



FACULTAD DE INGENIERÍAS Y CIENCIAS AGROPECUARIAS

IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN DE  
CLIENTES Y EL CONTROL DEL PERSONAL DE CAMPO EN LA EMPRESA  
XANDA LINKWORKS CÍA LTDA.

Trabajo de Titulación presentado en conformidad a los requisitos  
establecidos para optar por el título de Ingeniero en Electrónica y  
Redes de información

Profesor Guía  
Eduardo Fernando Solís Acosta  
M.Sc. en Auditoría de Sistemas Tecnológicos

Autor  
Gustavo David Hoyos Maroto

Año  
2015

### **DECLARACIÓN DEL PROFESOR GUÍA**

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

---

Eduardo Fernando Solís Acosta  
Magister en Auditoria de Sistemas Tecnológicos  
C.C.: 1803005071

### **DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE**

Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.

---

Gustavo David Hoyos Maroto

C.C.: 1717575623

## **AGRADECIMIENTOS**

Agradezco a todas las personas que estuvieron siempre a mi lado, apoyándome e incentivándome a cumplir con esta meta. Sé que el proceso fue largo y complicado; pero, después de todo, este es el resultado. Quiero agradecer a mi familia por su paciencia, apoyo y comprensión. Este logro es para todos ustedes.

## **DEDICATORIA**

Esta tesis la dedico a mi familia. Mi papá, mi mamá y mis hermanos y a mi novia, quienes estuvieron siempre pendientes y ayudándome en todo lo que necesité durante mi periodo de estudios universitarios. Ustedes son las personas más importantes y a quienes más quiero.

## RESUMEN

XANDA LINKWORKS Cía. Ltda es una empresa que presta servicios de cobranzas, tele mercadeo y actualizaciones en Ecuador. Tiene cuatro años en el mercado de cobranzas. Actualmente cuenta con un cliente en la provincia de Morona Santiago, en la ciudad de Macas. Se trata de Cable Express.

Cable Express es una empresa que provee servicio de televisión por cable a Macas. Para poder realizar el cobro del servicio brindado por Cable Express, el personal de campo de XANDA LINKWORKS Cia. Ltda se encarga de visitar a los abonados en su domicilio. Es decir, los gestores de Xanda Linkworks se encargan de recolectar y actualizar la información de los distintos clientes que se encuentran en las bases de datos de Cable Express.

Este personal realiza su tarea sin contar con la tecnología y medios adecuados, por lo que no existe una administración óptima de la información de los clientes, así como el control y evaluación del personal de la empresa.

Esta es la razón por la que se propone el desarrollo e implementación de un nuevo software que cuente con aplicaciones que optimicen el tiempo del personal y que regulen la gestión que se realiza con los procesos que involucran los clientes de manera remota.

La metodología de desarrollo que se empleó para el desarrollo de la aplicación fue RAD (rapid application development). El sistema generó un orden y control en la gestión de cobranzas en terreno y se concluyó que la aplicación ayuda a mejorar y entender el proceso de gestión de cobranzas

## ABSTRACT

Xanda Linkworks Cia. Ltda. is a company that provides collection services, telemarketing and updates in Ecuador. It has four years in the collections market. It, currently, has a client in Morona Santiago province, in the city of Macas. This client is Cable Express. Cable Express is a company that provides cable television service to Macas. To carry out the collection of service provided by Cable Express, the field staff of Xanda Linkworks Cia Ltda. visit the subscribers at their homes. Xanda Linkworks managers are responsible for collecting and updating information of different customers that are in the database Cable Express. This staff works without appropriate technology and media. This make that Xanda Linkworks doesn't have optima lmanagement of customer information. Moreover, the monitoring and evaluation of company personnel are not controlled. For this reason, it is necessary the development and implementation of new software with applications that optimize staff time and management the processes that involve remote clients. The development methodology that was used to develop the application was RAD (rapid application development), the system generated order and control in the management of collections and concluded that the application helps to improve the operation and better understand the management process of collections.

# ÍNDICE

INTRODUCCIÓN .....	1
1. MARCO TEÓRICO .....	8
1.1 Sistemas operativos para teléfonos móviles.....	8
1.2 PHP (hipertext Preprocessor).....	9
1.3 Conceptos Básico de Web Services.....	14
1.4 Conceptos básicos de Microsoft SQL SERVER .....	21
1.5 Conceptos básicos de Android.....	27
1.6 Sistema de Posicionamiento Global .....	33
1.7 Conceptos básicos de arquitectura basada en capas .....	34
1.8 Patrón de diseño Modelo – Vista- Controlador.....	36
1.8 Determinación de la metodología de desarrollo.....	37
1.8.1 Modelo RAD (Rapid Application Development).....	41
2. MODELO DE GESTIÓN.....	43
2.1 Estudio del Sistema existente de gestión de cobranza .....	43
2.1.1 Situación actual de la gestión de cobranza.....	43
2.1.2 Estudio del Sistema existente .....	46
2.1.3 Manejo de la información de gestión de cobranza .....	49
2.1.4 Destino de la información de gestión de cobranza.....	49
2.1.5 Procesamiento de la información de gestión de cobranza.....	50
3. MODELADO DEL PROCESO .....	54
3.1 Arquitectura de la aplicación.....	54
3.1.1 Capa de base de datos .....	54
3.1.2 Capa de Negocio .....	55
3.1.3 Capa de Presentación.....	61
4. MODELADO DE LOS DATOS .....	63
4.1 Datos de la solución.....	63
4.1.1 Abonado.....	63

4.1.2 Dirección .....	65
4.1.3 Teléfono .....	66
4.1.4 Contratos .....	66
4.1.5 Tipo de contrato .....	67
4.1.7 Historial de mora .....	68
4.1.6 Gestiones .....	69
4.1.7 Tipos de gestión.....	70
4.1.8 Usuarios .....	71
4.1.9 Tipos de Usuarios .....	72
4.1.10 Asignación.....	72
4.2 Diagrama relacional de la solución y tablas que la componen .....	80
<b>5. GENERACIÓN DE LA APLICACIÓN.....</b>	<b>82</b>
5.1 Capa del modelo .....	82
5.2 Capa del Controlador .....	86
5.3 Capa de la vista.....	91
5.4 Diagrama de Paquetes del prototipo .....	102
<b>6. PRUEBAS Y ENTREGA .....</b>	<b>104</b>
6.1 Pruebas de Funcionalidad.....	104
6.2 Comparación del proceso actual de cobranza con el proceso usando la aplicación.....	107
6.3 Entrega.....	110
<b>7. CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>111</b>
7.1 Conclusiones .....	111
7.2 Recomendaciones.....	113
<b>REFERENCIAS .....</b>	<b>114</b>
<b>ANEXOS .....</b>	<b>116</b>

## INTRODUCCIÓN

La globalización es un factor importante en la evolución tecnológica que actualmente estamos viviendo.

Cada día podemos observar la alta competitividad de los diferentes productos y servicios que existen en el mercado. En este sentido, el mundo empresarial y de negocios no se queda afuera.

Una empresa exige mucho tiempo, esfuerzo y recursos que si no son bien utilizados, pueden terminar siendo solo un gasto. Por esta razón, es necesario siempre estar al tanto de las exigencias y expectativas del mercado.

El éxito de una empresa se puede asegurar mediante el uso de técnicas y herramientas que permitan que la empresa brinde eficiencia, efectividad y seguridad en sus operaciones.

Tomando esto como antecedente, vamos a tomar como ejemplo a la Empresa XANDA LINKWORKS Cía. Ltda.

XANDA LINKWORKS es una empresa multinacional, dedicada a prestar servicios de cobranzas, tele mercadeo y actualizaciones. Tiene cuatro años en el mercado de cobranzas. Cuenta con clientes como el Banco de Guayaquil y el Banco del Pichincha.

Debido a su gran crecimiento, XANDA LINKWORKS busca establecerse como una de las mejores empresas de cobranza masiva del Ecuador. El personal de XANDA LINKWORKS Cía. Ltda. se divide en dos áreas. A estas se las divide en personal de planta y personal de campo.

XANDA LINKWORKS Cía. Ltda. ha tenido excelentes resultados. Se trata de una empresa que ha incrementado su facturación, por lo que su personal también ha aumentado en los últimos meses. Cabe destacar que este

crecimiento ha generado inconvenientes en su operación que buscan ser resueltos para dar más eficiencia a la operación y evitar el desperdicio de recursos.

La empresa busca mejorar el proceso de cobranza que su personal de gestión de campo realiza. Ya que por el hecho de que el trabajo sea en la calle, se vuelve complicado el control de las actividades que ejerce este personal.

Los gestores de campo de XANDA LINKWORKS Cía. Ltda. se encarga de realizar la cobranza en el campo. Además, recolectan y actualizar la información de los distintos clientes que se encuentran en las bases de datos que se les entrega para su gestión.

Hoy en día, los resultados del personal de campo de XANDA LINKWORKS Cia. no son medibles ya que, no se tiene un control completo de las actividades que estas personas realizan. A pesar de que se tenga un control de ruta, nada garantiza que el personal esté cumpliendo con sus responsabilidades y esté dedicando su tiempo completamente al trabajo.

Una opción para poder solucionar las falencias que tiene el personal de campo es desarrollar una solución tecnológica que busque optimizar su trabajo.

El trabajo de campo, como se mencionó anteriormente, consiste en recolectar y actualizar la información de los clientes que se encuentran en la base de datos que el personal de cobranza debe gestionar. Pero esta tarea se realiza de manera empírica, debido a que no disponen de una herramienta que les permita una administración óptima de la información de los clientes, así como el control y evaluación del personal de la empresa.

Por esta razón es importante realizar el desarrollo e implementación de un sistema informático que permita mejorar las falencias que hasta el momento ha venido presentando la empresa.

Se propone que la solución informática busque optimizar el tiempo de la gestión del personal y que regulen los procesos que involucra el trabajo de campo.

Con todo esto se espera que al poner en funcionamiento este software haya una mejor productividad y control en el manejo del proceso de cobranza en el terreno.

Los créditos en el país han tenido un gran progreso. Años anteriores los requerimientos que pedían las instituciones financieras eran complicados de cumplir.

Hoy en día, las personas pueden acudir a un local comercial y obtener un crédito directo tan sólo con presentar su cédula de identidad. Esto ha provocado que la tasa de créditos haya crecido, de acuerdo con un informe sobre la evolución del crédito y tasas de interés efectivas referenciales realizado por el Banco Central del Ecuador en el 2012.

En enero del 2012 hubo 558,913 transacciones con un monto de 2,605.4 dólares. Del mismo modo, las mutualistas registraron 5,7171 operaciones con un monto promedio de USD 4,003.1 dólares. Por su parte, los bancos registraron 382,891 operaciones, con una variación positiva de 3.99% (USD 3,065.8 dólares) (BCE, 2012).

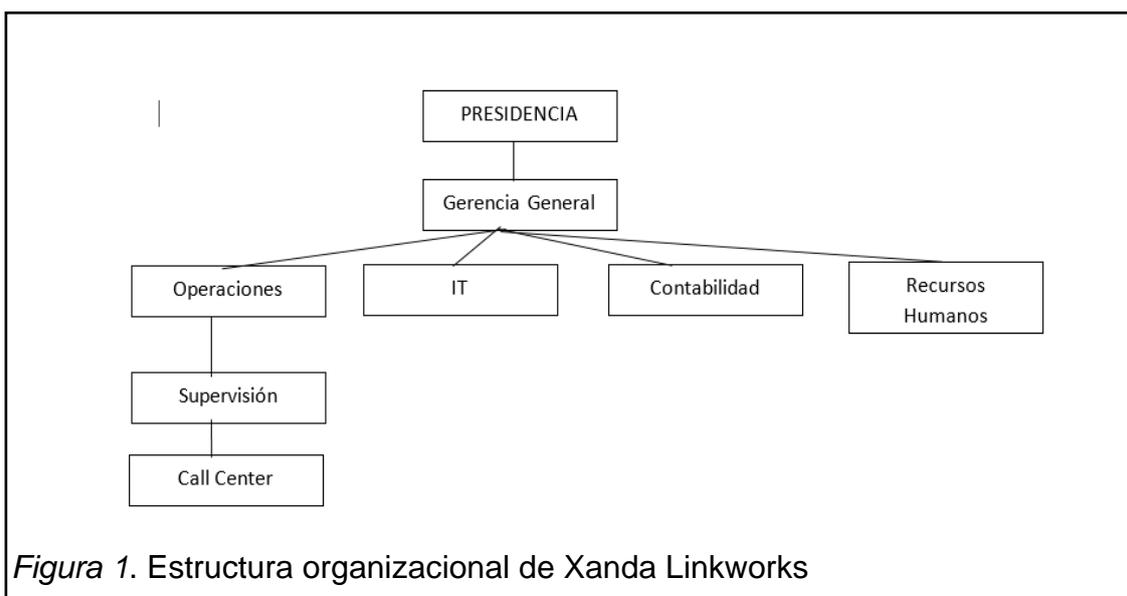
Asimismo, Los créditos en el Ecuador a partir del año 2009 tienen una transformación positiva, especialmente en la banca privada y pública, así como en las cooperativas (BCE, 2012).

Al existir una evolución en la actividad crediticia, el nivel de cartera es más alto y, por ende, el número de cartera por vencer también va a subir. Por esta razón, es necesaria la creación de empresas que se dediquen a recuperar cartera.

Debido a esta necesidad del mercado nace XANDA LINKWORKS Cía. Ltda. El objetivo de la empresa es ofrecer un servicio integral de recaudación y cobranzas a instituciones con políticas de reingeniería y que necesitan externalizar las labores administrativas inherentes a su giro comercial.

Esto permite que las empresas que contratan los servicios de XANDA LINKWORKS Cía. Ltda. disminuyan sus costos operacionales y logren invertir este dinero en tareas comerciales y/o de producción, generando así un valor agregado importante a la gestión de la empresa.

En la figura 1 se detalla se diagrama organizacional de la empresa Xanda Linkworks.



*Figura 1. Estructura organizacional de Xanda Linkworks*

### **Alcance**

Este proyecto comprenderá el diseño de un prototipo de una aplicación móvil que permite gestionar la cobranza y actualización de datos de clientes, realizada en el campo por parte de los gestores terrenos de la empresa Xanda Linkworks.

El diseño del prototipo de la aplicación se realizará en base a todos los requerimientos presentados por la empresa. Es decir, incluirá la recopilación de

la base teórica fundamental, levantamiento de los procesos y creación de una aplicación móvil basada en Android para su implementación en un entorno de pruebas.

El prototipo de la aplicación móvil permitirá la ejecución de la gestión en el campo, la cual comprende la consulta e ingreso de la información de los clientes a la base de datos. Para esto se aplicará una definición de parámetros de validación que permita que los datos sean registrados correctamente. De esta manera se espera que el personal operativo y de terreno pueda registrar la información del cliente de manera remota. Una vez ingresados los datos, tales como: fechas, números de contacto, entre otros, estos serán validados.

El sistema manejará la información de los clientes, la confirmación de las visitas, la geo-referenciación, la evidencia fotográfica y el control laboral y de eficiencia del personal de campo.

El diseño del prototipo de la aplicación usará una estructura de red y software que usará las siguientes herramientas:

- Web Service – PHP
- NAT
- Sistema operativo para dispositivos móviles: Android

Una vez finalizado el diseño, el prototipo se someterá a pruebas para verificar el desempeño de la misma. Xanda Linkworks analizará los resultados del prototipo para tomar decisiones sobre el uso del mismo en un ambiente de producción

### **Justificación**

XANDA LINKWORKS Cía. Ltda. desea controlar a su personal de campo por lo cual busca una solución que le permita realizar esta gestión. Uno de los problemas que actualmente se presentan en la empresa es la falta de

seguimiento del personal de terreno. El gestor de campo tiene su hoja con las visitas del día.

Después de registrar su gestión en la hoja de visita, ellos deben regresar a la empresa e ingresar todos los clientes visitados al sistema que actualmente maneja la empresa. Sin embargo, debido a que este proceso no se encuentra estandarizado no se puede saber si los gestores realizaron o no su trabajo. Solo se puede confiar en su palabra. Muchas veces se ha comprobado que las notificaciones no llegan a los clientes.

La implementación de una aplicación móvil basada en el sistema operativo Android, busca solventar los problemas de gestión y control del personal de campo. Esta aplicación busca mejorar la gestión de terreno para que los recursos usados por la empresa para esta actividad se optimicen

## **OBJETIVOS**

### **Objetivo General**

Implementar una aplicación de gestión de cobranza para dispositivos móviles que disminuya el tiempo y optimice recursos dirigidos al personal de campo de la empresa XANDA LINKWORKS Cía. Ltda.

### **Objetivos específicos**

- Analizar el requerimiento de la empresa XANDA LINKWORKS Cia. Ltda. respecto a la necesidad de un software móvil para la gestión de sus clientes en el terreno.
- Desarrollar un prototipo de una aplicación móvil basada en Android para la gestión de campo de los clientes de la empresa XANDA LINKWORKS Cía. Ltda.
- Implementar la aplicación móvil basada en Android en un entorno de pruebas en la empresa XANDA LINKWORKS Cía. Ltda.

- Probar la aplicación móvil basada en Android en un entorno de pruebas en la empresa XANDA LINKWORKS Cía. Ltda.

**Metodología a utilizar**

Se usará el método inductivo para el desarrollo del software tanto en dispositivo móvil como en el web Service. Asimismo, se empleará el método experimental para probar la aplicación móvil en parámetros de funcionamiento.

## MARCO TEÓRICO

### 1.1 Sistemas operativos para teléfonos móviles

El gran desarrollo tecnológico de los últimos tiempos ha hecho que los sistemas para equipos móviles sean importantes para los usuarios. Las empresas de desarrollo de aplicaciones en encontraron en esto un importante nicho de mercado (informatica-hoy, 2014).

Los avances tecnológicos han hecho que todas las personas se sientan identificadas con cada programa, equipo, material, entre otros que se desarrolla. Esto se ha dado porque cada avance mejora la calidad de vida del usuario (informatica-hoy, 2014).

Componentes de los sistemas operativos en dispositivos Móviles:

- **Capas**

Un sistema Operativo está formado por capas, al igual que los componentes informáticos más grandes.

- **Kernel**

El Kernel también llamado núcleo es la parte del software que permite ingresar al hardware del equipo (informatica-hoy, 2014). El núcleo trabaja con las capas así como con “los controladores, la administración de los distintos procesos, almacenamiento de datos, y el control de la memoria del sistema (informatica-hoy, 2014).

- **Middleware**

Se basa en un grupo de módulos que hace que las aplicaciones diseñadas puedan correr sin ningún tipo de problema (informatica-hoy, 2014).

El usuario puede ver cómo funciona el middleware. Este permite ejecutar servicios que agilitan el desarrollo de aplicación evitando la repetición de código.

Entre las funciones que tiene esta capa se encuentran:

- Mensajes y motores de comunicación.
- Seguridad.
- Personalización del equipo.
- Códec multimedia.
- Intérpretes de páginas Web y servicios WAP

- **Entorno de ejecución de aplicaciones**

Esta es la capa que suministra los elementos necesarios para la creación y desarrollo de software. Es decir, esta es la capa que tiene los elementos que se requieren para crear aplicaciones relacionadas con el sistema operativo (informatica-hoy, 2014).

- **Interfaz de usuario**

Se emplea para interactuar con el equipo. Está formado por: botones, pantallas, menú, etc. Además, tiene los elementos que se necesitan para facilitar la experiencia del usuario (informatica-hoy, 2014). La pantalla táctil es una de las mejores que ofrece este sistema. Sin duda, el poder tocar ha hecho que el mercado de los teléfonos móviles se revolucione.

## 1.2 PHP (hipertext Preprocessor)

Se trata de un lenguaje de código abierto utilizado para el desarrollo web, que puede ser incrustado en HTML (informatica-hoy, 2014). Hoy en día, se ha vuelto muy común ver más y más portales web contruidos con PHP.

Ejemplo:

"Tomado del manual de PHP en línea (PHP.ORG, s.f.)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
```

```
<head>
```

```
<title>Ejemplo</title>
```

```
</head>
<body>

    <?php
        echo "¡Hola, soy un script de PHP!";
    ?>
</body>
</html>
```

- En la primera línea es una etiqueta relativa a la versión/variante de HTML que declaramos usar, las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!")
- El código de PHP está encerrado entre las etiquetas especiales de comienzo y final <?php y ?> que permiten entrar y salir del "modo PHP".
- Las etiquetas de apertura y cierre de html, delimitan el código HTML.
- Las etiquetas de apertura y cierre head, delimitan la cabecera del código HTML.
- Las etiquetas body, delimitan el cuerpo del código HTML y que es donde suele ir el contenido central de un documento HTML (texto, imágenes, formularios, etc.).
- Dentro de las etiquetas body, además de código HTML, podemos tener código PHP embebido.

PHP se diferencia de Java Script por que el código corre en el servidor que lo procesa, reúne los datos y el servidor genera una página HTML.

El usuario ejecutara el script aunque no sabrá el código subyacente. El servidor web puede ser incluso configurado para procesar los ficheros HTML con PHP (Achour, 2014).

Este tipo de lenguaje suele ser utilizado para crear contenido dinámico y poder interactuar con el usuario; la información generalmente está contenida en una base de datos, lo mejor de usar PHP es que es extremadamente fácil de utilizar para el principiante, además permite a la mayoría de los programadores crear aplicaciones complejas e involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

### **Habilidades de PHP**

Está orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos:

- Puede procesar la información de formularios
- Generar páginas con contenidos dinámicos
- Enviar y recibir cookies.

PHP puede hacer cualquier cosa que pueda hacer otro programa CGI, ya que está enfocado a la programación de scripts del lado del servidor (Achour, 2014).

Existen tres campos principales donde se usan scripts de PHP.

- Scripts del lado del servidor.

Es el foco principal. Se necesitan tres cosas para que esto funcione:

- Analizador de PHP (módulo CGI o servidor),
- Servidor web
- Navegador web.

Se necesita tener un servidor, con una instalación de PHP instalada. Se puede acceder al programa PHP con un navegador. (Achour, 2014).

- Scripts desde la línea de comandos.

Se puede crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Únicamente se necesita el analizador de PHP para utilizarlo de esta manera. Es ideal para scripts ejecutados regularmente usando cron (en \*nix o Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden usarse para tareas simples de procesamiento de texto (Achour, 2014).

- Escribir aplicaciones de escritorio.

PHP tal vez no sea el lenguaje más apropiado para crear aplicaciones de escritorio con una interfaz gráfica de usuario, pero si se quisiera utilizar algunas características avanzadas de PHP en aplicaciones del lado del cliente, se puede utilizar PHP-GTK para escribir dichos programas. También es posible de esta manera escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal (Achour, 2014).

El programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable.

PHP puede usarse en todos los principales sistemas operativos, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS; es posible crear aplicaciones con una interfaz gráfica para el usuario, utilizando alguna extensión como PHP-Qt, PHP-GTK, WxPHP, WinBinder, Roadsend PHP, Phalanger, Phc o HiP Hop VM. También puede ser usado desde la línea de comandos, de la misma manera como Perl o Python pueden hacerlo; a esta versión de PHP se la llama PHP-CLI; PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Esto incluye cualquier servidor web que pueda utilizar el binario de PHP FastCGI, como lighttpd y nginx. PHP funciona tanto como módulo como procesador de CGI (Achour, 2014).

Algunas de las capacidades de PHP son:

- La creación de imágenes,
- Ficheros PDF e incluso películas Flash (usando libswf y Ming) generadas sobre la marcha.
- Genera fácilmente cualquier tipo de texto, como XHTML y cualquier otro tipo de fichero XML.

PHP puede autogenerar estos ficheros y guardarlos en el sistema de ficheros en vez de imprimirlos en pantalla, creando una caché en el lado del servidor para contenido dinámico (Achour, 2014).

PHP tiene un amplio abanico de conexión a bases de datos. Al usar extensiones específicas o utilizar una capa de abstracción como PDO, o por medio de la extensión ODBC. Escribir una página web con acceso a una base de datos es increíblemente fácil

- PHP cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros.
- Puede crear sockets de red puros e interactuar usando cualquier otro protocolo.
- PHP tiene soporte para el intercambio de datos complejos de WDDX entre virtualmente todos los lenguajes de programación web.
- PHP posee soporte para la instalación de objetos Java y usarlos de forma transparente como objetos de PHP.
- HP no requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).

Algunas estadísticas indican que alrededor del 60% de los sitios Web en Internet utilizan PHP; es un lenguaje de programación que goza de gran popularidad y difusión, a esta circunstancia hay que añadir el hecho de que páginas web de mucha importancia como Wikipedia o Yahoo se apoyan en este lenguaje, lo cual demuestra que es un lenguaje muy potente que puede servir desde para pequeñas páginas web hasta para grandes portales (Achour, 2014).

### **1.3 Conceptos Básico de Web Services**

Al final de la década de 1990, la WWW (denominada en lo sucesivo la Web) se propaga de forma explosiva en todo el mundo, ya que la banda ancha estaba al alcance del hogar, el negocio y de las empresas. Por esta razón la Web se convierte en la infraestructura esencial en la vida familiar y empresarial. En el usuario, hay un deseo de obtener la información de manera instantánea y de diferentes partes del mundo. (Nava, 2014).

La Web tradicional se basa en la suposición de que la gente va a utilizar el cliente Web mediante el protocolo http. Los datos proporcionados por el servidor estarán representados en formato html. El Navegador Web es una aplicación de cliente que analiza el formato html, y basándose en la información de estilo, coloca los datos en la pantalla.

Los datos en pantalla de son de tipo html y se pueden asociar con información de estilo, pero no se puede asociar con información semántica de los datos en sí.

Un servicio web es cualquier sistema de software que permite la interacción máquina a máquina sobre una red. De acuerdo con esto la interacción permite generar peticiones que tengan sentido dentro del el contexto de la información que se desea obtener.

## **Modelo de web services**

La arquitectura básica del modelo de web services describe a un:

- consumidor
- un proveedor
- ocasionalmente un corredor

Los tres actores pueden realizar las operaciones de publicar, encontrar y enlazar.

El funcionamiento básico consiste en:

1. El proveedor publica sus servicios a un corredor.
2. El consumidor se conecta al corredor para encontrar los servicios que necesita.
3. Una vez conectado el corredor realiza un lazo entre el consumidor y el proveedor.

Cada miembro puede ejecutar alguno o todos los roles.

Los web Services están montados con varias tecnologías que trabajan con los estándares que están surgiendo para garantizar la seguridad y operatividad, de modo de lograr que el uso combinado de varios Web Services, independientemente de la o las empresas que los proveen, esté siempre asegurado.

### **Requerimientos de los Web Services:**

- Una forma estándar de representación de datos

XML es la opción obvia para este requerimiento.

JSON es una opción que cada vez tiene más adeptos por su fácil esquema y semántica clara.

- Un formato común mensajes.

SOAP (Simple Object Access Protocol).

## REST (Representational State Transfer)

- Un lenguaje común para describir los servicios.  
WSDL. Es un lenguaje basado en XML desarrollado en forma conjunta por IBM y Microsoft.

RESTFUL. Es una arquitectura que usa xml o json y http de una manera más simple, sin complejidad adicional en los patrones de intercambio de información.

- Una forma de descubrir los Web Services.

UDDI (Universal Description, Discovery and Integration) es un catálogo de negocios de internet que especifica un mecanismo para publicar y localizar los servicios por parte de los proveedores y consumidores respectivamente como se muestra en la figura 2.

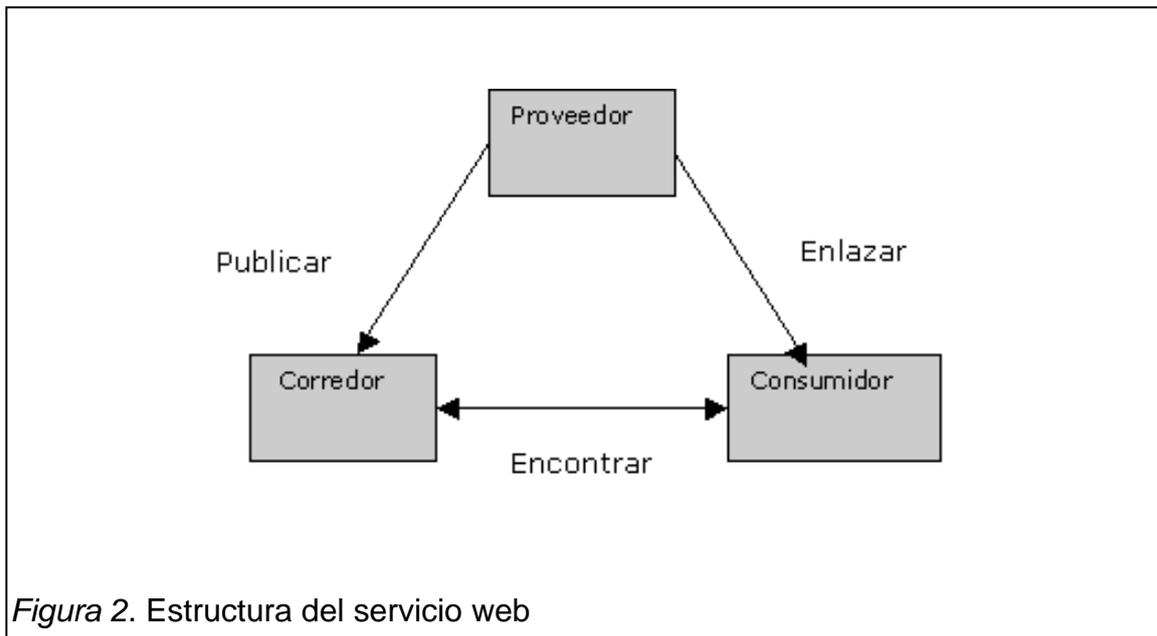


Figura 2. Estructura del servicio web

Los web Services comienzan a ser la pieza principal de la nueva generación de sistemas distribuidos. Por esta razón se analizará por qué son tan importantes en la actualidad.

**Interoperabilidad**

Cualquier web service puede interactuar con otro. Ya que pueden ser implementados en cualquier lenguaje, los desarrolladores no necesitan cambiar sus ambientes de desarrollo para producir o consumir los web services.

**Ubicuidad**

Los Web Services se comunican utilizando HTTP, XML o JSON. Cualquier dispositivo que soporte estas tecnologías pueden acceder o implementar web Services.

**Encapsular reduce la complejidad**

Todos los componentes en un modelo de Web Services son servicios web. Lo importante es la interface que el servicio suministra y no como está implementado, de esta manera se hace más simple la comunicación a través del servicio web reduciendo la complejidad.

**Fácil de utilizar:**

El concepto detrás de los Web Services es fácil de entender, Existen frameworks que permiten a los desarrolladores programar Web Services de forma rápida y fácil.

**Soporte de la Industria:**

Todas las empresas de software importantes soportan SOAP y REST.

**• Retos de los Web Services:**

Los Web Services debes superar algunos retos que les permitan ser estructuras simples y claras a la hora de comunicar datos. Los siguientes puntos analizaran estos retos.

**• Descubrimiento:**

¿Cómo un Web Services se anuncia para ser descubierto?

¿Qué pasa si el servicio sufrió algún cambio luego de ser anunciado?

WSDL, REST y UDDI son dos nuevos estándares que manejan este punto.

- **Confiabilidad:**

Algunos Web Services pueden ser más confiables que otros.

¿Cómo puede ser medida esa confiabilidad y como se comunica esta medición?

¿Qué pasa cuando un Web Services está fuera de línea temporalmente? ¿

¿Utilizamos un servicio alternativo o esperamos a que el servicio este de nuevo on-line?

- **Seguridad:**

Diversos Web Services son publicados para ser utilizados sin restricción, pero muchos otros no ya que van a necesitar autenticación para ser consumidos solo por usuarios autorizados.

¿Cómo autentifica a los usuarios que consumen un Web Services?

¿Lo hace a nivel del método que lo implementa?

¿Se debe utilizar otro Web Services para realizar la autenticación?

- **Responsabilidad:**

En el caso de que un Web Service sea de uso privado, ¿Cómo puedo definir cuantas veces se puede consumir el Web Service?

¿Su uso tiene costo?

¿Cómo se alerta que el Web Service ya no está más en línea?

- **Tecnologías asociadas**

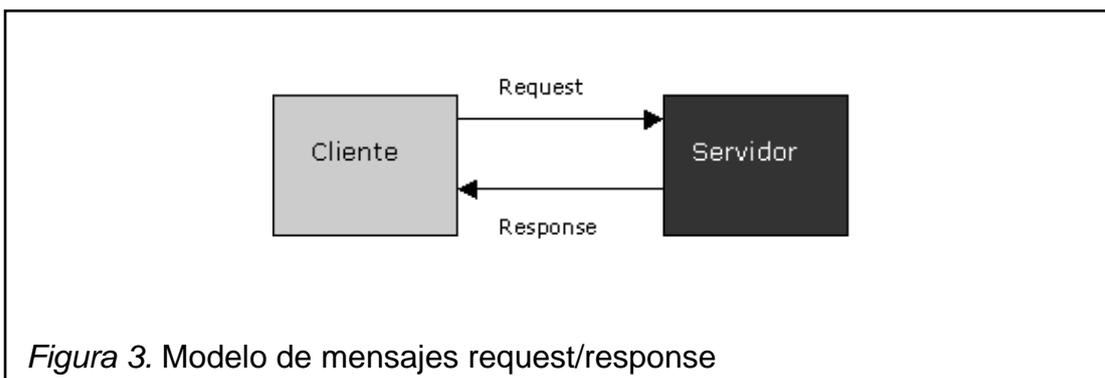
El modelo de Web Services se basa en ciertas tecnologías emergentes. Estas tecnologías son SOAP, REST, WSDL.

### **SOAP (Simple Object Access Protocol)**

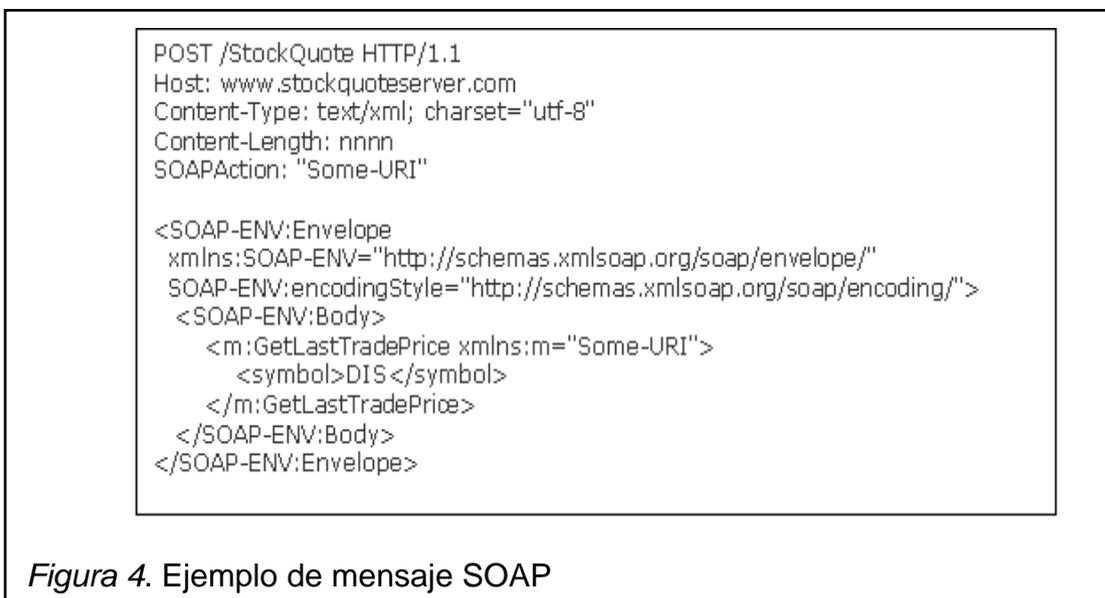
SOAP es un protocolo para el intercambio de información; es decir; dos objetos en distintos procesos pueden comunicarse a través del Web Service. Este es uno de los protocolos más utilizados para realizar el intercambio de información.

Se basa en XML, puede ser manejado en combinación con una gran variedad de protocolos de comunicación. Se utiliza HTTP para transportar la información y XML para representar la misma.

En la figura 3 se detalla el modelo de mensajes que maneja SOAP



En la figura 4 se detalla un ejemplo de un mensaje SOAP embebido en un request HTTP.



El ejemplo de la figura 4 invoca al servicio StockQuote llamando al método GetLastTradePrice con el símbolo DIS por parámetro.

Este es la respuesta al requerimiento anterior, el cual retorna el precio de la acción solicitada:

### **WSDL: Web Services Description Language**

WSDL es un protocolo basado en XML que describe los accesos al un Web Service. Se puede decir que es el manual de operación de los servicios web, ya que, nos indica cuáles son las interfaces que provee el servicio web y los tipos de datos necesarios para su utilización.

Características de WSDL:

- Provee información de los distintos métodos que provee el Web Service.
- Provee el método de acceso al Web Service.
- Da los formatos que deben de tener los mensajes que se envían y reciben

### **REST**

REST (Representational State Transfer). Es un tipo de arquitectura de desarrollo web que usa el estándar HTTP en su totalidad.

REST permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo que entienda HTTP. Por esta razón es más simple y convencional que otras alternativas que se han usado como SOAP y XML-RPC.

### **Niveles de calidad de REST**

Existen tres niveles de calidad en REST a la hora de desarrollar una aplicación web especialmente una API. Estos niveles se toman de un modelo llamado Richardson Maturity Model en honor su creador Leonard Richardson. Leonard Richardson es un ingeniero de UCLA que desarrollo de la arquitectura orientada a recursos.

Los 3 niveles son:

- Uso correcto de las URIs  
Este uso busca la universalización del servicio

- Uso correcto de HTTP.  
Este uso busca la correcta aplicación de los métodos HTML a la hora de transportar datos
- Implementación de Hipermedia.  
Esta implementación busca conectar mediante vínculos las aplicaciones clientes con las APIs.

Cada uno de estos niveles permite que REST sea muy uniforme al momento de comunicar datos. Esta uniformidad le permite usar diferentes lenguajes de transporte de información como son XML y JSON

#### **1.4 Conceptos básicos de Microsoft SQL SERVER**

##### **Base de datos relacional**

Una base de datos se puede definir como un conjunto de información que pertenece al mismo contexto y que se encuentra agrupada o almacenada para su uso posterior (informatico, 2014).

El objetivo de agrupar los datos de una manera ordenada es generar una estructura ordenada de datos que permita un crecimiento escalable y que cumpla normas específicas basadas en las mejores prácticas y manteniendo un estándar.

En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta (informatico, 2014).

##### **Sistema Administradora de Base de Datos (SGBD)**

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan (informatico, 2014).

Los sistemas de gestión facilitan el manejo de los datos usando un lenguaje de programación para bases de datos que permite una interacción con el usuario de una manera entendible, ordenada y programable.

Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System (informatico, 2014).

## **SQL SERVER**

### **Concepto:**

SQL SERVER es un sistema administrador de Base de Datos Relacional, Cliente Servidor, que permite una mayor escalabilidad de explorar objetos de Base de Datos y la integración de secuencias de los comandos en la base de Datos OLTP y OLAP. Contiene las versiones 2000, 2005, 2008, 2008 R2 y 2012, esta última versión fue presentada en este año.

En sus últimas dos versiones SQL SERVER facilita una plataforma integral empresarial con procedimientos analíticos integrados en la cual se incluye:

- El procesamiento Analítico en Línea (OLAP).
- Minería de Datos (OLAP).
- Las Herramientas de gestión y administración.
- El almacenamiento de datos y desarrollo de informes.

SQL SERVER permite que las empresas construyan y desarrollen aplicaciones de inteligencia empresarial robustas, sin complicaciones y controlar el costo en el desarrollo de estas aplicaciones (informatico, 2014). Permite a realizar los siguientes aspectos:

- Desarrollar e innovar aplicaciones empresariales.
- Optimizar la productividad de los TI, reduce la complejidad en la creación y administración de la aplicación de base de datos.
- Aumentar las capacidades de los programadores con un entorno de desarrollo Flexible y actual.
- Compartir datos a través de múltiples plataformas y aplicaciones.

Principales características de SQL SERVER 2008:

- Comprensión de datos.
- Paralelismo de tablas particionada.
- Optimización de Star Join Query.
- Integración optimizada con el sistema Office.
- Reporting Services Mejorado.
- Las consultas distribuidas en el motor de base de datos.
- Los altos niveles de seguridad, encriptación de datos.
- Programabilidad en el motor de Base de Datos.
- Desarrollo de base de datos espejos.
- Integración en XML de datos Relacionados.
- Arquitectura de interfaz virtual.
- Entre otros

### **Creación de base de datos**

Puede utilizar SQL Server para realizar procesamiento de transacciones, almacenar y analizar datos, y generar nuevas aplicaciones de base de datos.

SQL Server es una familia de productos y tecnologías que cumple los requisitos de almacenamiento de datos de los entornos de procesamiento de transacciones en línea (OLTP) y procesamiento analítico en línea (OLAP) (informatico, 2014). SQL Server es un sistema de administración de bases de datos relacionales (RDBMS) que:

- Administra el almacenamiento de datos para las transacciones y los análisis.
- Almacena datos en una amplia gama de tipos de datos, incluyendo texto, numérico, lenguaje de marcado extensible (XML) y objetos grandes.
- Responde a las solicitudes de las aplicaciones cliente.
- Utiliza Transact-SQL, XML u otros comandos de SQL Server para enviar solicitudes entre una aplicación cliente y SQL Server.
- El componente RDBMS de SQL Server es responsable de lo siguiente:
- Mantener las relaciones existentes entre los datos de una base de datos.
- Garantizar que los datos se almacenan correctamente y que no se infringen las reglas que definen las relaciones entre los datos.
- Recuperar todos los datos hasta un punto de coherencia conocida, en caso de que se produzca un error del sistema.

### **Bases de datos OLTP**

Las tablas relacionales organizan los datos de una base de datos OLTP para reducir la información redundante y aumentar la velocidad de las actualizaciones. SQL Server permite que un número elevado de usuarios realicen transacciones y modifiquen simultáneamente datos en tiempo real en bases de datos OLTP.

### **Bases de datos OLAP**

La tecnología OLAP se utiliza para organizar y resumir grandes cantidades de datos de manera que un analista pueda evaluar los datos rápidamente y en tiempo real. Microsoft SQL Server Analysis Services organiza estos datos para admitir una amplia variedad de soluciones empresariales, desde informes y análisis corporativo hasta modelado de datos y ayuda a la toma de decisiones (informatico, 2014).

### **Guardar las bd en sql server**

Al crear una base de datos, es importante entender cómo SQL Server 2008 almacena los datos para que pueda calcular y especificar la cantidad de espacio en disco que debe asignar a los archivos de datos y registros de

transacciones (informatico, 2014). Considere los datos e instrucciones siguientes sobre el almacenamiento de datos:

- Todas las bases de datos tienen un archivo de datos principal (.mdf) y uno o más archivos de registro de transacciones (.ldf). Una base de datos también puede tener archivos secundarios de datos (.ndf). Estos archivos físicos tienen los nombres de los archivos del sistema operativo y los nombres de los archivos lógicos que se pueden usar en las instrucciones Transact-SQL. La ubicación predeterminada para todos los archivos de datos y los registros de transacciones es C:\Archivos de programas\Microsoft SQL Server\MSSQL10\_50.MSSQLSERVER\MSSQL\DATA.
- Al crear una base de datos, una copia de la base de datos model, donde se incluyen las tablas del sistema, se copia en la base de datos y el resto de la base de datos se rellena con páginas vacías.
- Los datos se almacenan en bloques de 8 kilobytes (KB) de espacio en disco contiguo llamado páginas. Esto significa que una base de datos puede almacenar 128 páginas por megabyte (MB).

Consideraciones a utilizar antes de crear una Base de Datos:

A la hora de planear una nueva base de datos, debe tener en cuenta varios aspectos (informatico, 2014). Ellos incluyen, pero sin limitarse a, los siguientes:

- Propósito del almacenamiento de datos. Las bases de datos OLTP y OLAP tienen propósitos diferentes y, por tanto, tienen distintos requisitos de diseño.
- Rendimiento de transacciones. Las bases de datos OLTP suelen tener un requisito alto en cuanto al número de transacciones que se pueden procesar por minuto, hora o día. Un diseño eficiente con un nivel adecuado de normalización, índices y particiones de datos puede lograr un rendimiento de transacciones muy alto.

- Crecimiento potencial del almacenamiento físico de datos. Unas cantidades de datos elevadas requieren un hardware adecuado en cuanto a memoria, espacio en disco duro y capacidad de la unidad central de procesamiento (CPU). Una estimación de la cantidad de datos que su base de datos almacenará durante los meses y años venideros ayudará a garantizar que la base de datos siga funcionando eficazmente. Puede configurar las bases de datos para que los archivos crezcan automáticamente hasta alcanzar un tamaño máximo especificado. Sin embargo, el crecimiento automático de los archivos puede afectar el rendimiento. En la mayoría de las soluciones de bases de datos basadas en servidor, debe crear la base de datos con unos archivos correctamente dimensionados, supervisar el uso del espacio y reasignar más espacio sólo cuando sea necesario (informatico, 2014).
- Ubicación de los archivos. El lugar donde coloca los archivos de base de datos puede afectar el rendimiento. Si tiene la posibilidad de utilizar varias unidades de disco, puede repartir los archivos de base de datos en más de un disco. Esto permite a SQL Server aprovechar varias conexiones y varios cabezales de disco para lograr una lectura y una escritura eficientes de los datos (informatico, 2014).

Las bases de datos OLTP se ajustan a las necesidades de la empresa Xanda Linkworks.

Actualmente, la empresa usa este motor de base de datos para la gestión de sus diferentes clientes que se ajustan a cada tipo de negocio.

En un futuro la empresa usara el modelo OLAP y así usar la minería de datos para mejorar el manejo de los mismos y tener estadísticas que permitan analizar y mejorar su gestión.

## 1.5 Conceptos básicos de Android

Android se trata de un sistema operativo utilizado en teléfonos móviles.

iPhone ingresa al mercado a través de Apple. El Iphone se caracteriza por tener un estilo diferente y una interfase que cualquier persona la puede utilizar. Al salir al mercado, la idea de que Iphone revolucionaría el mundo móvil empezó a salir un rumor que decía que Google crearía un sistema que estaría compuesto de sus servicios web, mayoritariamente, siendo éstos: Google maps y el servidor de correo electrónico Gmail.

Al correr este rumor por la red, miles y miles de noticias empezaron a hablar sobre esta nueva creación. Sin embargo, el tiempo pasó y este nuevo invento nunca salió a la venta. Lo que sí era cierto es que Google estaba preparando algo grande que funcionaría con una plataforma gratuita.

Tiempo después, Google sacó al mercado el nuevo sistema operativo que trabaja en GNU/Linux. Este sistema recibió el nombre de Android.

Android trabaja con 34 empresas, siendo algunas de ellas: HTC, Samsung, Qualcomm, Motorola, entre otras.

Lo que quería Google con esta creación era ampliar su poder a los equipos móviles, especialmente en el área publicitaria. Además, ésta permitiría regular el acceso de las personas a la red.

### **Definición de android**

El sistema operativo Android se define de la siguiente manera:

Android es una plataforma de software utilizado en equipos móviles. Éste está formado por un sistema operativo, middleware y aplicaciones base. Cuenta con varios instrumentos relacionados a Linux. Aunque Android puede parecer una

maravilla, hay que recalcar que éste no puede funcionar como un sistema operativo sin funcionar con otras aplicaciones.

Esta plataforma funciona mediante un código libre, que no exige ningún tipo de pago. Se pueden crear aplicaciones mediante el SDK, utilizando Java y haciéndolo correr con Dalvik. La creación de aplicaciones hace que los teléfonos móviles brinden mayores beneficios a los usuarios. Con un solo teléfono, el usuario puede realizar una infinidad de tareas.

### **Linux Kernel**

Android funciona para servicios como seguridad, memoria, procesos en Linux 2.6.

La máquina virtual que utiliza Android, permite mejorar el funcionamiento de la memoria del dispositivo.

Se prevé que Android continúe mejorando sus servicios. Un aspecto característico de Android es que no discrimina entre núcleo y aplicaciones exteriores al equipo. Esto hace que el usuario personalice completamente su teléfono, mejorando su experiencia con equipos móviles.

### **Aplicaciones rompe fronteras**

Android revoluciona el mundo de las aplicaciones móviles, ya que los datos del internet pueden ser conjugados con los de un teléfono móvil. Con este sistema, la creación de aplicaciones que permitan dar a conocer la ubicación de una persona es mucho más sencilla.

### **Rápido y fácil desarrollo de aplicaciones**

Android no sólo permite crear aplicaciones que mejoran la calidad de vida de los usuarios, sino que éste también permite acceder bibliotecas que ayudan a construir diversos tipos de aplicaciones móviles.

Hoy en día es posible obtener la ubicación exacta de donde se encuentra el equipo. Anteriormente eso no era posible.

### **Android runtime**

Una ventaja de este sistema operativo es que cuenta con bibliotecas sencillas que le brinda la funcionalidad accesible en las bibliotecas de Java más importantes.

Las aplicaciones de este sistema funcionan bajo exclusivo proceso. Un ejemplo de esto constituye Dalvik, ya que ésta se ha creado de tal manera que un equipo pueda proceder a operar varias máquinas virtuales de una manera correcta.

### **Herramientas de desarrollo**

El Android SDK dispone de herramientas que brindan mayor facilidad en la creación de aplicaciones en Android. Aquí, la herramienta fundamental constituye "Android Emulator" y "Android Development Tools".

### **Emulador android**

Se trata de terminal virtual que funciona en un equipo de computación. Sirve para delinear y limpiar aplicaciones en un espacio parecido al de un aparato que existe verdaderamente.

### **Android development tools plugin para eclipse IDE**

El "ADT plugin" mejora el espacio de Eclipse, ya que hace que los procesos de limpieza y creación sean más sencillos. Éste motiva a la creación de las aplicaciones. Se diferencia porque brinda apertura a otras formas de desarrollo asociadas con Eclipse IDE, establece un asistente que fabrica los directorios y datos requeridos para elaborar una aplicación. También, reduce y automatiza el camino necesario para crear aplicaciones nuevas.

### **Dalvik debug monitor service - DDMS**

Funciona con la "Dalvik Virtual Machine". Esta herramienta ayuda a controlar los procesos emulador-equipos y los limpia. "Dalvik Virtual Machine" permite

detener un procedimiento, limpiarlo, producir informes, realizar impresiones de monitor, entre otros.

### **Android debug bridge - ADB**

Sirve para colocar aplicaciones con la extensión .apk en una solicitud de emulador- equipo para llegar usar comandos de línea.

### **Herramienta android asset packaging - AAPT**

Esta herramienta crea archivos de formato .apk. Estos archivos se caracterizan porque poseen imágenes de dos elementos del código.

### **Android interface description language - AIDL**

Esta herramienta ayuda a producir código que va destinado a una interface de interproceso.

### **Sqlite3**

Esta herramienta se creó para mejorar el trabajo de los programadores.

Ayuda a ingresar a los archivos de datos "SQLite" empleados por las aplicaciones.

### **Traceview**

Genera un reporte visual sobre los resultados de la investigación de los datos obtenidos por medio de bitácoras que pueden ser producidas a través de una aplicación Android.

### **Mksdcard**

Sirve para producir una imagen de disco. Ésta puede ser empleada en conjunto con el emulador, para hacer parecer que existe una tarjeta externa de almacenamiento.

### **Dx**

Sirve para transformar datos de tipo "bytecode" estándar (".class") en información "Android bytecode" (".dex").

## **Activitycreator**

Produce archivos "Ant build" que permite reunir todas las aplicaciones Android.

## **Características de android.**

Android permite la creación de una máquina virtual llamada Dalvik, la cual es ha sido mejorada para el uso especial de los equipos móviles. Otra característica es que dispone de un buscador fundado en WebKit. Así también genera gráficos de gran calidad, ya que cuenta con librerías de dos dimensiones.

- SQLite: Sirve para generar un almacenamiento de información.
- Sistema multimedia: Incluye audio-video, imágenes o fotografías de diferentes estilos y capacidad.
- Telefonía GSM
- Bluetooth, red 3G, y red inalámbrica
- GPS y Cámara

## **Arquitectura de android**

Las aplicaciones bajo las que se construye Android están integradas por un usuario de correo electrónico, servidor de mensajes escritos, agenda, localización, navegador, contactos, entre otros.

Todas éstas se hacen mediante la aplicación de programación Java.

## **Framework de aplicaciones**

La arquitectura de Android fue creada para hacer más sencilla la reutilización de los elementos que lo constituyen.

Las aplicaciones de este sistema pueden dar a conocer el alcance de sus capacidades para que otras aplicaciones se valgan de éstas para funcionar de mejor manera.

Una capa de servicios aprovechables para las aplicaciones de Android está compuesta por:

- Las vistas pueden ayudar a construir otras aplicaciones
- Distribuidores de datos que consienten el ingreso a información producida por otras aplicaciones o, se puede dar el caso, que se deba participar con su información propia.
- Administrador de recursos que facilita el ingreso a cadenas, gráficos, y archivos.
- Administrador de notificaciones que genera alertas.
- Administrador de actividades que determina el estado de vida y el tiempo de las aplicaciones.

### **Librerías**

Android está formado por librerías C/C++ usadas por los integrantes de este sistema. Algunos de los framework de Androidson: System C library, librerías de medios; de gráficos, 3d, SQLite, etc.

### **Runtime de Android**

Las librerías de Android tienen gran parte de las funcionalidades presentes en las librerías base de Java. En este sentido, las aplicaciones de este sistema corren un proceso específico, valiéndose de la máquina virtual Dalvik.

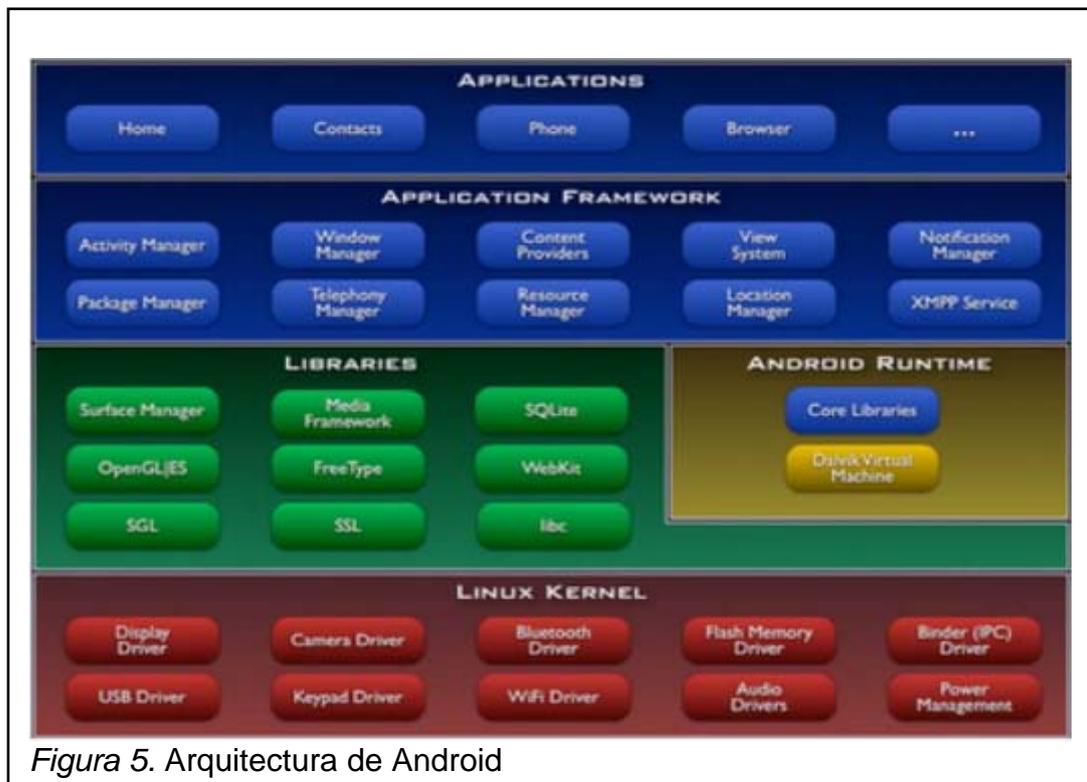
Esta máquina virtual ha sido escrita de forma que un dispositivo pueda correr en múltiples máquinas virtuales de forma eficiente. Esta máquina se rige en registros.

