



**UNIVERSIDAD DEL AZUAY**

**FACULTAD DE CIENCIAS DE LA ADMINISTRACION**

**ESCUELA DE INGENIERIA DE SISTEMAS Y TELEMÁTICA**

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

**Trabajo de graduación previo a la obtención del título de Ingeniero de Sistemas y Telemática**

**Autor:**

Belén Arízaga Jaramillo

**Director:**

Ing. Diego Chacón, Mhd.

**Cuenca, Ecuador**

**2014**

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

***DEDICATORIA:***

Dedico este trabajo a mis padres, a mis hermanos, a mi cuñada a mi sobrino y a mí enamorado por su gran apoyo que incondicionalmente me ha impulsado a seguir adelante y no rendirme durante esta ardua etapa.

Ana Belén.

***AGRADECIMIENTOS:***

Agradezco a las personas más cercanas que estuvieron para transmitirme los conocimientos y sus experiencias ya que me han servido para llegar hasta donde estoy.

Agradezco también a la Universidad del Azuay por las buenas enseñanzas impartidas durante los 5 años de la carrera, así como a los profesores que me ayudaron en mi desarrollo profesional.

Ana Belén.

## ***Índice de Ilustraciones y Tablas:***

- Ilustración #1, *Capas Software Android* – Android Developers  
Fuente: <http://developer.android.com/about/index.html>  
07/03/2014
- Ilustración #2, *Diseño de Navegación*  
21/03/2014
- Ilustración #3, *Consumo del Servicio Web desde la Aplicación*  
Fuente: [http://www.hipertexto.info/documentos/serv\\_web.htm](http://www.hipertexto.info/documentos/serv_web.htm)  
23/03/2014
- Ilustración #4, *Modelo Arquitectónico del Servicio Web*  
01/04/2014
- Ilustración #5, *Estructura de la Base de Datos del Sistema*  
25/05/2014
- Ilustración #6, *Tabla users*  
26/05/2014
- Ilustración #7, *Tabla elements*  
26/05/2014
- Ilustración #8, *Tabla tc*  
27/05/2014
- Ilustración #9, *Tabla element\_types*  
28/05/2014
- Ilustración #10, *Respuesta del Servicio Web*  
29/05/2014
- Ilustración #11 – *Logo RDomо*  
02/06/2014
- Ilustración #12 – *Pantalla de Logueo*  
02/06/2014
- Ilustración #13 – *Error de usuario o contraseña vacíos*  
02/06/2014
- Ilustración #14 – *Error de IP o dominio*  
02/06/2014
- Ilustración #15 – *Error de usuario o contraseña*  
02/06/2014

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

- Ilustración #16 – Pantalla del Menú de la Aplicación  
03/06/2014
- Ilustración #17 – Pantalla del Control de Luces  
04/06/2014
- Ilustración #18 – Pantalla del Control de Luminosidad  
05/06/2014
- Ilustración #19 – Pantalla del Control de Temperatura  
06/06/2014
- Ilustración #20 – Control de los Umbrales de Temperatura  
07/06/2014
- Ilustración #21 – Pantalla de Control de Riego  
08/06/2014
- Ilustración #22 – Control de Acceso, Abriendo Puerta  
09/06/2014
- Ilustración #23 – Control de Acceso, Puerta Abierta  
09/06/2014
- Ilustración #24 – Configuración de Sesión del Usuario  
10/06/2014
  
- Tabla #1, *Características del SO Android* – Android Developer  
Fuente: <http://developer.android.com/about/index.html>  
07/02/2014

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

## Contenido

Dedicatoria .....	ii
Agradecimientos .....	iii
Índice de Ilustraciones y Tablas .....	iv
Resumen.....	v
Abstract.....	vi
<b>Introducción .....</b>	<b>1</b>
Contenido .....	3
<b>CAPÍTULO 1 .....</b>	<b>3</b>
<b>Fundamentos teóricos .....</b>	<b>3</b>
1.1.    Introducción a Android .....	3
<b>1.1.1. WAP (Wireless Application Protocol, Protocolo de Redes Inalámbricas) .....</b>	<b>4</b>
<b>1.1.2. Plataformas Móviles Propietarias. ....</b>	<b>5</b>
<b>1.1.3. Google incursiona en campos inalámbricos. ....</b>	<b>6</b>
<b>1.1.4. Fabricantes: Diseño de Dispositivos Android.....</b>	<b>7</b>
<b>1.1.5. Gratuita y de Código Abierto. ....</b>	<b>8</b>
<b>1.1.6. Curva de Aprendizaje e Integración Segura.....</b>	<b>8</b>
<b>1.1.7. Un mercado gratuito, Android y una Plataforma en crecimiento. ....</b>	<b>9</b>
1.2.    Introducción a las aplicaciones móviles en Android.....	9
<b>1.2.1 Entre las principales características del sistema operativo móvil están: .....</b>	<b>10</b>
<b>1.2.2 Capas software de Android: .....</b>	<b>14</b>
1.2.2.1. Aplicaciones .....	14
1.2.2.2. Marco de Trabajo de Aplicaciones .....	14
1.2.2.3. Bibliotecas .....	15
1.2.2.4. Runtime de Android .....	15
1.2.2.5. Núcleo Linux .....	15
<b>1.2.3 Desarrollo de aplicaciones en Android: .....</b>	<b>15</b>
1.2.3.1. Aplicaciones: .....	15
1.2.3.2. Google Play.....	18
1.3    Introducción a la domótica.....	18
<b>1.3.1. Gestión de la domótica .....</b>	<b>20</b>
1.3.1.1. Energía Eléctrica. ....	20
1.3.1.2. Confort. ....	20
1.3.1.3. Seguridad. ....	20
1.3.1.4. Comunicaciones.....	21
<b>1.3.2. Sistema domótico .....</b>	<b>21</b>
<b>CAPÍTULO 2 .....</b>	<b>22</b>
<b>Análisis y diseño de la aplicación .....</b>	<b>22</b>

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

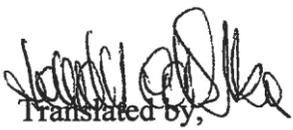
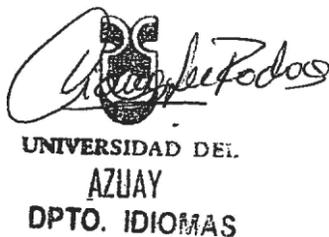
2.1	Desarrollo de la aplicación.....	22
2.2	Alcance de la aplicación.....	23
2.3	Objetivos de la aplicación móvil.....	24
2.4	Estructura de contenido.....	24
2.5	Diseño de Interfaz.....	25
2.6	Diseño de Navegación.....	26
2.7	Proyecto.....	26
	<b>2.7.1. Desafíos orientados al proyecto.....</b>	<b>26</b>
	2.7.1.1. Heterogeneidad.....	26
	2.7.1.2. Extensibilidad.....	27
	2.7.1.3. Seguridad.....	27
	2.7.1.4. Escalabilidad.....	27
	2.7.1.5. Tratamiento de fallos.....	27
	2.7.1.6. Concurrencia.....	28
	2.7.1.7. Transparencia.....	28
	<b>CAPÍTULO 3.....</b>	<b>28</b>
	<b>Comunicación entre la aplicación y el sistema.....</b>	<b>28</b>
	3.1. Modelo Arquitectónico.....	30
	3.2. Capas de Software.....	30
	<b>3.2.1. Requisitos de Diseño.....</b>	<b>30</b>
	3.2.1.1. Prestaciones o Rendimiento.....	30
	3.2.1.2. Calidad de Servicio.....	31
	3.2.1.3. Uso de cache y replicación.....	31
	3.2.1.4. Fiabilidad.....	31
	<b>3.2.2. Modelos Fundamentales.....</b>	<b>31</b>
	<b>3.2.3. Comunicaciones TCP/UDP.....</b>	<b>32</b>
	<b>CAPITULO 4.....</b>	<b>33</b>
	<b>Implementación y Pruebas.....</b>	<b>33</b>
	4.1. Implementación de la aplicación.....	33
	4.2. Implementación de la aplicación con el sistema.....	37
	<b>CAPÍTULO 5.....</b>	<b>38</b>
	<b>Recomendaciones y conclusiones.....</b>	<b>38</b>
	5.1 Conclusiones.....	38
	5.2 Recomendaciones.....	39
	<b>ANEXOS.....</b>	<b>40</b>
	Anexo #1 - Manual de Usuario.....	41
	Anexo # 2 – Código de la aplicación.....	50
	<b>Bibliografía.....</b>	<b>58</b>

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

### **ABSTRACT**

This graduation paper is about the development of a mobile application with an Android operating system for the control of a domotic system (home automation) implemented with Raspberry PI and Arduino. This application has a simple interface which encourages user interest, facilitating the daily activities that can become tedious, by providing the required comfort.

The fundamental basis of this application is in the construction and implementation of web services that are responsible for communication between the domotic architecture and its application. In order to understand its operation, the theoretical foundation that includes Android and domotics technology is explained.



Translated by,  
Lic. Lourdes Crespo

## Resumen

En la presente monografía se desarrolla una aplicación móvil con el sistema operativo Android para el control de un sistema domótico implementado con Arduino y Raspberry Pi. Esta aplicación cuenta con una interfaz sencilla, lo que incentiva el interés del usuario, facilitando así las actividades cotidianas que se pueden tornar tediosas y proporcionando el confort requerido.

La base fundamental de esta aplicación, se encuentra en la construcción e implementación de los servicios web que son los encargados de la comunicación entre la arquitectura domótica y la aplicación. Para comprender el funcionamiento se explican los fundamentos teóricos que engloba la tecnología Android y la domótica.

## **Responsabilidad**

La autora se responsabiliza de los criterios y conceptos vertidos en esta monografía

---

Belén Arizaga Jaramillo

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

## ***Introducción***

**Los dispositivos móviles revolucionaron la tecnología provocando en las personas una gran acogida de los mismos, tanto ha sido su crecimiento que se puede decir que hoy la mayoría de teléfonos son inteligentes. Esto ha generado una gran demanda de accesorios para los mismos, y por supuesto aplicaciones nativas para controles personalizados, es así que en las tiendas de aplicaciones de cada plataforma crecen sin cesar el número y la variedad de las mismas. Sin embargo se puede observar que en el Ecuador la creación de nuevas aplicaciones para la gestión y control de arquitecturas domóticas específicamente hablando no es muy extensa. Por otro lado la implementación de equipos domóticos dentro de un hogar tiene costos muy elevados, aún más si solicitamos una pantalla de control que viene con su propia aplicación para el manejo, los costos suben. Una de las soluciones sería la fabricación y desarrollo de propios equipos domóticos para el Ecuador, sin embargo para que esto suceda es necesario esperar que el conocimiento y la necesidad de la domótica crezcan.**

**La manera más fácil de estar al día en la tecnología, u obtener lo que queremos es adquiriendo algo ya hecho, pagando un valor por ello. Es debido a esta razón que el incentivo en la creación no es tan elevado en el Ecuador. Seguramente la mayoría pueden preguntarse para que intentar hacer algo que ya existe y es muy bueno. Sin embargo esta ideología puede cambiar, empezando desde uno mismo, y demostrando con resultados positivos el avance tecnológico que se puede crear. Es así que se pretende desarrollar una aplicación con una interfaz sencilla para el usuario, que le permita experimentar todo el verdadero confort que conlleva una arquitectura domótica. Es importante considerar que el diseño de dicha aplicación es muy importante para que tenga el enganche necesario con los usuarios.**

## CAPÍTULO 1

### Fundamentos teóricos

#### 1.1. Introducción a Android

Empezaremos citando la evolución de los teléfonos móviles, para luego poder comprender la utilidad de estos en la sociedad y como poco a poco se fueron introduciendo los teléfonos inteligentes específicamente Android.

La tecnología es desarrollada para facilitar a los usuarios en los aspectos posibles, así pues ponemos un ejemplo de como el teléfono móvil revoluciona a la sociedad. En muchas ocasiones olvidamos cosas importantes en casa o cuando vamos al supermercado olvidamos si tenemos o no tal producto, en la antigüedad teníamos que hacer uso de la memoria o dejarlo para comprar en la próxima ocasión, con la llegada del teléfono únicamente es necesario una llamada.

Los teléfonos móviles aparte de mantenernos comunicados y conectados nos permiten tener acceso a un sinnúmero de opciones como dirigirnos y saber en dónde estamos, o como llegar a ese lugar que deseamos, con solo esta simple razón es motivo suficiente para poder llamarlos teléfonos “Inteligentes”.

Las primeras generaciones de teléfonos móviles fueron diseñadas por fabricantes de terminales. La competencia en este mercado era feroz y se escondían como sus dispositivos estaban diseñados internamente esto permitía que no se puedan realizar aplicaciones para los mismos.

En esta dura competencia Nokia sobresalió con algo, un pequeño juego llamado “Snake” al contrario otros fabricantes incursionaban de la misma manera pero con juegos como “tetris” y “Tic-Tac-Toe”.

Sobre estos teléfonos podríamos decir que no eran esencialmente perfectos y poderosos, pero lo importante de ellos fue que cambiaron la forma de ver la comunicación en las personas y de la misma manera que la tecnología fue creciendo estas también es decir se tenía más áreas de cobertura y cada vez más personas lo utilizaban y empezaron a dejar de ser una novedad y convertirse en una solución. Otro aspecto que también fue cambiando con el transcurso del tiempo era que los usuarios eran más exigentes, es decir estos comenzaban a pedir más características, juegos, etc., pero había un problema general. Los fabricantes no tenían recursos y sobre todo

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

motivación para proporcionar a los usuarios un servicio que posea entretenimiento información y permitirles estar comunicados.

### ***1.1.1. WAP (Wireless Application Protocol, Protocolo de Redes Inalámbricas)***

En esos momentos la conectividad a internet en los teléfonos no era aún explotada en los móviles aunque teníamos en la Web una gran cantidad de contenidos atractivos para los usuarios, se empezó a incursionar en estos campos, pero había un problema, aun los teléfonos no tenían la capacidad de soportar la resoluciones que exigía la web debido a que las pantallas de los teléfonos eran de poca capacidad, otro impedimento era que tenían baja cantidad de almacenamiento y de procesamiento y los teléfonos no podían gestionar las transacciones requeridas por los navegadores web, aparte de esto los costos de banda ancha eran altos para los usuarios. Para solucionar estos problemas se ideó algo, el estándar WAP, este es un estándar en forma sencilla de HTTP, que es el protocolo básico de internet. Los navegadores WAP a diferencia de sus hermanos mayores los Web fueron diseñados para que cumplieran requisitos básicos del teléfono en memoria y ancho de banda del mismo.

El protocolo WAP permitió realizar sitios web personalizados y poco cargados de contenidos que permitían la navegación fácil y rápida desde dispositivos móviles, esta solución WAP también ofrecía a los operadores móviles la posibilidad de crear contenidos específicos generando suscripciones para los mismos generando dinero por costes de datos asociados a estas suscripciones.

Uno de los sitios que más éxito tuvo al ser publicado en WAP fue “ESPN.com” debido a que los usuarios podían leer contenidos de interés, esto los llevo a incrementar el número de sus lectores ya que se publicaban resultados deportivos en su portal WAP. Las aplicaciones más comerciables del protocolo WAP fueron catálogos sencillos, es decir estos contenían fondos de pantalla y tonos de llamada, estos le permitían al usuario tener la posibilidad de personalizar sus teléfonos por primera vez. Una de las cosas importantes que se debe mencionar es el método de pago, para realizar este se llenaba un sencillo formulario, este era enviado al usuario en manera de imagen a su teléfono, este se podría gestionar mediante varios mecanismos de costo extra como: SMS (Servicio de Mensajes Cortos), EMS (Servicio de Mensajería Multimedia) y WAP Push.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

Al principio de esta tecnología los navegadores WAP eran muy lentos y tediosos, debido a que era toda una travesía escribir una URL web por razones de la velocidad de respuesta del teléfono y esto también ocasionaba una difícil navegación por las paginas, otro limitante que ya mencionamos era la pantalla, esto no permitía a las páginas web dar una experiencia satisfactoria a los usuarios ya que no se les proveía de contenidos atractivos. Esto provoco que no se realice sitios WAP si no que se promocione contenidos a través de código SMS, estos proveían de promociones o contenidos de interés del usuario a un coste extra.

Como podemos observar la tecnología WAP no llego a satisfacer las expectativas de los proveedores móviles y era por una gran razón ya mencionada, la pantalla no brindaba el tamaño idóneo para navegar. (Lauren Darcey, Shane Conder, 2012)

### ***1.1.2. Plataformas Móviles Propietarias.***

Uno de los grandes retos de los operadores móviles era la capacidad de satisfacer a todos los segmentos de población, en especial al de 18 a 25 años, este segmento está compuesto de la gente más joven que con dinero disponible está dispuesto a gastarlo en personalizar su teléfono, pero ellos al mismo tiempo miraban sus consolas de video juegos, sus reproductores multimedia y se preguntaban si se podría tener uno que tenga todo incluido. Ellos se decían que las consolas les proporcionaban muchas horas de entretenimiento con 5 botones, entonces porque un teléfono no podía implementar esto, de la misma manera otros usuarios miraban su Palm, Cámaras digitales, BlackBerry, iPod e incluso a sus portátiles y se hacían la misma pregunta.

El avance de la tecnología permitió que muchos componentes como la memoria, la batería e incluso las pantallas bajaran de precio y al mismo tiempo mejoren su capacidad, esto permitió que se empiece a instalar versiones compactas de Sistemas Operativos como Windows y Linux en dispositivos pequeños como los celulares, esto llevo a incursionar en un mercado de los SmartPhones.

Los fabricantes de terminales o dispositivos móviles, entendieron que no pueden seguir vendiendo si es que ellos continúan con sus políticas de ocultar como esta su producto desarrollado internamente. Esto permitió que salgan muchas plataformas propietarias, en algunos SmartPhones se instalaba sistemas operativos como: PalmOs (Conocido como GarnetOs), RIM BlackBerryOs, Sun por su parte utilizó la ahora popular plataforma llamada Java específicamente surgió J2ME conocido como Java Micro

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

Edition (Java ME), y así una cantidad de plataformas propietarias como Symbian OS usadas en el Nokia, Samsung, SonyEricson y el conocido OS x iPhone que entro en el mercado en el 2008.

Así cada una de estas tenían una gran gama de servicios para sus usuarios, de cierta manera cada plataforma se especializa en algo en específico es decir algunas eran más apropiadas para vender juegos otras para aplicaciones de negocio, etc. Esto nos dice que no había una “mejor plataforma” todo esto dependía específicamente de la región o segmento de la población. (Lauren Darcey, Shane Conder, 2012)

### ***1.1.3. Google incursiona en campos inalámbricos.***

Las primeras incursiones de Google en el mercado móvil no fue fácil debido a que se encontraron con muchos obstáculos debido a una gran razón que la libertad que contaban los usuarios de internet no era la misma que los subscriptores de telefonía móvil es decir los usuarios de internet pueden elegir entre una gran variedad de marcas de ordenadores, sistemas operativos, proveedores de internet y distintas aplicaciones Web.

