

**PROTOTIPO DE SISTEMA INFORMÁTICO PARA LA CAPTURA DE  
INFORMACIÓN AMBIENTAL EN ESPACIOS CERRADOS Y MONITOREO EN  
TIEMPO REAL. CASO DE ESTUDIO FACULTAD DE INGENIERÍA DE  
SISTEMAS (UNAB).**

**CRISTIAN RENE CALDERÓN CALDERÓN**



**UNIVERSIDAD AUTONOMA DE BUCARAMANGA  
FACULTAD DE SISTEMAS  
TECNOLOGÍA Y SOCIEDAD  
BUCARAMANGA  
2015**

**PROTOTIPO DE SISTEMA INFORMÁTICO PARA LA CAPTURA DE  
INFORMACIÓN AMBIENTAL EN ESPACIOS CERRADOS Y MONITOREO EN  
TIEMPO REAL. CASO DE ESTUDIO FACULTAD DE INGENIERÍA DE  
SISTEMAS (UNAB).**

**CRISTIAN RENE CALDERÓN CALDERÓN**

**DIRECTORES:**

**RENÉ ALEJANDRO LOBO.**

**JUAN CARLOS GARCIA OJEDA.**

**PhD(c), MSc, BEng.**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA  
FACULTAD DE SISTEMAS  
TECNOLOGÍA Y SOCIEDAD  
BUCARAMANGA  
2015**

## Contenido

1. RESUMEN .....	10
2. INTRODUCCION.....	11
3. JUSTIFICACIÓN Y PROBLEMA.....	14
4. OBJETIVOS .....	17
Objetivo general:.....	17
Objetivos específicos: .....	17
5. ANTECEDENTES.....	18
6. ESTADO DEL ARTE .....	20
7. MARCO TEORICO .....	30
7.1 CONCEPTOS GENERALES .....	30
7.1.1 Diseño de Hardware .....	30
Hardware libre.....	30
Desventajas del Hardware Libre .....	33
Según su naturaleza.....	34
7.1.2 Diseño de Software.....	35
«Software libre» .....	35
7.2 Esquema general “Cliente Servidor” .....	37
7.2.1 Tipos de servidores.....	39
7.2.2 Tipos de Nubes .....	40
7.3 Internet de las cosas-Telefonía móvil.....	41
Telefonía móvil.....	41
Aplicativo móvil .....	41
Sistemas operativos para dispositivos móviles.....	46
API, Interface de Programación de Aplicaciones .....	47
7.4 Notificaciones push.....	49
Integración de la tecnología push en aplicaciones móviles.....	50
7.5 Lenguajes y estadísticas .....	55
7.5.1 Android .....	55

8.	DISEÑO METODOLOGICO .....	59
9.	RESULTADOS.....	62
9.1	CARACTERIZACIÓN DE LAS CONDICIONES ÓPTIMAS DE TEMPERATURA Y HUMEDAD EN INTERIORES .....	62
9.1.1	Temperatura .....	64
9.1.2	Transferencia de calor .....	65
9.1.3	Temperatura ambiente.....	65
9.1.4	Temperatura efectiva .....	66
9.1.5	Humedad.....	66
9.1.6	Zona confortable.....	67
9.1.7	Zona de conformidad térmica.....	67
9.2	ANALIZAR Y DISEÑAR UN SISTEMA INFORMÁTICO EN FUNCIÓN DE: REQUERIMIENTOS FUNCIONALES, MODELO DE BASE DE DATOS, ARQUITECTURA WEB, Y ARQUITECTURA MÓVIL (ANDROID). .....	69
9.2.1	Requerimientos del sistema .....	69
9.2.2	Arquitectura del sistema .....	70
9.2.3	Hardware del sistema iCrCThings. ....	72
9.2.5	Desarrollo de software. ....	75
9.2.6	Resultados y discusión del objetivo.....	82
9.3	IMPLEMENTAR EL PROTOTIPO DE SISTEMA INFORMÁTICO PARA LA TOMA DE INFORMACIÓN AMBIENTAL EN ESPACIO CERRADO Y REALIZAR PRUEBAS FUNCIONALES EN LA FIS. ....	84
9.3.1	Pruebas.....	84
9.3.2	Resultados y discusión del objetivo.....	89
10.	CRONOGRAMA DE ACTIVIDADES.....	97
11.	CONCLUSIONES .....	98
12.	TRABAJO FUTURO .....	100
13.	BIBLIOGRAFÍA.....	101

## LISTADO DE ILUSTRACIONES

pág.

Figura 1 : Arduino UNO .....	31
Figura 2 : Raspberry Pi.....	31
Figura 3 : Intel Galileo.....	32
Figura 4 : Dispositivos de entrada y salida conectados a servidor.....	37
Figura 5 : AirBot .....	42
Figura 6 : WaterBot.....	42
Figura 7 : Sensordrone .....	43
Figura 8 : Lapka Environmental Monitor .....	43
Figura 9 : Sensaris.....	44
Figura 10 : Air Quality Egg .....	44
Figura 11 : Electronic Nose Sensor.....	45
Figura 12 : PressureNet.....	45
Figura 13 : Broadcom Microchip.....	46
Figura 14 : iGeigie.....	46
Figura 15 : GCM Proceso General GCM .....	51
Figura 16 : ADF Mobile.....	54
Figura 17: Sistemas operativos Smartphone en todo el mundo .....	55
Figura 18 : Metodología modelo prototipo .....	59
Figura 19 : Metodología arquitectura prototipo.....	61
Figura 20: Zona de confort térmico.....	67
Figura 21 : Diagrama general del sistema de monitoreo.....	71
Figura 22 : Sketch iCrCThings.....	74
Figura 23 : Organización de la BD.....	76
Figura 24: Login y registro plataforma web.....	78
Figura 25 : Plataforma web con datos cargados.....	79
Figura 26 : Plataforma web con graficas cargadas.....	79
Figura 27 : Aplicación Móvil iCrCThings .....	81
Figura 28 : Graficas Móvil app iCrCThings .....	81
Figura 29 : Prototipo iCrCThings .....	83
Figura 30: Plano piso 2° Facultad Ingeniería Sistemas (UNAB).....	85
Figura 31 : Topología de red UNAB simulada.....	87
Figura 32 : Topología de red servidor –nodo.....	88

Figura 33 : Datos alojados en BD.....	90
Figura 34 : Temperatura nodos mañana-tarde.....	91
Figura 35 : Humedad relativa nodos mañana-tarde.....	92
Figura 36 : Temperatura y humedad relativa máxima nodos mañana.....	93
Figura 37 : Temperatura y humedad relativa máxima nodos tarde.....	94
Figura 38 : Zona confort térmico nodos mañana.....	95
Figura 39 : Zona confort térmico nodos tarde.....	96
Figura 40 : Diagrama de Gantt Cronograma actividades 2014-2015.....	97

## LISTADO DE TABLAS

pág.

Tabla 1 : Estado del arte.....	21
Tabla 2: Clasificación hardware libre.....	31
Tabla 3 : Clasificación hardware según su naturaleza. ....	34
Tabla 4 : Tipos de servidores.....	39
Tabla 5 : Tipos de Nubes.....	40
Tabla 6 : Aplicativos desarrollados con sensores ambientales.....	42
Tabla 7 : Sistemas operativos Smartphone en todo el mundo .....	56
Tabla 8 : Informe reportes 2015 EE.UU. Smartphone suscritos.....	56
Tabla 9 : Top de plataformas Smartphone EE.UU.....	57
Tabla 10 : Ecuación de balance térmico I.....	63
Tabla 11 : Ecuación de balance térmico II.....	63
Tabla 12 : Ecuación de balance térmico III.....	64
Tabla 13 : Parámetros DTH22 .....	73
Tabla 14 : Clasificación nodos según color .....	86
Tabla 15 : Resultado estabilidad Ping servidor-Nodo iCrCThings.....	86
Tabla 16 : Resultado temperatura y humedad relativa nodos.....	89

## 1. RESUMEN

En las organizaciones educativas en las cuales la mayor parte de los procesos (clases, tareas administrativas, actividades de investigación) se llevan a cabo en entornos cerrados, usualmente se utiliza una variada gama de mecanismos de medición con el fin de registrar variables ambientales. La mayoría de estos sistemas están constituidos por dispositivos o sensores, por ejemplo: un data logger, uno o varios sensores, y un software que en su conjunto permiten llevar a cabo la gestión de tales variables.

En este sentido, se propuso el desarrollo de un prototipo de sistema informático, único, automático y escalable que reconozca en forma simple, económica y robusta múltiples terminales y sensores comerciales cerrados que registren nivel de temperatura freático, humedad y temperatura del ambiente en tiempo real.

Con el desarrollo del proyecto se logró la integración del área de sistemas ambientales en un sistema de monitoreo, que permite supervisar continuamente las variables fundamentales de espacios indoor. Para ello se construyó un prototipo basado en el uso de una plataforma libre en hardware (Arduino) y software que utiliza los principios de Redes Inalámbricas de Sensores que permitan el monitoreo de condiciones, procesamiento de información y notificaciones de eventualidades en tiempo real.

El escenario de estudio para el desarrollo del proyecto y realización de las pruebas para la recolección de la información, fue la Facultad de Ingeniería de Sistemas de la Universidad Autónoma de Bucaramanga.

## 2. INTRODUCCION

Hoy por hoy la evolución, el desarrollo y la investigación por los nuevos protocolos de hardware, software y comunicación han llevado al despliegue de redes fiables que optimizan los recursos limitados (Procesamiento, Tamaño). La toma de decisiones correctas conlleva a la captura exponencial de datos generados por sensores, siendo los datos una manera eficiente de demostrar que tan satisfactorio es un experimento

Tecnologías convencionales muy pocas veces podrán lidiar con la gran cantidad de datos generados por WSN (por sus siglas en Inglés, *Wireless Sensor Networks*) capturadas. Es así como las nuevas herramientas surgen para procesar y almacenar grandes volúmenes de datos (Big Data).

Los nuevos conceptos MySQL y de almacenamiento evolucionan cuando los datos en tiempo real surgen. La nueva tecnología IoT pretende llevar a su máximo y desafiar soluciones de datos (Big) y WSN sin coexistir.

Las redes *Ad Hoc* no requieren infraestructuras existentes. A diferencia de los de WLAN o redes celulares. El cerebro de cada nodo WSN es el microcontrolador que procesa las lecturas de sus propios sensores y, en algunos casos, las lecturas de nodos adyacentes, así que los sensores en diferentes nodos situados cerca uno del otro pueden ser altamente correlacionados y por lo tanto la cantidad de datos de los sensores puede ser transmitida desde los dos nodos de sensores y así reducir de manera cooperativa la transmisión.

Abordando estos conceptos y comprendiendo su funcionamiento, el objetivo de este trabajo fue construir un prototipo con una plataforma libre en hardware y software integrando WSN que monitoree condiciones, procese información y notifique en tiempo real eventualidades.

Es por eso que se construyó un prototipo, utilizando tecnología libre y escalable (Arduino) que respondiera a esta necesidad. El prototipo fue implementado en la universidad autónoma de Bucaramanga (UNAB), facultad de ingeniería de sistemas. El sistema captura, almacena y procesa información todo esto en tiempo real.

En la construcción del prototipo se utilizó el Arduino WiFi Shield que permite conectarse a internet utilizando una especificación inalámbrica 802.11 (Wi-Fi). Se basa en el estándar 802.11b/g Sistema HDG204 LAN inalámbrica en paquete. Un AT32UC3 que proporciona una red IP que soporta los protocolos de transporte TCP y UDP. Este módulo está integrado a una placa Arduino Mega 2560 microcontroller ATmega2560 que cuenta con 54 pines digitales de entrada/salida, 16 entradas analógicas 4 UARTs (hardware puertos serie), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un header ICSP y un botón de reinicio.

El Atmega2560 tiene 256 KB de memoria flash para almacenar código (de los cuales 8 KB se utiliza para el cargador de arranque), 8 KB de SRAM y 4 KB de EEPROM (que puede ser leído y escrito con la biblioteca EEPROM). Se usó esta tecnología debido a su baja potencia, bajo costo y batería de larga duración, así como la facilidad de programación en C del microcontrolador. Las placas Arduino constan de bibliotecas para la conexión de sus respectivas Shields y a su vez la interacción de entradas y salidas tanto analógicas como digitales.

Para los sensores de monitoreo de temperatura y humedad se utilizaron sensores analógicos y digitales de bajo costo para demostrar la prueba de concepto. Se utilizaron librerías que facilitan la interconexión entre placas Arduino y dispositivos móviles (Temboo, Parse).

Se diseñó un aplicativo móvil y Web que lleva el control de todos los datos recolectados por los sensores y su almacenamiento en la base de datos. Las gráficas muestran registros históricos de todas las eventualidades capturadas en el caso de estudio. Las Push Notifications alertan al usuario de las irregularidades gracias al uso de librería Parse.

El documento está organizado de la siguiente manera. Justificación y planteamiento de problema, Objetivos, antecedentes, Estado del arte donde se discuten algunos trabajos relacionados, marco teórico donde desglosa todo lo anteriormente mencionado, Metodología a seguir en la construcción de proyecto.

En las siguientes secciones se describen el entorno de prueba y los resultados. Finalmente concluimos el documento y se discuten algunas extensiones futuras de nuestro prototipo actual.

### 3. JUSTIFICACIÓN Y PROBLEMA

En todo ámbito comercial, empresas, compañías y educación. Donde se encuentren entornos cerrados, de los cuales se deba registrar parámetros ambientales, con periodicidad se utiliza una variada gama de mecanismos de medición con el fin de registrar el comportamiento estacional y ambiental de diferentes variables. Por tal motivo, es común emplazar en cada uno estos sitios equipamiento de diferentes gamas industriales, con los cuales se recopila una serie de inconstantes ambientales (Temperatura-Humedad) con el fin de vigilar dichos parámetros dentro de estos medios cerrados. La mayoría de estos mecanismos están constituidos por dispositivos o sensores como lo son: un data-logger, uno o varios sensores, y un software, los cuales permiten gestionarlo, es decir, configurar, iniciar, moderar y reinicializar. Dichos sensores, se hallan en contacto continuo con el ambiente, y constan de transductores que cambian la vacilación de la inconstante física en diferenciaciones automáticas que son detectadas y acumuladas por el data-logger en una memoria interna.

El carácter de trabajo de este tipo de equipos es autónomo, es decir, una vez inicializados no demandan intervención manual hasta la inmediata descarga de datos, reconfiguración e inicialización. Uno de los problemas que se abordó en este trabajo es en la alineación de los equipos y en la descarga de los datos anteriormente almacenados. Comúnmente, ante la falta de equipos de transferencia de datos en tiempo real, estas tareas ineludiblemente se ejecutan in-situ, manejando un software diferente y especial de tipo propietario, implementado en cada data-logger.

Por lo tanto, hay industrias que brindan estos módulos de transferencia remota ajustados solamente a algunos modelos de sus equipos. Sin embargo éstos

tienen un alto costo, y si bien es una elección válida la de descarga de datos todavía debe ser ejecutada en forma manual.

Es significativo subrayar que mientras haya un único dispositivo que no posea funcionalidades automáticas y remotas, la descarga manual es ineludible, más aún, resultaría costoso tener un modelo de transmisión/recepción inalámbrica por cada equipo ubicado en un punto de monitoreo.

Dado lo anterior, el proceso para conseguir los datos rastreados radica en visitar diariamente estos puntos de análisis y procesarlos mediante el software que provee el fabricante para manipular el equipo, suprimir su memoria, alinear parámetros explícitos y posteriormente reinicializarlo, retornando así a su modo autónomo. Todo lo anterior crea un inconveniente extra al instante de procesar y manejar la información obtenida.

Organizar estos datos para su posterior procesamiento no es una labor insignificante, día a día por cada punto de vigilancia se crean cientos de datos. De modo que, manipular esta información, ya sea para nutrir modelos ambientales, o para otro fin, demanda un pre-procesamiento de los datos, debido a la falta de conformación común entre los dispositivos.

Para solucionar estos problemas, y con el fin de optimizar el proceso de recolección, traspaso y centralización de los datos explorados, en este trabajo se han de utilizar tecnologías inalámbricas, equipos de administración de redes y sistemas que proveen una gran ayuda y experimentada confiabilidad.

Asimismo, serán combinados con un lenguaje de categorización de alto nivel viabilizando computarizar e integrar la descarga de datos a través de un prototipo de sistema informático, dando como resultado un conjunto de monitoreo y recolección de datos ambientales automático.

El principal objetivo de este trabajo es diseñar e implementar un prototipo de sistema informático, único, automático y escalable que reconozca en forma simple, económica y robusta múltiples terminales y sensores comerciales cerrados que registren nivel de temperatura freático, humedad y temperatura del ambiente, etc. Todo esto en tiempo real.

Con la solución del proyecto se busca integrar el área de sistemas ambientales en un sistema de monitoreo, que permita supervisar continuamente las variables fundamentales para el buen funcionamiento. Construir un prototipo con una plataforma libre en hardware y software integrando WSN que monitoree condiciones, procese información y notifique en tiempo real eventualidades.

## 4. OBJETIVOS

### **Objetivo general:**

- Diseñar e implementar prototipo de sistema informático para la captura de información ambiental en espacios cerrados y monitoreo en tiempo real. Caso de estudio FIS.

### **Objetivos específicos:**

- Caracterizar condiciones de temperatura y humedad en espacios cerrados. Caso de estudio FIS.
- Analizar y diseñar un sistema informático en función de: requerimientos funcionales, modelo de base de datos, arquitectura web, y arquitectura móvil (Android).
- Implementar el prototipo de sistema informático para la toma de información ambiental en espacio cerrado y realizar pruebas funcionales en la FIS.

## 5. ANTECEDENTES

En el año 2005, los estudiantes *Derly Carolina Vera Durán* y *Josue Barrios Rodríguez* de la facultad de ingeniería de sistemas de la Universidad Autónoma de Bucaramanga desarrollaron el proyecto **sistema de monitoreo en servidores con generación de reportes a telefonía inalámbrica y web**. (Vera Duran & Barrios, 2005). En este proyecto se planteó la idea de implementar un sistema informático con la capacidad de monitorear servidores y verificar su continuo funcionamiento. Este proyecto brindo a los administradores de red una aplicación que les permitía tener acceso a la información de los servidores sin necesidad de trasladarse al lugar del suceso o falla. Además, el software desarrollado ofrecía la posibilidad de enviar comandos remotos para recuperar el servicio de algún servidor expuesto a alguna falla.

En el año 2006, los estudiantes *Cesar Augusto López Cortes*, *Luis Carlos Rodríguez* y *Ricardo Bustos Barba* de la facultad de Ingenierías Fisicomecánicas programa de ingeniería mecatrónica desarrollaron el proyecto **sistema de monitoreo mediante cámaras inalámbricas para vigilancia**. (Lopez Cortes, Rodriguez, & Barba, 2006). En este proyecto se planteó un hardware controlado por un sistema informático capas de supervisar y vigilar un área de trabajo. Brindando a las personas encargadas de la seguridad y administradores tener vigilado un área de trabajo en específico.

Finalmente en el año 2008 los estudiantes *Daniel Forero Vargas* y *Rafael Andrés Jaraba García* de la facultad de ingeniería de sistemas de la Universidad Autónoma de Bucaramanga desarrollaron el proyecto **Validación de una red de sensores inalámbricos para el control de la calidad del aire en la zona industrial de Bucaramanga**. (Forero Vargas & Jaraba Garcia, 2008). Este proyecto busco automatizar la medición y control de la concentración de monóxido de carbono de las emisiones atmosféricas de la ciudad de Bucaramanga. Este

sistema permitía capturar los datos obtenidos y administrarlos luego en una interfaz amigable y así lograr optimizar el método actual de la regulación de la calidad del aire.

## 6. ESTADO DEL ARTE

En la siguiente sección del documento se hace referencia a todas aquellas innovaciones que son parte fundamental del proceso de construcción del prototipo. Comenzando con una breve explicación sobre el proceso que ha tenido el hombre para alcanzar la perfección de monitorear condiciones físicas.

El hombre ha tenido la necesidad sobre el control de su medio y todas las variables que lo componen, solo hasta el siglo XVII se dieron a conocer sistemas de regulación de temperatura entre ellos los aplicados en el horno y la incubadora de Drebbel<sup>1</sup> (El regulador de temperatura de Drebbel). El principio utilizado en la regulación de temperatura fue el siguiente, si la temperatura del horno subía se dilataba el contenido de un depósito de alcohol de tal forma que se desplazaba un juego de palancas que abría un orificio de salida de gases.

Así mismo la ejecución y desarrollo para el monitoreo de temperaturas en diferentes ambientes es parte fundamental de cualquier proceso de seguridad y normatividad de cuartos fríos y espacios cerrados. Reduciendo riesgos y problemas al medio ambiente.

Entre las novedades se encuentran dispositivos capaces de regular las temperaturas.

- Numerosas opciones de alarma (SMS, e-mail, LED) en caso de violación de los valores límite o de interrupción del suministro eléctrico.
- Completo sistema de gestión de informes.
- Gran variedad de sondas para mediciones desde -200 °C hasta más de +1000 °C.

---

<sup>1</sup> Cornelius Drebbel fue un destacado inventor de origen holandés quien creó una incubadora con un termostato de mercurio que le permitía mantener constante la temperatura.

- Dispositivos WSN que procesan información y sobrescriben en plataformas (cloud).

