**UNIVERSIDAD TÉCNICA DE MACHALA**

**UNIDAD ACADEMICA DE INGENIERÍA CIVIL**

**CARRERA DE INGENIERÍA DE SISTEMAS**

**TRABAJO DE TITULACIÓN**

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA DE SISTEMAS**

**TEMA:**

**“DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN Y SOPORTE DE LAS ACTIVIDADES DE CAMPO REALIZADAS POR EL PERSONAL OPERATIVO DE CONSTRUCCIONES E INSTALACIONES ELÉCTRICAS (CONIEL CIA.LTDA.)”**

**AUTORA:**

**LOAIZA GONZAGA ANDREA ANABELL**

**TUTOR:**

**ING. FAUSTO REDROVÁN CASTILLO, MG. SC.**

**CO-TUTOR**

**ING. JOFFRE CARTUCHE VARGAS**

**MACHALA - EL ORO - ECUADOR**

**2014**

**CERTIFICACIÓN DEL TUTOR**

Ing. FAUSTO REDROVÁN CASTILLO, Mg. Sc, Profesor de la Unidad Académica de Ingeniería Civil de la Universidad Técnica de Machala, en calidad de Tutor de la Tesis de Grado titulada: “DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN Y SOPORTE DE LAS ACTIVIDADES DE CAMPO REALIZADAS POR EL PERSONAL OPERATIVO DE CONSTRUCCIONES E INSTALACIONES ELÉCTRICAS (CONIEL CIA.LTDA.)”, elaborado por la estudiante ANDREA ANABELL LOAIZA GONZAGA, egresada de la carrera de Ingeniería de Sistemas de la Escuela de Informática, certifico que la mencionada Tesis estuvo bajo mi dirección y supervisión ajustándose a los procedimientos académicos y metodológicos establecidos por la Facultad, razón por la que autorizo su presentación para el trámite legal correspondiente.

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

Ing. Fausto Redrován Castillo, Mg. Sc.

**CERTIFICACIÓN DEL CO-TUTOR**

Ing. JOFFRE CARTUCHE VARGAS, Profesor de la Unidad Académica de Ingeniería Civil de la Universidad Técnica de Machala, en calidad de Co-Tutor de la Tesis de Grado titulada: “DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN Y SOPORTE DE LAS ACTIVIDADES DE CAMPO REALIZADAS POR EL PERSONAL OPERATIVO DE CONSTRUCCIONES E INSTALACIONES ELÉCTRICAS (CONIEL CIA.LTDA.)”, elaborado por la estudiante ANDREA ANABELL LOAIZA GONZAGA, egresada de la carrera de Ingeniería de Sistemas de la Escuela de Informática, certifico que la mencionada Tesis estuvo bajo mi dirección y supervisión ajustándose a los procedimientos académicos y metodológicos establecidos por la Facultad, razón por la que autorizo su presentación para el trámite legal correspondiente.

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

Ing. Joffre Cartuche Vargas

**DECLARACIÓN DE AUTORÍA**

El desarrollo y levantamiento de información, expuesta en este trabajo de investigación denominado: **“DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN Y SOPORTE DE LAS ACTIVIDADES DE CAMPO REALIZADAS POR EL PERSONAL OPERATIVO DE CONSTRUCCIONES E INSTALACIONES ELÉCTRICAS (CONIEL CIA.LTDA.)”,** es de exclusiva autoría y responsabilidad de Andrea Anabell Loaiza Gonzaga con C.I.N° 070569847-0.

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

Andrea Anabell Loaiza Gonzaga

**AGRADECIMIENTO**

A Dios por iluminar mi camino, dame fuerzas para vencer los obstáculos y culminar mi carrera.

A mis padres dignos de ejemplo, trabajo y constancia quienes me han brindado su amor incondicional, han estado en todos los momentos de mi vida apoyándome y alentándome a seguir adelante.

A una persona especial cuyo respaldo y ánimos han sido decisivos en momentos de angustia y desesperación.

A mi tutor de tesis Ingeniero Fausto Redrován, quien durante todo este tiempo me colaboró en el desarrollo de este trabajo.

A mis profesores que han sido participes en mi formación académica dentro de esta prestigiosa Institución.

Agradecer a todas aquellas personas que en mayor o menor medida han ayudado a que este trabajo se desarrolle.

***Andrea Loaiza***

**DEDICATORIA**

Quiero dedicar este trabajo en primer lugar a Dios, por ser quien guía mi vida y por darme la fortaleza para poder alcanzar esta meta.

A mis padres Daniel Loaiza y Herminia Gonzaga, quienes han sido el pilar fundamental de mi vida, por su apoyo, consejos, comprensión y por ayudarme con los recursos necesarios para estudiar.

A mis hermanos por estar siempre presentes, acompañándome para poderme realizar y por enseñarme a encarar ñas adversidades sin desfallecer en el intento.

***Andrea Loaiza***

**“DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN Y SOPORTE DE LAS ACTIVIDADES DE CAMPO REALIZADAS POR EL PERSONAL OPERATIVO DE CONSTRUCCIONES E INSTALACIONES ELÉCTRICAS (CONIEL CIA.LTDA.)”**

Andrea Anabell Loaiza Gonzaga

**Resumen**

La Compañía de construcciones e instalaciones eléctricas Coniel Cia.Ltda ha brindado un buen servicio a la comunidad en el área de ingeniería eléctrica, desempeñándose en estos últimos años bajo la adjudicación de contratos de control de perdida de energía por la CNEL EP.

En la empresa el personal operativo ejecuta procesos manuales para el tratamiento y acceso a la información, por lo que se genera cierto retraso en las actividades que realizan diariamente.

Esto me ha motivado a poner en funcionamiento una propuesta innovadora desarrollada para la gestión de las actividades realizadas por el personal operativo con el fin de desarrollar e implementar una aplicación móvil que permita acceder a la información desde el sitio de trabajo y optimizar el tratamiento de los datos.

Esta aplicación interactiva, rápida y funcional ha sido desarrollada siguiendo los fundamentos de gestión de proyectos de la metodología ágil Mobile-D, junto con tecnologías como el Sistema Operativo Android, el lenguaje de programación Java y con acceso a un servidor con base de datos PostgreSQL.

Por lo que la aplicación se convirtió en gran apoyo para la compañía ya que aceleró la ejecución de sus procesos.

***Palabras Clave:*** Aplicación móvil, acceso a la información, procesos manuales, retraso.

**Abstract**

INTRODUCCIÓN

La utilización inadecuada de las nuevas tecnologías de información, genera retraso en los procesos productivos dentro de las empresas u organizaciones, ya que el tratamiento de la información se la realiza mediante tareas manuales.

En la actualidad el avance tecnológico en el ámbito empresarial ha permitido que las organizaciones tengan una mayor rentabilidad, agilidad en la ejecución de sus procesos, rápido y fácil acceso a la información y la transparencia en los servicios o productos que ofertan.

La información es el activo más importante dentro de una organización por lo que debe ser confiable, integra y de fácil acceso.

Grandes empresas como Coca-Cola que cuentan con personal operativo refiriéndose a los trabajadores encargados de tomar pedidos desde los puntos de venta al cliente (tiendas, bares, etc.), cuentan con herramientas que facilitan la transferencia de información en tiempo real, ayudando de esta manera a agilizar los procesos y brindando fácil acceso a la información desde cualquier punto.

Las empresas que se dedican a brindar servicios específicamente de control de pérdida de energía a nivel nacional, no utilizan herramientas tecnológicas para realizar actividades operativas de campo y además el personal encargado no cuenta con la capacitación requerida para usar nuevas tecnologías, por lo que en la compañía de Construcciones e Instalaciones Eléctricas CONIEL CIA.LTDA aún utilizan métodos tradicionales como: el uso de formatos impresos para el control de las actividades operativas, la utilización de cámaras para la captura de fotografías concernientes al trabajo realizado, y la comunicación a través de radios o teléfonos celulares que en muchas veces no cuentan con la claridad necesaria para el intercambio de información eficiente.

Para mejorar el desarrollo de las actividades que realizan las compañías dedicadas a los trabajos de control de perdida de energía, es necesario implementar nuevas técnicas o mecanismos para optimizar el uso de tecnologías como Aplicaciones móviles que permitan cubrir las falencias que se han venido dado en la ejecución de los trabajos.

Para el desarrollo e implementación de la aplicación móvil se utilizó la plataforma Android aplicando la metodología ágil Mobile-D y con una base de datos POSTGRESQL para el almacenamiento de la información.

JUSTIFICACIÓN

Los procesos automatizados son factores de vital importancia en cualquier empresa. El desarrollo de las tecnologías de información ha permitido la evolución de los dispositivos móviles capaces de manejar sistemas tecnológicos en distintas áreas.

Las actividades diarias desarrolladas en el campo por el personal operativo de CONIEL CIA.LTDA, requieren que la información sea procesada y almacenada de manera eficiente, hecho que agilizará los procesos de otras actividades.

Con el desarrollo de una aplicación que se ejecute desde un dispositivo móvil se resolverán las necesidades y requerimientos de la empresa, además se logrará un control integral de las actividades y se optimizará el procesamiento de los datos.

La portabilidad de los dispositivos móviles facilitan su traslado al lugar del trabajo, además esto incrementará la confiabilidad de la información, mediante la disminución de pérdida o alteración de los datos por el ejecución manual de este proceso.

Considerando lo señalado se propone denominar este proyecto como **“DESARROLLO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA LA GESTIÓN Y SOPORTE DE LAS ACTIVIDADES DE CAMPO REALIZADAS POR EL PERSONAL OPERATIVO DE CONSTRUCCIONES E INSTALACIONES ELÉCTRICAS (CONIEL CIA.LTDA.)”**

Con esta aplicación se busca obtener una herramienta con opciones de tareas comunes realizadas por el personal operativo que necesitan soporte para un eficiente tratamiento de la información en lo posterior, haciendo uso de las nuevas tecnologías y del auge de las aplicaciones móviles para brindar un mejor desarrollo en las actividades de la compañía CONIEL CIA.LTDA.

# MARCO REFERENCIAL

## PLANTEAMIENTO DEL PROBLEMA

La compañía de Construcciones e Instalaciones Eléctricas CONIEL CIA.LTDA es una empresa dedicada a trabajos de Ingeniería Eléctrica, siendo su punto fuerte los contratos de pérdidas y control de energía adjudicados a través del Sercop (www.compraspublicas.gob.ec) por la CNEL EP Unidad de Negocios El Oro.

En CONIEL CIA.LTDA, las actividades operativas en el campo son realizadas por cuadrillas de trabajo conformadas por tres personas encargadas de trasladarse a lugares estratégicos o determinados anticipadamente por CONIEL para la realización de trabajos de electrificación correspondientes a control de pérdidas de energía y abastecimiento del servicio a zonas o abonados determinados, cada cuadrilla consta de un anotador, que es el encargado de capturar fotos a razón de justificar cada acción a tomar y llenar plantillas de datos (también llamadas fichas) con información relevante de las operaciones que se realicen, las mismas que pueden ser: servicios nuevos, cambios de medidor o mantenimientos del servicio dependiendo de las necesidades del cliente y lo notificado a realizar.

Dicha información al final de día es trasladada a la oficina central de la compañía para su posterior procesamiento, ingreso al Sistema SICO (Sistema Comercial de la CNEL EP) y tabulación.

### PROBLEMA CENTRAL

La gestión de información actual desde el sitio de trabajo, genera retraso en los procesos productivos y disminuye el porcentaje de confiabilidad e integridad de la información registrada e ingresada en la empresa.

### PROBLEMAS PARTICULARES

* En la empresa CONIEL CIA.LTDA las TIC’s son desconocidas y no utilizadas por el personal.
* En ocasiones el personal operativo no cuenta con información necesaria para el registro de la actividad en la ficha (formato para el registro de la actividad realizada) desde el lugar de trabajo y requiere de ayuda del personal administrativo.
* La toma de Información en el sitio muchas veces es inconsistente e incompleta.
* Los procesos realizados por el personal operativo son netamente manuales.
* La empresa posee computadoras potentes que pueden ser utilizadas para la implementación de un servidor, pero únicamente son usadas para la tabulación, búsqueda y registro de la información.
* La empresa no cuenta con la información de ubicación de cada cuadrilla a lo largo del día.

## PREGUNTAS CIENTIFICAS

### PREGUNTA CENTRAL

¿La implementación de la aplicación móvil para la optimización de las actividades realizadas por el personal operativo permitirá gestionar la información desde el sitio de trabajo y brindará mejoras en los procesos productivos de la compañía CONIEL CIA.LTDA?

### PREGUNTAS COMPLEMENTARIAS

* ¿Con la capacitación al personal de la empresa Coniel Cia. Ltda. se logrará un mejor uso de las TIC’s?
* ¿Se logrará mejorar el acceso a la información desde el sitio de trabajo?
* ¿Se mejorará la integridad y confiabilidad de la información?
* ¿La aplicación móvil permitirá llevar una gestión automatizada de los procesos que realiza el personal operativo?
* ¿Se implementará un servidor haciendo uso de la potencia del hardware que posee la empresa?
* ¿A lo largo del día se contará con la información de ubicación donde se encuentra cada cuadrilla?

## OBJETIVOS

### OBJETIVO PRINCIPAL

* Implementar una aplicación móvil para la gestión y soporte de las actividades de campo realizadas por el personal operativo de CONIEL Cia.Ltda utilizando la metodología Mobile-D”.

