aplicación móvil en el servidor web de IDbox y realizar las mismas pruebas sobre un smartphone para comprobar que todo sigue funcionando de manera adecuada.

#### 3.1 Herramientas utilizadas durante el proceso de desarrollo

**Microsoft Visual Studio 2010**[RGAM2010] es un entorno de desarrollo para Windows, permitiendo desarrollar aplicaciones en varios lenguajes de programación como son Visual C++, Visual C#, Visual J# o Visual Basic .NET.

Por su parte **Google Chrome** y **Safari** son los navegadores web que mejor funcionan con **Sencha Touch**, debido a que ambos están basados en **Webkit**. Esta es una de las razones por las que **Windows Phone** no soporta el framework **Sencha Touch**, ya que implementa el navegador **Microsoft Internet Explorer**, el cual no soporta aplicaciones basadas en **HTML5** y **CSS3**.

Para comprobar el correcto funcionamiento en los dispositivos móviles, se han utilizado los siguientes dispositivos:

- **Iphone** (Apple iOS)
- **Ipod** (Apple iOS)
- **Samsung Galaxy Nexus** (Google Android)

## 3.2 Tecnología utilizada

### 3.2.1 Sencha Touch

Debido al constante crecimiento de los dispositivos móviles entre nosotros, ha surgido la necesidad de realizar desarrollos software para este tipo de dispositivos. El problema inicial a la hora de desarrollar aplicaciones para dispositivos móviles es que en la actualidad existen multitud de plataformas diferentes, teniendo cada una de ellas un lenguaje de programación distinto con el que desarrollar.

Para hacer frente a esta problemática surge **Sencha Touch**, que elimina esta limitación permitiendo realizar aplicaciones web a partir de su framework basado en **JavaScript**, **HTML5** y **CSS3**. Utilizando este framework los desarrolladores pueden crear aplicaciones para varias plataformas sin necesidad de tener que reescribir el código, lo que supone una gran ventaja.

En cambio, un primer problema de una aplicación web respecto a una aplicación nativa es que depende de un navegador web para poder ser ejecutada, lo que puede causar algunos problemas, como que la aplicación se comporte de manera diferente en función del navegador, que algunos elementos no se muestren, que los elementos **JavaScript** que un navegador funcionan perfectamente en otro no lo hagan, etc. Por ello para el desarrollo de este proyecto hemos utilizado los navegadores **Google Chrome** y **Safari** ya que ambos están basados en **WebKit**. **WebKit** es un motor de navegación web open-source que básicamente controla como se muestran las páginas web, maneja los elementos **JavaScript** e implementar los estándares web.

Otro problema es que una aplicación web no se instala en el dispositivo, mientras que una aplicación nativa si. La aplicación web se instala en un servidor web, en este caso en el servidor en el que también esta alojada la versión de escritorio de **IDbox**, y los usuarios acceden a ella a través de una dirección web (Figura 7). Gracias a esto, la aplicación puede ser utilizada desde cualquier dispositivo que disponga de una navegador web, sin necesidad de que sea exclusivamente el teléfono móvil, aunque ese es el dispositivo para el que la aplicación ha sido desarrollada y optimizada[CB2012].

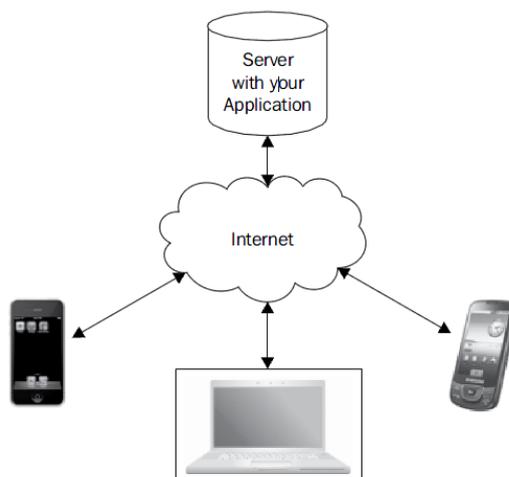


Figura 7 - Aplicación web

### **3.2.2 Servicios Web**

La aplicación IDbox mobile obtiene los datos con los que realizar las operaciones de una serie de servicios web. Un servicio web es un sistema software diseñado para permitir la comunicación entre diferentes maquinas, con diferentes plataformas y con programas distintos [CCV2002]. Dichos servicios web ofrecen datos de tipo histórico y datos en tiempo real, además de ofrecer el listado con todas las señales de las que dispone el sistema.

### **3.3 Sumario**

En esta sección se han descrito las herramientas utilizadas para desarrollar y depurar la aplicación así como la tecnología utilizada. En cuanto a la tecnología se ha optado por desarrollar la aplicación utilizando el framework Sencha Touch y para obtener los datos con los que trabaja la aplicación se utilizan servicios web.



## 4 Desarrollo de la aplicación

Como se ha visto en el capítulo anterior, el desarrollo de esta aplicación se ha dividido en un total de seis iteraciones y en este capítulo se detallarán cada una de ellas. En cada iteración, el sub-apartado de desarrollo sirve para explicar como se ha ido construyendo la aplicación desde su fase inicial hasta alcanzar la aplicación final.

### 4.1 Iteración 1: Desarrollo de pantalla principal y filtrado de datos

La primera iteración tiene como principal objetivo el comenzar a desarrollar la pantalla principal, diseñando los botones y funcionalidad de la misma. La primera decisión a tomar entonces en este momento es considerar que secciones o apartados formarán parte de la aplicación y como se accederá a dichos apartados. Tras analizar varias opciones se llega a la conclusión de que lo más adecuado para la correcta funcionalidad y usabilidad de la aplicación es dotar a la pantalla principal de cuatro botones, que permitirán a los usuarios acceder a las secciones principales de la aplicación. También se decide que la aplicación web deberá incluir un enlace a la versión web de **IDbox** permitiendo a los usuarios acceder a dicha aplicación si así lo requieren.

La pantalla principal contará con estos cuatro botones: **Indicadores**, **Análisis**, **Alarmas** y **Sinópticos**, como podemos ver en la Figura 8. Dichos botones permitirán a los usuarios acceder a las principales secciones del sistema e interactuar con la aplicación. En esta primera iteración el único botón al que se otorgara parte de funcionalidad será el botón de **Análisis**. Dicho botón permitirá en esta primera iteración realizar el filtrado de los pids que se encuentran en el sistema.