Es importante destacar que se debe seguir generando ideas para el mejoramiento de la calidad de vida y fortalecer la conservación del medio ambiente.

A continuación se evidencia el estado del arte detallado de las investigaciones e innovaciones, ejemplos o aplicativos de monitoreo afines al caso de interés tabla 1.

**Tabla 1 : Estado del arte**

<b>Palabras clave</b>	Internet of Things, Cloud computing,RFID,Wireless sensor networks,API key,Environmental parameters monitoring,Big Data,Arduino		
<b>Base de datos utilizadas</b>	Google Scholar,IEEE Xplorer		
<b>Total referencias recuperadas</b>	Google Scholar : 8	IEEE Xplorer : 6	Empresas: 3
<b>Fecha de búsqueda</b>	Google Scholar: Enero 25, 2015,IEEE Xplorer:Febrero 16,2015		
<b>Criterios para incluir y excluir referencias</b>	Rango de fecha de búsqueda: Entre los años 2002-2015. Pertinencia con el tema de investigación: <ul style="list-style-type: none"> <li>• Sistemas de monitoreo de condiciones ambientales.</li> <li>• Técnicas y tecnologías que permitan conocer el IoT.</li> <li>• Aspectos relacionados hardware libre (Arduino) monitoreo.</li> </ul>		



Autor/Autores	AÑO	TITULO	PROBLEMA	SOLUCION	TIPO DE DOCUMENTO
(Lita, Electron., Cioc, Visan, & Dorobantu, 2009)	2009	Technologies for wireless data acquisition and monitoring of environmental parameters	Correcta selección del método de transmisión de radio inalámbrico.	Implementación de sistemas distribuidos, módulos GM862 GSM / GPRS	Paper experimental
(Rossi, Departamento de Electr. Eng., Krishnamachari, & Kuo, 2004)	2004	Hybrid data and decision fusion techniques for model-based data gathering in wireless sensor networks [environmental monitoring applications]	Recolección de datos en las redes inalámbricas de sensores para la vigilancia del medio ambiente	Estimaciones de los parámetros del modelo. En lugar de transmitir las mediciones del sensor primas.	Sección libro Paper experimental
Xiaolei Zhang and Xianzhen Ren	2014	Wireless data transmission between iOS client and web server	Transmisión de datos mediante un Web Service entre iOS y un servidor	Creación de bases datos en MySQL en un servidor basado en Linux y hacer consultas de esta base por medio de un Web Service	Exploratorio

Sheng-Po Kuo ; Hsiao-Ju Kuo ; Yu-Chee Tseng	2009	The Beacon Movement Detection Problem in Wireless Sensor Networks for Localization Applications	La localización como tema crítico en las redes de sensores inalámbricos	Mejorar la precisión de los sistemas de localización en caso de que haya movimientos inadvertidos de algunas balizas	Paper experimental
Mare Srbinovska,Cvetan Gavrovski ,Vladimir Dimcev	2015	Environmental parameters monitoring in precision agriculture using wireless sensor networks	Las reformas en la política de la agricultura, a inpuesto la implementación de cultivos innovadores respetuosos al medio ambiente	una arquitectura de red inalámbrica de sensores para el invernadero de hortalizas con el fin de lograr el cultivo científico y menores costes de gestión desde el aspecto de la	Paper experimental
				vigilancia del medio ambiente	
Jayavardhana Gubbia,	2013	Future Generation	El futuro del internet de las	Conceptos e introducción (IoT),	Paper

Rajkumar BuyyabSlaven Marusica, Marimuthu Palaniswamia		Computer Systems	cosas. Convergencia de WSN. Interacción de nubes públicas y privadas.	Computación Ubicua con la perfección de objetos cotidianos.	experimental
Joshua A. Tauber,.	2002	Indoor Location Systems for Pervasive Computing	La mayoría de los sistemas de localización se desarrollan en torno a tecnologías específicas , y por lo tanto se restringido por las limitaciones de la tecnología utilizada	Técnicas y tecnologías para estimar la localización de la computación ubicua en interiores	Estado del Arte
Binghao Li., James	2006	Indoor Positioning	La ausencia de conocimiento en	Se describen los diferentes sistemas	Paper

Salter., Andrew G. Dempster., Chris Rizos..		Techniques Based on Wireless LAN	tecnologías Indoor.	de posicionamiento basados en Wi-Fi mencionando sus ventajas y desventajas.	Experimental
Dliruv Paiidya., Ravi Jain., Einil Lupu.,	2003	Indoor location estimation using multiple wireless technologies	En gran mayoría de los sistemas de localización se desarrollan en torno a tecnologías específicas , y por lo tanto se	Estudio de la viabilidad del uso de múltiples tecnologías inalámbricas de forma simultánea para la estimación de ubicación	Paper Experimental
			restringido por las limitaciones de la tecnología utilizada		
Vijaykumar Thakor,Prof.Chirag Bhatt, Prof. Mehul Patel	2015	Precision Agriculture Prospects using Wireless Sensor Network	Mejor la eficiencia de las prácticas agrícolas. Datos en tiempo real, sensores inalámbricos, control automatizado de riego.	Decisiones más inteligentes. En la toma de datos, mayor eficiencia de recortar y aumentar el plan de producción.	Paper Experimental

<p>Aslam, M.S.  Nimbus Centre For  Embedded Syst. Res.,  Cork Inst. of Technol.,  Cork, Ireland  Rea, S. ; Pesch, D.</p>	2012	<p>Service Provisioning for  the WSN Cloud</p>	<p>Crecimiento de la  infraestructura actual.  Impulsados por el concepto de  "Smart Cities"</p>	<p>Necesidad de una red Wireless  Sensor reutilizable.</p>	<p>Paper  Exploratorio</p>
<p>Moreira, N.  Venda, M. ; Silva, C. ;  Marcelino, L. ;  Pereira, A.</p>	2011	<p>Sensor - Mobile  application to monitor a  WSN</p>	<p>WSNs están surgiendo como un  nuevo paradigma, la  recopilación de información de  forma colaborativa, donde los  sensores recopilan procesan y  comparten información.</p>	<p>Desarrollo de una aplicación móvil  para controlar remotamente una  WSN.</p>	<p>Paper de  Resultados</p>
<p>Baghyalakshmi, D.</p>	2011	<p>WSN based temperature</p>	<p>Toma de decisiones en la</p>	<p>Vigilar continuamente la temperatura</p>	<p>Paper</p>

Ebenezer, J. Satyamurty, SAV.		monitoring for High Performance Computing cluster	implementacion de los WSN	del nodo, almacenando datos y transmitiendo solo cuando se solicita.	Experimental
Vongsagon Boonsawat, Jurarat Ekchamanonta, Kulwadee Bumrunghet, and Somsak Kittipiyakul	2010	XBee Wireless Sensor Networks for Temperature Monitoring	Reducir el costo y consumo de la energía mediante WSN	Implementación de un microcontrolador y un Arduino XBEE basado en el estándar 802.15.4	Paper Experimental
(SOLTRACK, 2012)	2012	Refrigerados-Soltrack	la repartición de mercancías por medio de transporte refrigerado representa entonces un desafío importante en término de logística	Monitorear constantemente la temperatura de cuartos fríos, cajas refrigeradas y vehículos de carga. Determinando así el comportamiento relevante de la temperatura.	Empresa

(measureinstruments, 2001-2014)	2014	Termometria de silos- Monitoreo de temperatura de granos en silo o celdas	Investigación y desarrollo de sensores, medidores, controladores para la medición de humedad y temperatura de materiales.	Lectura continúa de humedad y temperatura de suelos y tierra en 2 profundidades, a campo o invernaderos.	Empresa
(Industry, 2014)	2014	Aplicaciones- Alimentarias,Labguard	Sistema de monitoreo inalámbrico de parámetros de temperatura y ambiente.	Solución completa de monitoreo y trazabilidad, incluyendo software, hardware y servicios garantizando fluida y efectiva operación.	Empresa

Fuente: (Calderon Calderon , Estado del arte, 2015)

## 7. MARCO TEORICO

### 7.1 CONCEPTOS GENERALES

#### 7.1.1 Diseño de Hardware

##### **Hardware libre**

Se denomina hardware libre, electrónico libre o máquinas libres a aquellos dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita. La filosofía del software libre es aplicable a la del hardware libre y por eso forma parte de la cultura libre. Algo que tiene en común el hardware con el software es que ambos corresponden a las partes tangibles de un sistema informático. Un ejemplo de hardware libre es la arquitectura UltraSparc<sup>2</sup> cuyas especificaciones están disponibles bajo una licencia libre.

A continuación en la tabla 2 se muestra un breve resumen de tecnologías de hardware libre más comunes.

---

<sup>2</sup> Gama de procesadores creado por Sun Microsystems, a partir del diseño RISC. El modelo de la imagen es un Sun Enterprise 250 (2 procesadores UltraSPARC) del CCH Naucalpan -UNAM-. Los últimos modelos de UltraSparc es el UltraSparc IV.

**Tabla 2: Clasificación hardware libre**

Hardware Libre	
Arduino	Descripción
<b>Figura 1 : Arduino UNO</b>	<p>Plataforma de hardware libre basada en una placa de entradas y salidas simple. Posee entorno de lenguaje de programación Processing / Wiring.</p> <p>Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo puede descargarse gratuitamente.</p> <p>Las plataformas Arduino están basadas en los microcontroladores Atmega168, Atmega328, Atmega1280, ATmega8 y otros similares, chips sencillos y de bajo coste que permiten el desarrollo de múltiples diseños. (ARDUINO)</p>
	
Fuente: (ARDUINO)	
Hardware Libre	
Raspberry Pi	Descripción
<b>Figura 2 : Raspberry Pi</b>	<p>Es una placa computadora (SBC) de bajo costo desarrollada en el Reino Unido por la Fundación Raspberry Pi.</p> <p>Incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz (el firmware incluye unos modos Turbo para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía), un procesador gráfico (GPU)</p>
	

	VideoCore IV, y 512 MB de memoria RAM aunque originalmente al ser lanzado eran 256 MB.
Fuente: (raspberrypi.org)	
	Dimensiones de 85,6 x 53,98 x 17 mm, posibilidad de mostrar video 1080p o conectarse a redes y a internet y administrar dispositivos de domótica.
<b>Intel galileo</b>	
<b>Figura 3 : Intel Galileo</b>	Placa electrónica basada en el Quark Intel ® SoC procesador X1000 de aplicaciones, un sistema de 32 bits Intel Pentium en un chip. Una ranura de tamaño completo mini-PCI Express, puerto Ethernet de 100 Mb, ranura Micro-SD, RS-232 puerto serial, puerto Host USB, puerto USB Cliente y 8MByte flash NOR vienen de serie en el tablero. (ARDUINO)
	
Fuente: (Arduino)	Compatible con el Arduino Software Development Environment (IDE), cuenta con una ranura de tamaño completo mini-PCI Express, puerto Ethernet de 100 Mb, ranura Micro-SD, RS-232 puerto serial, puerto Host USB, puerto USB Cliente y 8MByte flash NOR .

Fuente: (Calderon Calderon , Estado del arte, 2015)

## **Ventajas del Hardware Libre**

- Independencia tecnológica, permite a las naciones no depender de ninguna otra que le provea los recursos necesarios para su desarrollo.
- Favorece la calidad del hardware, a los estándares abiertos y que sean más económicos.
- El trabajo colaborativo sobre los diseños permite la reutilización y la adaptación de los mismos.
- Disminuye los costos y tiempos de diseño en sus trabajos.
- Se libera a los productores de los propietarios de alianzas globales.

## **Desventajas del Hardware Libre**

Las cuatro libertades del Software libre no se pueden aplicar directamente en el Hardware, naciendo estas dificultades:

- Los diseños son específicos y únicos, dificultando su reproducción.
- La reproducción implica primero la prueba de los componentes, para luego fabricar el dispositivo, lo que eleva los costos.
- La disponibilidad de los chips, haciéndose dificultoso para muchos países en vías de desarrollo.
- La fabricación de hardware lleva implícito infraestructura de diseño, simulación, producción e implementación, al contrario de lo que se da en el software libre. (EcuRed.cu, 2014)

## Según su naturaleza

Dada su diferente naturaleza, al hablar de hardware abierto o libre hay que especificar de qué tipo de hardware se está hablando. A continuación se describen cada uno del diferente hardware según su naturaleza tabla 3:

**Tabla 3 : Clasificación hardware según su naturaleza.**

<b>Hardware reconfigurable</b>	<b>Hardware estático</b>
Esta descrito mediante un lenguaje de descripción de hardware. Naturaleza completamente diferente al hardware estático. Tiene una manera muy similar a como se realiza el software, mediante archivos planos de texto, que contienen su código fuente.	Reúne un conjunto de materiales tangibles de muchos sistemas electrónicos.
	Tiene una existencia física (Tangible).
	Problemas: Diseño físico único, tiene asociado un coste si se desea realizar cualquier modificación, disponibilidad de los componentes a la hora de fabricar un
Licenciamiento libre ejemplo GPL.	
Problemas con las herramientas de desarrollo	

necesarias para programar estos dispositivos.	componente se dificulta en ocasiones. (González, González, & Gómez, 2003)
Para hacer que el hardware reconfigurable sea libre basta con aplicar la licencia GPL al código escrito.	Las patentes en el mundo del desarrollo del hardware son una problemática al momento de desarrollar cualquier dispositivo.
	No toda persona está en la capacidad de diseñar hardware debido a la infraestructura de desarrollo que este demanda.

Fuente: (Calderon Calderon , Estado del arte, 2015)

### 7.1.2 Diseño de Software

#### ¿Qué es el software libre?

«**Software libre**» es el software que respeta la libertad de los usuarios y la comunidad. En grandes líneas, significa que los usuarios tienen la libertad para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, el

«software libre» es una cuestión de libertad, no de precio. Para entender el concepto, piense en «libre» como en «libre expresión», no como en «barra libre».

Promovemos estas libertades porque todos merecen tenerlas. Con estas libertades, los usuarios (tanto individualmente como en forma colectiva) controlan el programa y lo que este hace. Cuando los usuarios no controlan el programa, decimos que dicho programa «no es libre», o que es «privativo». Un programa que no es libre controla a los usuarios, y el programador controla el programa, con lo cual el programa resulta ser un instrumento de poder injusto.

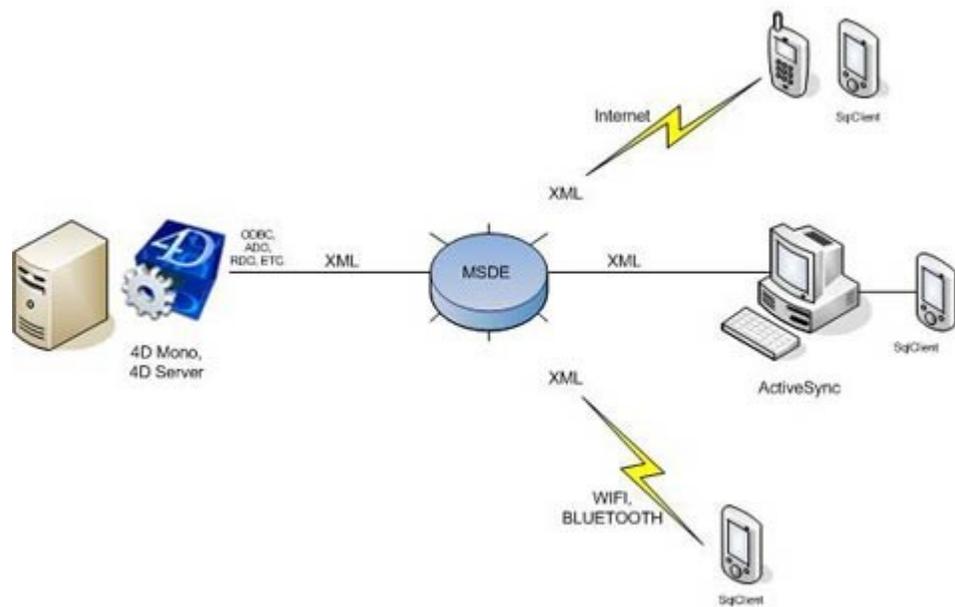
Un programa es software libre si los usuarios tienen las cuatro libertades esenciales:

- La libertad de ejecutar el programa como se desea, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para ayudar a su prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.

Un programa es software libre si otorga a los usuarios todas estas libertades de manera adecuada. De lo contrario no es libre. Existen diversos esquemas de distribución que no son libres, y si bien podemos distinguirlos en base a cuánto les falta para llegar a ser libres, nosotros los consideramos contrarios a la ética a todos por igual. (gnu.org, 2014)

## 7.2 Esquema general “Cliente Servidor”

Figura 4 : Dispositivos de entrada y salida conectados a servidor



Fuente: (galeon.com, 2013)

Como vemos, tenemos una máquina servidora que se comunica con variados clientes, todos demandando algún tipo de información. Esta información puede ser

desde archivos de texto, video, audio, imágenes, emails, aplicaciones, programas, consultas a base de datos, etc.

Los servidores descritos en la tabla 4 hacen parte de la plataforma cliente servidor implementada en el proyecto desarrollado.

## 7.2.1 Tipos de servidores

Tabla 4 : Tipos de servidores

DENOMINACIÓN DEL SERVIDOR	DESCRIPCIÓN
Servidor Web	Almacena principalmente documentos HTML (son documentos a modo de archivos con un formato especial para la visualización de páginas web en los navegadores de los clientes), imágenes, videos, texto, presentaciones, y en general todo tipo de información. Además se encarga de enviar estas informaciones a los clientes.
Servidor de Base de Datos	Da servicios de almacenamiento y gestión de bases de datos a sus clientes. Una base de datos es un sistema que nos permite almacenar grandes cantidades de información. Por ejemplo, todos los datos de los clientes de un banco y sus movimientos en las cuentas.

Fuente: (Sierra & aprenderaprogramar.com)

### 7.2.2 Tipos de Nubes

El proyecto implemento el concepto de nube hibrida explicada a continuación tabla 5.

**Tabla 5 : Tipos de Nubes**

<b>DENOMINACIÓN NUBE</b>	<b>DESCRIPCIÓN</b>
Dependiendo de las necesidades de cada empresa existen diversos tipos de nubes, el modelo de servicio ofrecido y la implementación de la misma.	
Nubes Publicas	Modelo estándar de computación en la nube, se compone de servidores externos al usuario. Donde se pueden tener acceso a aplicaciones de forma gratuita o de pago.
Nubes Privadas	La plataforma o montaje de la misma se encuentra al interior de cada empresa y no ofrece servicio a terceros. La nube privada generalmente es una plataforma para la obtención de hardware solamente (Maquinas de almacenamiento e infraestructura de red (IaaS <sup>3</sup> )). Son una buena elección para las compañías que necesitan proteger sus datos.
Nubes Hibridas	Estas plataformas integran recursos locales de la nube privada con la nube pública. Los servicios de la nube privada aumentan cuando se fusionan los servicios de la nube pública.

<sup>3</sup> Infraestructure as a Service (IaaS). Capacidad de proceso (CPU) y almacenamiento en la nube, sólo se paga por lo que se usa en ella. (xatakaon, 2011), (computacionennube, 2010).

Las empresas se benefician al poder tener control de su infraestructura y aprovechar la computación en la nube pública cuando sea oportuno.

Fuente: (computacionennube, 2010).

### **7.3 Internet de las cosas-Telefonía móvil**

#### **Telefonía móvil**

Dispositivo inalámbrico electrónico para acceder y utilizar los servicios de la red de telefonía móvil. Funciona mediante una red de celdas, donde cada antena repetidora de señal es una célula. La Real Academia Española da como definición, denominándola móvil al “sistema telefónico en el que la conexión entre el aparato portátil y la central se realiza mediante ondas hercianas.” (Quees.la, 2014)

#### **Aplicativo móvil**

Aplicación móvil o App diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Se encuentran disponibles a través de plataformas de distribución móviles como Android, iOS, BlackBerry OS, Windows Phone, entre otros.

La aplicación tiene el único y principal fin de realizar una tarea específica, a menudo básica y de rápida ejecución de fácil uso para el usuario común no avanzado.

En definitiva, una aplicación informática sirve para ahorrar tiempo y dinero al usuario y, por eso, constantemente nuevas aplicaciones son desarrolladas ya sea por usuarios básicos, avanzados o programadores, a los efectos de simplificar el

uso de un ordenador lo más posible. (ABC, 2015) A continuación se describen aplicaciones móviles y trabajos que implementan sensores ambientales tabla 6.

**Tabla 6 : Aplicativos desarrollados con sensores ambientales.**

<b>Aplicaciones con sensores ambientales</b>	
AirBot	Descripción
<b>Figura 5 : AirBot</b>	
	<p>Es un "robot de contador de partículas" desarrollado por la Universidad Carnegie Mellon que monitorea los contaminantes del aire que pueden causar problemas respiratorios como el asma.</p>
Fuente: (treehugger, 2013)	
WaterBot	
<b>Figura 6 : WaterBot</b>	
	<p>Desarrollado por Carnegie Mellon, los WaterBot realizan pruebas de calidad del agua. Los datos son transmitidos por un módulo ZigBee, los datos se transmiten en tiempo real. Ayudando a las personas que viven cerca de fuentes de agua a permanecer informados.</p>

Fuente: (treehugger, 2013)

**Sensordrone**

**Figura 7 : Sensordrone**



Desarrollado por Kickstarter, herramienta que puede detectar muchas variables del entorno, incluyendo gases, temperaturas y humedad. Ejecuta aplicaciones poniendo a prueba cada cosa. Dispositivos iPhone

Fuente: (treehugger, 2013)

**Lapka Environmental Monitor**

**Figura 8 : Lapka Environmental Monitor**



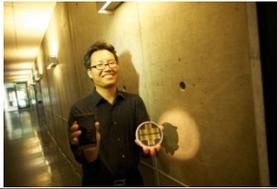
Conjunto de sensores ambientales que se conectan a un iPhone. Entre las especificaciones puede detectar la radiación, la retroalimentación electromagnética, nitratos en los alimentos crudos, y la temperatura y la humedad.

