

DISEÑO DE UN APP DE ANALITICA DE USO DE REDES WIFI

LABRADOR GONZALEZ CHRISTIAN LEONARDO

UNIVERSITARIA AGUSTINIANA
FACULTAD DE INGENIERIA
INGENIERIA DE TELECOMUNICACIONES
BOGOTA D.C

2018

DISEÑO DE UN APP DE ANALITICA DE USO DE REDES WIFI

LABRADOR GONZALEZ CHRISTIAN LEONARDO

Asesor de Trabajo

GOMEZ RUIZ CARLOS ANDRES

Trabajo de grado para optar al título como
Profesional de Ingeniería en Telecomunicaciones

UNIVERSITARIA AGUSTINIANA
FACULTAD DE INGENIERIA
INGENIERIA EN TELECOMUNICACIONES
BOGOTA D.C

2018

Nota de aceptación

Presidente del Jurado

Firma del Jurado

Firma del Jurado

Bogotá 10/07/2018

Agradecimientos

Quiero agradecer al docente Andrés Gómez Ruiz por su valiosa colaboración y asesoría, por su tolerancia, comprensión y apoyo incondicional.

A la empresa SDT Ingeniería, por las herramientas y asesorías prestadas en el desarrollo del proyecto.

A los profesores de la Universidad, que contribuyen en sus clases con su orientación y estímulo a culminar esta meta.

A la Universitaria Agustiniana por brindarme la oportunidad de superación profesional en beneficio de esta y futuras generaciones.

Al Señor Jesús Todo poderoso por darme la perseverancia, la fortaleza y sabiduría para realizar y culminar este proyecto.

A la mujer que me incentivo para iniciar esta carrera educativa,

Por sus consejos, su apoyo, su sacrificio, su voz aliciente en todas las dificultades de este proyecto y sobretodo su amor. Mi madre.

A la empresa SDT Ingeniería por brindarme su apoyo y

Confianza en mí para el desarrollo de este proyecto.

Glosario

A

ADT: Android Development Tools plugin.

APK: Android Application Package.

I

IRC: Internet Relay Chat.

J

JSON: JavaScript Object Notation.

P

PAYLOAD: Carga útil.

PHP: Hypertext Pre-processor.

S

SDK: Software Development Kit ó Kit de Desarrollo de Software.

SIP: Session Description Protocol.

SSL: Secure Sockets Layer.

X

XML: Protocolo de mensajería y comunicación de presencia.

V

VoIP: Voice over Internet Protocol / Voz sobre protocol de Internet.

Resumen

El presente documento contiene la documentación del proyecto de desarrollo un App que logre mostrar gráficas dinámicas del uso de las redes inalámbricas implementadas por la empresa SDT Ingeniería. Se ha conseguido realizar las gráficas analíticas en función del tiempo realizando consultas a las bases de datos que tienen construidas; el app deberá contar con cuentas para funcionarios o agentes con un sistema de autenticación, y donde por medio de su Smartphone se podrá acceder a consultas de las bases de datos en MySQL, cuyo resultado produce gráficas estadísticas de interés para la operación de la empresa SDT.

En este proyecto de investigación, en particular, se desarrolla un app para sistemas operativos Android, en lenguaje de programación para desarrollar dichas aplicaciones es JavaScript y herramientas de desarrollo SDK (Software Development Kit o Kit de desarrollo de Software) entre otras.

CONTENIDO

Introducción	1
Planteamiento problema	2
Descripción del problema	2
Pregunta problema	2
Justificación	3
Objetivos	4
Objetivos generales	4
Objetivos específicos	4
Marco teórico	5
Marco conceptual	10
Protocolos de consulta de clientes a base de datos	10
Xml	10
Estructura cliente-servidor	11
Sistemas operativos	13
Servidores	14
Marco legal	15
Metodología	17
Administración proyecto	18
Cronograma de actividades	18
Presupuesto	18
Desarrollo del problema de investigación	18
Infraestructura de red	19
Implementación de servicio web json	22
MySQL en Ubuntu	23
Código del json	26
Coke php	28
Estructura del desarrollo (App)	32
Frameworks	32
Modelo vista controlador	35

Framework ionic	36
Diseño de red móvil	39
Diseño de software funcional	40
Desarrollo del App	42
Desarrollo front App	46
Plan de pruebas	51
Ejecución de pruebas	55
Conclusiones	57
Recomendaciones	58
Referencia	59

Lista de tablas

Tabla 1. Cronograma de actividades	18
Tabla 2. Investigación de mercado de dispositivos móviles.	55

Lista de Figuras

Figura 1. Estructura de base de datos	10
Figura 2. Subconjunto de SMGL	11
Figura 3. Ejemplo de una estructura de un documento xml	11
Figura 4. Diagrama cliente-servidor via internet	12
Figura 5. Arquitectura general de un servicio de mensajería instantánea	12
Figura 6. Arquitectura general del núcleo un sistema operativo	13
Figura 7. Diagrama estadísticas de sistemas operativos móviles	14
Figura 8. Esquema general cliente-servidor	14
Figura 9. Proceso de la metodología en el desarrollo de app	19
Figura 10. Ciclo de solicitud CokePHP	31
Figura 11. Arquitectura MCC en Dispositivos Móviles	40

Figura 12. Infraestructura de diseños de software	41
Figura 13. Front del menú del dispositivo móvil	42
Figura 14. Desarrollo front del aplicativo	43
Figura 15. Código manifest.xml	43
Figura 16. Configuración de carpetas de desarrollo	44
Figura 17. Configuración de librerías independientes	44
Figura 18. Configuración carpeta MainActivity	45
Figura 19. Configuración de clase Connection	45
Figura 20. Configuración de clase Connection	46
Figura 21. Autenticación a la app	47
Figura 22. Actualización y cargue de información a las bases de datos SDT	47
Figura 23. Listado consulta de los clientes SDT	48
Figura 24. Graficas analíticas de la lectura de información wifi	49
Figura 25. Graficas analíticas de la lectura de información wifi	50
Figura 26. Configuración de conexión de servicios al as bases solicitadas	51
Figura 27. Enfoque de pruebas Móviles	51
Figura 28. Pantallas de pruebas de batería, cpu, red	55
Figura 29. Pantallas de pruebas cpu, red	56
Figura 30. Pantalla de pruebas en memoria y almacenamiento	56

Introducción

El presente trabajo representa la investigación y desarrollo frente a una problemática presentada por la empresa SDT de ingeniería en cuanto a la atención de requerimientos en las consultas de información sobre el uso de las redes inalámbricas implementadas por dicha empresa, y que reposa en bases de datos centralizadas; lo anterior desde una interfaz amigable y portable basada en Smartphones y lectura de informaciones analíticas gráficas.

Para la atención de este problema se va hacer un abordaje de una solución de software basado en una estructura de telecomunicaciones en una arquitectura cliente – servidor con creación de un servicio web por demanda bajo tecnología de Json de servicio, y conexión a bases de datos por medio del lenguaje de programación PHP. Dando así una solución con un servidor virtual en la nube y una serie de clientes de software de apps en sus smartphones que logran realizar la interacción con clientes finales.

Planteamiento del Problema

Descripción del problema

Actualmente la empresa SDT de Ingeniería cuenta con unas plataformas de Bases de Datos de MySQL y solo se pueden realizar las consultas por medios electrónicos web, esta labor requiere la dedicación de personal exclusivo en unas interfaces basadas en estaciones de trabajo fijas y horarios de atención fijos, sin embargo existe la necesidad de flexibilizar estas labores a que puedan ser atendidas desde dispositivos móviles y en cualquier horario y lugar donde se encuentre el personal; dado esto es necesario desarrollar una infraestructura que permita autenticar al personal de forma remota y darles acceso al control de las bases de datos y puedan realizar las consultas mostrando graficas dinámicas sobre el uso de las redes inalámbricas, en función del tiempo .

Pregunta Problema

¿Qué condiciones debe tener una herramienta móvil de análisis de datos sobre uso de redes inalámbricas de la empresa SDT Ingeniería?

Justificación

El desarrollo de este proyecto cobra gran importancia para la dinamización de las labores técnicas de seguimiento al uso y desempeño de las redes inalámbricas implementadas por la empresa SDT Ingeniería, ya que provee a las diversas áreas de la compañía de interfaces móviles para realizar consultas en tiempo real, y que permitan identificar patrones de comportamientos típicos para labores como diagnósticos de la operación de cada red.

Técnicamente El desarrollo de este proyecto generará una solución basada en una infraestructura cliente-servidor con un JSON+PHP + MySQL con que centralice todas consultas y descentralice los usuarios de la empresa, logrando así crear una alternativa tecnológica flexible a horarios y locaciones, urgencias laborales.

Personalmente es interesante el desarrollo de este proyecto en la creación de un App que sirva para facilitar las experiencias que tenga un usuario con sus funciones laborales, ya que generaría un gran desempeño y satisfacción para el área en el que se labora y así lograr tener más desarrollo y experiencia laboral en el área de las telecomunicaciones.

Así con el desarrollo y elaboración de este proyecto se podrían incluir nuevos aprendizajes e ideas para las generaciones que crea la Universitaria Agustiniiana de ingenieros de telecomunicaciones.

Objetivos

Objetivo General.

Diseño de un App de analítica de uso de redes WiFi de la empresa SDT Ingeniera, por medio de tecnologías de sistemas operativos Móviles.

Objetivos Específicos.

Implementar un servicio web (web Service) centralizado y en la nube que permita la consulta dinámica de datos desde aplicaciones APP hasta las bases de datos MySQL.

Diseñar un App que tenga funciones de autenticación centralizada contra base de datos del servidor centralizado y pueda acceder a consultas remotas de datos.

Dotar el APP de funciones de gráfica analítica de datos estadísticos y funcionalidades asíncronas.

Marco Teórico

Las Apps en relaciones a las bases de datos se llevan usando desde hace mucho tiempo pero en sistemas de datos realmente grandes, cuando se ejecutan millones de transacciones en periodos de tiempo suficientemente cortos, es muy fácil encontrar consultas SQL que saturan por completo el servidor durante segundos, minutos o incluso horas. (García R, 2016)

A lo largo de la historia se ha visto varios avances desde el inicio de la creación en las base de datos, a mediados de los años 60. En una primera aproximación, se puede decir que una base de datos es un conjunto de información relacionada que se encuentra agrupada o estructurada y desde el punto de vista informático, una base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos. (Contreras L, 2017).

En 1960 se crea y se populariza, el uso de los discos, cosa que fue un adelanto muy efectivo en la época, debido a que a partir de este soporte se podía consultar la información directamente, sin tener que saber la ubicación exacta de los datos. En esta misma época se dio inicio a las primeras generaciones de bases de datos de red y las bases de datos jerárquicas, ya que era posible guardar estructuras de datos en listas y árboles. (Museo Informático, 2011)

Otro de los principales logros de los años sesenta fue la alianza de IBM y American Airlines para desarrollar SABRE, un sistema operativo que manejaba las reservas de vuelos, transacciones e informaciones sobre los pasajeros de la compañía American Airlines. Y, posteriormente, en esta misma década, se llevó a cabo el desarrollo del IDS desarrollado por Charles Bachman (que formaba parte de la CODASYL) supuso la creación de un nuevo tipo de sistema de bases de datos conocido como modelo en red que permitió la creación de un standard en los sistemas de bases de datos gracias a la creación de nuevos lenguajes de sistemas de información.

CODASYL (Conference on Data Systems Languages) era un consorcio de industrias informáticas que tenían como objetivo la regularización de un lenguaje de programación estándar que pudiera ser utilizado en multitud de ordenadores. Los miembros de este consorcio

pertenecían a industrias e instituciones gubernamentales relacionadas con el proceso de datos, cuya principal meta era promover un análisis, diseño e implementación de los sistemas de datos más efectivos; y aunque trabajaron en varios lenguajes de programación como COBOL, nunca llegaron a establecer un estándar fijo, proceso que se llevó a cabo por ANSI. (Chaparro S, 2014)

Como consecuencia de esto, durante la década de 1970, Lawrence J. Ellison, más conocido como Larry Ellison, a partir del trabajo de Edgar F. Codd sobre los sistemas de bases de datos relacionales, desarrolló el Relational Software System, o lo que es lo mismo, lo que actualmente se conoce como Oracle Corporation, desarrollando así un sistema de gestión de bases de datos relacional con el mismo nombre que dicha compañía llamada “Oracle” (M Informatico, 2017)

Posteriormente en la época de los ochenta también se desarrollará el SQL (Structured Query Language) o lo que es lo mismo un lenguaje de consultas o lenguaje declarativo de acceso a bases de datos relacionales que permite efectuar consultas con el fin de recuperar información de interés de una base de datos y hacer cambios sobre la base de datos de forma sencilla; además de analiza grandes cantidades de información y permitir especificar diversos tipos de operaciones frente a la misma información, a diferencia de las bases de datos de los años ochenta que se diseñaron para aplicaciones de procesamiento de transacciones.

Pero cabe destacar que ORACLE es considerado como uno de los sistemas de bases de datos más completos que existen en el mundo, y aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace relativamente poco, actualmente sufre la competencia del SQL Server de la compañía Microsoft y de la oferta de otros Sistemas Administradores de Bases de Datos Relacionales con licencia libre como es el caso de PostgreSQL, MySQL o Firebird que aparecerían posteriormente en la década de 1990. (Informático M, 2018).

Seguidamente en 1983 comenzó el auge de la comercialización de los sistemas relacionales, y SQL comenzó a ser el estándar de la industria, ya que las bases de datos relacionales con su sistema de tablas (compuesta por filas y columnas) pudieron competir con las bases jerárquicas y de red, como consecuencia de que su nivel de programación era sencillo y su nivel de programación era relativamente bajo. (Cárdenas S y McConnell J, 2017)

Esto tenía un gran avance y por eso en 1989 Jarkko Oiharinen crea Internet Relay Chat (IRC),

un programa para la comunicación instantánea a servidores a través de canales multiusuario conocidos como “grupos”. Esta aplicación posibilita las charlas grupales, los mensajes privados y hasta el intercambio de ficheros. Así en 1990 ayuda a superar un apagón mediático en la Unión Soviética, y para esta altura, internet se vuelve más popular, por lo que muchos desarrolladores comienzan a imitar a esta poderosa herramienta. IRC es un protocolo de comunicación en tiempo real basado en texto, que permite la conferencia entre 2 o más personas y que está clasificado dentro de la mensajería instantánea (Abdreu T, Regalado E. 2008).

En la década de 1990 la investigación en bases de datos giró en torno a las bases de datos orientadas a objetos. Las cuales han tenido bastante éxito a la hora de gestionar datos complejos en los campos donde las bases de datos relacionales no han podido desarrollarse de forma eficiente. Así se desarrollaron herramientas como Excel y Access del paquete de Microsoft Office que marcan el inicio de las bases de datos orientadas a objetos. Fue también en esta época cuando se empezó a modificar la primera publicación hecha por ANSI del lenguaje SQL y se empezó a agregar nuevas expresiones regulares, consultas recursivas, triggers y algunas características orientadas a objetos, que posteriormente en el siglo XXI volverá a sufrir modificaciones introduciendo características de XML, cambios en sus funciones, estandarización del objeto sequence y de las columnas auto numéricas. Y además, se creará la posibilidad de que SQL se pueda utilizar conjuntamente con XML, y se definirá las maneras de cómo importar y guardar datos XML en una base de datos SQL. Dando así, la posibilidad de proporcionar facilidades que permiten a las aplicaciones integrar el uso de XQuery (lenguaje de consulta XML) para acceso concurrente a datos ordinarios SQL y documentos XML. Y posteriormente, se dará la posibilidad de usar la cláusula order by. Aunque el boom de la década de los noventa será es el nacimiento del World Wide Web a finales de la década, ya que a través de este se facilitará la consulta a bases de datos. (Informática M, 2012)

Las conversaciones se desarrollan en los llamados canales de IRC, que pueden ser locales al servidor al que se conectan los clientes o no. Los usuarios del IRC utilizan una aplicación cliente para conectarse con un servidor en el que funciona una aplicación INCD (IRC Daemon o servidor IRC), que gestiona los canales y las conversaciones. Para adicionarle seguridad al protocolo se puede utilizar SSL opcionalmente. También se puede decir que un servidor se conecta con otros servidores para expandir la red IRC e intercambian todo su tráfico, de forma

tal que todos los servidores de una red tienen copia de todos los mensajes de las salas de chat y así se crea la ilusión del lado de los clientes de que están conectados a un gran servidor. La manera en que los usuarios acceden a las redes IRC es conectando un cliente al servidor. (Abdreu T, Regalado E. 2008).