Figura 8 - Captura pantalla principal

Al pulsar en el botón **Análisis** el usuario accederá a una pantalla de consulta (Figura 9) en la cual dispondrá de una caja de texto denominada **Buscar** y que llevará el valor establecido por defecto al valor **“Cualquiera”**. El usuario podrá introducir en esta caja de texto la descripción de los pids o el nombre del pid/s que desea buscar. También dispondrá en esta pantalla de

consulta de un **Combo box** en el que podrá elegir la agrupación en la que desea realizar la búsqueda de pids. Por defecto dicho combo box establece el valor a **“Todas”**, buscando los pids en todo el sistema. El resto de valores de las agrupaciones seleccionables en el combo son obtenidos de un servicio web, que dispone de un método diseñado para tal fin. Cada consulta a un servicio web conlleva la utilización de una máscara de carga que permite al usuario saber que la aplicación está realizando una petición de datos y no dar la falsa sensación de que algo ha fallado en la aplicación en caso de que la petición al servicio web tarde más tiempo del previsto.



**Figura 9 - Captura pantalla consulta**

En esta primera pantalla ya aparece la barra de herramientas, realizada utilizando la clase **ToolBar**. En ella se agregan los botones de **Home**, **Tiempo real**, **Históricos**, **Gráficas** y **Agrupaciones temporales**. En esta primera iteración todos estos botones están sin funcionalidad asociada, simplemente se incluyen para mostrar la funcionalidad que albergará la aplicación en iteraciones futuras.

Si el usuario mantiene las opciones por defecto y posteriormente pulsa el botón **Filtrar**, obtendrá un listado de todos los pids que existen en el sistema. La Figura 10 muestra un ejemplo del resultado obtenido a partir de una consulta realizada. En caso de que realice alguna modificación en los valores por defecto, obtendrá un listado con los pids que cumplen los requisitos que el usuario ha establecido, mostrando PID de cada punto y el tipo de dato de dicho pid (analógico o digital).



Pid	Tipo
Cau01	Analógico
Cau02	Analógico
Cau03	Analógico
Cau04	Analógico
Cau05	Analógico
Cau06	Analógico
Cau07	Analógico
Cau08	Analógico
Cau09	Analógico
Cau10	Analógico
ADEMO1A	Analógico
ADEMO2A	Analógico

Figura 10 - Pantalla de Filtro

#### 4.1.1 Desarrollo de la iteración

La base de la aplicación está sustentada en un tipo de clase de la librería de Sencha Touch denominado **NavigationView**. Esta clase funciona básicamente como una pila de ventanas, permitiendo al usuario navegar por la aplicación mediante apilado y desapilado de ventanas.

La pantalla principal es la única ventana que nunca puede ser desapilada, mientras que el resto pueden ser apiladas o desapiladas en cualquier momento.

Para realizar la pantalla principal se utilizó un tipo de clase denominada **Container**, en el cual se añadieron el logo principal de la pantalla, los cuatro botones principales de la aplicación y el enlace a la versión web de **IDbox**. Los botones se crean utilizando un tipo de clase de Sencha Touch denominada **Button**, utilizando el atributo `text` para establecer el nombre de cada uno de los botones. Para el logo de la pantalla inicial se utiliza una clase **Panel** en la que se inserta la imagen mediante un objeto de la clase **Image** en el cual se introduce la ruta en la que se encuentra la imagen, así como su alto y su ancho. Por último el enlace web se crea con un simple hipervínculo apuntando a la dirección web de la versión **IDbox** de escritorio.

En esta primera iteración solo se le otorga funcionalidad al botón de análisis. Para ello se utiliza un controlador creado utilizando la clase **Controller** que responde al evento **"tap"** que salta cuando el usuario pulsa sobre el botón al que está suscrito. El manejador del evento se encarga de crear la instancia de la ventana de consulta y se la envía a la vista para que la muestre por pantalla. En la ventana de consulta, que se genera utilizando un **Panel** al que se le añaden los diversos componentes, disponemos de un campo de búsqueda realizado utilizando un **Searchfield** y un **Combo box** de tipos de agrupaciones para que el usuario pueda filtrar datos en función de los parámetros que desee. También en esta ventana de consulta aparece el botón de filtrar, generado utilizando la misma clase que en los botones de la pantalla principal, que a su vez está suscrito a un nuevo controlador que se encarga de crear la ventana

filtro y enviársela a la vista, que solicita al modelo los datos nuevos para mostrarlos por pantalla. Por último, en esta ventana de **Consulta** aparece por primera vez la barra de herramientas generada utilizando la clase **ToolBar** a la que se añaden los botones de **Home**, **Tiempo real**, **Históricos**, **Gráficas** y **Agrupaciones temporales**.

Para obtener los datos se realizan llamadas a los servicios web correspondientes, en este caso al servicio web que retorna los pids de los que dispone el sistema. Para realizar esta petición al servicio web se utiliza el método **request** de la clase **Ajax** al que se le pasan como parámetros la dirección del servicio web, así como el filtro y la agrupación que ha seleccionado previamente el usuario. Cada vez que se realiza una llamada a un servicio web se crea una máscara de carga para informar al usuario de que la petición se está realizando y cuando ésta finaliza, la máscara se deshabilita. Para crear esta máscara se utiliza el método **setMasked** con valor **true** y **setMasked** con valor **false** para deshabilitar la máscara.

## 4.2 Iteración 2: Desarrollo de Consulta de históricos y Tiempo Real

En la segunda iteración, el objetivo es incrementar la funcionalidad de la aplicación agregando la posibilidad de realizar consultas de datos históricos y consultas en tiempo real. Estas dos nuevas funcionalidades estarán incluidas en la sección de **Análisis** y para poder utilizarlas el usuario debe haber realizado un filtro anteriormente de la manera especificada en la primera iteración.

Una vez el usuario obtiene el listado con los puntos con los que desea trabajar, dispondrá en la barra de herramientas de la parte inferior de la pantalla dos botones que le permitirán acceder a la consulta de datos en tiempo real e históricos, como se puede observar en la Figura 11.



Figura 11 - Botones Tiempo Real e Históricos

Una vez obtenido el listado de pids, el usuario podrá seleccionar uno o varios pids y posteriormente pulsar el botón **Tiempo Real** o **Datos Históricos** en función de la operación que desee realizar. Una vez seleccione alguna de estas opciones, obtendrá o bien un listado con los valores en tiempo real de los pids seleccionados o bien un listado con los datos históricos de los pids seleccionados. En el caso de los históricos, y debido a las limitaciones de los dispositivos móviles, los datos mostrados en un primer momento corresponden al primer pid seleccionado por el usuario. Para acceder al resto de pids la aplicación dispone de una opción de configuración en la pantalla de históricos y que será detallada más adelante.