### OBJETIVOS ESPECIFICOS

* Capacitar al personal con el fin de lograr la actualización de conocimientos en el campo de nuevas tecnologías mediante cursos impartidos por profesionales en el tema
* Dar soporte necesario de información al personal operativo en los distintos puntos de trabajo mediante un módulo de búsqueda de datos para de esta manera abolir el sistema actual basado en fichas y obtener una mejor confiabilidad de la información registrada.
* Diseñar una aplicación móvil para automatizar los procesos manuales realizados por el personal operativo mediante la herramienta de desarrollo Android Studio.
* Implementar un servidor interno de la empresa con el fin de centralizar la información mediante el uso objetivo de nuevas tecnologías.
* Brindar geolocalización de los grupos de trabajo para llevar un mejor control de los mismos mediante la implementación de un módulo de monitorización que se añadirá al servidor web interno.

## ALCANCE

Previo el desarrollo del proyecto de tesis se realizará una investigación en la compañía de Construcciones e Instalaciones Eléctricas Coniel Cia.Ltda., poniendo énfasis a los procesos que realiza el personal operativo durante sus actividades diarias e identificar y analizar los problemas que tienen con el manejo de la información, para de esta manera integrarlos en una aplicación que permitirá un eficiente tratamiento de los datos desde el sitio de trabajo y mejorar la confiabilidad e integridad de la información.

La aplicación será diseñada de tal manera que sea intuitiva y de fácil manejo, para que el personal operativo pueda acceder a las opciones de forma sencilla, pudiendo así obtener mejores resultados en la ejecución de las actividades en tiempo real.

Para la realización del proyecto de titulación se utilizará un lenguaje de programación Java, el IDE Android Studio, base de datos PostgreSQL y dispositivos móviles con Sistema Operativo Android, además para un mejor control y gestión del proyecto se usará la metodología ágil Mobile-D que contemplará las siguientes acciones:

|  |  |  |  |
| --- | --- | --- | --- |
| **MÓDULO** | **FUNCIÓN** | **SUBFUNCIÓN** | **DESCRIPCIÓN** |
| **Login de la Aplicación** | Ingreso de usuario y contraseña | Validación y Verificación de los datos | Valida y verifica los datos ingresados, la disponibilidad de la sesión, las sesiones activas en el servidor y finalmente da apertura a una sesión.  También comprueba si el usuario que se encuentra realizando la petición de ingreso a la aplicación tiene acceso al contrato que ha seleccionado.  La persona encargada de recolectar la información de la actividad realizada ingresa a la aplicación mediante un usuario y una contraseña y accede al menú principal. |
| Comprobación de la identidad del usuario ingresado |
| Ingreso a la aplicación |
| Comprobación de Sesiones del servidor |
| Comprobación de sesiones disponibles |
| Apertura de Sesión |
|  | Comprobación de acceso a contrato |
| **Ingreso de Actividades** | Ingreso de información de las actividades realizadas | Búsqueda, Validación y Verificación de datos de abonado y de información de medidores | El personal operativo puede ingresar información necesaria de la actividad realizada, así mismo puede buscar datos del abonado y de los medidores asignados a él en el sistema comercial SICO, además puede capturar las imágenes para constatar la ejecución de su trabajo. Puede buscar las actividades realizadas para verificar si ya han sido ingresadas. |
| Búsqueda de actividades realizadas |
| Ingreso de detalle de instalación del servicio |
| Captura de fotografías de la actividad realizada |
| **Búsqueda** | Búsqueda de información de abonados por criterio en el sistema comercial SICO. | Búsqueda de datos por cuenta (código único de abonado) | Este módulo realiza búsquedas por criterios al sistema comercial SICO, y provee información necesaria para el personal operativo. |
| Búsqueda de datos por medidor |
| Búsqueda de datos por nombre de abonado |
| Búsqueda de datos por geocódigo |
| **Fotos** | Capturar fotografías de las actividades realizadas | Ingreso de cuenta para almacenar las fotografías | Este módulo permite capturar las fotografías de la actividad realizada, así mismo seleccionar la fecha para crear una nueva cuenta en la que almacenará las imágenes o seleccionar una cuenta de la lista a la que desee añadir más fotos o eliminar alguna(s) de la galería.  También puede seleccionar una imagen de la galería de la cuenta seleccionada para visualizar en tamaño real. |
| Capturar imágenes |
| Selección de fecha y de cuenta para captura de imágenes |
| Añadir más imágenes a una cuenta seleccionada |
| Eliminar imágenes seleccionadas |
| Visualización de imágenes |
| **Geolocalización** | Detectar ubicación del sitio en el que se encuentran las cuadrillas | Detectar ubicación actual de las cuadrillas | Esta sección de la aplicación permite tener un mejor control de la ubicación actual de las cuadrillas, además de conocer su propia ubicación y obtener la ruta más corta hacia una cuadrilla especifica. |
| Detectar mi ubicación |
| Ruta más cercana a una cuadrilla especifica |

**Tabla 1.** Detalle de alcance del proyecto

**CÁPITULO II**

# MARCO TEÓRICO

## ANTECEDENTES HISTÓRICOS DE LA EMPRESA

### RESEÑA HISTÓRICA DE CONIEL CIA.LTDA

La compañía de Construcciones e Instalaciones Eléctricas CONIEL CIA.LTDA, tiene sus inicios debido a la gran demanda de servicios eléctricos en nuestra Provincia. Fue constituida el 02 de julio del 2002, resultado de la asociación entre el Tnlg. Julio Loaiza y el Sr. Luis Pérez, para de esta manera dar paso al crecimiento de la empresa Privada cuyo único objetivo era ganar experiencia en el ámbito profesional y brindar servicios de calidad a la ciudadanía. Con el pasar de los años, la compañía es fuertemente constituida y cuenta con un amplio número de contratos firmados con la CNEL EP, los mismos que conforman su experiencia profesional. La empresa ha cumplido a cabalidad con las actividades de la prestación de los servicios adjudicado con la CNEL EP. La Compañía ha ido creciendo y ganando terreno en la Provincia y haciéndose acreedora a muy buenas referencias de trabajo por parte de la CNEL EP.

Actualmente es una compañía fuertemente establecida que brinda servicios a la comunidad trabajando en conjunto con la CNEL EP. (Coniel Cia.Ltda, 2013)

### UBICACIÓN



**CONIEL CIA.LTDA**

**Figura 1.** Croquis de ubicación de la Empresa Coniel Cía. Ltda.

***Fuente:*** Coniel Cia.Ltda

**Dirección:** Machala, Santa Rosa e/ Arízaga y Gral. Manuel Serrano.  
**Teléfono:** 2938-581

### ORGANIGRAMA ESTRUCTURAL

**ORGANIGRAMA DE FUNCIONES**

**GERENTE**

(Planificación control en la ejecución de los trabajos)

**ABOGADO**

Persona encargada de los reglamentos, normas estatales que permiten la eficiencia y constitución de la misma.

**CONTADOR**

Ente regulador de las normas tributarias y responsable del control de la compañía.

**REPRESENTANTE TÉCNICO**

(Fiscalizador de los Trabajos a ejecutar)

**SECRETARIA**

(Recepción de información y transferencia de información, novedades, reportes de las mismas)

**DIGITADORES**

(Recepción de información, digitación a través del sistema SICO novedades, reportes de las mismas)

**SUPERVISORES**

(Enrutamiento de servicio, control del personal y de las acciones a cumplir de la gestión.)

**GRUPO 1**

**(4 personas)**

**(4 )**

**Bachilleres en Electricidad para ejecutar los trabajos de:**

* Cambio de Medidores - Acometidas
* Corte y Reconexión de Energía Eléctrica, etc.

Herramientas: Un radio para comunicación, una escalera, cinturones, guantes, cascos, conos, cable, pinzas, destornilladores, playo etc.

**Vehículo**

Cámara fotográfica, equipo de comunicaciones.

**PERSONAL OPERATIVO**

**DE ESTA MANERA SE INTEGRAN LOS GRUPOS DE TRABAJOS**

**Figura 2.** Estructura Organización de CONIEL Cia.Ltda.

***Fuente:*** Coniel Cia.Ltda

## ANTECEDENTES CONCEPTUALES

### TELEFONÍA MOVIL

#### Concepto

“La telefonía fija o móvil, es aquella que hace referencia a las líneas y equipos que se encargan de la comunicación entre terminales telefónicos, generalmente enlazados entre ellos con la central por medio de conductores metálicos.” (Procoop, 2010)

“La telefonía celular es un sistema de comunicación telefónica totalmente inalámbrica, en este caso los sonidos se convierten en señales electromagnéticas, que viajan a través del aire, siendo recibidas y transformadas nuevamente en mensaje a través de antenas repetidoras o vía satélite.” (Botero, 2012)

La telefonía móvil es un sistema que permite el acceso a un nuevo medio de comunicación con el objeto de proveer el contacto entre usuarios.

#### Historia

Martínez (2001) afirma que en los Estados Unidos al introducir el primer radioteléfono en 1973 se lo consideró a Martin Cooper quien trabajaba en Motorola como el “padre de la telefonía celular” ya que fue el pionero en esta tecnología, pero fue en Tokio Japón en 1979 donde aparece el primer sistema comercial lanzado por la compañía NTT (Nippon Telegraph & Telephone Corp). En el año de 1983 la entidad reguladora de Estados Unidos adopta reglas para la creación del servicio de telefonía y pone en marcha el primer sistema comercial de telefonía celular en la ciudad de Chicago. Desde entonces la tecnología inalámbrica tuvo gran aceptación en varios países ya que la vieron como una alternativa a la telefonía convencional alámbrica, por lo que a los pocos años de utilizarse se empezó a saturar el servicio, y esto llevo a la necesidad de transformar los sistemas analógicos a digital con el fin de dar acceso a más usuarios, además de implementar técnicas que permitan el acceso múltiple al canal. (pág. 2)

#### Dispositivos Móviles

“Los dispositivos móviles son aparatos de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales.” (Álvarez, 2008)

“Son aquellos suficientemente pequeños para ser transportados y empleados durante su transporte. Normalmente se sincronizan con un sistema de sobremesa para actualizar aplicaciones y datos, es aquel que es capaz de comunicarse o acceder a una red sin cables.” (Fernández, 2006)

Son dispositivos que permiten acceder a los servicios que brinda la telefonía móvil o celular a través de conexiones inalámbricas, son de fácil de transportar y permiten el manejo de información desde cualquier sitio.

1. **Características**

Para Guevara (2010), los dispositivos móviles se caracterizan por ser aparatos pequeños de memoria limitada, fáciles de movilizar, con diseños específicos destinados a una función principal, capacidades especiales para el procesamiento, conexión permanente a una red, y además de versatilidad para el desarrollo de otras funciones.

1. **Ventajas**

Las ventajas de un dispositivo móvil son la variedad de funcionalidades que posee, la fácil portabilidad, y la posibilidad de mantener una conexión a una red permanente. (Rabajoli, 2007)

#### Aplicaciones móviles

Según Quintanilla (2013), las aplicaciones móviles son programas que brindan una variedad de servicios como de entretenimiento, de información, que además son pensadas para facilitar la consecución de una tarea determinada o asistir en operaciones y gestiones del día a día.

De acuerdo con Reynaga (2013), una aplicación móvil es un software diseñado para ser ejecutada en dispositivos móviles inteligentes, y que además para el desarrollo de las mismas se deben tomar en cuenta una gran variedad de características como el tamaño de la pantalla, datos específicos de software y configuraciones.

### SISTEMAS OPERATIVOS PARA DISPOSITIVOS MOVILES

Rivera (2012) menciona que el sistema operativo móvil se diferencia con el de una pc tradicional en que no se pueden ejecutar aplicaciones robustas sobre él, pero en general se desempeña con rapidez. Es una plataforma que interpreta lo que el usuario desea que la terminal realice y lo ejecuta con inteligencia, esto permite la interacción real con lo que se puede hacer a través de las capacidades del hardware de un equipo.

Además el sistema operativo permite la gestión de los recursos entre las aplicaciones que los conforman, son más simples, orientados hacia la conectividad inalámbrica y a necesidades específicas.