En la actualidad, las tres grandes compañías que dominan el mercado de las bases de datos son IBM, Microsoft y Oracle. Por su parte, en el campo de internet, la compañía que genera gran cantidad de información es Google. Aunque existe una gran variedad de software que permiten crear y manejar bases de datos con gran facilidad, como por ejemplo LINQ, que es un proyecto de Microsoft que agrega consultas nativas semejantes a las de SQL a los lenguajes de la plataforma .NET. El objetivo de este proyecto es permitir que todo el código hecho en Visual Studio sean también orientados a objetos; ya que antes de LINQ la manipulación de datos externos tenía un concepto más estructurado que orientado a objetos; y es por eso que trata de facilitar y estandarizar el acceso a dichos objetos. (Informático M, 2012)

Cabe destacar que Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows que soporta varios lenguajes de programación tales como Visual C++, Visual#, Visual J#, ASP.NET y Visual Basic.NET, aunque se están desarrollando las extensiones necesarias para otros, cuyo objetivo es permitir crear aplicaciones, sitios y aplicaciones web, así como servicios web a cualquier entorno que soporte la plataforma .Net, creando así aplicaciones que intercomunicuen entre estaciones de trabajo, páginas web y dispositivos móviles. (Informático, 2012)

Por otra parte se veían los grandes avances tecnológicos y de desarrollo en los teléfonos móviles por compañías como Huawei, Nokia, Motorola, Apple, Samsung entre otras que formaron la iniciativa Wireless Village en el 2001, cuya finalidad es la de crear especificaciones universales para las conexiones a bases de datos móvil. Wireless Village es un conjunto de especificaciones para la mensajería instantánea y los servicios de presencia. Se pretende que sea un estándar para teléfonos móviles y dispositivos móviles para utilizar estos servicios a través de plataformas. Muchos teléfonos inalámbricos incluyen capacidades de conexiones a bases de datos de servidores diseñadas para conectarse a servicios de transferencias de archivos en IMPS en la red de una compañía, anteriormente conocida como el protocolo Wireless Village. Que

suele ser renombrado y personalizado por el operador de telefonía móvil o eliminado completamente si no se admite a nivel de red. La mayoría de los teléfonos Nokia y Motorola tienen esa característica, al igual que muchos teléfonos Sony Ericsson. (González C, 2017).

Las redes sociales también avanzaron en las transferencias de conexiones a bases de datos creadas por las plataformas, En los avances que se veían y la gran importancia de las conexiones a las distintas bases de datos creadas en el mundo en los Smartphone de las compañías de telefonía móvil se empezaron a desarrollar App con todos los protocolos de conexiones a BD, así en el 2009 aparece Whatsapp creadas por el ucraniano Jan Kuom y Brian Acton, un norteamericano trabajó para Adobe, Apple y Yahoo! antes de incursionar como cofundador de la plataforma de mensajería móvil que fue comprada el 19 de febrero de 2014 por la extraordinaria suma de 19.000 millones de dólares, una de las transacciones más abultadas en la historia de la tecnología. Whatsapp es una aplicación de mensajería instantánea, actualmente gratuita, para teléfonos inteligentes, que envía y recibe mensajes mediante Internet. Además de utilizar la mensajería en modo texto, los usuarios pueden crear grupos y enviarse mutuamente, imágenes, vídeos, documentos, grabaciones de audio y llamadas de Vip. (Burle S. 2014).

En la actualidad 5.3 billones de personas utilizan aplicaciones con conexiones a bases de datos a servidores como Windows live, Whatsapp, Facebook, twitter Messenger entre otras donde transfieren más de 20mil millones de paquetes de transferencias al día. (Aparicio. 2016)

Marco Conceptual

Protocolos de consultas a bases de datos.

Los protocolos de consulta son los que nos permiten realizar las comunicaciones a bases de datos, se encuentran gran variedad de estos protocolos para su desarrollo, estos protocolos pueden ser abiertos o privados dependiendo en el ámbito de desarrollo que se haya utilizado por ejemplo la mensajería instantánea para una empresa o en común para el público. Los protocolos se basan en una plataforma donde se realizan el intercambio de datos como XML y estos pueden ser utilizados en aplicaciones de mensajería instantánea. (López de Aberasturi, 2014). Así como lo visualizamos en la figura 1 donde un cliente se comunican dos personas por medio de los protocolos IM.

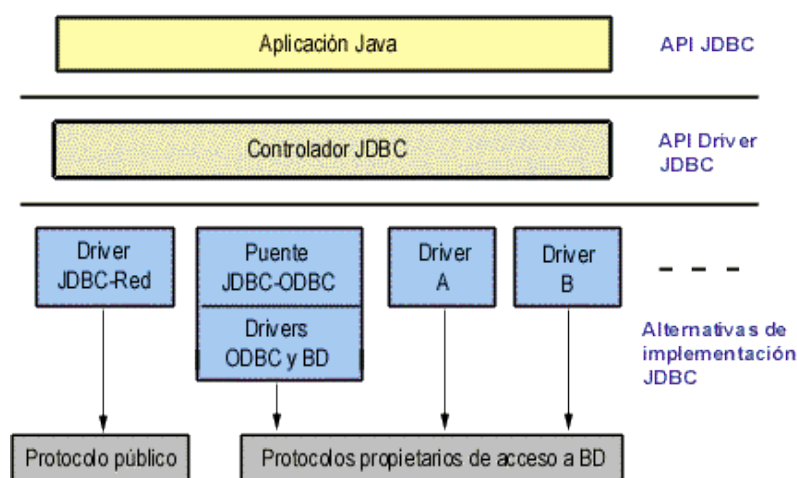


Figura 1. Estructura de bases de Datos. (Lamarca M. 2009)

Xml

Lenguaje de Marcado Extensible o Lenguaje de Marcas Extensible, es una meta-lenguaje que permite definir lenguajes de marcas desarrollado, utilizado para almacenar datos en forma legible. Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información



Fig. 2. Subconjunto de SGML(Estándar Generalised Mark-up Language). (U Madrid. 2013)