Fuente: (treehugger, 2013)

<b>Sensaris</b>	
<b>Figura 9 : Sensaris</b>	
	<p>Sensor que se instala en la muñeca, dando mediciones instantáneas de la calidad del aire en cualquier lugar donde estes. Los datos son enviados a través de Bluetooth a su dispositivo móvil.</p>
Fuente: (treehugger, 2013)	
<b>Air Quality Egg</b>	
<b>Figura 10 : Air Quality Egg</b>	
	<p>Desarrollado por Kickstarter, Air Quality Egg sensor ambiental en el hogar, recoge lecturas de muy alta resolución sobre concentraciones de NP2 y CO. Se comunica de forma inalámbrica con la estación base.</p>
Fuente: (treehugger, 2013)	

<b>Electronic Nose Sensor</b>	
-------------------------------	--

**Figura 11 : Electronic Nose Sensor**



Desarrollado por la Universidad de California en Riverside, dispositivo multi-sensor capaz de detectar pequeñas cantidades de productos químicos en el aire peligrosos como pesticidas, las emisiones de combustión, fugas de gas, y los agentes de guerra química. No está disponible al mercado.

Fuente: (treehugger, 2013)

<b>PressureNet</b>	
--------------------	--

**Figura 12 : PressureNet**

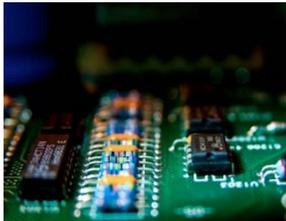


Aplicación con Android que mide la presión atmosférica, y proporciona esas mediciones para los científicos que a su vez lo utilizan para comprender mejor lo que está sucediendo con el clima. La aplicación utiliza sensores atmosféricos que ya se encuentran en muchos de los teléfonos Android.

Fuente: (treehugger, 2013)

<b>Broadcom Microchip</b>	
---------------------------	--

**Figura 13 : Broadcom Microchip**



Aprovechar la gran cantidad de sensores que los smartphones contienen para así reunir información precisa sobre el entorno del usuario. Lo cual podría proporcionar a los científicos valiosos datos para monitorear y mediar contaminantes y otras amenazas ambientales.

Fuente: (treehugger, 2013)

**iGeigie**

**Figura 14 : iGeigie**



Desarrollado después de la catástrofe de Fukushima en Japón, se acopla con un iPhone, los usuarios pueden escuchar los clics que indican la cantidad de radiación que se encuentra en la zona. El principal objetivo de los desarrolladores es crear una red de sensores para la radiación nuclear donde los datos pueden ser mapeados y grupos gubernamentales

Fuente: (treehugger, 2013)

Fuente: (Calderon Calderon , Estado del arte, 2015)

**Sistemas operativos para dispositivos móviles**

El Sistema Operativo (SO) móvil de un teléfono o tableta significa la interacción real con lo que podemos hacer a partir de las capacidades del hardware que conforman un equipo. A manera de traductor, esta plataforma interpreta lo que el usuario quiere que la terminal realice y cada vez, lo ejecuta con mayor inteligencia.

Una de las cualidades más atractivas de un sistema operativo móvil es la rapidez que en general se desempeña. No precisa apagar el equipo completamente, sino dejarlo en un estado de suspensión para ahorrar energía, las aplicaciones se lanzan en pocos segundos, la instalación es transparente para el usuario y muchos periféricos son actualmente compatibles con los dispositivos más comunes. Tal pareciera que la única diferencia con una PC tradicional es que todavía no soportan aplicaciones robustas como podrían ser las enfocadas en diseño o edición de video profesional. (PCWorld, 2012)

Ejemplos de sistemas operativos móviles:

- Android
- iOS
- Windows Phone
- BlackBerry OS
- Symbian OS
- Firefox OS
- 

### **API, Interface de Programación de Aplicaciones**

Una API (Application Programming Interface o Interfase de Programación de Aplicaciones) es un conjunto de funciones que facilitan el intercambio de mensajes o datos entre dos aplicaciones. Es una forma de que dos aplicaciones que

trabajan al mismo tiempo –como podría ser un procesador de texto y una hoja de cálculo– se comuniquen e intercambien datos.

Un API es un conjunto de reglas para escribir funciones o hacer llamados a subrutinas y acceder a otras funciones en una librería. Los programas que usan estas reglas o funciones en sus llamadas API pueden comunicarse con cualquiera que use dicha API.

Las API abren distintos tipos de diálogos con el proveedor para obtener o actualizar información en el mismo, entre ellos:

- Acceso a bases de datos
- Comunicación cliente/servidor
- Comunicación peer-to-peer
- Comunicación en tiempo real
- Event-driven (orientada a eventos)
- Store and forward
- Procesamiento de transacciones

La API toma la información y hace que todo el trabajo específico de comunicación sea transparente para la aplicación. Hay cuatro tipos de APIs que posibilitan el intercambio de datos entre diferentes aplicaciones de software en plataformas individuales o distribuidas:

- **Remote Procedure Calls (RPCs) o Llamadas a Procedimientos Remotos**

Con las RPCs, los programas pueden comunicarse a través de procedimientos (o tareas) que actúan en buffers compartidos de información.

- **Standard Query Language (SQL)**

Un lenguaje de acceso de datos sin procedimientos que permite el intercambio de datos entre aplicaciones, accediendo a bases de datos comunes.

- **Transferencia de archivos**

Habilita el intercambio de datos enviando archivos formateados entre aplicaciones.

- **Envío de mensajes**

Provee el intercambio de datos por comunicaciones de interprogramación directa, a través de pequeños mensajes formateados entre aplicaciones emparejada

#### **7.4 Notificaciones push.**

**La tecnología push es un tipo de comunicación en la que es el servidor el que inicia la petición al cliente** (el móvil, por ejemplo) cuando tiene una información o notificación nueva, permitiendo un importante ahorro de recursos y tiempo respecto a la tecnología convencional pull<sup>4</sup>.

Un **ejemplo muy sencillo** para entender la tecnología push es el sistema de recepción de correo electrónico de los dispositivos BlackBerry. La gran novedad de esta tecnología fue que permitió recibir los email según llegaban al servidor de correo, al enviar éste una notificación push al dispositivo para que recogiese los mensajes. La diferencia con respecto al sistema convencional de correo electrónico es evidente, con el Outlook convencional nos vemos obligados a pulsar

---

<sup>4</sup> Se utiliza la tecnología pull (tirar) cuando se navega por el World Wide Web para buscar y descargar información en el ordenador. Esto contrasta con la tecnología push (empujar), cuando los datos son entregados directamente al ordenador del usuario.

“Enviar y Recibir” para actualizar nuestra bandeja de entrada, o por el contrario programar nuestro servicio de correo para que se actualice cada cierto intervalo de tiempo, siendo todo el proceso ineficiente.

Las dos ventajas fundamentales de la mensajería push con respecto a las basadas en tecnología pull son:

- **La notificación es instantánea**, recibimos los mensajes en el acto según llegan al servidor.

-Las notificaciones push no requieren que la aplicación esté ejecutándose en el dispositivo y realizando llamadas constantes al servidor, sino que permite que ésta esté en segundo plano y se active al recibir un mensaje. Esto implica un **importante ahorro de batería y recursos del teléfono**. Un ejemplo esclarecedor es la aplicación WhatsApp para iPhone, la aplicación más exitosa de la historia. Aunque tengamos la aplicación apagada o en segundo plano, siempre que alguien nos escriba será la notificación push la que se encargue de iniciar el proceso de reactivación de la app y que recibamos el mensaje.

### **Integración de la tecnología push en aplicaciones móviles**

BlackBerry fue la primera plataforma móvil que integró la tecnología push para el servicio de email de sus dispositivos. La compañía RIM, (Research In Motion) llegó a un acuerdo con los operadores de telefonía para establecer una conexión permanente en la que basar sus servicios. Esta tecnología fue puesta a disposición de los desarrolladores a través del BlackBerry push API. Recientemente, y a través de una actualización del BB 10 webworks SDK beta, los

programadores en HTML5 para esta plataforma podrán aprovechar al fin la infraestructura push para sus aplicaciones.

La siguiente compañía en lanzar su propia infraestructura de notificaciones mediante tecnología push fue Apple a través del APNs (Apple Push Notification service) publicado junto a la versión 3.0 de iOS en Junio de 2009. Este servicio se basa en una conexión IP abierta permanentemente para dar soporte a las necesidades de las apps de los desarrolladores. La creación y diseño de este servicio se enmarcó dentro de la búsqueda de Apple de optimización del gasto de batería. La solución tomada por la compañía consistió en que las aplicaciones no se ejecutasen en segundo plano, pero que tuviesen a su disposición un servicio (el APNs) para que pudiesen ser activadas en caso de necesidad, como en el caso de la mensajería. El contenido que se puede enviar bajo este servicio se ve limitado a 256 bytes.

A continuación se ilustra el proceso General GCM<sup>5</sup>. Figura 15

**Figura 15 : GCM Proceso General GCM**



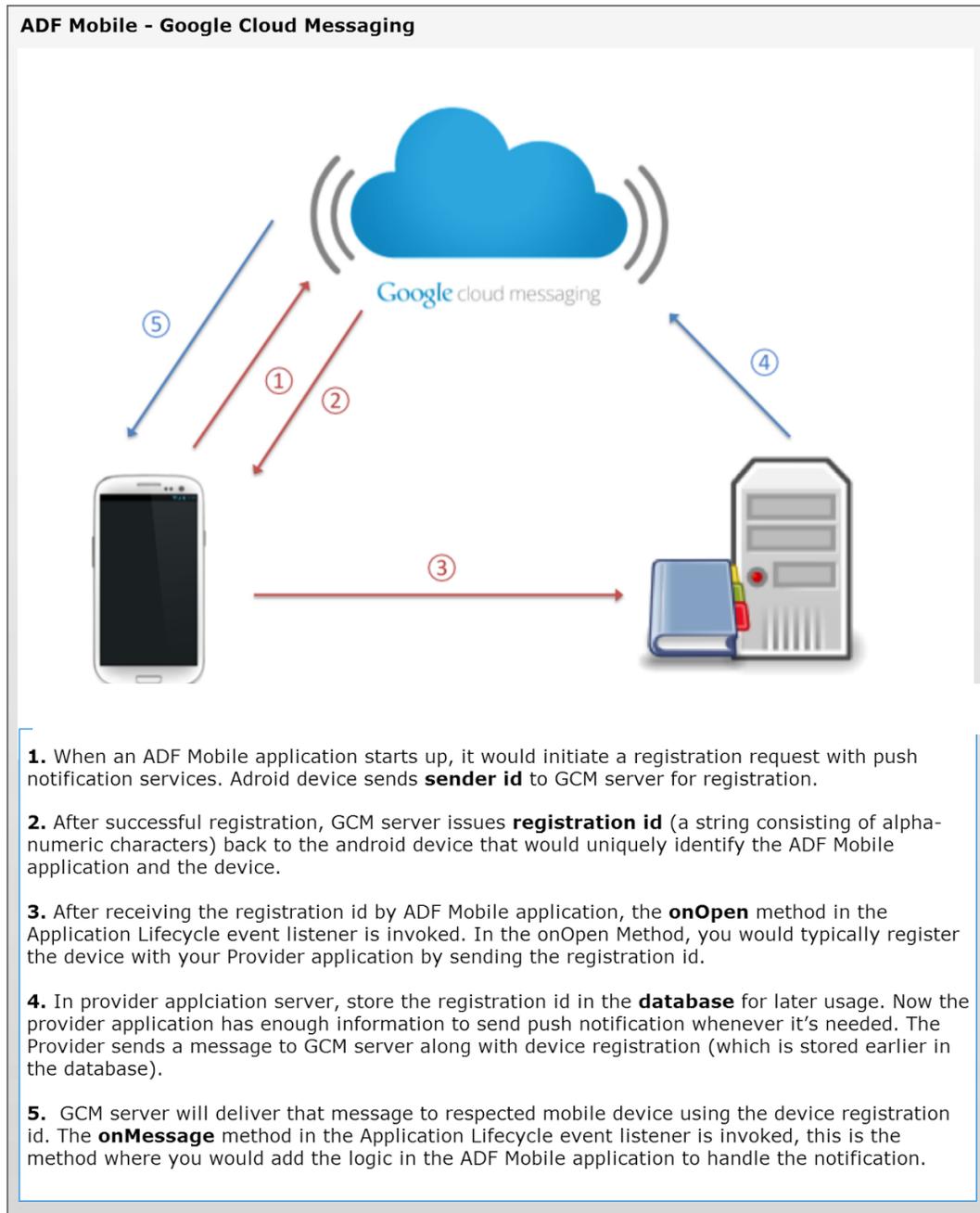
Fuente: (sgoliver, 2014)

La última plataforma en incorporar un servicio push fue Android a través del C2DM (Cloud to Device Messaging) de Google. Este servicio, como sucedía en los casos anteriores, se encarga de la gestión de cola y envío de las notificaciones a la app cliente. De la misma manera, las aplicaciones no tienen que estar ejecutándose en el momento de recibir la notificación para reactivarse e interactuar con el servidor. Recientemente, Google ha actualizado su servicio al denominado GCM (Google Cloud Messaging) que sustituye a C2DM, y al que dejará de dar soporte. Los requisitos para los móviles que quieran utilizar la infraestructura son disponer una versión de Android 2.2 o superior y tener instalado Google Play. En cuanto a las restricciones de tamaño de las notificaciones, con el nuevo GCM pasarán a 4Kb desde los 1014 bytes del C2DM.

Dato curioso es la liberación por parte de Instagram de node2dm, un servidor node.js para enviar notificaciones push a los dispositivos Android a través del servicio C2DM y dar libertad a la comunidad para que trabaje sobre él y mejore el servicio que le ha permitido gestionar más de 5 millones de notificaciones en su primer día. (startcapps, 2014)

El diagrama ilustra el concepto de trabajo visión general de ADF Mobile con soporte Push Notificación.

**Figura 16 : ADF Mobile**



Online diagramming & design **creately**.com

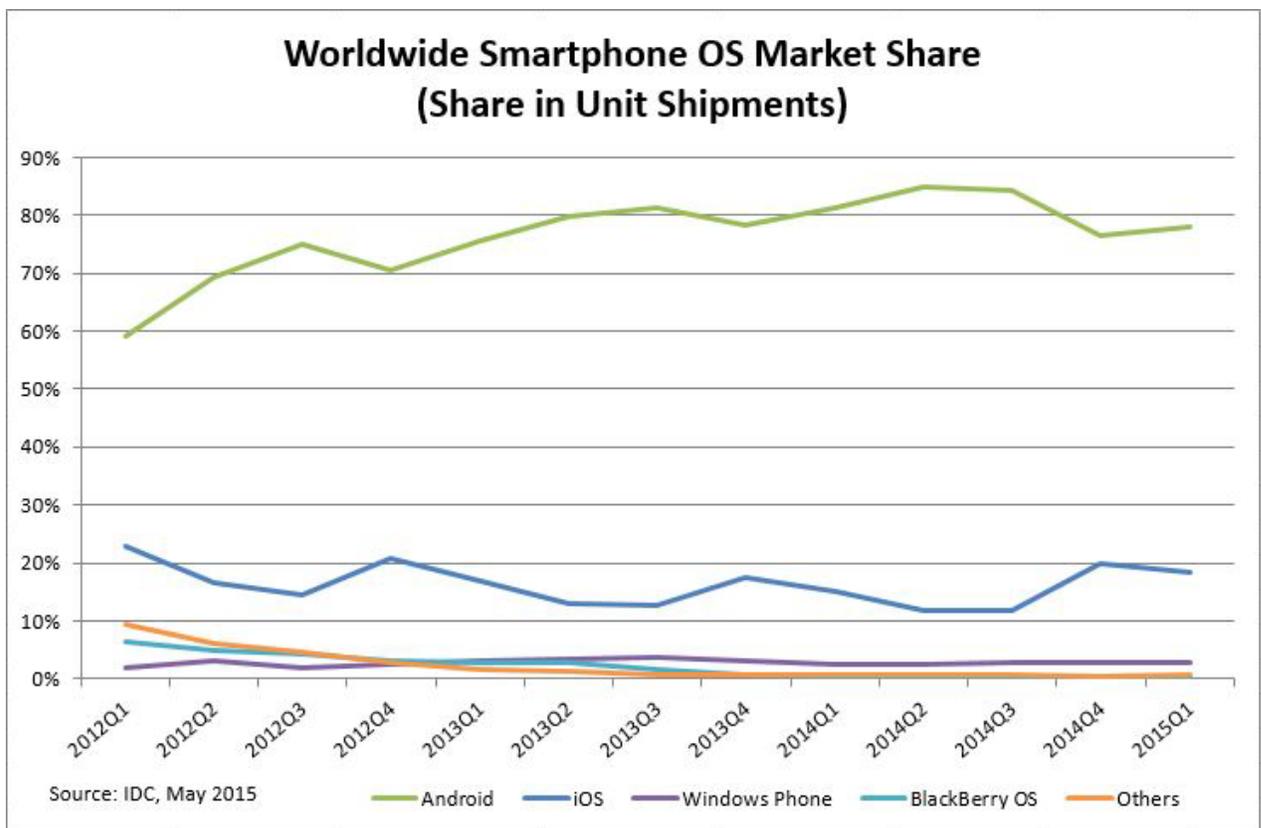
Fuente: (deepakcs, 2013)

## 7.5 Lenguajes y estadísticas

### 7.5.1 Android

Las gráficas estadísticas y tablas presentadas a continuación hacen parte de la investigación de las plataformas de sistemas operativos para dispositivos móviles implementadas en todo el mundo. Figura 17, Tabla 7.

**Figura 17: Sistemas operativos Smartphone en todo el mundo**



Fuente: (IDC, 2015)

**Tabla 7 : Sistemas operativos Smartphone en todo el mundo**

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
Q1 <sup>6</sup> 2015	78.0%	18.3%	2.7%	0.3%	0.7%
Q1 2014	81.2%	15.2%	2.5%	0.5%	0.7%
Q1 2013	75.5%	16.9%	3.2%	2.9%	1.5%
Q1 2012	59.2%	22.9%	2.0%	6.3%	9.5%

Fuente: (IDC, 2015)

**Tabla 8 : Informe reportes 2015 EE.UU. Smartphone suscritos.**

*Top Smartphone OEMs  
3 Month Avg. Ending Feb. 2015 vs. 3 Month Avg. Ending Nov. 2014  
Total U.S. Smartphone Subscribers Age 13+  
Source: comScore MobiLens*

Smartphone OEM	Share (%) Nov-14	Share (%) Feb-15	Point Change
Total Smartphone Subscribers	100%	100%	0
Apple	41,8%	41,7%	-0,1
Samsung	29,7%	28,6%	-1,1
LG	7,6%	8,3%	0,7
Motorola	5,2%	5,1%	-0,1
HTC	3,7%	3,8%	0,1

<sup>6</sup> Los vendedores envían un total de 334,4 millones teléfonos inteligentes en todo el mundo en el primer trimestre de 2015 (1Q15), hasta el 16,0% de los 288,3 millones de unidades en 1Q14, pero por 11,4 % de los 377,6 millones de unidades vendidas en el 4Q14. Android domina el mercado con una participación de 78,0% en el 1Q15.

Fuente: (MobiLens, 2015)

**Tabla 9 : Top de plataformas Smartphone EE.UU**

Top Smartphone Platforms 3 Month Avg. Ending Feb. 2015 vs. 3 Month Avg. Ending Nov. 2014 Total U.S. Smartphone Subscribers Age 13+ Source: comScore MobiLens			
<b>Smartphone OEM</b>	<b>Share (%) Nov-14</b>	<b>Share (%) Feb-15</b>	<b>Point Change</b>
Total Smartphone Subscribers	100%	100%	0
Android	52,6%	52,8%	0,2
Apple	41,8%	41,7%	-0,1
Microsoft	3,4%	3,5%	0,1
BlackBerry	2,0%	1,8%	-0,2
Symbian	0,1%	0,1%	0,0

Fuente: (MobiLens, 2015)

Por tanto Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y Blackberry OS. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma.

El sistema permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla en un lenguaje de programación muy conocido como es Java.