Figura 12 - Pantalla Tiempo Real



Figura 13 - Pantalla Históricos

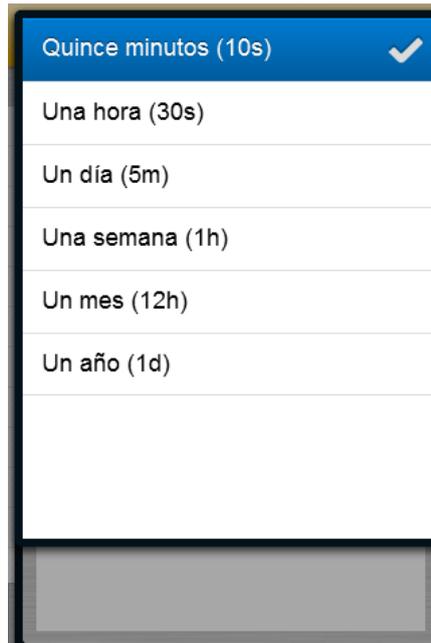
Como se puede apreciar en la Figura 12 la pantalla de **Tiempo Real** dispone de un botón en la esquina superior derecha que permite al usuario detener o reanudar la ejecución en tiempo real. Esta funcionalidad es útil en el caso de que el usuario seleccione demasiados puntos a la vez y estos no entren en su totalidad en la pantalla, haciendo difícil observar sus resultados sin detener la ejecución.

A su vez, la pantalla de históricos (Figura 13) dispone de un botón de configuración que permite al usuario acceder al resto de pids seleccionados previamente (en el caso de haber seleccionado más de uno), modificar la resolución con la que se obtienen los datos o establecer el intervalo de fechas en el cual se desea realizar la consulta de datos históricos. Esta pantalla de configuración dispone de dos pestañas **“Predefinido”** o **“Personalizado”** (Figura 14).



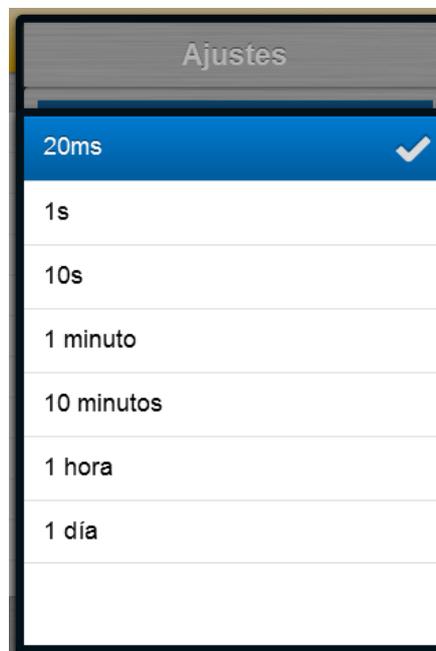
**Figura 14 - Pantalla configuración tiempo real**

En la pestaña **“Predefinido”**, los intervalos de tiempo ya vienen predefinidos pudiendo elegir el usuario entre los valores mostrados en la Figura 15.



**Figura 15 - Combo-box intervalos de tiempo y resolución**

En cambio en la pestaña **“Personalizado”** el usuario puede elegir la fecha de inicio y fin del intervalo que desea seleccionar para visualizar los datos así como la resolución con la que desea que se muestren. Para la pestaña **“Personalizado”** el usuario puede elegir entre las resoluciones mostradas en la Figura 16.



**Figura 16 - Combo-box resolución**

En caso de que el usuario no seleccione ningún pid antes de pulsar los botones de tiempo real o datos históricos, la aplicación mostrará un mensaje emergente informando al usuario que debe seleccionar algún punto, como se puede ver en la Figura 17.



Figura 17 - Mensaje emergente “seleccione puntos”

Por último, cuando el usuario accede a la pantalla de tiempo real tiene la opción de girar el dispositivo hasta la posición tumbada o “landscape” y poder visualizar una gráfica con los valores en tiempo real, mostrada en la Figura 18.

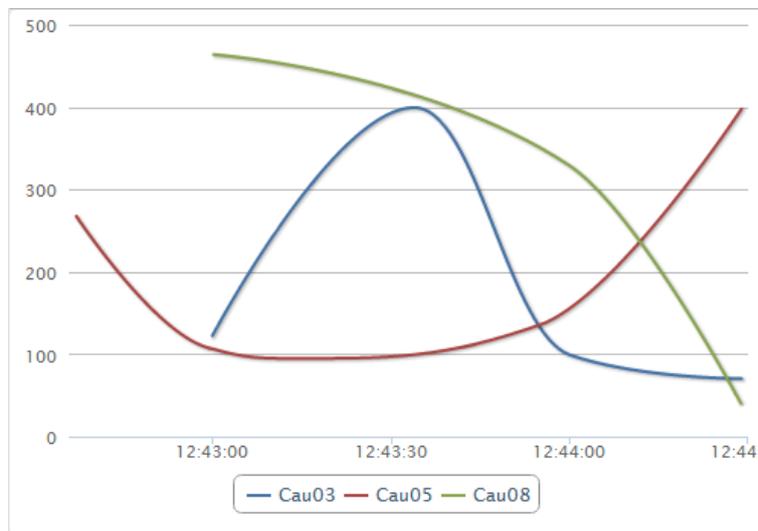


Figura 18 - Gráfica en tiempo real

#### 4.2.1 Desarrollo de la iteración

En esta segunda iteración las funcionalidades añadidas son las de consultar datos en tiempo real y datos históricos. Para implementar esta funcionalidad se crearon dos nuevas ventanas, una para cada opción.

Para la ventana de tiempo real se ha utilizado la clase **Panel** en la que se añade la lista de pids que el usuario ha seleccionado mostrando sus valores en tiempo real. Esta lista se genera utilizando la clase **Grid** a la que se agrega un objeto de la clase **Store**, que es el que recibe los

datos del servicio web y los almacena conforme a un modelo establecido con un objeto de la clase **Model**.

Para obtener los valores en tiempo real de cada pid se realiza una llamada a un servicio web que es el que devuelve los datos, utilizando el método **request** de la clase **Ajax**. Esta llamada se produce cuando el usuario pulsa el botón tiempo real (habiendo seleccionado previamente algún pid) generando un evento que es manejado por un controlador creado utilizando la clase **Controller** que se encarga del tiempo real. Dicho controlador crea la nueva ventana y solicita los datos al modelo, que los envía a la vista para mostrarlos por pantalla. Cuando se crea la nueva ventana de tiempo real y se muestra por pantalla el botón situado en la barra superior cambia y pasa a ser un botón que permite pausar o reanudar la ejecución de la consulta en tiempo real. Este botón se introdujo para que, en caso de que el usuario elija una gran cantidad de puntos no entrando todos en la pantalla, pueda pausar la ejecución y comprobar si los valores que devuelven los datos que no son visibles a simple vista son correctos. Si no se implementase esta funcionalidad, debido al refresco que se produce en la pantalla cada vez que se reciben nuevos datos, sería muy difícil comprobar los valores de los puntos que no se ven a simple vista en la pantalla. Mientras se está realizando la consulta de datos en tiempo real, el usuario tiene la posibilidad de girar el dispositivo colocándolo en posición tumbada o **"landscape"** con lo que se generara una nueva ventana en la que se mostrara una gráfica con los valores en tiempo real. Esta nueva ventana está diseñada utilizando la clase **Panel** y posteriormente añadiendo una gráfica generada utilizando la librería JavaScript **HighCharts**[Highcharts]. Los valores se van obteniendo en tiempo real a partir de llamadas a los servicios web y en esta ocasión en vez de volcarlos sobre la lista de puntos seleccionados se vuelcan sobre cada una de los puntos de la gráfica.