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">

<Edit_Mensaje>
  <Mensaje>
    <Remitente>
      <Nombre>Nombre del remitente</Nombre>
      <Mail>Correo del remitente </Mail>
    </Remitente>
    <Destinatario>
      <Nombre>Nombre del destinatario</Nombre>
      <Mail>Correo del destinatario</Mail>
    </Destinatario>
    <Texto>
      <Asunto>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Asunto>
      <Parrafo>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Parrafo>
    </Texto>
  </Mensaje>
</Edit_Mensaje>

```

Fig. 3. Ej. De una estructura de un documento XML (Gonzalo H 2016)

Estructura cliente-servidor

Esta estructura consiste en un sistema donde un cliente realiza peticiones de datos o información a un programa, ósea el servidor y este es el encargado de darle respuesta. Esta se aplica en los sistemas operativos ya que hay se encuentran multiusuario distribuido en una red de computadoras, la interacción que se tiene entre un cliente y un servidor es el soporte que se

brinda en la mayor parte de las comunicaciones por redes, y ya esta ayuda a comprender las bases sobre las que están contruidos los algoritmos distribuidos. Como se observa en la figura 4 la arquitectura cliente-servidor es un modelo de diseño de software en el que tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los que lo solicitan los clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. (E.V.A UCI. 2011)

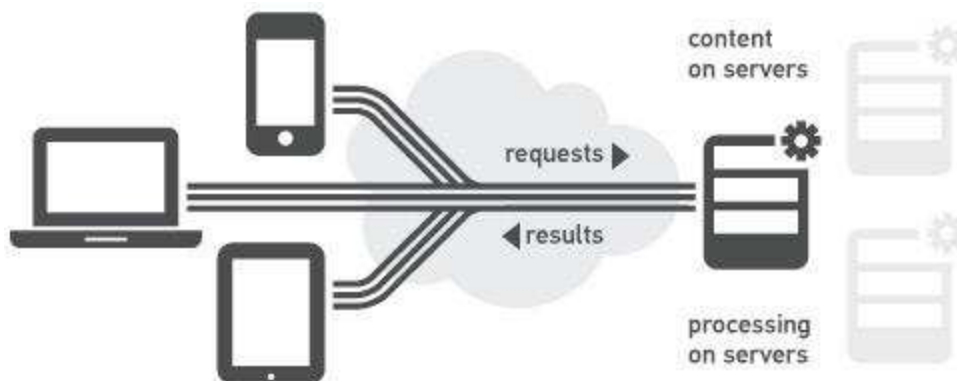


Fig. 4, Un diagrama cliente-servidor vía Internet (U.I.A 2011)

La arquitectura común de los sistemas de mensajería instantánea, ésta en el sistema cliente-servidor permite a los servicios un cierto control sobre los usuarios que los utilizan; mantiene una infraestructura que permite el servicio centralizado.

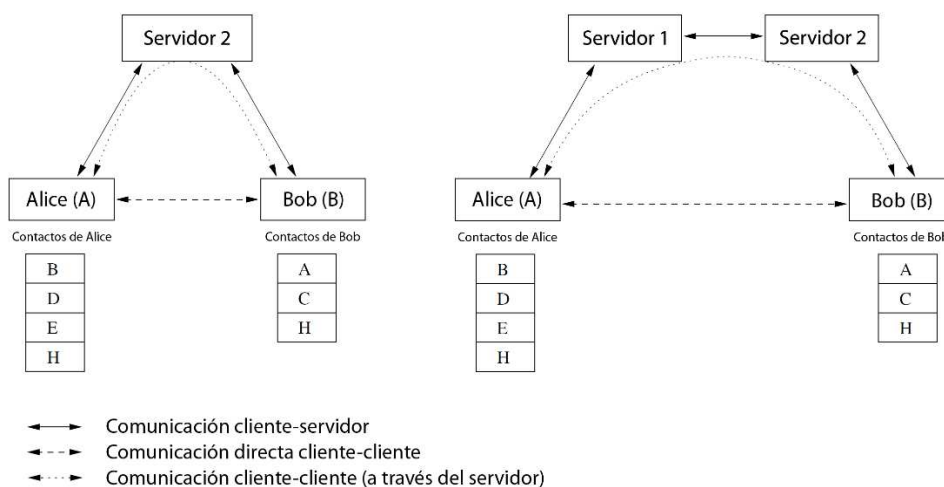


Figura. 5 Arquitectura general de un servicio de mensajería instantánea (U.P.M. Delgado L. 2013)

De acuerdo con la Figura 5, se toma un ejemplo donde los usuarios 'Alice' y 'Bob' se registran en el servidor (autenticación, actualización del estado, etc.) y acceden a través de él a las solicitudes de estado del resto de sus contactos. Si Alice quiere comunicarse con Bob, simplemente tendrá que enviar al servidor una petición de contacto y éste establecerá la comunicación entre ellos. En el caso de que Bob quiera enviar un fichero a Alice, enviará una solicitud al servidor, éste preguntará a Alice y en caso de que la acepte, se indicará a Bob la información necesaria para que se conecte con Alice. (Delgado L. 2013)

En la figura 5 tomando las bases de investigación de la universidad encontraremos la arquitectura cliente-servidor es un modelo de diseño de software en el que tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras. (U.I.A 2011)

Sistemas operativos

Los sistemas operativos se basan en el desarrollo de un software, y este es el que permite iniciar o encender un ordenador o dispositivo móvil y así gestionar todos los recursos del ordenador o dispositivo tanto del hardware como del software, permitiendo así la comunicación entre usuario y el ordenador o el dispositivo. En los sistemas operativos móviles usados para dispositivos móviles se encarga en parte del acceso, protección y administración de usuarios; comunicación entre usuarios y entre procesos. En la figura 6 encontramos la base de un SO móvil Unix.

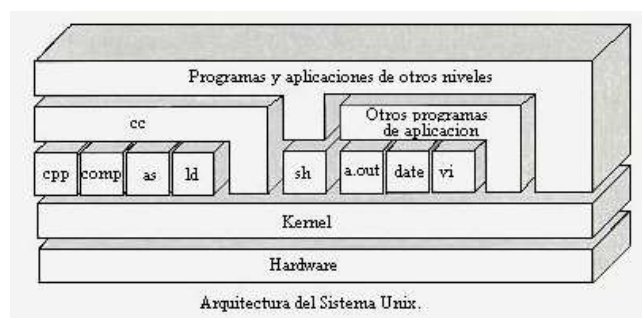


Figura 6. Arquitectura general del núcleo un sistema operativo. (UCT, 2017)

Los sistemas operativos en los PCs que utilizan Windows o Linux, en los SO móviles los dispositivos móviles tienen sus sistemas operativos como Android, iOS, Windows Phone, entre otros.

En la figura 7 de acuerdo a unas estadísticas NetMarketShare, la cuota de mercado de sistemas operativos móviles a principios de 2017 es el siguiente: Android 66,71 % (en países como España las diferencias son más significativas, donde Android tiene más del 90% de la cuota de mercado), iOS 29,55 %, Windows Phone 1,41 %, BlackBerry OS 0,37 %

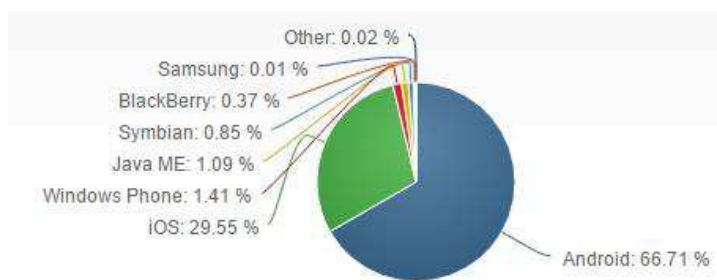


Figura 7. Diagrama de estadísticas del mercado en Sistemas Operativos Móviles (Acuña J. 2017)

Servidores

Los servidores son los ordenadores o maquina informática la cual está al servicio o apoyo otras máquinas o dispositivos las cuales son usadas por los clientes y los que le suministran a estos, todo tipo de información. En la figura 8 tendremos el esquema general, de cliente servidor y en este se basa gran parte de internet.

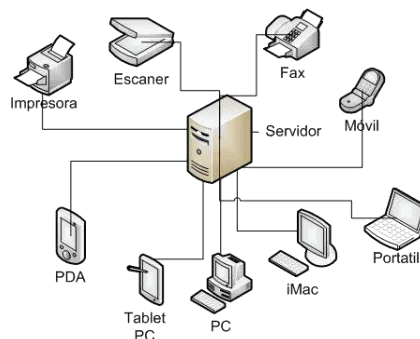


Figura 8. Esquema general Cliente-Servidor (A.P.R. 2008)

Como se puede observar en la figura 8, Se encuentra una máquina (servidor) la cual realiza la comunicación con variados clientes, todos solicitando algún tipo de información. Esta información puede ser desde archivos de texto, video, audio, imágenes, emails, aplicaciones, programas, consultas a base de datos, entre otros. (Room. 2016)

Marco legal

Algunas normas y leyes colombianas que se deben tener presente para el desarrollo de este proyecto son las que se describen por el congreso nacional de la república de Colombia y el ministerio de tecnologías de la información de acuerdo a la ley 1341 de TIC, ya que es la entidad que se encarga de diseñar, adoptar y promover las políticas, planes, programas y proyectos del sector de las tecnologías de información y las comunicaciones.

El ministerio de tecnologías de la información hace énfasis en la ley 1341 de TIC debe contar con el desarrollo de contenidos y aplicaciones, la protección a los usuarios, la formación de talento humano en estas tecnologías y su carácter transversal son pilares para la consolidación de las sociedades de la información y del conocimiento e impactan en el mejoramiento de la inclusión social y de la competitividad del país. Este marco legal provee una visión unificada aplicable.

También se maneja la neutralidad tecnológica donde se ve el principio normativo y regulatorio según el cual el Estado garantiza “la libre adopción de tecnologías, teniendo en cuenta recomendaciones, conceptos y normativas de los organismos internacionales competentes e idóneos en la materia, que permitan fomentar la eficiente prestación de servicios, contenidos y aplicaciones que usen las TIC y garantizar la libre y leal competencia, y que su adopción sea armónica con el desarrollo ambiental sostenible (artículo 2 Ley 1341/09).

Frente a la realidad de que la penetración de telefonía móvil es más importante que la penetración de computadores en los países en desarrollo, si los proveedores de redes y servicios móviles pueden ofrecer servicios convergentes, así como contenidos y aplicaciones, se convertirán en un motor de masificación del acceso y uso de TIC

También se evalúa al permitir que las redes de telecomunicaciones se conviertan en una plataforma dinamizadora del mercado de contenidos y aplicaciones, la convergencia tecnológica genera nuevas oportunidades económicas para los usuarios. En efecto, una expansión de la demanda por contenidos y aplicaciones es un factor impulsador del crecimiento económico. La expansión del sector de medios y entretenimiento a través de la telefonía móvil ya era una realidad para finales de 2007

La Ley brinda una definición de tecnologías de la información y las comunicaciones, así como del sector económico relacionado. Esto es un logro significativo ya que aun en escenarios internacionales no se ha llegado a un consenso de definición de TIC. Así mismo, esta definición permite circunscribir los límites de la intervención y/o de las políticas de promoción en el sector TIC. Adicionalmente, esta definición claramente incorpora los nuevos aspectos de énfasis sectorial como son las tecnologías de la información, las aplicaciones y los dispositivos.

El 5 de enero de 2009, el Congreso de la República de Colombia promulgó la Ley 1273 denominado “De la Protección de la información y de los datos”- y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones, se deben tener en cuenta normas y artículos importantes como los siguientes

Artículo 269E: USO DE SOFTWARE MALICIOSO. El que, sin estar facultado para ello, produzca, trafique, adquiera, distribuya, venda, envíe, introduzca o extraiga del territorio nacional software malicioso u otros programas de computación de efectos dañino

Artículo 269G: SUPLANTACIÓN DE SITIOS WEB PARA CAPTURAR DATOS PERSONALES. El que con objeto ilícito y sin estar facultado para ello, diseñe, desarrolle, trafique, venda, ejecute, programe o envíe páginas electrónicas, enlaces o ventanas emergentes, incurrirá en pena de prisión de y en multa.

En la misma sanción incurrirá el que modifique el sistema de resolución de nombres de dominio, de tal manera que haga entrar al usuario a una IP diferente en la creencia de que acceda a su banco o a otro sitio personal o de confianza, siempre que la conducta no constituya delito sancionado con pena más grave.

Es primordial mencionar que este artículo tipifica lo que comúnmente se denomina “phishing”, modalidad de estafa que usualmente utiliza como medio el correo electrónico pero que cada vez con más frecuencia utilizan otros medios de propagación como por ejemplo la mensajería instantánea o las redes sociales. Según la Unidad de Delitos Informáticos de la Policía Judicial (Dijín).

Metodología

Esta investigación se hace desde un enfoque cuantitativo por medio de experimentación y el diseño de un prototipo tecnológico basado en el desarrollo de software, por un Reporting Services que proporcione un diseñador gráfico de consultas basado en datos con los que se pueden crear consultas y recuperar datos de una base de datos relacional para un conjunto de datos de informe del Diseñador de gráficas.

Cuando se indica del enfoque cuantitativo se habla de un procedimiento de decisión que pretende señalar, entre ciertas alternativas, usando magnitudes numéricas que pueden ser tratadas mediante herramientas del campo de la estadística. Por eso la investigación cuantitativa se produce por la causa y efecto de las cosas. Por ejemplo, si tienes una unidad monetaria y compras un chicle ya no tendrás esa unidad monetaria.

Para que exista metodología cuantitativa se requiere que entre los elementos del problema de investigación exista una relación cuya naturaleza sea representable por algún modelo numérico ya sea lineal, exponencial o similar. Es decir, que haya claridad entre los elementos de investigación que conforman el problema, que sea posible definirlo, limitarlos y saber exactamente dónde se inicia el problema.

Administración del proyecto

Tabla 1.

Cronograma de Actividades

Acciones	Seman a 1	Seman a 3	Seman a 5	Seman a 7	Seman a 9	Seman a 11	Seman a 13	Seman a 15
	Seman a 2	Seman a 4	Seman a 6	Seman a 8	Seman a 10	Seman a 12	Seman a 14	Seman a 16
Investigación Anteproyecto	X							
Creación Front App		X			X			
Creación PHP			X	X				
Creación de Json			X	X				
Conexiones al servidor				X	X			
Organización de Tablas en Bases de Datos			X	X	X			
Diseño de Plan de Pruebas					X			
Ejecución de Pruebas					X	X		
Levantamiento de Documentación						X	X	
Cierre y Entrega							X	X

Nota: Tabla construida a partir de los aportes de (Labrador González, 2018)

Presupuesto

El presupuesto se complementó de \$300.000 por temas de investigación y cursos de desarrollo.

Desarrollo del problema de investigación

La metodología se encuentra enmarcada en cinco fases como se muestra en la figura 9, denominadas: análisis, diseño, desarrollo, pruebas de funcionamiento y entrega. A continuación

se describe cada una de las actividades que intervienen en el desarrollo de la propuesta.

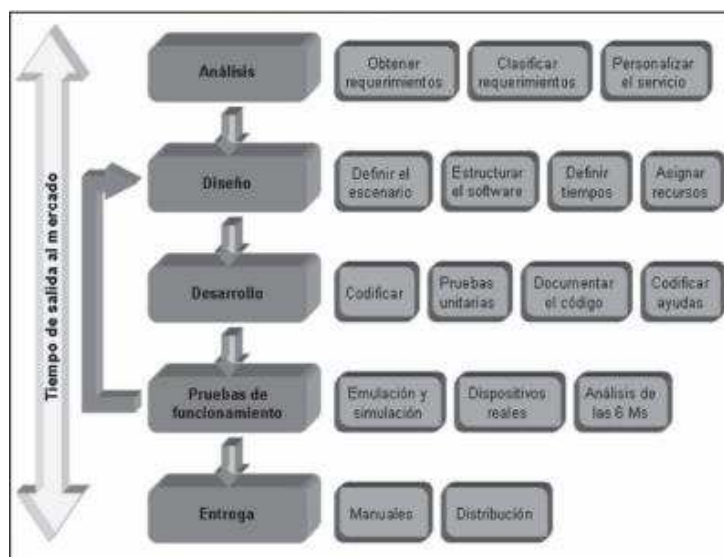


Figura 9. Proceso de la metodología en el desarrollo de App Móvil.

Infraestructura de Red

En el desarrollo se identificaron los tipos de nubes que pueden intervenir en el App por la estructura de soluciones que nos muestran entre ellos manejos:

- Nubes públicas: En estas se encuentran las aplicaciones de muchos clientes pueden estar compartiendo recursos de servidores, sistemas de almacenamiento y otras infraestructuras de la nube.
- Nubes privadas: Dirigidas a los clientes que necesitan una alta protección de datos y escritos SLA (Service Level Agreement). Estas nubes cuentan con una infraestructura administrada por un solo cliente el cual es el controla y dirige que aplicaciones deben correr y donde, ya que el cliente sería el propietario del servidor, red, y disco y decide que usuarios están autorizados a utilizar la infraestructura de la nube.
- Nubes híbridas: estas nubes combina los modelos de las nubes públicas y las nubes Privadas, ya que el cliente es propietario de algunas partes de la infraestructura y también comparte otras, aunque de una manera controlada.

En el desarrollo encontramos varios proveedores de nubes públicas las cuales son las que nos basaríamos en el desarrollo del App; entramos entre las más utilizadas en el mercado las siguientes:

- Amazon web services (AWS): es una plataforma de servicios que ofrecen en sus nubes en potencias de cómputo, manteniendo buenos desempeños en almacenamiento de base de datos, realiza las entregas de contenido entre otras funciones dando una gran ayuda a las empresas a escalar y avanzar en sus desarrollos. Sus usos son muy reconocidos en aplicaciones como Dropbox, Foursquare, HootSuite. AWS ha conseguido ser una de las ofertas internacionales más importante de la computación de servidores en la nube, ya que es considerado como el pionero en este campo.

AWS ofrece un amplio portafolio de productos globales basados en la nube, las cuales se dirigen para aplicaciones de informática, bases de datos, móviles, redes, análisis, almacenamiento, herramientas de administración, IoT, herramientas para desarrolladores, seguridad y entre otras. Todos sus servicios ofrecidos ayudan a las empresas avanzar grandemente, reduciendo costos de TI y escalar. ASW cuenta con el respaldo de una amplia variedad de cargas de trabajo, como las aplicaciones web y móviles, desarrollo de juegos, el almacenamiento y procesamiento de datos, almacenamiento en general, el archivado entre otras.

- Google Cloud: conocida también como nube de google, es también una plataforma que ha obtenido todas las aplicaciones y desarrollos web que ofrece google, en ella se crean ciertos tipos de soluciones por medio de la tecnología que manejan en su nube ya que es de destacar su rapidez y la estabilidad que tiene su infraestructura en las aplicaciones del buscador. Su nube se caracteriza por proveer resultados de búsqueda en milisegundos, también posee un gran espacio de almacenamiento para todos los usuarios de Gmail y su red global conecta en todos los sitios del planeta, y al usar la plataforma de nube de google Cloud tiene acceso a todas las innovaciones de google.

En Google Cloud le permite al usuario utilizar la infraestructura de base, el análisis de datos y un sistema de aprendizaje automático de google, GC cuenta con una amplia seguridad y cuenta con todas las funciones que necesitaría cualquier usuario o cualquier empresa, ya que brinda todo lo que se necesita para crear y escalar aplicaciones dado recursos informáticos desde máquinas virtuales con relación precio/rendimiento que ofrece ventajas probadas hasta una plataforma de

desarrollo de aplicaciones totalmente administradas como compute Engine, App Engine, Container Engine. Maneja el almacenamiento y bases de datos escalables, recientes y de alto rendimiento para sus aplicaciones, bases como Cloud Storage, Cloud SQL, Cloud Bigtable. Maneja miento de herramientas para productos de red definidos mediante el software de última generación en la red de fibra privada de google como Cloud Virtual Network, Cloud Load Balancing, Cloud CDN. Maneja también procesos en Big Data donde hace almacenamientos de datos totalmente administrado, procesamiento de streaming y por lotes, exploración de datos, Hadoop/Spark y mensajería fiable. Su identidad y seguridad controla el acceso y la visibilidad de los recursos en una plataforma protegida por el modelo de seguridad de google.

- Microsoft Azure: desarrollado por Microsoft como servicio de la nube, en ella ofrece Varios servicios para aplicaciones, servicios de almacenamiento de aplicaciones en alguno de los centros de procesamiento de datos de Microsoft para que se ejecute sobre su infraestructura (Cloud Computing) hasta servicios de comunicación segura y federación entre aplicaciones. Es muy respaldado ya que es uno de los líderes que maneja plataforma de nube.

Microsoft Azure es uno de los más desatacados ya que en sus servicios de procesos ejecuta distintas aplicaciones basadas en Windows server, y pueden ser creadas por medio de .net framework en lenguajes de programación tales como C# o visual Basic, C++, Java entre otros lenguajes muy utilizados. Su almacenamiento es una de las características más fuertes ya que puede almacenar objetos binarios grandes (blobs) donde proporcionan colas para la comunicación entre los componentes de las aplicaciones de Windows azure donde ofrece un tipo de tablas con un lenguaje de consulta simple. También tiene la posibilidad de desplegar de una forma sencilla a máquinas virtuales con Windows server o con distribuciones de Linux gracias a sus servicios de infraestructura. Azure maneja un sistema de controlador de tejidos donde ejecuta un gran número de máquinas en un solo centro de datos de Windows azure formando un conjunto armónico, los servicios de proceso y almacenamiento se implementa encima de toda esta eficacia de procesamiento. Su red de entrega de contenido (CDN) en el almacenamiento del cache de los datos a los que se accede frecuentemente cerca de sus usuarios agiliza el acceso a estos datos, Azure también permite conexiones de las empresas u organizaciones para la iteración de aplicaciones en la nube como si estuvieran dentro del propio firewall de la empresa u organización. Su sistema también maneja una administración de identidad y acceso por medio de

una solución Active Directory permite gestionar de forma centralizada y sencilla el control de acceso y la identidad, y esto la hace una de las soluciones perfectas para la administración de cuentas y la sincronización con directorios locales.

- Ibm cloud: también es un proveedor de la nube que tiene una buena integración en los entornos de la nube pública y privada, si infraestructura es muy segura, escalable y flexible, lo que brinda soluciones empresariales que hacen que IBM Cloud el líder del mercado de la nube híbrida, ya que maneja beneficios claves como tecnologías abiertas que garantizan la interoperabilidad y la flexibilidad, maneja una integración híbrida donde describe datos existentes con una nube híbrida integrada, usa datos y analíticas accesibles donde muestra a los clientes datos más detallados con recursos de computación analítica avanzados y poderosos.

Implementación de servicio web Json

Dados los recursos de la empresa SDT de ingeniería se usará Microsoft Azure como la plataforma de servidor desarrollado para la aplicación ya que las máquinas virtuales azure pueden crearse a través de Azure Portal. Este método proporciona una interfaz de usuario basada en el explorador para crear y configurar máquinas virtuales y todos los recursos asociados. Esta guía de inicio rápido le lleva paso a paso por la creación de una máquina virtual y la instalación de un servidor web en ella.

Desde allí se crea la máquina virtual por medio de la creación del par de claves SSH, ya que desde un Shell de bash, se ejecuta este comando, la salida del comando influye en el nombre del archivo de clave pública.

```
bash
```

```
ssh-keygen -t rsa -b 2048
```

En la creación de la máquina virtual, SDT con sus usuarios en Microsoft Azure realizaron el desarrollo del servidor GNU/Linux Ubuntu server 14.04 LTS ya que este esta soportado durante cinco años por Canonical ZFS, también es compatible con LXD OpenStack alojando el controlador, cuenta con Nova LXD donde implementa instancias de OpenStack como

contenedores del sistema, incrementando dramáticamente el paquete, usa el Juju OpenStack por densidad de nodos con una implementación automatizada de OpenStack en contenedores del sistema LXD.

GNU/Linux Ubuntu server 14.04 LTS cuenta con la certificación de Microsoft para alojar Windows Server 2012 y Windows Server 2008 como invitados, en virtud de su Programa de validación de virtualización de servidores (SVVP), trae la versión actualizada de Mitaka incluida la instalación automatizada, la puesta en cola / notificación y la integración de la base de datos como servicio, en conclusión usamos este servidor ya que tendremos OpenStack software, Juju, MAAS, LXD on IBM LinuxONE mainframe OpenStack software, Juju, MAAS, LXD on IBM LinuxONE mainframe.

MySQL en Ubuntu 14.04

MySQL es un sistema de gestión de bases de datos de código abierto, comúnmente instalado como parte de la popular LAMP (Linux, Apache, MySQL, PHP / Python / Perl). Utiliza una base de datos relacional y SQL (Structured Query Language) para gestionar sus datos.

La versión corta de la instalación es simple: actualizar el índice de paquetes, instalar el mysql-server paquete, y luego ejecutar los scripts de seguridad e inicialización de base de datos incluidos.

```
$ sudo apt-get update
$ sudo apt-get install mysql-server
$ sudo mysql_secure_installation
$ sudo mysql_install_db
```

Hay dos formas de instalar MySQL. Se puede utilizar cualquiera de una de las versiones incluidas en el repositorio de paquetes APT por defecto (que son 5.5 y 5.6), o puede instalar la versión más reciente (actualmente 5.7) añadiendo manualmente primer repositorio de MySQL.

La usada fue MySQL-server paquete de APT, que simplemente instala la última versión de su distribución de Linux. En el momento de la escritura, que es 5.5, pero siempre se puede actualizar a otra versión posterior. En la instalación de MySQL también se debe instalar el índice de paquetes en el servidor e instalar el paquete con apt-get.

```
$ sudo apt-get update
$ sudo apt-get install mysql-server
```

Se solicita crear una contraseña root durante la instalación. Elegir uno seguro y asegúrese de recordar que, ya que lo necesitará más adelante.

En primer lugar, actualizar el índice de paquetes en el servidor.

```
$ sudo apt-get update
```

Se instala el MySQL 5.5, instale el mysql-server-5.5 o 5.6 paquete

```
$ sudo apt-get install mysql-server-5.5
```

```
$ sudo apt-get install mysql-server-5.6
```

Se solicita crear una contraseña de root durante la instalación.

En la configuración de MySQL en primer lugar, se ejecuta el script de seguridad incluido. Esto cambia algunas de las opciones predeterminadas menos seguros para cosas como las conexiones de root remotos y usuarios de ejemplo.

```
$ sudo mysql_secure_installation
```

Esto le pedirá la contraseña de root que ha creado en el paso uno. Puede pulsar enter para aceptar los valores predeterminados para todas las preguntas siguientes, con la excepción de la que le pregunta si desea cambiar la contraseña de root. Usted acaba de establecer que en el paso uno, por lo que no tiene que cambiar ahora.

A continuación, vamos a inicializar el directorio de datos de MySQL, que es donde MySQL almacena sus datos. La forma de hacerlo depende de la versión de MySQL que se está ejecutando. Puede comprobar su versión de MySQL con el siguiente comando.