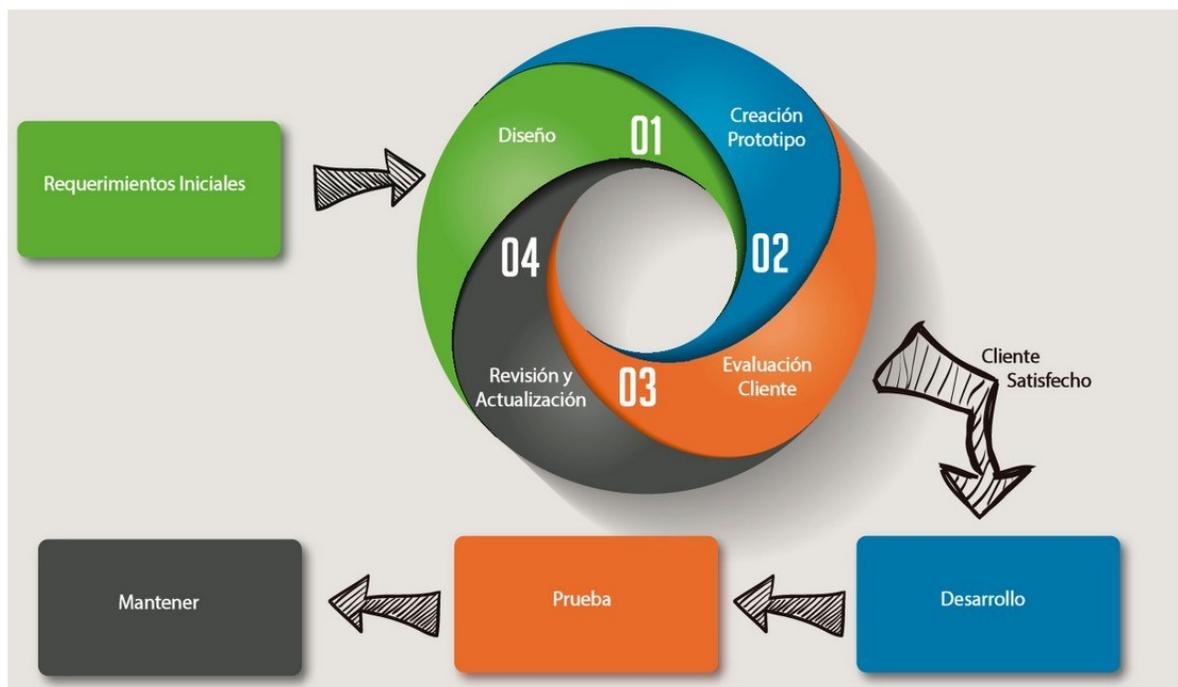
Principales características más importantes del sistema operativo:

- Código abierto.
- Núcleo basado en el Kernel de Linux.
- Adaptable a muchas pantallas y resoluciones.
- Utiliza SQLite para el almacenamiento de datos.
- Ofrece diferentes formas de mensajería.
- Navegador web basado en WebKit incluido.
- Soporte de Java y muchos formatos multimedia.
- Soporte de HTML, HTML5, Adobe Flash Player, etc.
- Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software.
- Catálogo de aplicaciones gratuitas o pagas en el que pueden ser descargadas e instaladas (Google Play).
- Bluetooth.
- Google Talk desde su versión HoneyComb, para realizar videollamadas.
- Multitarea real de aplicaciones.

## 8. DISEÑO METODOLOGICO

En la implementación del diseño metodológico del proyecto formulado, se llevó a cabo el proceso investigativo paso a paso de todas las actividades para el buen desarrollo del trabajo final. En el desarrollo del prototipo en su fase inicial se utilizó el siguiente modelo representado en la Figura 18 a continuación.

**Figura 18 : Metodología modelo prototipo**



Fuente: (Calderón Calderón, 2015)

En esta sección se describe la implementación del prototipo que abarca 4 pilares fundamentales. En el desarrollo de un sistema telemático no invasivo de monitoreo de condiciones ambientales y generación de informes en tiempo real. (Reportes, Sistema Electrónico, Computacional y Control), la subdivisión de las actividades ayudo al desarrollo de forma ordenada de cada uno de los objetivos.

El proceso de diseño metodológico fue inicializado con la:

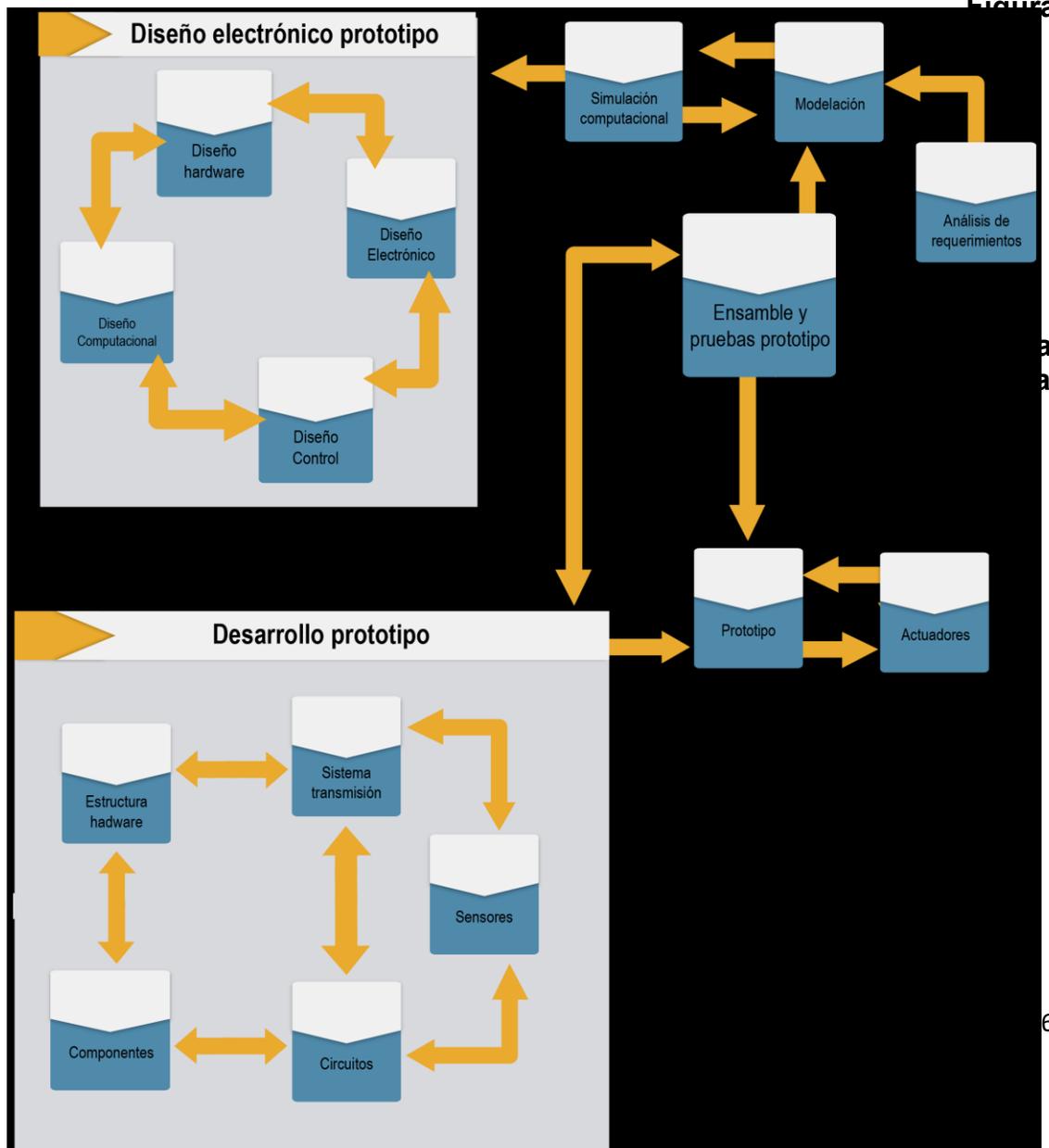
- Selección del tema de investigación.
- Exploración bibliográfica respecto al tema seleccionado.
- Fortalecimiento del problema en estudio, justificación y objetivos.
- Construcción del cronograma de actividades, presupuesto y recursos necesarios.
- Consolidación del estado del arte y marco teórico.
- Ejecución del estudio comparativo de herramientas de hardware y software libre y tecnología IoT.
- Selección hardware y software para el desarrollo y viabilidad del proyecto.
- Propuesta de prototipo funcional no invasivo para el monitoreo y generación de informes IoT.
- Configuración de la plataforma de conexión entre el modelo de arquitectura hardware con cada uno de sus componentes a una base de datos, al portal web y a la interfaz móvil.

Finalmente se determinó el comportamiento en el medio y ajusto sistema de reportes con notificaciones push.

Verificado el funcionamiento y desarrollo del proyecto con esta metodología, se dió paso a efectuar documentación. La cual recopilo toda la información de planos, circuitos, manual de uso entre otros.

### Diseño metodológico arquitectura

El desarrollo del prototipo en su fase de arquitectura concluyo con el desarrollo del siguiente diagrama metodológico figura 19.



Figura

Fuente: (Calderón Calderón, 2015)

## **9. RESULTADOS**

### **9.1 CARACTERIZACIÓN DE LAS CONDICIONES ÓPTIMAS DE TEMPERATURA Y HUMEDAD EN INTERIORES**

Dentro de las condiciones de trabajo, la temperatura es un factor determinante en el rendimiento de un trabajador. Temperaturas muy altas o demasiado bajas repercuten en el operador de manera perjudicial, no solo para su labor sino también en su salud.

La temperatura normal del cuerpo varía según la persona, la edad, la actividad y el momento del día. La temperatura corporal normal promedio que generalmente se

acepta es de 98.6° F (37° C). Algunos estudios han demostrado que la temperatura "normal" cuerpo puede tener una amplia gama, desde 97 ° F (36.1 ° C) a 99 ° F (37.2 ° C). (MACKOWIAK, 2014)

Una temperatura por encima de 100.4° F (38° C) generalmente significa que usted tiene una fiebre causada por una infección o una enfermedad.

La temperatura corporal normalmente cambia a lo largo del día. Para temperaturas superiores a 40.6 ° C el mecanismo de sudor puede fallar y causar una elevación rápida del núcleo y con el tiempo la muerte.

El intercambio de calor entre el cuerpo y su entorno se puede representar por la siguiente ecuación de balance térmico:

$$A = M \pm R \pm C - E$$

**Tabla 10 : Ecuación de balance térmico I**

Ecuación de balance térmico → Equilibrio térmico → cuando el cuerpo no gana ni pierde calor.	
A	Calor acumulado en el organismo
M	Ganancia de calor por el metabolismo
R	Ganancia o pérdida de calor por radiación
C	Ganancia o pérdida de calor por convección
E	Pérdida de calor por evaporación del sudor

Fuente: (Riesgolaboral.net, 2014)

**Tabla 11 : Ecuación de balance térmico II**

1	$M \pm R \pm C = 0$	No hay calor acumulado
---	---------------------	------------------------

		No necesita evaporar calor para lograr el equilibrio térmico
		Situación de confort térmico
2	$M \pm R \pm C - E = 0$	No hay equilibrio térmico
		No necesita evaporar sudor para lograr el equilibrio, ya que hay una tensión térmica.

**Tabla 12 : Ecuación de balance térmico III**

3	$M \pm R \pm C - E > 0$	No hay equilibrio térmico → Estrés térmico por calor.
		Se gana calor.
		El organismo incrementa la temperatura
4	$M \pm R \pm C < 0$	Se pierde calor
		No se necesita sudar y la temperatura del cuerpo desciende
		No hay equilibrio térmico → Estrés térmico por frío.

Fuente: (Riesgolaboral.net, 2014)

### 9.1.1 Temperatura

Propiedad de los sistemas que determina si están en equilibrio térmico. El concepto de temperatura se deriva de la idea de medir el calor o frío relativos y de la observación de que el suministro de calor a un cuerpo conlleva un aumento de

su temperatura mientras no se produzca la fusión o ebullición. En el caso de dos cuerpos con temperaturas diferentes, el calor fluye del más caliente al más frío, hasta que sus temperaturas sean idénticas y se alcance el equilibrio térmico.

### **9.1.2 Transferencia de calor**

El calor tiende a pasar desde los puntos en los que la temperatura es alta hacia aquellos en los que es inferior.

De acuerdo con los materiales en los cuales se está realizando la transferencia de calor se tienen diferentes procesos como son:

- **Conducción:** Cuando la transferencia de calor se realiza a través de sólidos o fluidos que no están en movimiento,
- **Convección:** Cuando la transferencia se realiza a través de fluidos en movimiento.
- **Radiación:** Cuando el calor es transferido de un cuerpo a otro sin soporte material alguno.

### **9.1.3 Temperatura ambiente**

Es la temperatura experimentada por una persona en un ambiente dado. Esta temperatura es el resultado del intercambio de calor por conducción (a través de

pisos o herramientas) y radiación (Muros, plafones, sol).

#### **9.1.4 Temperatura efectiva**

Temperatura representada por el efecto combinado de la temperatura ambiente, la humedad relativa de 20% a 60% y el movimiento del aire, en la sensación de calor o frío que percibe el cuerpo humano, equivalente a la temperatura del aire en reposo que produce un efecto idéntico.

#### **9.1.5 Humedad**

Medida de concentración de agua o vapor de agua en un sólido, un líquido o un gas. A continuación se presentan los tipos de humedad:

##### **Humedad atmosférica**

En Meteorología, como humedad atmosférica se denomina la cantidad de vapor de agua que hay en la atmósfera. Como tal, es medida con el higrómetro.

##### **Humedad absoluta**

Como humedad absoluta se denomina el peso del vapor de agua que, por unidad de volumen, contiene el aire de ambiente. Por lo general, se mide en gramos por metro cúbico.

##### **Humedad relativa**

La humedad relativa es el porcentaje de la masa de vapor de agua que hay en un volumen de aire, y la que este tendría si estuviera saturado. Mientras más próximo se encuentra este valor a 100% es indicativo de que más húmedo está el ambiente.

##### **Humedad Específica**

Es la relación entre la masa de agua o vapor de agua y la masa total. En el caso del aire se expresa en gramos de vapor de agua por kilogramo de aire húmedo.

#### **9.1.6 Zona comfortable**

Es el intervalo normal de temperatura efectiva. Se recomiendan temperaturas de **18.8 °C y 22.9 °C** como límites externos para la regulación termostática.

#### **Temperatura operativa**

Es la temperatura del cuerpo de un trabajador. Se determina por los efectos acumulativos de todas las fuentes y receptores de calor.

#### **Fatiga por calor**

Se presenta cuando hay un ascenso máximo en la temperatura del cuerpo de un individuo de  $1\text{ }^{\circ}\text{C}$

#### **9.1.7 Zona de conformidad térmica**

Para el estudio de temperaturas se ha determinado una zona de confort térmico para áreas donde se realiza trabajo ligero y sedentario durante 8 horas.

Este intervalo se encuentra a temperaturas entre los  $18.9\text{ }^{\circ}\text{C}$  y  $26.1\text{ }^{\circ}\text{C}$ , con una humedad relativa de 20% a 80%.

Sin embargo la ropa y la radiación de calor afectan el sentido individual de comodidad dentro de esta zona de conformidad Figura 20.

### **Figura 20: Zona de confort térmico**

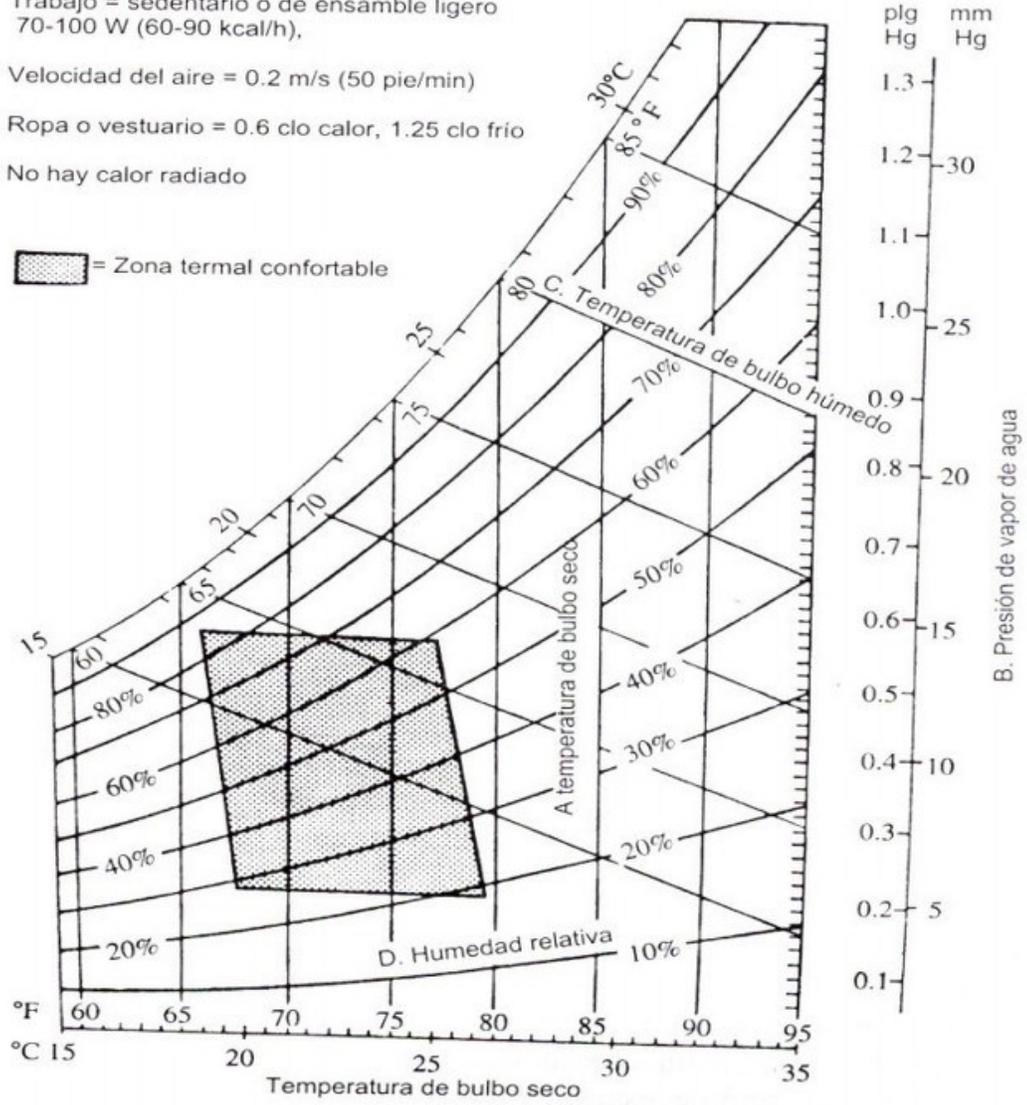
Trabajo = sedentario o de ensamble ligero  
70-100 W (60-90 kcal/h),

Velocidad del aire = 0.2 m/s (50 pie/min)

Ropa o vestuario = 0.6 clo calor, 1.25 clo frío

No hay calor radiado

 = Zona termal confortable



Fuente: (Niebel & Benjamín)

## **9.2 ANALIZAR Y DISEÑAR UN SISTEMA INFORMÁTICO EN FUNCIÓN DE: REQUERIMIENTOS FUNCIONALES, MODELO DE BASE DE DATOS, ARQUITECTURA WEB, Y ARQUITECTURA MÓVIL (ANDROID).**

De lo expuesto anteriormente y siguiendo paso a paso lo propuesto en el diseño metodológico Figura 18. Se desarrolló un sistema que permite monitorear en forma continua condiciones ambientales de espacios interiores cerrados y detectar cuando la temperatura y humedad aplicadas al caso de estudio se encuentran fuera de los límites de la zona de confort térmica. Previniendo posibles alteraciones y de tal manera que se puedan tomar acciones correctivas en el momento necesario.

### **9.2.1 Requerimientos del sistema**

- El sistema debe permitir monitorear condiciones ambientales (Temperatura y humedad) de forma inalámbrica sin un mayor margen de error.
- El sistema debe permitir monitorear en tiempo real las variables necesarias para identificar posibles alteraciones en el espacio a supervisar.
- El sistema debe llevar un registro permanente y actualizable de los datos capturados tales como: temperatura, humedad, fecha y hora de tal forma que el usuario pueda detectar anomalías y tomar correctivos necesarios en el menor tiempo posible.
- El sistema debe ser accesible desde cualquier dispositivo con acceso a internet, además debe contar con una aplicativa móvil plataforma (Android). El cual muestre en tiempo real las estadísticas de las variables a

monitorear, a su vez esta aplicación debe contar con Push Notificación que debe ser activada en forma inmediata cuando una variable este fuera de su margen normal.

- El sistema debe ser de bajo costo en su hardware, autónomo y escalable.

### 9.2.2 Arquitectura del sistema

En la Figura 21 se representa la arquitectura de sistema desarrollado, compuesto a nivel de hardware por:

- **Prototipo iCrCThings** (*Nombre con el cual se hace referencia al prototipo ensamblado*): este módulo está compuesto por un sensor de temperatura y humedad encargado de medir las variables que actualmente el ambiente este generando, con el fin de obtener los datos en tiempo real de las fluctuaciones térmicas del espacio interior a procesar. El prototipo también cuenta con una placa Arduino Mega2560, una placa Arduino wifi Shield y una pantalla. Este módulo puede ser escalable y ajustable tanto a las necesidades del cliente como al espacio a supervisar.
- **Modulo servidor:** Se encarga de coordinar la red conformada por el iCrCThings y Arquitectura móvil de tal manera que los datos obtenidos por el prototipo iCrCThings. Son procesados, almacenados y mostrados en una interfaz GUI (*Interfaz gráfica de Usuario*).
- **Aplicación Móvil:** Este prototipo consta de un aplicativo desarrollado en plataforma Android con una interfaz que procesa los datos almacenadas en el módulo servidor de tal forma que se puedan visualizar en una gráfica

estadística, además está en constante escucha de notificaciones tipo Push provenientes del api (Tembo) que se activa desde prototipo iCrCThings.