Muchos de los servicios de Google son gratuitos y gestionados por anuncios, y muchas de sus aplicaciones Web compiten directamente con aplicaciones disponibles en los dispositivos móviles, esto nos les dio mucho resultado así que Google busco una estrategia diferente que era buscar una plataforma más abierta el cual beneficie tanto a usuarios como a desarrolladores, es decir escogieron el modelo de Internet, esto permite escoger software sea gratuito o sea de pago, esto facilita y permite algo fundamental que es el libre mercado.

Creación de Open Handset Alliance.

La OHA (Open Handset Alliance, Alianza de terminales abiertos) se creó en el 2007, esta es una alianza comercial de las más grandes empresas de éxito en las plataformas móviles, en este grupo encontramos fabricantes desarrolladores y los proveedores de servicio.

Android Inc., es una empresa que fue creada por Andy Rubin, esta empresa fue comprada por Google el 2005, en esa mismo año empezaron a trabajar junto a la OHA (Open Handset Alliance, Alianza de terminales abiertos) en una plataforma abierta no propietaria basada en la tecnología creada por Android Inc., el cual tenía un gran

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

objetivo, este era eliminar los problemas que obstaculizaban la integración en una sola comunidad de móviles. (Lauren Darcey, Shane Conder, 2012)

#### ***1.1.4. Fabricantes: Diseño de Dispositivos Android.***

La gran mayoría de los miembros de la OHA son fabricantes de terminales o dispositivos móviles, entre estos podemos encontrar LG, HTC, Sony, Samsung, Motorola, Dell, Sony Ericsson y el resto también está compuesta por empresas que fabrican semiconductores, el más grande de estos es Intel.

El primer dispositivo en salir a la venta con un Sistema Operativo Android es fue comercializado por HTC y fue el T-Mobile G1 este fue lanzado en principios del 2008, en años posteriores, por ejemplo, en el 2009 y finales del 2010 la plataforma Android tuvo tanta acogida que esta empezó a liderar el mercado de los SmartPhones ganando terreno sobre sus plataformas competidoras como BlackBerry, iOS, y Windows Mobile.

En el mes de mayo Google Inc., publica los resultados de las estadísticas de Android en la conferencia anual llamada Google IO, específicamente hasta el mes de julio del 2011 Google público que más de 300 dispositivos Android fueron vendidos en más de 200 países alrededor de todo el mundo, es decir más de 500.00 dispositivos habían sido activados, una de las principales ventajas que permitió que Android tenga tanta acogida es que: los fabricantes daban un servicio extra, es decir tanto el fabricante y el operador móvil permitían al usuario contar con un soporte técnico gratuito sobre el dispositivo.

Otra gran ventaja de los dispositivos Android parte del hecho de que muchos de estos no tienen el precio habitual de un SmartPhone es decir su precio es mucho más reducido y en algunos casos estos dispositivos se ofrecen gratuitamente con la activación del servicio en alguna operadora, esto permitió que, por primera vez los usuarios puedan contar con un dispositivo inteligente con todas sus funciones sin la necesidad de ser un usuario de poder alto de adquisición.

Algo que no puede dejar de ser mencionado es el hecho que Google logro ventas de dispositivos gracias al masivo desarrollo de aplicaciones móviles para su plataforma. En el año 2011 en el Market de Android había disponible alrededor de 250000 aplicaciones y esto sin tomar en cuenta las aplicaciones vendidas de manera individual es decir sin pasar por el Market, este número tan alto es debido a una razón grande, Android al no ser un sistema operativo propietario a reducido los costes del operador

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

en licencias y derechos de autor al contrario de los sistemas propietarios. (Lauren Darcey, Shane Conder, 2012)

#### ***1.1.5. Gratuita y de Código Abierto.***

Como mencionamos anteriormente Android es una plataforma de código abierto es decir ni los desarrolladores ni los fabricantes de dispositivos pagan derecho de autor o gastos por licencia para desarrollar en la misma, el nombre de la licencia que Android provee es GNU GPL2 (Generic Public License Version 2, Licencia Publica General Versión 2) esta es una licencia susceptible a mejora es decir que cualquier mejora del Sistema Operativo solamente debe cumplir los términos de la licencia.

Android también posee otro tipo de licencia bajo cual esta se distribuye la ASL/Apache 2 (Licencia de Software Apache), esta licencia permite la distribución de código fuente, este puede ser abierto como cerrado, es decir los fabricantes de dispositivos pueden mejorar la plataforma sin necesidad de publicar estas al mundo del código abierto con esto se puede acceder a mejoras que son relacionadas a un solo x dispositivo. Otro beneficio es que los desarrolladores de Android pueden publicar sus aplicaciones bajo la licencia que ellos deseen.

Las herramientas de desarrollo de Android, es decir su kit de desarrollo es completamente gratuito, no se paga suscripción, y el SDK de Android se lo descarga de la página oficial y solamente se acepta sus términos y condiciones.

Pero se tiene más motivos para establecer que Android es gratuita, estos son:

Es completa debido a que se partió de una base muy estable y esto permite tener un enfoque integral de cómo desarrollar para esta.

Abierta, es decir todos tienen acceso a sus características sin límite alguno. (Lauren Darcey, Shane Conder, 2012)

#### ***1.1.6. Curva de Aprendizaje e Integración Segura.***

Java es el lenguaje de desarrollo para Android, aparte Android posee construcciones de programación típicas como: hilos, procesos, estructuras y sobre todo la encapsulación. Android también permite el uso de librerías típicas de java, las más usadas son las de grafico de tareas, base de datos, específicamente para el uso de la base de datos el gestor de esta es el SQLite. Los desarrollador también tienen la posibilidad de competir sanamente unos con otros sin distinción de estatus económico

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

debido a que todos los desarrolladores pueden acceder a las mismas APIs de Google y también tienen todo el acceso a el hardware interno permitiendo realizar aplicaciones mucho más potentes.

Uno de los potenciales que Android permite a sus desarrolladores es la completa integración, es decir las aplicaciones pueden interconectarse entre ellas obteniendo datos o información una de la otra, realizar este tipo de operaciones también conlleva una gran responsabilidad, pero de igual manera Android es lo suficientemente robusto para garantizar el éxito de este tipo de operaciones. (Lauren Darcey, Shane Conder, 2012)

### ***1.1.7. Un mercado gratuito, Android y una Plataforma en crecimiento.***

En la amplia gama de posibilidades que permite Android para la publicación de aplicaciones los desarrolladores están en la libertad de elegir cuál de ellas aplicar para garantizar sus ingresos, entre algunas de ellas tenemos: software gratuito, shareware, software de prueba, aplicaciones con anuncios o aplicaciones de pago. Con esto Android busca cumplir el objetivo con el cual fue creado, este busca eliminar las restricciones con respecto al desarrollo que ya las hemos mencionado anteriormente.

En sus inicios Android y sus desarrolladores tenían que enfrentarse a un gran reto, debido a que es una herramienta nueva los SDK sufrían revisiones frecuentes y por ende modificaciones continuas al mismo y esto se complementaba a la falta de documentación completa, uno de los típicos problemas que se encontraban era el hecho de que no todos los dispositivos eran compatibles con la última versión de la plataforma. Este hecho daba como consecuencia que los desarrolladores debían utilizar diferentes versiones del SDK para poder abarcar a todos los usuarios, pero con el paso del tiempo estos tipos de problemas se han ido perdiendo debido a que Android se ha convertido en una plataforma bien asentada con cimientos estables, y sobre todo con el apoyo global es decir Android tiene una comunidad de foros de ayuda muy activa y generosa. (Lauren Darcey, Shane Conder, 2012)

## **1.2. Introducción a las aplicaciones móviles en Android.**

Android como ya sabemos es un Sistema Operativo y una plataforma de software en el cual se desarrollan aplicaciones, pero como todo Sistema Operativo debe tener una arquitectura sobre el cual se ejecuta. Linux es el sistema operativo base de Android, de

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

la misma manera cada aplicación Android se ejecuta bajo su propia máquina virtual, la ventaja de realizar estas operaciones es que hay una probabilidad mucho más baja de que las aplicaciones provoquen que el dispositivo falle y halla un error global.

A diferencia de sistemas operativos para dispositivos móviles como iOS o Windows Phone, el código fuente de Android se mantiene abierto. Soporta multimedia, interfaz táctil, características basadas en voz y una amplia variedad de componentes de hardware. Si se añade a esto un entorno de desarrollo gratuito que incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software, Android constituye una opción interesante para nuevos desarrolladores. (Purdy.)

### 1.2.1 Entre las principales características del sistema operativo móvil están:

**Característica**      Especificaciones

<b>Diseño de dispositivo</b>	La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales. (Purdy.)
<b>Almacenamiento</b>	SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos. (Android Developers)
<b>Conectividad</b>	Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+ y WiMAX. (Android Developers)
<b>Mensajería</b>	SMS y MMS son formas de mensajería, incluyendo mensajería de texto y ahora la Android Cloud to Device Messaging Framework (C2DM) es parte del servicio de Push Messaging de Android.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

	(Android Developers)
<b><i>Navegador web</i></b>	El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome. El navegador por defecto de Kit Kat obtiene una puntuación de 100/100 en el test Acid3. (Android Developers)
<b><i>Soporte de Java</i></b>	Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner. (Android Developers)
<b><i>Soporte multimedia</i></b>	Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP. (Android Developers)
<b><i>Soporte para streaming</i></b>	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player. (Android Developers)

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

<p><b><i>Soporte para hardware adicional</i></b></p>	<p>Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D. (Android Developers)</p>
<p><b><i>Entorno de desarrollo</i></b></p>	<p>Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo integrado es Eclipse (actualmente 3.4, 3.5 o 3.6) usando el plugin de Herramientas de Desarrollo de Android. (Android Developers)</p> <p>Actualmente se creó el propio entorno de desarrollo para aplicaciones móviles, Android Studio es un nuevo entorno de desarrollo de Android basado en IntelliJ IDEA. Al igual que en Eclipse con el ADT Plugin, Android Studio proporciona herramientas para desarrolladores de Android integradas para el desarrollo y la depuración. (Android Developers)</p>
<p><b><i>Google Play</i></b></p>	<p>Google Play es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC. (Android Developers)</p>
<p><b><i>Multi-táctil</i></b></p>	<p>Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir patentes de otras compañías). Más tarde, Google publicó una actualización para el Nexus One y el Motorola Droid que activa el soporte multi-táctil de forma nativa. (Android Developers)</p>
<p><b><i>Bluetooth</i></b></p>	<p>El soporte para A2DF y AVRCP fue agregado en la versión 1.5; el envío de archivos (OPP) y la exploración del directorio telefónico</p>

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

	fueron agregados en la versión 2.0; y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2. (Android Developers)
<b>Videollamada</b>	Android soporta videollamada a través de Google Talk desde su versión HoneyComb. (Android Developers)
<b>Multitarea</b>	Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj, a diferencia de otros sistemas de la competencia en la que la multitarea es congelada. (Como por ejemplo iOS, en el que la multitarea se limita a servicios internos del sistema y no a aplicaciones externas) (Android Developers)
<b>Características basadas en voz</b>	La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema. (Android Developers)
<b>Tethering</b>	Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o inferiores mediante aplicaciones disponibles en Google Play (por ejemplo PdaNet). Para permitir a un PC usar la conexión de datos del móvil Android se podría requerir la instalación de software adicional. (Android Developers)

Tabla #1, Características de SO Android, Android Developers

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

### 1.2.2 Capas software de Android:

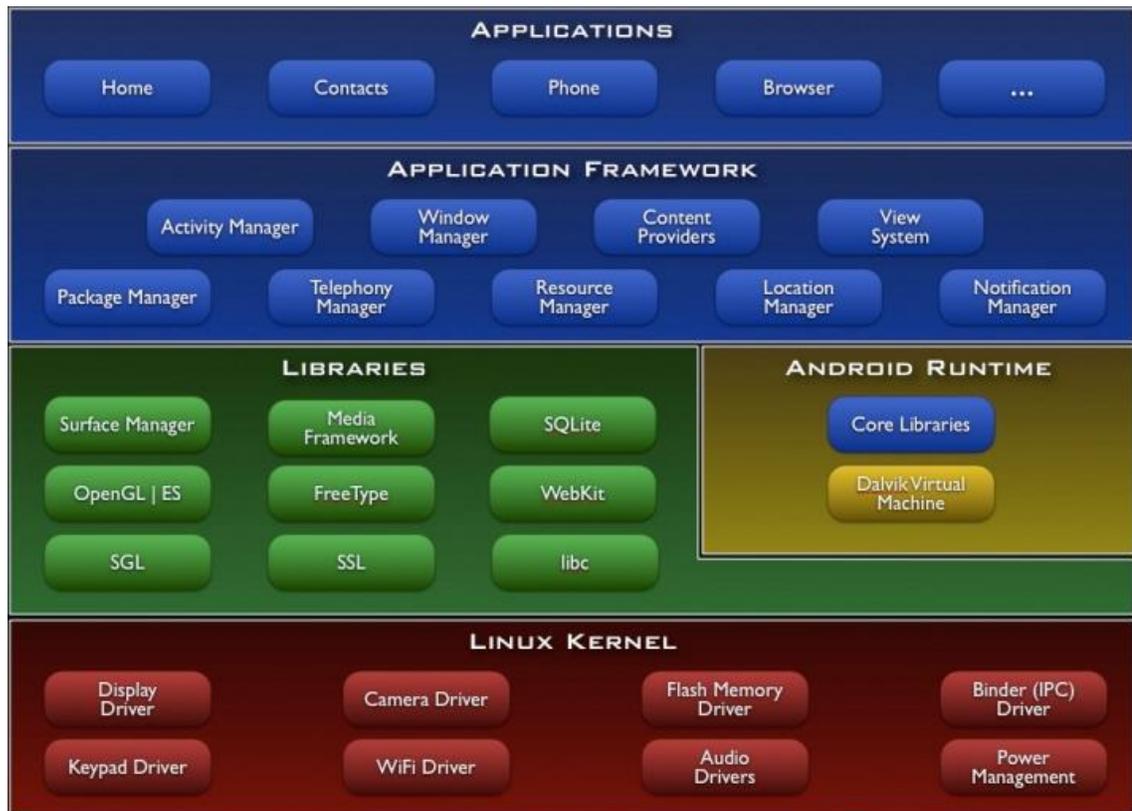


Ilustración #1, Capas software Android – Android Developers

#### 1.2.2.1. Aplicaciones

Las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

#### 1.2.2.2. Marco de Trabajo de Aplicaciones

Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

#### *1.2.2.3. Bibliotecas*

Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.

#### *1.2.2.4. Runtime de Android*

Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato dex por la herramienta incluida "dx".

#### *1.2.2.5. Núcleo Linux*

Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

### **1.2.3 Desarrollo de aplicaciones en Android:**

#### *1.2.3.1. Aplicaciones:*

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C o C++. El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

de Java y estar en posesión del kit de desarrollo de software o «SDK» provisto por Google el cual se puede descargar gratuitamente.

Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos.

Las aplicaciones para dispositivos móviles en Android son realizadas en java, para lo cual los desarrolladores tienen la opción de hacerlo en eclipse o en la propia herramienta llamada Android Studio. Estas herramientas han permitido el crecimiento de las aplicaciones para este sistema operativo, así mismo podemos conocer que cada día existe un sin número de aplicaciones nuevas las mismas que intentan buscar un plus para los usuarios y en las mismas ponen todo su ingenio y creatividad para una mejor experiencia y aceptación.

Habitualmente cuando hablamos de aplicaciones móviles, es inevitable pensar que estas van a hacer uso del Internet para cumplir con cierto objetivo o función. Sin embargo existen muchas otras que no necesitan el consumo de datos de la nube, puesto que su objetivo o función no lo amerita, en una diferente situación existen aplicaciones que funcionan parte con Internet y parte sin, es decir para lograr que el consumo de datos no sea excesivamente caro o simplemente si el dispositivo puede almacenar la información y trabajar desde aquí y nada más al final del día actualizar la información del dispositivo con la nube.

Así pues el paquete android.net proporciona acceso a propiedades de conectividad del dispositivo mediante de la clase ConnectivityManager: la misma que nos permite saber si el dispositivo está conectado a alguna red, y de qué tipo es (WiFi, GSM,...)

La manera de comunicarse desde la aplicación a la nube se la puede realizar utilizando las librerías que Android nos proporciona como son: HttpClient, HttpPost, HttpGet, las mismas que gestionan el envío y la recepción de data desde un servicio web específico.

**HttpClient:** está pensada para implementar clientes HTTP, los mismos que encapsulan una mezcla heterogénea de objetos necesarios para ejecutar solicitudes HTTP al manipular las cookies, autenticación, gestión de la conexión, y otras características. Seguridad de los hilos de los clientes HTTP depende de la implementación y la configuración del cliente específico. (Google, 2008)

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

- La principal función de la biblioteca Apache HttpClient es ejecutar métodos HTTP (GET, POST, PUT, HEAD,...)
- La ejecución de un método HTTP implica el intercambio de peticiones HTTP y respuestas HTTP, que normalmente son realizadas internamente por HttpClient de manera transparente al programador.
- El programador proporciona un objeto petición que ha de ser ejecutado y HttpClient se encarga de transmitir la petición y de recibir la respuesta a través de una conexión TCP, elevando una excepción si hay algún problema.

**HttpPost:** El método POST se utiliza para solicitar que el servidor de origen acepta la entidad incluida en la solicitud como un nuevo subordinado del recurso identificado por el Request-URI en el Request-Line. POST se ha diseñado para permitir un método uniforme para cubrir las siguientes funciones: (Google, 2008)

- Anotación de los recursos existentes
- Publicar un mensaje en un tablón de anuncios, grupos de noticias, listas de correo, o grupo similar de artículos
- Proporcionar un bloque de datos, tales como el resultado de la presentación de una forma, a un proceso de manejo de datos
- Prolongación de una base de datos a través de una operación de anexión

**HttpGet:** El método GET significa recuperar cualquier tipo de información (en forma de una entidad) es identificado por el Request-URI. Si el URI de solicitud se refiere a un proceso de producción de datos, es que los datos producidos que se volvió como la entidad en la respuesta y no el texto de origen del proceso, a menos que el texto pasa a ser la salida del proceso.

GetMethod seguirán redirigir las solicitudes desde el servidor HTTP de forma predeterminada. Este comportamiento puede desactivarse llamando setFollowRedirects (false). (Google, 2008)

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

### 1.2.3.2. *Google Play*

Google Play es la tienda en línea de software desarrollado por Google para dispositivos Android. Una aplicación llamada "Play Store" que se encuentra instalada en la mayoría de los dispositivos Android y permite a los usuarios navegar y descargar aplicaciones publicadas por los desarrolladores. Google retribuye a los desarrolladores el 70% del precio de las aplicaciones.

Por otra parte, los usuarios pueden instalar aplicaciones desde otras tiendas virtuales (tales como Amazon Appstore o SlideME) o directamente en el dispositivo si se dispone del archivo APK de la aplicación.

## 1.3 Introducción a la domótica.

La palabra domótica nace en Francia, ya que ellos son autores de adaptar términos propios a las nuevas disciplinas, surgió así la palabra "Domotique", que a su vez es la contracción de las palabras "domo" e "informatique". El mismo que hace referencia a "domo" como el hogar, y "informatique" cómo la informática o automatización. De hecho, la enciclopedia Larousse definía en 1988 el término domótica como: "el concepto de vivienda que integra todos los automatismos en materia de seguridad, gestión de la energía, comunicaciones, etc.". Es decir, el objetivo es asegurar al usuario de la vivienda un aumento del confort, de la seguridad, del ahorro energético y de las facilidades de comunicación. (Delgado Hidalgo Luis, Moreno Demera Walter, Veliz Velastegui Ninfa, Leyton Edgar, 2009)

Es importante remarcar el hecho de que automatización es distinto de domótica, puesto que en muchos casos las personas llaman a una automatización un sistema domótico; es importante aclarar que la domótica incluye la automatización. Sin embargo, la automatización no incluye a la domótica. Explícitamente la domótica es un sector más amplio, y la automatización es un subconjunto de ésta.