#### Sistemas Operativos para móvil

Montoya (2012) explica los siguientes sistemas operativos para móvil:

|  |  |
| --- | --- |
| Sistema Operativo | Descripción |
| Android | Desarrollado por Android Inc, firma que en el 2005 fue comprada por Google. Es un sistema operativo para teléfonos inteligentes. |
| Symbian | Este sistema operativo para terminales móviles se creó con el objetivo de competir con otros sistemas operativos del mismo propósito como el de Palm o el Windows Phone de Microsoft y ahora Android de Google Inc. Y iOS de Apple Inc. |
| Ios | IOS es un sistema operativo móvil basado en la manipulación directa, es decir incluye gestos como deslices, toques, pellizcos, lo que hace posible la interacción a través del contexto de la interfaz. |
| BlackBerry OS | Da soporte para varios métodos de entrada como touchpad y pantallas táctiles, además permite la ejecucion de varias tareas a la vez. |
| Windows Phone | Fue desarrollado por Microsoft, es un sistema operativo móvil compacto con interfaz de usuario natural destinado para +ser utilizado en teléfonos inteligentes. |

**Tabla 2.** Sistemas Operativos para móvil

***Fuente:*** Montoya (2012)

### SISTEMA OPERATIVO ANDROID

#### Conceptos

Ribas (2013) afirma que:

Android es una plataforma de desarrollo libre, y de código abierto: El núcleo del sistema está basado en un Linux (versión 2.6 para versiones 3.0 del kernel para posteriores) al que se le han hecho ciertas modificaciones para que pueda ejecutarse en teléfonos y terminales móviles. Android es el nombre esencial para un sistema operativo enfocado al uso del mismo en dispositivos móviles, tomando en cuenta que al inicio se lo creo para ser usado solamente en teléfonos celulares. En la actualidad se puede encontrar a Android en todo tipo de dispositivos como tablets, Smartphone, netbooks, entre otros. (pág. 18)

Paredes Velasco, Santacruz Valencia y Domínguez Mateos (2012) consideran que:

Android es una plataforma formada por un conjunto de software en estructura de pila (software stack) que incluye un sistema operativo, software para conectar aplicaciones (middleware) y aplicaciones base. El SDK (Software Development Kit, Kit de Desarrollo de Software) de Android proporciona varias herramientas y API (Applications Programming Interface, Interfaz de Programación de Aplicaciones) que son necesarias para desarrollar aplicaciones Android. (pág. 15)

Android es un sistema operativo para teléfonos móvil basado en el Kernel de Linux, de código libre, posee una amplia gama de aplicaciones disponibles. Permite el acceso a aplicaciones propias del sistema operativo como a (GPS, Cámara, Llamadas, Mensajes, etc), además de que da acceso a escritura y lectura a memoria tanto interna como externa.

#### Historia

Según Lara Cancela y Sara Ostos (2012), en el 2005 Google compra la firma Android Inc., y en ese entonces promociono a fabricantes de dispositivos y operadoras una plataforma para dispositivos móviles basada en el kernel de Linux que provee un sistema flexible y actualizable. La Open Handset Alliance, es una alianza dedicada al desarrollo de estándares abiertos para dispositivos móviles, que está conformada por un sin número de empresas entre fabricantes, desarrolladores y operadores de servicios, quienes adoptaron como principal producto el buscador y las aplicaciones para el uso en móviles de google.

De acuerdo con Castillo (2012), la compañía Android Inc. quien estaba desarrollando el sistema operativo para móvil Android, fue adquirida en el 2005 por Google. Por lo que en el 2008, el dispositivo T-Mobile fabricado por HTC y desarrollado conjuntamente con Google, fue el primero en poseer el sistema operativo en su versión 1.0 de Android.

#### Características

Para Ribas (2013), android posee grandes características como el soporte para audio, video y formato de imágenes, servicio de telefonía, permite la reutilización y la sustitución de los componentes, usa la máquina virtual Dalvik optimizada para dispositivos móviles, tiene un navegador integrado, permite el almacenamiento de datos estructurados a través de SQLite, y provee de servicios como Bluetooh, Wi-fi, Cámara, GPS. (pág. 19)

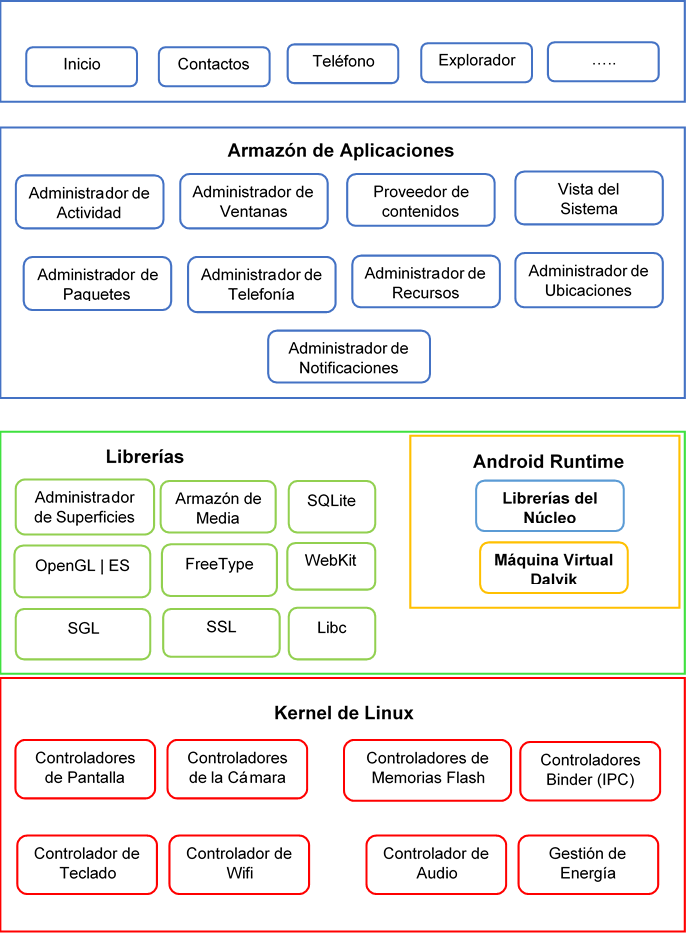
#### Arquitectura de Android

De acuerdo con Ribas (2013), el sistema operativo Android posee una arquitectura conformada por capas de software, en la que en la parte inferior hay un conjunto de drivers que se basan en Linux, y también un conjunto de librerías que pueden ser accedidas a través del nivel superior denominada Framework de aplicaciones. Además posee una pila de software en la que se encuentran el sistema operativo, middleware y aplicaciones básicas para el usuario.(pág. 42)

|  |  |
| --- | --- |
| Capa | Descripción |
| Aplicaciones | En esta capa se incluyen las aplicaciones básicas del dispositivo |
| Framework de Aplicaciones | Esta capa permite a las librerías javas el acceso a los recursos de niveles anteriores a través de la máquina virtual de Dalvik. Además está formada por clases y servicios que usan las aplicaciones para la ejecucion de un trabajo. |
| Las librerías nativas | Están escritas en C o C++, se compilaron para la arquitectura de hardware usadas por el dispositivo, además fueron instaladas en el por el proveedor del mismo. Son contenidas en la capa inmediata superior al kernel de Linux. |
| Runtime | En un dispositivo móvil se pueden correr múltiples máquinas virtuales de manera eficiente, ya que cada proceso de una aplicación obtiene una instancia a la máquina virtual. |
| Kernel Linux | El manejo de la memoria, el control de los procesos u otros servicios relativos al sistema operativo están basados en Linux, ya que provee una capa de abstracción de hardware para Android. |

**Tabla 3.** Descripción de las capas de la Arquitectura de Android

***Fuente:*** Ribas (2013), págs. 42-43



**Framework de Aplicaciones**

**Aplicaciones**

**Figura 3.** Arquitectura de Android

***Fuente:*** Adaptado de Ribas (2013, pág. 44

#### Ventajas y Desventajas

Según Santa María (2014), Android es un sistema operativo de código libre, que cuenta con un sin número de aplicaciones disponibles para descargar, además de que es un sistema compacto que permite la ejecución de varias tareas a la vez, aunque cabe destacar que así como posee grandes cualidades, una de sus desventajas es la poca duración de la batería.

#### Versiones

Martínez (2013) detalla las versiones del sistema operativo Android:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Versión | Fecha de Lanzamiento | Descripción | Logo |
| Apple Pie | 1.0 | 23 Septiembre del 2008 | Fue lanzado el 23 de septiembre de 2008, el primer smartphone en el que fue instalado es el HTC Dream. Tenía soporte para aplicaciones básicas como SMS y cámara. | D:\Datos\Pictures\Abecedario-versiones-Android.jpg |
| Banana Bread | 1.1 | 9 Febrero del 2009 | En esta versión se dio soporte para diseños de sistemas básicos, además de la posibilidad de adjuntar archivos en mensajes. | http://tuchillan.cl/wp-content/uploads/2013/08/original.jpg |
| Cupcake | 1.5 | 30 Abril del 2009 | Contaba con un rediseño completo en su interfaz, además de transiciones animadas, mejoras en la velocidad de la cámara, teclado en la pantalla y soporte de bluetooth stereo. |  |
| Donut | 1.6 | 15 Septiembre de 2009 | Fue una pequeña actualización, pero vino empaquetaba con un cuadro de búsqueda mejorado, cámara y aplicación de galería, y una renovada Android Market. | D:\Datos\Pictures\Abecedario-versiones-Android.jpg |
| Eclair | 2.0–2.1 | 25 Octubre de 2009 | Se rediseño la interfaz del navegador, se da soporte nativo de flash para la cámara, zoom digital en las fotos. | D:\Datos\Pictures\Abecedario-versiones-Android.jpg |
| Froyo | 2.2 | Mayo de 2010 | Incorpora el motor de Java V8 y ofrece a los usuarios un aumento de velocidad gracias al compilador JIT que permite iniciar las solicitudes más rápido y además de mejorar el rendimiento general del sistema. | D:\Datos\Pictures\Abecedario-versiones-Android.jpg |
| Gingerbread | 2.3 | 6 Diciembre de 2010 | Se vio un considerable incremento en la velocidad de ejecucion de tareas, además se renovó el diseño de la interfaz. | D:\Datos\Pictures\Abecedario-versiones-Android.jpg |
| HoneyComb | 3.0–3.1 | 22 Febrero de 2011 | Nueva interfaz de usuario con una barra de sistema en la parte inferior de la pantalla que permitía el acceso rápido a notificaciones, estados y botones de navegación suavizados y el Action Bar. | D:\Datos\Pictures\Abecedario-versiones-Android.jpg |
| Icecream Sandwich | 4.0 | 19 Octubre 2011 | Se implementó el diseño Holo en las interfaces de usuario e íntegro el sistema operativo en sus versiones para Tablets y Smartphones. | D:\Datos\Pictures\Abecedario-versiones-Android.jpg |
| Jelly Bean | 4.1 | Junio 2012 | Mejoro la fluidez de las interfaces de usuario, brindando estabilidad y mejor rendimiento de la misma. | D:\Datos\Pictures\Abecedario-versiones-Android.jpg |
| KitKat | 4.4 | Diciembre 2013 | Es la versión vigente del sistema operativo para móviles, con una interfaz impecable. | D:\Datos\Pictures\Abecedario-versiones-Android.jpg |
| Lollipop | 5.0 | Noviembre 2014 | Este posee soporte para 64 bits y funciona en teléfonos, tablets y tvs. | **lillipop** |

**Tabla 4.** Versiones del Sistema Operativo Android

***Fuente:*** Adaptado de Martínez (2013)

#### Componentes de una Aplicación Android

Ribas Lequerica (2013) considera que:

Para diseñar una aplicación en Android, es necesario tener claros los elementos que la componen y la funcionalidad de cada uno de ellos. Android trabaja en Linux, y cada aplicación utiliza un proceso propio. Se distinguen por el ID, un identificador para que solo ella tenga acceso a sus archivos. Los componentes son los elementos básicos con los que se construyen el proyecto. Habrá tantas actividades como ventanas distintas tenga la aplicación. (pág. 41)



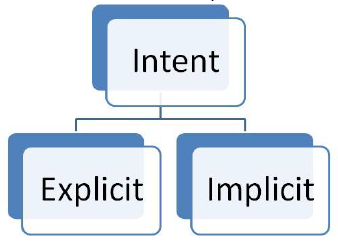
**Figura 4.** Componentes de una Aplicación Android

***Fuente:*** Ribas (2013), pág. 41

|  |  |
| --- | --- |
| **Componente** | **Descripción** |
| **Activity** | Una Activity es una clase que permite representar la parte gráfica de la aplicación es decir es donde se mostraran las vistas (Views). |
| **Broadcast Intent Receivers** | Es un receptor de mensajes emitidos por el sistema. |
| **Service** | Es un componente que se ejecuta en background, y que periódicamente se conecta a un servidor para buscar cambios en la información. |
| **Content providers** | Provee una capa de abstracción para acceder a los datos almacenados de tal manera que sean accesibles entre aplicaciones. |
| **Fragment** | Su cometido principal es la reutilización tanto de código de lógica de trabajo como de las interfaces de esos códigos. |
| **Intents** | Los intentos son objetos de la clase que contiene datos del mensaje a transmitir. |

**Tabla 5.** Descripción de los componentes de una Aplicación Android

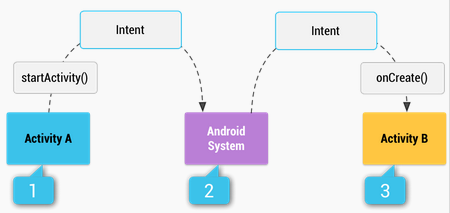
***Fuente:*** Adaptado de (Ribas Lequerica, 2013, pág. 49)



**Fuente:** Libro Desarrollo de Aplicaciones para Android 2013

**Figura 5.** Tipos de Intent

* ***Intent Explícito:***Este tipo de Intent es aquel en el que se conoce exactamente qué componente lanzar, el cual especificaremos a través del nombre de componente. Se lo utiliza cuando se desea lanzar un servicio o actividad, típicamente de nuestra propia aplicación.
* ***Intent Implícito*:**En este caso, no se conoce el nombre de componente (estará vacío), pero si se sabe la acción que se desea ejecutar. De esta forma, se le pide al sistema operativo que busque por usted qué aplicación podría realizar dicha acción y, en el caso de que haya varias, nos dará a elegir cuál utilizar. (pág. 50)

******

**Figura 6.** Ejemplo de ejecución de Intent

***Fuente:*** Libro Desarrollo de Aplicaciones para Android 2013

#### Estado de los procesos

De acuerdo con Ribas (2013), los procesos son situados en orden jerárquico por Android, basándose en sus estados. Cada aplicación de Android corre su propio proceso, el cual se crea cuando se ejecuta la aplicación y permanece ahí hasta que deja de trabajar o hasta cuando el sistema requiere de memoria para ejecutar otras aplicaciones, la jerarquía en la que se basa Android es la siguiente:

* **Actividad de primer plano (Foreground Activity):** Es el Activity con la que el usuario interactúa.
* **Actividad Visible:** Es la Activity que se ejecuta algo así como un cuadro de dialogo, que aunque no es

Aquella actividad que es visible al usuario pero no es la Foreground Activity, es decir, tiene algo por encima de ella en pantalla, algo como un cuadro de diálogo que aunque no permite interactuar con ella sí permite verla.