En cuanto a la ventana de históricos el funcionamiento es muy similar. Para implementarla se utiliza la clase **Panel** a la que se añade la lista de datos históricos obtenida del modelo utilizando para ello la clase **Grid**, obtenida a partir de un plugin de Sencha Touch ya que no incluye este componente de manera predeterminada. Para obtener los datos, el controlador asociado al botón de históricos genera una nueva clase y solicita al servicio web los datos en función de los pids seleccionados por el usuario y un intervalo por defecto de un año anterior al día actual. Los datos se almacenan utilizando un objeto de la clase **Store** que sigue el formato establecido por el objeto de la clase **Model**. Cuando la ventana de históricos se muestra por pantalla el botón de la barra superior se modifica para permitir al usuario configurar los parámetros en los que quiere realizar la consulta de datos históricos. Cuando el usuario pulsa este botón de configuración, el controlador lanza un evento que responde ha esa pulsación y crea una nueva ventana utilizando un **TabPanel** de dos pestañas. En la primera pestaña denominada **"Predefinido"** se añaden dos **Combo box**, en los que se permite al usuario seleccionar el pid del cual desean obtener los datos, así como la resolución con la que se obtendrán los mismos. Para la pestaña **"Personalizado"** se añaden un **Combo box** para elegir el pid del cual se desean consultar datos, dos **Datepicker** para establecer el intervalo de tiempo en el cual se desea realizar la petición y un último **Combo box** que permite seleccionar la resolución con la que se obtendrán los datos.

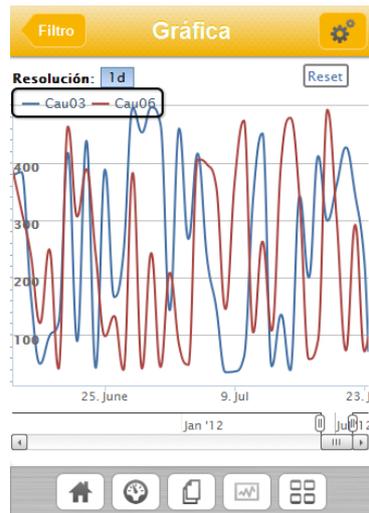
### 4.3 Iteración 3: Desarrollo de gráficas de tendencia

A continuación continuamos con el desarrollo de la iteración 3, agregando funcionalidad a la sección de análisis de la aplicación. Para poder visualizar gráficas de tendencia el usuario debe acceder a la sección de análisis desde el menú principal y posteriormente realizar un filtrado de datos, al igual que en las iteraciones anteriores. Una vez el usuario realice el filtrado de datos, tendrá la posibilidad de consultar gráficas de tendencia seleccionando uno o varios pids de la lista y posteriormente pulsando en el botón **Gráficas** situado en la barra de herramientas, como se puede apreciar en la Figura 19.



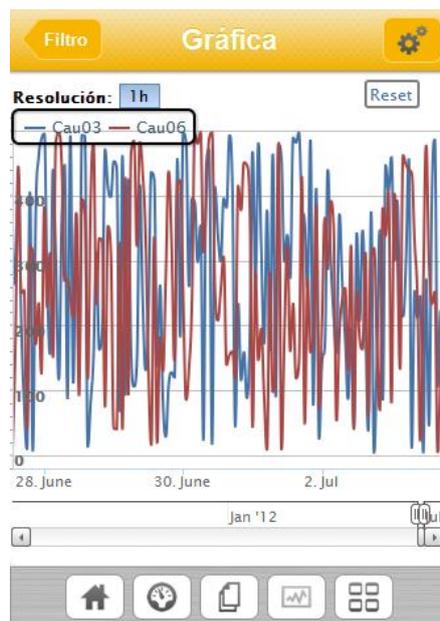
**Figura 19 - Botón gráficas de tendencia**

Una vez el usuario ha realizado el filtro puede seleccionar uno o varios pids de la lista y posteriormente pulsar en el botón graficas para obtener una gráfica de tendencia con una resolución inicial de 1 día ( 1 año), es decir, la gráfica muestra 1 valor por día durante el intervalo de un año desde el día actual. El resultado se puede ver en la Figura 20.



**Figura 20 - Pantalla gráficas de tendencia**

Una vez generada la gráfica, el usuario puede realizar zoom sobre la misma, lo que aumentará la resolución para mostrar datos en rangos más precisos. Cuando el usuario desee volver a la resolución inicial, puede pulsar el botón **“Reset”** situado en la parte superior derecha de la gráfica. En la Figura 21 se puede observar como la resolución varía al realizar zoom sobre la gráfica:



**Figura 21 - Aumento de resolución en gráficas de tendencia**

Esta pantalla de gráficas de tendencia también dispone de una opción de configuración que permite a los usuarios modificar de manera más rápida la resolución con la que se muestra la gráfica, sin necesidad de realizar zoom.

Dicha pantalla de configuración dispone de dos pestañas **“Predefinido”** y **“Personalizado”**, mostradas en la Figura 22, al igual que ocurría en la pantalla de configuración de datos

históricos. De hecho, esta pantalla ofrece la misma funcionalidad que la anterior únicamente adaptada en este caso a las gráficas de tendencia. De este modo, un usuario puede desde la pestaña “**Predefinido**” elegir la resolución que desea obtener en la gráfica mientras que en la pestaña “**Personalizado**” puede, además de elegir la resolución, seleccionar las fechas de inicio y fin para el intervalo en el que desea generar la gráfica de tendencia. Esto permite al usuario una gran libertad y usabilidad para poder elegir los intervalos que le interesan en cada momento.



Figura 22 - Pantalla configuración gráficas de tendencia

#### 4.3.1 Desarrollo de la iteración

En esta nueva iteración la funcionalidad añadida consiste en poder visualizar gráficas de tendencia a partir de datos históricos. Para ello se crea un controlador que lanza el evento cuando se pulsa el botón de gráficas de tendencia. Al lanzar este evento el controlador, generado utilizando la clase **Controller**, crea la nueva ventana de gráficas que consiste en un **Panel** en el que se añade la gráfica de tendencia. Los datos suministrados a la gráfica se obtienen de los datos históricos obtenidos del mismo servicio web que en la consulta de históricos. Una vez se obtienen los datos la vista muestra por pantalla la grafica

La grafica está diseñada utilizando **Highcharts**, que es una librería JavaScript que permite diseñar multitud de tipos de gráficas. Permite realizar zoom en la grafica, seleccionando con el dedo el intervalo que se desea ampliar. Al ampliar la grafica, también varía la resolución con la que se obtienen los datos obteniendo en cada ampliación datos más precisos. Cuando se pulsa el botón “**Reset**” de la gráfica, la aplicación realiza una petición al servicio web utilizando el método **request** de la clase Ajax y pasando como parámetros la dirección del servicio y la resolución inicial, esto es 1 año (1 día).