Inicialmente el término "Domótica" definía los sistemas electrónicos y eléctricos susceptibles de ser integrados en una vivienda (seguridad, automatización, entretenimiento, comunicaciones, etc.) hoy en día, se entiende por Domótica los sistemas de control y automatización. Considerando que las principales aplicaciones

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

de domótica se encuentran dentro de las áreas funcionales en la gestión energética, el confort en la seguridad y en la comunicación. (Domótica y Hogar Digital)

Una definición más técnica del concepto sería: "conjunto de servicios de la vivienda garantizados por sistemas que realizan varias funciones, los cuales pueden estar conectados entre sí y a redes interiores y exteriores de comunicación. Gracias a ello se obtiene un notable ahorro de energía, una eficaz gestión técnica de la vivienda, una buena comunicación con el exterior y un alto nivel de seguridad".

Como sabemos la definición de domótica varía según la importancia que se le quiere dar, sin embargo se puede encontrar una definición que contiene los componentes más importantes de todas estas definiciones, obteniendo así el siguiente significado completo y compacto.

Domótica es la disciplina que aborda la planificación, el diseño, la construcción, la implantación y la evaluación de todas aquellas tecnologías, productos y servicios que facilitan la interacción entre el individuo y su entorno a través de acciones preferiblemente naturales, sea cual sea éste.

Existen otras interpretaciones de la palabra domótica que hoy en día encajan correctamente con el significado que se quiere transmitir de la misma. Es así que se divide la palabra en "Domo-TIC-A", sin acento. Donde "domo" signifique, la casa, la vivienda, de acuerdo a su significado latino. Así pues "TIC" esté por "Tecnologías de la Información y de la Comunicación", y "A", por automatización.

De una manera general, un sistema domótico dispondrá de una red de comunicación y diálogo que permite la interconexión de una serie de equipos a fin de obtener información sobre el entorno doméstico y, basándose en ésta, realizar unas determinadas acciones sobre dicho entorno. (Mdhd., 2014)

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

### **1.3.1. Gestión de la domótica**

La domótica se encarga de gestionar principalmente los siguientes cuatro aspectos del hogar:

#### *1.3.1.1. Energía Eléctrica.*

En este campo, la domótica se encarga de gestionar el consumo de energía, mediante temporizadores, relojes programadores, termostatos, etc. También se aprovecha de la tarifa nocturna, mediante acumuladores de carga.

#### *1.3.1.2. Confort.*

La domótica nos proporciona una serie de comodidades, como pueden ser el control automático de los servicios de: Calefacción, Agua caliente, Refrigeración, Iluminación y la gestión de elementos como accesos, persianas, toldos, ventanas, riego automático, etc.

#### *1.3.1.3. Seguridad.*

La seguridad que nos proporciona un sistema domótico es más amplia que la que nos puede proporcionar cualquier otro sistema, pues integra tres campos de la seguridad que normalmente están controlados por sistemas distintos:

- Seguridad de los bienes: Gestión del control de acceso y control de presencia, así como la simulación de presencia. Alarmas ante intrusiones.
- Seguridad de las personas: Especialmente, para las personas mayores y los enfermos. Mediante el nodo telefónico, se puede tener acceso (mediante un pulsador radiofrecuencia que se lleve encima, por ejemplo) a los servicios de ambulancias, policía, etc.
- Incidentes y averías: Mediante sensores, se pueden detectar los incendios y las fugas de gas y agua y mediante el nodo telefónico, desviar la alarma hacia los bomberos, por ejemplo.

También se pueden detectar averías en los accesos, en los ascensores, etc.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

#### *1.3.1.4. Comunicaciones.*

Este aspecto es imprescindible para acceder a multitud de servicios ofrecidos por los operadores de telecomunicaciones. La domótica tiene una característica fundamental, que es la integración de sistemas, por eso hay nodos (pasarela residencial) que interconectan la red domótica con diferentes dispositivos, como Internet, la red telefónica, etc. (Luís Delgado Hidalgo, Walter Moreno Demera, Ninfa Véliz Velástegui, Edgar Leyton)

#### **1.3.2. Sistema domótico**

Para que se pueda hablar de Domótica, la edificación tiene que ser inteligente, es decir, ha de incorporar elementos o sistemas basados en las nuevas tecnologías de la información; como poseer una red de cableado, red inalámbrica o ambas, que le aseguren la posibilidad de implantar servicios de control y gestión. A la vez, esta red debe permitir el paso de varias señales al mismo tiempo, y todo ello gestionado eficazmente desde uno o varios puntos. Si a esto le sumamos el control de las instalaciones del edificio con una especial atención al gasto, ocio, confort, conseguiremos lo que se llama integración de servicios, o un sistema domótico completo. (Mdhd., 2014)

Teniendo un sistema domótico se puede crear un sin número de aplicaciones para su control, puesto que cada ámbito o integración que se realice en el sistema debe tener la posibilidad de controlarse y gestionarse por el usuario. Al incorporar las nuevas tecnologías de la información el sistema permite no solo un control y gestión por parte del usuario dentro de la edificación, sino además se presta para ofrecerle un plus al usuario, que es poder controlar al sistema por medio del internet.

Es así que nace la iniciativa de crear no sólo aplicaciones específicas para dispositivos propios domóticos que gestionen el sistema, sino más bien la posibilidad de crear aplicaciones que funcionen en los populares dispositivos inteligentes. Las aplicaciones móviles se vuelven así parte fundamental de un sistema domótico, ya que cumplen con uno de sus objetivos que es la comodidad del usuario.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

Entonces una aplicación móvil diseñada para un sistema domótico busca principalmente integrar todo el sistema y no dar soluciones aisladas, es importante considerar que toda acción que la aplicación permita realizar en el sistema domótico tiene que ser accesible al usuario para realizarla de manera manual. Entonces ahí si estamos hablando de un sistema domótico completo que está integrado y que permite la comunicación interna como externa del mismo, y que además cumple con la facilidad de uso para el usuario y está funcional para cualquier escenario.

## **CAPÍTULO 2**

### **Análisis y diseño de la aplicación**

Este capítulo pretende explicar cómo se va a desarrollar la aplicación y cuáles son sus objetivos, su alcance, y cómo se va a diseñar la aplicación desde su aspecto visual. La parte fundamental para la creación de la aplicación es su interface y cómo vamos a manejar los colores, imágenes y temas disponibles. Esta combinación es de vital importancia. Se deberá buscar una armonía entre los colores siguiendo los estándares para el diseño de interface.

#### **2.1 Desarrollo de la aplicación.**

La aplicación que se pretende realizar nos sirve de ayuda para la gestión y control de un sistema domótico, el mismo que utiliza Arduino y Raspberry Pi para su funcionamiento. Esta aplicación nos sirve para mostrar que se puede hacer un sistema completo con software libre.

Se propone la creación de una aplicación móvil capaz de ofrecer comodidad y control al usuario para gestionar ciertos aspectos domóticos como son el control de acceso, riego, iluminación y climatización, expresados en un prototipo que simula la arquitectura domótica de un hogar.

Para la aplicación de la interfaz móvil se ha considerado varios aspectos del objetivo de la aplicación. Para esta aplicación se ha seleccionado colores, café oscuro para el fondo con textos de cabecera de color blanco ya que son colores que van a resaltar

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

con las imágenes que van a expresar los diferentes controles que se tiene, además de que contribuye a una mejor percepción para los usuarios. Para todos los textos se ha escogido el color blanco, que resalta ante un color oscuro de fondo, además que el blanco muestra con claridad lo que se requiere expresar. El logo que se ha creado tiene como elemento principal un estilo flor con un decorado, ya que se ha pensado en un diseño ligero y que represente también la sencillez y el confort que brinda la aplicación móvil. Los menús a utilizarse en la aplicación han sido pensados para que el usuario maneje con facilidad, además de que se ha considerado en poner todo con imágenes para que la aplicación sea bastante intuitiva en cuanto al control de los aspectos que se le proporcionan al usuario.

## **2.2 Alcance de la aplicación**

El proyecto tiene como alcance crear una aplicación operativa en Android para dispositivos móviles capaz de controlar un sistema domótico usando herramientas de software y hardware abiertos (Arduino y Raspberry Pi) y Eclipse para Android. La aplicación debe ser dinámica y mostrar los estados actuales de los dispositivos que se encuentran conectados y cambian de estado, además mostrará el número de dispositivos conectados únicamente. Al ser una aplicación de uso para el control y gestión de un sistema, se considera que los usuarios de la misma van a ser de todo tipo, por lo que la principal característica de la aplicación es la facilidad de uso y que es muy intuitiva para cualquier tipo de usuario. Como resultado se tendrá un prototipo de una aplicación para gestionar el control de un hogar domótico.

Para el análisis del alcance del proyecto se definen las etapas de creación del mismo:

- Coordinación en el diseño de la base de datos.
- Desarrollo del servicio web de comunicación
- Diseño de una interfaz de usuario intuitiva
- Implementación de la interfaz junto con el servicio web para la comunicación
- Período de prueba de la aplicación
- Pruebas entre el funcionamiento de la aplicación con el sistema domótico

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

Mediante estos pasos conseguiremos llegar a nuestra meta inicial de la aplicación.

Uno de los aspectos principales de una aplicación es su diseño de interfaz, la misma que debe cumplir con los requerimientos necesarios además de ser completamente intuitiva y fácil de usar. Es así que se ve reflejada la necesidad de realizar un análisis completo que incluye modboards, selección de colores y análisis de navegación para que la aplicación cumpla con los mínimos requerimientos establecidos.

Requerimiento de contenidos:

- Interfaz de fácil manejo
- Control de Iluminación
- Control de riego
- Control de acceso
- Control de calefactor/aire acondicionado

### **2.3 Objetivos de la aplicación móvil**

- Mostrar de manera clara como controlar el sistema domótico.
- Informar de manera fácil al usuario el estado actual de todos los elementos del sistema.
- Captar la atención del usuario, mediante imágenes y texto intuitivo únicamente.
- Desarrollar una presentación visual clara.
- Consistencia en la apariencia estética (íconos, fuentes, colores, organización).
- Organización de los elementos dependiendo el número existente de los mismos en el sistema.

### **2.4 Estructura de contenido**

El contenido de la aplicación es bastante sencilla e intuitiva, ya que va a tener una pantalla principal en la cual se tiene acceso directo a todas las funcionalidades. Ya que la aplicación es para el control de un sistema domótico, todas las funcionalidades deben estar visibles y desplegadas de una manera fácil e intuitiva para el usuario.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

La arquitectura jerárquica le permitirá al usuario moverse de sección a sección, como son:

- Menú principal.
- Control de acceso.
- Apertura de puerta.
- Control de riego.
- Inicio y finalización de riego.
- Control de iluminación.
- Control del nivel de iluminación.
- Encendido o apagado de iluminación.
- Control de temperatura.
- Encendido apagado Calefactor.
- Encendido apagado Aire Acondicionado.

## **2.5 Diseño de Interfaz**

Se manejará el encabezado para cada sección del sitio web.

- El logo de la aplicación se pondrá en la parte superior izquierda y al lado derecho se pondrá el nombre de la actividad donde nos encontramos.
- Se tendrá un menú principal con imágenes para gestionar el control de acceso, riego, iluminación y climatización.
- En la parte de cada gestión se encontrará imágenes que nos muestran el estado real de lo que está pasando en el sistema, además de conocer cuántos elementos conectados existen.
- En la parte superior izquierda se tendrá un botón que servirá para ir atrás en la navegación de la aplicación.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

## 2.6 Diseño de Navegación

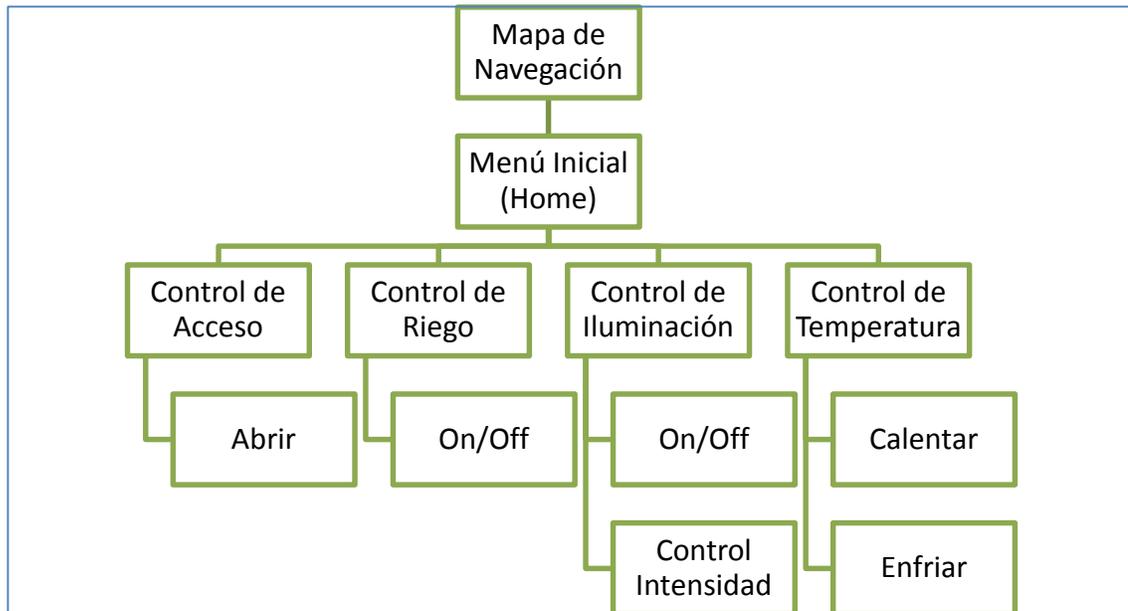


Ilustración #2, *Diseño de navegación*

## 2.7 Proyecto

La interfaz de la aplicación puede parecer bastante sencilla, pero lo que se ofrece al usuario es lo necesario y directo para su utilización, puesto que al usuario no le sirve una aplicación llena de cosas pero que confunda su uso, sino más bien una aplicación fácil de usar, informativa en todo sentido y que cumpla con los requerimientos establecidos.

### 2.7.1. *Desafíos orientados al proyecto*

#### 2.7.1.1. *Heterogeneidad*

Al basarse en esta plataforma se está inmerso en el uso de aplicaciones middleware enfocados para dispositivos como teléfonos inteligentes, tabletas que darán un enmascaramiento de la heterogeneidad.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

#### *2.7.1.2. Extensibilidad*

El sistema podrá extenderse, es decir en diversos dispositivos (hardware); cabe mencionar que así se trabaja conjuntamente dependiendo de la página de configuración del sistema domótico, que lo gestiona mediante una página web y de esta manera desde diversos dispositivos móviles accederemos a dicha información para reflejarla de acuerdo al sistema.

#### *2.7.1.3. Seguridad*

Para manejar la confidencialidad de la aplicación se considera un control de usuarios mediante la petición de datos como el usuario y contraseña. Para manejar la integridad, el sistema accede a la misma base de datos que el sistema domótico genera, la misma que se alojará en el servidor Raspberry Pi, que es el que controla el ingreso y actualización de toda la información, de esta manera se procesa información siempre actualizada y real.

Para manejar la disponibilidad del sistema, la aplicación manipulará los datos, consultas, reportes por medio de la web, por lo que se incorpora la menor carga a la aplicación, es decir número reducido de imágenes, sonidos, o archivos innecesarios.

#### *2.7.1.4. Escalabilidad*

Capacidad del sistema de poder añadir o disminuir recursos tanto en la aplicación móvil, como en el sistema en sí. Sin embargo la administración de este sistema estará dispuesto al crecimiento ya que los requerimientos cambian con el tiempo.

#### *2.7.1.5. Tratamiento de fallos*

La capacidad de recuperación de algún fallo con la manipulación de un gestor de base de datos que ayude en este sentido, con la administración de todos los datos almacenados en ella. Se maneja la redundancia de datos de la mejor manera sin exceder para tener como respaldo ante cualquier falla eléctrica o humana que haga perder toda la información.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

#### *2.7.1.6. Concurrencia*

Los usuarios pueden acceder al servidor para obtener información, el mismo que será administrado en base a privilegios, de esta manera gestionará las acciones más importantes de forma más rápida mientras que las de menor importancia se irán colocando en cola.

#### *2.7.1.7 Transparencia*

La aplicación brindará transparencia a los usuarios debido a que la información a la que accederán estará ubicada en la web, lo que le dará transparencia de concurrencia y movilidad ya que como está orientado a dispositivos móviles sólo necesitará tener acceso al INTERNET para descargarse la aplicación y emplearla.

## **CAPÍTULO 3**

### **Comunicación entre la aplicación y el sistema**

Este capítulo tiene como objetivo conocer un poco más sobre la comunicación entre la aplicación a través de los métodos POST y GET de la librería Android que ya explicamos y como estos se comunican con nuestro servicio web que va a ser alojado en el servidor Raspberry Pi, que es el que va a controlar todo el sistema domótico.

La siguiente imagen representa el escenario de la aplicación cliente, que puede estar desarrollada en cualquier lenguaje y corriendo sobre diferentes plataformas, pero en este caso está hecha en Android. En ella se muestra como la aplicación va a consumir el servicio del estado del sistema domótico y este a su vez consume servicios de los estados de los controladores (Arduinos por áreas) para que la aplicación desarrollada pueda realizar muchas actividades con tan sólo consumir los servicios y si es necesarios enviar sólo algunos parámetros para ejecutar operaciones complejas.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

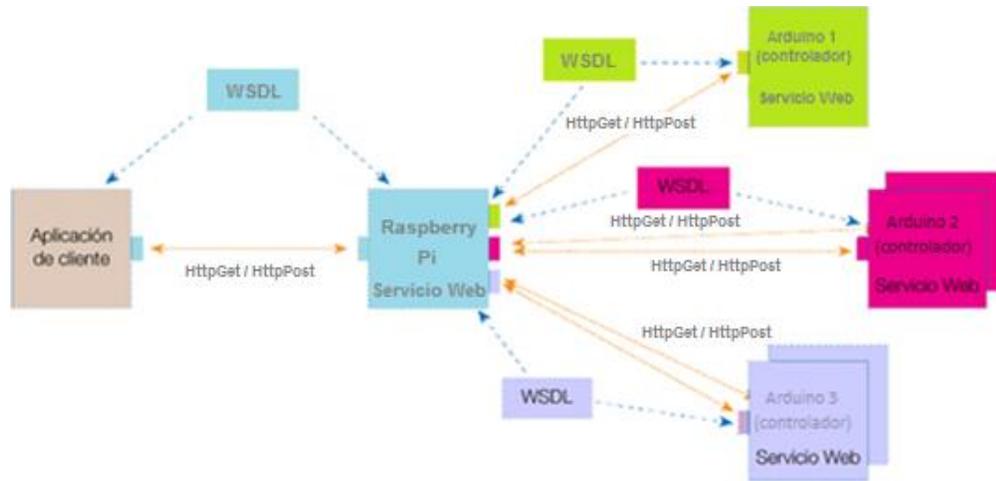


Ilustración #3 – Consumo del Servicio Web desde la Aplicación

WSDL.- (en ocasiones leído como como wisdel) son las siglas de Web Services Description Language, un formato XML que se utiliza para describir Servicios Web.