```
$ mysql --version
```

Se visualizará una salida como esta

```
Salida
mysql Ver 14.14 Distrib 5.7.11, for Linux (x86_64) using EditLine wrapper
```

Después tendremos pruebas de MySQL debería haber comenzado a ejecutarse automáticamente. Para probar esto, comprobar su estado.

```
$ service mysql status
```

Verá la siguiente salida (con un PID diferente).

```
Salida
mysql start/running, process 2689
```

Para una comprobación adicional, se puede intentar la conexión a la base de datos utilizando la mysql admin herramienta, que es un cliente que le permite ejecutar comandos administrativos. Por ejemplo, este comando le dice a conectarse a MySQL como **root** (-u root), solicite una contraseña (-p), y devolver la versión.

```
$ mysqladmin -p -u root version
```

Y se visualizará la siguiente salida

Salida

```

mysqladmin Ver 8.42 Distrib 5.5.47, for debian-linux-gnu on x86_64
Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Server version      5.5.47-0ubuntu0.14.04.1
Protocol version    10
Connection         Localhost via UNIX socket
UNIX socket        /var/run/mysqld/mysqld.sock
Uptime:            4 min 15 sec

Threads: 1  Questions: 602  Slow queries: 0  Opens: 189  Flush tables: 1  Open tables: 41

```

Esto ya indica que MySQL ya está funcionando.

CODIGO DEL JSON EN LA BASE DE DATOS UTILIZANDO PHP

Json básicamente significa JavaScript Object Notation. JSON es bastante simple, fácil y ligero y sirve para el intercambio y almacenamiento de datos desde y hacia la base de datos. Es una alternativa al conocido XML (Extensible Markup Language). Los datos en formato JSON se almacenan en el archivo JSON con la extensión “.json”. Los datos presentes en el archivo JSON son compatibles con todos los navegadores modernos actuales.

```
<?php
```

```
//connect and select the database
```

```
$connect = mysql_connect("localhost","root","") or die('Database Not Connected. Please Fix the Issue! ' . mysql_error());
```

```
mysql_select_db("jsondb", $connect);
```

```
// get the contents of the JSON file $jsonCont = file_get_contents('studJson.json');
```

```

//decode JSON data to PHP array $content = json_decode($jsonCont, true);

//Fetch the details of Student $std_id = $content['stdID'];

$std_Client = $content['stdData']['stdName'];

$std_Divices = $content['stdData']['stdBrand'];

$std_Zone = $content['stdData']['stdZone'];

$std_Date = $content['stdData']['stdNo'];

$std_Statistics = $content['stdData']['stdStatistics ']['stdStatistics t'];

$std_User = $content['stdData']['stdUser']['stdPasssword'];

$std_poll = $content['stdData']['stdpoll']['stdplace'];

$std_postalEvaluations = $content['stdData']['stdZone']['stdAnswer'];

$std_dept = $content['stdEdu']['stdDept'];

$std_sem = $content['stdEdu']['stdSemester'];

$std_major = $content['stdEdu']['stdMajor'];

//Insert the fetched Data into Database

$query = "INSERT INTO stdtable(std_id, std_name, std_User, std_Password, std_Client,
std_Devices, std_Date, std_zone, std_poll, std_place, std_semstr, std_major) VALUES('$std_id',
'$std_table', '$std_client', '$std_', '$std_no', '$std_street', '$std_city', '$std_country', '$std_postal',
'$std_dept', '$std_sem', '$std_major')";

if(!mysql_query($query,$connect)) {

die('Error : Query Not Executed. Please Fix the Issue! ' . mysql_error());

```

```

} else{

echo "Data Inserted Successully!!!";

} ?>

```

En el código anterior, lo primero que hacemos es establecer una conexión de base de datos mediante la función PHP “mysql_connect ()”. A continuación, después de la conexión, seleccionamos la base de datos usando la función de PHP “mysql_select_db ()”.

```

$datosclientes = json_decode($json, true);

$server = "app3.datawifi.co";

$user = "cristian";

$pass = ""; $bd = "cristian_json";

//Creamos la conexión

$conexion = mysqli_connect($server, $user, $pass,$bd)

or die("Ha sucedido un error inexperado en la conexion de la base de datos");

foreach ($datosclientes as $cliente) {

mysqli_query($conexion,"INSERT INTO clientes (nombre, tabla, dispositivos, estadística,
fechas, zonas, valoraciones, encuestas) VALUES
('".$dispositivos['marca'].",".$Zonas['lugar'].",".$Dispositivos['Marca/Nombre'].",".$Fechas['de
sdel'].",".$Zonas['localidad'].",".$Va loraciones['Respuestas].")"); }

mysqli_close($conexion); }

```

CakePHP

CakePHP está diseñado para hacer que las tareas comunes de desarrollo web sean simples y fáciles. Al proporcionar una caja de herramientas todo en uno para que pueda comenzar, las diversas partes de CakePHP funcionan bien juntas o por separado.

El objetivo de esta visión general es presentar los conceptos generales en CakePHP, y darle una visión general rápida de cómo esos conceptos se implementan en CakePHP.

La capa Modelo representa la parte de su aplicación que implementa la lógica comercial. Es responsable de recuperar los datos y convertirlos en los principales conceptos significativos de su aplicación. Esto incluye procesamiento, validación, asociación u otras tareas relacionadas con el manejo de datos. En el caso de una red social, la capa Modelo se ocuparía de tareas tales como guardar los datos del usuario, guardar asociaciones de amigos, almacenar y recuperar fotos de usuarios, buscar sugerencias para nuevos amigos, etc. Los objetos modelo pueden considerarse como "Amigo", "Usuario", "Comentario" o "Foto". Si quisiéramos cargar algunos datos de nuestra userstable, podríamos hacer:

```
use Cake \ ORM \ TableRegistry ;

$ users = TableRegistry :: get ( 'Usuarios' );

$ query = $ users -> find ();

foreach ( $ query as $ row ) {

echo $ row -> username ;

}
```

La capa de Vista representa una presentación de datos modelados. Al estar separado de los objetos del Modelo, es responsable de usar la información que tiene disponible para producir cualquier interfaz de presentación que su aplicación pueda necesitar.

Por ejemplo, la vista podría usar datos modelo para representar una plantilla de vista HTML que lo contenga, o un resultado con formato XML para que otros consuman:

// En un archivo de plantilla de vista, representaremos un 'elemento' para cada usuario.

```
<? php foreach ( $ users as $ user ) : ?>
```

```

<li class = "usuario">

<? = $ this -> element ( 'user_info' , [ 'user' => $ user ] ) ?>

</ li>

<? php endforeach ; ?>

```

La capa de Vista proporciona una cantidad de puntos de extensión como ver Plantillas , Elementos y Ver Celdas para permitirle reutilizar su lógica de presentación.

La capa de Vista no se limita solo a HTML o a la representación de texto de los datos. Se puede usar para entregar formatos de datos comunes como JSON, XML y, a través de una arquitectura conectable, cualquier otro formato que pueda necesitar, como CSV.

La capa de controlador maneja las solicitudes de los usuarios. Es responsable de generar una respuesta con la ayuda de las capas Modelo y Vista

Un controlador puede verse como un administrador que garantiza que todos los recursos necesarios para completar una tarea se deleguen a los trabajadores correctos. Espera peticiones de clientes, comprueba su validez de acuerdo con las reglas de autenticación o autorización, delega el procesamiento de datos al modelo, selecciona el tipo de datos de presentación que los clientes aceptan y finalmente delega el proceso de representación en la capa Vista. Un ejemplo de un controlador de registro de usuario sería:

```

función pública add ()

{

$ usuario = $ this -> Users -> newEntity ();

if ( $ this -> request -> is ( 'post' ) ) {

$ user = $ this -> Users -> patchEntity ( $ usuario , $ this -> request -> getData ());

```



```

if ( $ this -> Users -> save ( $ usuario , ['validate' => 'registration' ])) {

$ this -> Flash -> success ( __ ( 'Ahora estás registrado.' ));

} else {

$ this -> Flash -> error ( __ ( 'Hubo algunos problemas.' ));      }

}

$ this -> set ( 'usuario' , $ usuario );

}

```

Puede observar que nunca hemos representado explícitamente una vista. Las convenciones de CakePHP se encargarán de seleccionar la vista correcta y renderizarla con los datos de vista que preparamos set:

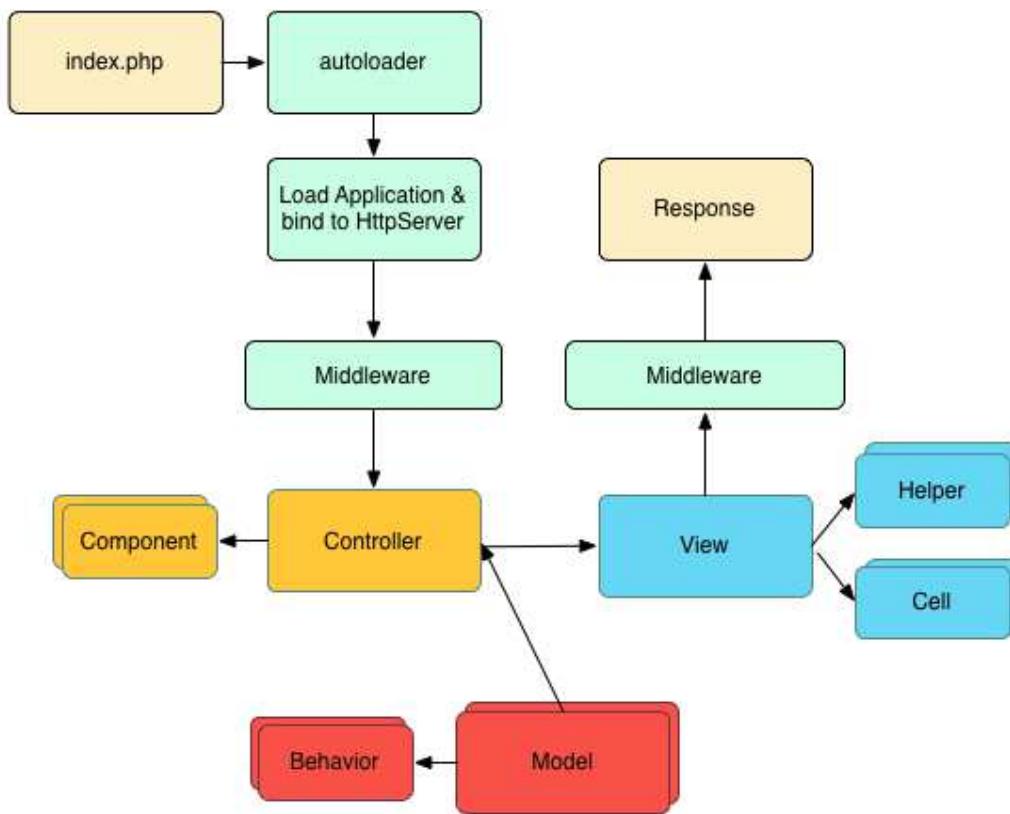


Figura 10.Ciclo de Solicitud CakePHP (Coke P, 2017).

El ciclo típico de solicitud de CakePHP comienza con un usuario que solicita una página o recurso en su aplicación. En un nivel alto, cada solicitud pasa por los siguientes pasos:

- Las reglas de reescritura del servidor web dirigen la solicitud a `webroot / index.php`.
- Su aplicación está cargada y vinculada a un `HttpServer`.
- El `middleware` de su aplicación se inicializa.
- Se envía una solicitud y respuesta a través del `Middleware PSR-7` que usa su aplicación. Típicamente esto incluye trampas de errores y enrutamiento.
- Si no se devuelve ninguna respuesta del `middleware` y la solicitud contiene información de enrutamiento, se seleccionan un controlador y una acción.
- Se llama a la acción del controlador y el controlador interactúa con los Modelos y Componentes requeridos.
- El controlador delega la creación de respuestas en la Vista para generar la salida resultante de los datos del modelo.
- La vista utiliza `Helpers` y `Cells` para generar el cuerpo y los encabezados de la respuesta.
- La respuesta se envía de vuelta a través del `Middleware`.
- El `HttpServer` emite la respuesta al servidor web.

Estructura del desarrollo (App)

Frameworks

Existe gran variedad de frameworks para el desarrollo móvil, en las investigaciones que realizamos encontramos cada vez más opciones para desarrollar proyectos multiplataforma para IOS, Android entre otros sistemas operativos. En el siguiente listado mencionaremos

Las plataformas de desarrollo para IOS y Android.

- PhoneGap: la plataforma de desarrollo de Adobe en ella se cuenta la distribución Open Source de Apache Cordova, el Framework de referencia para el desarrollo de WebApps(*apps* multiplataforma basadas en tecnología web abierta como HTML, CSS y JS Adobe ofrece a sus clientes no solo las herramientas de diseño líderes del mercado, también una

poderosa herramienta multiplataforma para crear WebApps sin necesidad de adquirir los conocimientos necesarios para programar en los lenguajes nativos de iOS y Android, pero sin renunciar al acceso de las APIs que controlan el hardware de los dispositivos como cámara o GPS. Ofreciendo de esta forma una experiencia completa de *App* nativa. Su principal ventaja, es que no hay modalidades ni precios, Adobe PhoneGap es gratuita.

- Xamarin: plataforma de Microsoft para el desarrollo de Android y IOS una de las plataformas favoritas de muchos desarrolladores ya que cuenta apps nativas para los sistemas operativos móviles y también para Windows 10. Cuenta con un kit de herramientas de interfaz de usuario multiplataforma respaldada de forma nativa que permite a los desarrolladores crear interfaces de usuario. Las interfaces de usuario se representan mediante los controles nativos de la plataforma de destino, lo que permite que las aplicaciones desarrolladas en Xamarin.Forms conserven la apariencia adecuada para cada plataforma.

- Appery.io: este framework maneja sus desarrollos en la nube, esta cuenta con una plataforma de desarrollo de *apps* basada *en la nube* y que no requiere de instalación en el disco duro local. Utilizado, según sus creadores, por grandes fabricantes de la telefonía como Samsung o AT&T. Destacando su compatibilidad con las principales soluciones de Base de Datos e información empresarial como Oracle, SQL Server, MySQL o Sharepoint. Con soporte para Apache Cordova para acceder a las funcionalidades nativas de las diferentes plataformas de desarrollo como iOS, Android y Windows Phone (acelerómetro, cámara, GPS, micrófono, almacenamiento, etc.), y basado en tecnologías web como JS (JavaScript), jQuery Mobile, AngularJS y Bootstrap. Además de ofrecer opciones de trabajo en grupo con sincronización en tiempo real, gestión de usuarios y permisos, opciones de versiones y copias de seguridad.

- Appcelerator: una plataforma que se destaca, ya que cuenta con características similares a las anteriores plataformas mencionadas, pero con la particularidad de estar basada en eclipse, otro de los más conocidos framework de código abierto para el desarrollo de

aplicaciones. Appcelerator también usa la corriente de utilizar la tecnología web y los estándares abiertos como JavaScript para crear apps compatibles con cualquier dispositivo y sistema operativo, accediendo como en los casos anteriores, también a las APIs nativas que dan soporte al hardware.

Appcelerator no solo es un entorno de desarrollo, también ofrece funcionalidades de verificación automatizada de las apps para depurar errores de funcionamiento, así como acceso online al sistema de estadísticas de uso de cada uno de los desarrollados realizados con la plataforma o su conexión y compatibilidad con sistemas de Bases de Datos como MongoDB o MySQL.

- AppMachine: se puede definir como un constructor visual de apps todo tipo de público, tiene un asistente organizado por diferentes categorías de diseño desarrollo, configuración, promoción, publicación y análisis de métricas, con todo lo necesario para crear apps sencillas. AppMachine permite importar datos desde una web, redes sociales o servicios web mediante ficheros Excel, XML o JSON.

- Android Studio: Esta es la plataforma que hemos elegido para el desarrollo integrado del App en sistemas operativos de Android móviles, ya que Android Studio está basado en el software IntelliJ IDE, lo cual la hace soportable para Microsoft Windows, macOS y GNU/Linux y está enfocada específicamente a los desarrollos en Android puro. Esta plataforma fue elegida gracias a su soporte en la programación de aplicaciones, cuenta con herramientas Lint y funciona detectando el código no compatible entre diferentes arquitecturas o códigos confusos que no es capaz de controlar el compilador y así logra detectar los problemas de rendimiento y compatibilidad.

Cuenta con un sistema llamado ProGuard para optimizar y reducir el código del proyecto al exportarlo como apk esto es muy útil para los dispositivos con problemas de limitaciones de almacenamiento en su memoria, Cuenta también con la herramienta de Gradle la cual es la encargada de gestionar y automatizar la construcción de proyectos, como también el tipo de tareas de testing, compilado o empaquetado. Su diseño de editor también lo hace óptimo para la

edición de temas, su interfaz es especificada para el desarrollo de Android, posibilita el control de versiones accediendo a un repositorio de descargas de controladores de desarrollo, editor de diseño donde muestra la vista previa de los cambios realizados en el archivo XML, dando también alternativas en tiempo real de errores sintácticos, compatibilidad rendimiento antes de compilar la apk.

Por eso lo hemos desarrollado con parte de las herramientas en Android Studio ya que su entorno es recomendado para el desarrollo de aplicaciones en Android y dispone de todas las herramientas necesarias para el desarrollo, usando las mejores y más desarrolladas herramientas para facilitar la distribución del código y el trabajo.

- **Ionic:** Este framework fue en el que realizamos el desarrollo de la App y a continuación se documentara sus características y el paso a paso del desarrollo realizado.

Lo primero que se explicara, y de forma escueta, es qué es un MVC y de qué se compone:

El mvc

El mvc (Model-View-Controller o Modelo-Vista-Controlador), es un patrón de diseño que separa los datos, la lógica y las interfaces de usuario. Como su nombre indica, está separado en tres componentes: Modelo, Vista y Controlador. Está basado en la ideología de separación de conceptos y cumple perfectamente con los objetivos de los patrones de diseño.

Modelo: Es la capa encargada de los datos, es decir, la que se encarga de hacer peticiones a las bases de datos para enviar o recibir información. Estas bases de datos pueden estar alojadas de forma local en nuestra App o de forma remota en un servidor externo.

Vista: Se trata del código que nos permitirá presentar los datos que el modelo nos proporciona, como ejemplo podríamos decir que en una aplicación es el código HTML que nos permite mostrar la salida de los datos procesados.

Controlador: Es la capa que sirve de enlace entre la vista y el modelo. Envía comandos al modelo para actualizar su estado, y a la vista correspondiente para cambiar su presentación.

En el caso MVVM (Modelo Vista Modelo) la interacción entre la vista y el controlador será en los dos sentidos, el controlador muestra los datos en la vista y si en la vista hay un cambio de datos, se actualiza el modelo automáticamente.

framework Ionic

Ionic es una herramienta, gratuita y open source, para el desarrollo de aplicaciones híbridas basadas en HTML5, CSS y JS. Está construido con Sass y optimizado con AngularJS.

Sus principales características son alto rendimiento, Ionic está construido para ser rápido gracias a la mínima manipulación del DOM, con cero jQuery y con aceleraciones de transiciones por hardware. También Ionic utiliza AngularJS con el fin de crear un marco más adecuado para desarrollar aplicaciones ricas y robustas. Ionic no sólo se ve bien, sino que su arquitectura central es robusta y seria para el desarrollo de aplicaciones. Trabaja perfectamente con AngularJS. Otra característica es su centro Nativo Ionic se inspira en las SDK de desarrollo móviles nativos más populares, por lo que es fácil de entender para cualquier persona que ha construido una aplicación nativa para iOS o Android. Lo interesante, como sabéis, es que desarrollas una vez, y compilas para varios. También tiene la característica de tener un bonito diseño Limpio, sencillo y funcional. Ionic ha sido diseñado para poder trabajar con todos los dispositivos móviles actuales. Con muchos componentes usados en móviles, tipografía, elementos interactivos, etc. Por último y no menos importante un potente CLI Con un sólo comando podrás crear, construir, probar y compilar tus aplicaciones en cualquier plataforma.

Se analiza la anatomía de una aplicación iónica. Dentro de la carpeta que se creó, se tendrá una estructura de proyecto típica de Córdoba donde podamos instalar complementos nativos y crear archivos de proyecto específicos de la plataforma.

src / index.html es el principal punto de entrada para la aplicación, aunque su propósito es configurar scripts, CSS includes y bootstrap, o comenzar a ejecutar nuestra aplicación. No se pasara mucho de tiempo en este archivo. Para la aplicación funcione, con Ionic se busca la etiqueta <ion-App> en su HTML, y los siguientes scripts:

```
<!-- Ionic's root component and where the app will load -->  
<ion-app></ion-app>
```

<!-- The polyfills js is generated during the build process -->

<script src="build/polyfills.js"></script>

<!-- The vendor js is generated during the build process

It contains all of the dependencies in node_modules -->

<script src="build/vendor.js"></script>

<!-- The main bundle js is generated during the build process -->

<script src="build/main.js"></script>

<! - El componente raíz de Ionic y dónde se cargará la aplicación ->

<ion-App> </ ion-App>

<! - El polyfills js se genera durante el proceso de compilación ->

<script src = "build / polyfills.js"> </ script>

<! - El proveedor js se genera durante el proceso de compilación

Contiene todas las dependencias en node_modules ->

<script src = "build / vendor.js"> </ script>

<! - El paquete principal js se genera durante el proceso de compilación ->

<script src = "build / main.js"> </ script>

Todos estos scripts son generados por el sistema de compilación

./src/

Dentro del directorio src encontramos nuestro código. Aquí es donde tendrá lugar la mayor parte del trabajo para una aplicación Ionic. Cuando ejecutamos servicio iónico, nuestro código dentro de src / se transporta a la versión de JavaScript correcta que el navegador entiende

(actualmente ES5). Eso significa que podemos trabajar a un nivel superior utilizando TypeScript, pero compilar hasta la forma más antigua de JavaScript que el navegador necesita.

`src / App / app.module.ts` es el punto de entrada para nuestra aplicación.

Cada aplicación tiene un módulo raíz que básicamente controla el resto de la aplicación. Esto es muy similar a ng-app de Ionic 1 y AngularJS. Aquí también es donde se inicia la aplicación usando `ionicBootstrap`.

En este módulo, estamos configurando el componente raíz para `MyApp`, en `src / App / app.component.ts`. Este es el primer componente que se carga en la aplicación, y generalmente es un shell vacío para que se carguen otros componentes. En `app.component.ts`, se va configurando la plantilla para `src / App / app.html`, así que al echar un vistazo allí.

`./src/App/app.html` Está la plantilla principal para la aplicación en `src / App / app.html`:

```
<ion-nav id="nav" [root]="rootPage" #nav swipeBackEnabled="false"></ion-nav>
```