Para realizar la pantalla de configuración se utiliza un **TabPanel** en el que se establecen dos pestañas, una para la configuración predefinida y otra para la personalizada. En la pestaña “**Predefinido**” se implementa un **Radiofield** que permite elegir al usuario la resolución que quiere obtener en la gráfica. En cambio en la pestaña “**Personalizado**” se implementan dos **Datepicker** para seleccionar el intervalo en el cual se quieren obtener los datos y un **Combo**

**box** para elegir la resolución a la que se obtendrán los datos. Por último, en cada una de las dos ventanas se implementan los botones **Aceptar** y **Cancelar** utilizando la clase **Button**, cada uno con su método correspondiente en el controlador que realiza las operaciones correspondientes en función del botón que pulse el usuario.

#### 4.4 Iteración 4: Agregar puntos a agrupaciones temporales e interoperar con ellos

En la iteración 4 seguimos agregando funcionalidad a la sección Análisis, ya que es con diferencia la sección con más contenido de la aplicación y probablemente también la más importante. En esta ocasión vamos a agregar a la aplicación la posibilidad de agrupar puntos para poder manejarlos de manera más cómoda y eficiente.

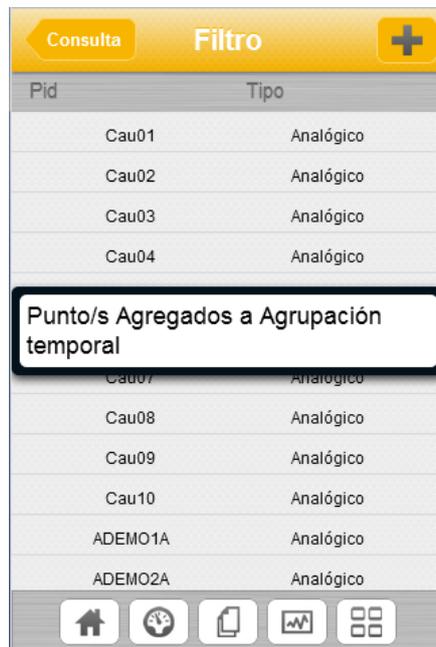
Mediante esta funcionalidad, un usuario puede seleccionar uno o varios puntos del listado obtenido al realizar el filtro y agregarlos a una agrupación temporal con la que trabajar de manera más sencilla. Esta agrupación temporal será eliminada cuando el usuario recargue por completo la aplicación.

Para agregar puntos a la agrupación el primer paso es acceder a la sección de Análisis y una vez allí realizar un filtro como se ha detallado anteriormente. Una vez realizado el filtro el usuario puede elegir los puntos que desea agregar a la agrupación y posteriormente pulsar en el botón situado en la parte superior derecha de la pantalla (Figura 23).



Figura 23 - Botón Agregar a agrupación temporal

En el momento de pulsar el botón agregar, la aplicación mostrará un mensaje emergente para advertir al usuario de que los puntos se han agregado correctamente a la agrupación temporal, mostrado en la Figura 24.



**Figura 24 - Mensaje emergente Agregar puntos agrupación temporal**

Una vez se han agregado puntos a la agrupación temporal el usuario puede acceder a dicha agrupación pulsando el botón **Agrupación temporal** situado en la barra de herramientas (Figura 25).



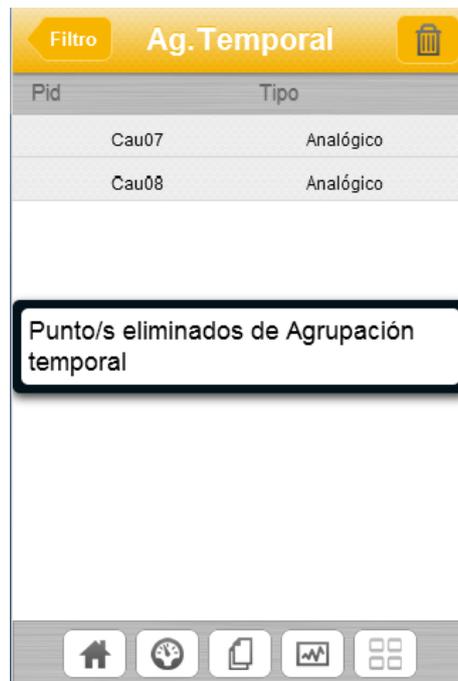
**Figura 25 - Botón Agrupación temporal**

Una vez dentro de la pantalla **Agrupación temporal**, el usuario puede realizar con los puntos agregados las mismas acciones que en la pantalla de filtro, pudiendo consultar datos en tiempo real, datos históricos o gráficas de tendencia.

Además el usuario dispone de la posibilidad de eliminar puntos de la agrupación temporal. Para ello el usuario debe seleccionar primero el/los puntos que desea eliminar y posteriormente pulsar el botón **Eliminar** situado en la parte superior derecha de la pantalla (Figura 26). Cuando el usuario realiza esta acción, la aplicación mostrará un mensaje emergente en la que informan que los puntos han sido eliminados y la lista de puntos de la agrupación es actualizada automáticamente (Figura 27).



**Figura 26 - Botón eliminar de agrupación temporal**



**Figura 27 - Mensaje emergente eliminar agrupación temporal**

Si el usuario pulsa el botón de **Agrupación temporal** sin haber agregado previamente ningún punto a dicha agrupación, la aplicación mostrará un mensaje emergente en el que informará al usuario que no existe ningún punto agregado a la agrupación (Figura 28).