SOAP.- (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

Un web service es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Así que distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet, esto significa que los web services aportan interoperabilidad.

La interoperabilidad se consigue mediante la adopción de estándares abiertos. El servicio web que se utilizará en esta aplicación está basado en un estándar abierto llamado SOAP (Simple Object Access Protocol) o XML-RPC (XML Remote Procedure Call). (George Coulouris, Jean Dollimore, 2012)

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

### 3.1. Modelo Arquitectónico



Ilustración #4 – Modelo Arquitectónico del Servicio Web

### 3.2. Capas de Software

Capa1: Cliente de aplicación móvil.

Capa2: Web Service.

Capa3: Servidor de Datos (Raspberry Pi).- Base de datos Mysql.

#### 3.2.1. Requisitos de Diseño

##### 3.2.1.1. Prestaciones o Rendimiento

La capacidad de respuesta de la aplicación debe ser rápida y eficaz, contar con el menor porcentaje de carga computacional, manejando un gestor de base de datos que garantice robustez, integridad y lo más importante seguridad ya que la información que se maneja dentro de esta aplicación es muy sensible.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

#### *3.2.1.2. Calidad de Servicio*

Los usuarios de la aplicación para dispositivos móviles serán atraídos por la alta calidad que la plataforma ofrece, ya que ANDROID proporciona soporte incorporado, en un futuro no tan lejano para potentes gráficos en 2D y 3D. Sin embargo para este caso se enfocará en enviar la respuesta solicitada por el usuario (petición datos del sistema domótico) de la manera más eficaz con una interfaz sencilla que no requiera conocimientos específicos.

#### *3.2.1.3. Uso de cache y replicación*

Al ser una tecnología de código abierto, los desarrolladores pueden utilizar el código de otros programadores y aplicaciones de esta manera se ahorraría algo de tiempo para mejorar aspectos de relevancia dentro de esta aplicación.

#### *3.2.1.4. Fiabilidad*

Al solicitar a los usuarios que ingresen un usuario y una contraseña, la aplicación se asegura que pueda controlar el sistema únicamente las personas que tienen la autorización de hacerlo, así también el usuario tendrá la opción de configurar el tiempo que desea que la cuenta este activa. Manteniendo una seguridad extra por si acaso el usuario se olvida cerrar su sesión.

### **3.2.2. Modelos Fundamentales**

Las principales entidades del sistema es, la aplicación Android, que toma la información necesaria de una base de datos en MySQL. Estas dos interactúan entre sí utilizando un WSDL (Web Services Descripción Lenguaje).

Con la utilización de un Web Service, distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet, esto significa que los web service aportan interoperabilidad.

La interoperabilidad se consigue mediante la adopción de estándares abiertos. El servicio web que se utilizará está basado en un estándar abierto llamado SOAP (Simple Object Access Protocol) o XML-RPC (XML Remote Procedure Call).

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

- **Modelo de Interacción.-** cuando intentamos tomar información de la base de datos MySQL, se puede producir un retardo por el tiempo de acceso a la red. Otro motivo de retardo depende del ancho de banda, ya que utilizamos internet para poder acceder a diferentes datos. Los sistemas distribuidos que se maneja son síncronos (límite de tiempo), ya que se necesita que la información llegue completa.

- **Modelo de Fallos.-** Muchas de las veces pueden ocurrir fallos en la transferencia de archivos, como por ejemplo que la información no llegue o llegue incompleta, estos fallos se producen por diferentes motivos uno de ellos sería por el ancho de banda, que puede encontrarse con una velocidad demasiado baja y los archivos nunca llegaran a su destino, otro motivo puede ser porque mientras se está recibiendo la información puede desconectarse del internet y no se recibiría la información correctamente. (George Coulouris, Jean Dollimore, 2012)

- **Modelo de Seguridad.-** En este proyecto se podrá hacer modificación de la información (desde Android hacia MySQL), esta información refleja el estado actual del sistema, por lo que se vuelve importante e imprescindible, por lo que al momento de recibir o enviar la información, el usuario deberá estar previamente con permisos que se le asigna cuando acceden al aplicativo con su respectivo usuario y contraseña, y durante el tiempo que es permitido que este activo. De esta manera se protege que personas no deseadas o incluso niños cambien el sistema domótico en una manera no deseada. (George Coulouris, Jean Dollimore, 2012)

### **3.2.3. Comunicaciones TCP/UDP**

Como la información es importante, debemos asegurar la recepción de la información, esto se logrará porque la capa TCP asegura que no existan mensajes perdidos, ya que los vuelve a transmitir.

Con la utilización de un botón invocaremos un web service (utilizando HttpGet). El web service nos retornará una cadena en formato JSON. Ya retornada la cadena JSON, utilizaremos la librería Json que nos ayudará a transformar esta cadena y pasarla a un objeto ArrayList para después utilizar dicha información para desplegar los elementos del sistema en sus diferentes ámbitos.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

## **CAPITULO 4**

### **Implementación y Pruebas**

En el presente capítulo se describirá el funcionamiento real de la aplicación con el sistema domótico encargado de desarrollarse mi compañero Orlando Chacón con su tema de monografía “Análisis para un sistema domótico con la arquitectura Arduino y Raspberry Pi, sobre TCP/IP”, así como también se describirá el proceso de la aplicación desde sus pruebas iniciales con el Web Service, base de datos y servidor local. Se pondrá como evidencia de todas las pruebas realizadas, los errores y las soluciones encontrados al momento de esta integración.

#### **4.1. Implementación de la aplicación**

Para realizar las pruebas de la aplicación, se puso la base de datos y el servicio web de respuesta en un servidor local, el mismo que es una réplica exacta del original que va a estar alojado en el servidor Raspberry Pi. Es de esta manera que la aplicación se fue integrando de una manera homogénea de acuerdo a los requerimientos acordados. Cabe mencionar que la base de datos utilizada es la misma, sin embargo es importante mencionar que la aplicación no tendrá acceso directo a todas las tablas, sino que el servidor Raspberry Pi es el encargado de interpretar las solicitudes enviadas por la aplicación, para que gestione las acciones a los diferentes actuadores según sea lo solicitado por la aplicación.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

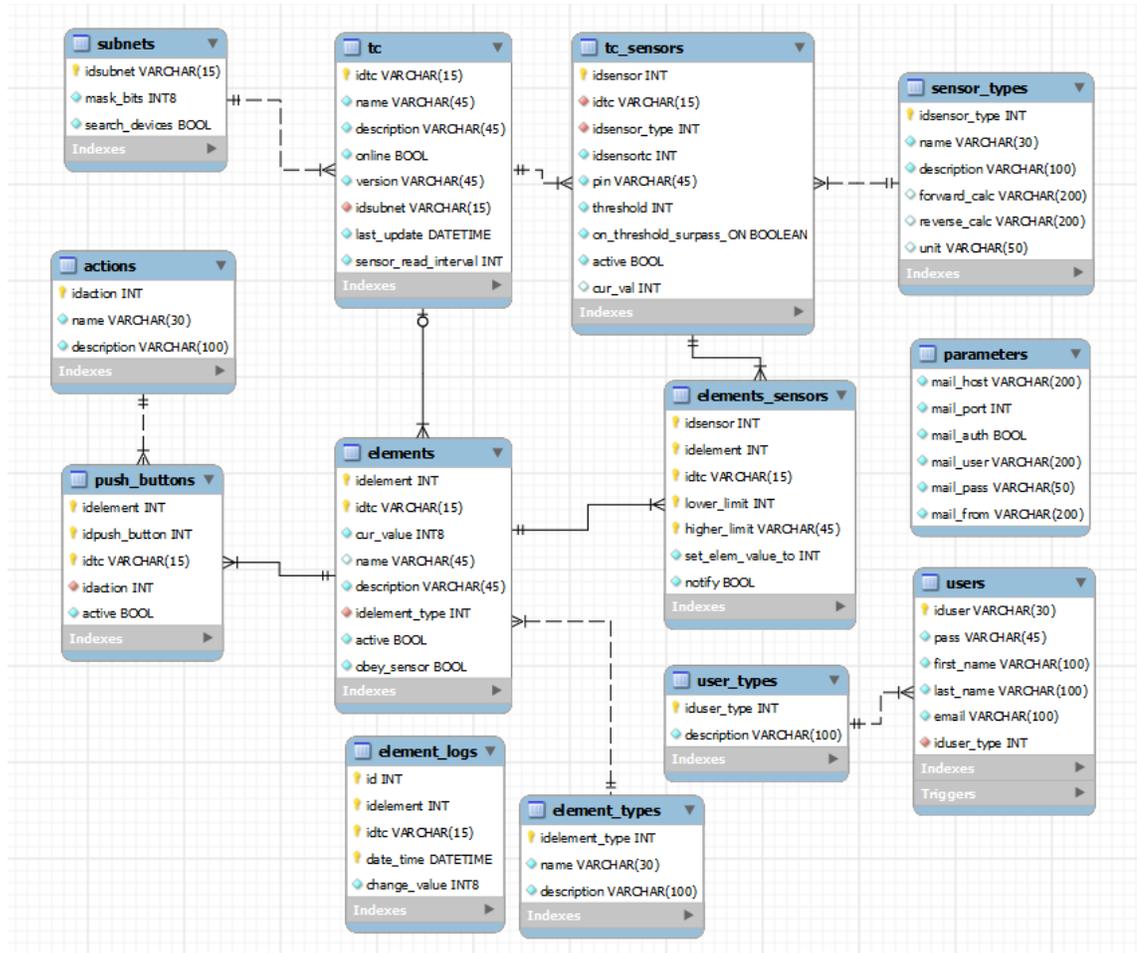


Ilustración #5 – Estructura de la Base de Datos del Sistema

La aplicación recibirá la información real de los elementos que el sistema posee, para de esta manera reflejarlo. Es por ello que se necesitará tener conocimiento de cuatro tablas específicas para que la aplicación pueda gestionar el sistema.

La tabla 'users' servirá para que la aplicación tenga restricción de acceso, es decir no todo el que tenga la aplicación puede controlar las acciones de la misma, es por ello que se solicitará la identificación, y únicamente los usuarios registrados en esta tabla tendrán permisos para gestionar la aplicación. Esta tabla tiene definido el campo administrator, el cual nos indica que usuarios son los administradores, es decir los que podrán controlar todo el sistema domótico, mientras que los usuarios que no son administradores no tendrán acceso a ciertas configuraciones dentro de la aplicación.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

iduser	pass	first name	last name	email	iduser type
belen	e2b256574907781e599f74071a00cafe	Belen	Arizaga	belen@belen.com	1
orlando	fef7185dd6e69d561bc286f3fe6e391f	Orlando	Chacon	orlando@igotel.net	1
juan	jua	Juan	Perez	juan	2

Ilustración #6 – Tabla users

La tabla 'elements' es la que tiene todos los elementos actuales que el sistema tiene registrado, la misma que por sus referencias proporciona la información correspondiente a su estado actual, es decir el valor que tiene, la descripción del elemento, el tipo de elemento que se puede conocer de acuerdo a la tabla 'element\_types', el idtc que es el identificador con el cual el servidor va a saber a qué actuador debe ir, está en la tabla 'tc' de la cual se toma también el valor online para saber si un elemento se dañó o se retiró, lo cual hace de la aplicación una muestra real del estado actual del sistema.

idelement	idtc	value	name	description	idelement type
1	192.168.0.1	0	Lampara	Lampara Garage	2
2	192.168.0.1	1	Lampara	Lampara Patio	2

Ilustración #7 – Tabla elements

idtc	name	description	online	version	idsubnet	last update	sensor read interval
192.168.0.1	outside	outside	1	1	192.168.0.0	2014-01-01	50
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Ilustración #8 – Tabla tc

Como se mencionó la tabla 'element\_types' proporciona la información referente al tipo de elemento que es, con esto se sabe qué valor recibe y a qué elemento se hace el tratamiento, con lo cual se refleja y se posiciona correctamente en cada ámbito. Se tiene 6 tipos de elementos, el uno de propósito general de encendido/apagado, el segundo para el control de luces encendidas/apagadas, el tres para el control de riego encendido/apagado, el cuatro control de climatización encendido/apagado; todos estos con valores de 0 o 1 para simular el encendido/apagado, el cinco para el control de acceso se abierto/cerrado, el 129 para el control de iluminación, e cual recibe valores entre 0 y 7.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

idelement type	name	description
1	ON/OFF General Purpouse	ON/OFF General Purpouse
2	ON/OFF Lighning	ON/OFF Lighning
3	ON/OFF AC	ON/OFF AC
4	ON/OFF Watering	ON/OFF Watering
5	AccessControl	AccessControl
129	+/- Lighning	+/- Lighning
NULL	NULL	NULL

Ilustración #9 – Tabla element\_types

Para obtener la información que se necesita para que la aplicación pueda funcionar, se hace el llamado al Servidor Web que devuelve todos los elementos que se van a gestionar con sus valores reales para que la aplicación pueda reflejarlo de acuerdo a lo que sucede. La respuesta del Servicio Web es en formato Json, esta información es interpretada y obtenida por la aplicación para realizar las acciones pertinentes.

```
[{"idtc":"192.168.1.177","idelement":1,"description":"DimmerB1","idelementType":129,"value":7,"online":true,"obeySensor":true}, {"idtc":"192.168.1.177","idelement":4,"description":"Lamp1","idelementType":2,"value":0,"online":true,"obeySensor":true}, {"idtc":"192.168.1.177","idelement":5,"description":"AC","idelementType":3,"value":1,"online":true,"obeySensor":true}, {"idtc":"192.168.1.177","idelement":7,"description":"PuertaPrincipal","idelementType":5,"value":0,"online":true,"obeySensor":true}, {"idtc":"192.168.1.181","idelement":1,"description":"DimmerB1","idelementType":129,"value":7,"online":true,"obeySensor":true}, {"idtc":"192.168.1.181","idelement":4,"description":"DimmerB2","idelementType":129,"value":0,"online":true,"obeySensor":true}, {"idtc":"192.168.1.181","idelement":7,"description":"Acceso","idelementType":5,"value":1,"online":true,"obeySensor":true}]
```

Ilustración #10 – Respuesta del Servicio Web

Es aquí donde hacemos el uso de las librerías propias de Android, mediante una petición de tipo HttpGet, obtenemos todos los datos referentes a los elementos del sistema, como se muestra en la figura anterior, adicional se utiliza la librería decodificadora Json para poner los elementos en un arreglo del tipo de los elementos, que en este caso es una clase llamada elements que tiene los atributos de respuesta del servidor. Es necesario explicar que la aplicación se actualiza al estado real cada 5 segundos, tiempo que se ha considerado prudente después de algunas pruebas de escritorio.

Mediante una petición tipo HttpPost, enviamos el usuario y la contraseña para el logeo, la misma que nos retorna una respuesta de correcto o incorrecto para saber si es un usuario valido, también nos retorna un campo para saber si el usuario que ingresó es administrador, también cuando queramos cambiar algún estado del sistema por medio de la aplicación, mandamos los valores respectivos que identifican al elemento, y cuál es el valor al que se está actualizando la aplicación.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

## **4.2. Implementación de la aplicación con el sistema**

Para la implementación de la aplicación con el sistema, se procedió a configurar y cambiar la aplicación para que el ingreso de la dirección IP o dominio se pueda modificar según la necesidad de la aplicación. Así mismo se vio la necesidad de que el inicio de sesión se haga en la pantalla inicial, y adicional se tenga un control de tiempo para validar la duración de la sesión.

En cuanto a lo visual se modificaron unos aspectos para que la aplicación sea más sencilla de manejar y de uso práctico para el usuario. Con la implementación en tiempo real, se pudo observar pequeños retrasos en el control de los elementos, por lo que se optimizó el código para una mejor reacción, quedando así un tiempo prudente para la reacción de los elementos. Se observó además que la aplicación se colgaba varias veces en ciertos puntos donde el usuario interactuaba buscando una respuesta rápida, por lo que se controló para que la aplicación responda únicamente cuando ha terminado todas sus tareas pendientes. Obteniendo como resultado una aplicación con control de fallos, y que asegure una respuesta correcta y efectiva. La aplicación fue puesta para varias pruebas, con las mismas se encontraron problemas de iteración o de códigos erróneos, lo que fue de gran ayuda ya que estos se solucionaron con éxito. Así mismo se añadieron funcionalidades necesarias, como el control del umbral de la temperatura, y también se bloquearon ciertas funcionalidades para los usuarios que no poseen el permiso de administrador, esto ha hecho de la aplicación una más robusta que protege que personas no autorizadas como los niños manipulen esta información delicada, que puede ocasionar daños o perjuicios en el sistema. Con lo cual se obtiene como resultado una aplicación bastante estable, con muchas consideraciones y mejoras que se pueden realizar a futuro si se desea que crezca el sistema.

Para entender el funcionamiento y el resultado de la aplicación se ha puesto como anexo el manual de la aplicación y el código fuente.

## CAPÍTULO 5

### Recomendaciones y conclusiones

#### 5.1 Conclusiones.

Después de haber realizado la aplicación, con sus respectivas pruebas e implementación real con el sistema domótico se ha podido conocer todas las ventajas y desventajas de haber elegido software y hardware libre. Concluyo que:

- Android es un sistema operativo muy extenso y al ser una plataforma de código abierto tiene mucho soporte en línea, lo cual es una gran ventaja para el desarrollo de aplicaciones móviles.
- La manera más común para el desarrollo de aplicaciones móviles en Android es a través del lenguaje Java, por lo cual se tiene muchas funcionalidades y posibles combinaciones de código, es por esta misma razón que muchas cosas podrían ser automatizadas o más sencillas para los desarrolladores, en vez de esto se tiene que realizar procedimientos y funciones propias.
- Android puede comunicarse con servicios web, lo que permitió realizar la implementación con el servidor central del sistema domótico, para consumir y enviar la información (comunicación), para obtener como resultado una aplicación personalizada del sistema. Sin embargo, es importante considerar el tiempo de respuesta de las peticiones para que la aplicación valide los bloqueos en el tiempo de espera.
- Es importante realizar las pruebas al momento de la implementación, con los resultados obtenidos se pudo mejorar y solucionar errores que impedían el entendimiento y la funcionalidad esperada por el usuario.
- Es importante considerar que esta aplicación es un prototipo que puede ser mejorada en aspectos como: el tiempo de respuesta, incremento de funcionalidades, un control más avanzado por parte de los usuarios, ya que solo así se puede conseguir una aplicación estable y que tenga una interfaz amigable con el usuario.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

## **5.2 Recomendaciones.**

Después de haber trabajado con Android y los servicios web por los últimos 3 meses se puede recomendar lo siguiente:

- En la web se puede encontrar mucha información de implementación de código que no necesariamente es la adecuada, por lo que se recomienda hacer las pruebas necesarias antes de utilizarlo para verificar su veracidad.
- Tomarse el tiempo necesario para realizar un diseño de interfaz correcto, la parte visual es el factor principal de atracción e interacción con los usuarios para que la aplicación tenga éxito.
- Considerar que el Internet es el factor principal e indispensable para que la aplicación pueda funcionar y obtener la información (la data) necesaria.
- Se recomienda el uso de Arduino, Raspberry Pi, y Android para personas que les gusta experimentar con aplicaciones y funcionalidades nuevas, es decir que este sistema con aplicativo pequeño se puede implementar en sus propios hogares.
- No se recomienda el uso de estas herramientas para desarrolladores que buscan implementar soluciones para la venta, ya que se tienen limitaciones de alcance, de tiempo de respuesta y de conexión.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

## **ANEXOS**

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

## Anexo #1 - Manual de Usuario

Al instalar la aplicación en nuestro dispositivo móvil, la podemos encontrar con el siguiente logo y con el nombre de RDomó.