* **Actividad de Fondo (Background Activity):** Son las actividades que el usuario no ve y que han sido pausadas.
* **Proceso vacío:** Son los procesos que no tienen asociados ninguna actividad ni dependen de ningún otro componente (Service o BroadcastReceiver). Son los primeros en ser sacrificados por el sistema si se necesitan recursos. (págs. 54 - 55)

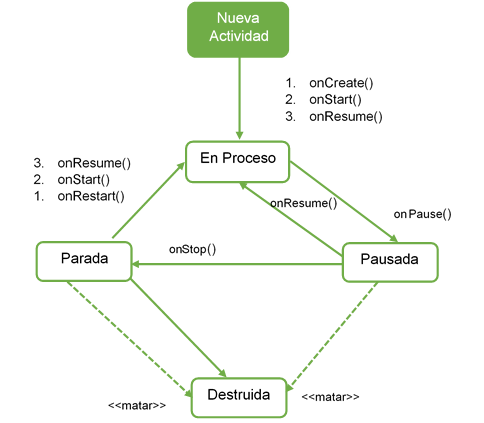
#### Ciclo de vida de una actividad

Ribas Lequerica (2013) afirma que:

Los componentes de las aplicaciones tienen unos ciclos de vida que dependerán de la situación en la que se encuentre en cada momento la aplicación, desde que se crea y es capaz de responder a eventos hasta que se destruye y se liberan todos los recursos utilizados, la aplicación pasara por diferentes estados.

A lo largo de una ejecución normal de una aplicación, sus Activity las podemos encontrar en alguno de los siguientes cuatro estados:

* **Activada:** Es cuando el usuario ve la actividad y puede interactuar con ella desde la pantalla, o dicho de otro modo, cuando está la primera pila de ejecución.
* **Pausada:** Cuando la actividad ha pasado ya a segundo plano, pero aun esta visible, es cuando otra actividad se coloca sobre la actividad que pasa a pausa, pero la nueva actividad no tapa del todo a la actividad anterior (bien porque sea transparente o porque sea de menor tamaño su interfaz); la actividad pausada pierde el foco pero se ve parte de ella.
* **Parada:** Cuando la actividad pasa a segundo plano y además está totalmente tapada por la nueva actividad, es decir queda totalmente eclipsada por la nueva interfaz.
* **Destruida:** La actividad no está ya disponible, se han liberado todos sus recursos y en caso de ser llamada, necesitaría comenzar un nuevo ciclo de vida. (págs. 50 - 51)



**Figura 7.** Estados de una Activity

***Fuente:*** Libro Desarrollo de Aplicaciones para Android 2013, pág. 52

***Editado:*** Andrea Loaiza

Según Ribas Lequerica (2013) los métodos para la gestión del ciclo de vida son:

* **onCreate()**: Es llamado cuando la actividad arranca por primera vez. Puede ser utilizado para realizar tareas de inicialización como crear la interfaz de usuario.
* **onStart()**: Se ejecuta cuando la aplicación va a ser visible al usuario.
* **onResume()**: Se ejecuta antes de que el usuario pueda interactuar con la Activity. En este momento la Activity se encuentra en la parte superior de la pila de ejecución.
* **onPause()**: Este método se ejecuta cuando la actividad está a punto de pasar a segundo plano, usualmente debido a que otra actividad ha sido lanzada frente a ella.
* **onStop()**: Este método es llamado cuando la actividad ya no está visible para el usuario y no será requerida por un tiempo.
* **onRestart()**: Si este método es llamado significa que una actividad que estaba detenida está volviendo a ser desplegada al usuario.
* **onDestroy()**: Es llamado justo antes de que la actividad sea destruida. Durante la destrucción de la Activity se perderán todos los datos asociados a ella, de modo que si vuelve a ser llamada se ejecutara un nuevo ciclo de vida, por lo que es uno de los lugares para controlar la persistencia de datos.
* **onSaveInstanceState()**: Android mandará llamar este método para permitir a la actividad el guardar un estatus por instancia (como la posición del cursor en un campo de texto). Usualmente no será necesario reescribir este método ya que la implementación por defecto guarda el estado de todos los controles de la interfaz de manera automática.
* **onRestoreInstanceState()**: Este método es llamado cuando la actividad está siendo reiniciada desde un estatus previamente almacenado en onSaveInstanceState().La implementación por defecto restaura el estado de la interfaz del usuario completamente. (pág. 53)

#### Acceso a Servicios web SOAP en Android

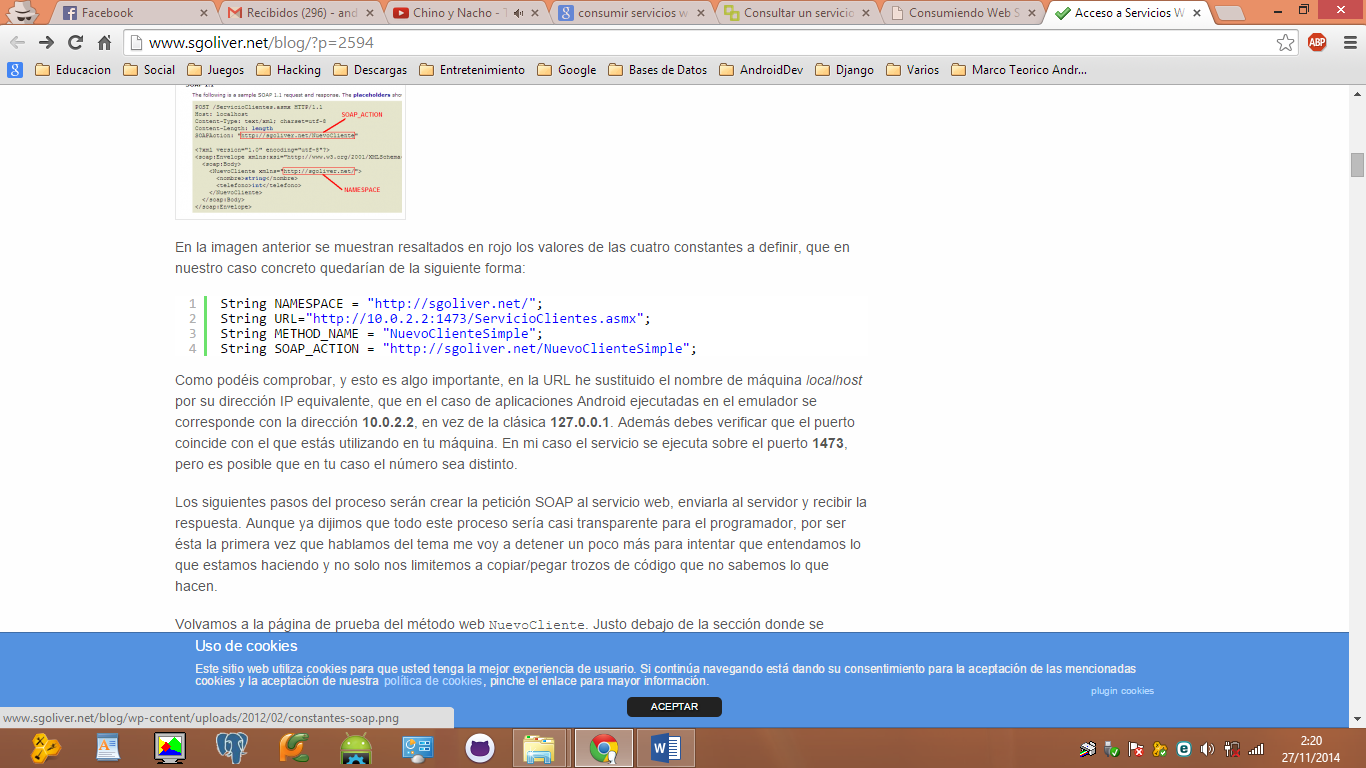
Gómez Oliver (2012) explica que:

Android no incluye ningún tipo de soporte para el acceso a servicios web de tipo SOAP. Es por esto por lo que se debe agregar una librería externa para hacer más fácil esta tarea. Entre la oferta actual, la opción más popular y más utilizada es la librería ksoap2-android. Esta librería es un fork, especialmente adaptado para Android, de la antigua librería kSOAP2. Este framework permitirá de forma relativamente fácil y cómoda utilizar servicios web que utilicen el estándar SOAP.

En la implementación del evento onClick del botón, que será el encargado de comunicarse con el servicio web y procesar el resultado, se definirán cuatro constantes que servirán en varias ocasiones durante el código:

* **NAMESPACE.** Espacio de nombres utilizado en nuestro servicio web.
* **URL.** Dirección URL para realizar la conexión con el servicio web.
* **METHOD\_NAME.** Nombre del método web concreto que vamos a ejecutar.
* **SOAP\_ACTION.** Equivalente al anterior, pero en la notación definida por SOAP.

Asignación de valores a las constantes:

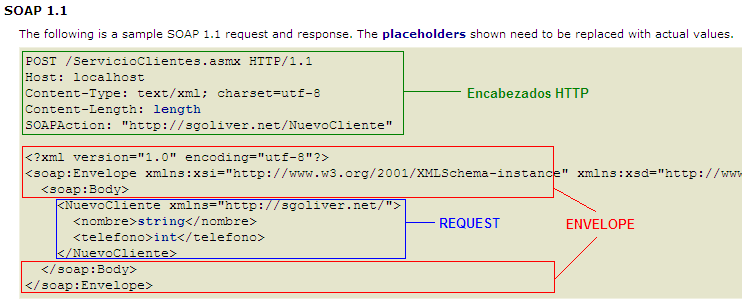


**Figura 8.** Ejemplo de asignación de valores a las constantes

***Fuente:*** http://www.sgoliver.net/blog/?p=2594

En la URL se ha sustituido el nombre de máquina *localhost* por la dirección IP equivalente. Además se debe verificar que el puerto coincide con el que se está utilizando en la máquina. En este caso el servicio se ejecuta sobre el puerto **1473**, pero es posible que en otras ocasiones el número sea distinto.Los siguientes pasos del proceso serán crear la petición SOAP al servicio web, enviarla al servidor y recibir la respuesta.