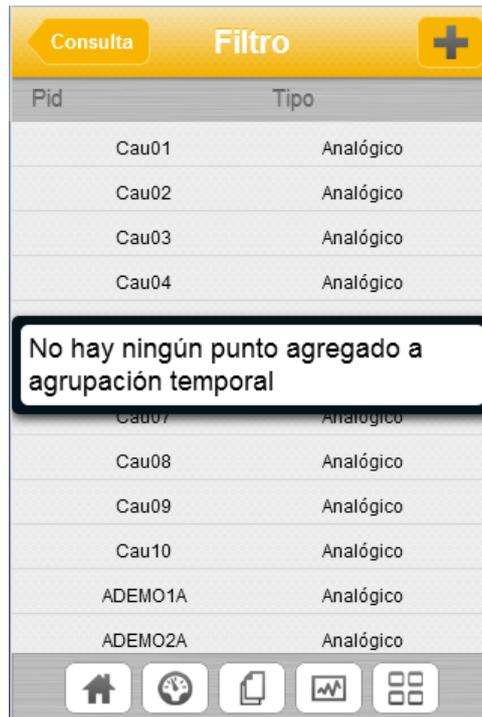


Figura 28 - Mensaje emergente no hay datos agrupación temporal

#### 4.4.1 Desarrollo de la iteración

Para el desarrollo de esta nueva iteración se utiliza una clase **Panel** en la que se añade una lista generada utilizando la clase **Grid** en la que se incluyen los pids seleccionados por el usuario, obtenidos mediante el método **getSelection()** de la clase **Grid** utilizada en la ventana de **Filtro** y que son añadidos a un nuevo objeto de la clase **Grid** que será la agrupación temporal. A su vez se crea un controlador utilizando la clase **Controller** que responda al evento provocado cuando el usuario selecciona algún pid de la lista filtrada y pulsa el botón agregar a agrupación temporal generado utilizando la clase **Button** y agregado a la barra de herramientas previamente generada en la iteración 1. También al añadir esta nueva pantalla a la pila **NavigationView**, se modifica el botón que aparece en la barra superior permitiendo al usuario eliminar los puntos que deseen de la agrupación. Para poder eliminar algún punto el usuario debe previamente seleccionarlos y pulsar el botón eliminar, lo que generara un evento que eliminará los elementos seleccionados de la lista de la agrupación temporal. Los mensajes emergentes se crean utilizando la clase **Panel** a la que se añade el texto que se quiere mostrar y se asocia al evento correspondiente.

## 4.5 Iteración 5: Desarrollo de Indicadores y Alarmas

Después de finalizar la iteración 4, la sección de análisis de la aplicación queda finalizada por lo que en la siguiente iteración comenzamos a desarrollar las secciones **Indicadores** y **Alarmas**.

Al acceder a la sección **Indicadores**, el usuario puede visualizar un panel con señales en tiempo real en las que en función del resultado obtenido las luces del panel se encienden de un color u otro. Mediante esta funcionalidad, el usuario puede visualizar rápidamente si alguna señal del sistema esta obteniendo un valor inadecuado o peligroso para el funcionamiento de la planta.

También mostrará información sobre diferentes datos de la planta, como en el caso del ejemplo la tabla de contadores o variables de entorno (Figura 29). Esta información podrá ser adaptada en función de las necesidades de cada cliente.

Inicio Indicadores		
Lecturas	CNA	CNT
Estado	<span style="color: red;">●</span>	<span style="color: red;">●</span>
Entorno	<span style="color: red;">●</span>	<span style="color: green;">●</span>
Produccion	<span style="color: green;">●</span>	<span style="color: red;">●</span>
Administracion	<span style="color: green;">●</span>	<span style="color: green;">●</span>
Contadores	CNA	CNT
Combustible	40%	70%
Dias Sin Parar	200	126
Dias Prox Parada	145	58
Variables entorno	CNA	CNT
Rio-Temp	9°C	7°C
Rio-Caudal	200	126
Viento-Velocidad	12 NE	2 SW
Clima-TempYHumed	24°C-30%	27°C-15%

**Figura 29 - Pantalla de indicadores**

Al pulsar sobre el panel de luces de la sección de indicadores, el usuario accederá a otra pantalla en la que podrá visualizar datos en tiempo real del estado de la planta. Con ello, el usuario puede obtener una visión mas detallada del funcionamiento del sistema (Figura 30).

Indicadores EstadoPlanta		
Lecturas	CNA	CNT
Potencia	418.09	411.45
Presión	153.75	189.38
Temperatura	31.22	173.54

**Figura 30 - Pantalla EstadoPlanta**

En las tablas de la parte inferior de la pantalla indicadores en las que aparecen los datos de contadores o variables de entorno el usuario podrá obtener gráficos de tendencia de dichos datos pulsando sobre las tablas, como se muestra en la Figura 31.

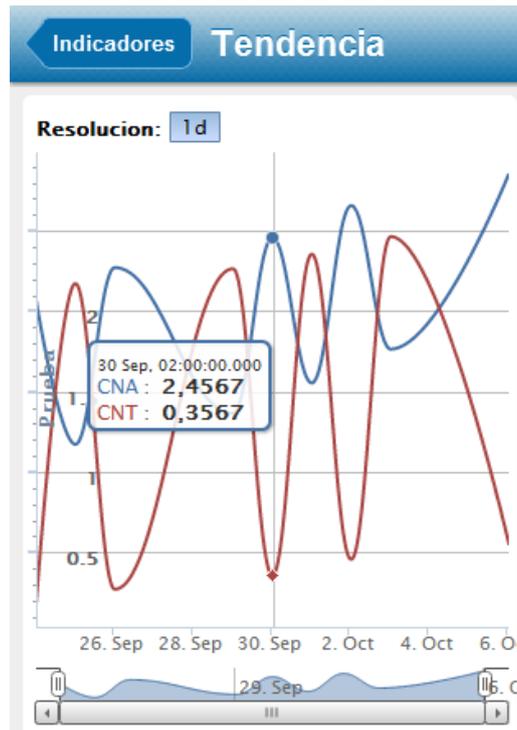


Figura 31 - Gráfica de tendencia indicadores

En cuanto a la sección de **Alarmas**, ésta incluirá una pantalla en la que el usuario podrá visualizar de manera rápida la cantidad de alarmas que se han producido en el sistema (Figura 32).

Inicio <b>Alarmas</b>		
PID	Tipo	Cantidad
Caudal cau01	Alarma Alta	253
Humedad hum01	Alarma Baja	50
Temperatura temp01	Alarma Crítica	80
Caudal cau02	Alarma Crítica	600
Humedad hum02	Alarma Crítica	90
Temperatura temp02	Alarma Alta	60
Caudal cau03	Alarma Baja	300
Temperatura temp03	Alarma Baja	25
Humedad hum03	Alarma Alta	70

Figura 32 - Alarmas del sistema

Al girar el dispositivo y colocarlo en posición tumbada o **“landscape”** se mostrara un gráfico de barras con las estadísticas de las alarmas producidas en el sistema (Figura 33).