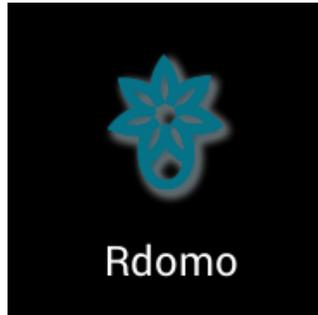


Ilustración #11 –Logo Rdomo

Si es la primera vez que abrimos la aplicación, o nuestra sesión ha caducado nos aparecerá la pantalla de logueo, la misma que nos permite ingresar el dominio o la página a la cual nos vamos a conectar para obtener la data, seguido podemos ingresar el usuario y la contraseña de nuestro sistema para poder acceder una vez que ha sido configurado.

### Pantalla de Inicio de Sesión

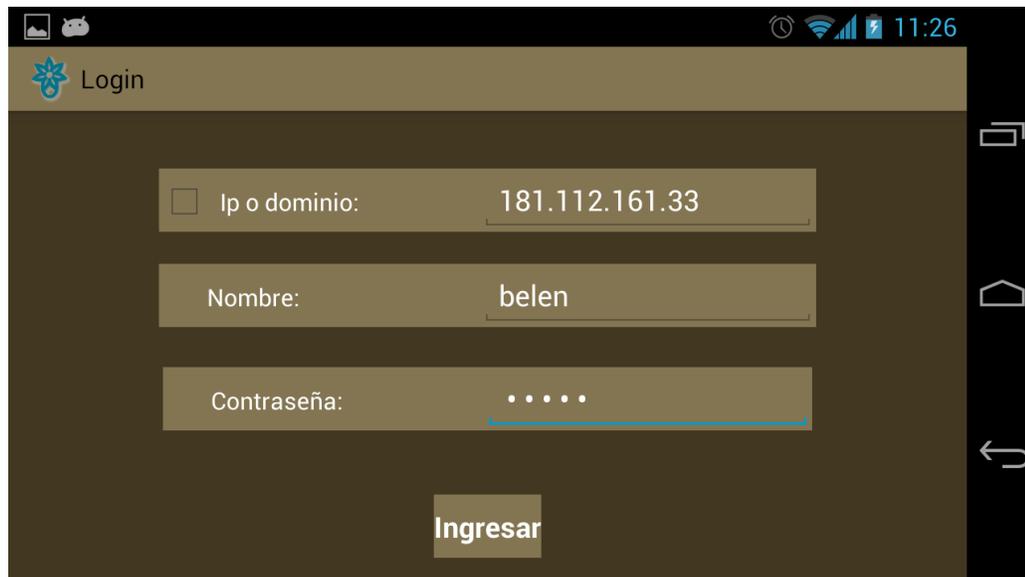


Ilustración #12 – Pantalla de Logueo

La aplicación responderá con tres mensajes de error, el primero es si el usuario o la contraseña están vacíos, otro es si el dominio o la dirección IP no es correcta, es decir no se puede conectar a ese servidor para obtener la data, el otro si es que el usuario o la contraseña no

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

coinciden con los datos almacenados en el servidor. Estos son los mensajes que la aplicación desplegará.

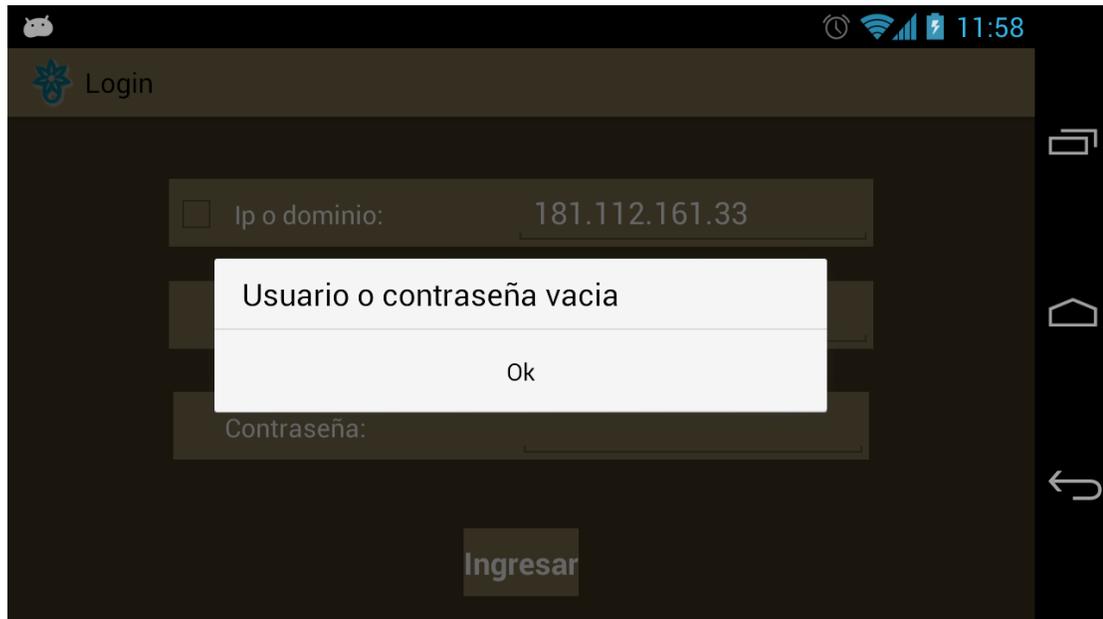


Ilustración #13 – Error de usuario o contraseña vacíos

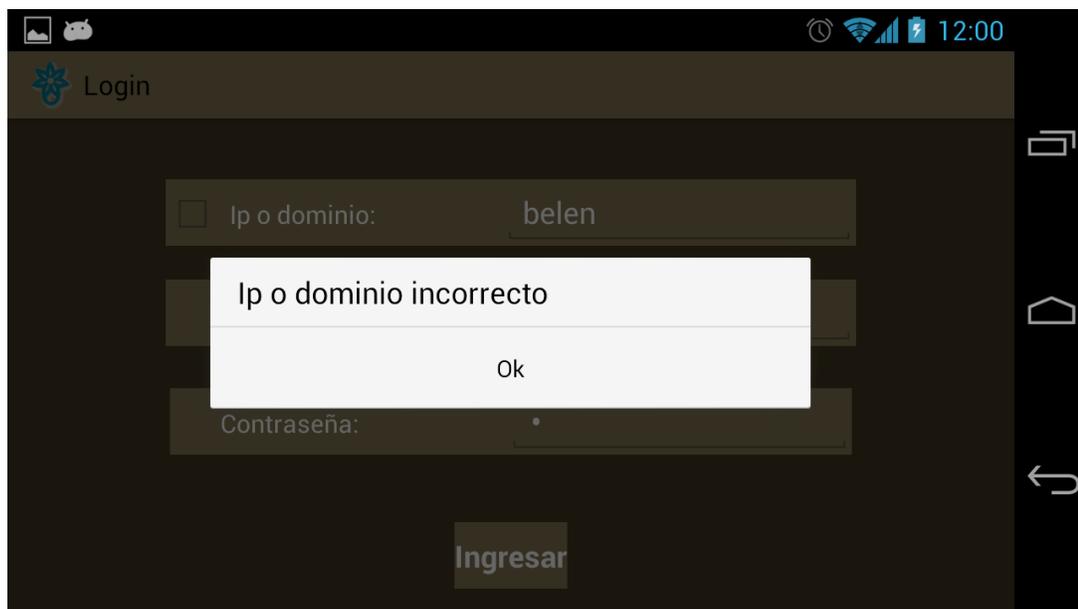


Ilustración #14 – Error de IP o dominio

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

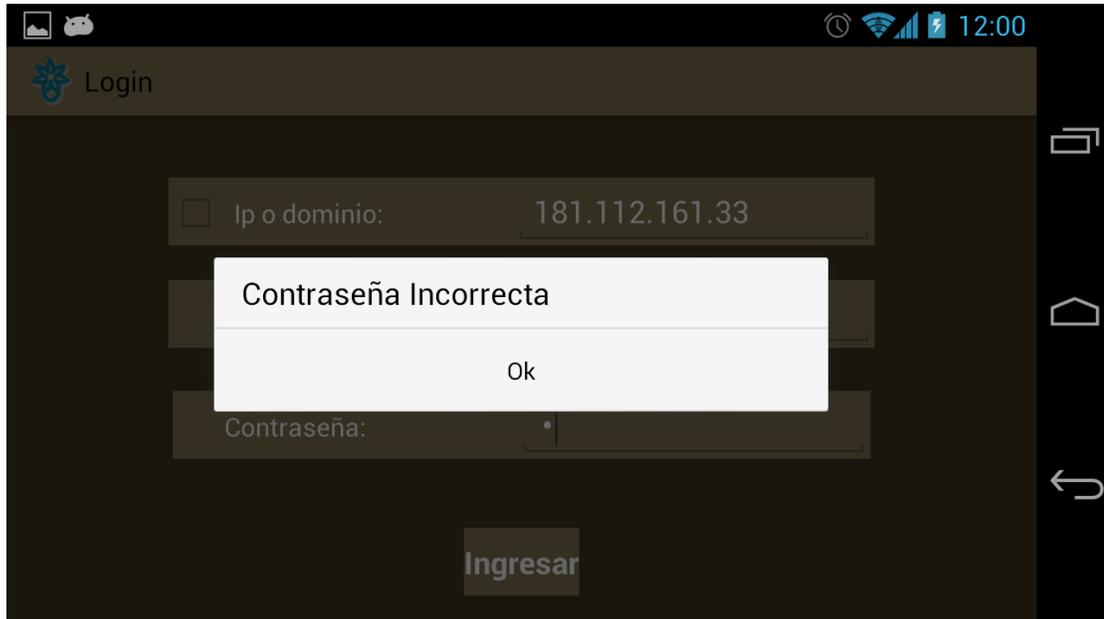


Ilustración #15 – Error de usuario o contraseña

Una vez que sea ingresado correctamente la información vamos a la siguiente pantalla, la cual es la que gestiona todo el sistema y la que nos muestra todas las acciones posibles.

Pantalla del Menú

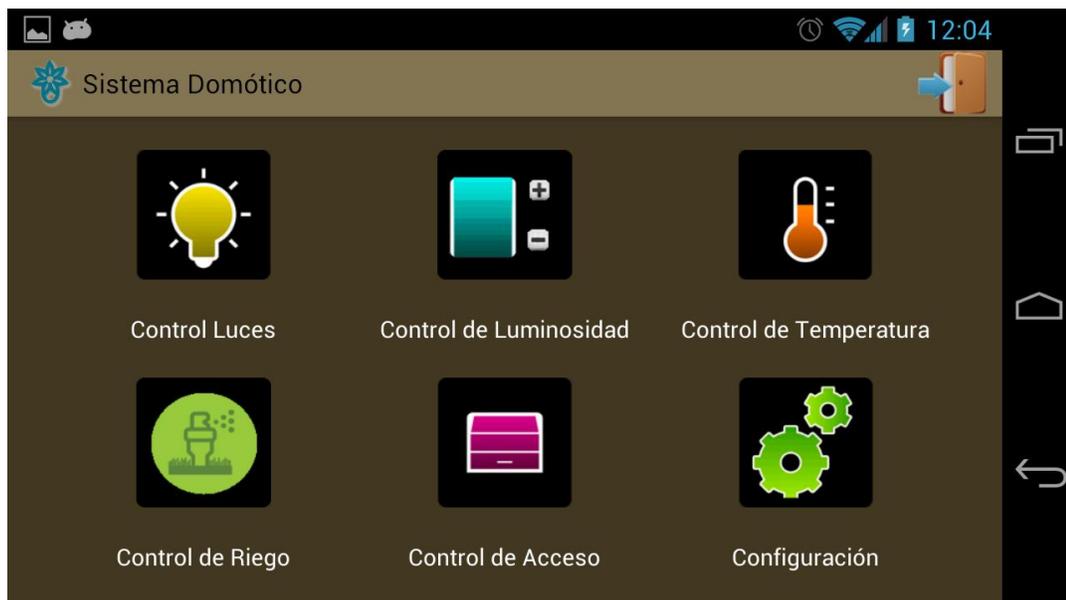


Ilustración #16 – Pantalla del Menú de la Aplicación

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

En el menú podemos encontrar los accesos para el control de luces, luminosidad, temperatura, riego, acceso y la configuración de la cuenta del usuario. Adicional como vemos en la barra de menú tenemos el icono para cerrar la sesión actual en el momento que lo deseemos.

#### Pantalla de Control de Luces

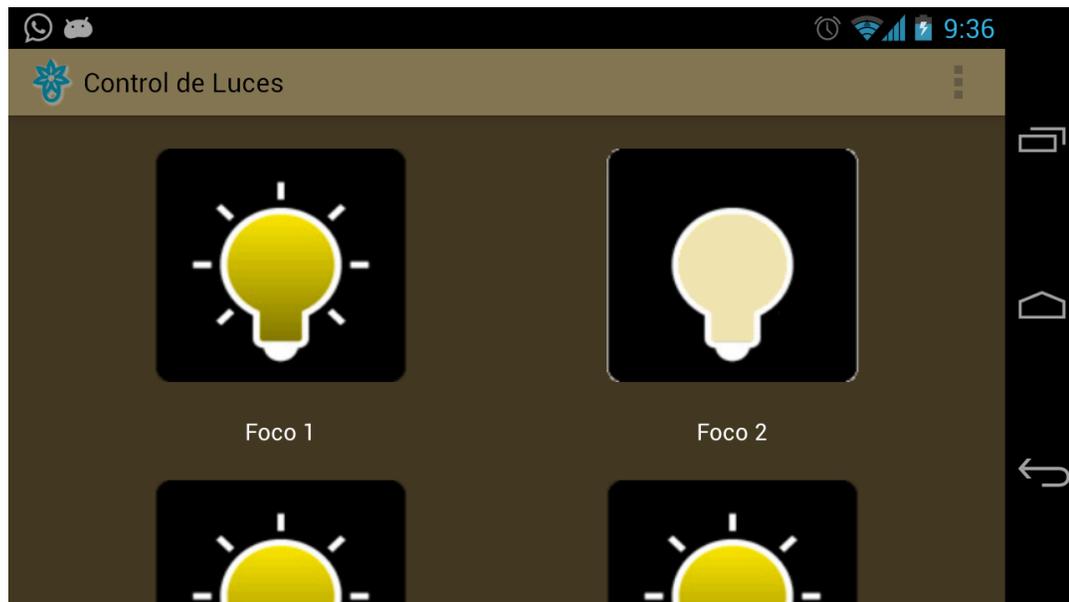


Ilustración #17 – Pantalla del Control de Luces

En esta pantalla se nos desplegará un estado real de todos los focos que estén conectados a nuestro sistema, teniendo además una interfaz muy amigable con el usuario, ya que nos muestra una imagen si es que el foco se encuentra prendido y otra imagen que representa el foco apagado, y adicional nos muestra la descripción que nos permitirá saber a qué foco del sistema pertenece. Podemos cambiar el estado de los elementos, encenderlos o apagarlos de acuerdo a nuestra necesidad con tan solo tocar cada elemento, así mismo si los focos son cambiados de estado directamente del sistema el estado será reflejado en la pantalla. Adicionalmente se tiene la implementación de un pulsado largo para que se activen o se desactiven el o los sensores que se encuentren enlazados a cada ítem.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

### Pantalla de Control de Luminosidad

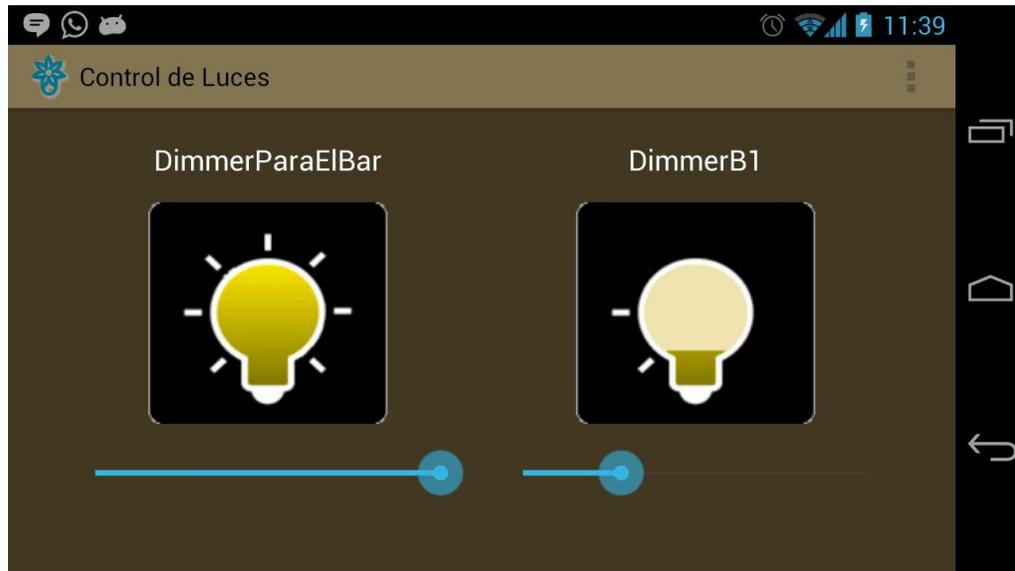


Ilustración #18 – Pantalla del Control de Luminosidad

Esta pantalla mostrara todos los elementos del tipo dimmer (cambio de intensidad de luz deseada) que se encuentren en el sistema, en la cual se tiene una clara visión de lo que sucede en el sistema, con ocho imágenes que representan el estado de los focos desde cero apagado hasta siete prendido al máximo, y cuenta con una barra para poder elegir la intensidad que se desea en cada elemento, sabiendo a que elemento del sistema pertenece de acuerdo a su descripción. Además se tiene la opción de activar o desactivar los sensores con un pulso largo sobre cada elemento, para saber si obedece o no a los sensores que obedecen a la intensidad de la luz que el sistema detecta.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

### Pantalla de Control de Temperatura

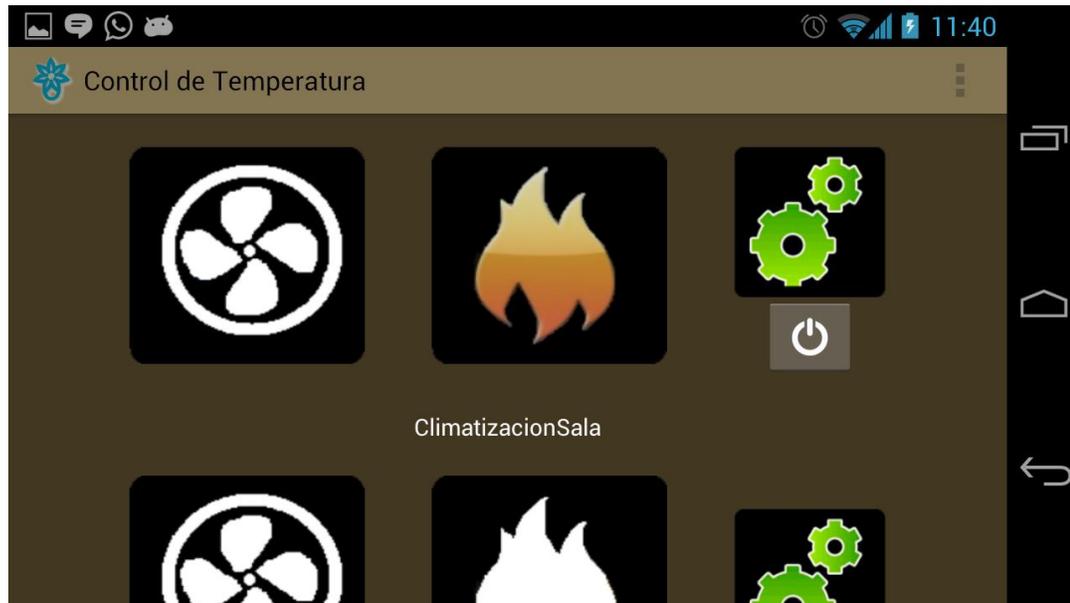


Ilustración #19 – Pantalla del Control de Temperatura

La pantalla de control de temperatura se encarga de gestionar un sistema, que puede ser separado por sectores o en un solo lugar de acuerdo a la configuración del sistema, la cual nos muestra cada gestión con el calefactor o el aire acondicionado, los mismos que se activan de acuerdo a los umbrales de temperatura establecidos. Lo que podemos realizar en esta pantalla es un encendido o apagado de los sensores del control de temperatura, entonces dependiendo de los umbrales que podemos configurar en el icono de configuración (tuercas) se va a encender o apagar el calefactor o aire acondicionado según lo leído, así mismo si la temperatura es la idónea .