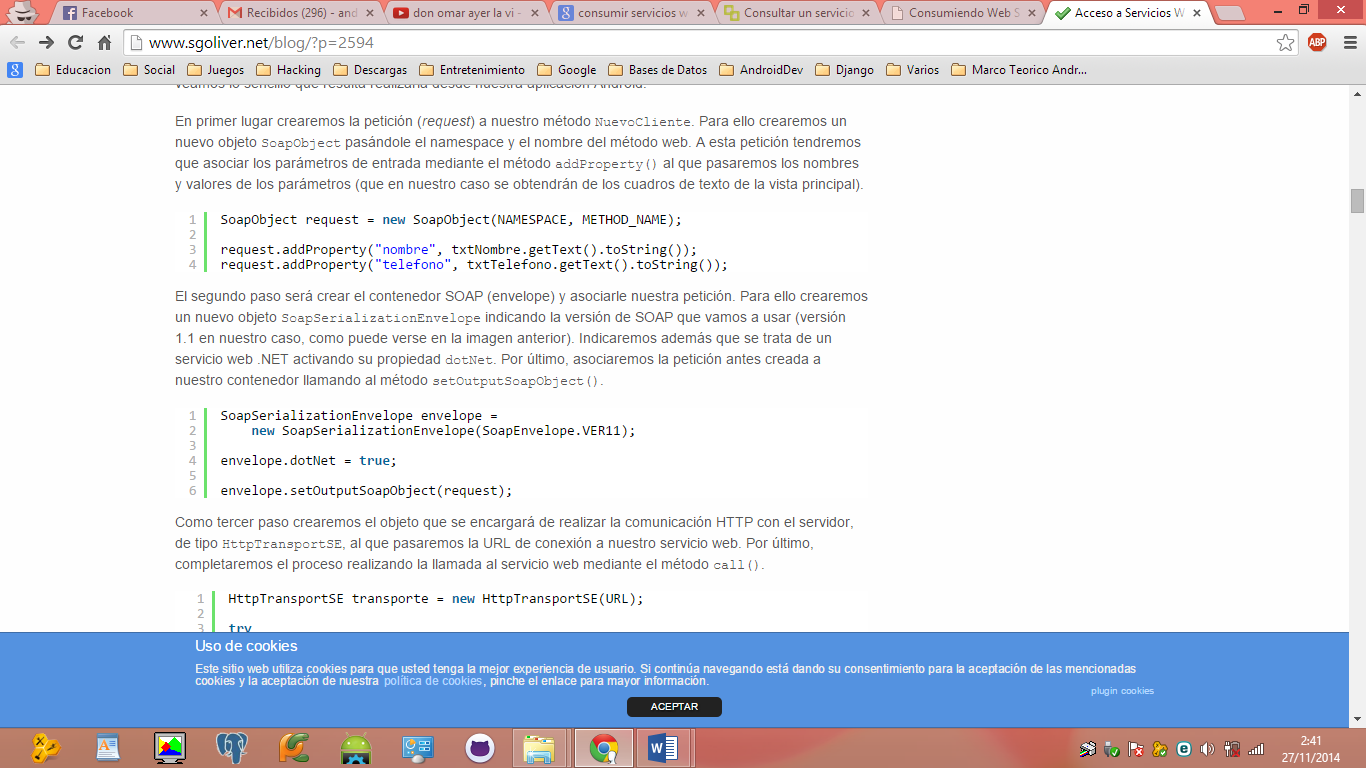
En la siguiente figura se observa que justo debajo de la sección donde se solicitan los parámetros a pasar al método se incluye también un XML de muestra de cómo tendría que ser la petición al servidor. Se ha marcado las tres partes principales de una petición de tipo SOAP. Empezando por la “parte interna” del XML, en primer lugar se encuentran los datos de la petición en sí (Request) que contiene el nombre del método al que se desea llamar, y los nombres y valores de los parámetros en entrada. Rodeando a esta información se añaden otra serie de etiquetas y datos a modo de contenedor estándar que suele recibir el nombre de Envelope. La información indicada en este contenedor no es específica de la llamada al servicio, pero sí contiene información sobre formatos y esquemas de validación del estándar SOAP. Por último, durante el envío de esta petición SOAP al servidor mediante el protocolo HTTP se añaden determinados encabezados como los que se observan en la imagen. Todo esto junto hará que el servidor sea capaz de interpretar correctamente la petición SOAP, se llame al método web correcto, y devuelva el resultado en un formato.



**Figura 9.** Ejemple de request y response

***Fuente:*** http://www.sgoliver.net/blog/?p=2594

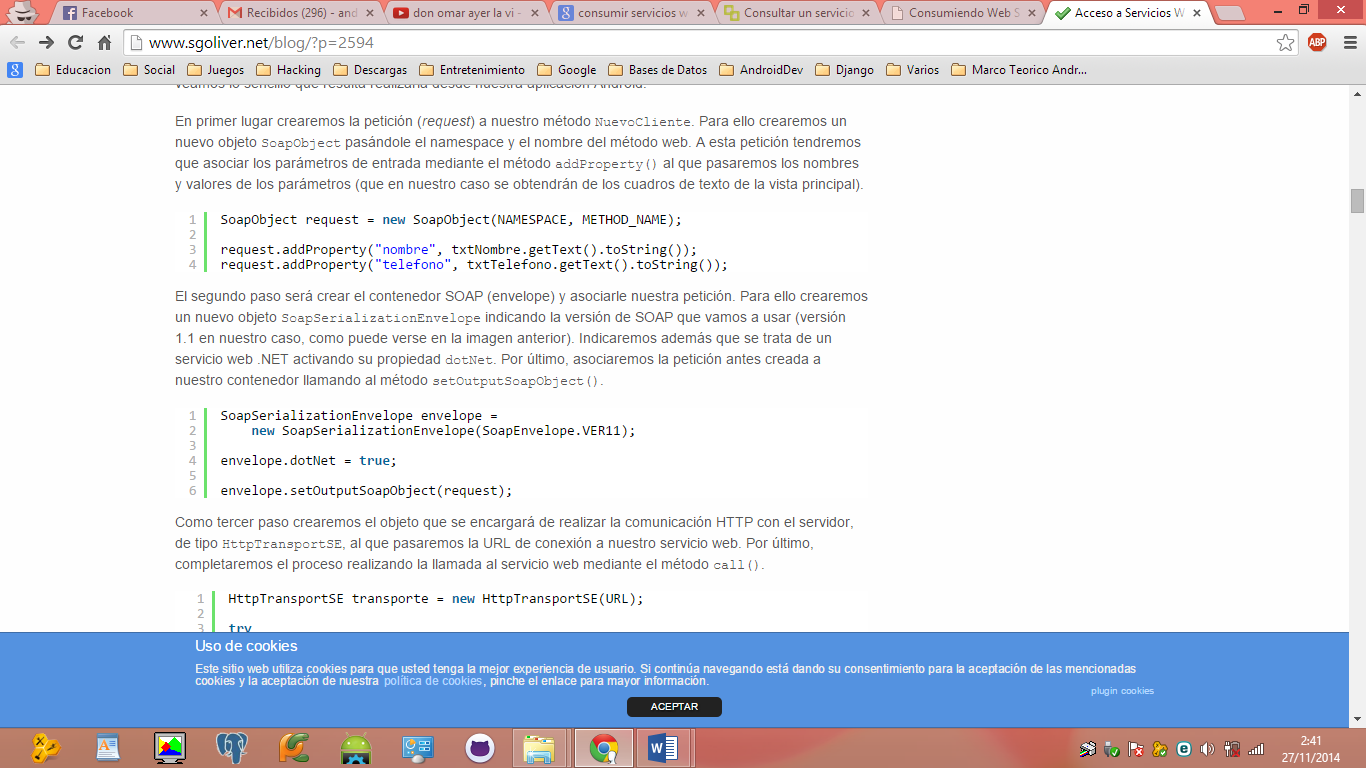
Para crear la petición (request) al método, se debe crear un nuevo objeto SoapObject pasándole el namespace y el nombre del método web. A esta petición se debe asociar los parámetros de entrada mediante el método addProperty() al que se pasarán los nombres y valores de los parámetros.



**Figura 10.** Ejemplo petición (request)

***Fuente:*** http://www.sgoliver.net/blog/?p=2594

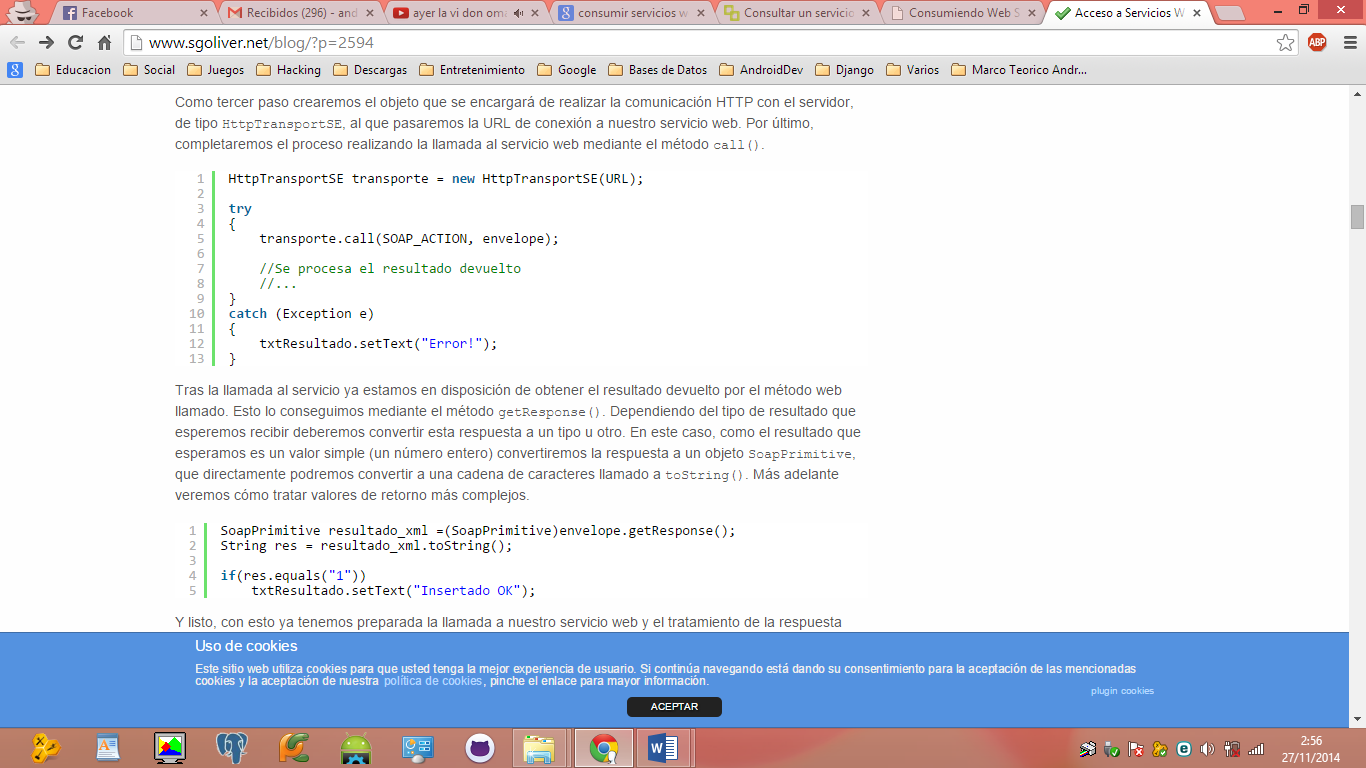
Se debe crear el contenedor SOAP (envelope) y asociarle la petición (request). Para ello se debe crear un nuevo objeto SoapSerializationEnvelope indicando la versión de SOAP que se va a usar (versión 1.1 en el caso de ejemplo). Además se debe indicar que se trata de un servicio web. Por último, se debe asociar la petición antes creada al contenedor llamando al método setOutputSoapObject ().



**Figura 11.** Ejemplo de creación del contenedor SOAP (envelope)

***Fuente:*** http://www.sgoliver.net/blog/?p=2594

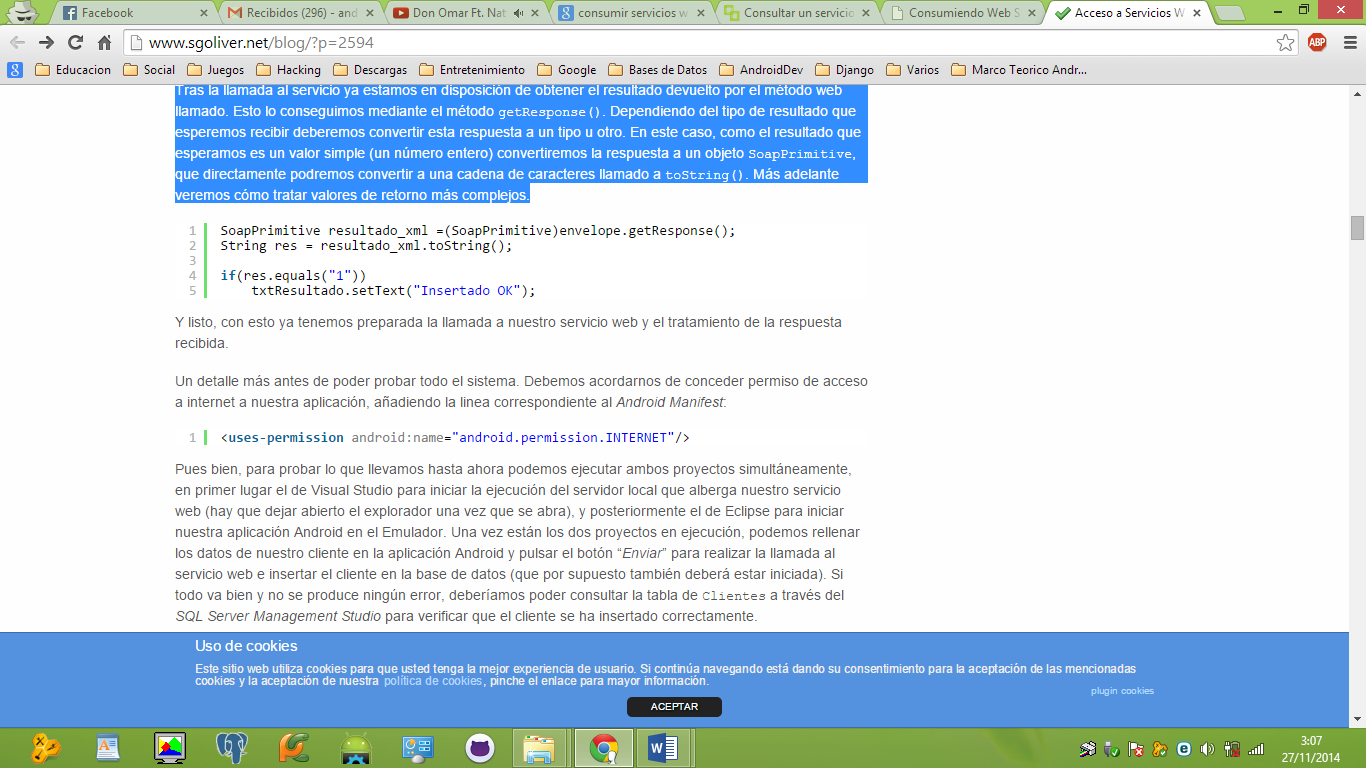
Ahora se crea el objeto que se encargará de realizar la comunicación HTTP con el servidor, de tipo HttpTransportSE, al que se pasará la URL de conexión al servicio web. Por último, se debe completar el proceso realizando la llamada al servicio web mediante el método call().