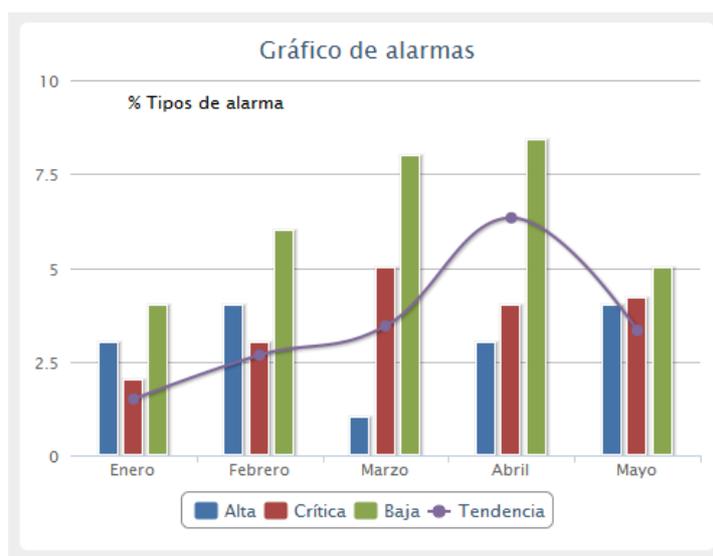


Figura 33 - Gráfico de alarmas

#### 4.5.1 Desarrollo de la iteración

En esta iteración se desarrollan dos de las secciones de la aplicación, **Indicadores** y **Alarmas**. Para realizar estas secciones se utiliza una clase **Panel** en la que se añaden los diversos componentes.

En la ventana indicadores se añade un **Grid** que muestra para cada señal una luz roja o verde en función del dato que recibe del servicio web y dos tablas con valores establecidos a modo de prueba para mostrar la funcionalidad que cada cliente puede establecer en esta sección. Las llamadas al servicio web se realizan de igual forma a la descrita en iteraciones anteriores, utilizando el método **request** de la clase **Ajax** y pasándole al método la dirección del servicio web y una serie de parámetros como la lista de pids. Las dos tablas también han sido diseñadas utilizando la clase **Grid** pero los datos que representan son datos a modo de prueba, establecidos directamente sin realizar ninguna llamada a servicio web.

En cuanto a la sección de **Alarmas**, a la clase **Panel** se le añade un **Grid** con el número de alarmas que se han producido en el sistema clasificadas por tipos. A este **Grid** se le pasan una serie de datos introducidos de manera directa a modo de prueba, al igual que en la sección de **indicadores**.

Al girar el dispositivo se añade una nueva ventana generada utilizando un **Panel** en la que se muestra un gráfico de barras generado con la ayuda de **HighCharts** en el que se muestran las estadísticas de alarmas producidas en el sistema.

## 4.6 Iteración 6: Desarrollo de gráficos Sinópticos

Por último, en la última iteración se desarrollara la sección de gráficos sinópticos. Esta sección permite al usuario visualizar una serie de gráficos sinópticos con señales en tiempo real del estado de la planta (Figura 34).

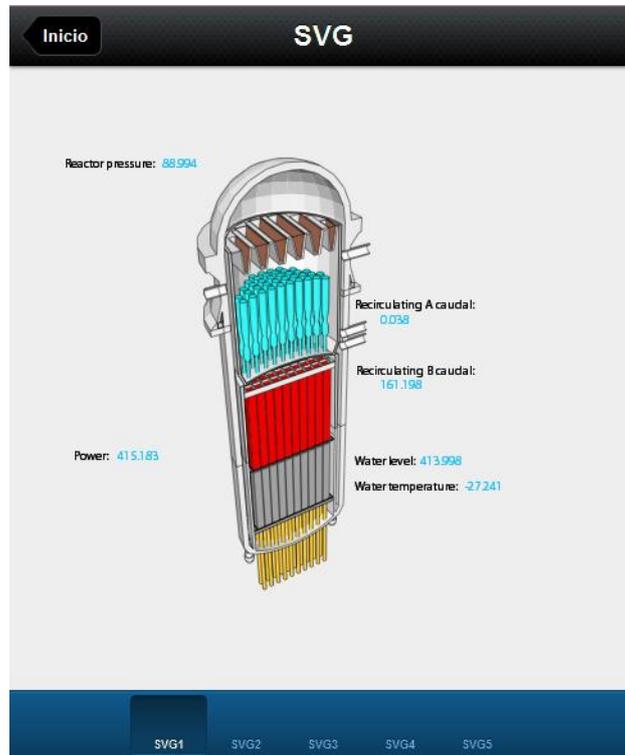


Figura 34 - Gráfico sinóptico

### 4.6.1 Desarrollo de la iteración

Para finalizar el desarrollo de las iteraciones nos queda realizar la sección de sinópticos. Para realizar esta sección se crea un **TabPanel** al que se le añade un grafico SVG en cada una de las pestañas. Los gráficos SVG ya habían sido desarrollados por la empresa para la aplicación web, por lo que el trabajo consistió en adaptarlos para que funcionaran en la versión móvil. Esta modificación fue realmente sencilla ya que prácticamente todo el trabajo realizado previamente para la versión web podía reutilizarse en la versión móvil.

En cada grafico SVG aparecen valores en tiempo real de las señales del sistema correspondientes. Para obtener estos valores se realiza una llamada al servicio web de datos en tiempo real utilizando el método **request** de la clase **Ajax** y pasándole por parámetro los pids que se desean ejecutar en tiempo real. Los datos se muestran por pantalla según se van obteniendo.

## 4.7 Sumario

En esta sección se ha descrito el desarrollo de la aplicación IDbox Mobile así como su funcionalidad. En cada iteración se han ido elaborando nuevas versiones del programa, añadiendo funcionalidad y consiguiendo cada vez un producto mas completo hasta alcanzar la versión final de la aplicación.



## 5 Evaluación y pruebas

Una vez finalizado el desarrollo completo de la aplicación se procede a realizar diversas pruebas en busca de posibles bugs que pueda tener la aplicación. Estas pruebas consisten en entregar la aplicación desarrollada a varios miembros del equipo de CIC ajenos al desarrollo de la aplicación y motivarles a encontrar posibles fallos en el sistema.

En las pruebas realizadas, los usuarios disponían en su teléfono personal de la dirección web para poder acceder a la aplicación e interactuar con ella. Gracias a estas pruebas algunas funcionalidades de la aplicación fueron modificadas, como por ejemplo el visor de gráficos sinópticos.

En un principio la ventana de gráficos sinópticos se diseñó utilizando un elemento de Sencha Touch denominado **Carousel** que permite realizar una lista de imágenes e ir pasando de una en una deslizando el dedo hacia un lateral de la pantalla. Esta solución fue desestimada una vez probada por los usuarios y visto que el gesto táctil era en ocasiones ignorado por la aplicación y también porque los tiempos de carga de las imágenes eran muy altos y el usuario debía esperar mucho tiempo entre la transición de imágenes, haciendo la aplicación poco usable. Por ello se decidió modificar dicho **Carousel** por el **TabPanel** actual, que permite realizar la misma función pero disminuyendo mucho esos tiempos de carga entre transiciones y además permitiendo acceder a cualquier imagen con una sola pulsación, mientras que en el **Carousel** debían pasarse las imágenes de una en una.