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

### Control de Umbrales de la Temperatura

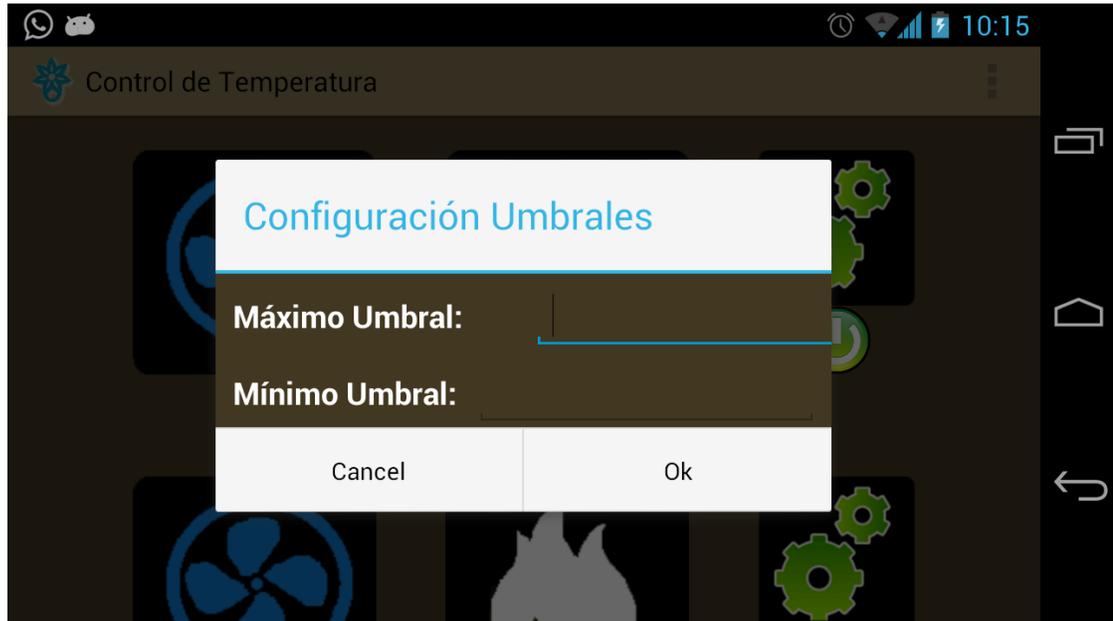


Ilustración #20 – Control de los Umbrales de Temperatura

Este dialogo nos permite elegir el umbral deseado para cuando el sensor de temperatura este activado, sea este el que elija según la configuración que elemento se prende o se apaga para lograr la temperatura deseada.

### Pantalla de Control de Riego

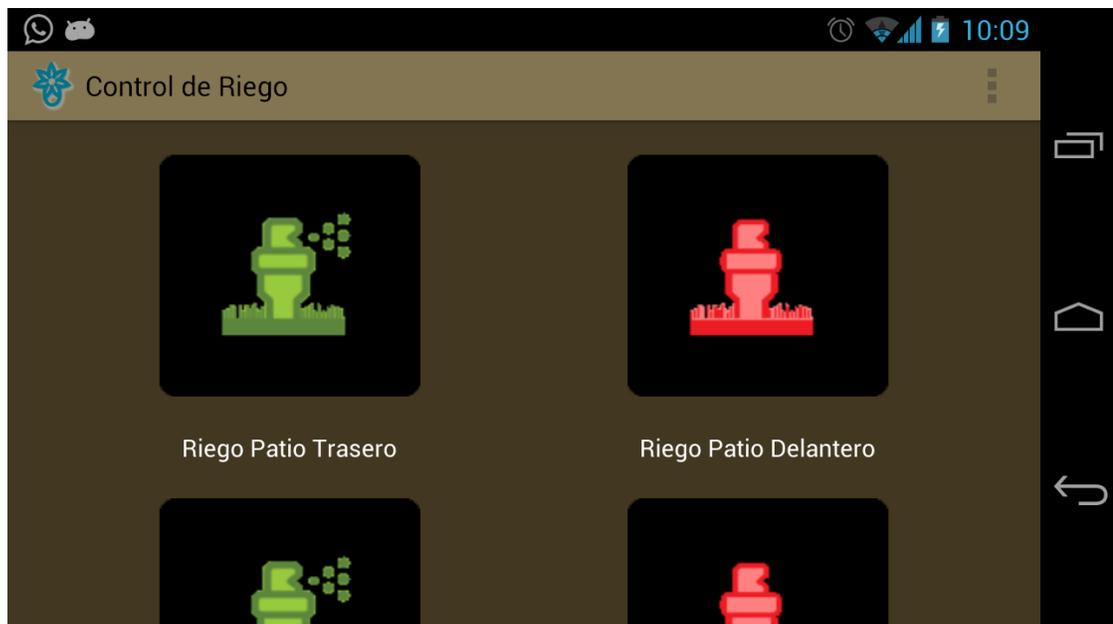


Ilustración #21 – Pantalla de Control de Riego

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

La pantalla de riego nos muestra dos tipos de imágenes, una en verde que nos indica que el riego de ese lugar esta encendido y así mismo una imagen en rojo significa que el control de riego está apagado.

#### Pantalla de Control de Acceso

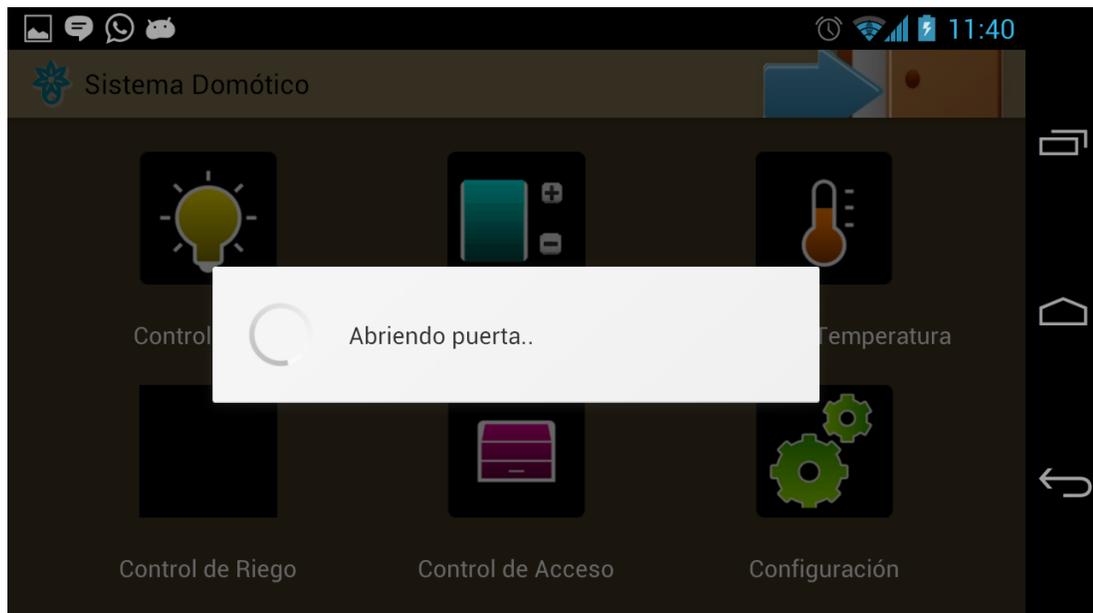


Ilustración #22 – Control de Acceso, Abriendo Puerta

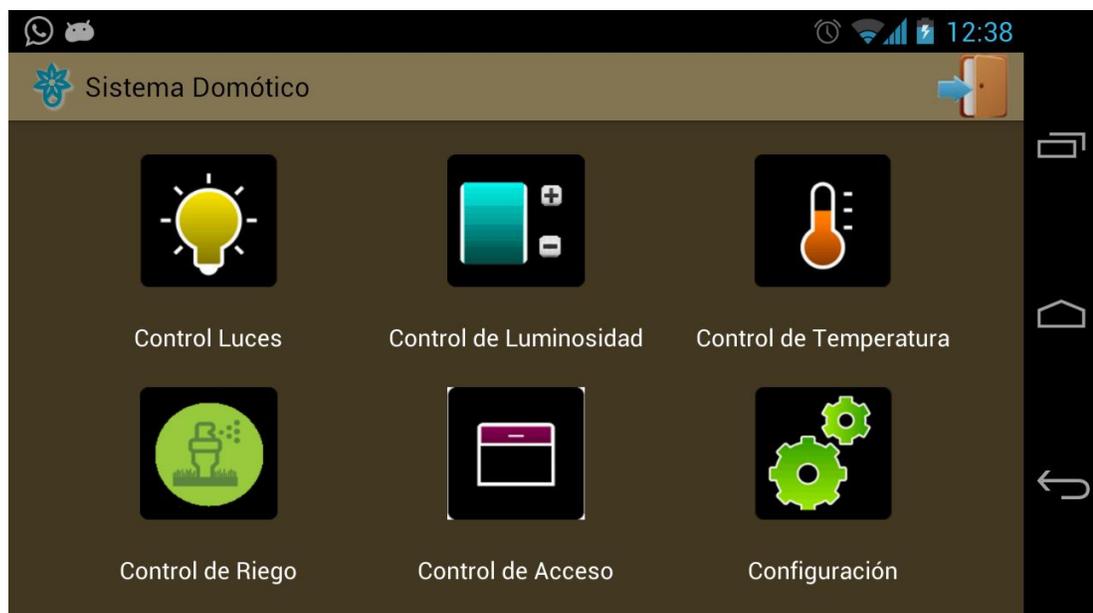


Ilustración #23 – Control de Acceso, Puerta Abierta

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

El control de acceso se lo maneja directamente en la pantalla de menú puesto que hace referencia a una sola puerta que va a ser la principal del sistema, al pulsar en el icono se puede observar un mensaje de que la puerta se está abriendo, y luego se observa una imagen con la puerta abierta, la misma que permanecerá hasta que la aplicación reciba la notificación que la puerta ha sido cerrada, inmediatamente la imagen regresara a su estado normal.

#### Pantalla de Configuración del Tiempo del Usuario

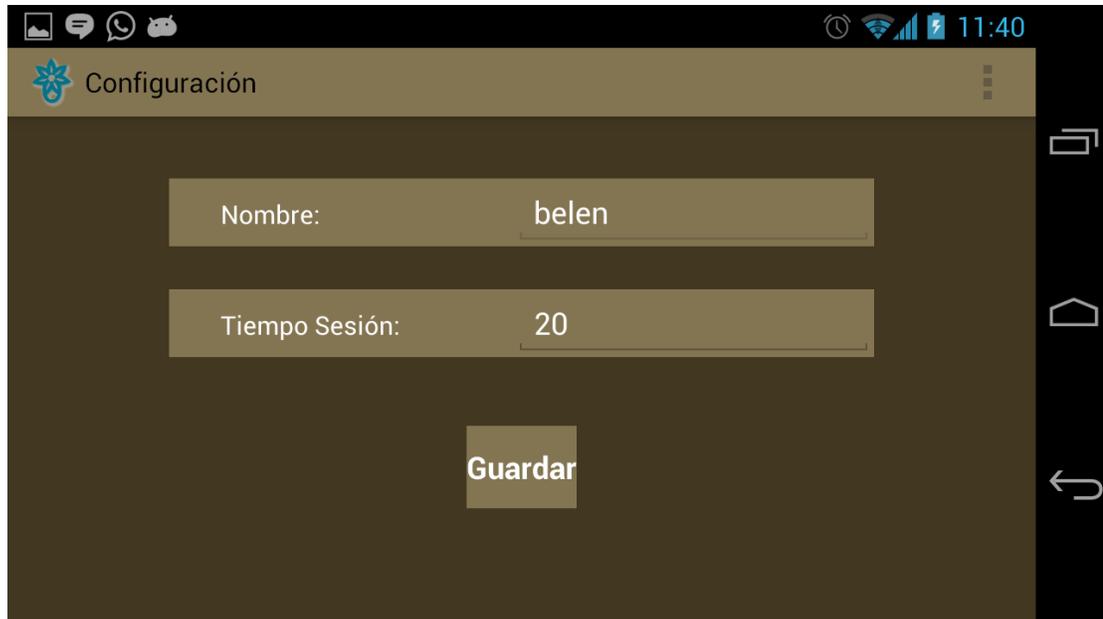


Ilustración #24 – Configuración de Sesión del Usuario

En esta aplicación se puede configurar el tiempo de actividad que se desea que tenga la cuenta, lo cual ayuda a que se cierre la sesión si es que olvidamos aplastar el botón de cerrar sesión, es decir que si pasa algo con nuestro teléfono podemos tener la confianza de que la aplicación está segura ya que cumple el tiempo en minutos.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

## Anexo # 2 – Código de la aplicación

Método POST para iniciar session.

```
private class Login extends AsyncTask<String, Void, Object>{

    @Override
    protected Integer doInBackground(String... params) {
        // TODO Auto-generated method stub
        String url = in.obtenerUrl();
        if (!url.equalsIgnoreCase(""))
        {
            String user = txtNombre.getText().toString();
            String pass = txtPass.getText().toString();

            url="Http://" + in.obtenerUrl() + "/api/ws.php?res="+ "login";
            try {

                HttpClient client = new
DefaultHttpClient();
                HttpPost post = new HttpPost(url);
                MultipartEntity mpEntity = new
MultipartEntity();
                mpEntity.addPart("user", new
StringBody(user,
Charset.forName("UTF-8")));
                mpEntity.addPart("pass", new
StringBody(pass,
Charset.forName("UTF-8")));

                post.setEntity(mpEntity);
                //Execute the post request
                HttpResponse responsel =
client.execute(post);

                //Get the response from the server
                HttpEntity resEntity = responsel.getEntity();
                Response = EntityUtils.toString(resEntity);
                Log.i("Response", Response);
                //Close the connection
                client.getConnectionManager().shutdown();

            } catch (Exception ex) {
                // TODO Auto-generated catch block
                ex.printStackTrace();
            }
        }
        return 1;
    }
    protected void onPostExecute(Object result){
        pd.dismiss();
        if (Response.equalsIgnoreCase("OK")) {
            in.asignarFecha(txtNombre.getText().toString(),
new Date());
            in.asignarTiempo(20);
            Intent intent = new Intent(MainActivity.this,
MenuActivity.class);
            startActivity(intent);
        }
    }
}
```

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

```

        } else {
            if (Response.equalsIgnoreCase("ERROR")) {
                AlertDialog.Builder alertbox = new
AlertDialog.Builder(
                    MainActivity.this);
                alertbox.setMessage("Contraseña
Incorrecta");
                alertbox.setNegativeButton("Ok", new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface
dialog, int whichButton) {
                        // Canceled.
                        dialog.cancel();
                    }
                });
                alertbox.show();
            }
            else{
                AlertDialog.Builder alertbox = new
AlertDialog.Builder(
                    MainActivity.this);
                alertbox.setMessage("Ip o dominio
incorrecto");
                alertbox.setNegativeButton("Ok", new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface
dialog, int whichButton) {
                        // Canceled.
                        dialog.cancel();
                        chk.setChecked(false);
                        mostrarDominio();
                    }
                });
                alertbox.show();
            }
        }
        super.onPostExecute(result);
    }
}

```

Método de llamada GET para obtener todos los elementos del sistema por el servicio web.

```
private class obtenerElementos extends AsyncTask<String, Void, Object>{
```

```

    @Override
    protected Integer doInBackground(String... params) {
        // TODO Auto-generated method stub
        elementos.clear();
        String url = in.obtenerUrl();
        if (!url.equalsIgnoreCase(""))
        {
            url="Http://" + in.obtenerUrl() + "/api/ws.php?res="+ "getelementos";
            HttpClient httpClient = new
DefaultHttpClient();
            HttpGet httpGet = new HttpGet(url);

```

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

```

        httpGet.setHeader("content-type",
"application/json");
        try{
            HttpResponse resp =
httpClient.execute(httpGet);
            String respStr =
EntityUtils.toString(resp.getEntity());
            JSONArray respJSON = new
JSONArray(respStr);

            for(int i=0;i<respJSON.length();i++){
                JSONObject obj =
respJSON.getJSONObject(i);
                String id_tc =
obj.getString("idtc");
                int id_element =
obj.getInt("idelement");
                String description =
obj.getString("description");
                int id_element_type =
obj.getInt("idelementType");
                int value = obj.getInt("value");
                boolean online =
obj.getBoolean("online");
                boolean sensor =
obj.getBoolean("obeySensor");

                if(id_element_type==2)
                    elementos.add(new
elementos(id_tc,id_element,description, id_element_type, value, online,
sensor));
            }
        }
        catch(Exception ex){
            Log.e("SERVICIO WEB SERVICE",
"Error!",ex);
        }
    }
    return 1;
}

protected void onPostExecute(Object result){
    //pd.dismiss();
    if(elementos.size()==0)
    {
        Toast.makeText(getApplicationContext(), "No hay
elementos disponibles", Toast.LENGTH_SHORT).show();
    }
    mostrarElementosDisponibles();

    super.onPostExecute(result);
}
}

```

Método para cambiar el estado de un elemento

```
private class encenderFoco extends AsyncTask<String, Void, Object>{
```

```
@Override
```