**Figura 21 : Diagrama general del sistema de monitoreo.**



Fuente: (Calderon & Tarazona, Diagrama general sistema de monitoreo, 2015)

En el servidor y aplicativo móvil el usuario podrá observar el comportamiento de las fluctuaciones térmicas del lugar procesado en tiempo real. El prototipo iCrCThings periódicamente escribe en la base de datos conforme una

eventualidad suceda y automáticamente activa la notificación push si esta fuera del rango. Todo esto sin que el usuario intervenga directamente el sistema.

### **9.2.3 Hardware del sistema iCrCThings.**

En la siguiente fase se muestran los componentes para la construcción del hardware del sistema.

#### **A. Hardware libre**

- **Arduino mega2560.**

Este módulo está integrado a una placa Arduino Mega 2560 microcontroller ATmega2560 que cuenta con 54 pines digitales de entrada/salida, 16 entradas analógicas 4 UARTs (hardware puertos serie), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un header ICSP y un botón de reinicio.

El Atmega2560 tiene 256 KB de memoria flash para almacenar código (de los cuales 8 KB se utiliza para el cargador de arranque), 8 KB de SRAM y 4 KB de EEPROM (que puede ser leído y escrito con la biblioteca EEPROM). Se usó esta tecnología debido a su baja potencia, bajo costo y batería de larga duración, así como la facilidad de programación en C del microcontrolador

- **Arduino Wifi Shield.**

En la construcción del prototipo se utilizó el Arduino WiFi Shield que permite conectarse a internet utilizando una especificación inalámbrica 802.11 (Wi-Fi). Se basa en el estándar 802.11b/g Sistema HDG204 LAN inalámbrica en

paquete. Un AT32UC3 que proporciona una red IP que soporta los protocolos de transporte TCP y UDP.

- **LCD display.**

Una vez escogida la tecnología a utilizar, cumpliendo con los requerimientos mencionados anteriormente se instala el modulo Lcd display en cual es adaptable a esta plataforma Arduino. Permitiendo la interfaz del usuario con los sensores instalados a la placa Arduino Mega2560.

### **B. Sensor de Temperatura y Humedad DTH22.**

Dados los requerimientos iniciales, se optó por elegir el sensor DTH22 fabricante MIKROELEKTRONIKA. El cual cumple con los estándares para la toma de datos en este espacio interior cerrado a evaluar.

Se eligió principalmente por sus parámetros compatibles con el hardware seleccionado y sus características relevantes Tabla 13.

**Tabla 13 : Parámetros DTH22**

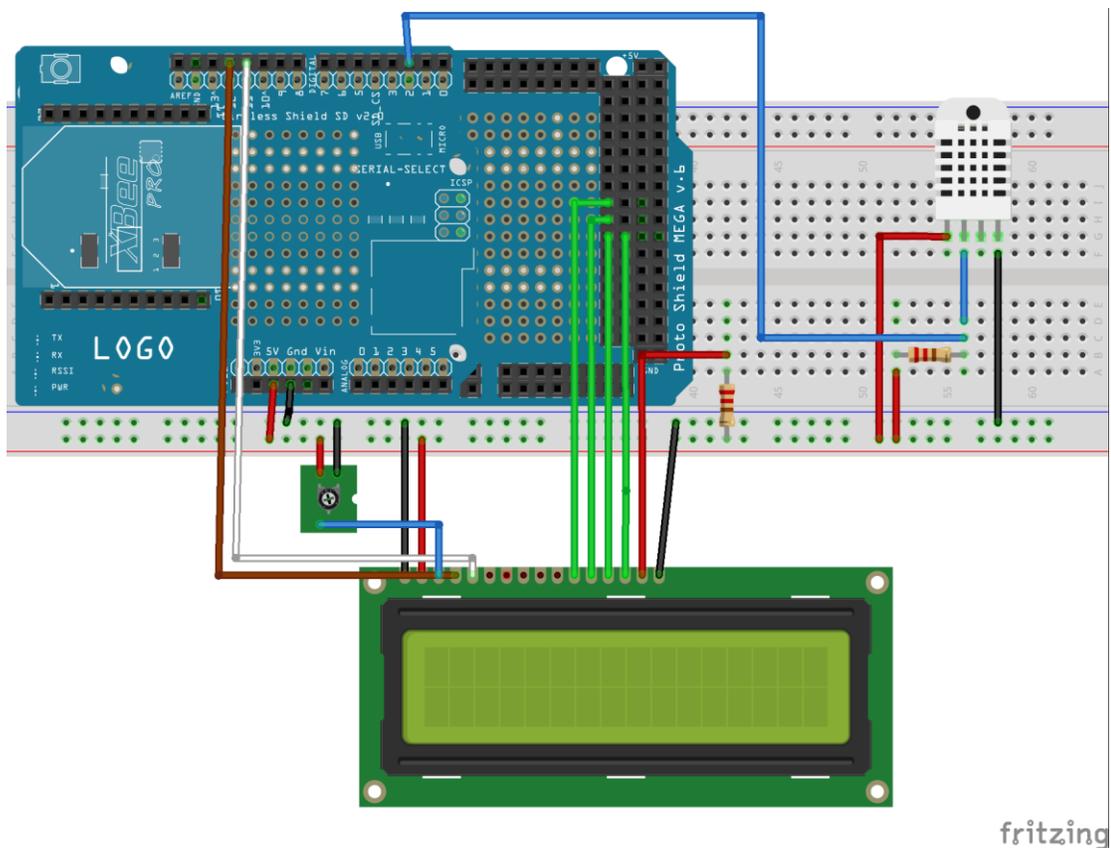
<b>Parámetro</b>	<b>DHT22</b>
Alimentación	$3.3V_{dc} \leq V_{cc} \leq 6V_{dc}$
Señal de Salida	Digital
Rango de medida Temperatura	De $-40^{\circ}C$ a $80^{\circ}C$
Precisión Temperatura	$<\pm 0.5^{\circ}C$
Resolución Temperatura	$0.1^{\circ}C$
Rango de medida Humedad	De 0 a 100% RH
Precisión Humedad	2% RH
Resolución Humedad	0.1%RH
Tiempo de sensado	2s
Tamaño	14 x 18 x 5.5mm

Fuente: (Electronics, 2013)

## 9.2.4 Diseño del dispositivo iCrCThings.

En la Figura 22. Se aprecian las diferentes conexiones realizadas para el funcionamiento del prototipo. Se emplea el Arduino Mega2560 y Arduino Wifi Shield. Previamente se instala el LCD display y sensor DTH22. La alimentación de energía que da soporte a este dispositivo es provista por una fuente de poder de 9 V DC 1 A.

Figura 22 : Sketch iCrCThings



Fuente: (Calderon Calderon, 2015)

Dado que todos los módulos son Arduino la compatibilidad de energía es de 5V ventaja a la hora de ensamblar el prototipo.

El sensor DHT22 tiene una conexión digital directamente al Pin 2 del Wifi Shield. El cual previamente fue programado para que sus pulsos digitales fueran interpretados y almacenados, dada la interacción entre el módulo Arduino Wifi Shield y la base de datos.

Los datos capturados provenientes del sensor e interpretados por el Arduino Mega2560. Viajan a través de internet utilizando el protocolo de datagrama de usuario IP/UDP, ideal para comunicaciones en tiempo real. Esta configuración depende del usuario como desee que las tramas de datos se envíen desde el modulo central Arduino Wifi Shield a una base de datos alojada en el servidor.

Luego de haber configurado el prototipo en su hardware se procede a generar la programación tanto en el Arduino, página web y aplicativo móvil.

### **9.2.5 Desarrollo de software.**

#### **A. Software Local Arduino**

El software desarrollado en esta plataforma tiene la capacidad de: leer, notificar, escribir y enviar datos por medio del Arduino Wifi Shield. Conectado de forma inalámbrica a la infraestructura de red del entorno. Este recolecta e interpreta los datos entrantes desde los sensores adaptados al dispositivo iCrCThings y genera nuevos datos con instrucciones para los dispositivos de red y móviles que dispongan el software. Basado en los requerimientos iniciales. Se desarrolló en la plataforma libre con la que cuentan estos módulos Arduino, programación en

lenguaje C. Tiene la capacidad de conectarse a las bases de datos en MySQL por medio de driver JDBC.

Como se observa en la Figura 23. Se cuenta con una base de datos que dispone de dos tablas: Usuario y sensores. Cada una almacena los respectivos datos enviados por el módulo iCrCThings. Al mismo tiempo cuando la temperatura o parámetro a evaluar se encuentra fuera del rango configurado por el usuario, se envía una notificación Push al dispositivo móvil.

**Figura 23 : Organización de la BD**

ICrCThings MySQL

Usuarios

Sensores

#	Nombre	Tipo de dat...	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	id	INT	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	user	INT	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	passwd	VARCHAR	50	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
4	tipo	INT	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

#	Nombre	Tipo de dat...	Longitud/Co...	Sin signo	Permitir...	Relle...	Predeterminado
1	id	DOUBLE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	temperatura	VARCHAR	45	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
3	humedad	VARCHAR	45	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
4	presion	VARCHAR	45	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
5	fecha	DATE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL
6	hora	TIME		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	NULL

Fuente: (Calderon Calderon, 2015)

La configuración específica de cada componente instalado en el Arduino Mega22560 utilizo las siguientes librerías para su correcto funcionamiento.

```
#include <SPI.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <DHT.h>
#include <Temboo.h>
#include "TembooAccount.h"
#include <LiquidCrystal.h>
```

- **Librería Temboo y su API Parse PushNotifications.**

Es uso de librerías nos permite realizar acciones complejas de una manera más sencilla. Una API es una "llave de acceso" a funciones que nos permiten hacer uso de un servicio web provisto por un tercero, dentro de una aplicación web propia, de manera segura. Por tanto la API Parse PushNotifications de la Librería Temboo fue desarrollada para interactuar de manera acorde con dispositivos de hardware libre (Arduino). Siendo de gran ayuda para cumplir los objetivos y requerimientos planteados en este proyecto.

## **B. Software Plataforma Web**

Las Figuras 24, 25 Y 26 corresponden a la plataforma web desarrollada para interactuar el usuario con módulo iCrCThings y los datos almacenados en la BD. Fue desarrollada en la plataforma de NetBeans 8.0.2, en La tecnología Java para la creación de páginas web con programación en el servidor (JSP) *acrónimo de Java Server Pages*.

- **Librería Gson**

Gson es una librería de Java que se puede utilizar para convertir objetos Java en su representación JSON. También se puede utilizar para convertir una cadena JSON a un objeto Java equivalente. Gson puede trabajar con objetos

Java arbitrarios incluyendo objetos preexistentes que usted no tiene dentro del código fuente. (google-gson, 2015).

- **Librería Highcharts**

Highcharts JS una librería de javascript con la cual se puede implementar multitud de opciones para estadísticas, desde típicas graficas de barras con registro histórico a complejas composiciones para detallar valores. Entre la multitud de graficas encontramos:

- Line & Scatter Charts
- Área Charts
- Column & Bar Charts
- Pie Charts
- Dynamic Charts
- Combination Charts
- Angular gauge
- Clock
- Rango de Áreas...

**Figura 24: Login y registro plataforma web**



Fuente: (Calderon Calderon, 2015)

**Figura 25 : Plataforma web con datos cargados**

<a href="#">Grafica</a> <a href="#">Administrar Inventario</a> <a href="#">Administrar Usuarios</a> <a href="#">Administrar Sensores</a> <a href="#">Salir</a>						
Seleccion	id	Temperatura °C	Humedad %	Presion kgf/cm2	Fecha	Hora
<input type="radio"/>	16459.0	27.60	72.60	0	2015-05-28	16:48:19
<input type="radio"/>	16458.0	27.30	72.90	0	2015-05-28	16:48:12
<input type="radio"/>	16457.0	27.00	73.30	0	2015-05-28	16:48:05
<input type="radio"/>	16456.0	26.70	73.30	0	2015-05-28	16:47:58
<input type="radio"/>	16455.0	26.30	71.10	0	2015-05-28	16:47:52
<input type="radio"/>	16454.0	26.10	68.20	0	2015-05-28	16:47:45
<input type="radio"/>	16453.0	26.10	66.00	0	2015-05-28	16:47:38
<input type="radio"/>	16452.0	26.10	66.00	0	2015-05-28	16:47:32
<input type="radio"/>	16451.0	26.10	66.00	0	2015-05-28	16:47:24
<input type="radio"/>	16450.0	26.10	66.00	0	2015-05-28	16:47:18
<input type="radio"/>	16449.0	26.10	66.00	0	2015-05-28	16:47:11
<input type="radio"/>	16448.0	26.10	65.90	0	2015-05-28	16:47:04

Fuente: (Calderon Calderon, 2015)

**Figura 26 : Plataforma web con graficas cargadas**



Fuente: (Calderon Calderon, 2015)

### **C. Software Plataforma móvil.**

Las Figuras 27, 28 corresponden a la plataforma móvil desarrollada para interactuar el usuario final en su dispositivo móvil con el módulo iCrCThings y los datos almacenados en la BD. Fue desarrollada en la plataforma de Android Studio v1.2.0, diseñada específicamente para sistema operativo Android, bajo el lenguaje de programación java orientada a objetos. Siendo el lenguaje predilecto para aplicaciones Android.

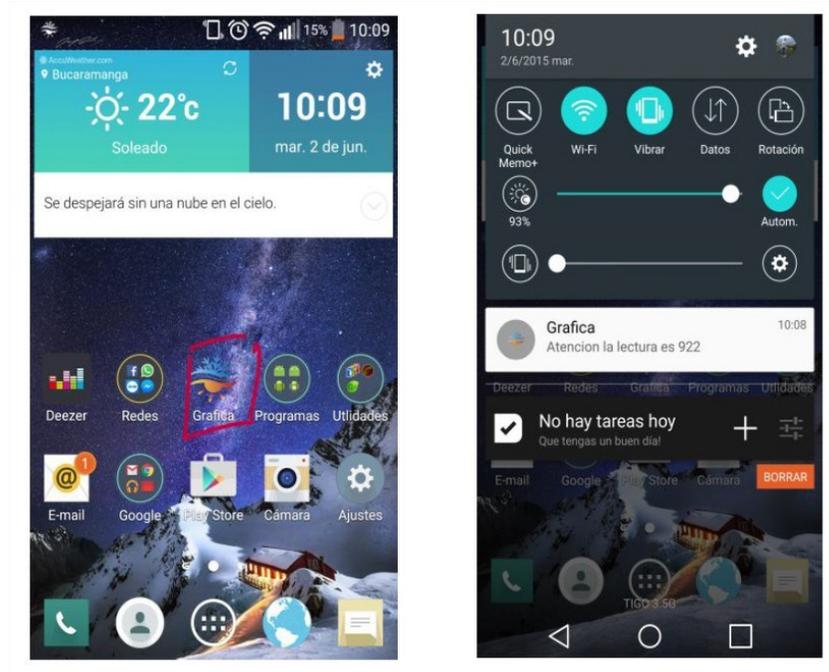
- **Librería Parse.**

Servicio web que permite enviar notificaciones Push entre muchos otros servicios y utilidades. Soporta dispositivos Android, IOS, Windows Phone etc. Parse cuenta con un plan gratuito de hasta un millón de notificaciones al mes. La utilización de esta librería es más fácil que Google Cloud Messaging plataforma oficial. Además en cuestión de minutos puede ser configurada.

- **API AndroidPlot.**

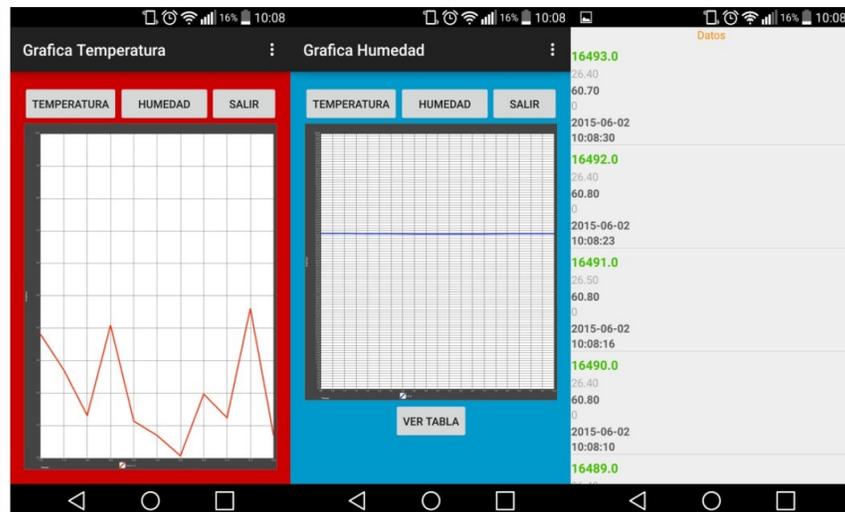
AndroidPlot es una API para la creación de gráficos dinámicos y estáticos dentro de su aplicación Android. Está diseñado desde cero para Android, compatible con todas las versiones de Android desde 1.6 en adelante y es utilizado por más de 500 aplicaciones en Google Play. (androidplot, 2015)

**Figura 27 : Aplicación Móvil iCrCThings**



Fuente: (Calderon Calderon, 2015)

**Figura 28 : Graficas Móvil app iCrCThings**



Fuente: (Calderon Calderon, 2015)

Para poder acceder tanto en plataforma web como al aplicativo móvil el usuario previamente debe estar registrado.

### **9.2.6 Resultados y discusión del objetivo.**

En la Figura 29 se visualiza el prototipo funcional. El cual pasó por diversas pruebas comprobando el correcto funcionamiento de los sensores, conexión a internet y notificación push antes de ser llevado a pruebas de campo.

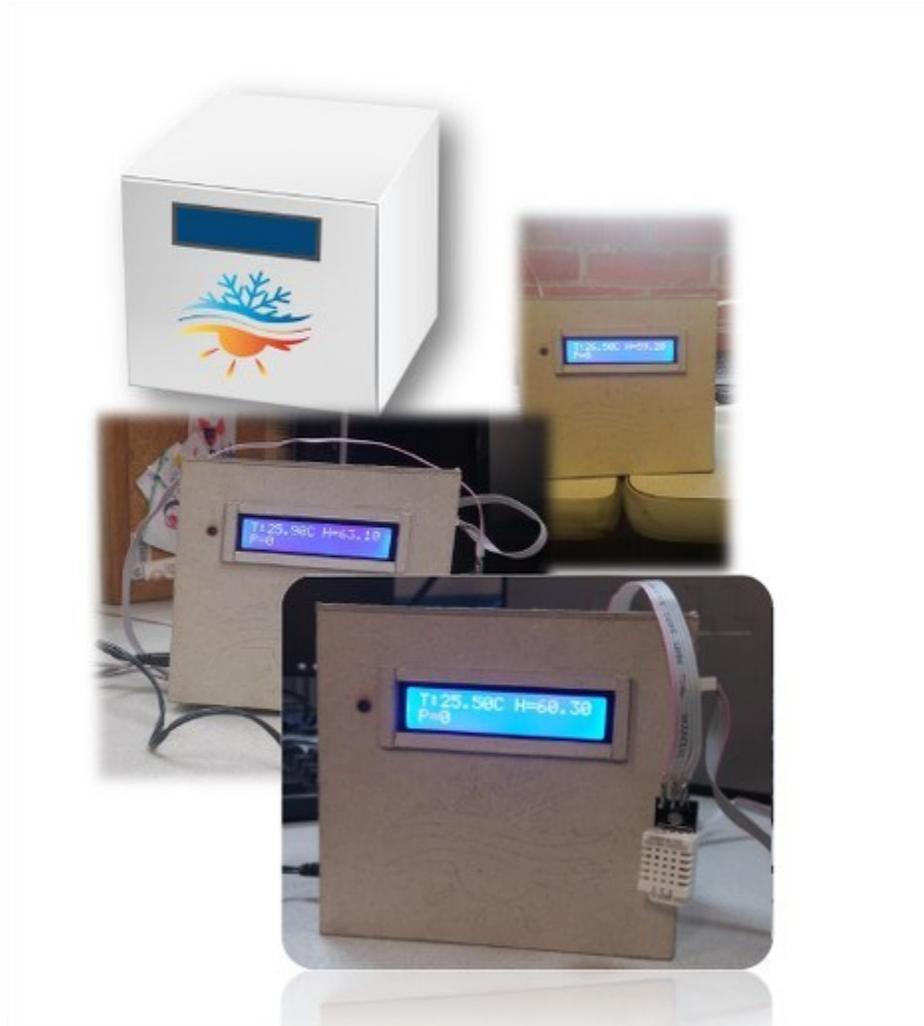
#### **A. Calibración y ajustes**

Los sensores previamente están calibrados en los laboratorios del fabricante lo que nos brindó confiabilidad en la toma de datos. El ajuste del factor máximo depende de la zona de confort que se esté evaluando.

Los sensores de humedad y temperatura respondieron satisfactoriamente a diferentes pruebas con un margen de error del 0.02 %.

El voltaje de entrada que soporta el prototipo tuvo que ajustarse con un regulador de voltaje dado que los Ampere (Unidad de intensidad de corriente eléctrica). Provocaban ruido en la señal al momento de enviar el dato por internet.