```
<ion-menu [content]="nav">
```

```
<ion-header>
```

```
<ion-toolbar>
```

```
<ion-title>Pages</ion-title>
```

```
</ion-toolbar>
```

```
</ion-header>
```

```
<ion-content>
```

```
<ion-list>
```

```
<button ion-item *ngFor="let p of pages" (click)="openPage(p)">
```

```
  {{p.title}}
```

```
</button>
```

```
</ion-list>
```

```
</ion-content>
```

```
</ion-menu>
```


En esta plantilla, se configura un menú de iones para que funcione como un menú lateral, y luego un componente de navegación iónica para actuar como el área de contenido principal. La propiedad [contenido] del menú de iones está vinculada a la variable de navegación local de nuestro ion-nav, por lo que sabe dónde debe animarse.

Componentes: Las aplicaciones iónicas están hechas de bloques de construcción de alto nivel llamados componentes. Los componentes le permiten construir rápidamente una interfaz para su aplicación. Ionic viene con una serie de componentes, incluidos modales, botones, ventanas emergentes y tarjetas.

Diseño de red móvil

La infraestructura de redes que se manejan para los dispositivos móviles se hace por medio de un código llamado Mobile country Code (MCC) y Mobile Network Code (MNC).), estos dispositivos móviles están conectados a las redes móviles a través de estaciones base (ejemplo base transceiver station (BTS), Access point, o satélite) que establecen y controlan las conexiones e interfaces funcionales entre las redes y los dispositivos móviles. Las peticiones de los usuarios de telefonía móvil y la información (por ejemplo ubicación) se transmiten a los procesadores centrales que están conectadas a los servidores que prestan servicios de red móvil. Los operadores de redes móviles pueden proporcionar servicios a los usuarios móviles como AAA (autenticación, autorización, y auditoría), basado en agentes y los datos de los suscriptores almacenados en bases de datos. Las peticiones de los suscriptores se entregan a la nube a través de Internet. En la nube los controladores, procesan las solicitudes para proporcionar a los usuarios móviles con el correspondiente servicio de cloud.

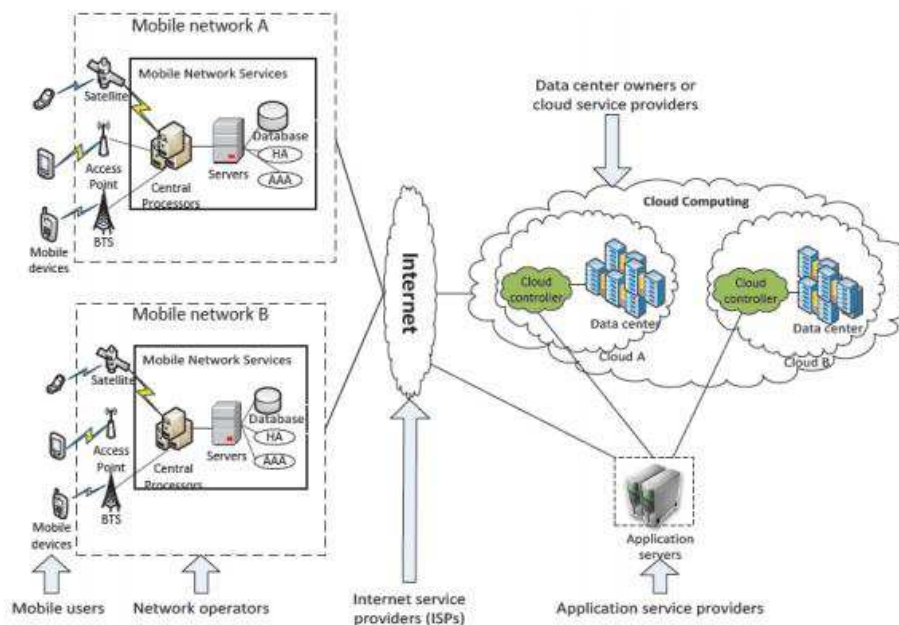


Figura 11. Arquitectura MCC en Dispositivos Móviles (Estigarribia. 2012)

Diseño de Software funcional

La estructura del desarrollo del software consiste en la conexión del web chat de páginas web de empresas u organizaciones donde son ofrecidos todos sus tipos de servicios, tales páginas cuentan con web chat donde sus clientes pueden realizar consultas más precisas de los servicios, se realizara las consultas por medio del servidor y la nube de Azure estas consultas podrán ser contestadas por agentes por medio de sus dispositivos móviles dándole un tiempo de respuesta optimo a los clientes. También se creara un sistema en el cual los agentes que cumplan con la labor de captar a los clientes, obtendrán un puntaje el cual será remunerado por su colaboración.

En la figura 12 vemos la infraestructura del software con la conexiones de hilos desde la web procesando la información en el servidor y enviándolas a los dispositivos móviles de los agentes usando las conexiones de red.

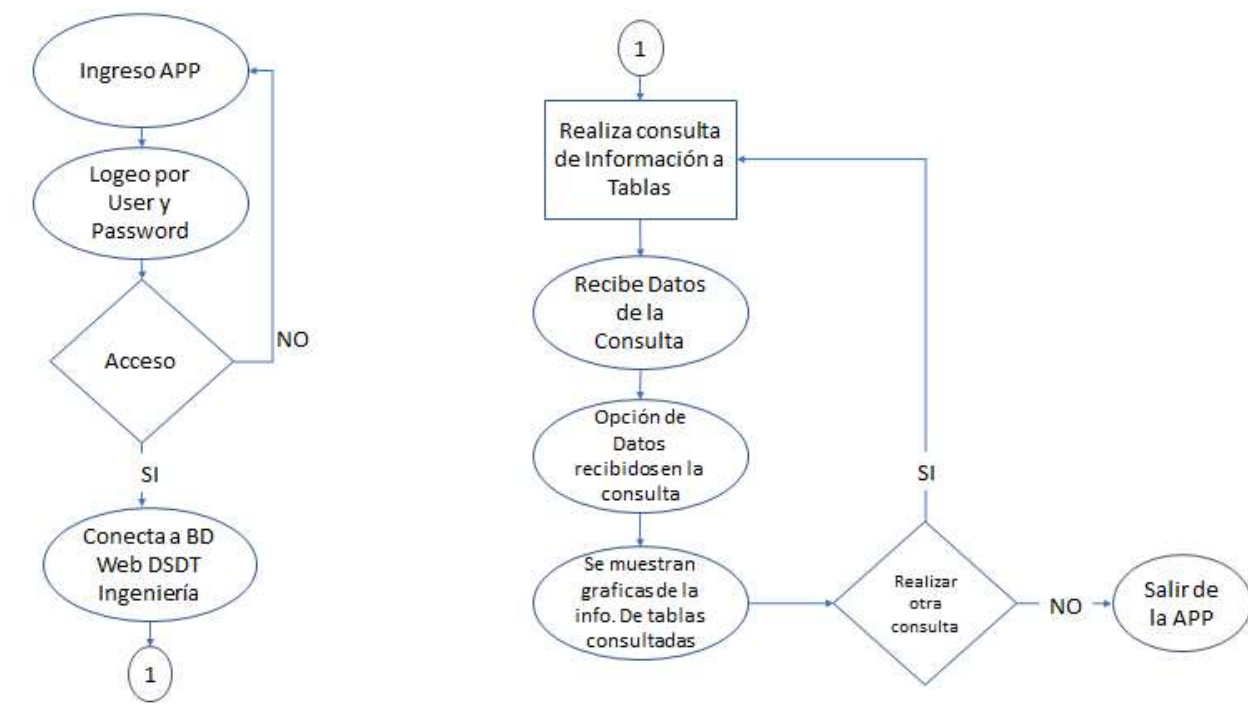


Figura 12. Infraestructura de Diseño de software (2018)

En la Figura 12 se observa la infraestructura de software que consiste en consultas a las Bases de datos de comunicación directos con un usuario en tiempo real, el cual el usuario realiza sus consultas a las bases de datos de mysql por web de la empresa, estas consultas realizadas por el usuario son enviadas como paquetes de datos al servidor procesando la información con los datos enviados por el cliente, verificando la identidad y creando un registro de la comunicación del cliente y el servidor a las bases de datos, después son cargados a la nube en la cual cuenta con los protocolos de comunicación que se especificó para estos servicios, los usuarios que ingresen estarán en estado disponibles por medio del App ya que estaría conectados directamente con el servidor de la base de datos, a este agente se le entregarían los paquetes de datos procesando las consultas realizadas por el usuario, por medio del software el cliente ya tendrá una comunicación directa con la base de datos se sabrá a que IP y puerto se establecen la comunicación. El agente podrá mostrar graficas de consultas por medio de su App que se encuentre conectada al servidor, repitiendo ese proceso y así llevando acabo todo el proceso en tiempo real. Cuando el agente se desconecte del App y se encuentre fuera de línea de la comunicación de los clientes se desconecta del servidor.

DESARROLLO DEL APP

Aquí observaremos los pantallazos de interfaz desarrollada en el App donde describiremos paso a paso su funcionamiento.

Como lo vemos en figura 13 en el menú del dispositivo móvil Android, encontraremos el icono de la Aplicación. Donde el agente podrá tener un fácil acceso para entrar en línea con el servidor configurado.



Figura 13. Front de Menú del Dispositivo Android (Labrador G. 2018)

Después de ingresar al icono de SDT SQL, ingresara automáticamente a la App, en donde se le solicitara ingresar sus datos con su usuario y contraseña para autenticarse en el servidor. En la figura 14 observaremos la configuración del desarrollo en la herramienta de Android Studio ya que usamos funcionalidades nativas del dispositivo.

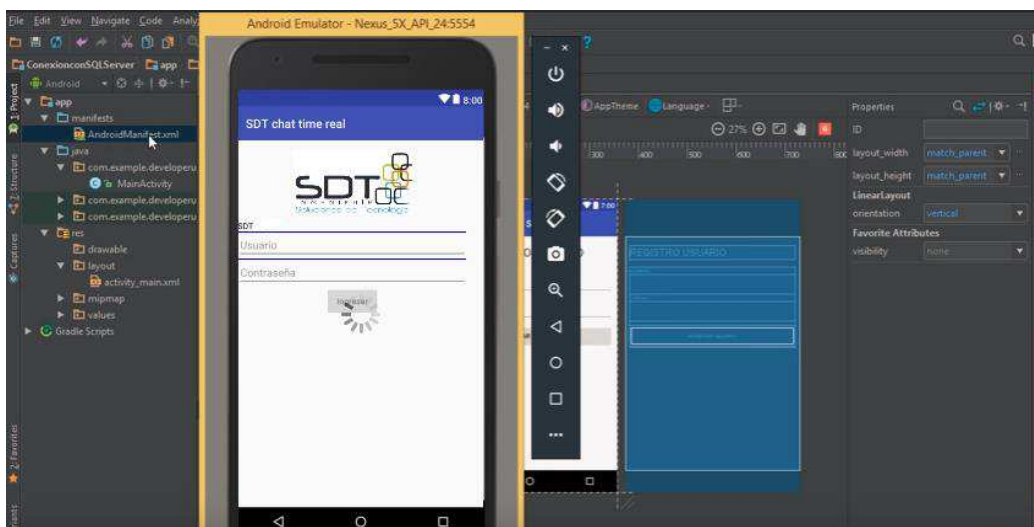


Figura 14. Desarrollo Front del Aplicativo (Labrador G. 2018)

En la figura 15 se muestra la evidencia del acceso por medio del Manifest.xml de Android Studio dándole permiso a internet para hacer conexión a la base de datos.

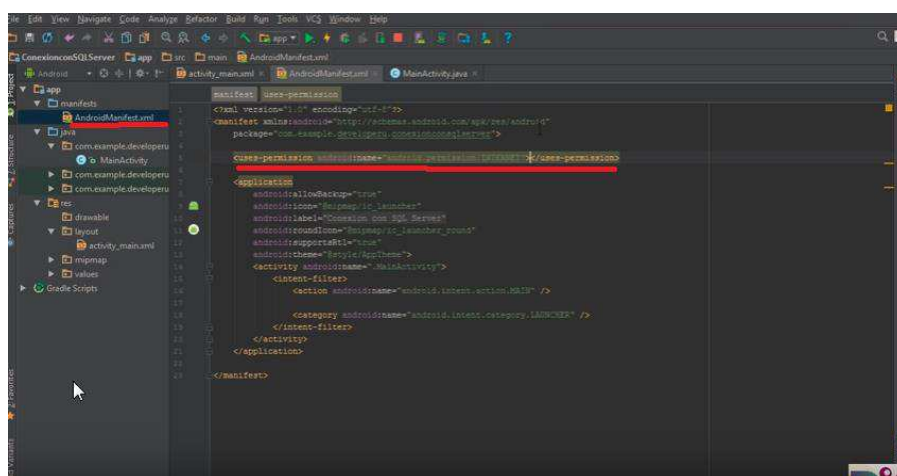


Figura 15. Código Manifest.xml (Labrador G. 2018)

También se realiza el uso de un Drawable conexión, en este caso el jtds de java, el cual nos permitirá conectarnos a una base de datos de sql server, para importar la librería o Driver, yendo a la carpeta Project en carpeta apps y libs, ingresando a la página del jtds descargara y descargamos el driver y lo acomodamos en la carpeta libs. Así como lo observamos en la figura 16 la configuración de las carpetas.

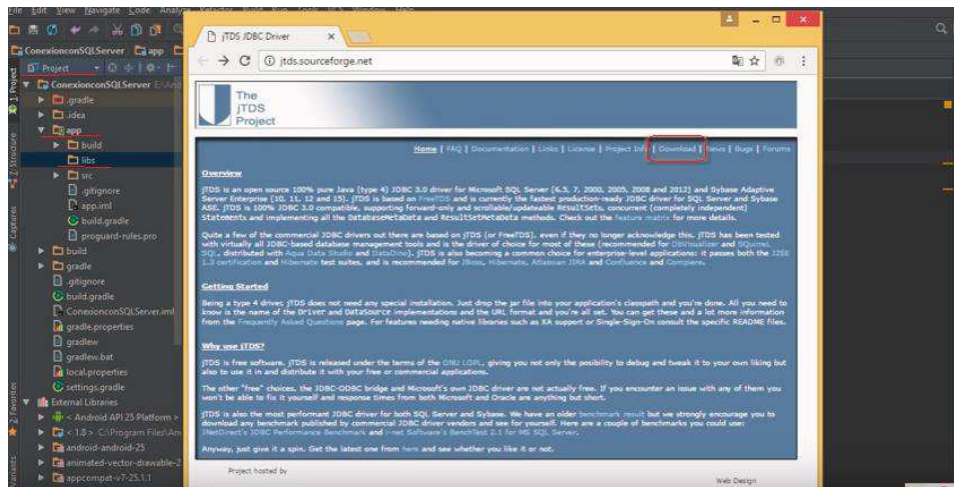


Figura 16. Configuración de carpetas de desarrollo (Labrador G. 2018)

En el Figura 17 se realiza la creación de la librería como dependencia y así creamos las librerías en desarrollo de Angular JS, para tener configuraciones distintas.

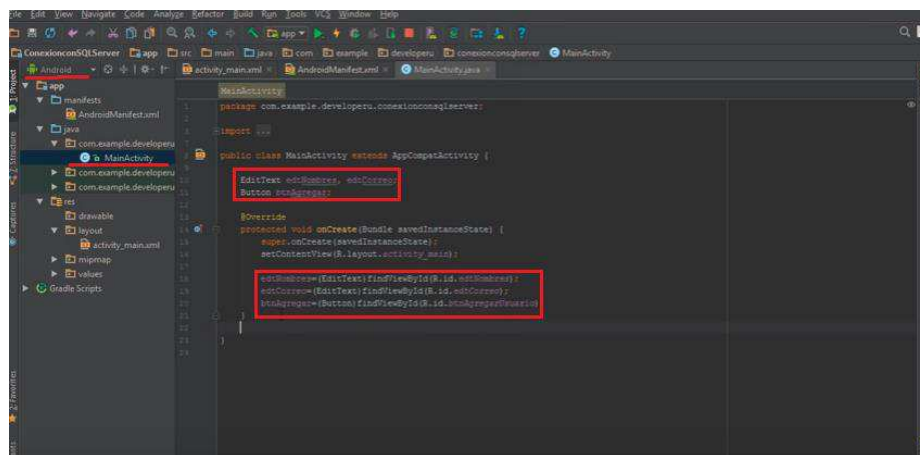


Figura 17. Configuración de librerías independientes. (Labrador G. 2018)

Nuevamente con el proyecto a modo Android, y como se muestra en la figura 18 que en la carpeta de MainActivity se declaran dos objetos EditText y otro de tipo EditBotton, y seguidamente se asocian los elementos de nuestro Layout con dichos objetos

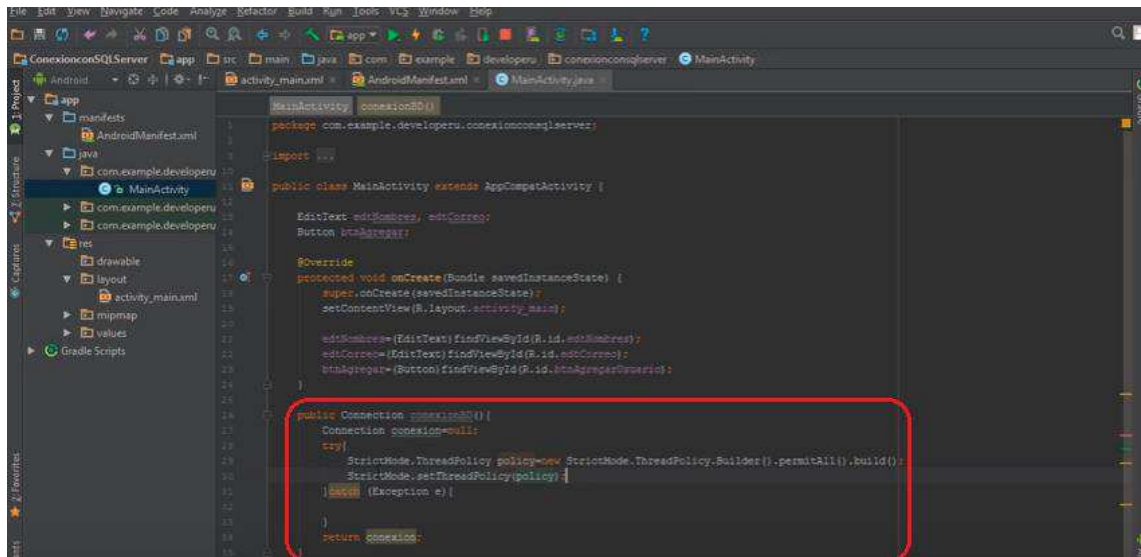


Figura 18. Configuración carpeta de MainActivity. (Labrador G. 2018)

Luego se crea la función para establecer la conexión con la base de datos que se quieren, y como primer paso agregamos el objeto de clase connection para crear la conexión con la base de datos, luego registramos el permiso de la aplicación mediante las líneas de código StrictMode. Este proceso se evidencia en la siguiente figura 19.

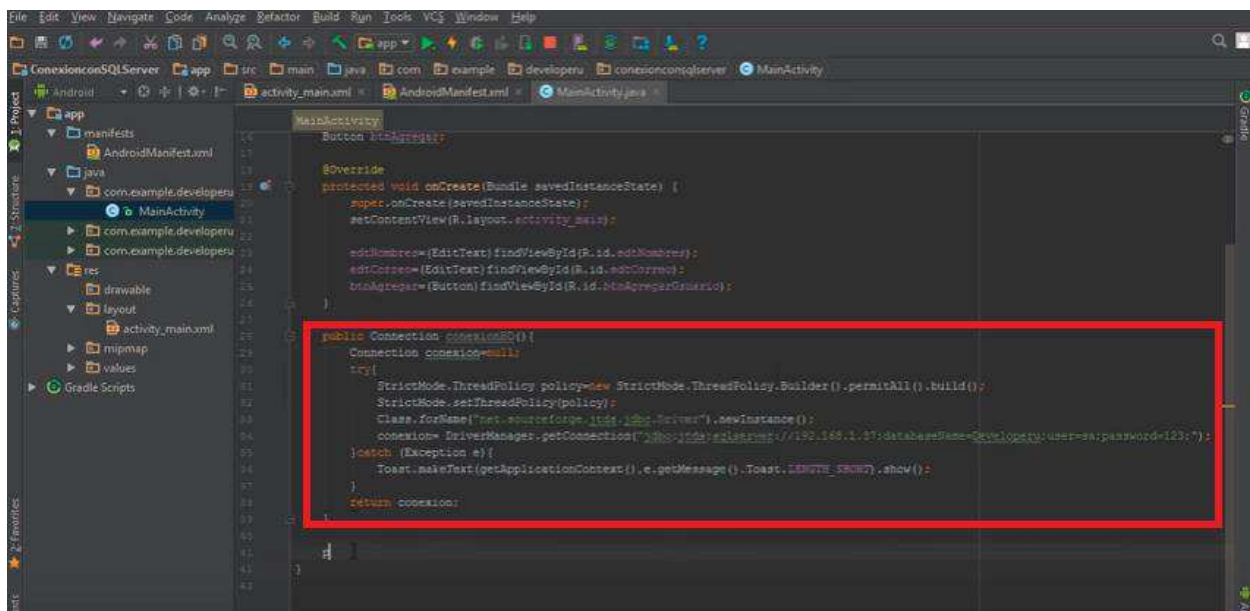


Figura 19. Configuración de Clase Connection. (Labrador G. 2018)

Ahora conectamos la aplicación con el SQL Server, por medio del class.forName de Java para declarar la conexión, luego con el objeto conexión y la clase DriverManager obtenemos la conexión a la base de datos y con los argumentos declaramos la dirección donde está la base de datos, en nombre de la base de datos, el usuario y su contraseña respectiva y luego hacemos una notificación “Toast”k para mostrar un mensaje posible de error.

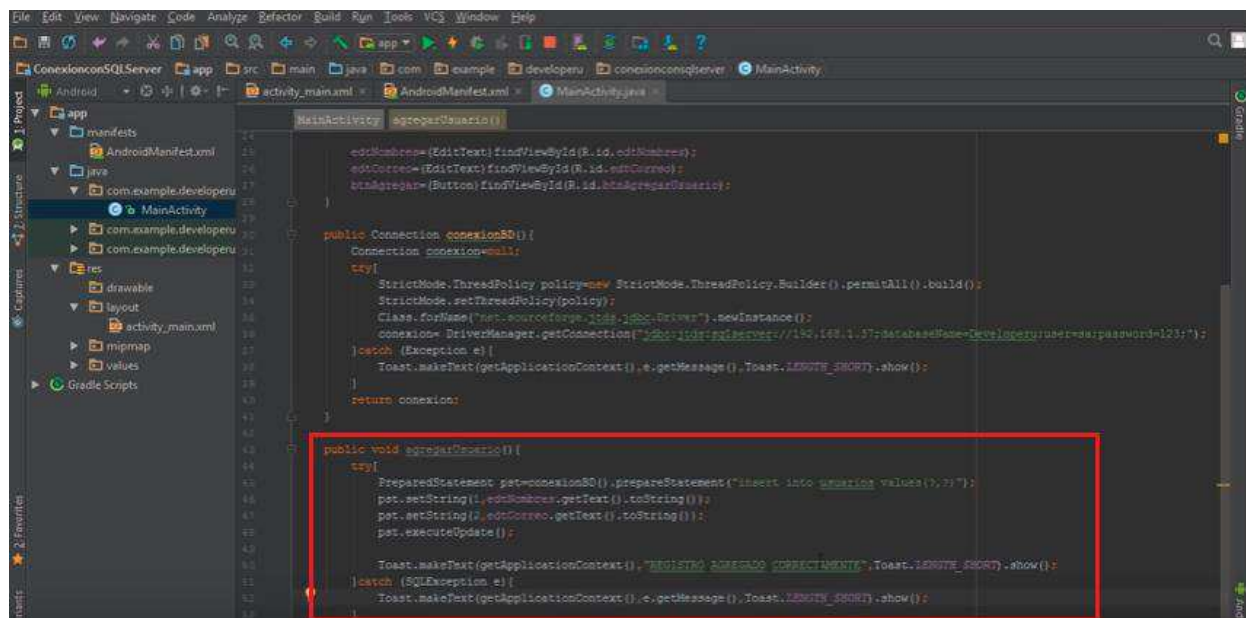


Figura 20. Configuración de Clase Connection. (Labrador G. 2018)

Después se agrega un método del cual se encargara de dar ingreso al usuario al agregarlo, haciendo uso de la clase “PreparedStatement” y nuestra función creada hacemos uso de una consulta en set con sus parámetros definidos y luego asignamos los valores a dichos parámetros en este caso vinculados al formulario y mediante la clase “pat.executeUpdate” ejecutamos la consulta creada. Y por último colocamos un “Toast” para dar mensaje de error o mensaje de confirmación.

Desarrollo front App

Creación de Front y Conexión con los servidores de las bases de Datos de Información Wifi.

En la figura 21. A la 25 se muestra en desarrollo del App desde su autenticación hasta la finalización de cierre de sesión.

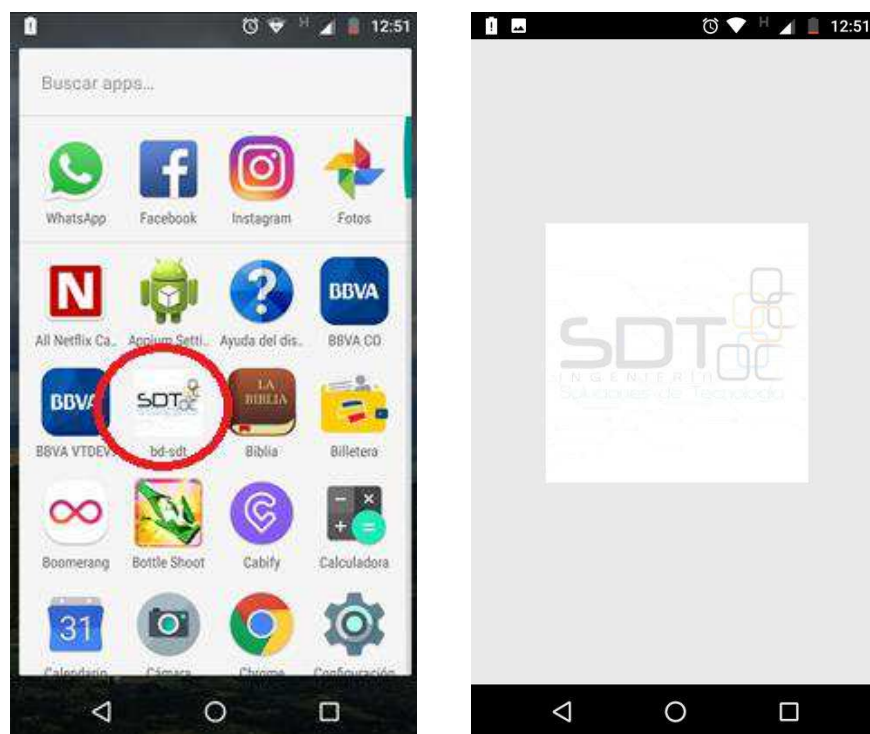


Figura 21. Autenticación a la App. (Labrador G. 2018)

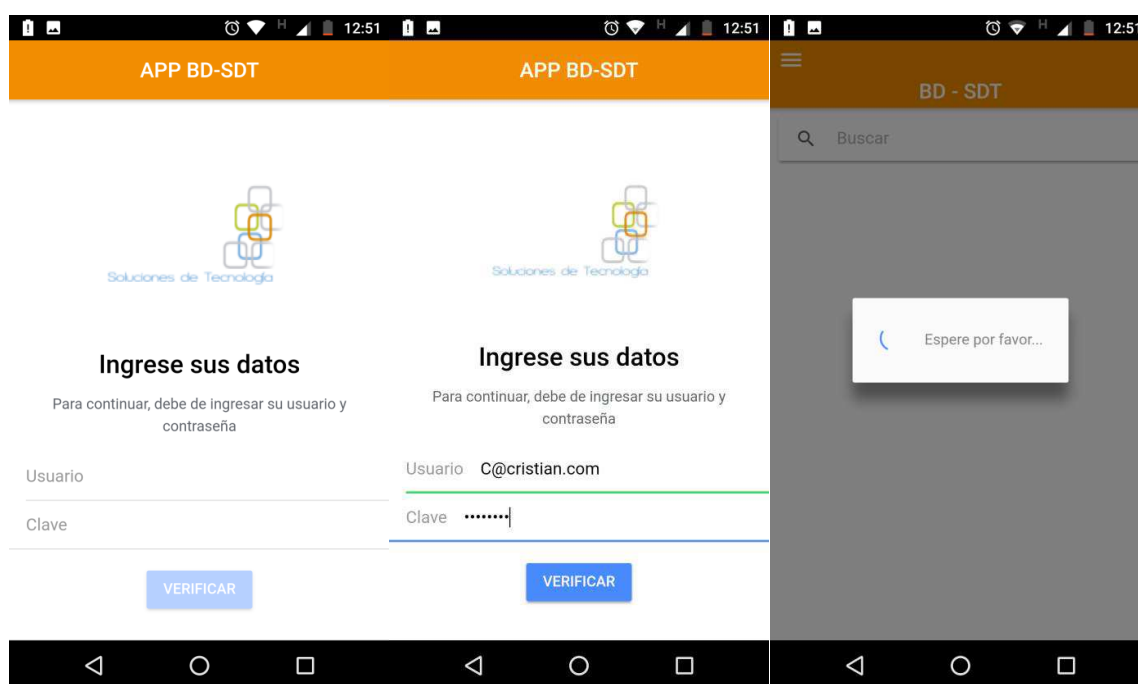


Figura 22. Actualizaciones y Cargue de Información dada por las bases de Datos solicitadas por SDT (Labrador G. 2018).

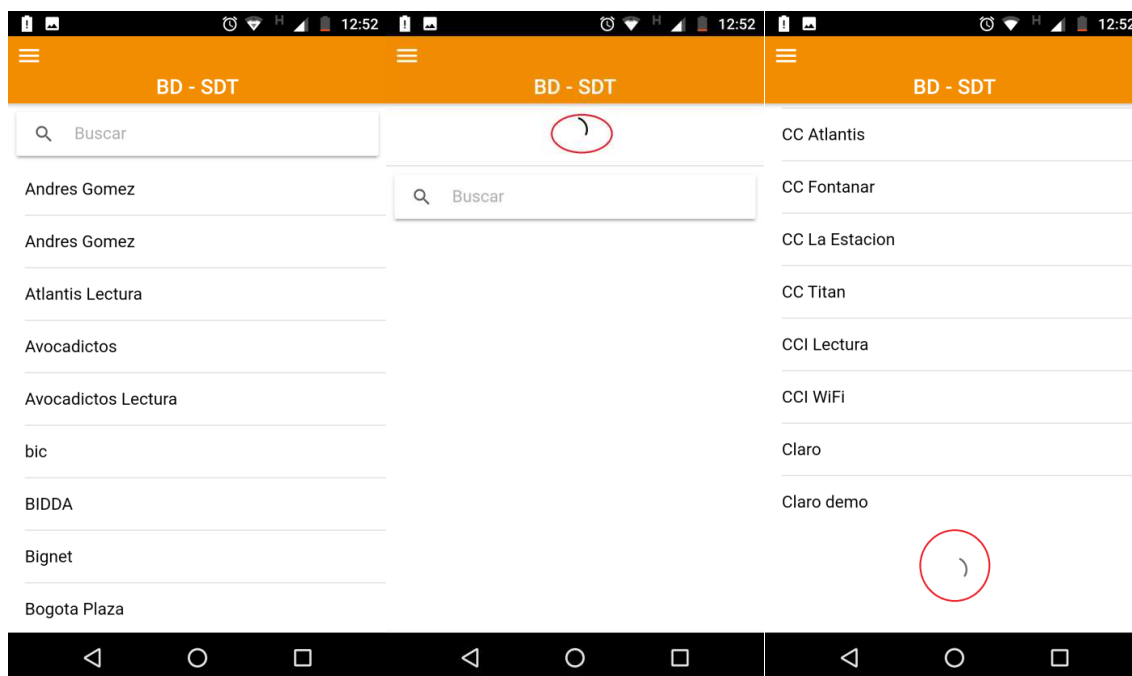


Figura 23. Listado Consulta de los Clientes SDT (Labrador G. 2018)

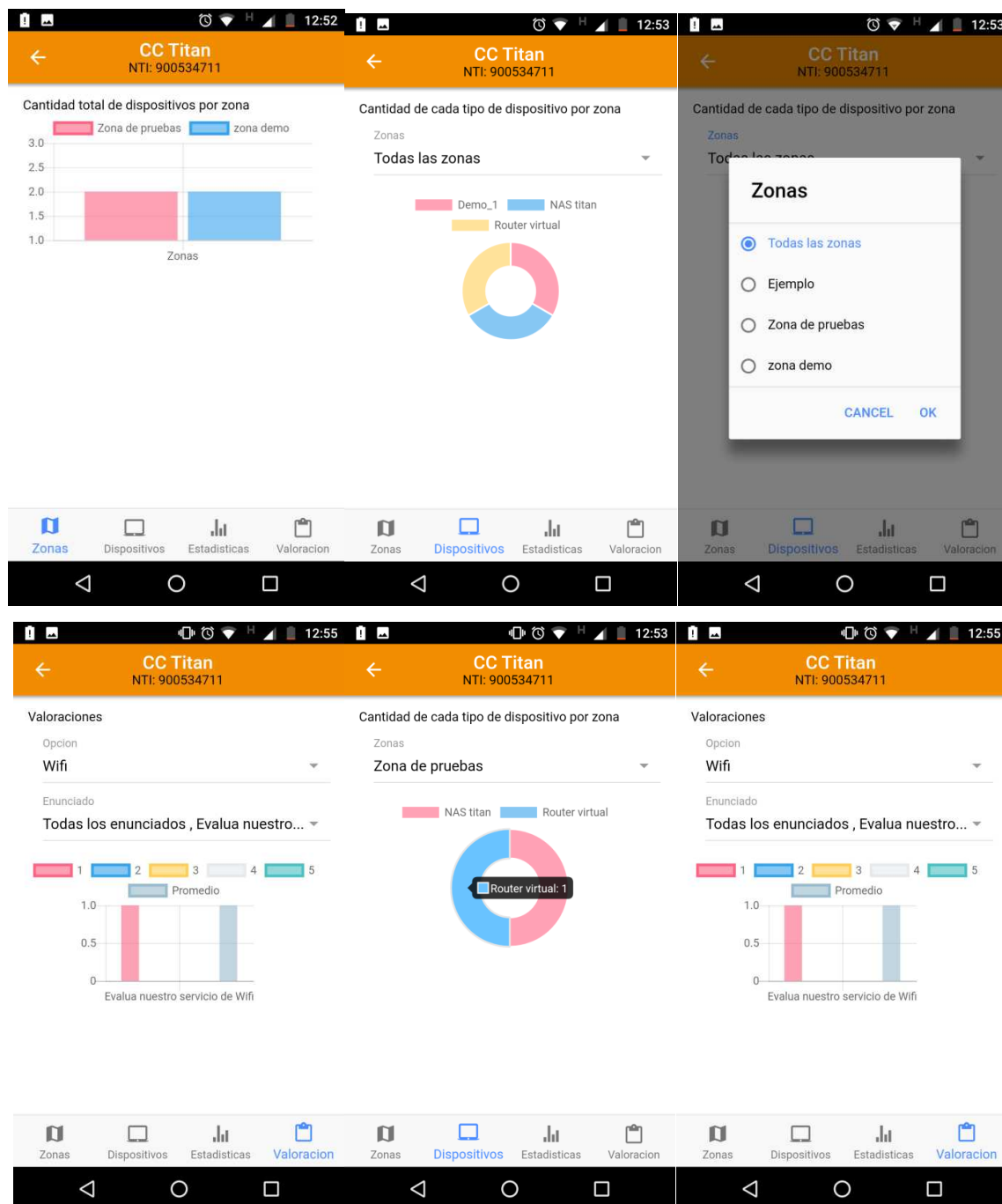


Figura 24. Graficas Analiticas de la lectura de Información Wifi (Labrador G. 2018)

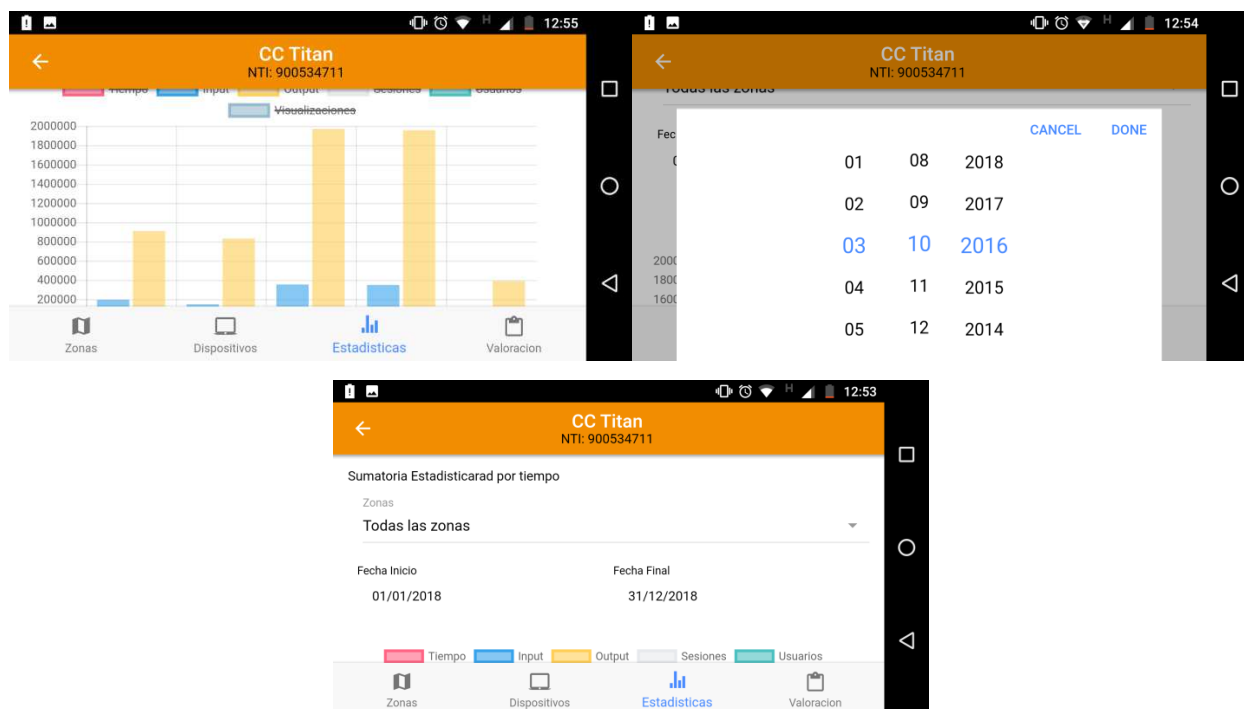


Figura 25. Graficas Analíticas de la lectura de Información Wifi (Labrador G. 2018)

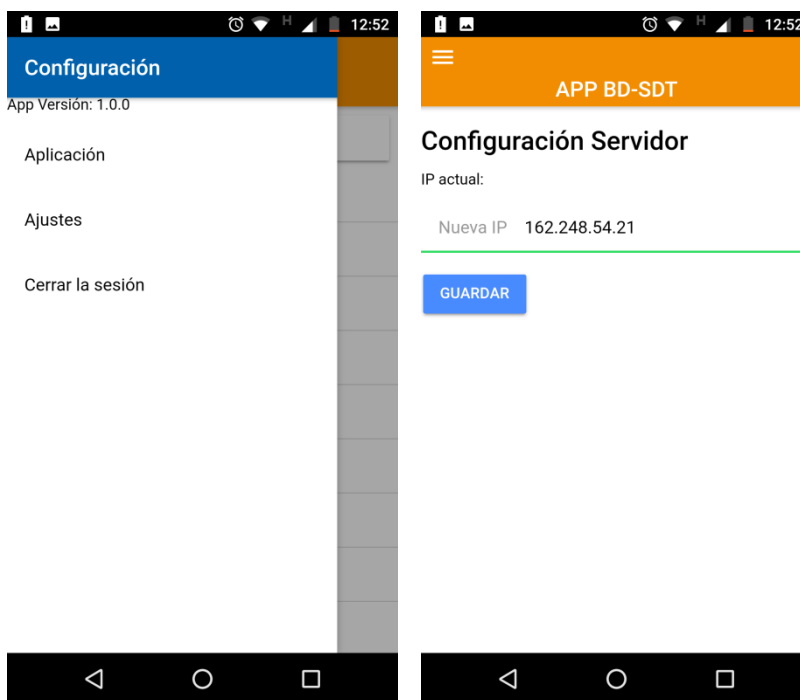


Figura 26. Configuración de Conexión de Servicio a las Bases solicitadas por el Usuario (Labrador G. 2018)

Plan de Pruebas para la App

Soportado sobre el personal capacitado, el método y las herramientas se tienen los tipos de pruebas a realizar y los atributos de calidad a verificar, los que podrá o no percibir el usuario final y para los cuales se abordarán dentro del alcance del servicio de pruebas para aplicaciones móviles, este alcance incluye:

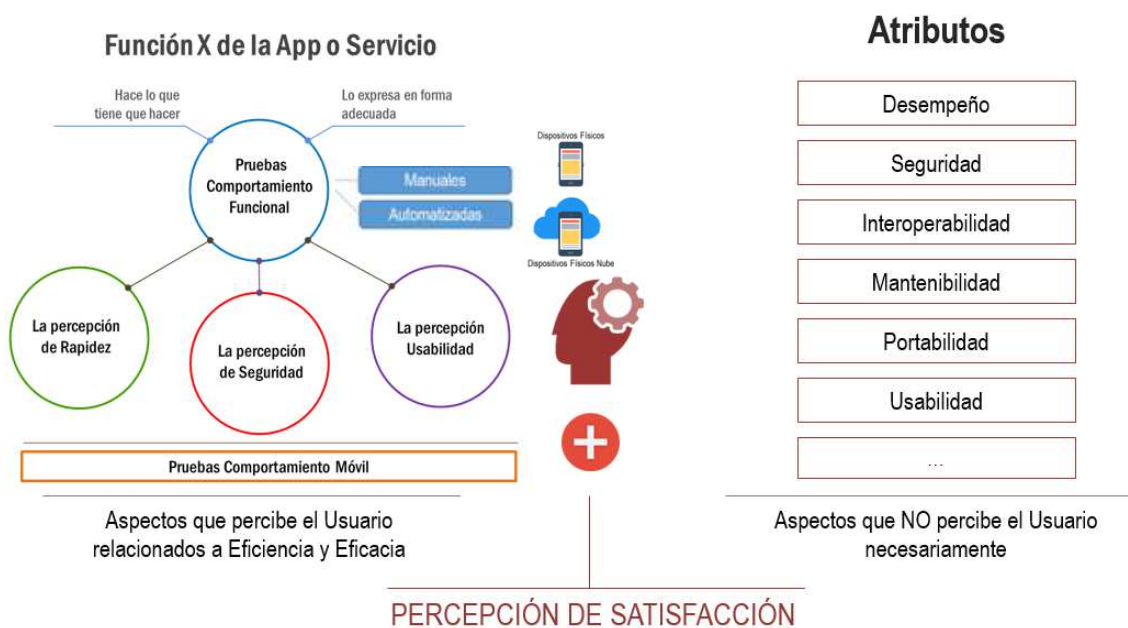


Figura 27. Enfoque de Pruebas Móviles (Choucair 2017)

Pruebas Comportamiento Funcional o Transaccional

Estas pruebas tienen como alcance realizar la ejecución de pruebas sobre las transacciones asociadas a la definición funcional de los requisitos, verificando lo que hace la funcionalidad y como lo expresa. Para el proyecto de Conexión a Bases de Datos se verificarán las funcionalidades.

Pruebas Percepción Seguridad y Performance

Estas pruebas están encaminadas a verificar las percepciones que podrá tener el usuario final sobre el performance de la aplicación y la seguridad que ésta muestra, teniendo en cuenta los

estándares de desarrollo.

Entre las verificaciones que se realizan se encuentran:

- Tiempo inicio de la App al ejecutarla en los dispositivos móviles y sistemas operativos
- Tiempo de carga entre pantallas y elementos de la aplicación móvil
- Información del consumo de batería, datos, CPU y memoria dados por los mismo dispositivos móviles y sistemas operativos
- Verificar enmascaramiento de los datos para ingreso de claves, número de pólizas y número de tarjetas o cuentas de medios de pago.
- Verificar autorización (enrolamiento) para cada tipo de usuario según las opciones disponibles.
- Verificar que no se esté instalando archivos corruptos en el dispositivo y que al desinstalar la aplicación se borre toda la información de la aplicación.
- Verificar que no se guarde información sensible en el dispositivo móvil.

Pruebas Comportamiento Móvil

Para las pruebas de comportamiento móvil se tienen definidas las verificaciones de las siguientes características o comportamientos, las cuales están ligadas netamente al comportamiento de la aplicación en un dispositivo físico y con un determinado sistema operativo.

Pruebas de Instalación y desinstalación:

En este tipo de pruebas se realiza la verificación del comportamiento de la aplicación al ser instalada, configurada y al ser desinstalada, permitiendo:

Revisar el comportamiento durante los diferentes métodos de instalación y desinstalación de la aplicación en el dispositivo.

- Verificar que al desinstalar la aplicación no quede información en el dispositivo.
- Comportamiento al mover la aplicación a una memoria externa y viceversa.
- El comportamiento de la aplicación al realizar cambios de la configuración inicial o predeterminada.

- Verificar características propias proporcionadas por la aplicación móvil, para actualización, instalación o configuración de funcionalidades.

Pruebas de Manejo de Datos

Con este tipo de pruebas se realizan verificaciones enfocadas a conocer el comportamiento de la aplicación respecto al manejo de datos para encontrar huecos de seguridad en el manejo inconsistente de estos, entre las validaciones a realizar se tienen:

- Verificar el almacenamiento de la información en la memoria interna del dispositivo por parte de la aplicación.
- Verificar el almacenamiento de la información en la memoria externa del dispositivo por parte de la aplicación.
- Verificar el acceso y manejo (consulta, modificación, eliminación) a la BD del dispositivo.

Pruebas de Manejo de eventos

Al realizar este tipo de pruebas se verifica el comportamiento de la aplicación cuando se presenten eventos que disparen ya sea la aplicación misma u otras aplicaciones, dentro de las verificaciones a realizar se tienen:

- Reanudar una aplicación luego de estar inactiva un tiempo determinado.
- Verificar que la aplicación se comporta correctamente al término de un evento de expiración mientras se está ejecutando.
- Verificar que la aplicación se inicializa al término de un evento de expiración.
- Verificar el comportamiento al generarse una notificación de otra aplicación.
- Verificar la percepción de calentamiento de la batería.

Pruebas de Influencias externas o Interrupciones

Al realizar este tipo de pruebas se pretende verificar el comportamiento de la aplicación, con respecto a las influencias que se puedan tener, al ser un dispositivo móvil, que está compuesto

por distintos componentes y elementos que lo hacen único, entre las pruebas a realizar se tienen.

- Verificar el comportamiento de la aplicación ante la entrada de mensajes.
- Verificar el comportamiento de la aplicación ante la entrada de llamadas.k
- Verificar que la aplicación funciona correctamente al realizar la inserción y remoción de tarjetas de memoria, utilización de puertos USB o utilización de otros puertos mientras está en ejecución.
- Verificar que la aplicación funciona correctamente al recibir un correo, mientras está en ejecución.
- Verificar que la aplicación funcione correctamente al estar conectado con audífonos y al tener conectado el cargador en el dispositivo.

Look and Feel (UI) y Usabilidad básica

Por medio de estas pruebas se realizan verificaciones respecto a la experiencia de usuario con la aplicación, realizando validaciones de:

- Verificar coherencia de menús.
- Verificar árbol de navegación.
- Verificar consistencia de fuentes y contrastes
- Verificar opciones de feedback.
- Verificar Legibilidad y Tiempo de Lectura
- Tamaño Elementos Aplicación
- Mensajes de Error, Técnicos y Confirmación
- Función de Cargar Aplicación
- Ortografía

Ejecución del plan de pruebas

Se realizan las pruebas en los siguientes equipos

Tabla 2

Investigación de mercado de dispositivos móviles.

| ID | Dispositivo | Tipo de Dispositivo | SO | Versión SO |
|----|-------------|---------------------|---------|------------|
| 1 | Motorola x | Smartphone | Android | 7.x |
| 2 | Samsung x | Smartphone | Android | 6.x |
| 3 | Huawei x | Smartphone | Android | 8.x |
| 4 | Polaroid x | Smartphone | Android | 6.x |

Nota: Tabla construida de investigaciones de usos en los dispositivos para las pruebas (Labrador González, 2018)

Se anexa algunos pantallazos con las evidencias de la ejecución de Pruebas de Batería, CPU, Consumo Funcionalidades, Estrés, Comportamiento Móvil, Interrupciones, Manejo de Eventos y de Datos, Usabilidad entre Otras.

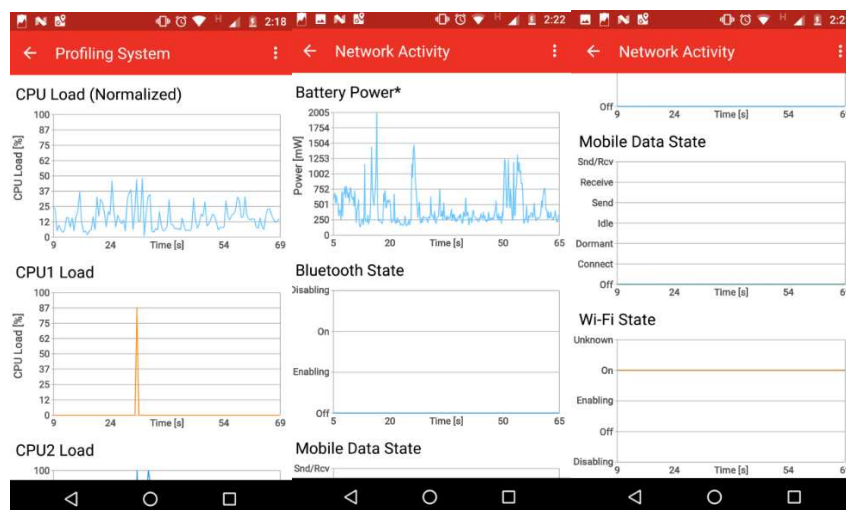


Figura 28. Pantallas de Pruebas de batería, cpu, red (Labrador G. 2018)

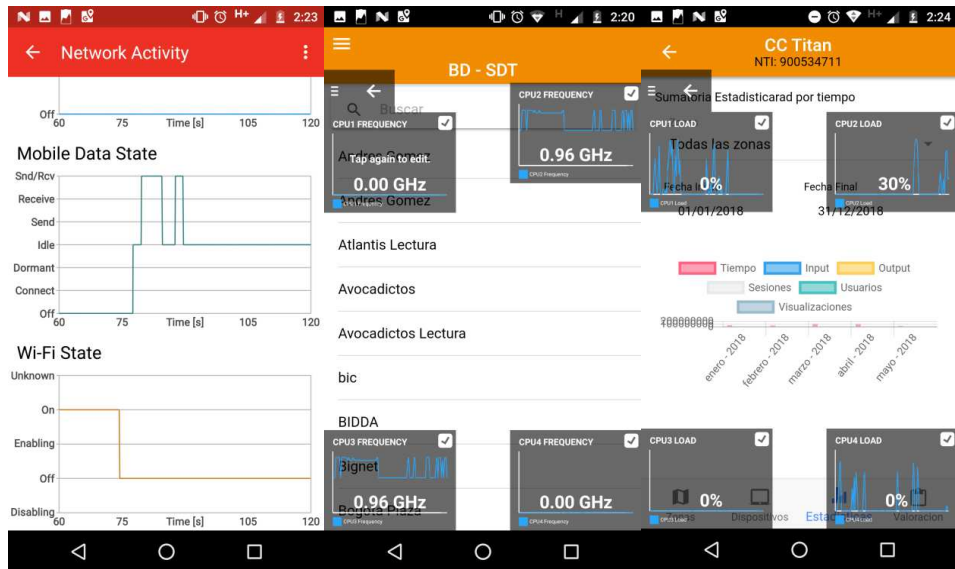


Figura 29. Pantalla de Pruebas de cpu, red (Labrador G. 2018)

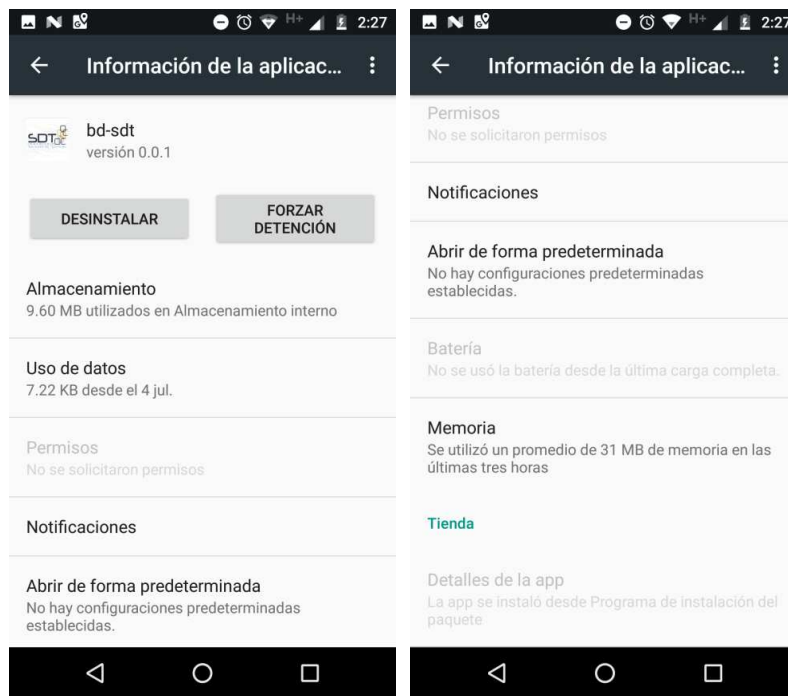


Figura 30. Pantalla de Pruebas en Memoria y Almacenamiento (Labrador G. 2018)

Conclusiones

Fue posible el desarrollo del App analítica con graficas estadísticas e informativas de uso de las redes inalámbricas implementadas por SDT de sus clientes, ya que por medio de esta se brindó soluciones efectivas de consultas de información por medio de tecnologías en sistemas operativos móviles.

También se logra implementar un servicio web (web Service) centralizado donde la nube de SDT permite la consulta dinámica de datos desde aplicaciones APP hasta las bases de datos MySQL.

La App tiene funciones de autenticación centralizada contra base de datos del servidor centralizado y pueda acceder a consultas remotas de datos. Por medio de la opción desarrollada en el direccionamiento de la Base de Datos a la cual quiera acceder el usuario. Aparte de la funcionalidad de la APP de funciones de gráficación analítica de datos estadísticos se logró tener funcionalidades asíncronas con los servicios. Ya que su desarrollo fue basado en una aplicación híbrida.

A partir de la investigación se determinó que el protocolo Database Cliente. Servidor es muy útil en aplicaciones móviles actuales, con la implementación que tiene el protocolo en un futuro en tecnologías, permitirá a las personas acceder a nuevos servicios y aplicaciones muy útiles en la vida cotidiana para consultas a bases de Datos SQL.

El sistema operativo Android es el más viable para desarrollar aplicaciones para dispositivos móviles por sus ventajas en costo, desarrollo y tiempo de aprendizaje. Utilizar el estándar de modelamiento Cliente-Servidor permitió reconocer con claridad los requerimientos y su interacción; lo cual permitió que el desarrollo de la aplicación se ajustara a las necesidades. El protocolo de comunicación en SQL Server es una excelente alternativa para la transmisión de datos gracias a sus ventajas en costo, efectividad y velocidad.

Gracias a que la tecnología que nos muestran brinda, se encuentra en una etapa de formación, es una gran oportunidad de implementar nuevas aplicaciones en las diferentes áreas que puedan mejorar el estilo de vida de las personas en su ámbito profesional.

Recomendaciones

Se recomienda hacer una actualización de la información a las bases de datos que tienen en la empresa SDT de Ingeniería.

Se recomienda solicitar el desarrollo para sistemas operativos ios, ya que se cuenta con el código en ionic el cual funciona para el funcionamiento en cualquier sistema operativo.

Referencias

Abdreu T, Regalado E. 2008. IRC cenco.sld.cu. URL: <http://www.cenco.sld.cu/node/38>

Acuña José. 2017. SO móviles. Wordpress.com. URL:
<http://jmacuna73.blogspot.com.co/2017/03/sistemas-operativos-moviles.html>

Aparicio D. 2011. Usuarios del mundo de MI. 20minutos.es URL:
<http://basesdedatosfecajaja.blogspot.com/2011/03/evolucion-historica.html>

CokePHP 2017. Desarrollo de PHP. URL: <https://book.cakephp.org/3.0/en/index.html>

General Redes. 2008. Peer to Peer. darkub.wordpress.com.
URL:<https://darkub.wordpress.com/2008/01/09/%C2%BFque-es-una-red-p2p-peer-to-peer/>

González C. 2018 novedades en S.O Móviles y Bases de Datos. URL:
<http://histinf.blogs.upv.es/2011/01/04/historia-de-las-bases-de-datos/>

Maquez G. 2015. Sistemas Operativos móviles. Arearecnologica.com. URL:
<http://www.areatecnologia.com/informatica/sistemas-operativos-moviles.html>

Microsoft Azura. SQL DataBase. MicrosoftAzura.com URL: <https://docs.microsoft.com/es-es/azure/sql-database/>

Perez J. 2015. Ionic FrameWork. URL: <http://www.phonegapSpain.com/que-es-y-como-empezar-con-ionic-framework/>

Symfony. 2014. Databases and the Doctrine ORM. URL:
<https://symfony.com/doc/current/doctrine.html>

Ionic. 2014. Project Structure URL: <https://ionicframework.com/docs/api/>

Informatica M. 2011. Bases de Datos. URL: <http://histinf.blogs.upv.es/2011/01/04/historia-de-las-bases-de-datos/>

Ricardo García. 2016. Análisis de bases de datos y tendencias tecnológicas URL:
https://rua.ua.es/dspace/bitstream/10045/57065/1/ANALISIS_DE_BASES_DE_DATOS_Y_TENDENCIAS_TECNOLOGI_GARCIA_CEBREIROS_RICARDO.pdf

Roam. 2017. Servidores Móviles. Aprenderaprogramar.com URL:
https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-sntp-dhcp&catid=57&Itemid=179

Tijerina Renny. 2016. Servidores. Emaze.com. URL:
<https://www.emaze.com/@AFCQQQFF>