**Figura 12.** Ejemplo de objeto para comunicación HTTP

***Fuente:*** http://www.sgoliver.net/blog/?p=2594

Tras la llamada al servicio ya se puede obtener el resultado devuelto por el método web llamado. Esto se lo consigue mediante el método getResponse (). Dependiendo del tipo de resultado que se espere recibir se deberá convertir esta respuesta a un tipo u otro. En este caso, como el resultado que se espera es un valor simple (un número entero) se convertirá la respuesta a un objeto SoapPrimitive, que directamente se puede convertir a una cadena de caracteres llamado a toString ().



**Figura 13.** Ejemplo de método getResponse

***Fuente:*** http://www.sgoliver.net/blog/?p=2594

SOAP es un protocolo que permite especificar los objetos y métodos de un servicio web a los que el cliente tendrá acceso.

### PROGRAMACIÓN ORIENTADA A OBJETOS

#### Introducción

Según Llobet Azpitarte, Alonso Jordá, Miedes De Elías, Ruiz Fuertes y Torres Goterris (2008):

La orientación a objetos promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo de software:

La falta de portabilidad del código y su escasa reusabilidad.

* Código que es difícil de modificar.
* Ciclos de desarrollo largos.
* Técnicas de codificación no intuitivas.

El concepto de programación orientada a objetos (POO) no es nuevo, lenguajes clásicos como SmallTalk se basan en ella. Dado que la POO se basa en la idea natural de la existencia de un mundo lleno de objetos y que la resolución del problema se realiza en términos de objetos, un lenguaje se dice que está basado en objetos si soporta objetos como una característica fundamental del mismo. No debemos confundir que esté basado en objetos con que sea orientado a objetos: para que sea orientado a objetos al margen que esté basado en objetos, necesita tener clases y relaciones de herencia entre ellas. (pág. 12)

La programación orientada a objetos intenta simular el mundo real mediante el uso de objetos, clases y herencia entre clases para el diseño de aplicaciones.

#### Propiedades de la programación orientada a objetos

#### Objetos

Llobet Azpitarte et al. (2008) definen que:

Un objeto es un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización. En este caso las estructuras de datos y los algoritmos usados para manipularlas están encapsulados en una idea común llamada objeto.

Esta definición especifica dos propiedades características de los objetos:

* En primer lugar, un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados.
* En segundo lugar, cada objeto no es un ente aislado, sino que forma parte de una organización jerárquica o de otro tipo. (pág. 13)

#### Las clases

“Una clase es la descripción de una familia de objetos que tienen la misma estructura (atributos) y el mismo comportamiento (métodos).” (Llobet Azpitarte, Alonso Jordá, Miedes De Elías, Ruiz Fuertes, & Torres Goterris, 2008, pág. 15)

Terrero y Paredes (2011) consideran que:

Las clases son las plantillas para hacer objetos. Una clase sirve para definir una serie de objetos con propiedades (atributos), comportamientos (operaciones o métodos), y semántica comunes. Hay que pensar en una clase como un molde. A través de las clases se obtienen los objetos en sí.Es decir antes de poder utilizar un objeto se debe definir la clase a la que pertenece, esa definición incluye:

* **Sus atributos:** Es decir, los datos miembros de esa clase. Los datos pueden ser públicos (accesibles desde otra clase) o privados (sólo accesibles por código de su propia clase. También se las llama campos.
* **Sus métodos:** La función miembro de la clase. Son las acciones (u operaciones) que puede realizar la clase. (pág. 37)

1. **Atributos**

“Un atributo es una característica de un objeto. Mediante los atributos se define información oculta dentro de un objeto, la cual es manipulada solamente por los métodos definidos sobre dicho objeto.” (Carballo, 2010)

1. **Métodos**

“Los métodos se utilizan de la misma forma que los atributos, excepto porque los métodos poseen siempre paréntesis, dentro de los cuales pueden ir valores necesarios para la ejecución del método (parámetros).” (Terrero & Paredes, 2011, pág. 39)

#### Herencia

Llobet Azpitarte et al. (2008) explican que:

Herencia es la propiedad por la que los ejemplares de una clase hija (subclase) pueden tener acceso tanto a los datos como al comportamiento (métodos) asociados con una clase paterna (superclase). La herencia significa que el comportamiento y los datos asociados con las clases hijas son siempre una extensión de las propiedades asociadas con las clases paternas. Una subclase debe reunir todas las propiedades de la clase paterna y otras más. (pág. 19)



***Fuente:*** Libro Introducción a la POO (Programación Orientada a Objetos)

**Figura 14.** Esquema de Herencia

1. **Encapsulamiento**

“Una clase se compone tanto de variables (propiedades) como de funciones y procedimientos (métodos). Hay una zona oculta al definir la clase (zona privada) que sólo es utilizada por esa clase y por alguna clase relacionada. Hay una zona pública (llamada también interfaz de la clase) que puede ser utilizada por cualquier parte del código.” (Terrero & Paredes, 2011, pág. 35)

1. **Polimorfismo**

“Cada método de una clase puede tener varias definiciones distintas. En el caso del parchís: partida.empezar (4) empieza una partida para cuatro jugadores, partida.empezar (rojo, azul) empieza una partida de dos jugadores para los colores rojo y azul; estas son dos formas distintas de emplear el método empezar, que es polimórfico.” (Terrero & Paredes, 2011, pág. 36)

### ENTORNO DE DESARROLLO

#### IDE Android Studio

Google Inc. (2009) explica que:

Android Studio es un nuevo entorno de desarrollo de Android basado en IntelliJ IDEA. Ofrece nuevas características y mejoras con respecto a Eclipse ADT. En la parte superior de las capacidades que usted espera de IntelliJ, Android Studio ofrece:

* Sistema de construcción a base de Gradle flexible.
* Construir variantes y generación de múltiple APK.
* Extensas plantillas soportadas por los servicios de Google y varios tipos de dispositivos.
* Editor de diseño Rich con soporte para la edición de tema.
* Herramientas para capturar rendimiento, usabilidad, compatibilidad de versiones, y otros problemas
* ProGuard y aplicación de firma de capacidades.

El IDE Android Studio es el entorno de desarrollo Android creado por Google, posee características que facilitan el desarrollo, la refactorización es más potente, incorpora una herramienta para el análisis del código, la edición es más fluida, además de que permite la previsualización de los recursos.

#### Android SDK

“Es un kit de desarrollo de software para Android consta de un conjunto de herramientas para el desarrollo que permite a los desarrolladores y programadores confeccionar aplicaciones y juegos para el sistema dicho. Se trata pues de una interfaz de programación de aplicaciones que hace uso del lenguaje de programación Java.” (Ramírez Hernández, 2011).

Es un conjunto de herramientas de programación que permiten el desarrollo de aplicaciones de una manera sencilla, a través del lenguaje de programación java.

#### Xml

Sagástegui Lescano (2008) afirma que:

XML son las siglas en inglés de eXtensible Markup Language (lenguaje de marcas ampliable), este lenguaje es del tipo metalenguaje y es extensible mediante el uso de etiquetas, fue desarrollado por el World Wide Web Consortium. Nació como una forma de reducir SGML usando para ello la parte que permite definir la gramática de lenguajes específicos. XML (Extensible Markup Language) es un lenguaje de etiquetas, es decir, cada paquete de información está delimitado por dos etiquetas como se hace también en el lenguaje HTML, pero XML separa el contenido de la presentación. Sus características más importantes son que permite separar el código de la presentación del mismo y que es completamente extensible mediante el uso de nuevas etiquetas creadas por el desarrollador.

XML es una tecnología que sirve para estructurar, almacenar e intercambiar información.

#### Lenguaje de programación Java

Según Terrero y Paredes (2011):

La compañía Sun describe el lenguaje Java como: “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico. (pág. 3)

“Java es un lenguaje de programación de alto nivel con el que se pueden escribir tanto programas convencionales como para internet.” (Ceballos, 2011, pág. 22)

Java es un lenguaje de programación multiplataforma, orientado a objetos y basado en clases.

#### PostgreSQL

Gibert Ginestà y Pérez Mora (2012) consideran que:

En 1996, se hizo evidente que el nombre “Postgres95” no resistiría el paso del tiempo. Elegimos un nuevo nombre, PostgreSQL, para reflejar la relación entre el Postgres original y las versiones más recientes con capacidades SQL. Al mismo tiempo, hicimos que los números de versión partieran de la 6.0, volviendo a la secuencia seguida originalmente por el proyecto Postgres. Durante el desarrollo de Postgres95 se hizo hincapié en identificar y entender los problemas en el código del motor de datos. Con PostgreSQL, el énfasis ha pasado a aumentar características y capacidades, aunque el trabajo continúa en todas las áreas.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. PostgreSQL utiliza un modelo cliente/servidor y usa *multiprocesos* en vez de *multihilos* para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

PostgreSQL es sistema de gestión de bases de datos relacional orientada a objetos, de código abierto.

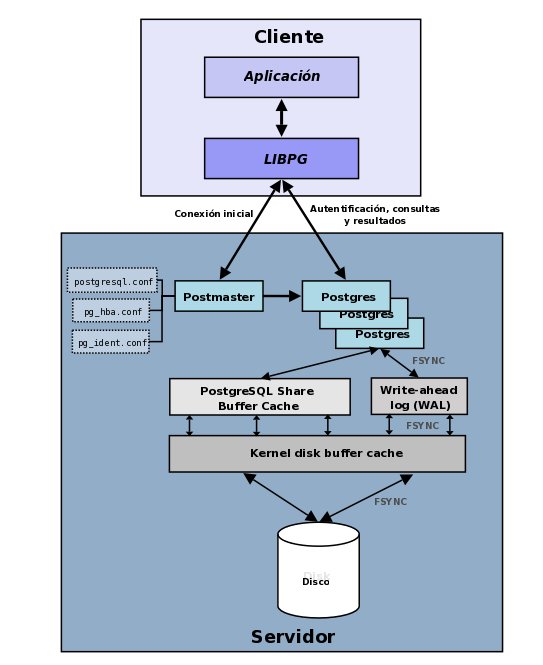
1. **Las principales mejoras en PostgreSQL incluyen:**

* Los bloqueos de tabla han sido sustituidos por el control de concurrencia multi-versión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde pg\_dump mientras la base de datos permanece disponible para consultas.
* Se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers).
* Se han añadido funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, forzado de tipos cadena literal, conversión de tipos y entrada de enteros binarios y hexadecimales.
* Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.
* La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20-40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada.

1. **Componentes de un sistema PostgreSQL**

Martínez (2010) explica lo componentes:

* **Aplicación cliente:** Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos.
* **Demonio postmaster:** Este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes.
* **Ficheros de configuración:** Los 3 ficheros principales de configuración utilizados por PostgreSQL, postgresql.conf, pg\_hba.conf y pg\_ident.conf
* **Procesos hijos postgres:** Procesos hijos que se encargan de autentificar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes.
* **PostgreSQL share buffer cache:** Memoria compartida usada por PostgreSQL para almacenar datos en caché.
* **Write-Ahead Log (WAL):** Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO).
* **Kernel disk buffer cache:** Caché de disco del sistema operativo
* **Disco:** Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.