### **Núcleo – Linux**

Android utiliza la versión 2.6 de Linux para que corran los servicios base del sistema: seguridad, gestión de memoria, gestión de procesos, stack de red, y modelo de drivers.

Asimismo, esta parte funciona como una capa de abstracción entre la parte tangible de equipo, así como de la intangible.

El diagrama a continuación muestra los principales componentes del sistema operativo Android:



## 1.6 Sistema de Posicionamiento Global

El ser humano ha tenido de curiosidad de conocer su ubicación en la Tierra. Por eso, es que se han creado varias técnicas de navegación, las cuales permiten una transportación aérea y marítima más segura y fiable.

Los avances de la tecnología en el campo de la ubicación geoespacial han sido sorprendentes.

En los años 70 apareció la navegación que empleaba satélites. En 1973 el GPS tuvo sus primeras pruebas. Este sistema está formado por las siguientes partes:

- SS (Space Segment),
- CS (Control Segment)

- US (User Segment)

### **Control Segment**

Básicamente este segmento mantiene la integridad tanto de los satélites como de los datos que ellos transmiten; el Segmento de Control Máster se encuentra ubicado en la base aérea Falcón en Colorado y es el nodo central de control de los satélites 24 horas al día, 7 días a la semana y 365 días del año. Los datos del estado de los satélites son enviados constantemente a esta estación para su control y monitoreo.

### **User Segment**

El GPS navegación en tres dimensiones (X, Y, Z), lo cual permite conocer la ubicación de una persona.

El GPS es capaz de calcular la ubicación de un usuario en el espacio y tiempo. Este proceso lo realiza mediante la recepción de diferente tipo de señales y un software de cálculo.

El receptor está formado por una antena, un receptor, un software específico y los sistemas de comunicación.

### **Space Segment**

Este conjunto de satélites hace que se asegure la presencia de cuatro satélites, como mínimo, en el horizonte y la superficie de la tierra.

## **1.7 Conceptos básicos de arquitectura basada en capas**

La arquitectura por capas es una de las técnicas más habituales para construir y organizar un sistema informático. El beneficio de este tipo de arquitectura es la modularidad, escalabilidad y seguridad que esta ofrece, ya que la capa más alta utiliza varios servicios definidos por la capa inferior, pero ésta no tiene conocimiento de la existencia de una capa superior.

En términos de seguridad, normalmente, cada capa oculta a las capas inferiores y, por el hecho de ser modular, la reparación del código en caso de errores es más simple. Esto se da porque cada capa tiene una función determinada, lo que facilita la detección de errores.

A continuación se describe una arquitectura de tres capas:

**Capa de base de datos:** En esta capa se encuentra el motor de la base de datos con el modelo de los datos y sus respectivas restricciones. Aquí se almacenan los datos del sistema y de los usuarios. La capa de base se comunica con la capa de negocio, a través de handlers, “aunque para esto también es necesario en algunos casos, que tengan procedimientos almacenados y funciones dentro de la capa”. (EcuRed, s.f.).

**Capa de negocio:** En esta capa se encuentra la lógica del negocio. Esta es la capa “donde residen las funciones que se ejecutan, se reciben las peticiones del usuario, se procesa la información y se envían las respuestas tras el proceso” (EcuRed, s.f.).

En esta capa se establecen todas las reglas que deben cumplirse. Se “comunica con la de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él” (EcuRed, s.f.).

**Capa de presentación:** Es la capa que se encarga de que el sistema interactúe con el usuario y viceversa. Esta capa le presenta la información al usuario. Se la conoce también como “interfaz gráfica y debe tener la característica de ser amigable, o sea, entendible y fácil de usar para el usuario. Esta capa se comunica únicamente con la capa intermedia o de negocio” (EcuRed, s.f.).

## 1.8 Patrón de diseño Modelo – Vista- Controlador

Se trata de un patrón de diseño que divide una aplicación de software en tres módulos identificables y de funcionalidad definida. Estos módulos son: modelo, vista y controlador. Es decir, modelo MVC.

A continuación se detalla cada módulo del patrón de diseño:

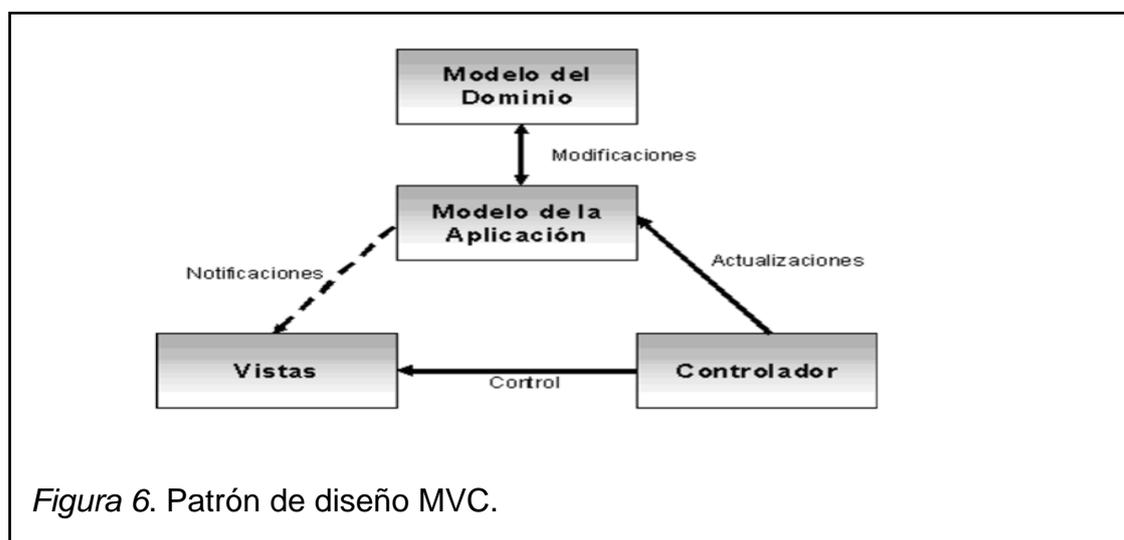
**El modelo:** Es un conjunto de clases que representan la información en la realidad que un sistema debe procesar sin tomar en cuenta la forma en que la información va ser mostrada o los métodos que se usen para que los datos estén dentro del modelo.

**Las vistas:** Son un conjunto de clases que muestran la información al usuario final que un modelo contiene. Cada vista se asocia a un modelo. Por esta razón, varias vistas pueden asociarse al mismo modelo.

Una vista despliega la información necesaria del modelo y se actualiza cada vez que el modelo cambie.

**En controlador:** Orienta el flujo del control de la aplicación en base a los mensajes que recibe externamente. Un ejemplo de esto son los datos introducidos por el usuario y las opciones de menú seleccionadas por el mismo. Esta capa tiene acceso tanto a la capa de vista como a la capa de modelo, pero las otras dos capas no conocen la existencia del controlador.

La figura 6 representa la estructura de un patrón de diseño MVC



## 1.8 Determinación de la metodología de desarrollo

Para la determinación de la metodología de desarrollo se analizaron los siguientes modelos:

- RAD
- RUP
- XP

La tabla 1 a continuación muestra un análisis de los modelos que se podrían utilizar en el desarrollo de la aplicación con el fin de determinar cuál es la metodología de desarrollo de software más viable y óptima para el desarrollo de la solución.

**Tabla 1. Análisis de metodologías de desarrollo de software para la implementación de la solución.**

<b>MODELO:</b>	<b>RAD</b>	
El desarrollo rápido de aplicaciones (RAD) es una metodología de desarrollo de software, que implica el desarrollo iterativo y la construcción de prototipos. El desarrollo rápido de aplicaciones es un término originalmente utilizado para describir un proceso de desarrollo de software introducido por James Martin en 1991		
<b>Ventajas</b>	<b>Cumplimiento para aplicación</b>	<b>Observación</b>
Se comprenden bien los requisitos y se limita el ámbito del proyecto	si	Al ser una aplicación pequeña este modelo se ajusta para la realización del mismo
Es fácil dividir al sistema en módulos	si	La aplicación se puede visualizar en módulos para cada actividad.
Se utiliza un enfoque de construcción basado en objetos reusables	si	Debido a que el web servicio se usa en toda la aplicación cumple la reusabilidad
<b>Desventajas</b>		

Continuación de la tabla 1.

Requiere recursos humanos suficientes como para crear el número correcto de equipos	no	Debido a que la aplicación es pequeña no es un impedimento que solo exista una persona en el desarrollo
Necesita que el cliente y el desarrollador se comprometan en las actividades necesarias para completar un sistema en un tiempo corto	si	Esta desventaja aplica ya que si se necesita que el cliente y el desarrollador se comprometan
<b>ETAPAS</b>		
Planificación de los requisitos	si	Se ajusta a las necesidades del proyecto
Diseño	si	Se ajusta a las necesidades del proyecto
Construcción	si	Se ajusta a las necesidades del proyecto
Implementación	si	Se ajusta a las necesidades del proyecto
<b>Conclusión</b>		
De acuerdo con lo expuesto anteriormente el sistema a desarrollarse cumple con las ventajas no se ve limitado por las desventajas y se ajusta a las etapas del modelo RAD siendo este modelo el que más se acerca a la necesidad del caso.		
<b>MODELO:</b>	<b>RUP</b>	
Es un proceso de ingeniería de software, que hace una propuesta orientada por disciplinas para lograr las tareas y responsabilidades de una organización que desarrolla software.		
Su meta principal es asegurar la producción de software de alta calidad que cumpla con las necesidades de los usuarios, con una planeación y presupuesto predecible		
<b>Ventajas</b>	<b>Cumplimiento para aplicación</b>	<b>Observación</b>
Provee un entorno de proceso de desarrollo configurable, basado en estándares	si	El desarrollo de la aplicación se puede ajustar a un estándar y puede ser configurable

Continuación de la tabla 1.

Permite tener claro y accesible el proceso de desarrollo que se sigue	si	Se ajusta a las necesidades del proyecto debido a que se tiene claro el proceso de desarrollo
Permite ser configurado a las necesidades de la organización y del proyecto	no	La organización no es tan grande y el proyecto es pequeño y aunque se puede configurar a la organización el uso del sistema se enfoca a una sola unidad organizacional
Provee a cada participante con la parte del proceso que le compete directamente, filtrando el resto	no	Solo existe una persona en el equipo de desarrollo.
<b>Desventajas</b>		
Método pesado	si	El modelo tiene muchas etapas para el tamaño del proyecto
Por el grado de complejidad puede ser no muy adecuado	si	El modelo es muy complejo a el tamaño del proyecto
En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios	si	El modelo requiere un presupuesto alto el cual la empresa no ha contemplado
<b>ETAPAS</b>		
Inicio	si	En esta etapa se levantan los requerimientos
Elaboración	no	En esta etapa se busca la arquitectura del sistema pero la ser programada en Android este ya tiene una arquitectura definida que es MVC
Construcción	si	
Transición	no	En esta etapa se entrega el producto final al usuario para retroalimentación y reinicio del modelo pero el cliente desea ver prototipos del sistema a desarrollarse que no necesariamente son el producto final
<b>Conclusión</b>		
De acuerdo con el análisis del modelo RUP se encontró que este es muy complejo, extenso y costoso para la aplicación a desarrollarse. Además se busca que se muestren prototipos de la aplicación pero el modelo no contempla este tipo de esquema.		

Continuación de la tabla 1.

<b>MODELO:</b>	XP (Extreme programming)	
En XP se trabaja estrechamente con el cliente, se hace pequeñas iteraciones cada dos semanas, donde no existe más documentación que el código en sí; cada versión contiene las modificaciones necesarias según como el cliente vaya retroalimentando al sistema, por eso es necesaria la disponibilidad del cliente durante el desarrollo.		
<b>Ventajas</b>	<b>Cumplimiento para aplicación</b>	<b>Observación</b>
Acortar los ciclos de desarrollo y reforzar la comunicación con el cliente	si	Se busca que la aplicación se desarrolle en un corto plazo
Involucrar al cliente desde el principio hasta el final de cada ciclo	no	El cliente se encuentre en una ciudad alejada a la del desarrollador
Menor tasa de errores	si	la aplicación busca la minimización de los errores en la aplicación
<b>Desventajas</b>		
Es recomendable emplearlo solo en proyectos a corto plazo	no	La aplicación es pequeña y se busca el corto plazo.
Altas comisiones en caso de fallar	si	No se tiene el presupuesto suficiente en caso de fallas
Requiere de un rígido ajuste a los principios de XP	no	El hecho de que el cliente no esté el cliente cerca del programador no permite ajustarse rígidamente a el modelo
<b>ETAPAS</b>		
Exploración	si	En esta fase se levantan los requerimientos y se familiariza con los procedimientos
Planeamiento	no	Se levanta el cronograma y la arquitectura del sistema pero se usan historias que requiere el modelo las cuales no se han levantado ya que este proceso se va implementar como prueba
Producción	no	Esta etapa requiere un tiempo máximo de 2 semanas para el desarrollo pero el cliente no puede estar en esos tiempos

Continuación de la tabla 1.

Mantenimiento	no	No se puede dar mantenimiento a un sistema que está en prototipo
Muerte	no	El cliente no levanto historias así que no se daría esta fase.
<b>Conclusión</b>		
De acuerdo con el modelo mostrado en este análisis el cliente no está cerca del programador para que este modelo funcione adecuadamente. Además la estructura de historias del modelo no se ajusta a las necesidades del cliente.		

### 1.9.1 Modelo RAD (Rapid Application Development)

De acuerdo a lo analizado posteriormente, el modelo que más se ajusta a las necesidades de la empresa para el desarrollo de la solución es el modelo RAD, ya que por su estructura de desarrollo rápido y su baja complejidad permite que el proyecto camine de una manera adecuada en su desarrollo para la ejecución de la solución.

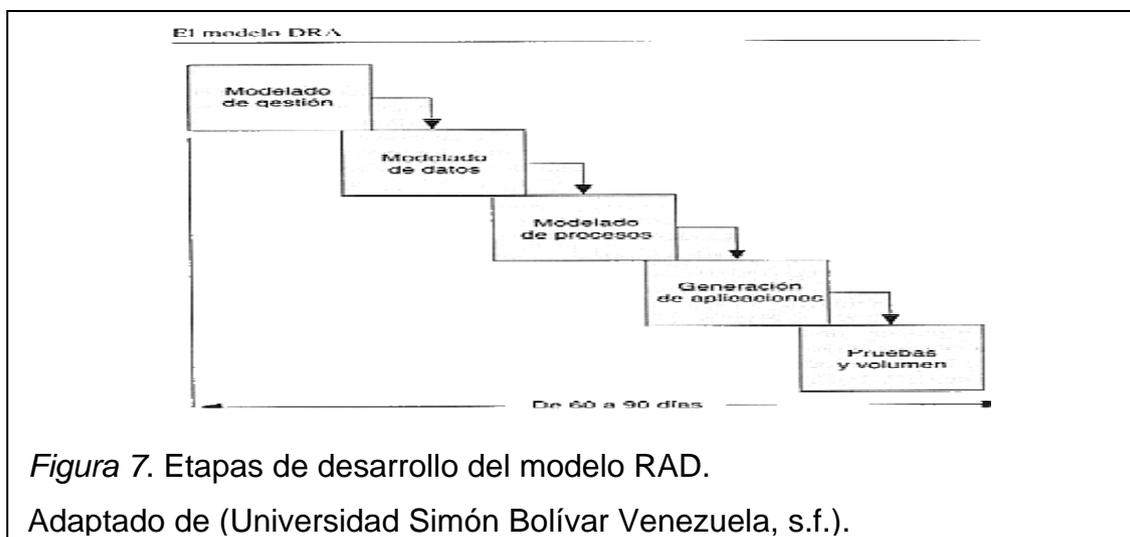
RAD es el modelo que más se ajusta a los tiempos de diseño e implementación de la solución para la gestión de cobranzas en el terreno.

El modelo de desarrollo RAD está estructurado por las siguientes etapas:

- **Modelo de gestión:** En esta fase se modela el flujo de información entre las diversas funciones de gestión. Este flujo debe responder preguntas, tales como: ¿Qué información maneja el proceso de gestión? ¿quién la genera? ¿A dónde va la información? y ¿quién la procesa? (Universidad Simón Bolívar Venezuela, s.f.)
- **Modelado de datos:** En esta fase se definen los atributos de cada objeto que se forma a través del flujo de información y de las relaciones que se dan entre ellos.

- **Modelado del proceso:** Las diversas descripciones del proceso se crean para añadir, alterar, eliminar o recuperar un objeto de datos (Universidad Simón Bolívar Venezuela, s.f.).
- **Generación de aplicaciones:** El modelo RAD reutiliza los componentes de programas creados o elabora componentes que se pueden reutilizar. No crea un software nuevo (Universidad Simón Bolívar Venezuela, s.f.).
- **Prueba y entrega:** En esta fase, el tiempo de duración de las pruebas es menor, pues los componentes ya fueron analizados y probados. Sin embargo, esto no significa que se tenga que comprobar cada componente nuevo (Universidad Simón Bolívar Venezuela, s.f.).

De acuerdo con lo analizado, la estructura a seguir para el desarrollo del software son las etapas del modelo RAD que se muestran en la figura 7:



## MODELO DE GESTIÓN

### 2.1 Estudio del Sistema existente de gestión de cobranza

#### 2.1.1 Situación actual de la gestión de cobranza

La empresa XANDA LINKWORKS usa actualmente el programa Microsoft Excel para la gestión de cobranzas en terreno de su cliente Cable Express, ubicado en la ciudad de Macas, Ecuador.

Cablexpress es una empresa que brinda servicios de Televisión por cable a la ciudad. Esta empresa envía la información de la base de datos de clientes activos, por cobrar y en mora. La empresa Xanda Linkworks se encarga de recuperar esta cartera.

El área de IT de Xanda Linkworks, ubicada en Quito, envía un archivo Excel (.xlsx) a la secretaria que labora en la sucursal de Xanda Linkworks en Macas, para realizar el proceso de cobranza. Este archivo contiene la base de datos de clientes a cobrar de Cablexpress.

En la figura 8 se muestra un ejemplo del archivo que recibe la secretaria de la sucursal de Xanda Linkworks en Macas. Los datos que se envían desde IT son los siguientes:

- Numero de Contrato
- Cedula o ruc
- Nombres
- Apellidos
- Dirección
- Sector
- Valor de suscripción
- Meses de mora
- Valor total a pagar

Nombres	Apellidos	Estado civil	Nombre de Contacto	Apellido del contacto	Parentesco	Telefono del contacto	Enero	Febrero
MARIO	ALVAREZ PAREDES	C	GLADYS	PAREDES	TIA	964278513	x	x
YURI	ALVARADO ESPINOZA	S	HERMIA	ESPINOZA	MADRE	32145894	x	x
DORIS GRACIELA	ALVAREZ SIMBAÑA	D	BRYAN	ALVAREZ	HERMANO	962585126		
CIA. LTDA.	ALMACENES MAVELI	S	MARCO	TAPIA	PADRE	994238623		x
DEL ROCIO	ACEVEDO ALEXANDRA	D	PABLO	SALLES	PRIMO	3545114		
SEGUNDO DAVID	ANDRADE PAREDES	C	CAROLINA	HERNANDEZ	ESPOSA	978854126		
FAUSTO FLORENCIO	ALVAREZ FERNANDEZ	S	HERMINIO	ALVAREZ	HERMANO	35632147		x
EDGAR PATRICIO	ALTAMIRANO CARDENAS	C	WILMA	CARDENAS	MADRE	932568451		
HUMBERTO MANUEL	ANDRADE AMOROSO	C	DIANA	BALAREZO	ESPOSA	996359847		x
RONAL ISAC	ALVAREZ BAUTISTA	C	YURIDIA	BAUTISTA	MADRE	23568452		
EDUARDO MARCELO	ANDRADE MEDINA	D	VERONICA	MEDINA	MADRE	995865327	x	x
JUAN ALBERTO	ALARCON JARAMILLO	C	MARIA	MERCHAN	ESPOSA	3514783		
MAGDALENA	ANDRAMUNO JARAMILLO	D	XAVIER	ANDRAMUNO	PADRE	995684269	x	x
ALBINO AUGUSTO	ANDRADE NIETO	S	PEDRO	ANDRADE	PADRE	26874592	x	x
BLANCA FABIOLA	ANDINO RAMOS	S	GLENDA	ANDINO	TIO	37895426	x	x
MARIA LUISA	ANDINO CASTILLO	C	ANGEL	FLORES	ESPOSO	2356898		x

*Figura 8.* Archivo de cartera de clientes por cobrar enviado desde Xanda Linkworks Quito hacia Macas

La secretaria de Xanda Linkworks, empleando el mismo programa que utilizan en Quito (Excel), realiza una repartición equitativa y sectorizada de la base de datos de clientes a cobrar a cada gestor.

Luego, ella procede a la impresión de esta información en un formato definido y se la entrega a cada gestor de cobranza en terreno. Este documento se lo conoce como hoja de trabajo y se lo entrega diariamente.

En la siguiente Figura 9 se muestra un ejemplo del archivo, con el formato específico, que se entrega a cada gestor para que realice su gestión de cobranza.

Marloon Granda Lopez								
Cedula	Nombres	Apellidos	Estado civil	Direccion	Enero	Febrero	Marzo	Abril
101324861	MARIO	ALVAREZ PAREDES	C	Av. 29 de mayo y calle P	x	x	x	x
102346640	YURI	ALVARADO ESPINOZA	S	Benjamin Delgado y Calle 2	x	x		x
102815768	DORIS GRACIELA	ALVAREZ SIMBAÑA	D	Av. Juan Padre Vigna				
190386376	CIA. LTDA.	ALMACENES MAVELI	S	12 de febrero y Calle N		x		x
201019807	DEL ROCIO	ACEVEDO ALEXANDRA	D	Benjamin Carrion y Calle 2				
300897022	SEGUNDO DAVID	ANDRADE PAREDES	C	Cuenca y Soasti				
301014882	FAUSTO FLORENCIO	ALVAREZ FERNANDEZ	S	Av. 13 de abril y Juan Rivadeneira		x	x	x
301273959	EDGAR PATRICIO	ALTAMIRANO CARDENAS	C	Calle K y Gral. Rumiñahui			x	
601733512	HUMBERTO MANUEL	ANDRADE AMOROSO	C	Calle A y Calle 1		x		x
923210785	RONAL ISAC	ALVAREZ BAUTISTA	C	Inca atahualpa y 6 de Marzo			x	x
1102848544	EDUARDO MARCELO	ANDRADE MEDINA	D	Juan Rivadeneira y Maria Jaramillo	x	x	x	x
1400003461	JUAN ALBERTO	ALARCON JARAMILLO	C	Inca Atahualpa y calle J				x
1400003540	MAGDALENA	ANDRAMUNO JARAMILLO	D	24 de mayo y Quito	x	x	x	x
1400083315	ALBINO AUGUSTO	ANDRADE NIETO	S	Av. 13 de abril y hacia la Urb. Israel	x	x	x	x
1400107379	BLANCA FABIOLA	ANDINO RAMOS	S	Calle I y Gral. Rumiñahui	x	x	x	x

*Figura 9.* Hoja de trabajo diario que se entrega a los gestores de cobranza de Xanda Linkworks en Macas

El gestor de cobranza en terreno se dirige a cumplir su labor en el sector establecido llevando los siguientes materiales:

- Hoja de trabajo diario
- Facturas
- Recibos

Se pueden dar varios eventos durante la jornada laboral del gestor. Estos se registran en la hoja de trabajo diario que se le entregó. Dentro de los eventos que se pueden presentar se encuentran:

- **Pago completo:** El cliente paga completamente el valor de su suscripción mensual.
- **Pago parcial:** El cliente paga parcialmente el valor de su suscripción mensual, dejando un saldo que el gestor debe cobrar en el transcurso del mes.
- **Negativa de pago:** El cliente se niega a pagar el valor de su suscripción mensual.
- **No se encuentra el abonado:** El abonado no se encuentra en el momento de la visita.
- **Dirección incorrecta:** La dirección dada no es la correcta para ese abonado.
- **Pago en oficina:** El abonado pagó previamente en oficina y el gestor pide el número de factura o recibo para comprobar el pago.

La secretaria recibe la información y la verifica nuevamente. Emplea la herramienta Excel para generar un informe diario de ingresos y gestión del personal de terreno. Al finalizar el informe, la secretaria envía la información por correo electrónico a las oficinas de Xanda Linkworks en Quito. Estos datos son ingresados, posteriormente, en el sistema de cobranzas de la empresa para ser analizados.

Finalmente, en las oficinas de Quito se realiza la nueva carga de trabajo diario con los datos enviados desde Macas del día anterior.

A su vez, Xanda Linkworks en Quito envía los listados de mora mensual al área técnica de Cablexpress Macas para que ellos generen el listado de corte de servicio a los abonados impagos.

Excel es una herramienta extraordinaria para su uso previsto. La realización de cálculos numéricos en filas y columnas y la visualización de datos o de resultados, con recursos gráficos, la hace una herramienta muy eficaz.

No obstante, Excel no es una base de datos. Los valores pueden introducirse sin que exista una estructura definida. Las celdas y los rangos pueden tener un nombre, pero los datos no hacen referencia a ellos de manera formal, más allá de la fila y de la columna.

El hecho de que la información de un documento de Excel sea entendida por una persona, no implica que los datos puedan ser procesados en un entorno digital. El lector humano es capaz de resolver muchas deficiencias, reparar las incoherencias y completar los vacíos de información, pero una herramienta digital no lo puede hacer.

### **2.1.2 Estudio del Sistema existente**

Xanda Linkworks no posee un sistema de cobranzas para la gestión terrena. Por esta razón, la empresa se apoya de la herramienta Microsoft Excel para llevar este proceso de una manera parcialmente ordenada.

Este proceso consta de tres formatos:

- **Formato de envío de la información:** Este formato contiene la información de la cartera de clientes para el cobro del pago mensual de la suscripción del servicio. Además, se detalla si los usuarios tienen meses de suscripción en mora.

Esta información genera el flujo de trabajo diario que se entrega a los gestores para proceder al cobro. De este formato se genera la repartición de carga de trabajo sectorizada a los diferentes gestores de cobranza de terreno.

- **Hoja diaria de trabajo:** Esta hoja se genera desde el formato de envío de información. En esta hoja se establece la carga de trabajo sectorizado para cada gestor, la cual debe ser llenada en el transcurso de la jornada laboral. El gestor llena esta hoja de trabajo de forma manual.
- **Informe de gestión diaria:** Este formato contiene el informe de gestión diaria realizada por el gestor de terreno. Lo realiza la secretaria de Xanda Linkworks en Macas. Aquí se detallan las incidencias presentadas en el día con cada cliente que visitó el gestor. El formato presentado en la figura 10 es el usado para el envío de la información

Informe Diario de cobranzas		
Mes	Marzo	
Año	2014	
Día	12	
Gestor	MARLON GRANDA LOPEZ	
	<b>Numero</b>	<b>Cobro total</b>
Facturas	5	60
Recibos	10	120
<b>Detalle de facturas</b>		
<b>Numero</b>	<b>Valor</b>	
037-001-000015846	16	
037-001-000015847	16	
037-001-000015848	16	
037-001-000015849	16	

*Figura 10.* Informe de gestión diaria realizado desde Macas para su envío a Quito.

Todo este proceso se realiza de forma manual, lo que conlleva a errores de digitación, pérdida de datos, alteración de la información, retrasos de envío y baja persistencia de los datos.

Asimismo, este proceso, por el hecho de ser manual, requiere de mayor tiempo, esfuerzo y consumo de recursos humanos y económicos. Estos procesos disminuyen la eficiencia de la gestión, pues se alargan los tiempos

de cobro. Además, el gestor pierde tiempo al regresar a las oficinas de Xanda Linkworks para entregar la hoja de trabajo diaria. La actualización de la información no es constante y a tiempo. Por ser manual, está sujeta a errores. Debido a estos errores, se encontró la necesidad de implementar un prototipo de software para dispositivos móviles que agilite la gestión del personal de terreno. Este prototipo busca reducir los errores de información, optimizar el tiempo y los recursos del personal de terreno de la empresa Xanda Linkworks y reducir costos.

Para el estudio del sistema existente se responderá a la pregunta: ¿Qué información maneja el proceso de gestión?

La información que maneja el proceso de gestión se obtiene de la base de datos de clientes activos de la empresa.

Los clientes activos son aquellos usuarios que tienen el servicio de audio y video por suscripción y cuyos pagos se encuentran al día. También se encuentran los usuarios que tienen varios meses de mora en los pagos y los usuarios que tienen el servicio suspendido por falta de pago.

La información que maneja este proceso es la siguiente:

1. Número de cédula o RUC del abonado
2. Nombres del abonado
3. Apellidos del abonado (en caso de empresa este dato es nulo)
4. Estado civil (en caso de empresa este dato es nulo)
5. Dirección del abonado
6. Teléfono del abonado
7. Estado civil del abonado (en caso de empresa este dato es nulo)
8. Nombre del contacto del abonado
9. Apellido del contacto del abonado
10. Parentesco del contacto del abonado
11. Teléfono del contacto del abonado
12. Correo electrónico del abonado

13. Número de contrato del abonado
14. Número de televisiones contratadas por el abonado
15. Valor a cancelar por el contrato
16. Valor total a cancelar por el servicio
17. Fecha de inicio de contrato
18. Tipo de contrato
19. Mes de pago
20. Meses en mora

### **2.1.3 Manejo de la información de gestión de cobranza**

Para el estudio del sistema existente se responderá a la pregunta: ¿Quién genera la información?

La información es generada por la empresa Cablexpress Macas. Esta es extraída de su base de datos de abonados activos, en proceso de cobro mensual que estén al día, en mora y en suspensión.

### **2.1.4 Destino de la información de gestión de cobranza**

Para el estudio del sistema existente se responderá a la pregunta: ¿A dónde va la información?

La información de gestión de cobranzas de la empresa Cablexpress Macas es recibida diariamente por Xanda Linkworks en Quito, vía correo electrónico.

Una vez recibida la información, el departamento de IT de Xanda Linkworks ordena la información en un formato establecido, que se mostró en el capítulo II de este proyecto, para enviarlo a las oficinas de Xanda Linkworks en Macas.

Cuando esta información es recibida, se reparte, equitativamente y de manera sectorizada, a los gestores de cobranza en terreno de Macas para proceder al cobro.

### 2.1.5 Procesamiento de la información de gestión de cobranza

Para el estudio del sistema existente se responderá a la pregunta: ¿Quién procesa la información?

La información de gestión de cobranza es procesada por los gestores de terreno. Estas personas, en cada una de sus áreas, se acercan a los usuarios para solicitar el pago del servicio. En este caso, pueden presentarse cinco escenarios:

1. **Pago completo:** El gestor de cobranza recibe el pago completo del servicio, ya sea de uno o varios meses. Luego el gestor entrega la factura o recibo de la transacción y lo registra en el formato establecido (hoja de trabajo diario), entregado por la secretaria encargada.
2. **Pago parcial:** El gestor de cobranza recibe el pago parcial del servicio, ya sea de uno o varios meses. El gestor no entrega la factura o recibo de la transacción hasta que no se complete el pago dentro del periodo determinado. Finalmente, el gestor registra en la hoja de trabajo diario, entregada por la secretaria encargada.
3. **Negativa de pago:** En este caso, el abonado se niega a pagar bajo alguna excusa. El gestor no registra la transacción.
4. **Dirección errónea:** El gestor no puede encontrar la dirección del abonado por diversas razones: mala digitación del texto, mal sectorizado, la dirección no existe. El gestor registra el evento en su hoja de trabajo.
5. **Visita con notificación:** El gestor visita al abonado. Todos los datos de ubicación son correctos, pero el gestor no localiza al usuario. En este caso se deja una notificación escrita o verbal de la visita. El gestor no registra el evento en su hoja de trabajo.

Al final de la jornada laboral, los gestores llevan a las oficinas de Xanda Linkworks en Macas las hojas de trabajo, los recibos, las facturas y el dinero recuperado.

Antes de proceder a la entrega del dinero recuperado, los gestores realizan un cuadro de caja para verificar lo recuperado contra las facturas y recibos.

Finalmente, la secretaria recibe estos valores y vuelve a verificar para generar un informe de cobranza diaria. Este informe es enviado a las oficinas de Xanda Linkworks en Quito, para su posterior análisis y generación de carga de trabajo para el siguiente día.

En la figura 11 se muestra el diagrama de actividades UML del proceso de cobranzas y actualización de datos del cliente.

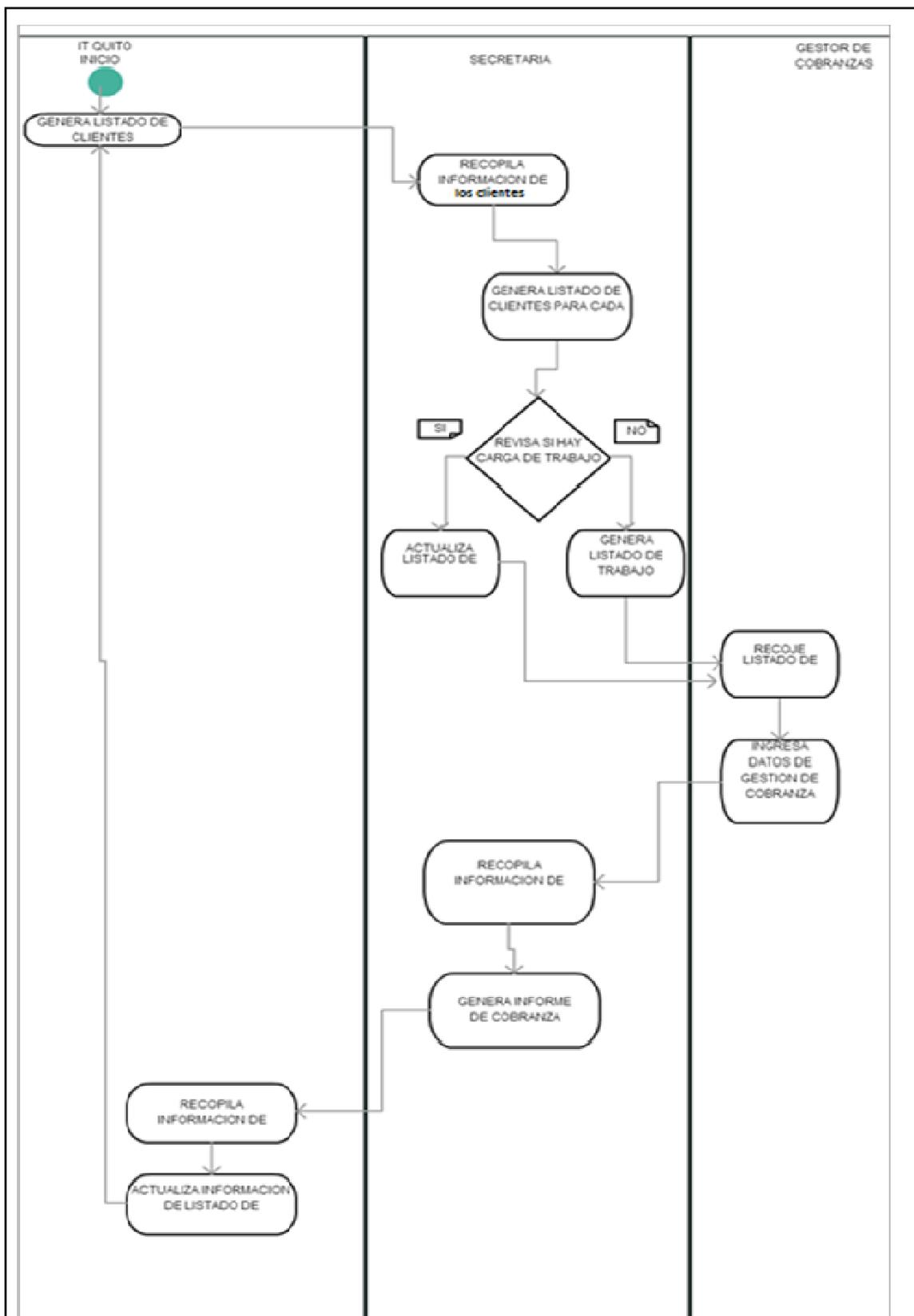


Figura 11. Diagrama de actividades UML del proceso de cobranza

De acuerdo con la figura anterior, la información que genera el gestor de terreno se produce en la calle y, para informar su gestión, debe retornar a la oficina llevando el dinero cobrado y la información de cobro y/o actualización de datos de los clientes que se le asignó a su carga de trabajo diario.

La solución informática que se plantea para mejorar este proceso es un prototipo de aplicación móvil que permita al gestor de terreno actualizar los datos de gestión de cobranza e información de los abonados, en el lugar donde se encuentre y así no perder tiempo en viajes innecesarios. Además, el prototipo de la aplicación busca ayudar al proceso de cobranza, aumentando datos de tipo geo referencial y fotográficos para una mejor ubicación del abonado y gestionarlo.

El prototipo de la aplicación móvil busca recolectar información sobre la gestión que realiza el personal en el terreno, ya que, actualmente, no se tiene información alguna de la misma. Como se mostró anteriormente en este capítulo, el informe que recibe la secretaria consta solo de los abonados cobrados, pero no se toma en cuenta si hubo visitas u otro tipo de contacto a los abonados del listado de trabajo del gestor. Por esta razón, el prototipo de la aplicación móvil busca recolectar la información de las visitas realizadas por el gestor y con esto tener un control y un registro de las actividades realizadas.

## MODELADO DEL PROCESO

### 3.1 Arquitectura de la aplicación

Para desarrollar el prototipo de la aplicación se empleó una estructura de desarrollo de software en tres capas. Las capas que se utilizaron son:

- Capa de base de datos
- Capa de negocio
- Capa de acceso

La figura 12 muestra la representación gráfica de la arquitectura y los componentes que comprenden cada una de las capas usadas para el prototipo.



Figura 12. Arquitectura del prototipo

#### 3.1.1 Capa de base de datos

En esta capa se encuentra el motor de base de datos, en la cual están todos los datos de la aplicación.

El diseño de la estructura de la base de datos se especificará en el siguiente capítulo.

### 3.1.2 Capa de Negocio

Para el desarrollo de la capa de negocio de la aplicación se establecieron todas las funciones y validaciones para que el cliente gestione los datos que residen en la capa anterior.

#### Funciones de la capa de negocios:

- **Login**

Esta función permite a un usuario registrarse en el prototipo de la aplicación, usando su nombre de usuario y su contraseña.

Para realizar este proceso, la capa de negocios hace la siguiente consulta a la tabla de Tbl\_Usuarios de la base de datos:

```
SELECT COUNT (*) FROM DBO.TBL_USUARIOS WHERE
USR_NOMBREUSUARIO='$USERNAME' AND
USR_PASSWORD='$PASSWORD' AND USR_STATUS='A'
```

Donde \$username es el Nombre de usuario, \$password es la contraseña y 'A' es el estatus del usuario.

Si el resultado de esa consulta es igual a 1, entonces se genera una nueva consulta:

```
SELECT USR_ID, USR_NOMBRES, USR_APELLIDOS,
USR_NOMBREUSUARIO, TUSR_ID FROM DBO. TBL_USUARIOS WHERE
USR_NOMBREUSUARIO='$USERNAME' AND
USR_PASSWORD='$PASSWORD'
```

Que posteriormente se envía a la capa de presentación para su posterior uso. Caso contrario, si el resultado es igual a 0, no se envía un dato de tipo nulo a

la capa de presentación para su posterior uso.

Si esta función pierde conexión con la base de datos, enviará un mensaje de error a la capa de presentación.

- **Asignación**

Esta función permite consultar la carga de trabajo asignada a un gestor, realizando la siguiente consulta a la base de datos:

```
SELECT B.PER_ID+' '+B.PER_NOMBRES+' '+B.PER_APELLIDOS AS
ASIGNACION, A.CONT_ID AS ID_CONTRATO FROM TBL_ASIGNACION A
INNER JOIN TBL_PERSONA B ON A.PER_ID=B.PER_ID WHERE
A.USR_ID=$USR_ID AND A.ASIG_ESTADO='$STATUS'
```

Donde \$usr\_id es la identificación del usuario y \$Status es el status de la asignación.

Los resultados de la consulta se envían a la capa de acceso para su posterior uso.

Si esta función pierde conexión con la base de datos, enviará un mensaje de error a la capa de presentación.

- **Contrato**

Esta función permite buscar los datos del contrato que son necesarios para el proceso de cobranzas, realizando la siguiente consulta:

```
SELECT A.CONT_ID, A.CONT_VALORPAGO, B.DIR_DIRECCION,
B.DIR_IMAGEN, B.DIR_LATITUD, B.DIR_LONGITUD, B.DIR_SECTOR,
A.TCONT_ID FROM [DBO]. [TBL_CONTRATOS] AS A INNER JOIN [DBO].
[TBL_DIRECCION] AS B ON A.DIR_ID=B.DIR_ID WHERE
A.CONT_ID='$CONT_ID'
```

Donde \$ cont\_id es la identificación del contrato.

Los resultados de la consulta se envían a la capa de acceso para su posterior uso.

Si esta función pierde conexión con la base de datos, enviará un mensaje de error a la capa de presentación.

- **Contratos Persona**

Esta función permite buscar los datos de los contratos de servicio de CATV que tiene una persona. Estos datos son necesarios para el proceso de cobranzas y se obtienen realizando la siguiente consulta:

```
SELECT A.CONT_ID, B.PER_NOMBRES + B.PER_APELLIDOS, B.PER_ID,
C.TCONT_TIPOCONTRATO, C.TCONT_ID, A.CONT_VALORPAGO,
A.CONT_STATUS FROM [DBO]. [TBL_CONTRATOS] AS A INNER JOIN
[DBO]. [TBL_PERSONA] AS B ON A.PER_ID=B.PER_ID INNER JOIN [DBO].
[TBL_TIPOCONTRATO] AS C ON A.TCONT_ID=C.TCONT_ID
WHERE B.PER_ID= '$ID_ABONADO'
```

Donde \$ Id\_Abonado es la identificación del abonado.

Los resultados de la consulta se envían a la capa de acceso para su posterior uso. Si esta función pierde conexión con la base de datos, enviará un mensaje de error a la capa de presentación.

- **Recuperar Información Contrato**

Esta función permite recuperar los datos de los contratos de servicio de CATV usando el número de contrato. Estos datos son necesarios para el proceso de cobranzas y se obtienen realizando la siguiente consulta:

```
SELECT C. PER_NOMBRES, C. PER_APELLIDOS, C. PER_ID, A.CONT_ID,
A.CONT_VALORPAGO, B. [DIR_DIRECCION], B. DIR_SECTOR,
B.DIR_IMAGEN, B. DIR_LATITUD, B. DIR_LONGITUD FROM [DBO].
[TBL_CONTRATOS] AS A INNER JOIN [DBO]. [TBL_DIRECCION] AS B ON
A.DIR_ID=B.DIR_ID INNER JOIN TBL_PERSONA AS C ON A.PER_ID=C.
PER_ID WHERE A.CONT_ID='$CONT_ID'
```

Donde \$ cont\_id es la identificación del contrato.

Los resultados de la consulta se envían a la capa de acceso para su posterior uso. Si esta función pierde conexión con la base de datos, enviará un mensaje de error a la capa de presentación.

- **Recuperar Mora Mes Valor Contrato**

Esta función permite recuperar los datos de los meses en mora y el valor total de la misma de los contratos de servicio de CATV, usando el número de contrato. Estos datos son necesarios para el proceso de cobranzas y se obtienen realizando la siguiente consulta:

```
SELECT COUNT (*) AS MESES_MORA, SUM (HIM_VALOR) AS
VALOR_MORA FROM TBL_HISTORIALMORA WHERE
CONT_ID='$CONT_ID'AND [HIM_STATUS] ='A'
```

Donde \$ cont\_id es la identificación del contrato.

Los resultados de la consulta se envían a la capa de acceso para su posterior uso. Si esta función pierde conexión con la base de datos, enviará un mensaje de error a la capa de presentación.

- **Recuperar Meses Mora Valor Contrato**

Esta función permite recuperar el detalle de la mora de los contratos de servicio de CATV, usando el número de contrato. Estos datos son necesarios para el proceso de cobranzas y se obtienen realizando la siguiente consulta:

```
SELECT * FROM TBL_HISTORIALMORA WHERE CONT_ID='$CONT_ID'AND
[HIM_STATUS] ='A'
```

DONDE \$CONT\_ID ES LA IDENTIFICACIÓN DEL CONTRATO.

Los resultados de la consulta se envían a la capa de acceso para su posterior uso. Si esta función pierde conexión con la base de datos, enviará un mensaje de error a la capa de presentación.

- **Insertar gestión**

Esta función permite insertar los datos de la gestión realizada por el agente de cobranzas en la tabla de gestiones. Estos datos son necesarios para el proceso de cobranzas y se obtienen realizando la siguiente consulta:

```
INSERT INTO [DBO]. [TBL_GESTIONES] ([TGST_ID, [CONT_ID],
[TCONT_ID], [PER_ID], [USR_ID], [TUSR_ID], [GST_FECHA],
[GST_VALORPAGADO], [GST_SALDO], [GST_NROFACTUR],
[GST_NRORECIVO], [GST_MESGESTIONMORA]) VALUES (<TGST_ID,
INT, >, <CONT_ID, INT, ><TCONT_ID, INT, ><PER_ID, CHAR (10),
><USR_ID, INT, >, <TUSR_ID, INT, >, <GST_FECHA, DATE, >,
<GST_VALORPAGADO, MONEY, >, <GST_SALDO, MONEY, >,
<GST_NROFACTUR, VARCHAR (100), >, <GST_NRORECIVO, VARCHAR
(100), ><GST_MESGESTIONMORA, INT, >)
```

Los resultados de la consulta se envían a la capa de base de datos para su posterior uso. Si esta función pierde conexión con la base de datos, enviará un mensaje de error a la capa de presentación.

- **Actualizar Direccion**

Esta función permite actualizar los datos de la dirección del abonado en la tabla de direcciones. Estos datos son necesarios para el proceso de cobranzas y se obtienen realizando la siguiente consulta:

```
UPDATE [DBO]. [TBL_DIRECCION] SET, [DIR_TIPO] = <DIR_TIPO,
VARCHAR (10), >, [DIR_DIRECCION] = <DIR_DIRECCION, VARCHAR
(100), >, [DIR_SECTOR] = <DIR_SECTOR, VARCHAR (50), >,
[DIR_LATITUD] = <DIR_LATITUD, REAL, >, [DIR_LONGITUD] =
<DIR_LONGITUD, REAL, >, [DIR_IMAGEN] = <DIR_IMAGEN, VARCHAR
(200), > WHERE [PER_ID]=$PERID
```

Donde \$ PerId es la identificación de la persona.

Los resultados de la consulta se envían a la capa de base de datos para su posterior uso. Si esta función pierde conexión con la base de datos, enviará un mensaje de error a la capa de presentación.

- **Actualizar Estatus Asignacion**

Esta función permite actualizar el estatus de la asignación en la tabla Tbl\_Asignacion. Estos datos son necesarios para el proceso de cobranzas y se obtienen realizando la siguiente consulta:

```
UPDATE [DBO]. [TBL_ASIGNACION] SET [ASIG_ESTADO] =
<ASIG_ESTADO, CHAR (1), > WHERE [CONT_ID] =$CONT_ID
```

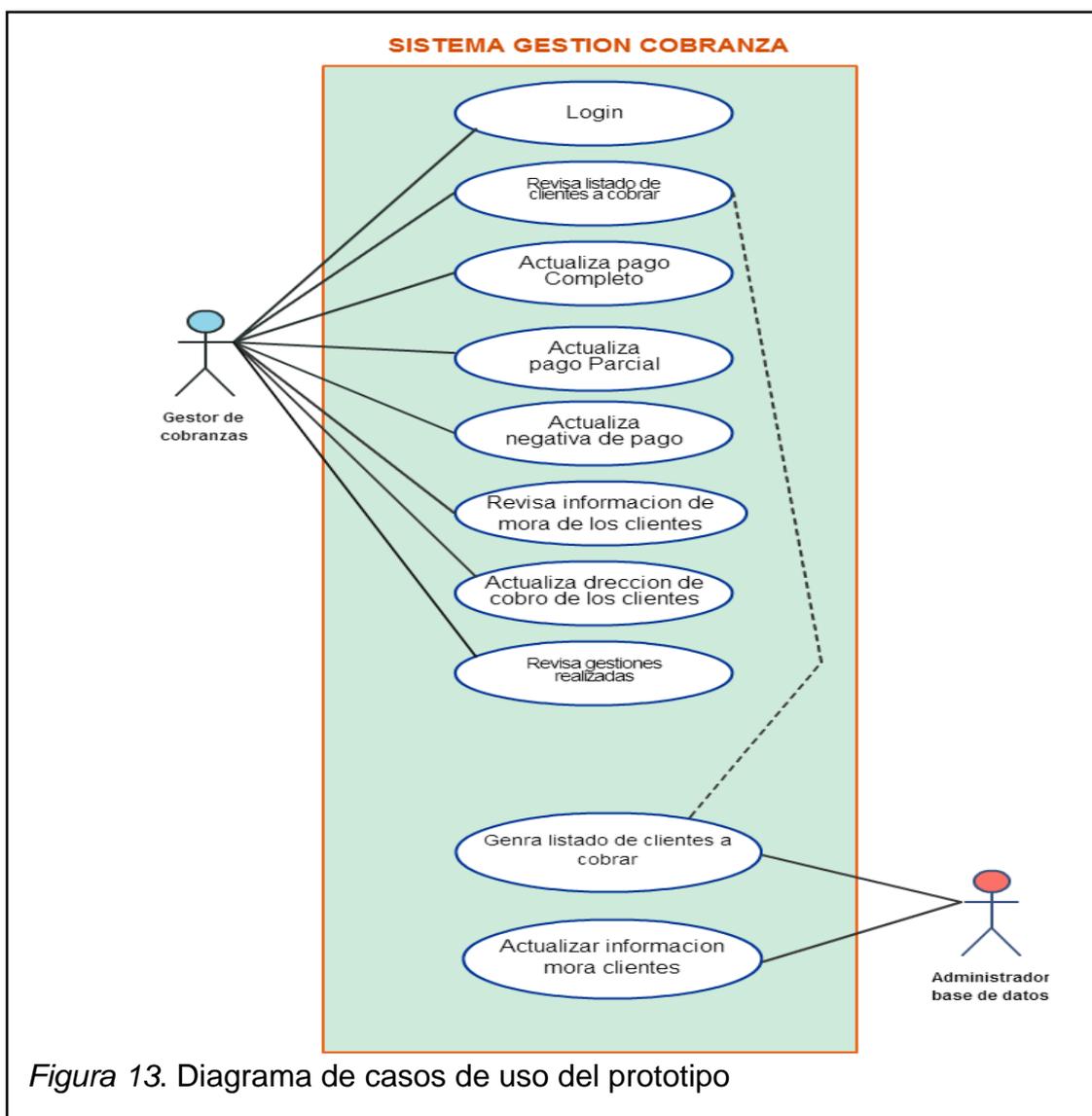
Donde \$ cont\_id es la identificación del contrato.

Los resultados de la consulta se envían a la capa de base de datos para su posterior uso. Si esta función pierde conexión con la base de datos, enviará un mensaje de error a la capa de presentación.

Todas las funciones descritas anteriormente fueron programadas como web services en lenguaje PHP. Estos servicios serán consumidos por la capa de presentación para el despliegue de información y actualización de datos hacia la capa de base de datos.

El método de conexión entre la capa de negocios y la capa de base de datos es el handler PDO. Este handler permite una fácil conexión entre la base de datos y la capa programada en PHP.

En la figura 13 se detalla el diagrama de casos de uso del prototipo.



### 3.1.3 Capa de Presentación

La capa de presentación contiene los componentes necesarios para la interacción con el usuario. En esta capa se manejan las interfaces de usuario que, en este caso es una interfaz gráfica, y los procesos encargados para el

manejo de los datos que ingresan a la capa de negocios y finalmente llegan a la capa de base de datos.

El sistema permitirá realizar las siguientes acciones:

- Gestionar pagos de abonados en el campo
- Buscar abonados para visualizar información de su contrato
- Buscar abonados para actualizar la información existente en la base de datos
- Visualizar el número de gestiones diarias realizadas por el usuario

## MODELADO DE LOS DATOS

En esta etapa, se analizarán los requisitos de la solución a implantarse para el desarrollo de una aplicación móvil para la gestión de clientes y el control del personal de campo en la empresa XANDA LINKWORKS Cía. Ltda.

La empresa Xanda Linkworks tiene la necesidad de contar con un sistema que permita llevar un mejor control de la gestión del personal de terreno en Macas, que sea amigable con el usuario. El control consiste en llevar a cabo un registro de todas las gestiones realizadas en el proceso de cobranza en terreno.

El sistema registrará cualquier operación, tanto de cliente como de la gestión. Es decir, operaciones de actualización de datos a los abonados, registrar la gestión realizada por el personal de terreno. Además, el sistema permitirá guardar la numeración de recibos y facturas pre impresas para cada abonado gestionado.

### 4.1 Datos de la solución

Para dicho registro se necesitaran datos como los siguientes:

#### 4.1.1 Abonado

Para poder realizar la gestión de cobro, se necesitan saber los datos de la persona o empresa a la cual le pertenece el contrato de servicios de audio y video por suscripción. Los datos necesarios para la gestión son:

- **Identificación de la persona (cédula de identidad. No aplica si es empresa):** Corresponde a la cédula de identidad del abonado. Tipo de dato: cadena de caracteres. Longitud fija 10 caracteres.
- **RUC:** Corresponde al registro único del contribuyente. Tipo de dato: cadena de caracteres. Longitud fija 13 caracteres.

- **Nombres del abonado:** Se refiere a los nombres del abonado. Tipo de dato: cadena de caracteres. Longitud variable 50 caracteres.
- **Apellidos del abonado (No aplica si es empresa).** Se refiere a los apellidos del abonado. Tipo de dato: cadena de caracteres. Longitud variable 50 caracteres.
- **Estado civil (No aplica si es empresa).** Se refiere al estado civil del abonado. Hay cinco tipos: soltero, casado, viudo, divorciado, unión libre. Tipo de dato: carácter (se especifica por la primera letra de estado civil que corresponde).
- **Nombre del contacto del abonado.** Se refiere al nombre del contacto del abonado. Tipo de dato: cadena de caracteres. Longitud variable 50 caracteres.
- **Apellido del contacto del abonado:** Se refiere al apellido del contacto del abonado. Tipo de dato: cadena de caracteres. Longitud variable 50 caracteres.
- **Parentesco del contacto del abonado:** Se refiere al parentesco del contacto del abonado. Son de tres tipos y llegan hasta el tercer grado de consanguinidad. Estos son: padres, hermanos, primos. Tipo de dato: cadena de caracteres. Longitud variable 10 caracteres.
- **Dirección de correo electrónico del abonado:** Se refiere a la dirección del correo electrónico del abonado. Tipo de dato: cadena de caracteres. Longitud variable 100 caracteres.

#### 4.1.2 Dirección

Para la gestión de cobranza de terreno se necesitan los siguientes datos de la dirección del abonado:

- **Identificación de la dirección:** Identificación única de la dirección. Tipo de dato: Entero.
- **Tipo de dirección:** son de dos tipos. Trabajo o residencia. Tipo de dato: cadena de caracteres. Longitud variable 10 caracteres.
- **Ciudad:** Se refiere a la ciudad de residencia del abonado. Tipo de dato: cadena de caracteres. Longitud variable 50 caracteres.
- **Dirección física del abonado:** Es la dirección completa que registró el abonado. Tipo de dato: cadena de caracteres. Longitud variable 100 caracteres.
- **Sector:** Es el sector establecido por Cablexpress Macas. Tipo de dato: cadena de caracteres. Longitud variable 50 caracteres.
- **Latitud:** Es la coordenada de GPS de la latitud de la dirección del abonado. Tipo de dato: real.
- **Longitud:** Es la coordenada de GPS de la longitud de la dirección del abonado. Tipo de dato: real.
- **Imagen:** Se refiere a la ubicación de la fotografía del predio físico. Tipo de dato: cadena de caracteres. Longitud variable 200 caracteres.

### 4.1.3 Teléfono

Para la gestión de cobranza de terreno se necesitan los siguientes datos de teléfono del abonado:

- **Identificación del teléfono:** Identificación única del teléfono. Tipo de dato: Entero.
- **Categoría:** Existen cuatro categorías. Residencial, comercial, popular y celular. Tipo de dato: cadena de caracteres. Longitud variable 15 caracteres.
- **Número:** Número de teléfono completo del abonado. Tipo de dato: cadena de caracteres. Longitud variable 10 caracteres.

### 4.1.4 Contratos

Aquí se encuentran los datos de los contratos de servicio de audio y video por suscripción que los abonados poseen una vez que han adquirido el servicio. Un abonado puede tener varios contratos de prestación de servicios. Pero un contrato, visto desde la parte administrativa, solo puede estar asignado a un abonado.

Los datos necesarios para la gestión de cobranza del personal de terreno son los siguientes:

- **Identificación o número del contrato:** Identificación única del contrato creado por Xanda Linkworks. Tipo de dato: entero.
- **Número de televisores contratados:** Es el número de televisores para los que el cliente contrató el servicio. Tipo de dato: entero.
- **Valor a pagar del contrato:** Valor a pagar por el servicio. Tipo de dato: entero.

- **Estatus del contrato:** Es el estado del contrato. En este campo se especifican dos valores: activo (A), para los clientes que mantienen el servicio activo aunque se encuentren en mora. Inactivo (I), son los clientes a los que se les canceló el servicio. Tipo de dato: carácter.
- **Fecha de inicio del contrato:** Fecha de inicio del contrato. Tipo de dato: fecha.
- **Fecha de finalización del contrato:** Fecha de finalización del contrato. Tipo de dato: fecha.

#### 4.1.5 Tipo de contrato

Dentro de la gestión de cobranza se debe saber qué tipo de contrato tiene el abonado. Para esto, se ha realizado la categorización de contratos de acuerdo a la siguiente la tabla 2:

**Tabla 2. Tipos de contratos de prestación de servicio de audio y video por suscripción de la empresa Cablexpress Macas**

Tipo de contrato	Valor estándar contrato	Valor de TV extra
HOGAR	12	1
NEGOCIO	14	2
HOTEL	65	2
HOSTAL	45	2

Los datos presentados en la tabla anterior fueron enviados por la empresa Cablexpress Macas, que es cliente de Xanda Linkworks.

A continuación se va a explicar el significado de cada uno de las categorías en las que se clasificó la información:

- **Tipo de Contrato.** Se refiere al lugar donde se instala el servicio. Este puede ser: hogar, negocio, hotel y hostel.

- **Valor estándar del contrato.** Es la pensión básica que el abonado paga por el servicio que recibe.
- **Valor de televisión extra.** Es costo que el abonado debe cancelar por cada televisión extra en la que requiere el servicio.

Los datos necesarios para la gestión de cobranza se especifican a continuación:

- **Identificación del tipo de contrato:** Identificador único creado por Xanda Linkworks. Tipo de dato: entero.
- **Tipo de contrato:** Se refiere al lugar donde se instala el servicio. Este puede ser: hogar, negocio, hotel y hostal. Tipo de dato: cadena de caracteres. Extensión variable de 10 caracteres.
- **Valor estándar del contrato:** Es la pensión básica que el abonado paga por el servicio que recibe. Tipo de dato: flotante.
- **Valor de televisión extra:** Es costo que el abonado debe cancelar por cada televisión extra en la que requiere el servicio. Tipo de dato: flotante.

#### 4.1.7 Historial de mora

Para el proceso de gestión de cobranza se necesita el historial de mora de los abonados, el cual consta de los siguientes datos:

- **Identificación del historial:** Identificador único creado por Xanda Linkworks. Tipo de dato: entero.
- **Mes de mora:** Se refiere al mes en el que se encuentra pendiente el pago. Tipo de dato: fecha.

- **Valor de la mora:** Se refiere al valor de mora del abonado. Tipo de dato: flotante.
- **Estatus de la mora:** Estado de actividad o inactividad de la mora. Tipo de dato: carácter. Pueden existir dos estados: Activo (A) e inactivo (I).

#### 4.1.6 Gestiones

Para el proceso de gestión de cobranza se requieren los siguientes datos con el fin de registrar la gestión realizada por el agente de cobranza.

- **Identificación de la gestión:** Identificador único creado por Xanda Linkworks. Tipo de dato: entero.
- **Fecha de la gestión:** fecha en la que se realizó la gestión de cobranza. Tipo de dato: fecha.
- **Valor pagado de la gestión:** Valor pagado por el abonado. Tipo de dato: flotante.
- **Saldo de la gestión:** Saldo pendiente del valor a cancelar por parte del abonado. Tipo de dato: flotante.
- **Número de factura:** Número de la factura entregada al abonado. Tipo de dato: cadena de caracteres. Longitud variable 100.
- **Número de recibo:** Número del recibo entregado al abonado. Tipo de dato: cadena de caracteres. Longitud variable 100.
- **Mes gestión mora:** Se refiere al mes del cual el abonado está pagando su saldo de mora. Tipo de dato: entero.

Cada gestión realizada contiene estos datos. Sin embargo, el tipo de gestión viene de otro repositorio de datos.

#### 4.1.7 Tipos de gestión

Para el proceso de gestión, se han establecido diferentes tipos como se puede ver en la tabla 3.

**Tabla 3. Tipos de gestiones de cobranza.**

Tipo de gestión	Descripción
Pago en oficina	Pago de servicio en oficina
Pago en domicilio	Pago de servicio en domicilio
Visita con notificación	Visita domiciliaria con notificación
Negativa de pago	Negativa de pago
Pago parcial en oficina	Pago parcial en oficina
Pago parcial en domicilio	Pago parcial en domicilio

Para el proceso de cobranzas se requieren los siguientes datos:

- **Identificador de gestión:** Identificador único creado por Xanda Linkworks. Tipo de dato: entero.
- **Tipo de gestión:** Tipo de gestión que realiza el agente de cobranza. Tipo de dato: cadena de caracteres: variable. Extensión máxima 10 caracteres.
- **Descripción de la gestión:** Descripción breve de la gestión. Tipo de dato: cadena de caracteres: variable. Extensión máxima 50 caracteres.
- **Estatus de la gestión:** Estado de actividad o inactividad de la gestión. Tipo de dato: carácter. Pueden existir dos estados: Activo (A) e inactivo (I).

Estos datos serán empleados en el registro del tipo de gestión que realiza el agente.

#### 4.1.8 Usuarios

Para realizar el proceso de cobranza, cada gestor se debe identificar. Para hacerlo, el personal de sistemas de Xanda Linkworks crea los usuarios cumpliendo los siguientes parámetros.

- **Identificación del usuario:** En este campo se encuentra la información de identificación del usuario. Se trata de un código creado por Xanda Linkworks. Tipo de dato: cadena de caracteres: variable. Extensión máxima 50 caracteres.
- **Nombre de usuario:** Es el nombre de pila de usuario. Tipo de dato:
  - cadena de caracteres: variable. Extensión máxima 50 caracteres.
- **Apellido de usuario:** Es el apellido del usuario. Tipo de dato: cadena de caracteres: variable. Extensión máxima 50 caracteres.
- **Nombre de usuario para la aplicación:** Es el nombre de usuario que se utilizará para registrarse en la aplicación. Tipo de dato: cadena de caracteres: variable. Extensión máxima 20 caracteres.
- **Contraseña:** Es la contraseña personal de cada usuario. Tipo de dato: cadena de caracteres: variable. Extensión máxima 32 caracteres.
- **Estatus:** Determina si el usuario está activo o inactivo. Tipo de dato: carácter.

Cada usuario tiene estos datos. Sin embargo, el tipo de usuario viene de otra tabla que contiene estos de datos.

#### 4.1.9 Tipos de Usuarios

Para el proceso de gestión, se han establecido diferentes tipos de usuarios que cumplen diferentes roles. Estos son:

- **Administrador.** Este rol permite visualizar y gestionar toda la información en el sistema.
- **Usuario.** Este rol permite visualizar y gestionar la información asignada gestor.

Para el proceso de cobranzas se requieren los siguientes datos:

- **Identificación del tipo de usuario:** Identificador único creado por Xanda Linkworks. Tipo de dato: Entero.
- **Tipo de usuario:** En este campo se encuentra la información del tipo de usuario. Aquí se especifica cuál es el rol del usuario en la solución. Tipo de dato: cadena de caracteres: variable. Extensión máxima 10 caracteres.
- **Descripción del usuario:** En este campo se encuentra una breve descripción del usuario. Tipo de dato: cadena de caracteres: variable. Extensión máxima 50 caracteres.
- **Estatus del usuario:** Determina si el usuario está activo o inactivo. Tipo de dato: carácter.

#### 4.1.10 Asignación

Para el proceso de cobranza se debe asignar a cada usuario su carga de trabajo diario. Para esto se necesita un repositorio de datos en el que se especifican los siguientes parámetros:

- **Identificador de asignación:** Identificador único creado por Xanda Linkworks. Tipo de dato: entero.
- **Fecha de asignación:** Fecha en la cual se asignó el abonado el gestor. Tipo de dato: fecha.
- **Estado de la asignación:** Estado de actividad o inactividad de la asignación. Existen dos estados: activo e inactivo. Tipo de dato: carácter.

El estado activo es aquel que determina que el abonado no ha cumplido con sus pagos o lo ha hecho parcialmente.

El estado inactivo se da cuando el abonado canceló, en su totalidad, el servicio en el mes solicitado o en el mes de mora.

A continuación se enumerará cada tabla y sus propiedades:

### Tabla persona

En la figura14 se muestran las columnas y las claves que contiene la tabla persona.

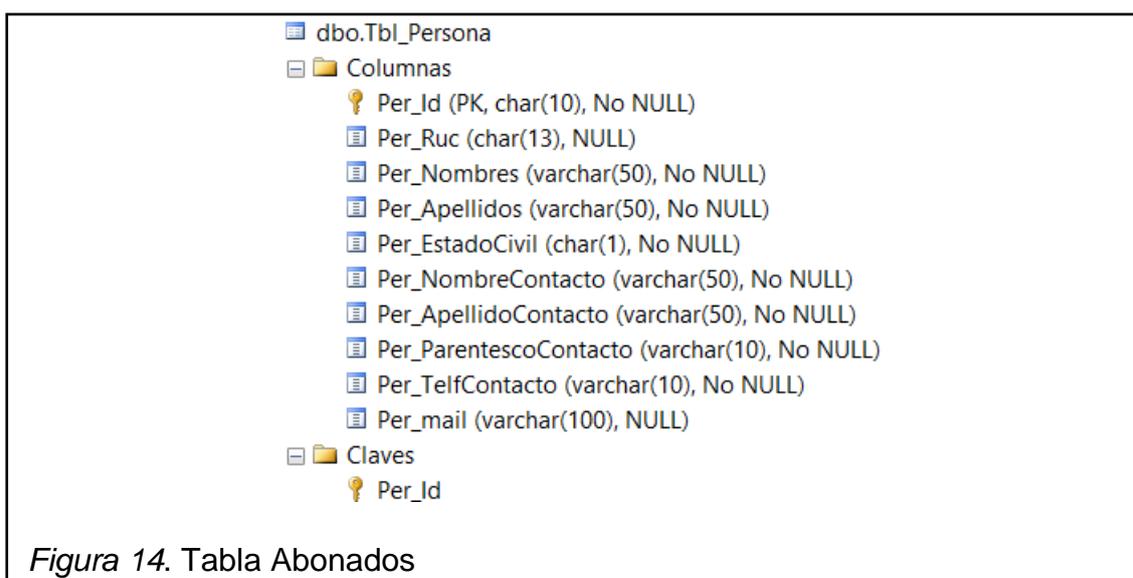


Figura 14. Tabla Abonados

### Tabla Asignación

En la figura15 se muestran las columnas y las claves que contiene la tabla asignación.



### Tabla contratos

En la figura 16 se muestran las columnas y las claves que contiene la tabla contratos.



### Tabla Dirección

En la figura 17 se muestran las columnas y las claves que contiene la tabla dirección.



### Tabla Gestiones

En la figura 18 se muestran las columnas y las claves que contiene la tabla gestiones.



### Tabla Historial mora

En la figura19 se muestran las columnas y las claves que contiene la tabla historial mora.

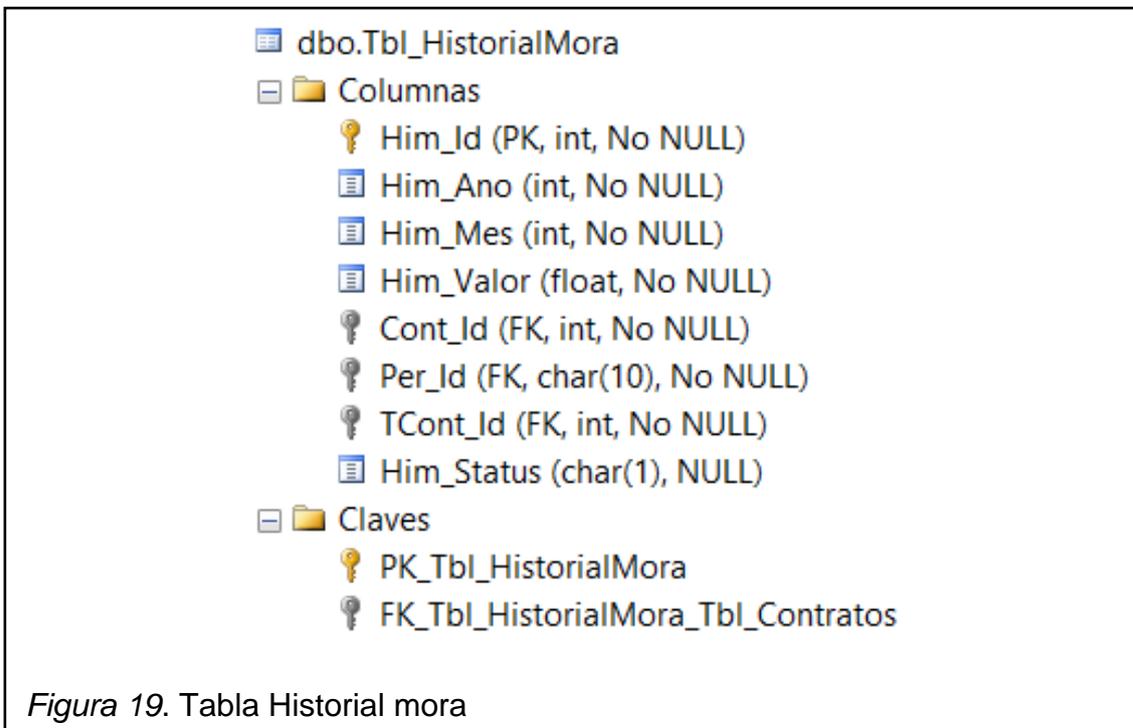


Figura 19. Tabla Historial mora

### Tabla Teléfono

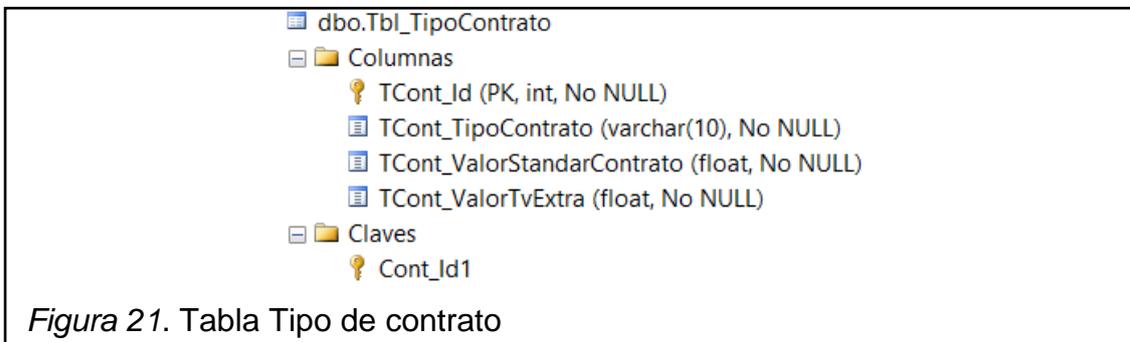
En la figura20 se muestran las columnas y las claves que contiene la tabla teléfono.



Figura 20. Tabla Teléfono

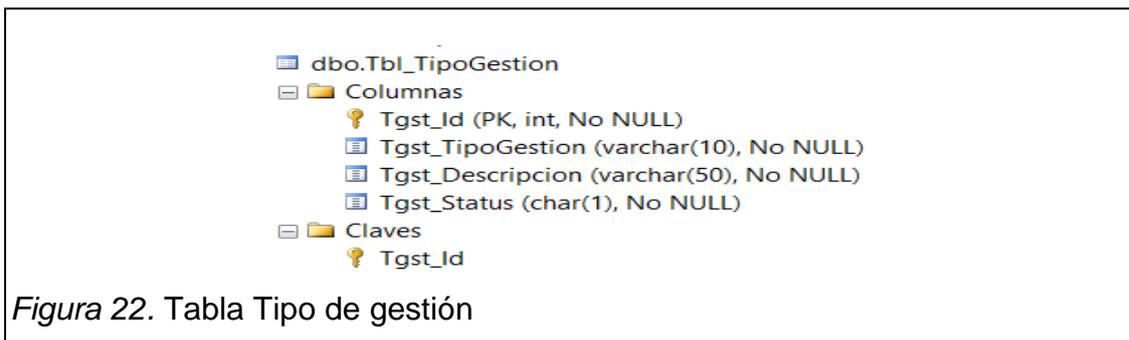
### Tabla Tipo de contrato

En la figura21 se muestran las columnas y las claves que contiene la tabla Tipo de contrato.



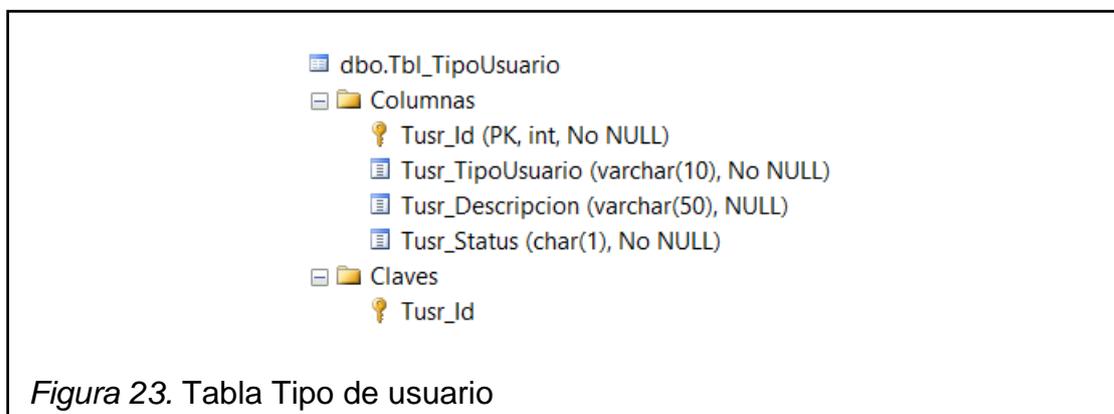
### Tabla Tipo de gestión

En la figura22 se muestran las columnas y las claves que contiene la tabla Tipo de gestión.



### Tabla Tipo de usuario

En la figura23 se muestran las columnas y las claves que contiene la tabla Tipo de usuario.



## Tabla Usuario

En la figura24 se muestran las columnas y las claves que contiene la tabla usuario.

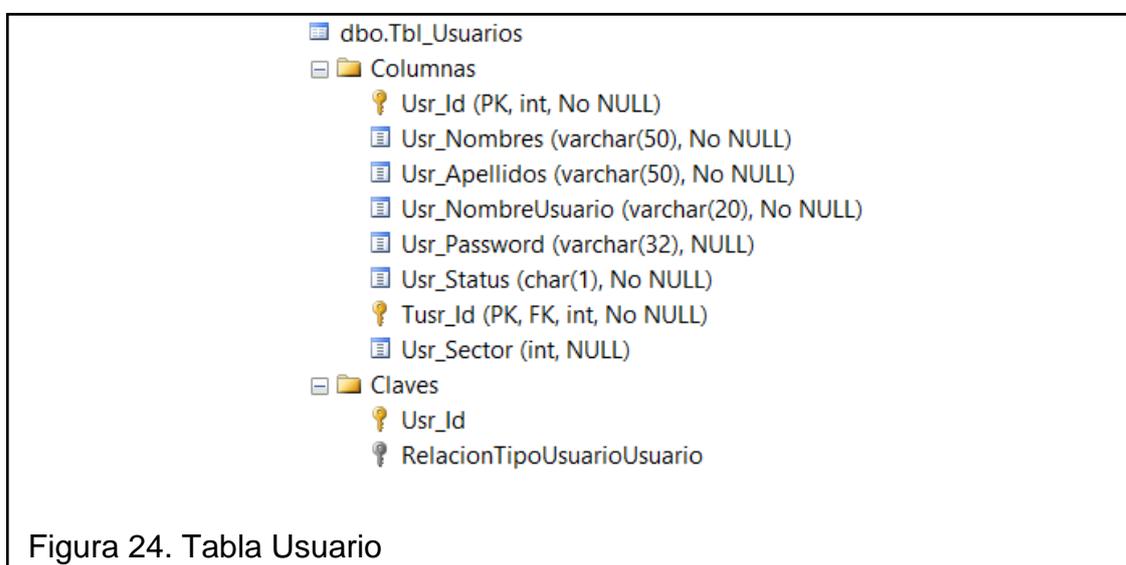


Figura 24. Tabla Usuario

Esta base de datos no se encuentra vacía. Los datos que fueron entregados se introducen en las tablas creadas.

Cablexpress Macas envía la información de sus clientes en un archivo de Excel. Toda esta información se normalizó, y se introdujo en las tablas de la base de datos para realizar la programación de las demás capas.

Además, debido a las relaciones entre las tablas se debe agregar datos ya que si están vacías se generaran errores al momento de crear más datos.

Las tablas que se llenaron con información necesaria para que la solución funcione son las siguientes:

- **Tbl\_Persona:** En esta tabla se insertaron los datos de todas las personas tanto naturales como jurídicas que maneja la empresa Cablexpress en la ciudad de Macas.
- **Tbl\_Telefonos:** En esta tabla se insertaron los datos de todos los

teléfonos que las personas tanto naturales como jurídicas registraron en la empresa Cablexpress en la ciudad de Macas.

- **Tbl\_Direccion:** En esta tabla se insertaron los datos de todas las direcciones donde da el servicio de CATV en la ciudad de Macas.
- **Tbl\_Contrato:** En esta tabla insertaron los datos de todos los tipos de contratos que maneja la empresa Cablexpress en la ciudad de Macas.
- **Tbl\_TipoContrato:** En esta tabla insertaron los datos de todos los contratos que maneja la empresa Cablexpress en la ciudad de Macas.
- **Tbl\_HistorialMora:** En esta tabla insertaron los datos del historial de mora de los clientes que maneja la empresa Cablexpress en la ciudad de Macas.
- **Tbl\_Usuarios:** En esta tabla insertaron los usuarios que usaran la aplicación.
- **Tbl\_TipoUsuario:** En esta tabla insertaron los tipos de usuarios que usaran la aplicación.
- **Tbl\_Asignacion:** En esta tabla insertó la asignación de trabajo para los usuarios de la aplicación.
- **Tbl\_TipoGestion:** En esta tabla insertaron los tipos de gestiones que realizan los gestores de cobranzas de terreno.

La tabla Tbl\_Gestiones no lleva ningún campo lleno ya que en esta de van a agregar los datos de las gestiones por parte de los usuarios de la aplicación.

## 4.2 Diagrama relacional de la solución y tablas que la componen

En la figura 25 se muestra el diagrama relacional de los datos modelados.

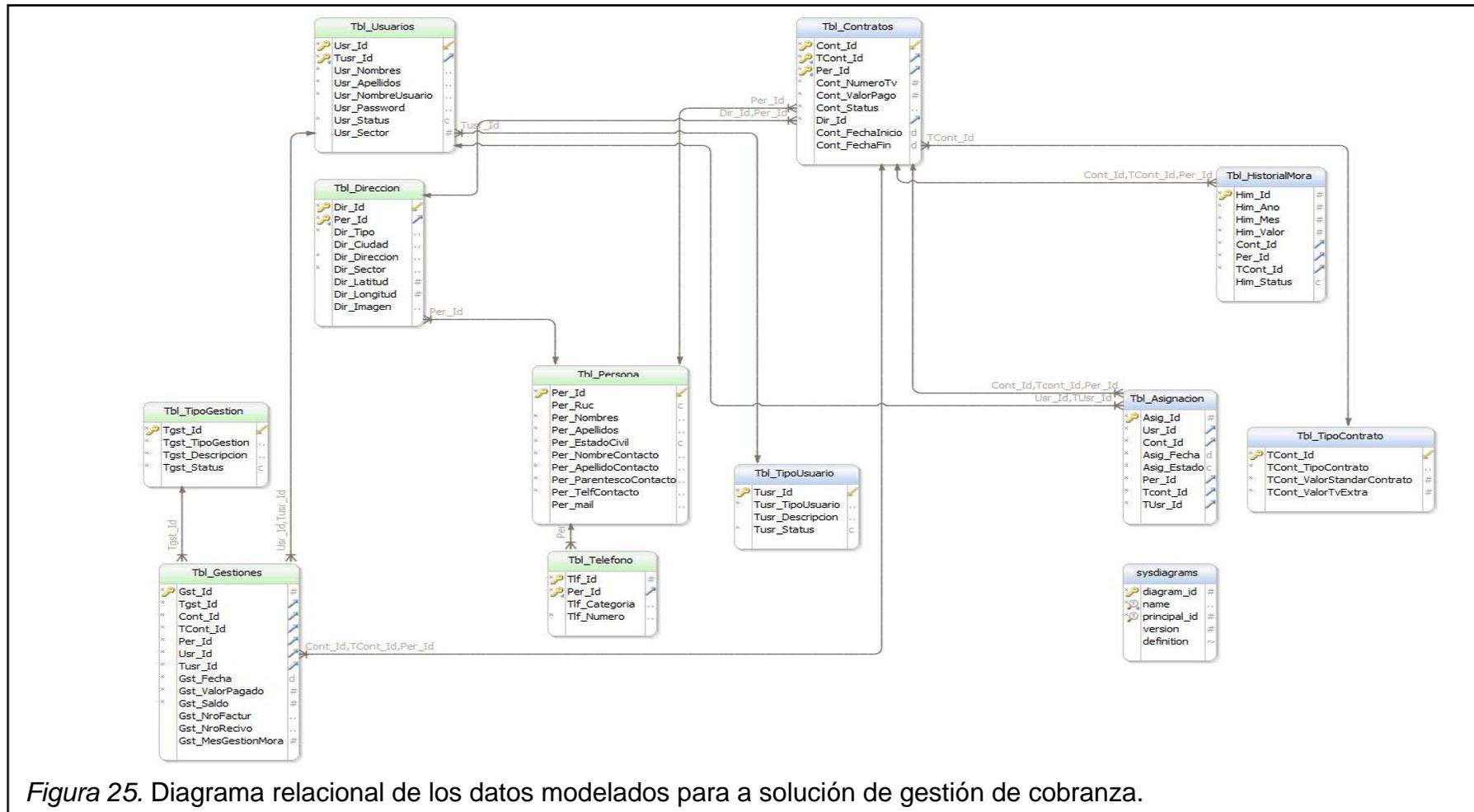


Figura 25. Diagrama relacional de los datos modelados para a solución de gestión de cobranza.

A continuación en la tabla 4 se detallara cada tabla y como cada rol de usuario podrá acceder a cada una y como se validaran los datos de entrada a cada uno

**Tabla 4. Permisos concedidos a los usuarios en las tablas de base de la base de datos.**

TABLAS	ROL	PERMISOS	VALIDACIONES
Tbl_Persona	USUARIO	LECTURA	N/A
	ADMINISTRADOR	LECTURA	N/A
Tbl_Telefonos	USUARIO	LECTURA	N/A
	ADMINISTRADOR	LECTURA	N/A
Tbl_Direccion	USUARIO	LECTURA Y ESCRITURA	Numero de caracteres permitido para el campo dirección de 100.
	ADMINISTRADOR	LECTURA Y ESCRITURA	Numero de caracteres permitido para el campo dirección de 100.
Tbl_Contrato	USUARIO	LECTURA	N/A
	ADMINISTRADOR	LECTURA	N/A
Tbl_TipoContrato	USUARIO	LECTURA	N/A
	ADMINISTRADOR	LECTURA	N/A
Tbl_TipoContratos	USUARIO	LECTURA	N/A
	ADMINISTRADOR	LECTURA	N/A
Tbl_HistorialMora	USUARIO	LECTURA	N/A
	ADMINISTRADOR	LECTURA	N/A
Tbl_Usuarios	USUARIO	LECTURA	N/A
	ADMINISTRADOR	LECTURA	N/A
Tbl_TipoUsuario	USUARIO	LECTURA	N/A
	ADMINISTRADOR	LECTURA	N/A
Tbl_Asignacion	USUARIO	LECTURA	N/A
	ADMINISTRADOR	LECTURA Y ESCRITURA	Solo caracteres numéricos en el campo de Usr_Id.
Tbl_TipoGestion	USUARIO	LECTURA	N/A
	ADMINISTRADOR	LECTURA	N/A
Tbl_Gestiones	USUARIO	LECTURA Y ESCRITURA	Gst_fecha: Esta echa debe estar dentro del año en curso. Gst_ValorPagado: Este campo solo debe contener caracteres numéricos. Gst_Saldo: Este campo solo debe contener caracteres numéricos. Gst_NroFactur: Este campo solo debe contener caracteres numéricos. Gst_Nro Recivo: Este campo solo debe contener caracteres numéricos Gst_MesGestionMora: Este campo solo debe contener caracteres numéricos
	ADMINISTRADOR	LECTURA Y ESCRITURA	Gst_fecha: Esta echa debe estar dentro del año en curso. Gst_ValorPagado: Este campo solo debe contener caracteres numéricos. Gst_Saldo: Este campo solo debe contener caracteres numéricos. Gst_NroFactur: Este campo solo debe contener caracteres numéricos. Gst_NroRecivo: Este campo solo debe contener caracteres numéricos Gst_MesGestionMora: Este campo solo debe contener caracteres numéricos

## GENERACIÓN DE LA APLICACIÓN

El objetivo de este capítulo es generar el prototipo en base a lo diseñado en los capítulos anteriores. Para lograr esto se usara en patrón de diseño Modelo, vista, controlador (MVC).

El patrón MVC posee 3 capas que son el modelo, la vista y el controlador. A continuación de detallar el desarrollo de cada una de las capas.

### 5.1 Capa del modelo

La capa del modelo aloja las clases que se conectan a la base de datos. Esta base de datos usa el motor de base de datos Microsoft SQL Server en su edición Express. En el anexo A se detallará el código fuente que fue necesario para la creación de la base de datos de la solución .La base de datos como tal se encontraba vacía, por lo cual, se procedió a volcar los dato necesarios para su correcta operación.

Para la codificación de la capa del modelo nos apoyamos del lenguaje de programación PHP para los servicios web y Java para el consumo de los mismos y el envío de la información proveniente de la capa del modelo donde se aloja la base de datos.

Para que la capa del controlador, que en este caso está desarrollada para el sistema operativo de dispositivos móviles Android se necesita de la ayuda de un servicio web. De esta manera PHP, al ser un lenguaje de programación para que soporte la creación de servicios web, es la herramienta perfecta para este fin.

La tecnología que se usó para serializar los datos que provienen de la capa de base de datos es Java Script Object Notation (JSON) por sus siglas en ingles. A continuación se dará una breve explicación de la tecnología JSON.

- **JSON (Java Script Object Notation )**

De acuerdo con la página web [www.json.org](http://www.json.org) dice:

**JSON** (Java Script Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación Java Script, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocidos como un *objeto*, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

En JSON, se presentan de estas formas:

Un *objeto* es un conjunto desordenado de pares nombre/valor. Un objeto comienza con (llave de apertura) y termine con (llave de cierre). Cada nombre es seguido por: (dos puntos) y los pares nombre/valor están separados por, (coma). (Json.org, 2014)

Un ejemplo simple de lo citado anteriormente es el siguiente:

El conjunto de datos de un usuario consta de los siguientes valores:

- Usr\_id: 5
- Usr\_Nombres: Marlon Alberto
- Usr\_Apellidos : Granda Lopez
- Usr\_NombreUsuario : mgrandal
- Tusr\_Id : 2

Estos datos convertidos a notación JSON se verán de la siguiente manera:

```
["Usr_id": "5", "Usr_Nombres": "Marlon Alberto", "Usr_Apellidos": "Granda Lopez", "Usr_NombreUsuario": "mgrandal", "Tusr_Id": "2"]
```

El lenguaje PHP puede gestionar JSON de manera nativa y lo hace con la siguiente sintaxis:

```
echo json_encode($myArray);
```

Donde myArray es un arreglo de datos generado en PHP. Este arreglo puede venir de cualquier fuente de datos ya sea una base de datos, etc.

Finalmente la conexión desde la capa de negocios hacia la base de datos se lo realiza con una herramienta que permite este cometido y es la Extensión Objetos de Datos de PHP (PDO) por sus siglas en ingles. A continuación se detallara brevemente el que es PDO.

- **Extensión Objetos de Datos de PHP (PDO)**

De acuerdo con el manual en línea de PHP en su sección de PDO dice:

La extensión Objetos de Datos de PHP (PDO por sus siglas (PHP.ORG) en inglés) define una interfaz ligera para poder acceder a bases de datos en PHP. Cada controlador de bases de datos que implemente la interfaz PDO puede exponer características específicas de la base de datos, como las funciones habituales de la extensión. Obsérvese que no se puede realizar

ninguna de las funciones de la bases de datos utilizando la extensión PDO por sí misma; se debe utilizar un controlador de PDO específico de la base de datos para tener acceso a un servidor de bases de datos.

PDO proporciona una capa de abstracción de acceso a datos, lo que significa que, independientemente de la base de datos que se esté utilizando, se usan las mismas funciones para realizar consultas y obtener datos. PDO no proporciona una abstracción de bases de datos; no reescribe SQL ni emula características ausentes. Se debería usar una capa de abstracción totalmente desarrollada si fuera necesaria tal capacidad.

PDO viene con PHP 5.1, y está disponible como una extensión PECL para PHP 5.0; PDO requiere las características nuevas de OO del núcleo de PHP 5, por lo que no se ejecutará con versiones anteriores de PHP. (PHP.ORG)

De acuerdo al texto citado anteriormente PDO permite una conexión a cualquier base de datos, independientemente del motor que se esté usando. Por esta razón se usa este método de conexión para traer los datos desde la base de datos hacia la capa de negocios.

Un ejemplo de uso de PDO en PHP es el siguiente:

```
<?php
/* Ejecutar una sentencia preparada vinculando variables de PHP */
$calorías = 150;
$color = 'red';
$gsent = $gbd->prepare ('SELECT name, colour, calories
    FROM fruit
    WHERE calories <? AND colour =?');
$gsent->bindParam (1, $calorías, PDO::PARAM_INT);
$gsent->bindParam (2, $color, PDO::PARAM_STR, 12);
$gsent->execute ();
?>
```

La librería PDO no viene de manera nativa en sistema de servicio web PHP así que debe ser instalada para su posterior uso.

Finalmente para poder publicar la capa de negocios como servicio web usando PHP se debe levantar un servidor web. Para este fin se usó un equipo con Windows 8 al cual se instaló el programa WAMP que permite tener un servidor web que soporta PHP. A este servidor se instalaron las librerías PDO para la conexión a la base de datos.

Para que el servicio web esté disponible en el internet se usó un DNS dinámico con la siguiente dirección: [www.antagus1984.no-ip.org](http://www.antagus1984.no-ip.org). Además para que el servidor pueda ser visto desde el internet se realizó un NAT estático y un PAT del puerto 8081 en el que el servidor web aloja los servicios web al puerto 80 de cara al internet.

En el anexo B se detallará un ejemplo de código PHP que se usó para el desarrollo de la capa del modelo. Este código recupera datos de la base de datos con una consulta en lenguaje SQL y los publica como servicio web en formato JSON.

## **5.2 Capa del Controlador**

Para que la capa de la vista que está programada en Java para Android se comunique con el controlador y así, llegue a la base de datos para interactuar con la información, debe consumir el servicio web que el negocio provee por medio de PHP.

El programa debe conectarse al servidor web para consumir los servicios web requeridos para la gestión de los datos. Para esto se usó dos clases que permiten realizar esta tarea. Estas clases son `HttpHandler.java` y `JsonParser.java`. El anexo C muestra un ejemplo del uso de estas 2 clases.

- **HttpHandler.java**

Esta clase es la encargada de conectarse al servidor web para consumir los servicios web de envío y recepción de datos. Aquí se manejan todas las excepciones en caso de falta de conexión u otros problemas referentes a la comunicación. Cabe destacar que para que una clase se comuniquen con el internet, se debe dar permisos a la aplicación para que se conecte a una red ya sea privada o pública.

Para dar permisos de conexión a la aplicación se debe agregar la siguiente línea de código en archivo `AndroidManifest.xml` que se encuentra en el anexo D.

El siguiente paso es hacer que el sistema entienda la trama que llega desde el web servicio, para esto se usa la clase `JsonParser.java`. Aquí toma los datos que viene del servicio web que envía tramas codificadas en JSON y los lee para convertirlos en datos que puedan ser usados en la aplicación en forma de objetos.

Para cada servicio web, la clase `JsonParser.java` debe tener una codificación para entenderlos y un objeto donde depositar los datos, ya sea una matriz o una lista que se puede operar según la conveniencia del programador.

En el anexo D se muestra un ejemplo de la codificación de `JsonParser.java` usada en la aplicación móvil.

Finalmente para que los servicios web puedan ser usados correctamente, el sistema operativo requiere que todas las peticiones hacia la red se hagan en un hilo secundario del procesador. Esto se debe a que si la conexión falla u otro tipo de problema se suscita, el programa que corre en el hilo principal no se cuelga haciendo que el dispositivo deje de funcionar correctamente.

- **AsyncTask**

La clase que permite correr operaciones en el hilo secundario es AsyncTask. Este método realiza las funciones programadas en este en un hilo secundario. Gracias a esta funcionalidad podemos instanciar la clase `HttpHandler.java` y conectarnos a la red sin ningún peligro de cuelgues ni errores por conexión.

En el anexo E se muestra un ejemplo del uso de AsyncTask en el sistema.

- **Servicio de GPS**

Una de las herramientas especiales que se usó para el desarrollo de la aplicación es el API de GPS que posee el sistema operativo Android. Esta herramienta nos permite tomar datos de la posición geográfica del dispositivo y usarla posteriormente para diferentes fines.

La API de GPS se usó en el sistema de cobranzas con el fin de dar una mejor ubicación a las direcciones de los abonados a los cuales el gestor de cobranzas de terreno debe hallar. En la ciudad de Macas el sistema de direcciones que existe es inexacto y no esta normado, por esta razón, la ubicación geográfica es una herramienta que ayuda resolver este inconveniente de una manera referencial y confiable.

Esta API genera los datos de latitud y longitud que son enviados desde el GPS del celular. Posteriormente se envían a guardar en la base de datos a través del servicio web destinado para este fin.

Para que el sistema pueda acceder a los servicios de GPS del dispositivo móvil. De debe modificar el archivo `androidmanifest.xml` agregando la siguiente línea:

```
<Uses-permission  
android:name="android.permission.ACCESS_FINE_LOCATION"></uses-  
permission>
```

El código del anexo F muestra cómo se usó el api para activar el GPS y recoger los datos de localización geográfica para enviarlos a la base de datos a la tabla Tbl\_Dirección que es donde se requieren los mismos.

- **Servicio de Fotografía**

Para una mejor ubicación del predio donde se realiza la cobranza, el sistema contempla la visualización de una fotografía que permita encontrar la dirección. Además, si la dirección no tiene fotografía alguna, el usuario debe tener la posibilidad de subir la foto del predio del abonado y guárdala en el servidor web para que posteriormente sea visualizado.

- **Visualización de la fotografía**

Con el objeto de ayudar a la ubicación del abonado, la aplicación, además de mostrar la dirección del predio donde se da el servicio, muestra la fotografía del lugar. Esto permite que el agente de cobranzas tenga una ayuda al ubicar la dirección.

Para este fin usamos un servicio web al cual le enviamos la información de la ubicación de la foto y él nos responde enviando la foto.

El siguiente código muestra como el sistema llama a la fotografía para ser mostrada enviándole al servidor web la dirección de su ubicación.

```
URL imageUrl = null;
```

```
URLConnection conn = null;
```

```
try {
```

```
imageUrl = new URL("http://antagus1984.no-ip.org/images/1.jpg");
```

```
conn = (URLConnection) imageUrl.openConnection();
```

```
conn.connect();
```

```
BitmapFactory.Options options = new BitmapFactory.Options();
```

```
options.inSampleSize = 2; // el factor de escala a minimizar la imagen, siempre es potencia de 2
```

```
Bitmap imagen = Bitmap Factory. Deco de Stream (conn. getInputStream ()),
new Rect (0, 0, 0, 0), options);
img. setImage Bitmap (imagen); } catch (IOException e) {e. printStackTrace();
```

El usuario podrá visualizar toda la información de la dirección, incluyendo la fotografía del predio como se muestra en la figura 26.



*Figura 26.* Formulario gestión que despliega la fotografía del predio del abonado

- **Envío de fotografía**

Con el objeto de ayudar a la ubicación del abonado, la aplicación permite actualizar los datos de ubicación del predio donde el abonado contrato el servicio. Esto permite que el agente de cobranzas actualice los datos de dirección del abonado incluyendo la fotografía del predio.

El código del anexo G muestra como la aplicación toma la fotografía y la sube al servidor web. Este código detalla cómo el programa toma el control de la cámara de fotos del dispositivo para tomar una foto. Posteriormente usa el método AsyncTask para subir la foto tomada al servidor.

Finalmente la aplicación envía la ubicación del archivo de fotografía a la base

de datos para futuras consultas como se muestra en la figura 27.



### 5.3 Capa de la vista

Para el desarrollo de la capa de la vista se usó el lenguaje de programación Java con su entorno de programación Eclipse destinado a codificación de aplicaciones móviles para el sistema operativo Android. Este entorno de desarrollo es más conocido como Android Development Tools o ADT por sus siglas en inglés

El siguiente código corresponde a la capa de la vista Listatrabajoactivity.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:orientation="vertical" >
    <TextView
```

```

        android:id="@+id/TxvListaTrabajo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Lista de Trabajo"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textStyle="bold" />

```

En la figura 28 se muestra el formulario final de manera gráfica después de su codificación.

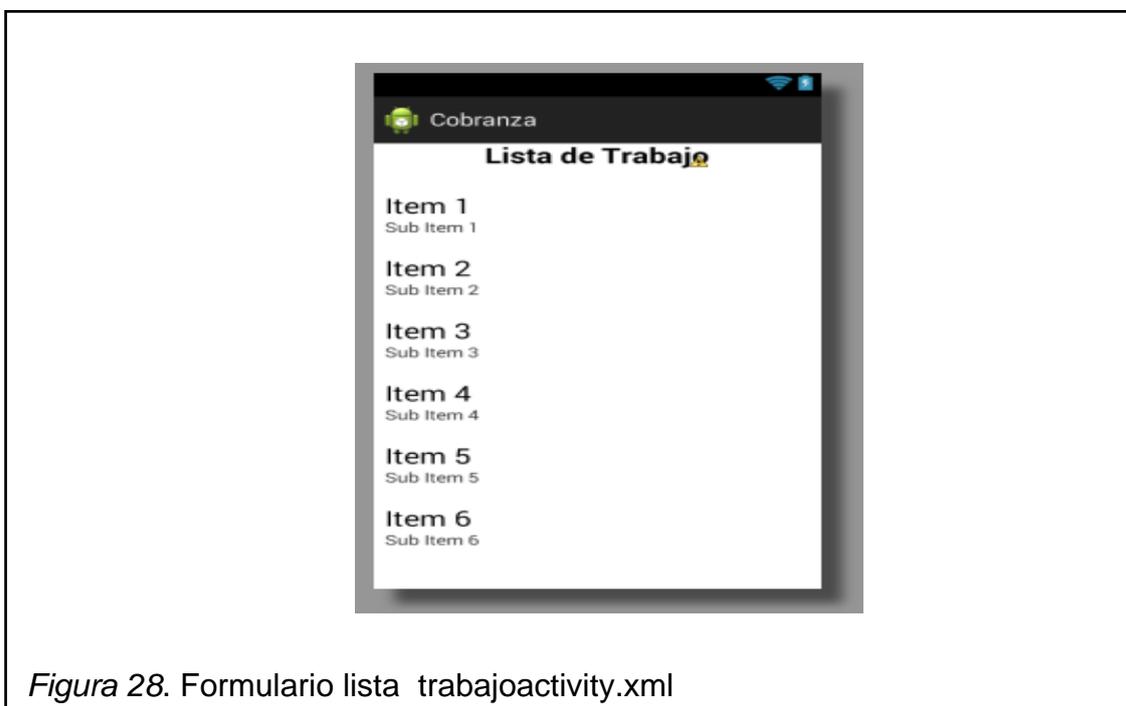


Figura 28. Formulario lista trabajoactivity.xml

A continuación se va a describir cada uno de los formularios que se han utilizado para el desarrollo de la capa de la vista.

- **Login**

Este formulario permite al usuario ingresar su nombre de usuario y contraseña para registrarse en la aplicación. Si los datos ingresados son correctos, el sistema avanzará al formulario menú principal. Caso contrario, el sistema muestra un error y se mantiene en el formulario como se muestra en la figura 29.

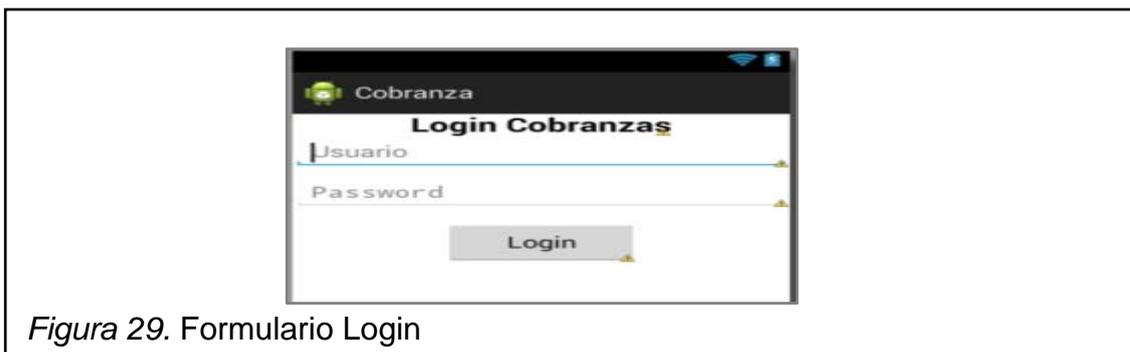


Figura 29. Formulario Login

- **Menú Principal**

El formulario de menú principal contiene las siguientes opciones:

- Lista de trabajo
- Búsqueda de abonados
- Información de gestión

Cada ítem que se despliega en esta lista puede ser seleccionado. Al realizar esta acción, el sistema despliega los formularios destinados: lista de trabajo, búsqueda de abonados o Información de gestión, dependiendo de la selección como se muestra en la figura 30.

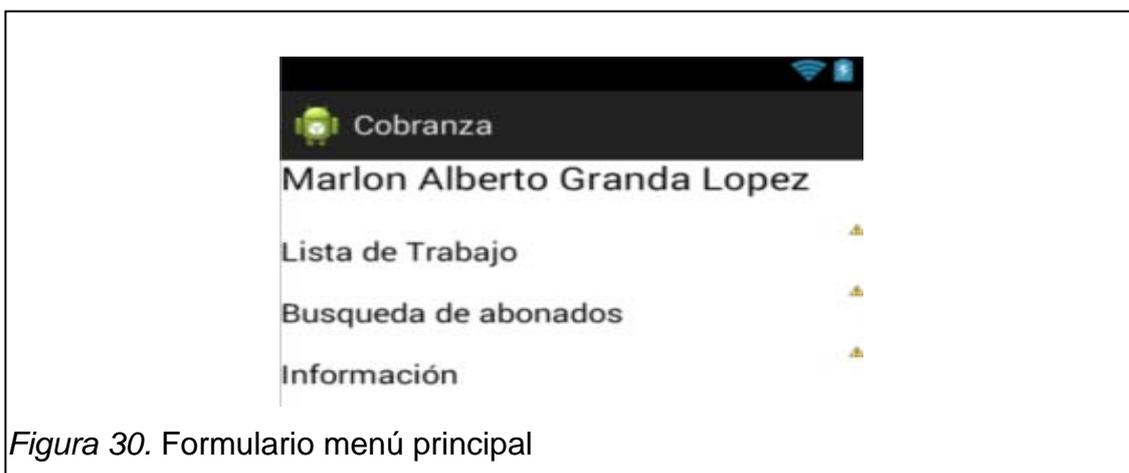


Figura 30. Formulario menú principal

- **Lista de trabajo**

Este formulario despliega la lista de trabajo asignada, que se genera dinámicamente, al usuario para que realice la gestión diaria.

La lista de trabajo consta de la siguiente información:

- Nombre del abonado
- Identificación

Cada ítem que se despliega en esta lista puede ser seleccionado. Al realizar esta acción, el sistema despliega el formulario de Gestión para el abonado seleccionado, como se muestra en la figura 31.

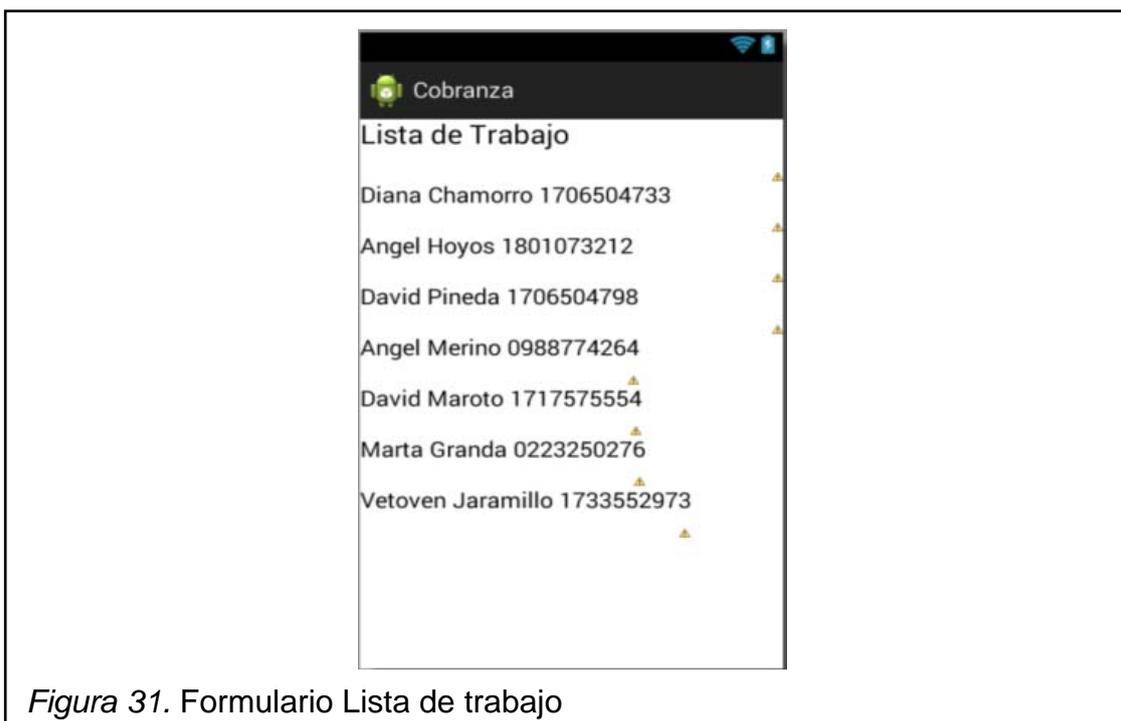


Figura 31. Formulario Lista de trabajo

- **Gestión**

Este formulario despliega los siguientes datos:

- Nombre del abonado
- Identificación
- Dirección
- Sector
- Latitud y longitud
- Fotografía del predio

Además el formulario contiene las siguientes opciones:

- Pago Completo
- Pago Parcial
- Negativa de pago
- Pago en oficina

Cada ítem que se despliega en esta lista puede ser seleccionado. Al realizar esta acción el sistema despliega los formularios: pago completo, pago parcial, negativo de pago o pago en oficina de acuerdo a la necesidad el usuario como se muestra en la figura 32.

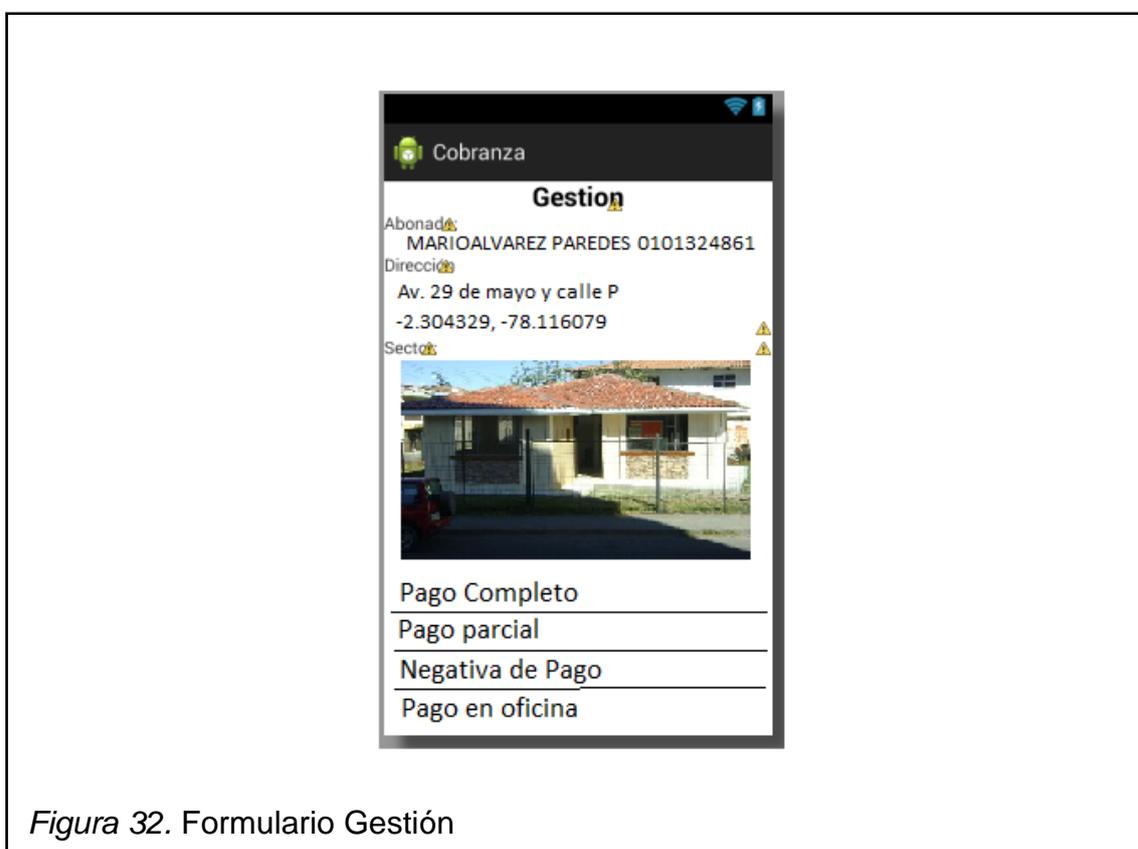


Figura 32. Formulario Gestión

- **Pago Completo**

Este formulario despliega la siguiente información:

- Valor total de pago
- Caja de texto de ingreso del documento que abala el pago

Además, el formulario mostrado en la figura 33 contiene un botón que al ser accionado permite guardar la gestión de pago. Esta acción muestra un mensaje de guardado y, posteriormente, el sistema regresa al formulario Lista de trabajo. También, este proceso actualiza el estado de la asignación de este abonado de activo a pasivo, por lo cual desaparece del formulario lista de trabajo.

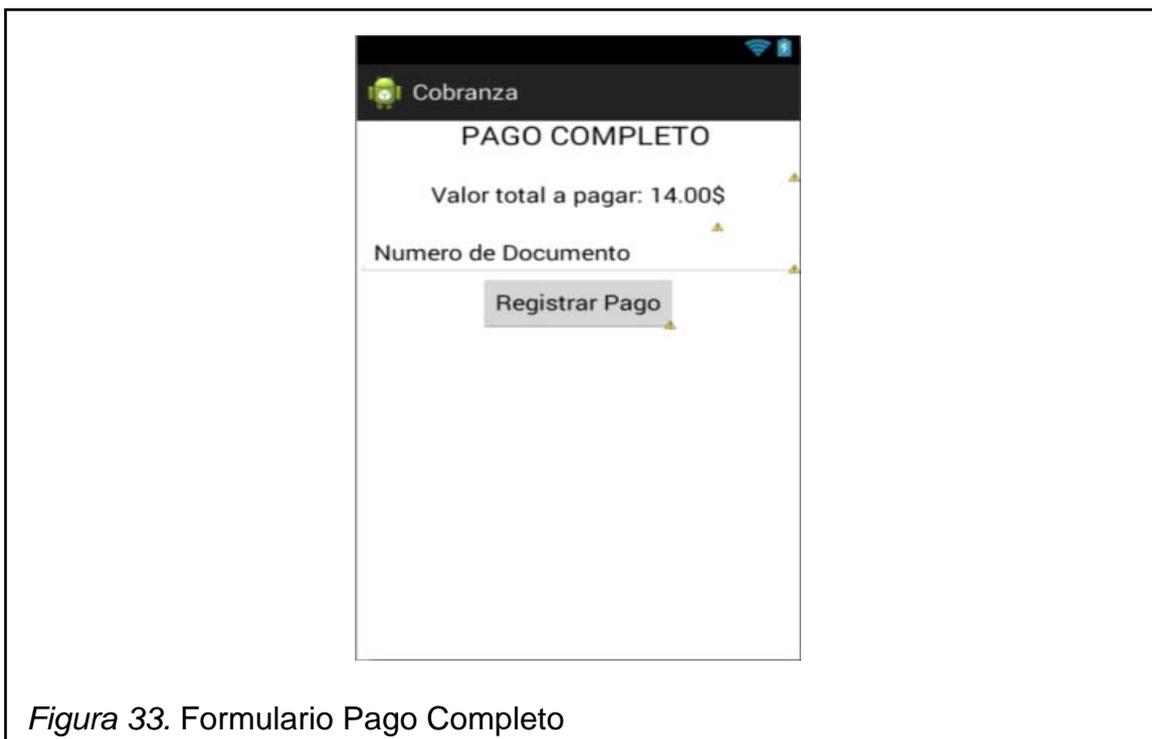


Figura 33. Formulario Pago Completo

- **Pago Parcial**

Este formulario despliega la siguiente información:

- Valor total de pago
- Caja de texto de ingreso del monto parcial del pago
- Caja de texto de ingreso del documento que abala monto parcial del pago

Para registrar el pago parcial este formulario posee un campo donde se puede ingresar la cantidad deseada. Cabe destacar que si el usuario ingresa una cantidad mayor o igual al valor total de pago, se producirá un error en el momento de guardar la gestión.

Además, el formulario mostrado en la figura 34 contiene un botón que al ser accionado permite guardar la gestión de pago. Esta acción muestra un mensaje de guardado y, posteriormente, el sistema regresa al formulario Lista de trabajo.

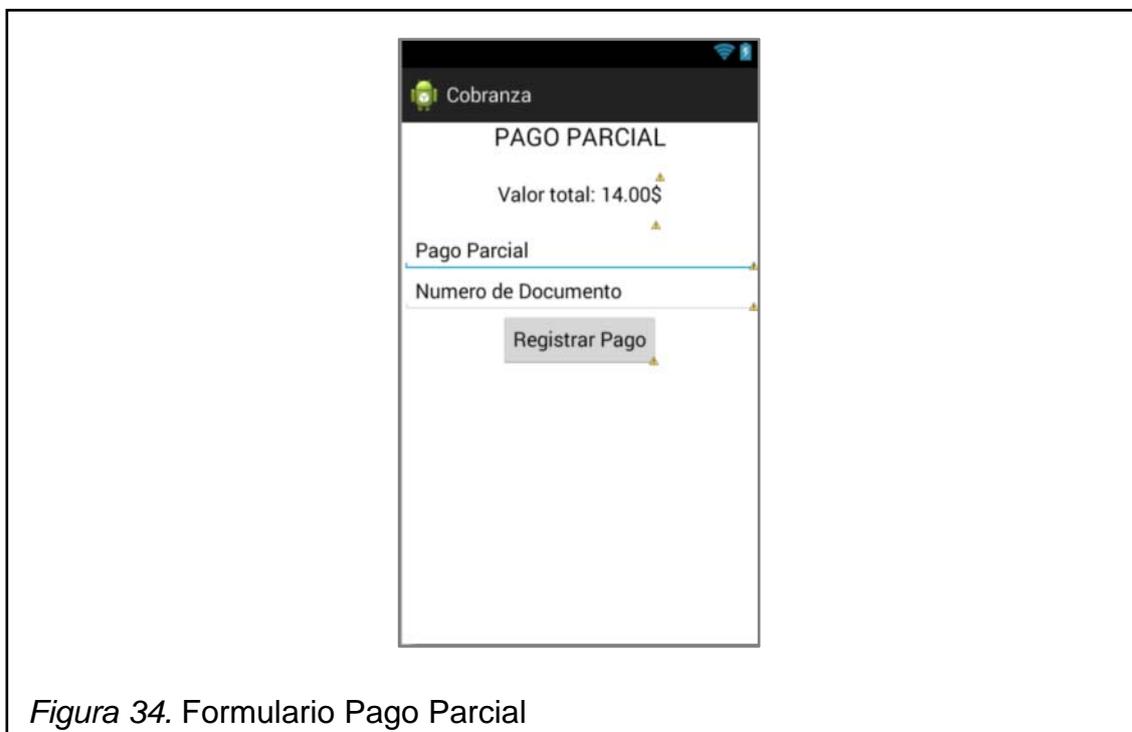


Figura 34. Formulario Pago Parcial

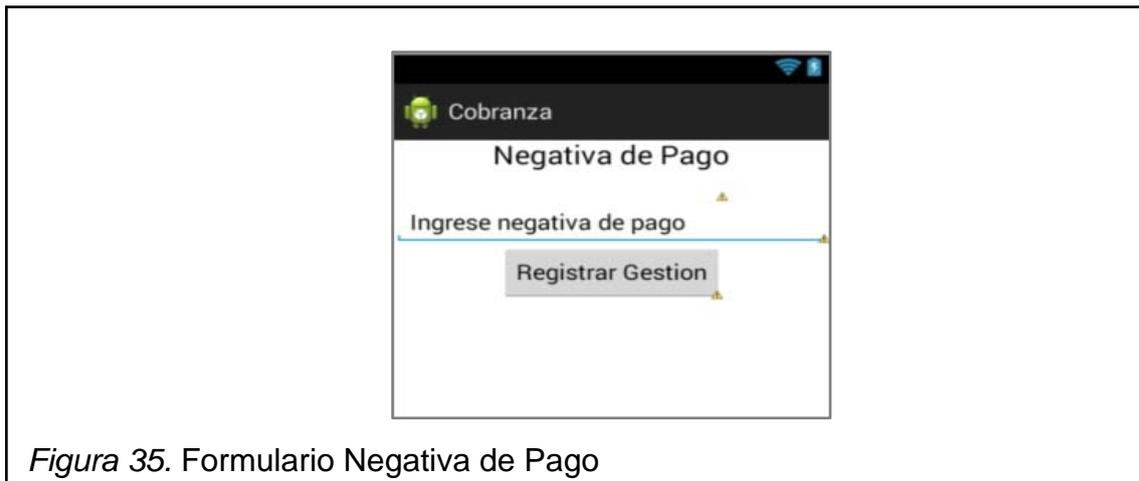
- **Negativa de pago**

Este formulario despliega la siguiente información:

- Caja de texto ingreso motivo de no pago

Para registrar la negativa de pago, este formulario posee un campo donde se puede ingresar una breve descripción del motivo de no pago. Cabe destacar que si el usuario no ingresa ningún carácter, en este campo producirá un error en el momento de guardar la gestión.

Además, el formulario mostrado en la figura 35 contiene un botón que al ser accionado permite guardar la gestión. Esta acción muestra un mensaje de guardado y, posteriormente, el sistema regresa al formulario Lista de trabajo.



*Figura 35. Formulario Negativa de Pago*

- **Pago en Oficina**

Este formulario despliega la siguiente información:

- Caja de texto de ingreso de valor registrado en el documento pagado e oficina.
- Caja de texto de ingreso del documento que abala monto pagado en oficina.

Este formulario mostrado en la figura 36 posee un campo que permite registrar el número de recibo o de factura de pago, que le ha sido entregada al gestor de cobranzas para comprobar el pago. Cabe destacar que si el usuario no ingresa ningún carácter, en estos campos producirá un error en el momento de guardar la gestión

Además, el formulario contiene un botón que al ser accionado permite guardar la gestión de pago. Esta acción muestra un mensaje de guardado y, posteriormente, el sistema regresa al formulario Lista de trabajo. Además, este proceso actualiza el estado de la asignación de este abonado de activo a pasivo, por lo cual desaparece del formulario lista de trabajo.



Figura 36. Formulario Pago en oficina

- **Búsqueda de Abonados**

Este formulario permite al usuario buscar un abonado usando su número de identificación.

El formulario contiene un campo para ingresar el número de identificación a buscar y un botón que envía la petición de búsqueda. Al accionar el botón, la aplicación despliega la siguiente información.

- Numero de contrato
- Nombre del cliente
- Identificación del cliente

Esta información desplegada puede ser seleccionada y, al realizar esta acción, se despliega el formulario Información de contrato como se muestra en la figura 37.



Figura 37. Formulario Búsqueda de Abonados

- **Información de contrato**

Este formulario despliega la siguiente información:

- Nombres del cliente
- Apellidos del cliente
- Número de identificación del cliente
- Número de identificación del contrato
- Dirección donde se presta el servicio
- Sector
- Imagen del predio
- Latitud del predio
- Longitud del predio

Además, este formulario tiene dos opciones que pueden seleccionarse y son:

- Actualizar dirección
- Consultar mora

Al realizar la acción de selección se despliegan los formularios: Actualización de dirección o Información de mora, según la necesidad del usuario como se muestra en la figura 38.



Figura 38. Formulario información de contrato

- **Actualización de dirección**

Este formulario permite al usuario actualizar la dirección del abonado. Aquí se pueden actualizar los siguientes datos.

- Caja de texto de ingreso de la Dirección
- Caja de texto de ingreso del Sector
- Botón radio del tipo de predio

Para actualizar los datos de Latitud y longitud del predio existe un botón que toma el dato de un GPS. Al accionar este botón, se realiza el proceso de toma de datos y finalmente los despliega en el formulario.

Para actualizar la imagen del predio existe un botón que toma el dato de una cámara de fotos. Al accionar este botón, se realiza el proceso de toma de foto y finalmente la despliega en el formulario.

Finalmente, el formulario que se muestra en la figura 39 tiene un botón de guardado que al ser accionado permite guardar la información que se desea actualizar y regresar al formulario menú principal.



*Figura 39. Formulario de actualización de dirección*

- **Información de mora**

Este formulario de la figura 40 permite consultar la información de mora de un abonado. Para este fin el formulario despliega la siguiente información:

- Número total de mese de mora
- Valor total de la mora
- Listado de detalles de los meses que el abonado se encuentra en mora.

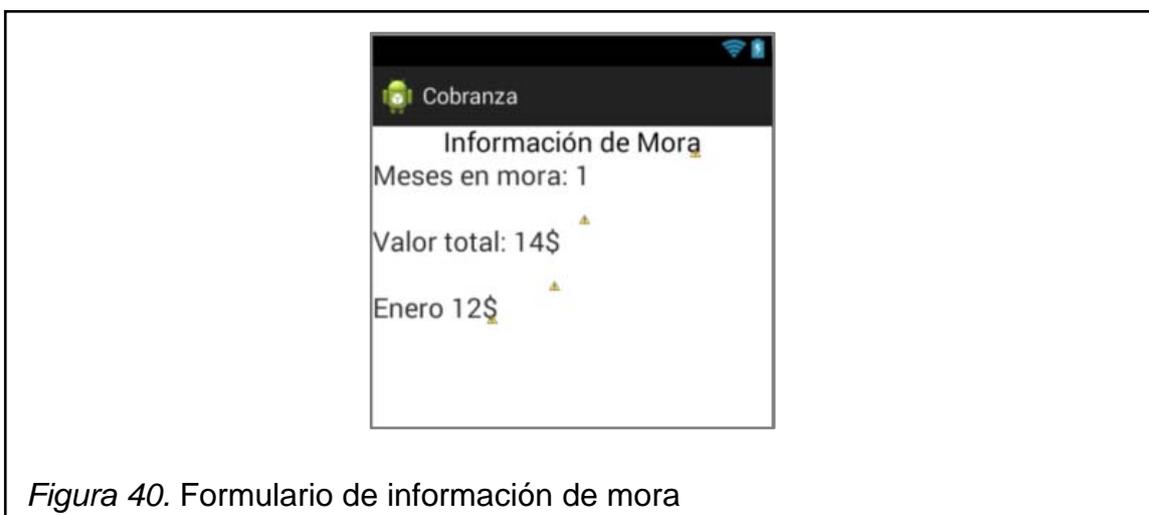
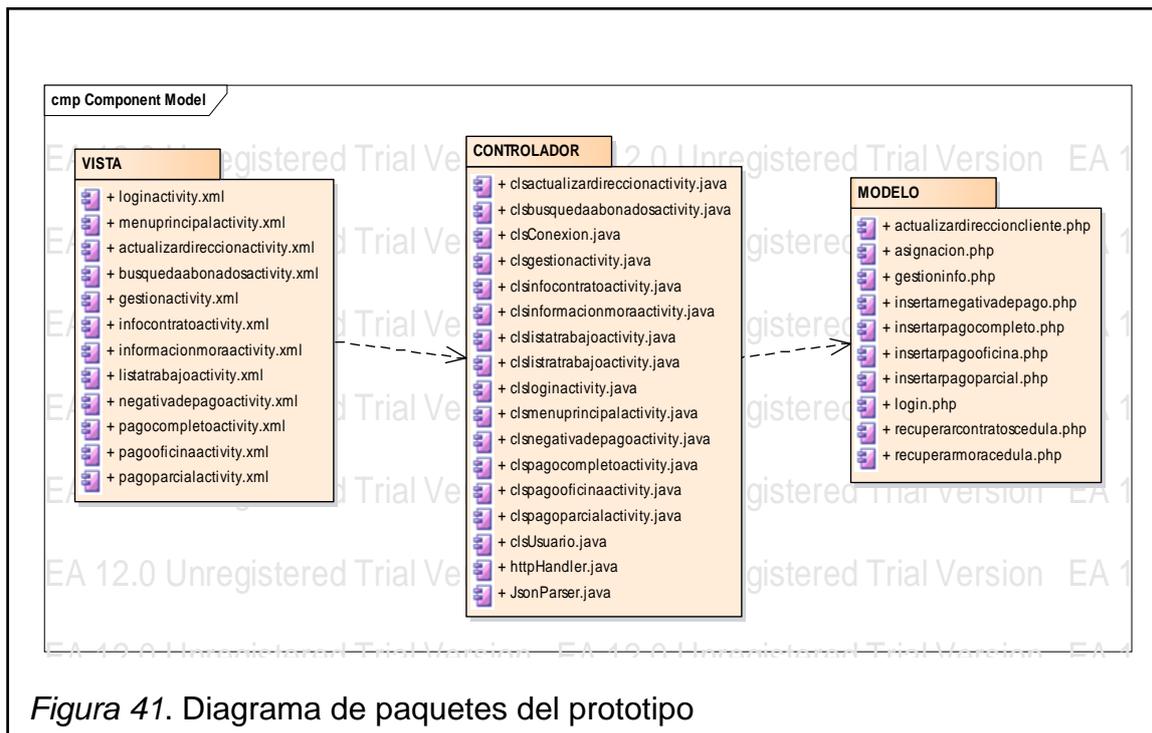


Figura 40. Formulario de información de mora

#### 5.4 Diagrama de Paquetes del prototipo

La figura 41 muestra el diagrama de paquetes del prototipo que detalla el patrón de diseño MVC



## CAPITULO VI

### PRUEBAS Y ENTREGA

El objetivo de este capítulo es realizar las pruebas del sistema programado y probar su desempeño en el dispositivo móvil en un ambiente de pruebas. Para este fin se ha compilado la solución he instalado en un teléfono móvil con sistema operativo Android. Finalmente se entregara al usuario final para que el sistema funcione en el ambiente de pruebas

Debido a la naturaleza del modelo de desarrollo RAD, cada parte del sistema ya se fue probando desde su desarrollo. Esto permite que el periodo de prueba sea más corto y se detecten errores en todo el proceso.

#### 6.1 Pruebas de Funcionalidad

Para realizar estas pruebas, nos centraremos en dos de las funcionalidades más importantes que son:

- La actualización de los datos de la dirección del abonado
- El ingreso de las gestiones del agente de cobranzas.

#### Prueba de actualización de datos de dirección del abonado

Para realizar esta prueba primero se establecerá el proceso por el cual el usuario actualiza datos, desde que ingresa al sistema hasta que actualiza los datos.

- **Formulario Login**

#### Proceso:

El usuario ingreso diferentes nombres de usuario y contraseñas.

**Resultado:**

Es sistema permitió pasar al siguiente formulario solo si los datos de usuario fueron correctos.

**Proceso:**

El usuario ingreso diferentes nombres de usuario y contraseñas pero sin conexión del terminal móvil la red.

**Resultado:**

Es sistema mostro un mensaje de error de conexión y se mantuvo en el formulario de Login.

- **Formulario Menú principal**

**Proceso:**

El usuario seleccionó la opción Búsqueda de abonados del Menú

**Resultado:**

Es sistema permitió pasar al siguiente formulario.

- **Formulario Búsqueda de abonado**

**Proceso:**

El usuario ingreso diferentes datos en cuadro de texto y envió la información a buscar.

**Resultado:**

Es sistema muestra los contratos existentes para otro tipo de datos no se envió nada.

**Proceso:**

El usuario ingreso diferentes datos en cuadro de texto y envió la información a buscar, pero sin conexión del terminal móvil la red.

**Resultado:**

Es sistema mostro un mensaje de error de conexión y se mantuvo en el formulario de Búsqueda de abonado.

**Proceso:**

El usuario seleccionó un contrato desplegado después de la búsqueda realizada.

**Resultado:**

Es sistema desplegó el siguiente formulario.

**Proceso:**

El usuario seleccionó un contrato desplegado después de la búsqueda realizada, pero sin conexión del terminal móvil la red.

**Resultado:**

Es sistema mostro un mensaje de error de conexión y se mantuvo en el formulario de Búsqueda de abonado.

- **Formulario Información del contrato**

**Proceso:**

El usuario seleccionó la opción de Actualización de dirección del formulario

**Resultado:**

Es sistema permitió pasar al siguiente formulario.

- **Formulario de Actualización de dirección**

**Proceso:**

El usuario inserto los datos de dirección nuevos, uso el botón de toma de datos de GPS y uso el botón de toma de fotografía y finalmente presiono el botón de guardado.

**Resultado:**

Es sistema mostró un mensaje de éxito al guardar y se dirigió al formulario de búsqueda de abonados.

**Proceso:**

El usuario dejó campos vacíos, y no uso el botón de toma de datos de GPS ni el botón de toma de fotografía y finalmente presiono el botón de guardado.

**Resultado:**

Es sistema mostró un mensaje de error por falta de datos y se mantuvo en el formulario.

**Proceso:**

El usuario inserto los datos de dirección nuevos, uso el botón de toma de datos de GPS y uso el botón de toma de fotografía y finalmente presiono el botón de guardado, pero el dispositivo se encontraba sin conexión a la red.

**Resultado:**

Es sistema mostro un mensaje de error de conexión y se mantuvo en el formulario de Actualización de dirección.

## **6.2 Comparación del proceso actual de cobranza con el proceso usando la aplicación**

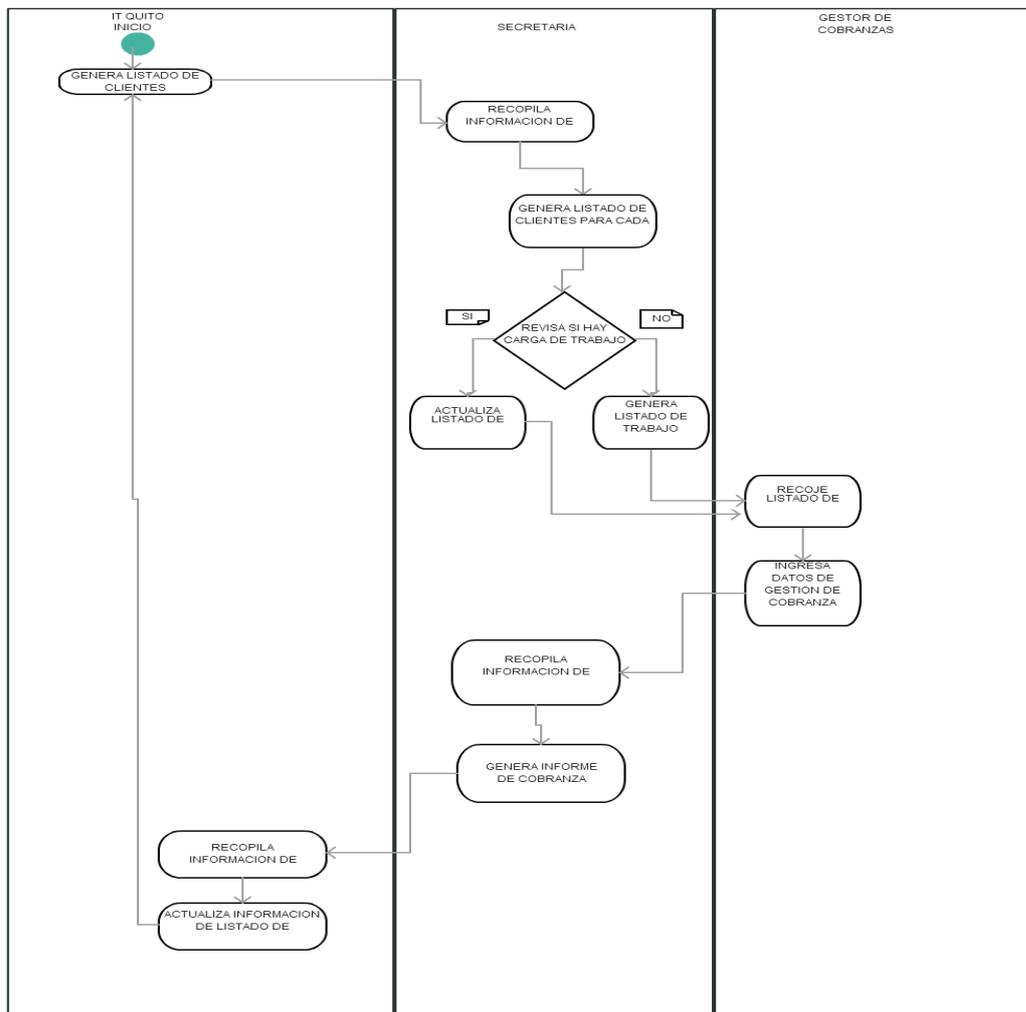
El proceso de cobranzas actualmente se maneja de forma manual, no existe un sistema que automatice el proceso de gestión de cobranza en el campo. Con el desarrollo de este prototipo se busca mejorar y controlar el mismo.

El proceso de cobranzas que usa el prototipo de la aplicación permite al gestor acceder a información que antes no la poseía de manera inmediata. Además, la intervención de la secretaria en el proceso de repartición de la carga de trabajo ha desaparecido y ahora la información de cuadro diario la genera la oficina de IT de Xanda Linkworks en Quito.

El informe de cuadro diario es usado por la secretaria para cuadrar la información generada por el prototipo con lo que el gestor tiene en dinero y papeles físicos ya sean recibos o facturas. Además, el personal de IT de Xanda Linkworks puede monitorear el tiempo en que se generan las gestiones y así en un futuro generar informes de tiempos de gestión

En la figura 42 se muestra el diagrama de responsabilidades del proceso actual y el proceso de cobranza con el prototipo de aplicación de gestión de cobranza.

### PROCESO ACTUAL



### PROCESO CON EL PROTOTIPO

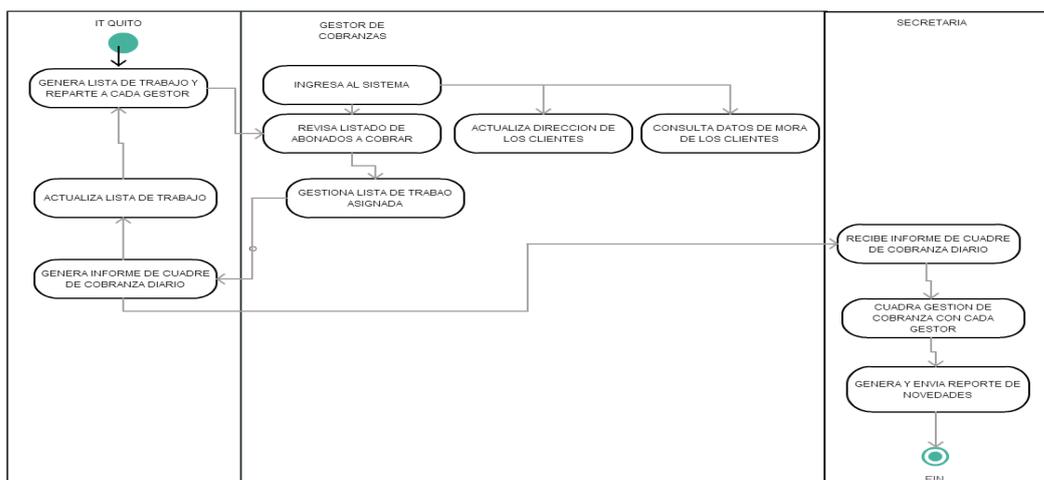


Figura 42. Diagrama de responsabilidades del proceso actual de cobranzas versus el proceso de cobranzas usando el prototipo

### 6.3 Entrega

Después de las pruebas realizadas se procedió a la entrega del sistema al cliente para las pruebas de campo. Para esto se instaló el software en un teléfono móvil de las siguientes características:

- Marca: Samsung
- Modelo: Galaxy S4
- Operadora: Movistar

En la figura 43 se muestra imágenes del sistema funcionando en el teléfono móvil

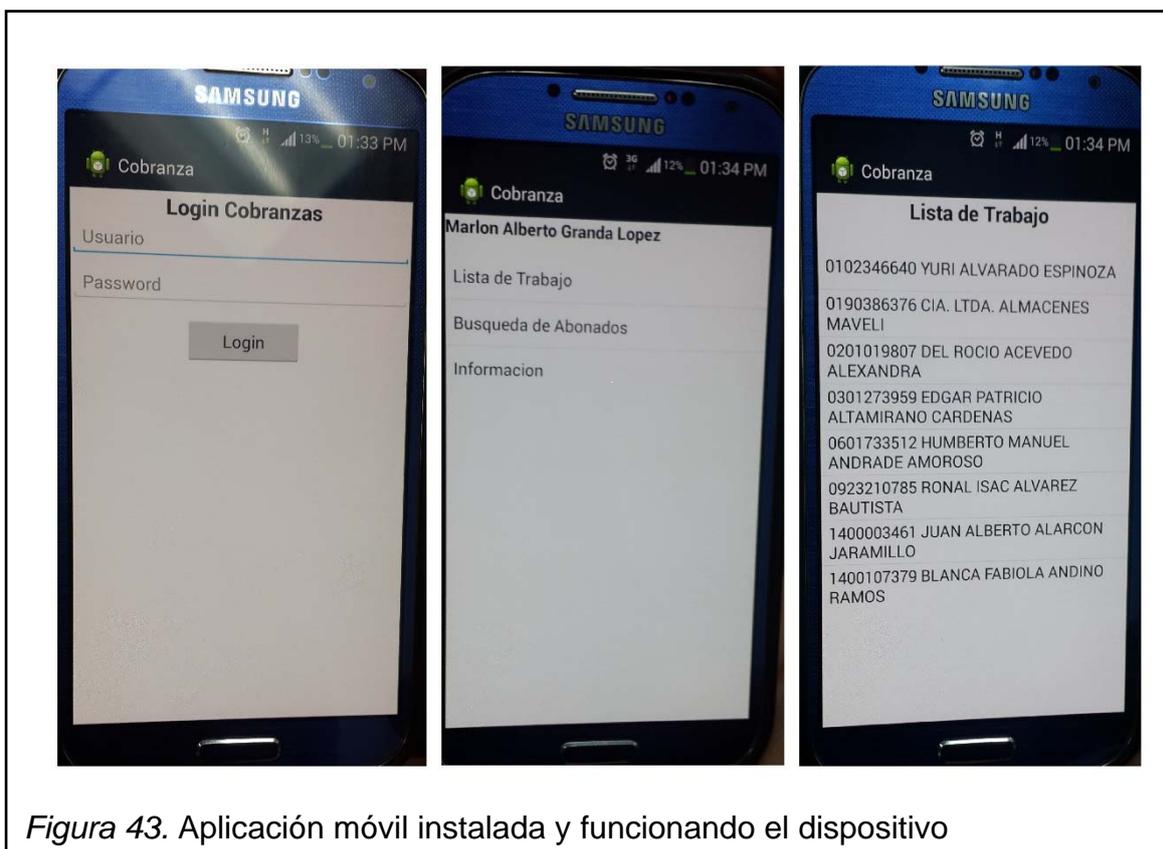


Figura 43. Aplicación móvil instalada y funcionando el dispositivo

## CAPITULO VII

### CONCLUSIONES Y RECOMENDACIONES

#### 7.1 Conclusiones

En el proyecto de tesis “Implementar un sistema de gestión para dispositivos móviles que disminuya el tiempo y optimice recursos dirigidos al personal de campo de la empresa XANDA LINKWORKS Cía. Ltda.” Se desarrolló un sistema usando varios lenguajes de programación, entre los cuales se pueden mencionar: PHP, Java y Sql. Usando los datos que la operación de cobranzas, las conclusiones a las que se llegaron son:

Se analizaron los requerimientos de la empresa XANDA LINKWORKS Cia. Ltda. Respecto a la necesidad de un software móvil para la gestión de sus clientes en el terreno y se determinó que un sistema basado en capas fue la mejor opción. Se llegó a esta conclusión debido a la escalabilidad y modularidad el sistema, pues este software, además de ser portado en dispositivos móviles, este puede escalar a varias plataformas.

Se desarrolló una aplicación móvil basada en Android para la gestión de campo de los clientes de la empresa XANDA LINKWORKS Cía. Ltda. Esta aplicación busca ordenar el sistema de gestiones que maneja la empresa ubicándolo en un repositorio e datos escalable y controlado. Además, este programa busca tener una fuente datos que pueda ser analizada para extraer informes de operación de cobranza.

En el proceso de desarrollo del prototipo, se utilizaron tecnologías como los servicios web que permitieron la conexión entre la capa de acceso a los datos que en este caso fue realizada en Java para Android y la base de datos. Este enlace se realizó gracias mensajes JSON que son compatibles para el lenguaje PHP con el que se realizó el servicio web y Java para Android.

Para realizar la evidencia fotográfica del predio del abonado, se usó las librerías de Java para Android que permiten el uso de la cámara integrada al dispositivo móvil. Estas librerías permitieron este cometido siendo esta funcionalidad de gran ayuda para la gestión de cobranzas.

Para realizar la toma de datos de GPS del predio del abonado, se usó las librerías de Java para Android que permiten el uso del GPS integrado al dispositivo móvil. Estas librerías permitieron este cometido siendo esta funcionalidad de gran ayuda para la ubicación del predio ya que, en la ciudad de Macas, las direcciones no son estandarizadas.

Se implementó la aplicación móvil basada en Android en la empresa XANDA LINKWORKS Cia. Ltda. Usando varios lenguajes de programación que facilitan este proceso y utilizando una metodología de software que permitió desarrollar y documentar todo el proceso de inicio a fin.

En el proceso de implantación se determinó que las tecnologías de consumo de información en la red como son los servicios web fueron la mejor solución a los problemas de conectividad entre la capa de presentación y la base de datos. Además, se determinó que el sistema operativo Android permite un desarrollo ordenado, escalable y que permite un sistema claro de conexión y consumo de servicios web

Se probó la aplicación móvil basada en Android en la empresa XANDA LINKWORKS Cia. Ltda., tanto en ambientes de prueba como en ambientes de producción, y se concluyó que el sistema ayuda a encontrar direcciones de una manera más gráfica. Esto se debe al uso de fotografías de predios que la aplicación muestra y a la referencia geográfica que indica, usando coordenadas de GPS. Esto permite que el gestor llegue a las coordenadas indicadas, extrayéndolas de la aplicación y usándola en diferentes tipos de sistemas de ubicación ya sea WAZE, Google maps, entre otras.

## 7.2 Recomendaciones

En el proyecto de tesis “Implementar un sistema de gestión para dispositivos móviles que disminuya el tiempo y optimice recursos dirigidos al personal de campo de la empresa XANDA LINKWORKS Cía. Ltda.” se tienen las siguientes recomendaciones.

La programación de la capa de presentación, la cual usa Java para Android, es muy compleja. La arquitectura de todo el sistema está basada en capas. Android, como tal, maneja tres capas más de las que previamente se diseñó. Por esta razón, se maneja gran cantidad de código que puede llegar desordenarse o no ser claro al final del proyecto.

La programación en capas, usando varios tipos de lenguajes, se puede volver confusa y difícil de manejar. Por esta razón se debería usar un Framework de desarrollo de aplicaciones móviles que ayude a solventar el problema de codificación multilenguaje y los problemas que esto puede traer.

## REFERENCIAS

- Achour, M. (2014). *PHP*. Recuperado el Octubre de 2014 de <http://www.php.net/manual/es/index.php>
- Androideity. (2012). *Android Developer Lab*. Recuperado el 10 de Agosto de 2014 de <http://androideity.com/>
- BCE. (2012). *Evolución del crédito y tasas de interés efectivas referenciales*. Recuperado el 1 de Julio de 2013, de <http://contenido.bce.fin.ec/docs.php?path=/documentos/Estadisticas/SectorMonFin/TasasInteres/Indice.htm>
- EcuRed. (s.f.). *Ecured*. Recuperado el 17 de Agosto de 2014, de [http://www.ecured.cu/index.php/Arquitectura\\_de\\_tres\\_niveles](http://www.ecured.cu/index.php/Arquitectura_de_tres_niveles)
- Guevara, L. (2014). *repositorio usfq*. Recuperado el 25 de Noviembre de 2014 de <http://repositorio.usfq.edu.ec/bitstream/23000/1476/1/102320.pdf>
- il-masacratore. (2014). *sql-server-express*. Recuperado el 20 de Agosto de 2014 de <http://www.dataprix.com/blog-it/bases-datos/que-tiene-como-instalar-sql-server-express-2012>.
- informatica-hoy. (2014). *Informática, tecnología e Internet sin complicaciones*. Recuperado el 21 de Agosto de 2014 de <http://www.informatica-hoy.com.ar/soluciones-moviles/Componentes-de-un-Sistema-Operativo-Movil.php>
- Informatico, M. (2014). *Rjprogramacion*. Recuperado el 8 de Noviembre de 2014 de <http://rjprogramacion.blogspot.com/2013/06/sql-server-conceptos-basicos.html>
- Json Org. (2014). *Introducción a JSON*. Recuperado el 2 de Diciembre de 2014 de <http://json.org/json-es.html>.
- Nava, S. (2014). *Servicios WEB*. Recuperado el 14 de Noviembre de 2014 de [http://www.capacinet.gob.mx/Cursos/Tecnologia%20amiga/desarrollador-desoftware/ServiciosWeb\\_SE.pdf](http://www.capacinet.gob.mx/Cursos/Tecnologia%20amiga/desarrollador-desoftware/ServiciosWeb_SE.pdf)
- PHP.ORG. (s.f.). *PHP*. Recuperado el 15 de Agosto de 2014 de <http://php.net/manual/es/intro.pdo.php>
- Universidad Simón Bolívar Venezuela, (s.f.). *Metodología RAD*. Recuperado el 7 de Agosto de 2014 de Laboratorio Docente de Computación:

<http://ldc.usb.ve/~vtheok/cursos/ci3711/apuntes/99-01-14/Info/Modelo%20RAD.htm>

VERA, T. (2012). *INGENIERÍA DEL SOFTWARE*. Recuperado el 21 de Agosto de 2014 de [http://ingsoftii.blogspot.com/2012\\_04\\_01\\_archive.html](http://ingsoftii.blogspot.com/2012_04_01_archive.html)

## **ANEXOS**

**ANEXO A**

Código de generación de la base de datos

```
USE [BDDXNLUIOCM01]
GO
/***** Object: Table [dbo].[Tbl_Asignacion]   Script Date: 20/08/2014
20:43:26 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Asignacion](
    [Asig_Id] [int] IDENTITY(1,1) NOT NULL,
    [Usr_Id] [int] NOT NULL,
    [Cont_Id] [int] NOT NULL,
    [Asig_Fecha] [date] NOT NULL,
    [Asig_Estado] [char](1) NOT NULL,
    [Per_Id] [char](10) NOT NULL,
    [Tcont_Id] [int] NOT NULL,
    [TUsr_Id] [int] NOT NULL,
    CONSTRAINT [PK_Tbl_Asignacion] PRIMARY KEY CLUSTERED
    (
        [Asig_Id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

```
GO
/***** Object: Table [dbo].[Tbl_Contratos]   Script Date: 20/08/2014 20:43:26
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Contratos](
    [Cont_Id] [int] NOT NULL,
    [Cont_NumeroTv] [int] NOT NULL,
    [Cont_ValorPago] [float] NOT NULL,
    [Cont_Status] [varchar](10) NOT NULL,
    [TCont_Id] [int] NOT NULL,
    [Per_Id] [char](10) NOT NULL,
    [Dir_Id] [int] NOT NULL,
    [Cont_FechaInicio] [date] NULL,
    [Cont_FechaFin] [date] NULL,
    CONSTRAINT [Cont_Id] PRIMARY KEY CLUSTERED
(
    [Cont_Id] ASC,
    [TCont_Id] ASC,
    [Per_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Tbl_Direccion]   Script Date: 20/08/2014 20:43:26
```

```
*****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
SET ANSI_PADDING ON  
GO  
CREATE TABLE [dbo].[Tbl_Direccion](  
    [Dir_Id] [int] NOT NULL,  
    [Dir_Tipo] [varchar](10) NOT NULL,  
    [Dir_Ciudad] [varchar](50) NULL,  
    [Dir_Direccion] [varchar](100) NOT NULL,  
    [Dir_Sector] [varchar](50) NOT NULL,  
    [Dir_Latitud] [real] NULL,  
    [Dir_Longitud] [real] NULL,  
    [Dir_Imagen] [varchar](200) NULL,  
    [Per_Id] [char](10) NOT NULL,  
    CONSTRAINT [Dir_Id] PRIMARY KEY CLUSTERED  
(  
    [Dir_Id] ASC,  
    [Per_Id] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]  
  
GO  
SET ANSI_PADDING OFF  
GO  
/***** Object: Table [dbo].[Tbl_Gestiones]   Script Date: 20/08/2014 20:43:26  
*****/  
SET ANSI_NULLS ON  
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Gestiones](
    [Gst_Id] [int] IDENTITY(1,1) NOT NULL,
    [Tgst_Id] [int] NOT NULL,
    [Cont_Id] [int] NOT NULL,
    [TCont_Id] [int] NOT NULL,
    [Per_Id] [char](10) NOT NULL,
    [Usr_Id] [int] NOT NULL,
    [Tusr_Id] [int] NOT NULL,
    [Gst_Fecha] [date] NOT NULL,
    [Gst_ValorPagado] [money] NOT NULL,
    [Gst_Saldo] [money] NOT NULL,
    [Gst_NroFactur] [varchar](100) NULL,
    [Gst_NroRecivo] [varchar](100) NULL,
    [Gst_MesGestionMora] [int] NULL,
    CONSTRAINT [PK_Tbl_Gestiones] PRIMARY KEY CLUSTERED
    (
        [Gst_Id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Tbl_HistorialMora]   Script Date: 20/08/2014
20:43:26 *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_HistorialMora](
    [Him_Id] [int] IDENTITY(1,1) NOT NULL,
    [Him_Ano] [int] NOT NULL,
    [Him_Mes] [int] NOT NULL,
    [Him_Valor] [float] NOT NULL,
    [Cont_Id] [int] NOT NULL,
    [Per_Id] [char](10) NOT NULL,
    [TCont_Id] [int] NOT NULL,
    [Him_Status] [char](1) NULL,
    CONSTRAINT [PK_Tbl_HistorialMora] PRIMARY KEY CLUSTERED
(
    [Him_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
GO
```

```
SET ANSI_PADDING OFF
```

```
GO
```

```
/***** Object: Table [dbo].[Tbl_Persona] Script Date: 20/08/2014 20:43:26
```

```
*****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
SET ANSI_PADDING ON
```

```
GO
```

```
CREATE TABLE [dbo].[Tbl_Persona](
```

```
[Per_Id] [char](10) NOT NULL,  
[Per_Ruc] [char](13) NULL,  
[Per_Nombres] [varchar](50) NOT NULL,  
[Per_Apellidos] [varchar](50) NOT NULL,  
[Per_EstadoCivil] [char](1) NOT NULL,  
[Per_NombreContacto] [varchar](50) NOT NULL,  
[Per_ApellidoContacto] [varchar](50) NOT NULL,  
[Per_ParentescoContacto] [varchar](10) NOT NULL,  
[Per_TelfContacto] [varchar](10) NOT NULL,  
[Per_mail] [varchar](100) NULL,  
CONSTRAINT [Per_Id] PRIMARY KEY CLUSTERED  
(  
    [Per_Id] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]  
  
GO  
SET ANSI_PADDING OFF  
GO  
/***** Object: Table [dbo].[Tbl_Telefono]   Script Date: 20/08/2014 20:43:26  
*****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
SET ANSI_PADDING ON  
GO  
CREATE TABLE [dbo].[Tbl_Telefono](  
    [Tlf_Id] [int] NOT NULL,  
    [Tlf_Categoria] [varchar](15) NULL,  
    [Tlf_Numero] [varchar](10) NOT NULL,
```

```

        [Per_Id] [char](10) NOT NULL,
CONSTRAINT [Tlf_Id] PRIMARY KEY CLUSTERED
(
    [Tlf_Id] ASC,
    [Per_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Tbl_TipoContrato]   Script Date: 20/08/2014
20:43:26 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_TipoContrato](
    [TCont_Id] [int] NOT NULL,
    [TCont_TipoContrato] [varchar](10) NOT NULL,
    [TCont_ValorStandarContrato] [float] NOT NULL,
    [TCont_ValorTvExtra] [float] NOT NULL,
CONSTRAINT [Cont_Id1] PRIMARY KEY CLUSTERED
(
    [TCont_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Tbl_TipoGestion]   Script Date: 20/08/2014
20:43:26 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_TipoGestion](
    [Tgst_Id] [int] NOT NULL,
    [Tgst_TipoGestion] [varchar](10) NOT NULL,
    [Tgst_Descripcion] [varchar](50) NOT NULL,
    [Tgst_Status] [char](1) NOT NULL,
    CONSTRAINT [Tgst_Id] PRIMARY KEY CLUSTERED
(
    [Tgst_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Tbl_TipoUsuario]   Script Date: 20/08/2014
20:43:26 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_TipoUsuario](
    [Tusr_Id] [int] NOT NULL,
    [Tusr_TipoUsuario] [varchar](10) NOT NULL,
    [Tusr_Descripcion] [varchar](50) NULL,
    [Tusr_Status] [char](1) NOT NULL,
    CONSTRAINT [Tusr_Id] PRIMARY KEY CLUSTERED
(
    [Tusr_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[Tbl_Usuarios]  Script Date: 20/08/2014 20:43:26
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Tbl_Usuarios](
    [Usr_Id] [int] NOT NULL,
    [Usr_Nombres] [varchar](50) NOT NULL,
    [Usr_Apellidos] [varchar](50) NOT NULL,
    [Usr_NombreUsuario] [varchar](20) NOT NULL,
    [Usr_Password] [varchar](32) NULL,
    [Usr_Status] [char](1) NOT NULL,
```

```
[Tusr_Id] [int] NOT NULL,  
[Usr_Sector] [int] NULL,  
CONSTRAINT [Usr_Id] PRIMARY KEY CLUSTERED  
(  
    [Usr_Id] ASC,  
    [Tusr_Id] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]  
  
GO  
SET ANSI_PADDING OFF  
GO  
ALTER TABLE [dbo].[Tbl_Asignacion] WITH CHECK ADD CONSTRAINT  
[FK_Tbl_Asignacion_Tbl_Contratos] FOREIGN KEY([Cont_Id], [Tcont_Id],  
[Per_Id])  
REFERENCES [dbo].[Tbl_Contratos] ([Cont_Id], [TCont_Id], [Per_Id])  
GO  
ALTER TABLE [dbo].[Tbl_Asignacion] CHECK CONSTRAINT  
[FK_Tbl_Asignacion_Tbl_Contratos]  
GO  
ALTER TABLE [dbo].[Tbl_Asignacion] WITH CHECK ADD CONSTRAINT  
[FK_Tbl_Asignacion_Tbl_Usuarios] FOREIGN KEY([Usr_Id], [TUsr_Id])  
REFERENCES [dbo].[Tbl_Usuarios] ([Usr_Id], [Tusr_Id])  
GO  
ALTER TABLE [dbo].[Tbl_Asignacion] CHECK CONSTRAINT  
[FK_Tbl_Asignacion_Tbl_Usuarios]  
GO  
ALTER TABLE [dbo].[Tbl_Contratos] WITH CHECK ADD CONSTRAINT  
[FK_Tbl_Contratos_Tbl_Direccion] FOREIGN KEY([Dir_Id], [Per_Id])  
REFERENCES [dbo].[Tbl_Direccion] ([Dir_Id], [Per_Id])  
GO
```

```
ALTER TABLE [dbo].[Tbl_Contratos] CHECK CONSTRAINT
[FK_Tbl_Contratos_Tbl_Direccion]
GO
ALTER TABLE [dbo].[Tbl_Contratos] WITH CHECK ADD CONSTRAINT
[Relacion Persona Contratos] FOREIGN KEY([Per_Id])
REFERENCES [dbo].[Tbl_Persona] ([Per_Id])
GO
ALTER TABLE [dbo].[Tbl_Contratos] CHECK CONSTRAINT [Relacion
Persona Contratos]
GO
ALTER TABLE [dbo].[Tbl_Contratos] WITH CHECK ADD CONSTRAINT
[RelacionTipoContrato Contrato] FOREIGN KEY([TCont_Id])
REFERENCES [dbo].[Tbl_TipoContrato] ([TCont_Id])
GO
ALTER TABLE [dbo].[Tbl_Contratos] CHECK CONSTRAINT
[RelacionTipoContrato Contrato]
GO
ALTER TABLE [dbo].[Tbl_Direccion] WITH CHECK ADD CONSTRAINT
[Relacion Persona Direccion] FOREIGN KEY([Per_Id])
REFERENCES [dbo].[Tbl_Persona] ([Per_Id])
GO
ALTER TABLE [dbo].[Tbl_Direccion] CHECK CONSTRAINT [Relacion
Persona Direccion]
GO
ALTER TABLE [dbo].[Tbl_Gestiones] WITH CHECK ADD CONSTRAINT
[FK_Tbl_Gestiones_Tbl_Contratos] FOREIGN KEY([Cont_Id], [TCont_Id],
[Per_Id])
REFERENCES [dbo].[Tbl_Contratos] ([Cont_Id], [TCont_Id], [Per_Id])
GO
ALTER TABLE [dbo].[Tbl_Gestiones] CHECK CONSTRAINT
[FK_Tbl_Gestiones_Tbl_Contratos]
GO
ALTER TABLE [dbo].[Tbl_Gestiones] WITH CHECK ADD CONSTRAINT
```

```
[FK_Tbl_Gestiones_Tbl_Usuarios] FOREIGN KEY([Usr_Id], [Tusr_Id])
REFERENCES [dbo].[Tbl_Usuarios] ([Usr_Id], [Tusr_Id])
GO
ALTER TABLE [dbo].[Tbl_Gestiones] CHECK CONSTRAINT
[FK_Tbl_Gestiones_Tbl_Usuarios]
GO
ALTER TABLE [dbo].[Tbl_Gestiones] WITH CHECK ADD CONSTRAINT
[RelacionTipoGestion Gestiones] FOREIGN KEY([Tgst_Id])
REFERENCES [dbo].[Tbl_TipoGestion] ([Tgst_Id])
GO
ALTER TABLE [dbo].[Tbl_Gestiones] CHECK CONSTRAINT
[RelacionTipoGestion Gestiones]
GO
ALTER TABLE [dbo].[Tbl_HistorialMora] WITH CHECK ADD CONSTRAINT
[FK_Tbl_HistorialMora_Tbl_Contratos] FOREIGN KEY([Cont_Id], [TCont_Id],
[Per_Id])
REFERENCES [dbo].[Tbl_Contratos] ([Cont_Id], [TCont_Id], [Per_Id])
GO
ALTER TABLE [dbo].[Tbl_HistorialMora] CHECK CONSTRAINT
[FK_Tbl_HistorialMora_Tbl_Contratos]
GO
ALTER TABLE [dbo].[Tbl_Telefono] WITH CHECK ADD CONSTRAINT
[Relacion Persona Telefono] FOREIGN KEY([Per_Id])
REFERENCES [dbo].[Tbl_Persona] ([Per_Id])
GO
ALTER TABLE [dbo].[Tbl_Telefono] CHECK CONSTRAINT [Relacion Persona
Telefono]
GO
ALTER TABLE [dbo].[Tbl_Usuarios] WITH CHECK ADD CONSTRAINT
[RelacionTipoUsuarioUsuario] FOREIGN KEY([Tusr_Id])
REFERENCES [dbo].[Tbl_TipoUsuario] ([Tusr_Id])
GO
ALTER TABLE [dbo].[Tbl_Usuarios] CHECK CONSTRAINT
```

```
[RelacionTipoUsuarioUsuario]
GO
```

## ANEXO B

- Archivo Loginand.php

```
<?php
try {
$dbh = new
PDO("sqlsrv:Server=GUSTAVOAMD;Database=BDDXNLUIOCM01",'sa',
'antagus');
} catch (PDOException $e)
{header('Content-type: application/json');
echo json_encode([]);}
If (isset($_POST['user']) && isset($_POST['c']) )
{$username = $_POST ['user'];
$password = $_POST ['password'];
$sth = $dbh->prepare ("SELECT COUNT(*) FROM dbo.Tbl_Usuarios WHERE
Usr_NombreUsuario='$username' and Usr_Password='$password' and
Usr_Status='A'");
$sth->execute ();
$result = $sth->fetchAll();
if( $result[0][0]==1){
    $sth1 = $dbh->prepare("SELECT
Usr_id,Usr_Nombres,Usr_Apellidos,Usr_NombreUsuario,Tusr_Id FROM
dbo.Tbl_Usuarios WHERE Usr_NombreUsuario='$username' and
Usr_Password='$password'");
    $sth1->execute();
    $result1 = $sth1->fetchAll(PDO::FETCH_ASSOC);
        header('Content-type: application/json');
        echo json_encode($result1);
    }
else{header('Content-type: application/json'); cho json_encode([]);}
```

```
else {echo "Could not complete query. Missing parameter";}>
```

- Archivo Asignacionand.php

```
<?php
try {
$dbh = new PDO("sqlsrv:Server=GUSTAVOAMD;
Database=BDDXNLUIOCM01",'sa', 'antagus');
} catch (PDOException $e)
{
    header('Content-type: application/json');
    echo json_encode($e);
}
If ( $_POST['usr_id'] != null and $_POST['usr_id'] != "")
{
$usr_id = $_POST['usr_id'];
$Status = 'A';
$sth = $dbh->prepare("select b.Per_Id+' '+b.Per_Nombres+' '+b.Per_Apellidos
as Asignacion, a.Cont_Id as Id_Contrato from Tbl_Asignacion a inner join
Tbl_Persona b on a.Per_Id=b.Per_Id where a.Usr_Id=$usr_id and
a.Asig_Estado='$Status'");
$sth->execute();
$result = $sth->fetchAll(PDO::FETCH_ASSOC);

header('Content-type: application/json');
echo json_encode(array_values($result));
}
else{
echo "Could not complete query. Missing parameter";}>
```

## ANEXO C

Código de la capa del controlador del formulario Listadetrabajoactivity.xml

```
package com.example.cobranza;
```

```
import com.example.cobranza.ClsUsuario;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.ExecutionException;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;

import com.example.cobranza.MainActivity.Tarea;

import android.R.integer;
import android.app.ListActivity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class ClsListaTrabajoAct extends ListActivity{
//TextView StrContenido;
private ProgressDialog pDialog;
private httpHandler ObjHttpHand=new httpHandler();
TextView StrContenido;
String[][] Dato;
```

```

ClsConexion ObjCon=new ClsConexion();
@Override

protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.listatrabajoactivity);
final ClsUsuario ObjClsUsuario=(ClsUsuario) getApplicationContext();
ListView LstVMenu=(ListView) this.findViewById(android.R.id.list);
StrContenido=(TextView) this.findViewById(R.id.TxvContenido);
try {
Tarea T=new Tarea();
ObjHttpHand.params=new ArrayList<NameValuePair>();
ObjHttpHand.params.add(new
BasicNameValuePair("usr_id",Integer.toString(ObjClsUsuario.getIntIdUsuario()
)));
T.execute();
try {

if (T.get().equals("Error de Conexion" ) || T.get().equals("Connect to .. timed
out")) {
//StrMensaje.setText("Error de Conexion");

}
else if (T.get().compareTo("[[]\r\n")!=0)
{
JsonParser ObjJp=new JsonParser();
try {
Dato=ObjJp.LeerJsonStringAsignacion(T.get().toString());
String[] Datos= new String[Dato.length];
for (int i = 0; i < Datos.length; i++)
{
Datos[i]=Dato[i][0];
}
}
}
}
}
}

```

```

ArrayAdapter<String> ArrAdAdaptador= new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,Datos);
setListAdapter(ArrAdAdaptador);
} catch (JSONException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

} else{
Toast.makeText(getApplicationContext(), "No hay
datos",Toast.LENGTH_LONG).show();
}
} catch (InterruptedException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (ExecutionException e) {
// TODO Auto-generated catch block
e.printStackTrace ();}
} catch (Exception e) {
// TODO: handle exception
//StrMensaje.setText(e.getMessage().toString());}

@Override
protected void onItemClick(ListView l, View v, int position, long id) {
super.onItemClick(l, v, position, id);
// ListView Clicked item index
int itemPosition    = position;
// ListView Clicked item value
String itemValue    = (String) l.getItemAtPosition(position);
StrContenido.setText("Clickc : \n Position : "+itemPosition+" \n ListItem : "
+itemValue);
Intent i = new Intent(getApplicationContext(),ClsGestionAct.class);
i.putExtra("StrIdNombre",itemValue);

```

```

i.putExtra("Id_Contrato",Dato[itemPosition][1]);
startActivity(i);
}
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
if (keyCode == KeyEvent.KEYCODE_BACK) {
onBackPressed();
}
return super.onKeyDown(keyCode, event);
}
public void onBackPressed() {
Intent myIntent = new Intent(getBaseContext(), ClsMenuPrincipalAct.class);
myIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);// clear back stack
startActivity(myIntent);
finish();
return;
}
class Tarea extends AsyncTask< String , Void, String>{
@Override
protected void onPreExecute() {
super.onPreExecute();
// Showing progress dialog
pDialog = new ProgressDialog(ClsListaTrabajoAct.this);
pDialog.setMessage("Conectando...");
pDialog.setCancelable(false);
pDialog.show();
}
@Override
protected String doInBackground(String... params) {
String StrJson = null;
StrJson = ObjHttpHand.post( ObjCon.getStrUrl()+"/ws/asignacionand.php");
if (pDialog.isShowing())
pDialog.dismiss();
}
}

```

```

return StrJson;
}
public void onBackPressed() {
Intent myIntent = new Intent(getBaseContext(), ClsMenuPrincipalAct.class);
myIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);// clear back stack
startActivity(myIntent);
finish();
return;} }

```

## ANEXO D

```
<uses-permission android:name="android.permission.INTERNET" />
```

A continuación se muestra la codificación de la clase HttpHandler.java:

```

package com.example.cobranza;
import android.R.string;
public class httpHandler {
public List<NameValuePair> params=new ArrayList<NameValuePair>();
public String post(String posturl){
HttpParams httpParams = new BasicHttpParams();
ConnManagerParams.setTimeout(httpParams, 10000);
HttpConnectionParams.setConnectionTimeout(httpParams,50000);
HttpConnectionParams.setSoTimeout(httpParams, 50000);
try {
                HttpClient httpclient = new DefaultHttpClient(httpParams);
/*Creamos el objeto de HttpClient que nos permitira conectarnos mediante
peticiones http*/
HttpPost httppost = new HttpPost(posturl);
/*El objeto HttpPost permite que enviemos una peticion de tipo POST a una
URL especificada*/
        //AÑADIR PARAMETROS
/*Una vez añadidos los parametros actualizamos la entidad de httppost, esto
quiere decir en pocas palabras anexamos los parametros al objeto para que al
enviarse al servidor envíen los datos que hemos añadido*/

```

```

httppost.setEntity(new UrlEncodedFormEntity(params));
/*Finalmente ejecutamos enviando la info al server*/
HttpResponse resp = httpclient.execute(httppost);
HttpEntity ent = resp.getEntity();/*y obtenemos una respuesta*/
String text = EntityUtils.toString(ent);
return text;
}
catch (ClientProtocolException e) {
return e.getMessage().toString(); }
catch (ConnectTimeoutException e) {
return e.getMessage().toString();
}
    Catch (Exception e) { return "Error de Conexion";}
}

```

## **ANEXO E**

```

Publicclass JsonParser {

    public List<String> LeerJsonStringLogin(String StrCadena) throws
JSONException {
JSONArray jsonarray = newJSONArray(StrCadena);
List<String> Lista= new ArrayList<String>();
JSONObject Obj=new JSONObject();
for (int i = 0; i < jsonarray.length(); i++) {
Obj=jsonarray.getJSONObject(i);
//And then read attributes like
Lista.add(Obj.getString("Usr_id"));
Lista.add(Obj.getString("Usr_Nombres"));
Lista.add(Obj.getString("Usr_Apellidos"));
Lista.add(Obj.getString("Usr_NombreUsuario"));
Lista.add(Obj.getString("Tusr_Id"));return Lista;}
}

```

```

public String[][] LeerJsonStringAsignacion(String StrCadena) throws
JSONException {
JSONArray jsonarray = newJSONArray(StrCadena);
String[][] Lista= new String[jsonarray.length()][2];
JSONObject Obj=new JSONObject();
for (int i = 0; i < Lista.length; i++) {
Obj=jsonarray.getJSONObject(i);
//And then read attributes like
for (int j = 0; j < Lista[i].length; j++) {

if (j==0) {
Lista[i][j]=(Obj.getString("Asignacion"));

}

if (j==1) {
Lista[i][j]=(Obj.getString("Id_Contrato")); } return Lista;}

```

```

public List<String> LeerJsonStringContrato(String StrCadena) throws
JSONException {

```

```

JSONArray jsonarray = newJSONArray(StrCadena);
List<String> Lista= new ArrayList<String>();
JSONObject Obj=new JSONObject();
for (int i = 0; i < jsonarray.length(); i++) {

```

```

Obj=jsonarray.getJSONObject(i);
//And then read attributes like

```

```

Lista.add(Obj.getString("Cont_Id"));
Lista.add(Obj.getString("Cont_ValorPago"));
Lista.add(Obj.getString("Dir_Direccion"));
Lista.add(Obj.getString("Dir_Imagen"));
Lista.add(Obj.getString("Dir_Latitud"));
Lista.add(Obj.getString("Dir_Longitud"));

```

```

        Lista.add(Obj.getString("Dir_Sector"));
        Lista.add(Obj.getString("TCont_Id"));}

return Lista;}

```

## ANEXO F

```

class Tarea extends AsyncTask< String , Void, String>{
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        // Showing progress dialog
        pDialog = new ProgressDialog(ClsListaTrabajoAct.this);
        pDialog.setMessage("Conectando...");
        pDialog.setCancelable(false);
        pDialog.show();
    }
    @Override
    protected String doInBackground(String... params) {
        String StrJson = null;
        StrJson = ObjHttpHand.post( ObjCon.getStrUrl()+"/ws/asignacionand.php");
        if (pDialog.isShowing())
            pDialog.dismiss();
        return StrJson;
    }
}

```

## ANEXO G

```

loc.getLatitude();
loc.getLongitude();
String coordenadas = "coordenadas:" + "Latitud = " + loc.getLatitude() +
"Longitud = " + loc.getLongitude();
Toast.makeText(
    getApplicationContext(),coordenadas,Toast.LENGTH_LONG).show();
}
public void onProviderDisabled(String provider)

```

```

{
    Toast.makeText(
        getApplicationContext(),"Gps Desactivado",Toast.LENGTH_SHORT ).show();
    }
    public void onProviderEnabled(String provider)
    {
        Toast.makeText(
            getApplicationContext(),"Gps Activo",Toast.LENGTH_SHORT ).show();
    }
}

```

## ANEXO H

```

btnCamara = (Button) findViewById(R.id.button1);
btnCamara.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        Uri output = Uri.fromFile(new File(foto));
        intent.putExtra(MediaStore.EXTRA_OUTPUT, output);
        startActivityForResult(intent, TAKE_PICTURE);
    }
});
class UploaderFoto extends AsyncTask{
    ProgressDialog pDialog;
    String Fotopredio = "";
    @Override
    protected void doInBackground(String... params) {
        miFoto = params[0];
        try{
            HttpClient httpClient = new DefaultHttpClient();
            httpClient.getParams().setParameter(CoreProtocolPNames.PROTOCOL_VERSION, HttpVersion.HTTP_1_1);
            HttpPost httpPost = new HttpPost("http://antagus1984.no-ip.org/upload.php");

```

```
File file = newFile(Fotopredio);
MultipartEntity mpEntity = newMultipartEntity();
ContentBody foto = newFileBody(file, "image/jpeg");
mpEntity.addPart("fotoUp", foto);
httppost.setEntity(mpEntity);
httpClient.execute(httppost);
httpClient.getConnectionManager().shutdown();
} catch(Exception e) {
    e.printStackTrace();
}
returnnull;
}
protectedvoidonPreExecute() {
    super.onPreExecute();
    pDialog = newProgressDialog(MainActivity.this);
    pDialog.setMessage("Subiendo la imagen, espere.");
    pDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
    pDialog.setCancelable(true);
    pDialog.show();
}
protectedvoidonPostExecute(Void result) {
    super.onPostExecute(result);
    pDialog.dismiss();
}
```