**Figura 29 : Prototipo iCrCThings**



Fuente: (Calderon Calderon, 2015)

### **9.3 IMPLEMENTAR EL PROTOTIPO DE SISTEMA INFORMÁTICO PARA LA TOMA DE INFORMACIÓN AMBIENTAL EN ESPACIO CERRADO Y REALIZAR PRUEBAS FUNCIONALES EN LA FIS.**

En la siguiente sección de resultados se explicara detalladamente el proceso de recolección, análisis y toma de decisiones con los datos recolectados en el escenario propuesto. Facultad de Ingeniería de Sistemas (UNAB).

#### **9.3.1 Pruebas**

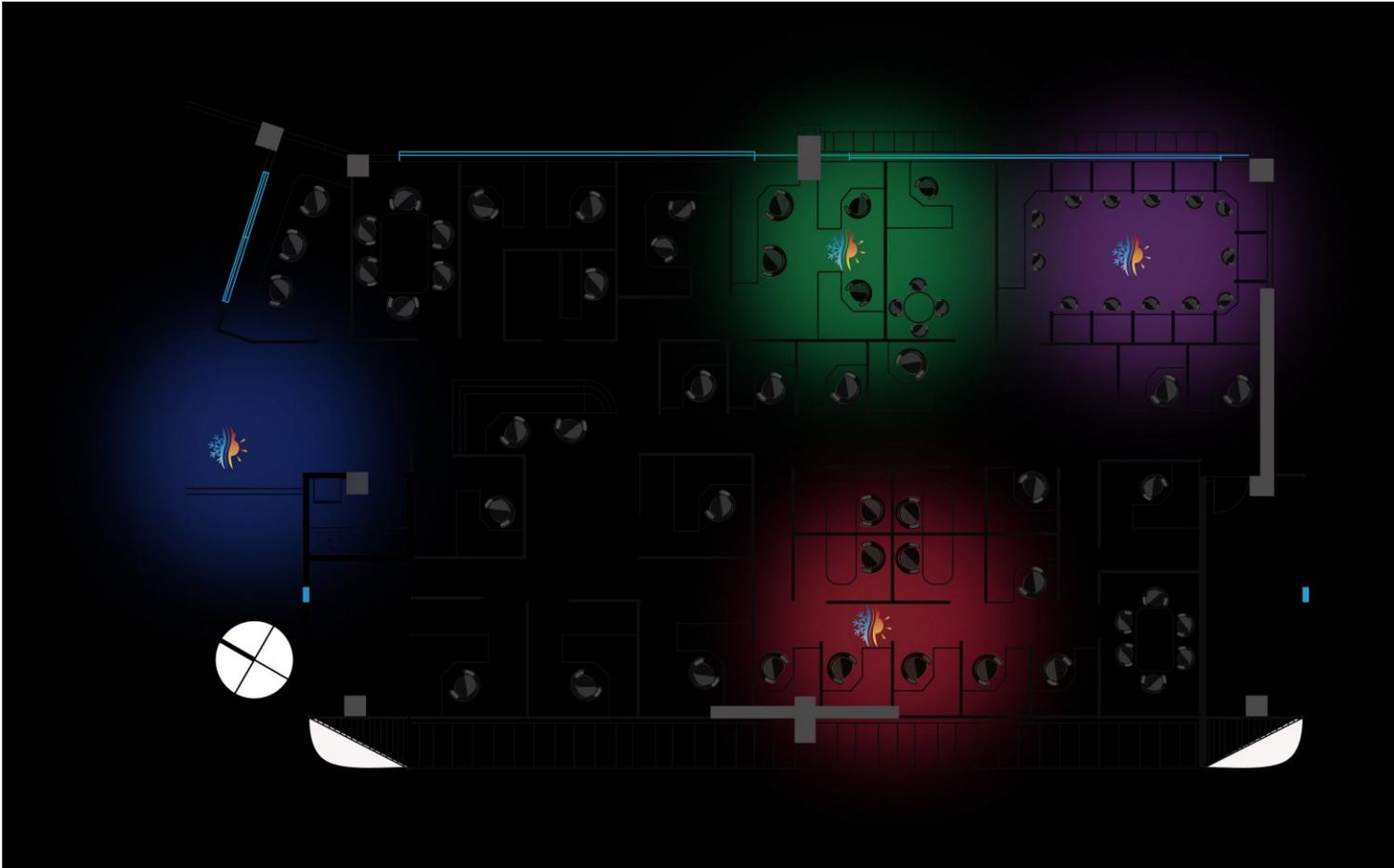
##### **A. Test de sensores.**

Los DTH22 (Pequeños dispositivos que nos permiten medir temperatura y humedad). Estos sensores están compuestos de un sensor capacitivo para medir humedad y de un termistor. Estos sensores digitales están calibrados en laboratorios presentando fiabilidad. No es necesario añadir ningún componente adicional al circuito. Caso contrario sucede con los sensores analógicos.

##### **B. Ubicación de sensores en el escenario propuesto.**

El proceso de recolección de datos conto con el despliegue del prototipo y un servidor dentro de las instalaciones de la Universidad Autónoma de Bucaramanga (UNAB).Faculta de Ingeniería de Sistemas, 2° piso-bloque L. Campus principal. La Figura 30 muestra el plano del piso donde se llevó a cabo la toma de datos. El prototipo se desplego en cuatro ubicaciones llamados nodos identificados con un color respectivo. El seguimiento de los nodos se realizó atreves de su plataforma web y móvil que se encuentra dentro del servidor previamente configurado.

Figura 30: Plano piso 2° Facultad Ingeniería Sistemas (UNAB)



Fuente: (Calderon Calderon, Plano piso 2° Bloque L Facultad de Ingenieria sistemas (UNAB), 2015)

Tabla 14 : Clasificación nodos según color	
	Centro de estudios
	Entrada Facultad
	Oficina director proyecto grado
	Oficinas Docentes Matemáticas

### C. Topología de red del escenario.

El escenario cuenta con una infraestructura de red previamente configurada.

En esta sección se simula cierta parte de la red de la Universidad Autónoma de Bucaramanga (UNAB) Figura 31 y se muestra la configuración del servidor con el nodo iCrCThings. Además se adjunta el resultado de estabilidad del ping Tabla 15 aplicado desde el servidor a cada ubicación del nodo a prueba.

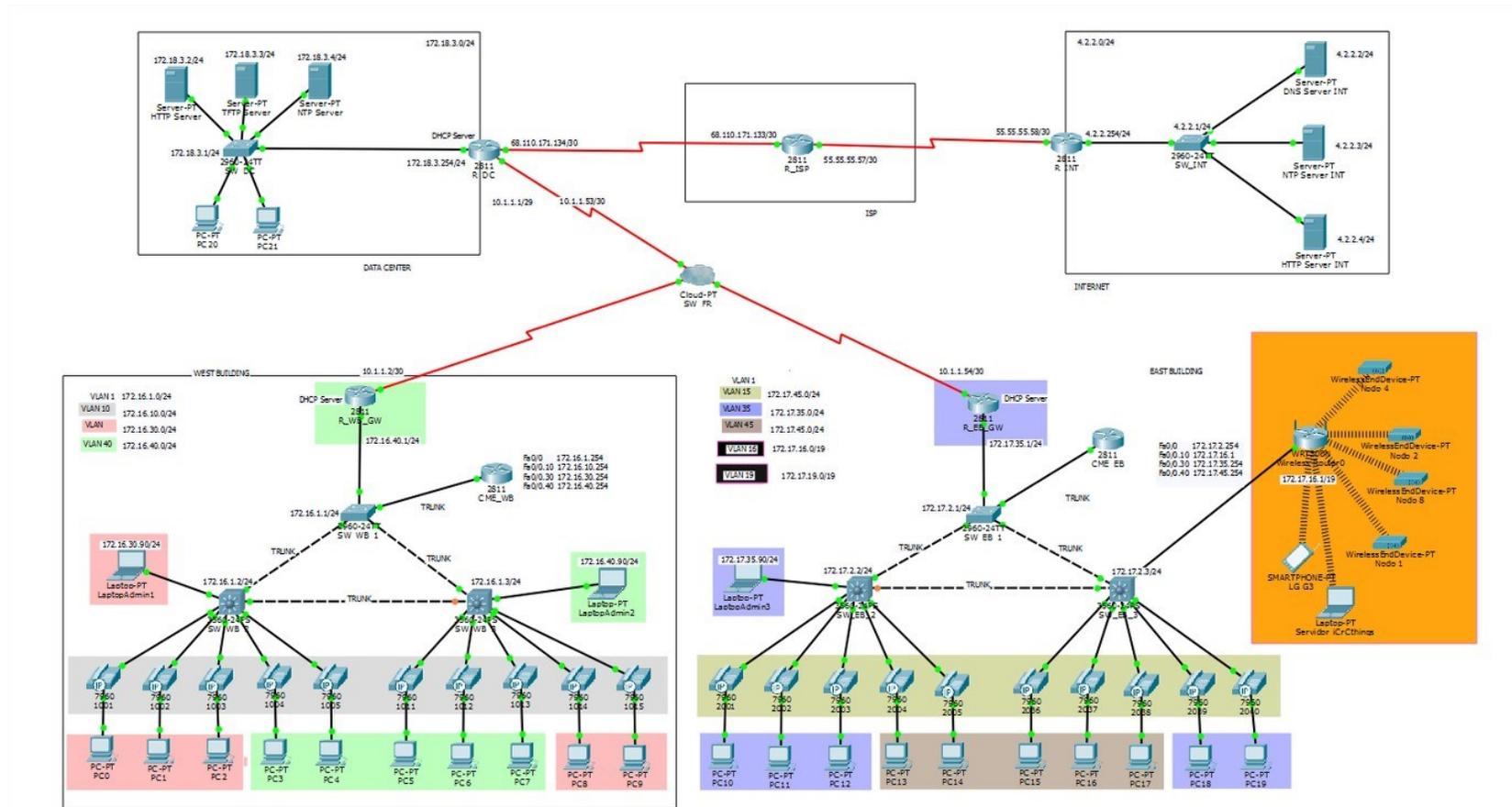
**Tabla 15 : Resultado estabilidad Ping servidor-Nodo iCrCThings**

Resultado estabilidad Ping servidor-Nodo iCrCThings							
Servidor IP		Mascara Subred		Dirección IP iCrCThings		Mascara Subred	
172.17.16.59		255.255.224.0 / 19		172.17.19.224		255.255.224.0 / 19	
# Sensor	Nodo	Delay Mínimo	Delay Máximo	Delay Media	Paquetes enviados	Recibidos	Perdidos
Nodo 1		1 ms	256 ms	12 ms	1061	1039	22
Nodo 2		1 ms	324 ms	11 ms	1277	1276	1
Nodo 4		2 ms	478 ms	13 ms	1129	1129	0
Nodo 8		2 ms	291 ms	11 ms	648	647	1

Fuente: (Calderon Calderon, Estabilidad servidor-Nodo, 2015)

- Topología de red UNAB simulada.

Figura 31 : Topología de red UNAB simulada

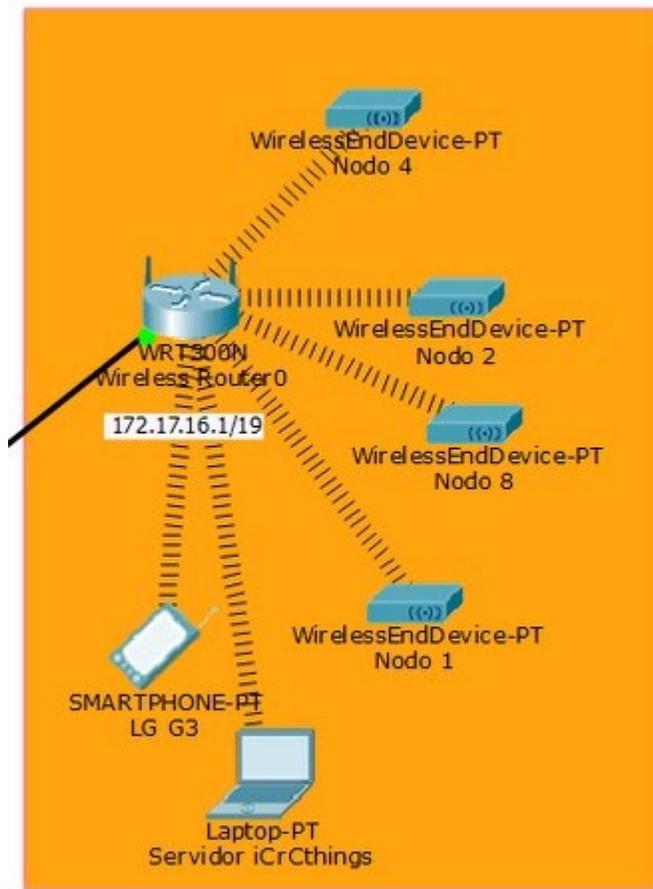


Fuente: (Calderon Calderon, Topología de red UNAB , 2015)

- **Topología de red escenario servidor –Nodo. Piso 2° Facultad Ingeniería de sistemas.**

La configuración del servidor y nodo se realizó dentro de la red previamente configura Figura 31. Direccionamiento IP Figura 32, Tabla 15.

**Figura 32 : Topología de red servidor –nodo**



Fuente: (Calderon Calderon, Topología de red UNAB , 2015)

### 9.3.2 Resultados y discusión del objetivo.

A continuación se muestra la prueba de campo para el monitoreo de temperatura entre 8 am a 5 pm. Estas mediciones se realizan en un día parcialmente nublado con temperaturas 21 °C iniciando la toma de datos y finalizando la muestra 24 °C. Con sensación térmica de 26 grados y 88 % de humedad. El sistema recoge las lecturas de los nodos en dos diferentes etapas (Mañana y Tarde), cada una comprende un tiempo promedio de 20 minutos. En este experimento se toman medidas en diferentes lugares del piso 2° y piso 7° para nuestro estudio se tienen en cuenta la toma de 4 nodos principales, 2 nodos ubicados en los exteriores del piso 2° y 1 nodo ubicado en el piso 7° zona totalmente diferente al estudio, con la cual se busca comparar otro ambiente de confort. En las figuras y tablas que se muestran a continuación es posible observar cómo se comporta en forma general el piso 2° frente a la alternancia de temperaturas en el transcurso del día. En este proceso, al igual que descende la temperatura exterior (máx. 24.8 °C), nodo 3 anexo Excel al documento. Luego una serie de mediciones (máx. 31.3 °C), nodo 7 anexo Excel. Los espacios interiores del piso 2° fluctúan térmicamente a la par de esas condiciones externas.

El resultado de estas mediciones se muestra en la Tabla 16. Los cuales previamente fueron almacenados en tiempo real en la base de datos alojada en el servidor Figura 33.

**Tabla 16 : Resultado temperatura y humedad relativa nodos**

Temperaturas (°C ) Nodos							
Etapa		Mañana			Tarde		
# Sensor	Nodo	T. máx.	T. min.	T. pro.	T. máx.	T. min.	T. pro.
Nodo 1		24,8	24,6	24,68	25,4	25,3	25,35
Nodo 2		24,8	24,6	24,68	26,6	26	26,45
Nodo 4		24,8	25	25,11	26,4	26	26,13
Nodo 8		26	25,6	25,75	27,6	25,9	26,20

Humedad R. (%) Nodos							
Etapa		Mañana			Tarde		
# Sensor	Nodo	HR. Máx.	HR. Min.	HR. Pro.	HR. Máx.	HR. Min.	HR. Pro.
Nodo 1		74,9	63,8	68,00	67,9	67,3	67,57
Nodo 2		69,8	65,9	68,83	65,1	62,7	63,31
Nodo 4		68,4	66,4	67,24	65,8	64,2	65,39
Nodo 8		63,6	62,8	63,26	73,3	65,6	66,80

Fuente: (Calderon Calderon, Resultado Temperatura y Humedad, 2015)

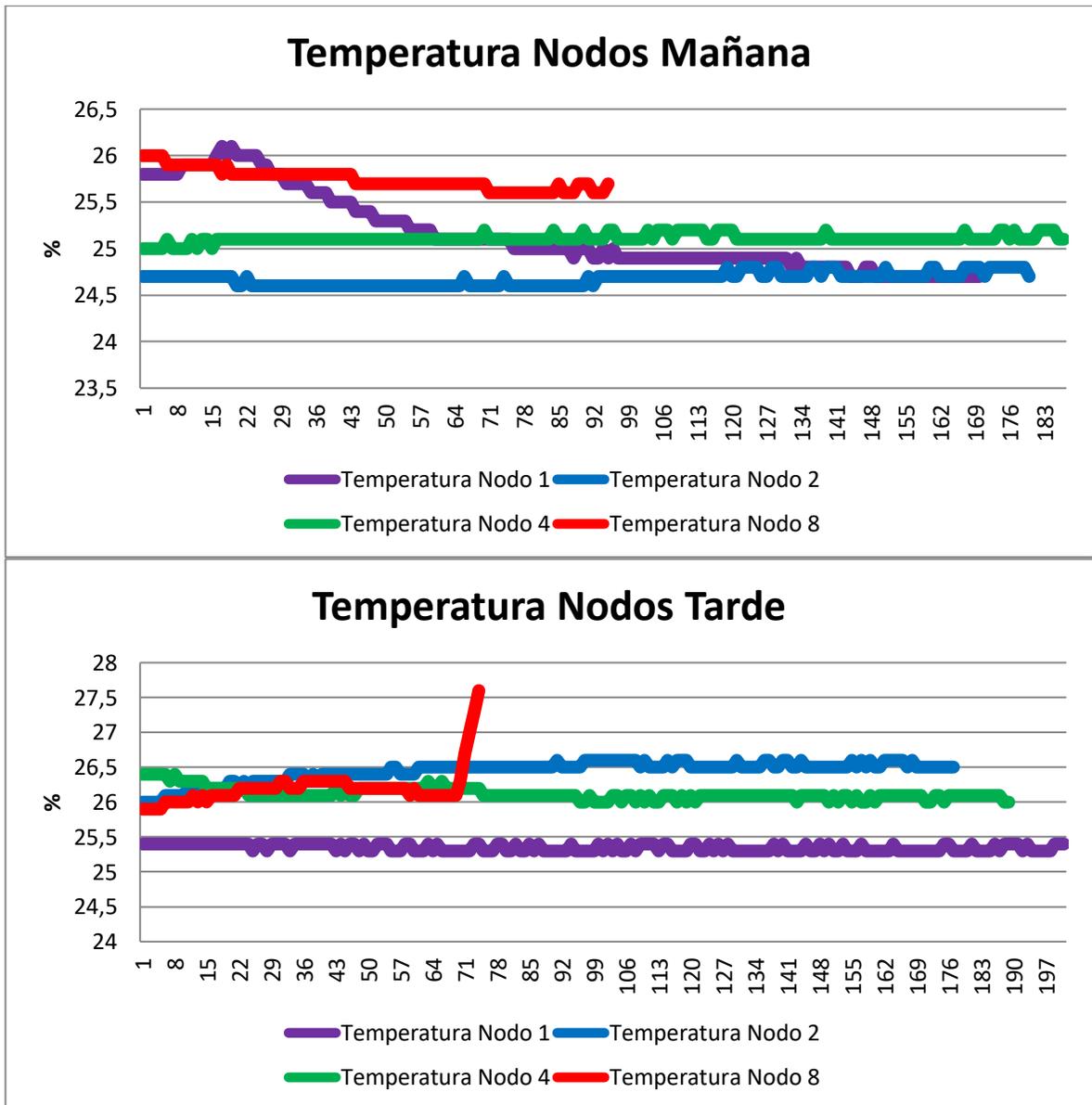
**Figura 33 : Datos alojados en BD**

id	temperatura	humedad	presion	fecha	hora
16459	27.60	72.60	0	2015-05-28	16:48:19
16458	27.30	72.90	0	2015-05-28	16:48:12
16457	27.00	73.30	0	2015-05-28	16:48:05
16456	26.70	73.30	0	2015-05-28	16:47:58
16455	26.30	71.10	0	2015-05-28	16:47:52
16454	26.10	68.20	0	2015-05-28	16:47:45
16453	26.10	66.00	0	2015-05-28	16:47:38
16452	26.10	66.00	0	2015-05-28	16:47:32
16451	26.10	66.00	0	2015-05-28	16:47:24
16450	26.10	66.00	0	2015-05-28	16:47:18
16449	26.10	66.00	0	2015-05-28	16:47:11
16448	26.10	65.90	0	2015-05-28	16:47:04
16447	26.10	65.80	0	2015-05-28	16:46:58

Fuente: (Calderon Calderon, 2015)