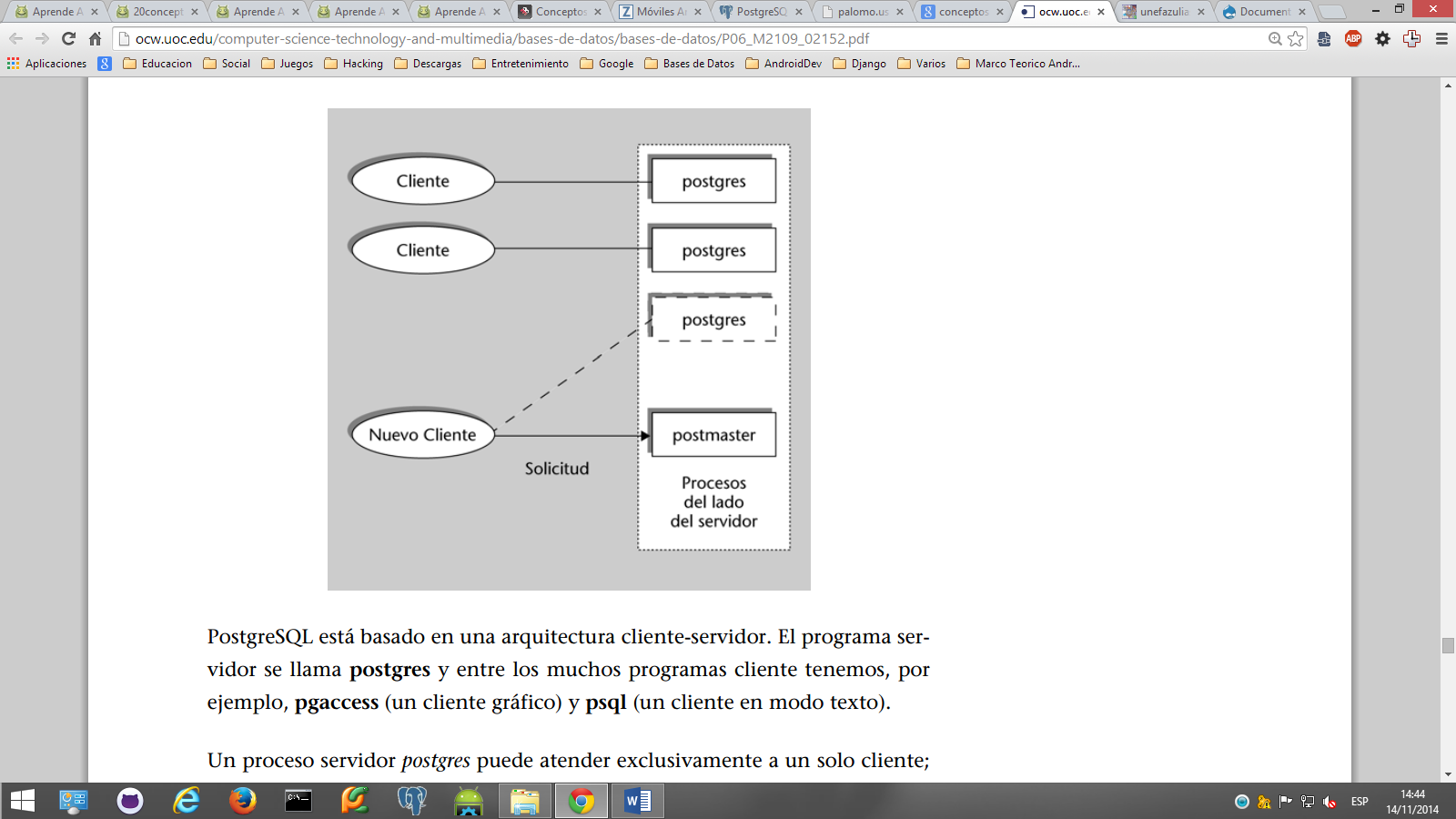


***Fuente:*** http://www.postgresql.org.es/sobre\_postgresql

**Figura 15.** Componentes de un Sistema PostgreSQL

1. **Arquitectura de PostgreSQL**

El siguiente gráfico muestra de forma esquemática las entidades involucradas en el funcionamiento normal del gestor de bases de datos:



**Figura 16.** Arquitectura de PostgreSQL

***Fuente:*** http://ocw.uoc.edu/

Según Gibert Ginestà y Pérez Mora (2012):

PostgreSQL está basado en una arquitectura cliente-servidor. El programa servidor se llama **Postgres** y entre los muchos programas cliente tenemos, por ejemplo, **pgaccess** (un cliente gráfico) y **psql** (un cliente en modo texto).

Un proceso servidor  *postgres* puede atender exclusivamente a un solo cliente; es decir, hacen fala tantos procesos servidor *postgres* como clientes haya. El proceso **postmaster** es el encargado de ejecutar un nuevo servidor para cada cliente que solicite una conexión.

Se llama **sitio** al equipo anfitrión (*host*) que almacena un conjunto de bases de datos PostgreSQL. En un *sitio* se ejecuta solamente un proceso *postmaster* y múltiples procesos *postgres.* Los clientes pueden ejecutarse en el mismo sitio o en equipos remotos conectados por TCP/IP.Es posible restringir el acceso a usuarios o direcciones IP modificando las opciones del archivo pg\_hba-conf, que se encuentra en /etc/postgresql/pg\_hba.conf. Este archivo, junto con /etc/postgresql/postgresql.conf es particularmente importante, porque algunos de sus parámetros de configuración por defecto provocan multitud de problemas al conectar inicialmente y porque en ellos se especifican los mecanismos de autenticación que usará PostgreSQL para verificar las credenciales de los usuarios.

Para habilitar la conexión a PostgreSQL desde clientes remotos, debemos verificar el parámetro tcpip\_socket = true en el fichero */etc/postgresql/postgresql.conf*.

A continuación, para examinar los métodos de autenticación y las posibilidades de conexión de clientes externos, debemos mirar el fichero */etc/postgresql/pg\_hba.conf*, donde se explicita la acción que hay que emprender para cada conexión proveniente de cada *host* externo, o grupo de *hosts.* (pág. 65)

#### Python

González Duque (2010) afirma que:

Es un lenguaje interpretado, interactivo y orientado a objetos que ofrece una gran cantidad de estructuras de datos de alto nivel por medio de un tipado dinámico y fuerte, además de estas características es multiparadigma y multiplataforma. Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos.

Python es un lenguaje de programación orientado a objetos, multiplataforma, dinámico que posee grandes estructuras de datos.

**Ventajas de Python**

* Python es un lenguaje muy expresivo, es decir, los programas Python son muy compactos: un programa Python suele ser bastante más corto que su equivalente en lenguajes como C. (Python llega a ser considerado por muchos un lenguaje de programación de muy alto nivel.)
* Python es muy legible. La sintaxis de Python es muy elegante y permite la escritura de programas cuya lectura resulta más fácil que si utilizáramos otros lenguajes de programación.
* Python ofrece un entorno interactivo que facilita la realización de pruebas y ayuda a despejar dudas acerca de ciertas características del lenguaje.
* El entorno de ejecucion de Python detecta muchos de los errores de programación que escapan al control de los compiladores y proporciona información muy rica para detectarlos y corregirlos.
* Python puede usarse como lenguaje imperativo procedimental o como lenguaje orientado a objetos.
* Posee un rico juego de estructuras de datos que se pueden manipular de modo Sencillo. (págs. 7-8)

#### Servicios Web

“Son un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios.” (W3C, 2013)

Los servicios web es una tecnología que utiliza una serie de protocolos que permiten el intercambio de información entre distintas aplicaciones.

1. **¿Para qué sirven los Servicios Web?**

“Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas.” (W3C, 2013)

#### Soap (Simple Object Access Protocol)

1. **Introducción**

Moscatelli (2008) considera que:

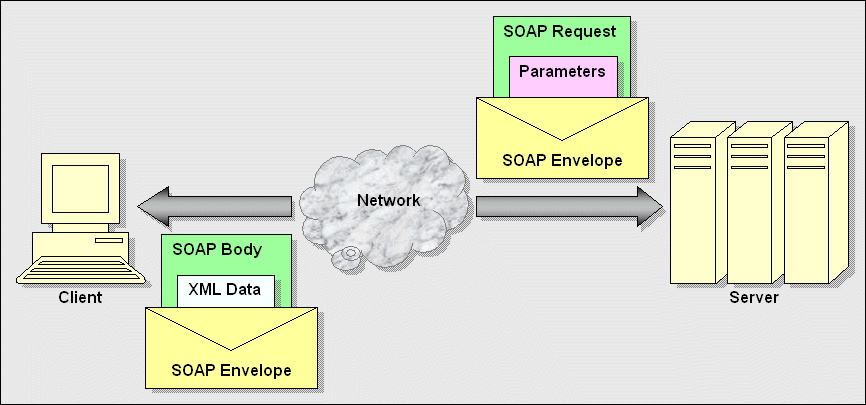
**S**imple **O**bject **A**ccess **P**rotocol (SOAP) surge a partir de la especificación realizada por Dave Winer, de la empresa UserLand Software a finales del año 1998, de un mecanismo basado en XML para realizar invocaciones RPC. A partir de dicha especificación y de la unión de las empresas DevelopMentor, UserLand Software y Microsoft surge públicamente la versión 0.9 de SOAP en setiembre de 1999. La versión 1.0 fue lanzada casi a continuación de la versión 0.9 (diciembre de 1999) y enviada a la IETF (Internet Engineering Task Force). Este organismo hizo de esta versión una recomendación oficial. Posteriormente se realizó la versión 1.1 de SOAP, donde, además de las empresas mencionadas, también participaron Lotus e IBM, entre muchas otras empresas de desarrollo de software. Esta nueva versión fue enviada al W3C (World Wide Web Consortium) en mayo del 2000, con el propósito de formar un grupo de trabajo en el área de los protocolos basados en XML. En julio de 2000 el grupo de trabajo en el área de XML del W3C edita el primer W3C Working Draft de la versión 1.2 de SOAP. La especificación del protocolo SOAP, básicamente, describe un formato de mensajes para comunicar aplicaciones. La filosofía de SOAP, para lograr dicha comunicación, es no inventar una nueva tecnología, sino combinar tecnologías existentes y de amplia aceptación en la industria de software. En particular, combina XML para la codificación de los mensajes y HTTP como protocolo de transporte (aunque no se excluye el uso de otros protocolos de tranporte). (pág. 3)

1. **¿Qué es SOAP?**

Moscatelli (2008) explica que:

SOAP define un mecanismo simple y liviano (en contraposición a sofisticado) para la comunicación, en un entorno distribuido o descentralizado, entre componentes de software o aplicaciones. La comunicación se realiza mediante mensajes codificados en XML y transportados por un protocolo de transporte (SOAP no mandata el uso de un protocolo de transporte en particular, aunque si define como es el transporte en caso de usar HTTP). En definitiva SOAP define un mecanismo para el intercambio de información, estructurada y tipeada, entre pares de aplicaciones en un entorno distribuido, teniendo como objetivos de diseño la simplicidad y la extensibilidad. SOAP no define por sí mismo la semántica de las aplicaciones, como ser un modelo de programación o algún tipo de semántica específica de una implementación, sino que provee un mecanismo simple para expresar la semántica de las aplicaciones, mediante un modelo modular de empaquetado de mensajes y la definición de como codificar los datos de las aplicaciones en dichos módulos. (pág. 4)

1. **Arquitectura básica de SOAP**



***Fuente:*** http://www.fing.edu.uy/

**Figura 17.** Arquitectura básica de SOAP

Moscatelli (2008) explica que:

En la figura se observa la arquitectura básica de un sistema, análogo al descripto, construido sobre SOAP y los mensajes que definen la interacción entre la aplicación cliente y la aplicación servidor. Generalmente la aplicación cliente envía un mensaje (REQUEST vía HTTP), el cual al ser recibido por la aplicación servidor genera una respuesta (RESPONSE) que es enviada a la aplicación cliente vía HTTP. Se observa además que en el caso de usar SOAP para realizar RPC, la invocación RPC se mapea naturalmente al REQUEST de HTTP y la respuesta RPC se mapea al RESPOSE de HTTP. (pág. 5)

1. **Protocolo SOAP**

Según Moscatelli (2008) la especificación del protocolo SOAP indica que el mismo consiste de 3 partes:

* El constructor SOAP ENVELOPE que define un framework para expresar qué hay en un mensaje, a quién está dirigido el mensaje y cuando es opcional o mandatorio.
* Las reglas de codificación que definen un mecanismo de serialización para ser usado para intercambiar instancias de tipos de datos.
* La representación SOAP RPC que define una metodología que puede ser usada para representar invocaciones a procedimientos remotos y sus respuestas (pág. 6).

#### Wsdl

“WSDL es sinónimo de Web Services Description Language. WSDL es un lenguaje para describir servicios Web y cómo acceder a ellos, está escrito en XML.” (Refsnes Data Inc, 2012)

Es un formato xml que sirve para describir servicios web, que describe la forma de comunicación, protocolos y mensajes necesarios para la interacción.

1. **Documento WSDL**

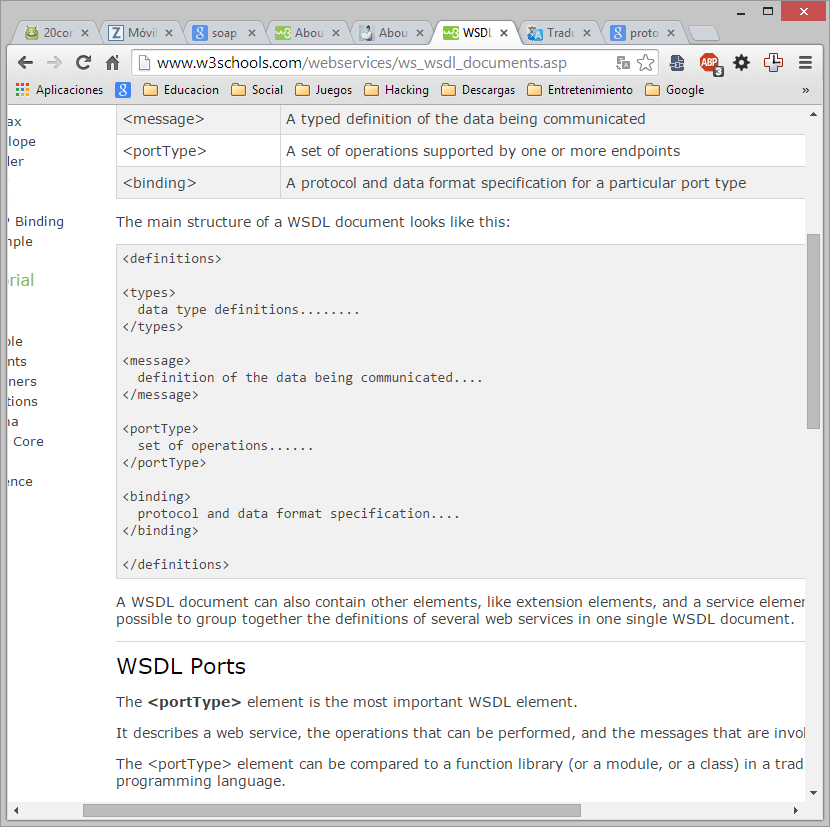
Un documento WSDL es un documento XML simple. Contiene un conjunto de definiciones para describir un servicio web utilizando estos elementos principales:

|  |  |
| --- | --- |
| **Elemento** | **Descripción** |
| <types> | Un contenedor para las definiciones de tipos de datos que utiliza el servicio web |
| <message> | Una definición con tipo de los datos que se comunica |
| <portType> | Un conjunto de operaciones con el apoyo de uno o más puntos finales |
| <binding> | Una especificación de protocolo y datos de formato para un tipo de puerto en particular |