En general todos los empleados que realizaron las pruebas con la aplicación quedaron satisfechos con la funcionalidad y usabilidad de la aplicación, destacando la facilidad para poder acceder a las consultas de datos en tiempo real e históricos y también de las grandes expectativas que podría tener la sección **Indicadores**, ya que permite a cada cliente implementar las señales o alertas más críticas de sus sistemas para en una simple pulsación saber si todo funciona de manera correcta o no.

Realizadas estas pruebas, se resolvieron algunos bugs encontrados y se dio por completado el desarrollo de la aplicación.

### 5.1 Sumario

En esta sección se ha descrito como se han realizado las pruebas sobre la versión final de la aplicación en búsqueda de posibles fallos o mejoras en la aplicación. Una vez utilizada la aplicación por algunos miembros del equipo de CIC ajenos al desarrollo se modificaron algunas funcionalidades de la aplicación y se llegó a la versión final del producto.



## 6 Conclusiones y trabajos futuros

### 6.1 Conclusiones

El desarrollo de este proyecto tenía como objetivo desarrollar un nuevo cliente para enriquecer el producto *IDbox*. El objetivo principal se ha cumplido y la herramienta *IDbox* ya dispone de un cliente web, un cliente móvil y un cliente Excel.

En el desarrollo de este proyecto han surgido algunas dificultades, principalmente relacionados con el framework elegido para desarrollar el proyecto, Sencha Touch.

Sencha Touch es un producto bastante novedoso y que en el inicio de este proyecto su versión 2.0 se encontraba en fase beta. A lo largo del desarrollo, fueron publicándose nuevas versiones de Sencha Touch que arreglaban bugs y añadían funcionalidades hasta llegar a una versión definitiva, que es la versión utilizada en el desarrollo final.

Otra de las dificultades fue que, al ser una tecnología bastante novedosa, no existe una gran cantidad de bibliografía por lo que realizar algunas tareas necesarias para el desarrollo de la aplicación conllevó una dificultad añadida.

El motivo de elegir Sencha Touch como base para realizar el proyecto fue meramente una cuestión de la empresa, la cual estaba interesada en desarrollar dicha aplicación utilizando este framework. La alternativa a la utilización de Sencha Touch hubiera sido JQuery Mobile [Dav2011].

JQuery Mobile es un framework JavaScript similar a Sencha Touch que permite desarrollar aplicaciones web adaptadas a los dispositivos móviles. Una de sus principales diferencias respecto a Sencha Touch es que permite utilizar la aplicación en un mayor número de dispositivos, ya que también funciona de manera correcta en Windows Phone, WebOS o Symbian, entre otros

En el apartado personal, la realización de este proyecto me ha permitido introducirme en un ambiente laboral, observando a mi alrededor como se desarrollan proyectos software en el “mundo real”. Considero que ha sido una experiencia muy gratificante tanto a nivel personal como profesional y que me servirá en el futuro.

En cuanto al aspecto meramente profesional, la realización de este proyecto me ha permitido aprender a desarrollar aplicaciones para dispositivos móviles, que es un mercado que actualmente está en expansión, por lo que puede ser una buena oportunidad de cara a mi futuro laboral.

## 6.2 Trabajos futuros

Los trabajos futuros a desarrollar dependerán principalmente de las necesidades de cada cliente. Cada cliente podrá adaptar parte de las funcionalidades de la aplicación al ámbito de su empresa, obteniendo una aplicación personalizada.

Aun así, algunos de los trabajos futuros a realizar de manera general serian:

- Agregar la posibilidad de visualizar gráficos sinópticos en versiones de Android anteriores de Android 3.0 ya que dichas versiones no soportan la visualización de gráficos SVG de manera nativa.
- Implementar aplicaciones nativas para cada sistema operativo a partir de la versión web ya desarrollada utilizando el entorno PhoneGap, lo que permitirá aplicaciones más estables con configuraciones adaptadas específicamente al sistema donde van a ser ejecutadas.

PhoneGap es un framework de desarrollo de aplicaciones móviles que permite desarrollar aplicaciones nativas para varios tipos de sistemas operativos sin necesidad de utilizar los lenguajes de programación propios de cada sistema, permitiendo desarrollar aplicaciones para varias plataformas en paralelo. Debido a la fragmentación que existe en el mercado de dispositivos móviles, con diversos sistemas operativos como iOS, Android, Windows Phone, Blackberry OS, Bada, Meego, etc la utilización de frameworks que permitan realizar desarrollos de este tipo se convierte en casi una necesidad.[GP2012]

## 7 Bibliografía

- [IDbox] Documentación del producto IDbox <http://idbox.cic.es/documento-es.pdf> .2012.
- [Rod2012] Aquilino Rodríguez Penin."Sistemas SCADA". Marcombo. 2012.
- [Kum2011] Ajit Kumar. "Sencha Touch Cookbook". Packt Publishing.2011.
- [iOS] Documentación iOS <https://developer.apple.com/library/ios/navigation/> .2012.
- [Android] Documentación Android <http://developer.android.com/guide/components/index.html> .2012.
- [CB2012] John Clark, Bryan Johnson. "Sencha Touch Mobile JavaScript Framework". Packt Publishing.2012.
- [Eis2002] J.David Eisenberg. "SVG Essentials". 2002.
- [GIEP2012] Javier Garzás, Emanuel Irrazábal ,Juan Enríquez, Mario Piattini. "Gestión Ágil de Proyectos Software". Kybele Consulting, 2012.
- [FFB2004] Elisabeth Freeman, Eric Freeman, Bert Bates. "Head First Design Patterns". O'Reilly Media.2004
- [Fla2011] David Flanagan. "JavaScript: The Definitive Guide". O'Reilly Media.2011
- [Firt2012] Maximiliano Firtman. "jQuery Mobile. Aplicaciones HTML5 para móviles".ANAYA Multimedia. 2012.
- [BSS2011] Michael Bowers, Dionysios Synodinos, Victor Sumner. "Pro HTML5 and CSS3 Design Patterns". Apress.2011
- [RGAM2010] Nick Randolph, David Gardner, Chris Anderson y Michael Minutillo. "Professional Visual Studio 2010".John Wiley & Sons Ltd. 2010.
- [CCV2002] Patrick Cauldwell, Rajesh Chawla y Vivek Chopra. "Servicios Web XML". ANAYA Multimedia.2002.
- [Highcharts] Web de la librería Highcharts <http://api.highcharts.com/highstock> 2012
- [Dav2011] David Matthew. "HTML5 Mobile Websites: Turbocharging HTML5 with JQuery Mobile, Sencha Touch, and Other Frameworks". Focal Press.2011.
- [GP2012] Rohit Ghatol, Yogesh Patel. "Beginning PhoneGap: Mobile Web Framework for JavaScript and HTML5".Apress.2012.