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

```

        protected Integer doInBackground(String... params) {
            // TODO Auto-generated method stub
            String url = in.obtenerUrl();
            if(!url.equalsIgnoreCase(""))
            {
                url="Http://"+in.obtenerUrl()+"/api/ws.php?res=setvalue&groupid="
+
                elementos.get(id).getId_element()+"&idtc="+elementos.get(id).getId_tc()+
                "&value="+valor;
                //Log.i("URL",url);
                HttpClient httpClient = new
DefaultHttpClient();
                HttpGet httpGet = new HttpGet(url);
                httpGet.setHeader("content-type",
"application/json");
                try{
                    HttpResponse resp =
httpClient.execute(httpGet);
                    String respStr =
EntityUtils.toString(resp.getEntity());
                }
                catch(Exception ex){
                    Log.e("SERVICIO WEB SERVICE",
"Error!", ex);
                }
            }
            return 1;
        }

        protected void onPostExecute(Object result){
            //new obtenerElementos().execute("");
            super.onPostExecute(result);
        }
    }
}

```

Método para activar o desactivar los sensores.

```

private class activarSensor extends AsyncTask<String, Void, Object>{

    @Override
    protected Object doInBackground(String... arg0) {
        // TODO Auto-generated method stub
        String url = in.obtenerUrl();
        if(!url.equalsIgnoreCase(""))
        {
            url="Http://"+in.obtenerUrl()+"/api/ws.php?res=shiftoBeySensor&gr
oupid="+
            elementos.get(id).getId_element()+"&idtc="+elementos.get(id).getId_tc();
            //Log.i("URL",url);
            HttpClient httpClient = new
DefaultHttpClient();

```

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

```

        HttpGet httpGet = new HttpGet(url);
        httpGet.setHeader("content-type",
"application/json");
        try{
            HttpResponse resp =
httpClient.execute(httpGet);
            String respStr =
EntityUtils.toString(resp.getEntity());
        }
        catch(Exception ex){
            Log.e("SERVICIO WEB SERVICE",
"Error!",ex);
        }
    }
    return 1;
}
protected void onPostExecute(Object result){
    Toast.makeText(getApplicationContext(), "Cambio de
estado del sensor", Toast.LENGTH_SHORT).show();
    super.onPostExecute(result);
}
}

```

### Método para obtener los elementos (Servicio Web)

```

function getAllFromDB($scope="all",$format="array"){
    if($scope=="this"){
        $mydb=new mysqlConn("SELECT
        e.idtc,
        e.idelement,
        e.description,
        e.idelement_type,
        IF(
        sen.cur_val IS NULL,
        e.cur_value,
        IF(
        sen.cur_val > (sen.threshold * 4)
        AND on_threshold_surpass_ON = 1,
        1, 0
        )
        ) AS cur_value,
        tc.online,
        e.obey_sensor
        FROM
        elements e
        LEFT OUTER JOIN
        (SELECT
        t.`idtc`,
        e1.`idelement`,
        ts.`cur_val`,
        ts.`threshold`,
        ts.`on_threshold_surpass_ON`
        FROM
        `tc` t,
        `tc_sensors` ts,

```

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

```

        `elements` e1
        WHERE t.`idtc` = ts.`idtc`
        AND t.`idtc` = e1.`idtc`
        AND ts.`pin` = e1.`idelement`
        AND e1.`idelement_type` = 5) AS sen
        ON e.`idtc` = sen.`idtc`
        AND e.`idelement` = sen.`idelement`, tc
        WHERE e.active = 1
        AND tc.idtc = e.idtc
        AND tc.idtc=?;");
        $mydb->bindParam(1,$this->idtc);
    }
    if($scope=="online"){
        $mydb=new mysqlCnn("SELECT
        e.idtc,
        e.idelement,
        e.description,
        e.idelement_type,
        IF(
            sen.cur_val IS NULL,
            e.cur_value,
            IF(
                sen.cur_val > (sen.threshold * 4)
                AND on_threshold_surpass_ON = 1,
                1, 0
            )
        ) AS cur_value,
        tc.online,
        e.obey_sensor
        FROM
        elements e
        LEFT OUTER JOIN
        (SELECT
            t.`idtc`,
            e1.`idelement`,
            ts.`cur_val`,
            ts.`threshold`,
            ts.`on_threshold_surpass_ON`
        FROM
            `tc` t,
            `tc_sensors` ts,
            `elements` e1
        WHERE t.`idtc` = ts.`idtc`
        AND t.`idtc` = e1.`idtc`
        AND ts.`pin` = e1.`idelement`
        AND e1.`idelement_type` = 5) AS sen
        ON e.`idtc` = sen.`idtc`
        AND e.`idelement` = sen.`idelement`,
        tc
        WHERE e.active = 1
        AND tc.idtc = e.idtc
        AND tc.online = ?;");
        $mydb->bindParam(1,1);
    }elseif($scope=="offline"){
        $mydb=new mysqlCnn("SELECT
        e.idtc,

```

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

```

e.idelement,
e.description,
e.idelement_type,
IF(
  sen.cur_val IS NULL,
  e.cur_value,
  IF(
    sen.cur_val > (sen.threshold * 4)
    AND on_threshold_surpass_ON = 1,
    1, 0
  )
) AS cur_value,
tc.online,
e.obey_sensor
FROM
elements e
LEFT OUTER JOIN
  (SELECT
    t.`idtc`,
    e1.`idelement`,
    ts.`cur_val`,
    ts.`threshold`,
    ts.`on_threshold_surpass_ON`
  FROM
    `tc` t,
    `tc_sensors` ts,
    `elements` e1
  WHERE t.`idtc` = ts.`idtc`
    AND t.`idtc` = e1.`idtc`
    AND ts.`pin` = e1.`idelement`
    AND e1.`idelement_type` = 5) AS sen
ON e.`idtc` = sen.`idtc`
AND e.`idelement` = sen.`idelement`,
tc
WHERE e.active = 1
AND tc.idtc = e.idtc
AND tc.online = ?");
$mydb->bindParam(1,0);
}elseif($scope=="all"){
$mydb=new mysqlCnn("SELECT
e.idtc,
e.idelement,
e.description,
e.idelement_type,
IF(
  sen.cur_val IS NULL,
  e.cur_value,
  IF(
    sen.cur_val > (sen.threshold * 4)
    AND on_threshold_surpass_ON = 1,
    1, 0
  )
) AS cur_value,
tc.online,
e.obey_sensor
FROM

```

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

```

elements e
LEFT OUTER JOIN
(SELECT
  t.`idtc`,
  e1.`idelement`,
  ts.`cur_val`,
  ts.`threshold`,
  ts.`on_threshold_surpass_ON`
FROM
  `tc` t,
  `tc_sensors` ts,
  `elements` e1
WHERE t.`idtc` = ts.`idtc`
  AND t.`idtc` = e1.`idtc`
  AND ts.`pin` = e1.`idelement`
  AND e1.`idelement_type` = 5) AS sen
ON e.`idtc` = sen.`idtc`
  AND e.`idelement` = sen.`idelement`,
tc
WHERE e.active = 1
  AND tc.idtc = e.idtc
  AND tc.online = 1;");
}
$mydb->execute();
$this->subnets="";

if($format=="array" or $format=="ssv"){
  $hosts=array();
  while($record = $mydb->getNextRecord()){
    array_push($hosts,$record['idtc']);
  }
}elseif($format=="ssv"){//idtc, Space Separated Values
  $hosts="";
  while($record = $mydb->getNextRecord()){
    $hosts=$record['idtc']." ";
  }
}elseif($format=="json"){
  $jsonArray = array();
  while($record = $mydb->getNextRecord()){
    array_push($jsonArray, array("idtc" =>
$record['idtc'],
    "idelement" => intval($record['idelement']),
    "description" => $record['description'],
    "idelementType" =>
intval($record['idelement_type']),
    "value" => intval($record['cur_value']),
    "online" => ($record['online']>0 ? true :
false),
    "obeySensor" => ($record['obey_sensor']>0 ?
true : false)));
  }
  $hosts=json_encode($jsonArray);
}
return $hosts;
}
}

```

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

## Bibliografía

- Android Developers*. (2008). Recuperado el 27 de 04 de 2014, de Develop, Api Guides:  
<http://developer.android.com/guide/index.html>
- Android Developers*. (2008). Recuperado el 27 de 04 de 2014, de Google services:  
<http://developer.android.com/google/index.html>
- Android Developers*. (2008). Recuperado el 27 de 04 de 2014, de Tools:  
<http://developer.android.com/tools/index.html>
- Arturo Baz Alonso, Irene Ferreira Artime, María Álvarez Rodríguez, Rosana García Baniello. (01 de 06 de 2008). Dispositivos móviles. Obtenido de <http://castor.edv.uniovi.es/~smi/5tm/09trabajos-sistemas/1/Memoria.pdf>
- Delgado Hidalgo Luis, Moreno Demera Walter, Veliz Velastegui Ninfa, Leyton Edgar. (26 de Febrero de 2009). Diseño Para El Control De Un Hogar Inteligente Mediante El Uso Del Sistema Domotico (X-10) Por Medio De La Red Electrica. Guayaquil, Guayas, Ecuador. Recuperado el 2 de Mayo de 2014, de <http://www.dspace.espol.edu.ec/handle/123456789/1068>
- George Coulouris, Jean Dollimore. (2012). *Distributed systems : concepts and design*. Boston: Boston : Pearson. doi: 978-0-13-214301-1
- Google. (2008). *Android developer*. Recuperado el 10 de 05 de 2014, de <http://developer.android.com/reference/packages.html>
- Harke, W. (2010). *Domótica para viviendas y edificios*. Barcelona, España: Marcombo.
- Lauren Darcey, Shane Conder. (2012). *Android Wireless Application Developmrrnt. Volume 1: Android Essentials*. (M. C. Polo, Trad.) doi:978-84-415-3194-9
- Luís Delgado Hidalgo, Walter Moreno Demera, Ninfa Véliz Velástegui, Edgar Leyton. (s.f.). DISEÑO PARA EL CONTROL DE UN HOGAR INTELIGENTE MEDIANTE EL USO DEL SISTEMA DOMOTICO (X-10) POR MEDIO DE LA RED ELECTRICA. Recuperado el 10 de Mayo de 2014, de <http://www.dspace.espol.edu.ec/bitstream/123456789/1068/1/1902.pdf>
- Mdhd., I. D. (2014). Material Para Domótica del Curso de Grado 2014. *Introducción Domótica*. Cuenca, Azuay, Ecuador. Recuperado el 08 de Mayo de 2014
- Purdy., K. (2009). "The complete Android Guide". USA. Recuperado el 07 de 03 de 2014, de [http://www.completeguides.net/01\\_The\\_Complete\\_Android\\_Guide](http://www.completeguides.net/01_The_Complete_Android_Guide)

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

Stefan Junesträd, Xavier Passaret, Daniel Vázquez. (2005). Domótica y Hogar Digital. En X. P. Stefan Junesträd, *Domótica y Hogar Digital* (pág. 244). Madrid, España: EDICIONES PARANINFO, S.A. doi:9788428328913

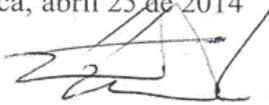
Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

**DOCTOR ROMEL MACHADO CLAVIJO,**  
**SECRETARIO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACION**  
**DE LA UNIVERSIDAD DEL AZUAY,**

**C E R T I F I C A:**

Que, el H, Consejo de Facultad de Ciencias de la Administración en sesión del 24 de abril de 2014, conoció la petición de la señorita ANA BELEN ARIZAGA JARAMILLO (46144) que denuncia su trabajo de titulación denominado: “**DESARROLLO DE UN APLICACIÓN EN ADROID PARA EL MANEJO DE UNA ARQUITECTURA DOMOTICA ESPECIFICA EN EL CONTROL DE ACCESO, RIEGO, ILUMINACION, CLIMATIZACION CON ARDUINO Y RASPBERRY PI**” presentada como requisito previo a la obtención del Grado de Ingeniera de Sistemas. El Consejo acoge el informe de la Junta Académica y aprueba la denuncia. Designa como Director de dicho trabajo al ingeniero Diego Chacón y como miembro del Tribunal Examinador al ingeniero Juan Carlos Salgado Arteaga. De conformidad a las disposiciones reglamentarias la denunciante deberá presentar su trabajo de graduación en un plazo no mayor a **TRES MESES** contados a partir de la fecha de aprobación, esto es hasta el 24 de julio de 2014.

Cuenca, abril 25 de 2014





Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

Oficio Nro. 037-2014-DIST-UDA

Cuenca, 14 de Abril de 2014

Señor Ingeniero

Xavier Ortega Vázquez

DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN

Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 14 de Abril del 2014, revisó la documentación del proyecto de monografía "Desarrollo de una Aplicación Android para el Manejo de un Arquitectura Domótica especificada en el Control de Acceso, Riego, Iluminación, Climatización con Arduino y Raspberry Pi", presentado por la estudiante Belén Arízaga con código 46144, estudiante de la Escuela de Ingeniería de Sistemas, previo a la obtención del título de Ingeniero de Sistemas.

La Junta considera que la documentación cumple con las normas legales y reglamentarias de la Universidad y de la Facultad de Ciencias de la Administración y avala la aprobación por parte del tribunal designado, así por su digno intermedio, el conocimiento y aprobación por parte del Consejo de Facultad.

Atentamente,

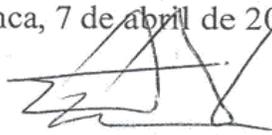
  
Ing. Marcos Orellana Cordero  
Director Escuela de Ingeniería de Sistemas y Telemática  
Universidad del Azuay

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

## CONVOCATORIA

Por disposición de la Junta Académica de Ingeniería de Sistemas y Telemática, **CONVOCO** a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación denominado: **“DESARROLLO DE UNA APLICACIÓN EN ANDROID PARA EL MANEJO DE UNA ARQUITECTURA DOMOTICA ESPECIFICA ENEL CONTROL DE ACCESO, RIEGO, ILUMINACION, CLIMATIZACION CON ARDUINO Y RASPBERRY PI”** presentado por la estudiante **ANA BELEN ARIZAGA JARAMILLO (46144)**, previa a la obtención del grado de Ingeniero de Sistemas, para el día **JUEVES 10 DE ABRIL DE 2014, a las 17h00**

Cuenca, 7 de abril de 2014



Dr. Romel Machado Clavijo  
Secretario de la Facultad

Ing. Diego Chacón



Ing. Juan Carlos Salgado Arteaga



Comunicado  
abm 4.  
62



Facultad de Ciencias de la Administración  
Escuela de Ingeniería de Sistemas y Telemática  
Desarrollo de una aplicación en Android para el manejo de  
una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con  
Arduino y Raspberry Pi.

Oficio Nro. 031-2014-DIST-UDA

Cuenca, 03 de Abril de 2014

Señor Ingeniero  
Xavier Ortega Vázquez  
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN  
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 3 de Abril del 2014, revisó el proyecto de monografía titulado "Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.", presentada por la estudiante Arízaga Jaramillo Ana Belén, estudiante de la Escuela de Ingeniería de Sistemas y Telemática, previo a la obtención del título de Ingeniera de Sistemas y Telemática.

La Junta considera que el diseño de trabajo de titulación cumple con los requisitos normados en la "Guía de Elaboración y Presentación de la Denuncia/Protocolo de Trabajo de Titulación", razón por la cual solicita, por su digno intermedio, notificar al tribunal designado y determinar lugar, fecha y hora de sustentación.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomienda como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Diego Chacón (docente del curso de graduación), y como miembro del Tribunal al Ing. Juan Carlos Salgado.

Atentamente,

Ing. Marcos Orellana Cordero  
Director Escuela de Ingeniería de Sistemas y Telemática  
Universidad del Azuay

---

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

Sustentación del Diseño de Monografía (Doctor Romel Machado Clavijo)

Fecha: 07-04-2014

**ESCUELA DE INGENIERIA DE SISTEMAS**

*Diseños de Monografía  
Escuela de Sistemas Y Telemática*

**Estudiante:** Ana Belén Arízaga Jaramillo con código 46144

**Tema:** "DESARROLLO DE UNA APLICACIÓN EN ANDROID PARA EL MANEJO DE UNA ARQUITECTURA DOMOTICA ESPECIFICA EN EL CONTROL DE ACCESO, RIEGO ILUMINACION"

**Para:** La obtención del título de Ingeniera de Sistemas y Telemática.

**Director:** Ing. Diego Chacón.

**Tribunal:** Ing. Juan Carlos Salgado.

DIA: Jueves

FECHA: 10-04-2014

HORA: 17h00

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.



1.

**ACTA**  
**SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN**

1.1. **Nombre del estudiante:** ANA BELEN JARAMILLO ARIZAGA

1.1.1 Código: 46144

1.2 **Director sugerido:** Ing. Diego Chacón

1.3 **Codirector (opcional):** \_\_\_\_\_

1.4 **Tribunal:** Ings. Juan Carlos Salgado Arteaga

1.5 **Título propuesto:** DESARROLLO DE UNA APLICACIÓN EN ANDROID PARA EL MANEJO DE UNA ARQUITECTURA DOMOTICA ESPECIFICA EN EL CONTROL DE ACCESO, RIEGO, ILUMINACION, CLIMATIZACION CON ARDUINO Y RASPBERRY PI

1.6 **Resolución:**

1.6.1 Aceptado sin modificaciones

1.6.2 Aceptado con las siguientes modificaciones:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

1.7 Responsable de dar seguimiento a las modificaciones (designado por la Junta Académica de entre los Miembros del Tribunal): Ing. Diego Chacón

1.7.1 No aceptado

• Justificación:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Tribunal**

.....

Ing. Juan C. Salgado A.

.....

.....

Ing. Diego Chacón

Secretario de Facultad

Fecha de sustentación: 10 de Abril de 2014



### RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN

**1.1 Nombre del estudiante:** ANA BELEN JARAMILLO ARIZAGA (46144)

**1.2 Director sugerido:** Ing. Diego Chacon

**1.3 Codirector (opcional):**

**1.4. Título propuesto:** DESARROLLO DE UNA APLICACIÓN EN ANDROID PARA EL MANEJO DE UNA ARQUITECTURA DOMOTICA ESPECIFICA EN EL CONTROL DE ACCESO, RIEGO, ILUMINACION, CLIMATIZACION CON ARDUINO Y RASPBERRY PI

**1.5 Revisores (tribunal):** Ing. Juan Carlos Salgado Arteaga

**1.6 Recomendaciones generales de la revisión:**

	Cumple totalmente	Cumple parcialmente	No cumple	Observaciones (*)
<b>Línea de investigación</b>				
1. ¿El contenido se enmarca en la línea de investigación seleccionada?	/			
<b>Título Propuesto</b>				
2. ¿Es informativo?	/			
3. ¿Es conciso?	/			
<b>Estado del arte</b>				
4. ¿Identifica claramente el contexto histórico, científico, global y regional del tema del trabajo?	/			
5. ¿Describe la teoría en la que se enmarca el trabajo	/			
6. ¿Describe los trabajos relacionados más relevantes?	✓			
7. ¿Utiliza citas bibliográficas?	/			
<b>Problemática y/o pregunta de investigación</b>				
8. ¿Presenta una descripción precisa y clara?	/			
9. ¿Tiene relevancia profesional y social?	/			
<b>Hipótesis (opcional)</b>				
10. ¿Se expresa de forma clara?	✓			
11. ¿Es factible de verificación?	/			
<b>Objetivo general</b>				
12. ¿Concuerda con el problema formulado?	/			
13. ¿Se encuentra redactado en tiempo verbal infinitivo?	/			
<b>Objetivos específicos</b>				
14. ¿Concuerdan con el objetivo general?	/			
15. ¿Son comprobables cualitativa o cuantitativamente?	/			
<b>Metodología</b>				



16. ¿Se encuentran disponibles los datos y materiales mencionados?	/			
17. ¿Las actividades se presentan siguiendo una secuencia lógica?	/			
18. ¿Las actividades permitirán la consecución de los objetivos específicos planteados?	/			
19. ¿Los datos, materiales y actividades mencionadas son adecuados para resolver el problema formulado?	/			
<b>Resultados esperados</b>				
20. ¿Son relevantes para resolver o contribuir con el problema formulado?	/			
21. ¿Concuerdan con los objetivos específicos?	✓			
22. ¿Se detalla la forma de presentación de los resultados?	✓			
23. ¿Los resultados esperados son consecuencia, en todos los casos, de las actividades mencionadas?	/			
<b>Supuestos y riesgos</b>				
24. ¿Se mencionan los supuestos y riesgos más relevantes?	/			
25. ¿Es conveniente llevar a cabo el trabajo dado los supuestos y riesgos mencionados?				No hay respuesta para la pregunta
<b>Presupuesto</b>				
26. ¿El presupuesto es razonable?	/			
27. ¿Se consideran los rubros más relevantes?	/			
<b>Cronograma</b>				
28. ¿Los plazos para las actividades son realistas?	/			
<b>Referencias</b>				
29. ¿Se siguen las recomendaciones de normas internacionales para citar?	/			
<b>Expresión escrita</b>				
30. ¿La redacción es clara y fácilmente comprensible?	✓			
31. ¿El texto se encuentra libre de faltas ortográficas?		/		Raspberry no Raspberry.

(\*) Breve justificación, explicación o recomendación.

- Opcional cuando cumple totalmente,



- Obligatorio cuando cumple parcialmente y NO cumple.

.....

.....

.....

.....



Oficio Nro. 031-2014-DIST-UDA

Cuenca, 03 de Abril de 2014

Señor Ingeniero  
Xavier Ortega Vázquez  
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN  
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 3 de Abril del 2014, revisó el proyecto de monografía titulado "Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.", presentada por la estudiante Arízaga Jaramillo Ana Belén, estudiante de la Escuela de Ingeniería de Sistemas y Telemática, previo a la obtención del título de Ingeniera de Sistemas y Telemática.

La Junta considera que el diseño de trabajo de titulación cumple con los requisitos normados en la "Guía de Elaboración y Presentación de la Denuncia/Protocolo de Trabajo de Titulación", razón por la cual solicita, por su digno intermedio, notificar al tribunal designado y determinar lugar, fecha y hora de sustentación.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomienda como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Diego Chacón (docente del curso de graduación), y como miembro del Tribunal al Ing. Juan Carlos Salgado.

  
Atentamente,

Ing. Marcos Orellana Cordero  
Director Escuela de Ingeniería de Sistemas y Telemática  
Universidad del Azuay

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.



UNIVERSIDAD DEL AZUAY

DOCTORA JENNY RIOS COELLO SECRETARIA, DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN DE LA UNIVERSIDAD DEL AZUAY

CERTIFICA:

Que, la Señorita Ana Belén Arízaga Jaramillo registrada con el código 46144 perteneciente a la Escuela de Ingeniería de Sistemas, luego de cumplir con todas las asignaturas de su Pensum de estudios, egresó de la Facultad en el día 20 de Julio de 2013.

Cuenca, Abril 07 del 2014



Derecho 54508  
vcf.-

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.



## 1 DATOS GENERALES

### 1.1 Nombre del estudiante:

Arízaga Jaramillo Ana Belén

### 1.2 Código:

46144

### 1.3 Contacto:

072816970/0984412032 belenarizaga7@gmail.com

### 1.4 Director sugerido:

Ing. Chacón Diego, Ms.

### 1.5 Tribunal Designado

Ing. Juan Carlos Salgado.

### 1.6 Línea de Investigación de la carrera:

*1203 Informática de computadores.*

*1203.17 Informática*

### 1.7 Área de estudio:

Domótica y Hogar Digital para dispositivos móviles con Android.

### 1.8 Título propuesto:

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.

### 1.9 Subtítulo

Control móvil de una arquitectura domótica

### 1.10 Estado del proyecto

Se dispone de una aplicación prototipo que realiza una de las funcionalidades propuestas.

## 2 Contenido

### 2.1 Motivación de la investigación:

Las aplicaciones móviles son hoy en día primordiales para los usuarios de teléfonos inteligentes, sin embargo en el Ecuador no existen muchas dedicadas al control de infraestructuras domóticas, las cuales nos ofrecen confort y control sobre nuestro hogar. Si utilizamos las apps que vienen por defecto con los equipos domóticos el valor es muy elevado. Por esta razón, se pretende crear una aplicación que controle una arquitectura basada en

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.



UNIVERSIDAD DEL AZUAY

estándares de comunicación existentes y dispositivos de bajo costo, como Arduino y Raspberry Pi.

## 2.2 Problemática:

Las aplicaciones móviles están tomando gran parte en el ámbito del desarrollo, sin embargo se puede observar que en el Ecuador la creación de nuevas aplicaciones para la gestión y control de arquitecturas domóticas no es muy extensa. Por otro lado la implementación de equipos domóticos dentro de un hogar tiene costos muy elevados, aún más si solicitamos una pantalla de control que viene con su propia aplicación para el manejo, los costos suben. Una de las soluciones sería la fabricación y desarrollo de propios equipos domóticos para el Ecuador, sin embargo para que esto suceda es necesario esperar que el conocimiento y la necesidad de la domótica crezcan.

Considerando además que en nuestro país el incentivo al desarrollo no es tan elevado, creemos muchas veces que es mejor comprar o utilizar productos y software ya desarrollado. Es por esto que se pretende desarrollar una aplicación con una interfaz sencilla para el usuario, que le permita experimentar todo el verdadero confort que conlleva una arquitectura domótica. Es importante considerar que el diseño de dicha aplicación es muy importante para que tenga el enganche necesario con los usuarios.

## 2.3 Pregunta de investigación

¿Se puede realizar una implementación entre una arquitectura domótica específica en el control de acceso, riego, iluminación y temperatura con una aplicación móvil?

## 2.4 Resumen

Con la realización de esta monografía se pretende crear una aplicación móvil en Android capaz de servir como intermediaria para el control de un sistema domótico específico en el control de acceso, de riego, de luces, y de climatización representado en una maqueta, que va a servir para cumplir la función principal de la domótica que es proporcionar la comodidad del usuario. La aplicación consta de un desarrollo amigable y fácil de entender para el usuario, utilizando también imágenes y estados actuales de los dispositivos para tener una interacción más dinámica y real.

## 2.5 Estado del Arte

### 2.5.1 Android

Android es un sistema operativo orientado a dispositivos portátiles como teléfonos inteligentes, netbooks, tabletas, entre otros. Fue desarrollado inicialmente por la compañía de software Android Inc., absorbida por Google en el 2005. Con una plataforma basada en el kernel de Linux, Google promocionó un nuevo sistema flexible y actualizable, llamando la atención de operadores y fabricantes de hardware y software.

A diferencia de sistemas operativos para dispositivos móviles como iOS o Windows Phone, el código fuente de Android se mantiene abierto. Soporta multimedia, interfaz táctil, características basadas en voz y una amplia variedad de componentes de hardware. Si se

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.



UNIVERSIDAD DEL  
AZUAY

añade a esto un entorno de desarrollo gratuito que incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software, Android constituye una opción interesante para nuevos desarrolladores. (Purdy, 2009)

### 2.5.2 Aplicaciones móviles en Android

Los sistemas móviles con sistema operativo Android tienen ciertas características que pueden dar algunas complicaciones para sus desarrolladores y que por lo tanto resultan retos para el desarrollo. Algunas debidas al tipo de dispositivos donde se ejecutan y otras dependen del tipo de trabajo desarrollado. (Arturo Baz Alonso, Irene Ferreira Artime, María Álvarez Rodríguez, Rosana García Baniello, 2008)

Una aplicación móvil es diseñada para tener una personalización específica de un negocio, juego, idea o aplicación propia. Su objetivo principal es proporcionarnos un servicio específico para el uso directo e intuitivo de la misma. Tal ha sido su crecimiento, que las aplicaciones hoy en día forman parte de todo tipo de aspecto y ámbito laboral. Hablaremos estrechamente de la relación entre las aplicaciones móviles y la domótica.

### 2.5.3 Android y Domótica

Puesto que el principal objetivo de la domótica es automatizar, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas. Proporcionando de esta manera al usuario el confort que requiere y dejando la posibilidad de controlarlo manualmente o a través de un dispositivo a distancia. Es por esta misma razón que la importancia de las aplicaciones móviles juega un papel sumamente importante en la optimización y control de nuestra casa; por ambientes; o por dispositivos específicos mediante el internet. (Harke, 2010)

En esta monografía se pretende realizar la aplicación móvil en Android para conjuntamente con el trabajo de un compañero demostrar el funcionamiento de un sistema domótico.

## 2.6 Hipótesis:

Se puede crear una aplicación en Android para controlar una arquitectura domótica (Arduino y Raspberry Pi).

## 2.7 Objetivo general:

Desarrollar una aplicación en Android para móviles capaz de controlar una arquitectura domótica en los ámbitos de control de acceso, luces, riego y temperatura usando (Arduino y Raspberry Pi).

## 2.8 Objetivos Específicos:

- Establecer el diseño para la aplicación considerando que tiene que ser de fácil manejo para los usuarios.
- Crear los servicios web para la comunicación entre la aplicación y el servidor de los dispositivos domóticos.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.



UNIVERSIDAD DEL AZUAY

- Implementar una demostración de la aplicación en conjunto con la arquitectura desarrollada.

### 2.9 Metodología:

Esta monografía es un proyecto de desarrollo tecnológico en el que se evaluará el cumplimiento de los objetivos planteados y la confiabilidad del producto. Para su desarrollo, y por la naturaleza del mismo, se utilizará un método experimental basado en pruebas exhaustivas de rendimiento y confiabilidad. Se pretende alcanzar los resultados especificados siendo constante con el cronograma y siguiendo las actividades de planificación, diseño, desarrollo, pruebas y corrección de errores. Al finalizar el desarrollo de la aplicación será necesario para su total funcionamiento, planificar reuniones con el encargado del desarrollo del sistema domótico, obteniendo así el resultado esperado que es el funcionamiento en conjunto del sistema.

### 2.10 Alcances y resultados esperados

Crear una aplicación operativa en Android para dispositivos móviles capaz de controlar un sistema domótico usando herramientas de software y hardware abiertos (Arduino y Raspberry Pi) y Eclipse para Android. La comunicación entre la aplicación y el sistema se realizará mediante servicios web sobre el protocolo RS232. La aplicación debe ser dinámica y mostrar los estados actuales de los dispositivos que se encuentran conectados y cambian de estado, además mostrará el número de dispositivos conectados únicamente. Como resultado se tendrá un prototipo de una aplicación para gestionar el control de un hogar domótico.

### 2.11 Supuestos y riesgos

Los riesgos identificados que podrían llevar al retraso o fracaso del proyecto son:

Fallas en el hardware debido a errores de fábrica o conexiones indebidas, lo cual retrasaría la prueba de compatibilidad entre la aplicación y el sistema.

Mal manejo de los servicios web para consumir o enviar información y crítica que definen la funcionalidad de la aplicación.

### 2.12 Presupuesto

Rubro-Denominación	Costo USD (Detalle)	Justificación ¿para qué?
	C. Total: \$40.00	Para tener un respaldo y ayuda sobre temas específicos que requieren más de una investigación normal, y que sirven también para el crecimiento en el desarrollo de aplicaciones móviles.
Suscripción a foros de discusión y ayuda específica en Android.		
	C. Unidad: \$35.00	Para realizar las pruebas de conexión entre la aplicación y el servicio web, para asegurar que la comunicación se
1 Tarjeta Arduino UNO		

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.



UNIVERSIDAD DEL AZUAY

- Implementar una demostración de la aplicación en conjunto con la arquitectura desarrollada.

## 2.9 Metodología:

Esta monografía es un proyecto de desarrollo tecnológico en el que se evaluará el cumplimiento de los objetivos planteados y la confiabilidad del producto. Para su desarrollo, y por la naturaleza del mismo, se utilizará un método experimental basado en pruebas exhaustivas de rendimiento y confiabilidad. Se pretende alcanzar los resultados especificados siendo constante con el cronograma y siguiendo las actividades de planificación, diseño, desarrollo, pruebas y corrección de errores. Al finalizar el desarrollo de la aplicación será necesario para su total funcionamiento, planificar reuniones con el encargado del desarrollo del sistema domótico, obteniendo así el resultado esperado que es el funcionamiento en conjunto del sistema.

## 2.10 Alcances y resultados esperados

Crear una aplicación operativa en Android para dispositivos móviles capaz de controlar un sistema domótico usando herramientas de software y hardware abiertos (Arduino y Raspberry Pi) y Eclipse para Android. La comunicación entre la aplicación y el sistema se realizará mediante servicios web sobre el protocolo RS232. La aplicación debe ser dinámica y mostrar los estados actuales de los dispositivos que se encuentran conectados y cambian de estado, además mostrará el número de dispositivos conectados únicamente. Como resultado se tendrá un prototipo de una aplicación para gestionar el control de un hogar domótico.

## 2.11 Supuestos y riesgos

Los riesgos identificados que podrían llevar al retraso o fracaso del proyecto son:

Fallas en el hardware debido a errores de fábrica o conexiones indebidas, lo cual retrasaría la prueba de compatibilidad entre la aplicación y el sistema.

Mal manejo de los servicios web para consumir o enviar información y crítica que definen la funcionalidad de la aplicación.

## 2.12 Presupuesto

Rubro-Denominación	Costo USD (Detalle)	Justificación ¿para qué?
Suscripción a foros de discusión y ayuda específica en Android.	C. Total: \$40.00	Para tener un respaldo y ayuda sobre temas específicos que requieren más de una investigación normal, y que sirven también para el crecimiento en el desarrollo de aplicaciones móviles.
1 Tarjeta Arduino UNO	C. Unidad: \$35.00	Para realizar las pruebas de conexión entre la aplicación y el servicio web, para asegurar que la comunicación se

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.



UNIVERSIDAD DEL AZUAY

está realizando correctamente.

*Equipos para el desarrollo de la Aplicación*

C. Celular de Pruebas:  
\$600.00

Para realizar las pruebas reales del funcionamiento de la aplicación, así como para realizar un desarrollo más rápido y ágil.

### 2.13 Financiamiento

Se espera que la Universidad pueda facilitar algunas placas Arduino con los Shields Ethernet, pero de no ser posible se adquirirán por cuenta del estudiante, esta parte es primordial para la realización de pruebas de conexión, por el desarrollo, pago de suscripción, el equipo móvil para las pruebas y el conocimiento queda por cuenta del estudiante que es el interesado en adquirir dichos conocimientos de refuerzo.

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.



## 2.14 Esquema tentativo

2.14.1 Abstract

2.14.2 Introducción

2.14.3 Objetivos

2.14.4 Capítulo 1. Fundamentos teóricos

2.14.4.1 *Introducción a las aplicaciones móviles en Android*

2.14.4.1.1 Conexión y comunicación con HttpClient, HttpPost, HttpGet

2.14.4.2 *Introducción a la domótica*

2.14.4.2.1 Componentes principales

2.14.5 Capítulo 2 Análisis y diseño de la Aplicación

2.14.5.1 *Control de Acceso*

2.14.5.2 *Control de Iluminación*

2.14.5.3 *Control de Riego*

2.14.5.4 *Climatización*

2.14.5.5 *Pantallas Principales*

2.14.5.6 *Definición de Funcionalidades*

2.14.6 Capítulo 3. Comunicación entre la Aplicación y el sistema

2.14.6.1 *Web Services*

2.14.6.2 *Desarrollo de la Aplicación*

2.14.7 Capítulo 4. Implementación y pruebas

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.



UNIVERSIDAD DEL AZUAY

## 2.15 Cronograma

Objetivo Específico	Actividad	Resultado esperado	Tiempo (semanas)
	-Realizar un esquema tentativo de las posiciones de los elementos (modboard).	-Definir los colores, los iconos y la mejor posición para lograr que sea de fácil uso para los usuarios.	- Media Semana (4 días)
<i>Establecer el diseño para la aplicación considerando que tiene que ser de fácil manejo para los usuarios.</i>	-Realizar las pantallas en la aplicación.	- Un diseño correcto con las funcionalidades necesarias.	- Media Semana (4 días)
	-Realizar el desarrollo de la aplicación.	-Realizar el funcionamiento de todas las funciones elegidas.	-Una semana y media (10 días)
<i>Crear los servicios web para la comunicación entre la aplicación y el servidor de los dispositivos domóticos.</i>	-Replicar la base de datos definida en mysql dentro de wamp server para pruebas.	-Simular el sistema domótico que va a estar implementado para pruebas.	-Media semana(4 días)
	-Realizar un servicio web en el servidor wamp server para pruebas ya que en servidor final se subirá el archivo en php.	-Realizar la conexión conjuntamente con la base y el equipo arduino para asegurar una conexión correcta.	-Media semana(4 días)
<i>Implementar una demostración de la aplicación en conjunto con la arquitectura desarrollada.</i>	-Reuniones con el proyecto que tiene la arquitectura domótica para subir los web services al servidor.	-Los web services deben funcionar correctamente cambiando solo la dirección ya que la base de datos es replicada.	-Una semana (7 días).
	-Reuniones con el proyecto que tiene la arquitectura domótica para realizar pruebas de conexión y de respuesta correcta.	-La aplicación debe recibir notificaciones de la arquitectura y viceversa para lograr una interacción real y dinámica.	-Una semana (7 días)

Desarrollo de una aplicación en Android para el manejo de una arquitectura domótica específica en el control de acceso, riego, iluminación, climatización con Arduino y Raspberry Pi.



## 2.16 Referencias

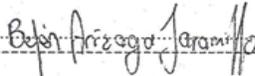
Arturo Baz Alonso, Irene Ferreira Artime, María Álvarez Rodríguez, Rosana García Baniello. (01 de 06 de 2008). Dispositivos móviles. Obtenido de <http://castor.edv.uniovi.es/~smi/5tm/09trabajos-sistemas/1/Memoria.pdf>

Harke, W. (2010). *Domótica para viviendas y edificios*. Barcelona, España: Marcombo.

Purdy., K. (2009). "The complete Android Guide". USA. Recuperado el 07 de 03 de 2014, de [http://www.completeguides.net/01\\_The\\_Complete\\_Android\\_Guide](http://www.completeguides.net/01_The_Complete_Android_Guide)

## 2.17 Anexos

### 2.18 Firma de responsabilidad (estudiante)



Belén Arízaga Jaramillo

### 2.19 Firma de responsabilidad (director sugerido)



Ing. Diego Chacón Mdhd

### 2.20 Fecha de Entrega