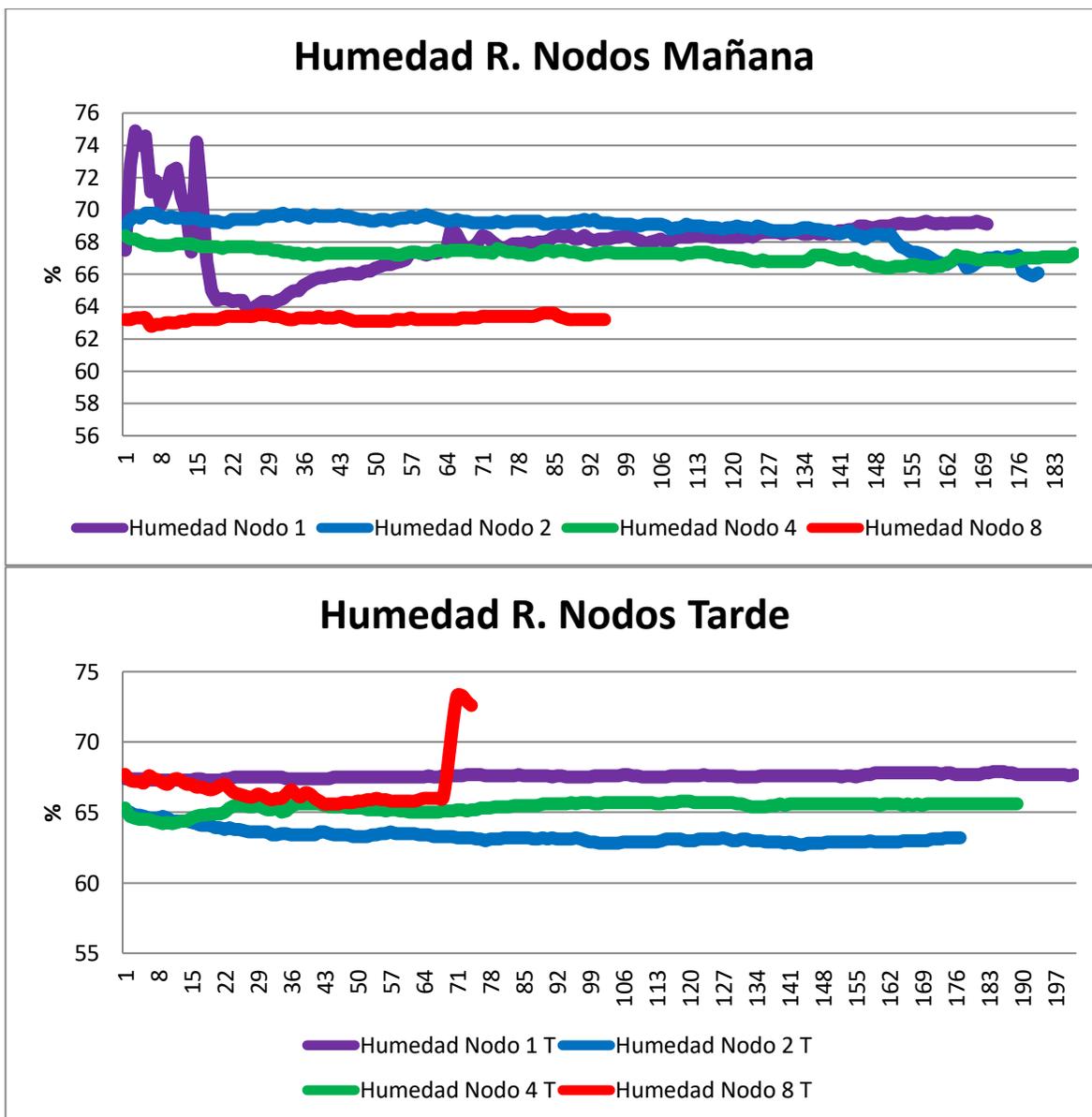
Luego del análisis de los resultados temperatura y Humedad Relativa a continuación los gráficos resultantes del experimento Figuras 34, 35, 36 y 37

**Figura 34 : Temperatura nodos mañana-tarde.**



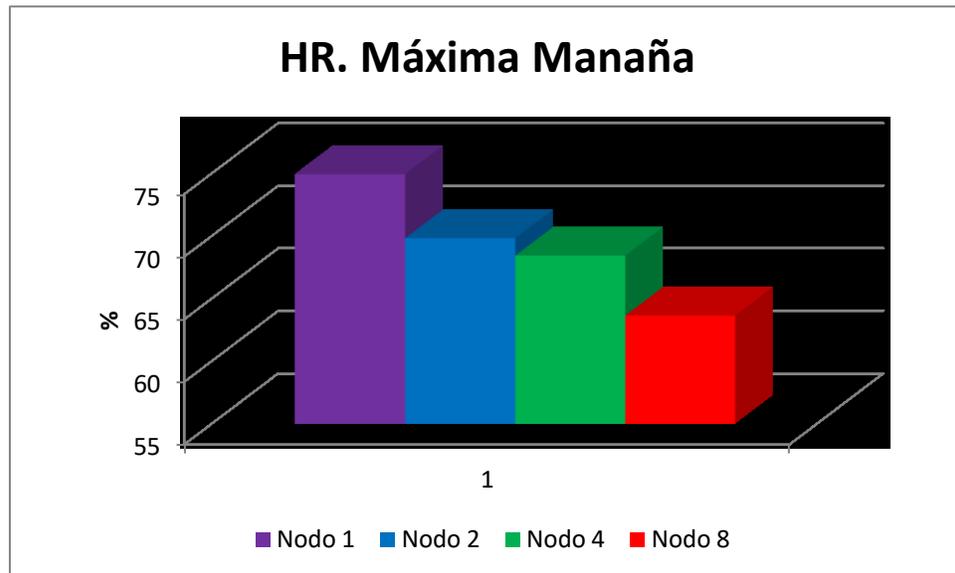
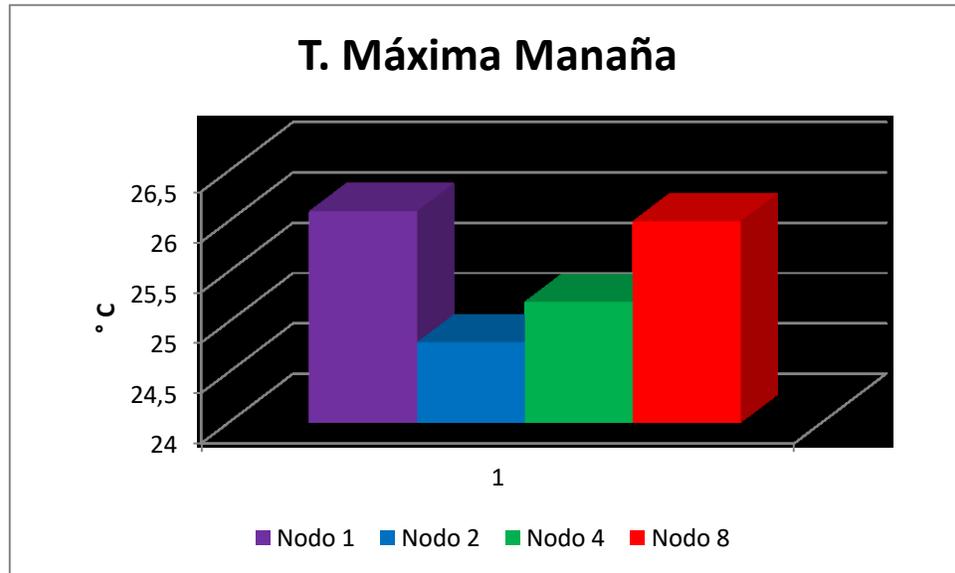
Fuente: (Calderon Calderon, 2015)

Figura 35 : Humedad relativa nodos mañana-tarde.



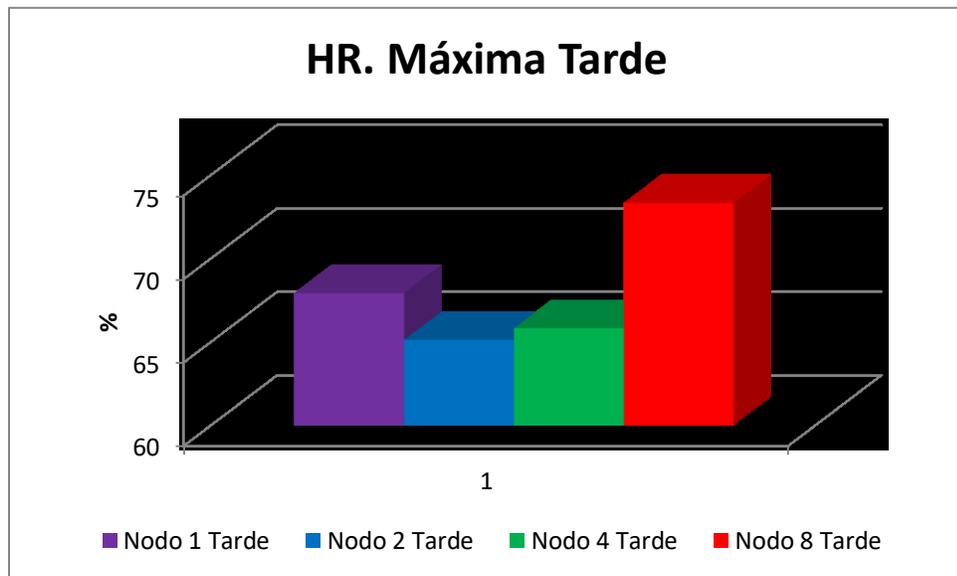
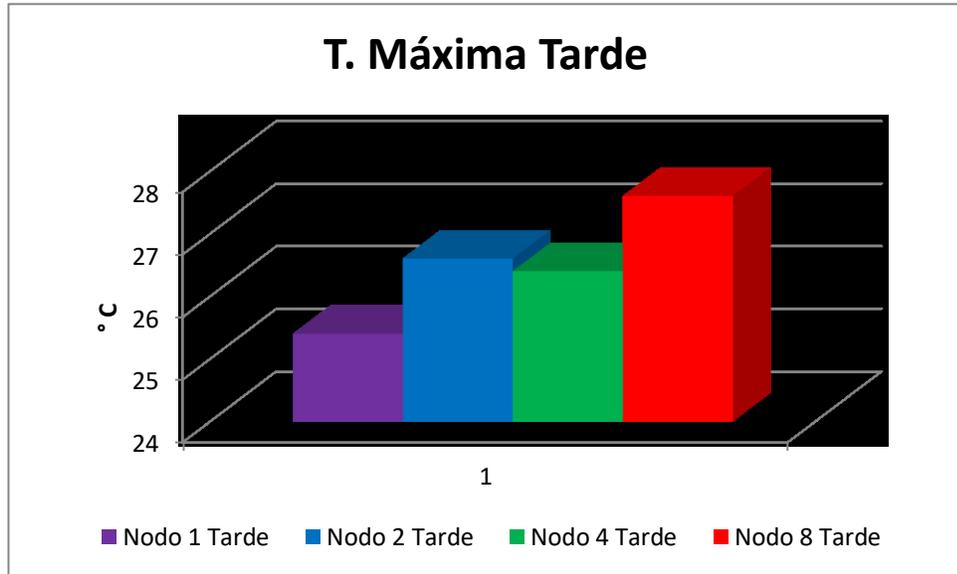
Fuente: (Calderon Calderon, 2015)

Figura 36 : Temperatura y humedad relativa máxima nodos mañana.



Fuente: (Calderon Calderon, 2015)

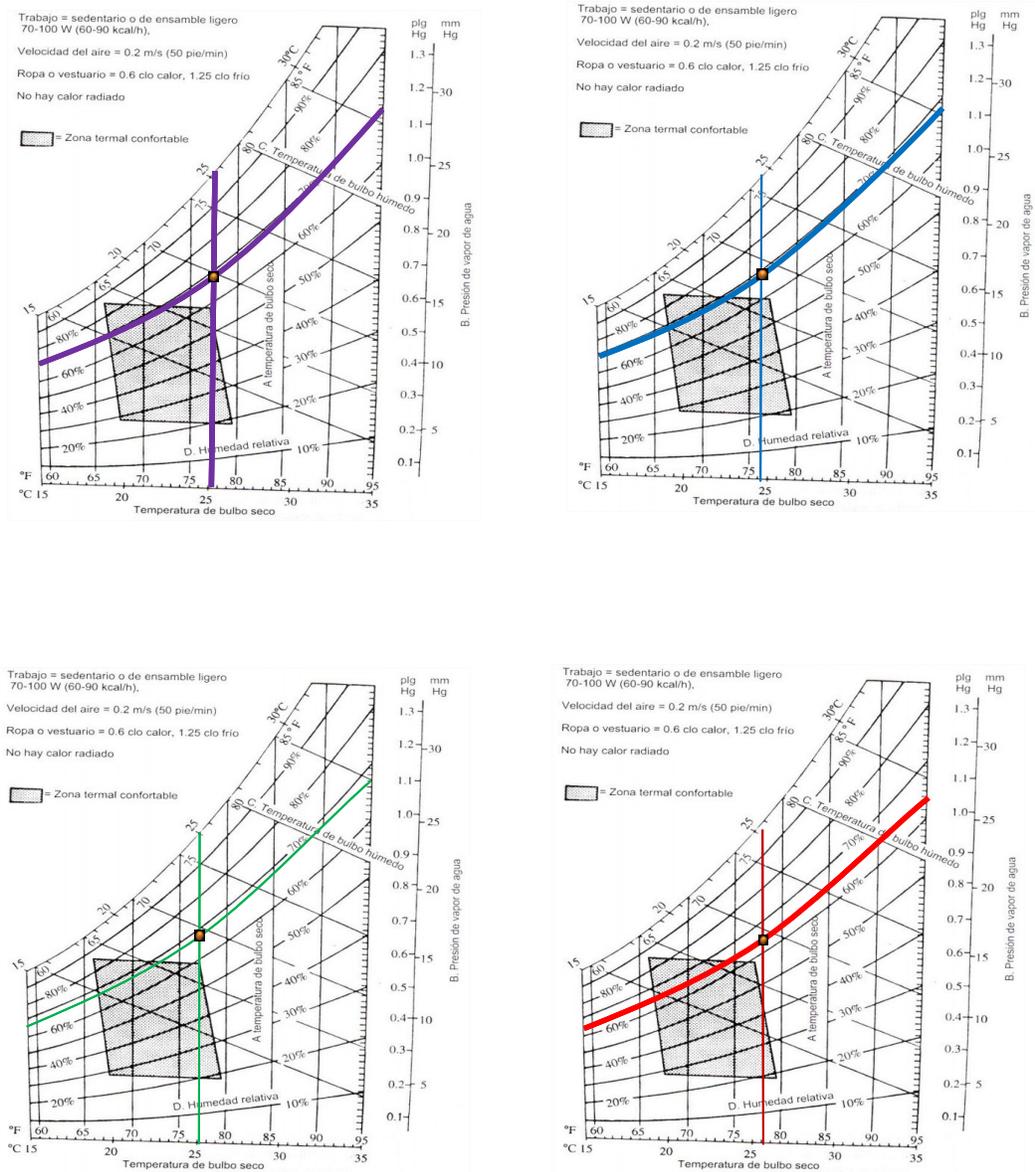
Figura 37 : Temperatura y humedad relativa máxima nodos tarde



Fuente: (Calderon Calderon, 2015)

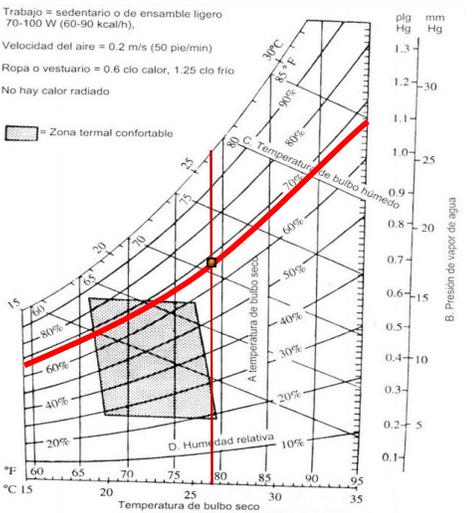
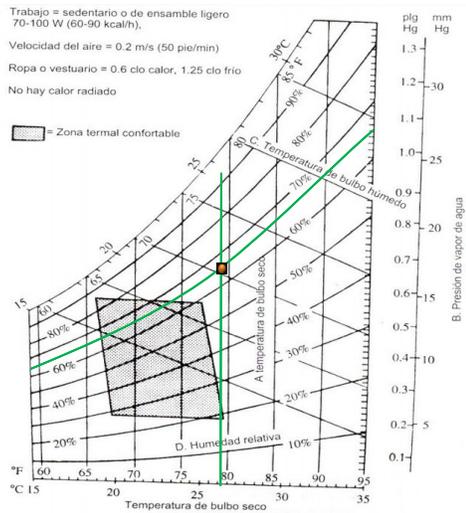
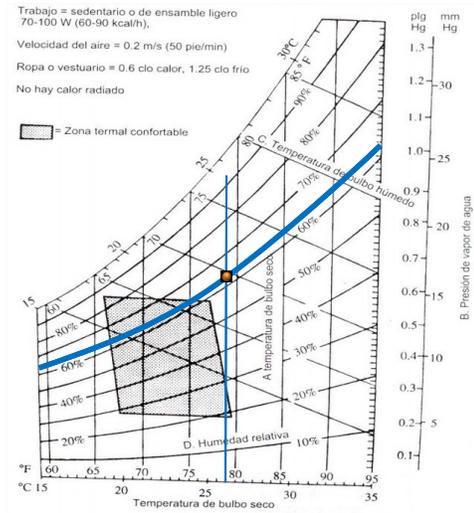
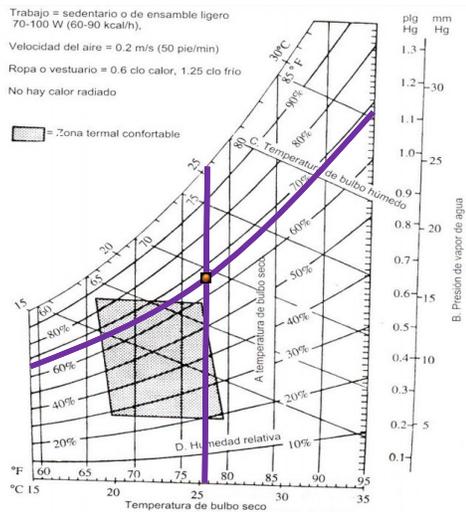
Con la interpretación de los resultados en esta sección se realiza análisis Zona de Confort Térmico Figura 20 en cada una de las etapas (Mañana-Tarde) Figuras 38,39.

**Figura 38 : Zona confort térmico nodos mañana**



Fuente: (Calderon Calderon, 2015)

**Figura 39 : Zona confort térmico nodos tarde**

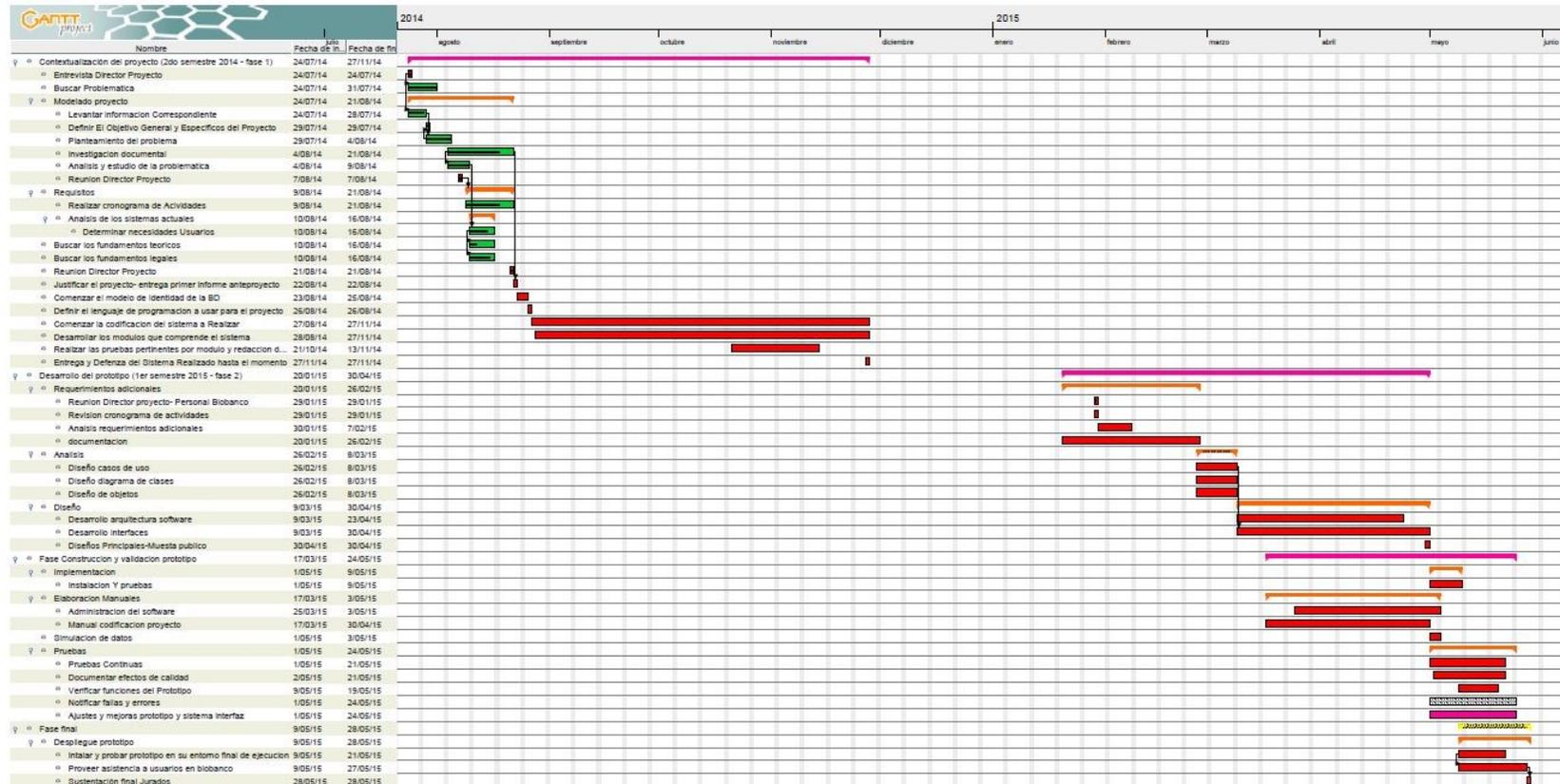


Fuente: (Calderon Calderon, 2015)

## 10. CRONOGRAMA DE ACTIVIDADES

Figura 40 : Diagrama de Gantt Cronograma actividades 2014-2015

# Diagrama de Gantt



Fuente: (Calderon, Diagrama de GANTT, 2014)

## 11. CONCLUSIONES

- Se diseñó y desarrollo un prototipo de sistema de información en tiempo real inalámbrico, Capaz de monitorear condiciones de temperaturas óptimas y generar notificaciones. Considerando que se realizó basado hardware libre, bajo costo, plataformas de visualización web y móvil. En las cuales el usuario final tiene una rápida noción de los factores climatológicos del lugar donde se encuentre, se cumplieron los objetivos y requerimientos iniciales. Esto constituye una herramienta de suma utilidad en el ámbito académico (monitoreo de condiciones óptimas, pruebas de campo etc.) o comercial (mejorar espacios donde se guarden productos de suma importancia que se necesiten monitorear constantemente).
- Tanto el software como el prototipo fueron desarrollados multipropósito y Handshaking<sup>7</sup>, lo que implica que en cualquier caso o estudio se puedan cambiar los sensores para medir otras variables (luz, gases, nivel de líquidos, etc.) siendo así las modificaciones mínimas.
- Es uso de librerías nos permite realizar acciones complejas de una manera más sencilla. Una API es una "llave de acceso" a funciones que nos permiten hacer uso de un servicio web provisto por un tercero, dentro de una aplicación web propia, de manera segura. Por tanto la API Parse PushNotifications de la Librería Temboo fue adaptada y desarrollada de manera tal que pueda interactuar con dispositivos de hardware libre

---

<sup>7</sup> Proceso automatizado de negociación que establece de forma dinámica los parámetros de un canal de comunicaciones establecido entre dos entidades antes de que comience la comunicación normal por el canal

(Arduino). Siendo de gran ayuda para cumplir los objetivos y requerimientos planteados en este proyecto.

- La implementación de tecnología de hardware libre (Arduino) facilitó el proceso de construcción del prototipo, Dada su gran variedad de componentes electrónicos que se pueden encontrar en el mercado. Los sensores fueron otro factor importante en el desarrollo del mismo ya que en nuestra investigación y estado del arte encontramos tecnologías sensoriales revolucionarias de muy bajo costo. Las cuales hoy día son auge por la gran expansión del mundo del IoT.
- El desarrollo en Android fue sencillo gracias a la forma en que fácilmente se pueden utilizar librerías de terceros, en este caso para las gráficas en tiempo real. Al ser programado en Java, el desarrollo de aplicaciones dispone de gran cantidad de documentación, tanto oficial como de terceros.
- Los resultados muestran que los espacios interiores de la facultad de ingeniería de sistemas de la Universidad Autónoma de Bucaramanga (UNAB) edificio L piso 2° no alcanzan los estándares mínimos de confort para la situación climática estudiada. Siendo uno de los principales problemas encontrados las elevadas ganancias de calor que durante el día son absorbidas por los muros del edificio y que en muy pocas ocasiones son disipadas por los mismos. La insuficiente ventilación otro factor no óptimo para lograr condiciones que se acerquen a las de confort.

## 12. TRABAJO FUTURO

A futuro se espera desarrollar las siguientes mejoras al prototipo iCrCThings y a todo el sistema que lo compone en general. Siendo este el pilar de la construcción de nuevas tecnologías del internet de las cosas.

- Disminución del tamaño del prototipo.
- Poder configurar los dispositivos (nodos) desde una plataforma remota en tiempo real.
- Ajustar el dispositivo con nuevas tecnologías de abastecimiento energético (células fotovoltaicas) y baterías de mayor durabilidad.
- Trabajar sensores inalámbricos con mayores longitudes de señal.
- Poder administrar toda clase de dispositivos electrónicos que cuenten con conexión a internet y se puedan programar para cambiar alguna funcionalidad. (Apagar y encender aires, etc).
- Con los datos capturados por el dispositivo poder alimentar BD de investigadores y así predecir cambios en el clima.
- Mejor la interfaz de usuario tanto en la plataforma web y móvil.
- Ajustar la programación de tal forma que genere reportes tipo Excel con los cuales se detalle con mayor claridad las mediciones.
- Poder contar con una mayor cantidad de dispositivos (tecnología WSN) y realizar pruebas con mayores lapsos de tiempo.

### 13. BIBLIOGRAFÍA

- ABC. (2015). *definición abc*. Recuperado el 2015 de 02 de 28, de <http://www.definicionabc.com/tecnologia/aplicacion.php>
- androidplot. (2015). *androidplot.com*. Recuperado el 25 de 05 de 2015, de <http://androidplot.com/>
- Arduino. (2015). *arduino.cc*. Recuperado el 16 de 04 de 2015, de <http://www.arduino.cc/en/ArduinoCertified/IntelGalileo>
- ARDUINO. (2015). *Intel Galileo*. Recuperado el 26 de 10 de 2014, de <http://arduino.cc/en/ArduinoCertified/IntelGalileo>
- Aslam, M.S., Rea, S. , & Pesch, D. (8 de 2012). *Service Provisioning for the WSN Cloud*. Recuperado el 25 de 03 de 2015, de <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6253607&queryText%3DWSN>
- Baghyalakshmi, D., Ebenezer, J., & SatyaMurty, S.A.V. (08 de 2011). *WSN based temperature monitoring for High Performance Computing cluster*. Recuperado el 16 de 09 de 2014, de *WSN based temperature monitoring for High Performance Computing cluster*
- bibliotecakatherinebrecht. (12 de 11 de 2011). *Proyecto Socio-Tecnológico III:Sistema de Control de Materiales Bibliográficos*. Recuperado el 18 de 08 de 2014, de Diagrama Gantt Fase 1, 2, 3 y 4 - RUP: Inicio, Elaboración, Construcción y Transición: <http://bibliotecakatherinebrecht.blogspot.com/2011/11/diagrama-gantt-fase-1-2-3-y-2-rup.html>
- Binghao Li, James Salter, Andrew G, Dempster, & v. (2006). *Indoor Positioning Techniques Based on Wireless LAN* . Recuperado el 18 de 02 de 2015, de [http://www.researchgate.net/profile/Chris\\_Rizos/publication/228998356\\_Indoor\\_positioning\\_techniques\\_based\\_on\\_wireless\\_LAN/links/0912f5089b73fef679000000.pdf](http://www.researchgate.net/profile/Chris_Rizos/publication/228998356_Indoor_positioning_techniques_based_on_wireless_LAN/links/0912f5089b73fef679000000.pdf)
- Calderon Calderon , C. R. (2015). Estado del arte. Bucaramanga, Santander, Colombia.
- Calderon Calderon, C. R. (25 de 05 de 2015). Bucaramanga, Santander, Colombia.
- Calderon Calderon, C. R. (25 de 05 de 2015). Estabilidad servidor-Nodo. Bucaramanga, Santander, Colombia.
- Calderón Calderón, C. R. (2015). Metodología arquitectura prototipo. Bucaramanga, Santander, Colombia.

Calderon Calderon, C. R. (25 de 05 de 2015). Plano piso 2° Bloque L Facultad de Ingenieria sistemas (UNAB). Bucaramanga, Santander, Colombia.

Calderon Calderon, C. R. (25 de 05 de 2015). Resultado Temperatura y Humedad. Bucaramanga, Santander, Colombia.

Calderon Calderon, C. R. (25 de 05 de 2015). Topología de red UNAB . Bucaramanga, Santander, Colombia.

Calderon, C. R. (25 de 08 de 2014). Diagrama de GANTT. Floridablanca, Colombia.

Calderon, C. R. (23 de 08 de 2014). Presupuesto Proyecto . Floridablanca, Colombia.

Calderon, C. R., & Tarazona, J. F. (05 de 05 de 2015). Diagrama general sistema de monitoreo. Bucaramanga, Santander, Colombia.

cibererbiobank. (s.f.). *ciberer-biobank*. Recuperado el 20 de 08 de 2014, de ¿Qué es un Biobanco?: [http://ciberer-biobank.es/index.php?option=com\\_content&view=article&id=1%3Aique-es-un-biobanco&catid=36%3Apreguntas-frecuentes&Itemid=41&lang=es](http://ciberer-biobank.es/index.php?option=com_content&view=article&id=1%3Aique-es-un-biobanco&catid=36%3Apreguntas-frecuentes&Itemid=41&lang=es)

clinicbiobanc. (s.f.). *El Biobanco del Hospital Clínic - IDIBAPS*. Recuperado el 22 de 08 de 2014, de El Biobanco es un banco de bancos: [http://www.clinicbiobanc.org/qui-som/sobre-el-biobanc/es\\_index.html](http://www.clinicbiobanc.org/qui-som/sobre-el-biobanc/es_index.html)

Colombia, F. S. (2008). *La Temperatura es un Factor Determinante en el Transporte de Mercancia Refrigerad*. Recuperado el 18 de 08 de 2014, de SATRACK: <http://www.frimac.com.co/nota.php?idd=1>

computacionnube. (8 de 08 de 2010). *Tipos de Nube*. Recuperado el 25 de 08 de 2014, de <http://www.computacionnube.org/13/tipos-de-nube/>

deepakcs. (19 de 06 de 2013). *ADF Mobile Push Notifications With Google Cloud Messaging (GCM) Part 1* . Recuperado el 26 de 08 de 2014, de <http://deepakcs.blogspot.com/2013/06/adf-mobile-push-notifications-with.html>

definicion.de. (2012). *WEB*. Recuperado el 22 de 08 de 2014, de <http://definicion.de/web/>

desdelinux.net. (2010). *DesdeLinux*. Recuperado el 26 de 10 de 2014, de <http://blog.desdelinux.net/un-ejemplo-de-hardware-libre-arduino/>

ecured. (18 de 07 de 2014). *Cornelius Drebbel*. Recuperado el 21 de 08 de 2014, de Cornelius Drebbel

EcuRed.cu. (2014). *Hardware libre*. Recuperado el 10 de 26 de 2014, de [http://www.ecured.cu/index.php/Hardware\\_libre](http://www.ecured.cu/index.php/Hardware_libre)

- Electron., C. &. (2009). Technologies for wireless data acquisition and monitoring of environmental parameters. 1 - 5.
- Electronics, O. (03 de 05 de 2013). *Parametros DTH22*. Recuperado el 25 de 05 de 2015, de <http://www.opiron.com/portfolio/todos-sobre-los-sensores-dht11-dht22-by-opiron-2/>
- Forero Vargas, D., & Jaraba Garcia, R. A. (2008). Validación de una red de sensores inalámbricos para el control de la calidad del aire en la zona industrial de Bucaramanga. Bucaramanga, Santander, Colombia.
- galeon.com. (2013). *ARQUITECTURAS CLIENTE/SERVIDOR*. Recuperado el 16 de 04 de 2015, de <http://bdjulian.galeon.com/aficiones1783434.html>
- Giordano Avellaneda, R. N. (27 de 06 de 2012). *Scielo*. Recuperado el 19 de 08 de 2014, de Giordano Avellaneda\*, Ricardo Noguera\*, Edgar Muñoz: [http://www.scielo.cl/scielo.php?script=sci\\_arttext&pid=S0718-50732012000300004](http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-50732012000300004)
- gnu.org. (05 de 08 de 2014). *El sistema operativo GNU*. Recuperado el 19 de 08 de 2014, de El sistema operativo GNU: <http://www.gnu.org/philosophy/free-sw.es.html>
- González, I., González, J., & Gómez, F. (6 de 09 de 2003). *Hardware libre: clasificación y desarrollo de hardware*. Recuperado el 16 de 08 de 2014, de <http://es.tldp.org/Presentaciones/200309hispalinux/8/8.pdf>
- google-gson. (2015). *Gson*. Recuperado el 25 de 05 de 2015, de <https://code.google.com/p/google-gson/>
- HelloPRO. (2014). Recuperado el 25 de 03 de 2015, de [https://www.google.com.co/search?q=espacio+cerrado&espv=2&biw=947&bih=605&source=lnms&tbm=isch&sa=X&ei=Jk5jVcDNOsevggTQ94CICQ&ved=0CAYQ\\_AUoAQ#tbm=isch&q=espacio+confinado+interiores&imgdii=JZZDSrZZNzDkRM%3A%3BJZZDSrZZNzDkRM%3A%3Bifj87ly3-7sDGM%3A&imgsrc=J2](https://www.google.com.co/search?q=espacio+cerrado&espv=2&biw=947&bih=605&source=lnms&tbm=isch&sa=X&ei=Jk5jVcDNOsevggTQ94CICQ&ved=0CAYQ_AUoAQ#tbm=isch&q=espacio+confinado+interiores&imgdii=JZZDSrZZNzDkRM%3A%3BJZZDSrZZNzDkRM%3A%3Bifj87ly3-7sDGM%3A&imgsrc=J2)
- <http://androidos.readthedocs.org/>. (2012). *Android OS*. Recuperado el 28 de 03 de 2015, de <http://androidos.readthedocs.org/en/latest/data/historia/>
- IDC. (05 de 2015). *Smartphone OS Market Share, Q1 2015*. Recuperado el 25 de 05 de 2015, de <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- Industry, b. (13 de 08 de 2014). *biomerieux*. Recuperado el 13 de 08 de 2018, de Aplicaciones Alimentarias - Labguard: [http://www.biomerieux.com.mx/servlet/srt/bio/mexico/dynPage?doc=MEX\\_IND\\_FDA\\_PRD\\_G\\_PRD\\_NDY\\_73](http://www.biomerieux.com.mx/servlet/srt/bio/mexico/dynPage?doc=MEX_IND_FDA_PRD_G_PRD_NDY_73)

- Jie Yin, Qiang Yang , & Lionel Ni. (12 de 03 de 2005). *Adaptive Temporal Radio Maps for Indoor Location Estimation*. Recuperado el 16 de 02 de 2015, de <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1392745&tag=1>
- Joshua A, T. (07 de 2002). Recuperado el 16 de 02 de 2015, de <http://people.csail.mit.edu/josh/papers/location.pdf>
- Lita, I., Electron., C. &, Cioc, I., Visan, D., & Dorobantu, I. (17 de 5 de 2009). *Technologies for wireless data acquisition and monitoring of environmental parameters*. Recuperado el 20 de 02 de 2015, de <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5206980&queryText%3DEnvironmental+parameters+monitoring>
- Lopez Cortes, C. A., Rodriguez, I. C., & Barba, R. B. (2006). sistema de monitoreo mediante cámaras inalámbricas para vigilancia. Bucaramanga, Santander, Floridablanca.
- MACKOWIAK, P. A. (2014). *Temperature Regulation and*. Recuperado el 16 de 04 de 2015, de [https://xa.yimg.com/kq/groups/23515872/1390971389/name/FIEBRE\\_Mandell.7.pdf](https://xa.yimg.com/kq/groups/23515872/1390971389/name/FIEBRE_Mandell.7.pdf)
- measureinstruments. (2001-2014). *MEASURE INSTRUMENTS - Buenos Aires – ARGENTINA*. Recuperado el 18 de 08 de 2014, de [http://www.measureinstruments.com.ar/Temperatura\\_Silos.html](http://www.measureinstruments.com.ar/Temperatura_Silos.html)
- MobiLens, c. (09 de 04 de 2015). Recuperado el 26 de 05 de 2015, de <http://www.comscore.com/lat/Insights/Market-Rankings/comScore-Reports-February-2015-US-Smartphone-Subscriber-Market-Share>
- Moreira, N, Venda, M. , Silva, C, & Marcelino, L. . (18 de 08 de 2011). *Sensor - Mobile application to monitor a WSN*. Recuperado el 16 de 08 de 2014, de <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5974264&queryText%3DWSN>
- Niebel, & Benjamín, W. (s.f.). *Tiempos y Movimientos*. Alfaomega.
- Ortal, A. C. (2014). *enpositivo.com*. Recuperado el 18 de 8 de 2014, de <http://enpositivo.com>
- PCWorld. (17 de 01 de 2012). Recuperado el 28 de 03 de 2015, de *Sistemas Operativos Móviles: Comunicación en tiempo real* : <http://www.pcworld.com.mx/Articulos/20734.htm#>
- Quees.la. (2014). *Qué es telefonía celular*. Recuperado el 16 de 02 de 2015, de <http://quees.la/telefonía-celular/>
- quien.net. (2012). *Biografía de Cornelius Drebbel - Quién fue*. Recuperado el 21 de 08 de 2014, de <http://www.quien.net/cornelius-drebbel.php>

- quiminet. (10 de 6 de 2005). *quiminet.com*. Recuperado el 21 de 08 de 2014, de Indicadores:  
<http://www.quiminet.com/articulos/que-son-los-indicadores-para-calibracion-en-temperatura-776.htm>
- quiminet. (29 de 9 de 2011). *quiminet.com*. Recuperado el 19 de 08 de 2014, de ¿Qué es la  
ultracongelación?: <http://www.quiminet.com/articulos/conserva-sus-muestras-biologicas-con-la-ultracongelacion-2576208.htm>
- raspberrypi.org. (2015). *Raspberrypi*. Recuperado el 16 de 04 de 2015, de  
<https://www.raspberrypi.org>
- raspberrypi.es. (2011-2014). *Raspberry Pi*. Recuperado el 26 de 10 de 2014, de  
<http://www.raspberrypi.es>
- Riesgolaboral.net. (2014). *Ergonomia AMBIENTAL*. Recuperado el 16 de 04 de 2015, de  
<http://slideplayer.es/slide/156495/>
- Rossi, L., Departamento de Electr. Eng., U. d., Krishnamachari, B., & Kuo, C.-C. (2004). Hybrid data  
and decision fusion techniques for model-based data gathering in wireless sensor  
networks [environmental monitoring applications]. Vehicular Technology Conference,  
2004. VTC2004-Fall. 2004 IEEE 60th (Volume:7 ).
- Sánchez, J. R. (s.f.). *lungnome*. Recuperado el 20 de 08 de 2014, de ¿Qué es un Biobanco?:  
<http://www.lungnome.org/es/interes.htm>
- sgoliver. (2014). *Notificaciones Push Android: Google Cloud Messaging (GCM). Introducción*.  
Recuperado el 25 de 08 de 2014, de <http://www.sgoliver.net/blog/?p=2682#>
- Sierra, M., & aprenderaprogramar.com. (s.f.). *Definición de servidor*. Recuperado el 26 de 08 de  
2014, de 2014:  
[http://aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftpsmtp&catid=57:herramientas-informaticas&Itemid=179](http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftpsmtp&catid=57:herramientas-informaticas&Itemid=179)
- SOLTRACK. (2012). *Refrigerados*. Recuperado el 2014
- startcapps. (2014). *Que son las notificaciones push*. Recuperado el 22 de 08 de 2014, de  
<http://www.startcapps.com/blog/que-son-las-notificaciones-push/>
- testo. (2014). *testo.com.ar*. Recuperado el 18 de 8 de 2014, de  
[http://www.testo.com.ar/es/home/productos/industria\\_farmaceutica\\_y\\_hospitales/bancos\\_de\\_sangre/monitoreo\\_en\\_bancos\\_de\\_sangre.jsp](http://www.testo.com.ar/es/home/productos/industria_farmaceutica_y_hospitales/bancos_de_sangre/monitoreo_en_bancos_de_sangre.jsp)

- treehugger. (5 de 03 de 2013). *10 environmental sensors that go along with you*. Recuperado el 06 de 2014, de <http://www.treehugger.com/clean-technology/environmental-sensors.html>
- UPVM, I. (2009). *Noma Mexicana IMNC*. Recuperado el 10 de 10 de 2014, de <http://ingenieriaindustrialupvmtareasytrabajos.files.wordpress.com/2012/06/norma.pdf>
- Veloza Cabrera, L. A., Wiesner Ceballos, C., Serrano López, M. L., Peñaranda Correa, N. R., & Huertas Salgado, A. (01 de 2010). Revista Colombiana de Bioética, vol. 5, núm. 1, enero-junio, 2010, pp. 121-141. Universidad El Bosque, Colombia.
- Vera Duran, D. C., & Barrios, R. J. (2005). sistema de monitoreo en servidores con generación de reportes a telefonía inalámbrica y web. Bucaramanga, Santander, Colombia.
- wikipedia. (19 de 08 de 2014). *Hardware libre*. Recuperado el 22 de 08 de 2014, de [http://es.wikipedia.org/wiki/Hardware\\_libre#Desventajas](http://es.wikipedia.org/wiki/Hardware_libre#Desventajas)
- wikipedia. (19 de 08 de 2014). *Telefonía móvil*. Recuperado el 22 de 08 de 2014, de [http://es.wikipedia.org/wiki/Telefonía\\_móvil](http://es.wikipedia.org/wiki/Telefonía_móvil)
- wikipedia. (31 de 03 de 2015). *Aplicación móvil*. Recuperado el 9 de 04 de 2015, de [http://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_m%C3%B3vil](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vil)
- xatakaon. (08 de 11 de 2011). *IaaS, PaaS, SaaS*. Recuperado el 16 de 08 de 2014, de <http://www.xatakaon.com/almacenamiento-en-la-nube/cuando-hablamos-de-la-nube-que-es-iaas-paas-saas>