**Tabla 6.** Elementos para describir un Servicio Web

**Fuente:** http://www.w3schools.com/

La estructura principal de un documento WSDL es algo similar a esto:



**Figura 18.** Ejemplo de la estructura de un documento WSDL

***Fuente:*** http://www.w3schools.com/

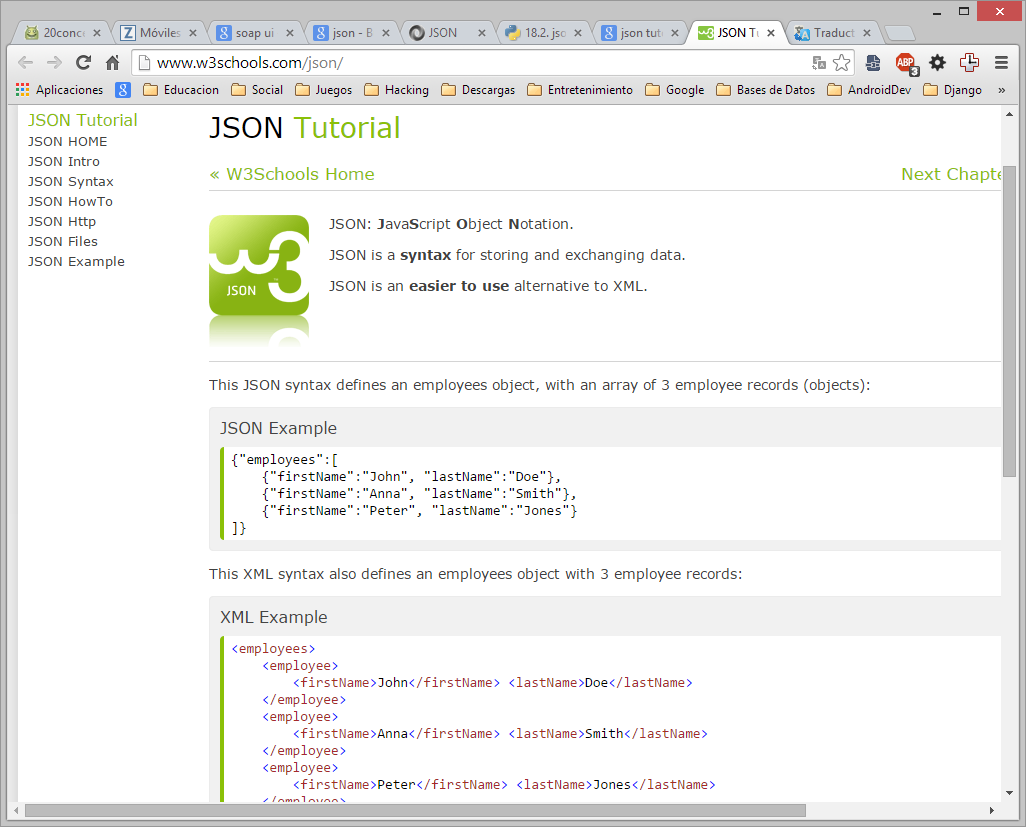
“Un documento WSDL puede contener también otros elementos, como elementos de extensión, y un elemento de servicio que permite agrupar las definiciones de varios servicios web en un documento único WSDL.” (Refsnes Data Inc, 2012)

#### ¿Qué es JSON?

Refsnes Data Inc (2012) afirma que:

JSON es sinónimo de JavaScript Object Notation, es un formato de intercambio de datos ligero independiente del lenguaje, es "auto-descripción" y fácil de entender.

Esta sintaxis JSON define un objeto empleados, con una serie de 3 registros de empleados (objetos):



**Figura 19.** Ejemplo definición de objeto Json

***Fuente:*** http://www.w3schools.com/json

1. **Al igual que XML**

* JSON es texto plano
* JSON es "auto-describe" (legible)
* JSON es jerárquica (valores dentro de los valores)
* JSON pueden recuperar con una HttpRequest

1. **Diferencia de XML**

* JSON no utiliza la etiqueta final
* JSON es más corto
* JSON es más rápido para leer y escribir
* JSON puede utilizar matrices

1. **Reglas de sintaxis en JSON**

* Los datos son en pares nombre / valor
* Los datos se separan por comas
* Las llaves tienen objetos
* Los corchetes tienen matrices

### METODOLOGÍA MOBILE-D

VTT Technology for business (2007) afirma que:

Es un metodología para el desarrollo ágil de software, que no solamente está orientada al desarrollo de aplicaciones móviles, también se puede usar en aplicaciones de seguridad, financieras, de logísticas, y de simulación.

Mobile-D se basa en la programación extrema (XP) para la implementación, Cristal methodologies para la escalabilidad y en el Proceso Unificado de Desarrollo (RUP) para la cobertura del ciclo de vida.

* La fase de exploración, siendo ligeramente diferente del resto del proceso de producción, se dedica al establecimiento de un plan de proyecto y los conceptos básicos, por lo tanto, se puede separar del ciclo principal de desarrollo (aunque no debería obviarse).Los autores de la metodología ponen además especial atención a la participación de los clientes en esta fase.
* Durante la fase de inicialización, los desarrolladores preparan e identifican todos los recursos necesarios. Se preparan los planes para las siguientes fases y se establece el entorno técnico (incluyendo el entrenamiento del equipo de desarrollo). Los autores de Mobile-D afirman que su contribución al desarrollo ágil se centra fundamentalmente en esta fase, en la investigación de la línea arquitectónica. Esta acción se lleva a cabo durante el día de planificación.
* En la fase de "producción" se repite la programación de tres días (planificación, trabajo, liberación) se repite iterativamente hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración de antemano (de ahí el nombre de esta técnica de TestDriven Development, TDD). Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el último día se lleva a cabo la integración del sistema (en caso de que estuvieran trabajando varios equipos de forma independiente) seguida de las pruebas de aceptación.
* En la fase de estabilización, se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funciona correctamente. Esta será la fase más importante en los proyecto multi-equipo con diferentes subsistemas desarrollados por equipos distintos. En esta fase, los desarrolladores realizarán tareas similares a las que debían desarrollar en la fase de "productización", aunque en este caso todo el esfuerzo se dirige a la integración del sistema. Adicionalmente se puede considerar en esta fase la producción de documentación.
* La última fase (prueba y reparación del sistema) tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos de cliente y se eliminan todos los defectos encontrados.

# BIBLIOGRAFÍA

Álvarez, J. (10 de Abril de 2008). *http://es.slideshare.net/.* Obtenido de http://es.slideshare.net/Jmaquino/dispositivos-moviles

Cancela, L., & Ostos, S. (2012). *https://sites.google.com/*. Obtenido de https://sites.google.com/site/swcuc3m/home/android/

Carballo, P. Y. (21 de Febrero de 2010). *Universidad de los Andes.* Recuperado el 4 de Noviembre de 2013, de Web del profesor: http://webdelprofesor.ula.ve/ingenieria/hyelitza/materias/programacion2/oxo/ProfaYusneyi\_Tema8\_POOClasesyObjetos.pdf

Ceballos, J. (2011). *Java 2.* Ra-Ma.

Gibert Ginestà, M., & Pérez Mora, O. (2012). *http://ocw.uoc.edu/.* Obtenido de http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06\_M2109\_02152.pdf

Gómez Oliver, S. (27 de Febrero de 2012). *sgoliver.net blog*. Obtenido de http://www.sgoliver.net/blog/?p=2594

González Duque, R. (2010). *Python para todos.* España: Autoedición.

Google Inc. (2009). *developer.android.com/*. Obtenido de https://developer.android.com/sdk/installing/studio.html

Guevara Soriano, A. (06 de Agosto de 2010). *Revista Seguridad*. Obtenido de http://revista.seguridad.unam.mx/numero-07/dispositivos-m%C3%B3viles

Kovach, P. S. (19 de Agosto de 2013). *SoyEntrepreneur*. Recuperado el 6 de Noviembre de 2013, de SoyEntrepreneur: http://www.soyentrepreneur.com/25686-17-momentos-clave-de-android.html

Llobet Azpitarte, R., Alonso Jordá, P., Miedes De Elías, E., Ruiz Fuertes, M., & Torres Goterris, F. (2008). *Introducción a la programación orientada a objetos con java.* Valencia: Universidad Politécnica de Valencia.

Marco, C. (2010). Sistema Operativo Android Todo lo que querías saber y tenías miedo a preguntar. *Tecnología con estilo Gadgets*, 62.

Martínez, E. (Mayo de 2001). *http://www.adecom.biz/.* Obtenido de http://www.adecom.biz/pdf/pdf\_agosto2005/La%20evolucion%20de%20la%20telefonia%20movil.pdf

Martínez, J. (21 de Mayo de 2013). *AndroidZone.org*. Obtenido de http://androidzone.org/2013/05/historia-de-android-la-evolucion-a-lo-largo-de-sus-versiones/

Martínez, R. (2 de 10 de 2010). *Postgresql.org.es*. Obtenido de http://www.postgresql.org.es/sobre\_postgresql

Montoya, J. (2012). *Sistemas Operativos para Moviles.* Obtenido de https://docs.google.com: https://docs.google.com/presentation/d/177BvKpFn3-B07mICmpHmsPsg9m35Sr-\_K6qBOFwVju4/edit#slide=id.p

Moscatelli, S. (21 de Agosto de 2008). *Fing.edu.uy.* Obtenido de www.fing.edu.uy/inco/grupos/lins/docsgen/soap/soap.doc

Nicolás, D. (30 de Junio de 2012). Jelly Bean: El nuevo "dulce" de Android. *Diario La tercera*, pág. 22.

Paredes Velasco, M., Santacruz Valencia, L., & Domínguez Mateos, F. (2012). *Programación Multimedia y Dispositivos Móviles. Cfgs.* Ra-ma.

Procoop. (23 de Marzo de 2010). *InfoProcoop*. Recuperado el 30 de Octubre de 2013, de http://www.infoprocoop.com.ar/index.php?option=com\_content&view=article&id=86:introduccion-a-la-telefonia-fija&catid=37:telefonia&Itemid=62

Rabajoli, G. (27 de Septiembre de 2007). *Pido Ayuda*. Recuperado el 30 de Octubre de 2013, de http://pidoayuda.blogspot.com/2007/09/ventajas-y-desventajas-de-los.html

Ramírez Hernández, E. (14 de Marzo de 2011). *Universitat Politécnica de Valéncia.* Recuperado el 5 de Noviembre de 2013, de Desarrollo de aplicaciones para dispositivos con sistema operativo Android: http://riunet.upv.es/bitstream/handle/10251/10299/Memoria.pdf?sequence=1&isAllowed=y

Refsnes Data Inc. (2012). *http://www.w3schools.com/*. Obtenido de http://www.w3schools.com/

Ribas Lequerica, J. (2013). *Desarrollo de Aplicaciones para Android.* España: Anaya Multimedia.

Rivera, A. (2012). Sistemas Operativos Móviles: Comunicación en tiempo real . *PCWorld*.

Sagástegui Lescano, W. (2008). *Aprenderaprogramar*. Obtenido de http://aprenderaprogramar.com/index.php?option=com\_content&view=article&id=102:ique-es-y-para-que-sirve-el-lenguaje-de-etiquetas-xml-extensible-markup-language&catid=46:lenguajes-y-entornos&Itemid=163

Santa Maria, F. (26 de Febrero de 2014). *Staffcreativa*. Obtenido de http://blog.staffcreativa.pe/android-ventajas-desventajas/

Sébastien, P. (2012). Android Guia de desarrollo de aplicaciones Smartphones y Tabletas. En P. Sébastien, *Android Guia de desarrollo de aplicaciones Smartphones y Tabletas* (pág. 20). Barcelona: Ediciones ENI.

Sébastien, P. (2012). Android Guia de desarrollo de aplicacions para Smartphones y Tabletas. En P. Sébastien, *Android Guia de desarrollo de aplicacions para Smartphones y Tabletas* (pág. 11). Barcelona: Ediciones ENI.

Terrero, H., & Paredes, J. (2011). *Desarrollo de Aplicaciones Java.* Fundación Código Libre.

Tomás, G. (15 de Octubre de 2012). *Computer Hoy*. Recuperado el 5 de Noviembre de 2013, de http://computerhoy.com/listas/moviles/5-mejores-versiones-android-os-1706

Tomás, G. J. (2013). El gran libro de Android 3ra Edición. En G. J. Tomás, *El gran libro de Android 3era Edición.* Barcelona: Ediciones Marcombo.

VTT Technology for business. (22 de Noviembre de 2007). *Agile.vtt.fi*. Obtenido de http://agile.vtt.fi/mobiled.html

W3C. (2013). *http://www.w3c.es/*. Obtenido de http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb