

**Universidad Autónoma Metropolitana - Azcapotzalco**

**División de Ciencias Básicas e Ingeniería**

**Licenciatura en Ingeniería en Computación**

**Reporte Final Proyecto de Integración**

**Modalidad:** Estancia Profesional

**Aplicación Móvil para la Autorización de Formatos Digitales**

**Alumno:**

Roberto Espejel Ramírez

Matricula: 207205750

**Asesora Externa:**

Ing. Natalia Chávez Calvo

Gerente

Area Intelligent Technology Manager

**Asesora:**

Dra. Silvia Beatriz González Brambila

Profesora Titular

Departamento de Sistemas

Trimestre 2018 - Invierno

24 de abril de 2018

Yo, Silvia Beatriz González Brambila, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como el Repositorio Institucional de UAM Azcapotzalco.

Asesora:

Handwritten signature of Silvia Beatriz González Brambila in cursive script, written over a horizontal line.

Yo, Natalia Chávez Calvo, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como el Repositorio Institucional de UAM Azcapotzalco.

Asesora Externa:

Handwritten signature of Natalia Chávez Calvo in cursive script, written over a horizontal line.

Yo, Roberto Espejel Ramírez, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la biblioteca Digital, así como en el Repositorio Institucional de la UAM Azcapotzalco.

Alumno:

Handwritten signature of Roberto Espejel Ramírez in cursive script, written over a horizontal line.

## Resumen

El presente reporte describe el desarrollo de un sistema para gestionar trámites internos de una empresa, se menciona los recursos empleados, así como también la elaboración un video tutorial de cómo es utilizado cada módulo dentro de la aplicación.

El sistema funcionó en un ámbito de aplicación para teléfono móviles, el resultado de este proyecto es llamado “*AuDiFoHHn*”. *AuDiFoHHn* es operada a través de sistema operativo *Android*, desde la versión 4.4.2 hasta la 7.1.1 en sus versiones más actuales.

Con el recurso de esta tecnología se puede interactuar con mucha información, alguna de esta pública y otra privada; para resguardar documentación confidencial de cada usuario en algunas ocasiones se recurre al Código QR, también conocida como código de respuesta rápida, se usa en estrategias de comunicación y marketing de las empresas como las revistas, periódicos, anuncios publicitarios y facturas, entre otros; ya que permite almacenar datos íntegros y auténticos ocupando menos espacio, además es económico generarlos.

La aplicación nativa para *Android* y la aplicación web en PHP generan ambas una aplicación móvil con el nombre “*AuDiFoHHn*” dirigida al sector empresarial hotelero, enfocada en gestionar trámites internos para los trabajadores utilizando Código QR para la seguridad en la aprobación de dichos formatos, así como también publicar información de cumpleaños del mes, eventos internos, vacantes, avisos de última hora, menú del comedor, aviso de defunción, cursos y capacitaciones.

Con la implementación de la aplicación “*AuDiFoHHn*” en teléfonos móviles se reducen los tiempos de respuesta a los trámites hechos por los trabajadores, de igual manera, informar y actualizar los últimos sucesos relacionados con la empresa y oportunidades de crecimiento a los empleados.

## Tabla de contenido

1. Introducción .....	7
2. Antecedentes .....	8
3. Justificación .....	10
4. Objetivo General .....	11
4.1 Objetivos Específicos.....	11
5. Marco Teórico .....	13
5.1 Sistema Operativo <i>Android</i> .....	13
5.2 Lenguaje de programación <i>Java</i> .....	14
5.3 IDE Android Studio .....	15
5.4 API 19 Android 4.4.....	15
5.5 MySQL .....	15
5.6 Elementos de una aplicación.....	16
5.6.1 Activity.....	16
5.6.2 Layout.....	16
5.6.3 View.....	17
5.6.4 Service .....	17
5.6.5 Content Provider .....	17
5.6.6 Fragment .....	17
5.6.7 Broadcast Receiver .....	18
5.6.7 Intent.....	18
5.7 AsyncTask .....	18
5.8 Dependencias .....	19
5.9 Ciclo de vida de una <i>Activity</i> .....	19
5.10 Aplicación Web.....	20
5.11 API REST.....	20
5.12 JSON .....	21
6. Desarrollo del Proyecto .....	22
6.1 Investigación previa .....	22
6.2 Requerimientos.....	22
6.3 Modelo Arquitectónico de la aplicación .....	24
6.4 Funcionamiento de la aplicación.....	25
6.5 Funcionamiento del servidor .....	25
6.6 Base de datos .....	25

6.7 Procedimiento .....	37
6.8 Uso del Sistema .....	45
7. Resultados .....	46
8. Conclusiones.....	55
8.1 Seguimiento a futuro .....	56
9. Bibliografía .....	58
10. Apéndices .....	61
10.1 AbrirGaleria.class .....	61
10.2 ActividadMensual.class .....	65
10.3 ContenedorFragmentMg.class .....	68
10.4 CorreoEstacionamientoMg.class.....	70
10.5 CorreoPermicongMg.class .....	74
10.6 CorreoPermisingMg.class.....	79
10.7 CorreoVacacionesMg.class.....	83
10.8 Cumpleaños.class.....	88
10.9 DetalleEvento.class .....	91
10.10 DetalleVacaciones.class .....	92
10.11 Estacionamiento.class .....	92
10.12 FragmentAceptar.class.....	96
10.13 FragmentDenegar.class.....	100
10.14 Genereqr.class.....	103
10.15 IntroApp.class.....	105
10.16 LoginGM.class .....	106
10.17 LoginTms.class.....	108
10.18 MainActivity.class.....	111
10.19 ModuloAutorizar.class .....	114
10.20 ModuloCalendario.class .....	115
10.21 ModuloFormatos.class.....	116
10.22 PermisoCon.class.....	121
10.23 PermisoSin.class .....	126
10.24 ScannerFinal.class.....	132
10.25 Vacaciones.class.....	135
10.26 Vacantesz.class.....	142
10.27 VerRegitroestacionamiento.class.....	144

10.28 VerRegistrocon.class .....	147
10.29 VerRegistropsin.class .....	150
10.30 VerRegistrovacaciones.class .....	153

## 1. Introducción

La empresa *Operadora Hotel Centro Historico, S de RL de CV* es socio de una gran cadena hotelera de la marca Hilton, una de ellas es el hotel *Hilton Mexico City Reforma* que ofrece servicios de centro de convenciones, así como de hotelería.

Cuenta con el departamento de Sistemas *IT* que tiene como propósito apoyar con todos los servicios de red, mantenimiento y prevención de fallas en equipos de ordenadores y servidores entre otros más; algunos servicios son locales y otros a distancia para los clientes, huéspedes, proveedores, empleados y propietarios.

Con los avances tecnológicos en la actualidad es más fácil la comunicación a distancia y toda información interna tiene que ser segura y accesible desde cualquier parte del mundo; es por ello que el presente proyecto involucrado tiene como propósito acceder a toda información en trámites internos de la empresa de forma digital disminuyendo gastos de papelería, así como también el tiempo de resolución de dichos trámites.

Esta propuesta de proyecto es una aplicación móvil que permite gestionar trámites internos de una empresa sustituyendo la firma convencional por una firma electrónica avanzada<sup>1</sup> en formatos digitales, cada trámite cuenta con número de miembro de equipo, nombre completo del trabajador, puesto, fecha de ingreso, departamento, área, así como también una fecha de inicio y fin para aquellas solicitudes que tengan que soportar vacaciones, permiso con goce de sueldo, permiso sin goce de sueldo y estacionamiento.

---

<sup>1</sup> Conjunto de datos electrónicos para identificar al emisor de un mensaje, al igual la integridad del mismo.

## 2. Antecedentes

Cualquier institución tanto gubernamental como privada debe contar con tecnología nueva, que le permita ser más eficiente, por ello las aplicaciones para Smartphone son esenciales para estar a la vanguardia con la competencia.

La UAM<sup>2</sup> AZC<sup>3</sup> [1], SEFIN<sup>4</sup> CDMX<sup>5</sup> [2], CONAGUA<sup>6</sup> [3], SAT<sup>7</sup> [4], INFONAVIT<sup>8</sup> [5], IMSS<sup>9</sup> [6], son ejemplos de instituciones gubernamentales que facilitan a los usuarios sus servicios, tales como:

App UAM permite consultar horarios, cursos impartidos, calendario escolar, datos escolares, historial académico, historial de pagos, reglamento interno de la institución [13].

App Tesorería CDMX permite pagar predial, tenencia vehicular, infracciones de tránsito, derecho por suministro de agua, constancias de adeudo, certificaciones de pago y consulta de horarios de atención en las oficinas de la tesorería, así como su ubicación [14].

App MeteoInfo (CONAGUA) tiene la característica de pronosticar el tiempo mostrando de lo más alto una resolución en mapa satelital [15].

App SAT Móvil ofrece servicios a los contribuyentes tales como escanear código QR contenido en el acuse de alta autenticación llamado e.firma portable, permite generar y restablecer contraseñas para autenticarse en los diferentes servicios que ofrece el SAT vía web y móvil, consultar indicadores económicos y fiscales en tiempo real, tutoriales, noticias, citas, quejas y denuncias, calendario fiscal, declaraciones y acuses de pago de impuestos, factura móvil y declaraciones y acuses de pago de impuestos [16].

App Infonavit se puede consultar información de crédito, opciones de pago, información de ahorro de Subcuenta de Vivienda, acceso directo al centro de llamadas Infonatel [17].

App IMSS Digital permite la localización de clínicas, alta y cambio del NSS, alta y cambio de clínica, consulta de vigencia de derechos, agendar citas médicas e información de salud [18].

---

<sup>2</sup> Universidad Autónoma Metropolitana

<sup>3</sup> Azcapotzalco

<sup>4</sup> Secretaria de Finanzas

<sup>5</sup> Ciudad de México

<sup>6</sup> Comisión Nacional del Agua

<sup>7</sup> Servicio de Administración Tributaria

<sup>8</sup> Instituto del Fondo Nacional de la Vivienda para los Trabajadores

<sup>9</sup> Instituto Mexicano del Seguro Social



En el sector privado se encuentran TELMEX<sup>10</sup> [7], TELCEL<sup>11</sup> [8], IZZI<sup>12</sup> [9], BANCOMER<sup>13</sup> [10], BANAMEX<sup>14</sup> [11], SANTANDER<sup>15</sup> [12], entre otros y no puede faltar el Hotel Hilton Mexico City Reforma, todas las empresas anteriores utilizan herramientas que ahorran tiempo y papelería de servicios que convencionalmente tardan en tiempo y no contribuyen al cuidado del medio ambiente.

App Telmex 2.0 permite consultar y pagar recibo Telmex, crecer el paquete de telefonía, activar beneficios Claro Video, recargar tiempo aire Telcel, localiza sitios WIFI y tiendas Telmex [19].

App Mi Telcel se puede realizar consultas y administrar los servicios de la línea telefónica, recargar tiempo aire, compra de paquetes telefónicos, pago de factura con tarjeta de crédito/débito y ahorrar tiempo al realizar trámites sin necesidad de acudir a un Centro de Atención [20].

App izzi gestiona la red WIFI<sup>16</sup>, crecer los servicios de TV, pagar y consultar estado de cuenta, agregar canales de TV o velocidad de internet y atención personalizada a través de un chat [21].

App Bancomer Móvil permite retirar dinero en un cajero automático sin requerir de la tarjeta, genera códigos de retiro y enviarlos a otra persona para poder retirar dinero de un cajero automático, transferencias a cuentas BBVA Bancomer u otros bancos, pagar servicios domésticos, consulta de saldo, movimientos, contratación de préstamos, alertas personalizadas para compras, cargos, retiros y depósitos desde \$50.00 MXN<sup>17</sup> [22].

App Citibanamex móvil consultar saldo, movimientos de cuentas, transferencias a cualquier banco, pago de tarjetas de crédito y otros bancos, pagar servicios e impuestos del SAT, bloquear temporalmente tarjetas de crédito, modificar el límite de pago a tarjetas y localizar cajeros automáticos [23].

App Santander móvil controla gastos, transferencias, pago de tarjetas de crédito propias y de terceros, depósitos y retiro de dinero creciente, localización de sucursales y cajero automáticos, consulta de saldo de tarjetas de crédito hipotecario, asociar y modificar cuentas a un número celular [24].

App Hilton Honors permite registrarse desde cualquier lugar y elegir la habitación, llave digital para acceso a la habitación, ordenar comida, bebidas, almohadas extras, uso de puntos para ofertas [25].

---

<sup>10</sup> Teléfonos de México, S.A.B de C.V.

<sup>11</sup> Es una marca propiedad de la empresa mexicana Radiomóvil DIPSA, S.A de C.V.

<sup>12</sup> Es una empresa de servicios de telecomunicaciones operada por Cablevisión, S.A.B de C.V.

<sup>13</sup> Es una institución financiera múltiple mexicana, BBVA Bancomer, S.A.

<sup>14</sup> Grupo Financiero Banamex

<sup>15</sup> Es un grupo bancario español.

<sup>16</sup> Es una tecnología que permite la interconexión inalámbrica de dispositivos electrónicos.

<sup>17</sup> Peso Mexicano.

### **3. Justificación**

Se busca crear una aplicación que permita la generación, autorización y consulta de solicitudes en formato digital.

La aplicación móvil cumple con la arquitectura cliente – servidor complementada de una conexión a la base de datos móvil, la cual almacena datos del trabajador tanto laboral como personal junto con su historial de trámite que lo relacione.

Los trámites que pueden realizar los usuarios registrados son: vacaciones, días con/sin goce de sueldo, pago de las horas extras (tiempo x tiempo), petición de correo electrónico del dominio de la empresa, alta de usuario y contraseña para la consulta de huéspedes, solicitud de permisos de usuario que permita consultar ocupación, actividades internas y externas que se llevarán en el centro de convenciones, así como el menú del comedor.

Los usuarios capaces de tener una firma electrónica avanzada utilizando Código QR cifrado serán aquellos que como puesto sea el de gerente, con el propósito de autorizar las solicitudes con respecto a su departamento.

Tanto la petición del trámite como su validación se realiza a través de teléfonos inteligentes para ofrecer a los usuarios fácil acceso y tiempos de respuesta adecuados. Además, no se necesitará la impresión de documentos, ya que todo estará almacenado en la base de datos.

## **4. Objetivo General**

Diseñar e implementar una aplicación móvil que permita generar, autorizar y gestionar las solicitudes internas en formato digital de una empresa empleando una firma electrónica avanzada en Código QR.

### **4.1 Objetivos Específicos**

A continuación, se presentan los objetivos específicos que se presentaron dentro de la propuesta de este proyecto, así como una breve explicación de cómo se lograron:

- Diseñar e implementar un módulo que mostrará toda información acerca de todos los eventos y actividades programadas cada mes dentro de la empresa; se logró la implementación del módulo con una pantalla dividida en pestañas para desplegar toda la información en relación a cumpleaños, vacantes, eventos y actividades del mes.
- Diseñar e implementar un módulo para generar y presentar todos los trámites internos de la empresa en formato digital a solicitud del usuario; se logró la implementación del módulo con una pantalla que contiene un menú desplegable para elegir vacaciones, permiso con goce de sueldo, permiso sin goce de sueldo y estacionamiento, como apoyo se ofrece un calendario y un campo para escribir el motivo de la solicitud.
- Diseñar e implementar un módulo que llene automáticamente los datos personales del solicitante; se logró la implementación del módulo con una pantalla que despliega la información laboral relevante para soportar la solicitud al trámite requerido por el trabajador.
- Diseñar e implementar un módulo que genere y decodifique una firma electrónica avanzada, para su autorización y finalización del trámite interno solicitado por el usuario; se logró la implementación del módulo con una pantalla que contiene un campo para introducir una serie de caracteres alfanuméricos conocidos solamente por el gerente departamental, en ella se encuentra un botón para generar el Código QR y otro botón para guardar la firma electrónica avanzada en el teléfono móvil inteligente; en el caso de la validación y decodificación se logró la implementación del módulo que escanea con la cámara del teléfono inteligente el Código QR que representa la firma digital del gerente, así como también abrir un explorador de archivos para leer y validar la firma electrónica avanzada a través de una imagen guardada en la galería de fotos.
- Diseñar e implementar un módulo que visualice la resolución al finalizar el proceso de solicitud, también guardará y mostrará como historial del trabajador; se logró la implementación del

módulo con una pantalla que muestra todas las peticiones realizadas durante todo un periodo laboral de un trabajador, así como la resolución a cada trámite realizado con un breve comentario opcional.

- Diseñar e implementar que visualice las calificaciones del *SALT*; se logró la implementación del módulo que muestra la meta y estado actual del servicio en general *SALT*.
- Diseñar e implementar la base de datos remota; se logró la implementación del módulo que almacena todos los datos de manera remota en un *Web Service*.

## 5. Marco Teórico

Para poder complementar los conocimientos y cubrir los objetivos del proyecto de integración es necesario tener un cierto nivel de información de las herramientas involucradas para desarrollar una aplicación móvil, por ello a continuación se presentan los principales conocimientos generales que se emplearon en el desarrollo del proyecto.

### 5.1 Sistema Operativo *Android*

El sistema operativo *GNU*<sup>18</sup>, tipo Unix, pero libre involucra una suite de software por lo que es conocido como Proyecto *GNU*. *GNU* es usado como un núcleo llamado Linux, su propósito es administrar los recursos de la máquina y comunicarse con el hardware [31].

*Android* es un entorno *Open-Source*<sup>19</sup>, este software no solo sirve para el ambiente empresarial o laboral sino también para estudios de carácter académico. Este sistema operativo está basado en el núcleo *Linux*<sup>20</sup>, para lo cual es diseñado para un extensa gama de dispositivos móviles y otros administrados por Google[32].

Android se basa en núcleo de Linux versión 2.6 para servicios básicos, con esto ofrece agilidad y portabilidad para aprovechar las diversas opciones de teléfonos celulares, muchas aplicaciones dependen del núcleo de Linux para servicios como procesadores, memoria y administración de sistemas de archivos, que trabaeneno en entorno Java y para poder funcionar son lanzadas sobre una máquina virtual llamada *Dalvik* [26].

*Dalvik* es una variación de la máquina virtual de *Java*, diseñada para Android, con compilación en tiempo real, tiene control sobre recursos como: controladores de pantalla, cámara, memoria flash, entre otros más (ver Figura 1). Después de la compilación se obtiene un archivo con extensión .apk, para lo cual el código de este programa es considerado una aplicación y es usado por Android para su instalación en el dispositivo móvil [27].

- Aplicaciones: Está formado por la aplicación instalada en el dispositivo.
- Framework de aplicaciones: Su objetivo es simplificar la reutilización de componentes, es decir, las aplicaciones pueden publicar sus capacidades y otras hacer uso de ellas.
- Librerías: Incluye un conjunto de bibliotecas *C/C++* usadas por los diferentes componentes de Android.

---

<sup>18</sup> Acrónimo recursivo de ‘*GNU’s Not Unix*’.

<sup>19</sup> Iniciativa de Código Abierto.

<sup>20</sup> Sistema Operativo de GNU.

- Tiempos de ejecución de Android: Es la capa donde corren las aplicaciones, las cuales utilizan la máquina *Dalvik*.
- Núcleo Linux: Proporciona servicios de seguridad, manejo de memoria, multiproceso, pila de protocolos y soporte para los drivers.

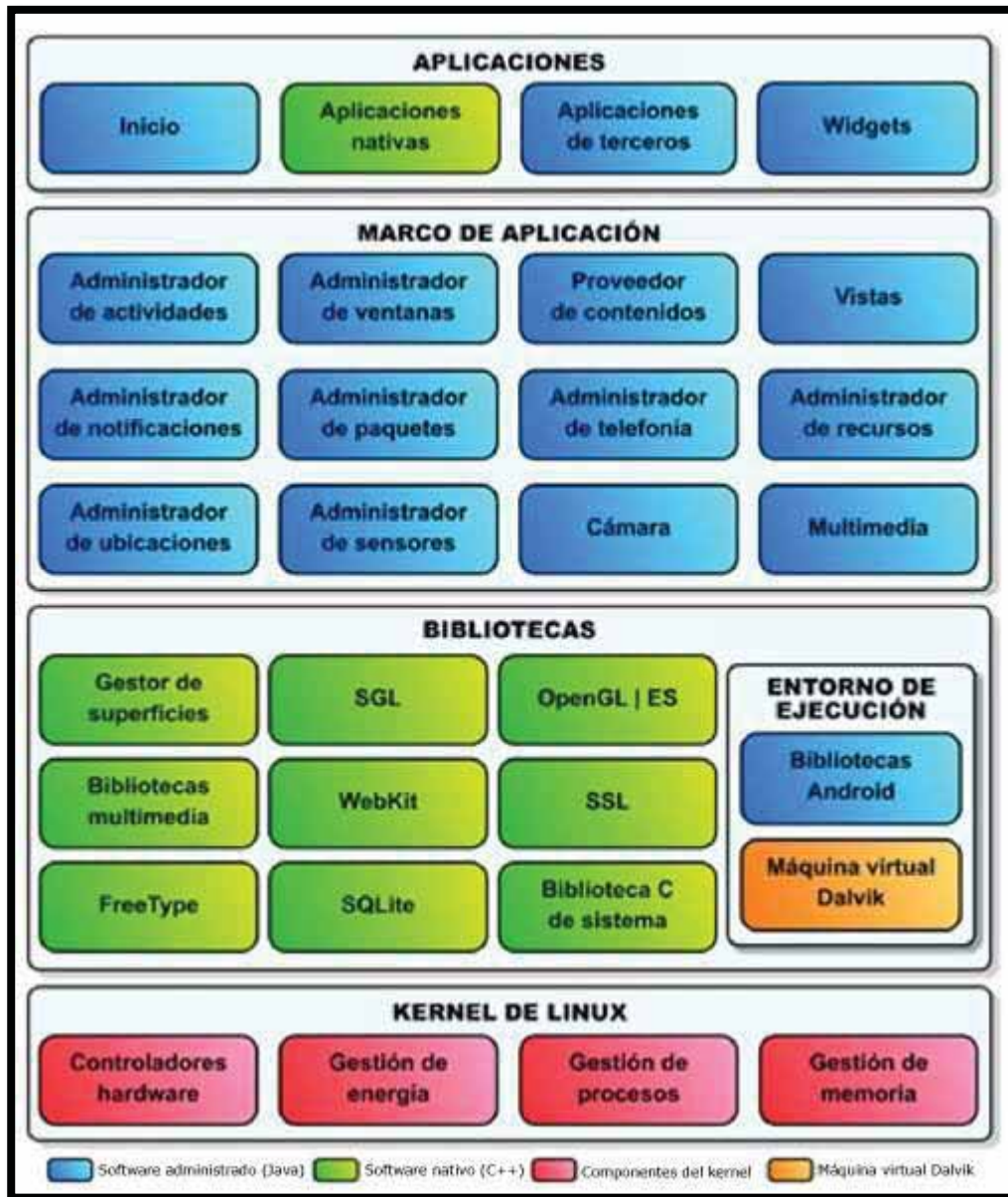


Figura 1. Arquitectura *Android*[33].

## 5.2 Lenguaje de programación *Java*

Java es un lenguaje de programación orientado a objetos, las características de este lenguaje son: orientado a objetos (encapsulación, herencia, polimorfismo, entre otras) es portable, dinámico,

produce applets, multihilo, alto rendimiento, seguro, robusto, distribuido, lenguaje simple, interpretado y compilado a la vez.

### 5.3 IDE Android Studio

Android Studio es un Entorno de Desarrollo Integrado, *IDE* por sus siglas en inglés: *Integrated Development Environment* para Android. Android Studio es un entorno de desarrollo integrado oficial para el sistema operativo Android. Está basado en lenguaje de programación Java, esto hace que sea multiplataforma, es decir, disponible para plataformas Microsoft Windows, MacOS y GNU/Linux.

El *SDK*<sup>21</sup> para Android Studio es Android *SDK* Tools el cual contiene depurador de código, biblioteca y, un simulador de teléfono basado en *QEMU* que es un emulador en interfaz gráfica que tiene la capacidad de virtualización dentro de un sistema operativo.

Algunas funciones del *IDE* son:

- *Android Virtual Device* (Emulador de dispositivos Android), la cual permite emular distintos dispositivos Android.
- *User Interface* (Interfaz de desarrollo visual), encargado de facilitar el desarrollo porque permite agregar componentes visuales de forma directa en la interfaz de la aplicación.

### 5.4 API 19 Android 4.4

Es un conjunto de reglas y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas, es usado como interfaz entre distintos programas y el usuario facilitando la interacción humano-software. El uso de este *API*<sup>22</sup> permite hacer el uso de funciones ya existentes en otro software, en el caso de modificar alguna funcionalidad, no se elimina, sino que se etiqueta como obsoleta, pero se puede seguir utilizando.

Para este proyecto la *API* mínima es Android 4.4 Nivel de *API* 19 su nombre comercial es postre “*KitKat*”.

### 5.5 MySQL

*MySQL*<sup>23</sup> es software libre y código abierto, por lo cual es desarrollo bajo la licencia pública general de *GNU* y licencia comercial, es programado en lenguaje *C*<sup>24</sup> y *C++*<sup>7</sup> lo que lo hace multiplataforma. El software es un sistema de administración relacional de bases de datos permitiendo una

---

<sup>21</sup> Kit de Desarrollo de Software (*Software Development Kit*).

<sup>22</sup> Interfaz de Programación de Aplicaciones (*Application Programming Interface*).

<sup>23</sup> Sistema de Gestión de Bases de Datos Relacional (*Database Management System, DBMS*).

<sup>24</sup> Lenguaje de Programación de nivel medio.

organización de datos en un conjunto de tablas con columnas y filas, cada columna guarda un determinado tipo de datos y un campo almacena el valor real de un atributo, en cuanto a las filas representan una relación predefinida de un objeto o entidad lo cual es marcado como un identificador único denominado clave primaria, mientras que filas de varias tablas pueden relacionarse con claves extranjeras[34].

*MySQL* es muy utilizada en aplicaciones Web combinada con *PHP*<sup>25</sup>, ya que la comunicación al motor de la base de datos relacional es a través de *SQL*<sup>26</sup> eso impacta en la velocidad de acceso y flexibilidad para la modificación de datos, es decir, para agregar, actualizar o eliminar filas de datos, recuperar subconjuntos de datos para aplicaciones de análisis y procesamiento de transacciones y para administrar todos los aspectos de la base de datos. Las reglas que utiliza garantizan la integridad de los datos precisos y fiables.

## 5.6 Elementos de una aplicación

Android está estructurado de una serie de elementos básicos que dan forma a la aplicación, tales como Activity, Layout, View, Service, Fragment, Content Provider, Broadcast Receiver, Intent. Estos elementos resultan ser imprescindibles para desarrollar aplicaciones [32].

### 5.6.1 Activity

Una *Activity* representa una sola pantalla con una interfaz de usuario para poder interactuar y realizar una acción. Por ejemplo, en una aplicación de llamada telefónica una *Activity* muestra los contactos y otra *Activity* las llamadas realizadas. Para el usuario final las *activities* se ven como si fuera una sola, pero cada una es independiente de aquellas que se relacionan entre sí.

En el IDE de Android Studio son representadas como subclase de una *Activity* principal. Cada vez que se inicia una *Activity* nueva, se detiene la *Activity* anterior conservándola en una pila de *activities* esto hace que cuando el usuario presiona el botón Atrás, la *Activity* de la pila se reanuda junto con los recurso y acciones que se interrumpieron, así como también al mismo tiempo se destruya en la pila de *activities* liberando conexiones de red o a la base de datos.

### 5.6.2 Layout

Se ejecuta en segundo plano para realizar operaciones de larga duración o para realizar tareas para procesos remotos.

Un *layout* define la estructura visual para una interfaz de usuario. Se pueden diseñar de dos maneras:

---

<sup>25</sup> Lenguaje de Programación Interpretado (*Hypertext Preprocessor*).

<sup>26</sup> Lenguaje de Consulta Estructurada (*Structured Query Language*).



- Declarar elementos de la IU<sup>27</sup> en *XML*. Android proporciona un vocabulario *XML* simple que coincide con las clases y subclases de vistas, como las que se usan para *layout*.
- Crear una instancia de elementos del diseño en tiempo de ejecución. Tu aplicación puede crear objetos *View* y *ViewGroup* con el fin de definirlos desde el código.

Los diferentes tipos de *layouts* para organizar las vistas son diseño lineal, diseño relativo o vista web, también incluyen adaptadores que son vistas de lista y vista de cuadrícula.

### 5.6.3 View

Es la clase donde se integran los componentes básicos con los que se construye la interfaz gráfica de la aplicación, es la responsable de dibujar y manejar eventos, tales como botones, campos de texto, listas desplegables, imágenes, etc. Todas las vistas en un *layout* están organizadas en un solo árbol de vistas, eso hace fácil manipular las propiedades y los métodos que las configuran para cada elemento declarado.

### 5.6.4 Service

Se ejecuta en segundo plano para realizar operaciones de larga duración o para realizar tareas para procesos remotos. Estos no cuentan con una interfaz de usuario. Por ejemplo, el reproducir música se hace en segundo plano mientras que el usuario está en una aplicación diferente. Los servicios son implementados como subclases de un servicio.

### 5.6.5 Content Provider

Los proveedores de contenido gestionan un sistema compartido de datos de una aplicación, donde se puede almacenar datos en el sistema de archivos, en una base de datos como *SQLite*, *MySQL*, *Microsoft SQL Server*, *PostgreSQL*, *Mongo DB*, *Redis*, *neo4j*, *cassandra*, en la nube, o en cualquier otro lugar de almacenamiento persistente que una aplicación puede tener acceso. Por medio de un proveedor de contenido que cuenta con aplicaciones para consultar o modificar datos. Cualquier aplicación con permisos necesarios puede consultar o modificar información de una persona en particular. Un proveedor de contenido se implementa como una subclase de *ContentProvider*.

### 5.6.6 Fragment

Se encarga de responder a mensajes de difusión en todo el sistema, muchos de estos se originan en el sistema.

Es representado como un comportamiento o una parte de la interfaz de usuario en una *Activity*. Se pueden combinar múltiples fragmentos en una sola *Activity* para crear una *IU* multipanel y volver a usar un fragmento en múltiples *activities*. Son capaces de recibir sus propios eventos de entrada, con

---

<sup>27</sup> Interfaz de Usuario.

la ventaja de agregar o quitar mientras la actividad se esté ejecutando y volver a usar en diferentes *activities*.

### 5.6.7 Broadcast Receiver

Se encarga de responder a un mensaje de difusión del propio sistema operativo u otras aplicaciones Android. Algunos eventos son: cuando hay actualizaciones para las aplicaciones, cuando el sistema operativo arranca o una aplicación no responde, así como también cuando la orientación cambia de vertical a horizontal, cuando recibe una llamada o un mensaje. El uso *BroadcastReceiver* es que una aplicación se entere de algún evento del sistema operativo y no de lo que está haciendo una aplicación.

### 5.6.7 Intent

Es la manera en que la misma aplicación se comunica consigo misma en tiempo de ejecución.

- El método *startActivity()* inicia una actividad o también obtiene la actividad existente para realizar algo nuevo.
- El método *putExtra()* funciona para enviar datos a otra *Activity*.
- El método *getIntent()* obtiene todos los datos enviados de una *Activity*.
- El método *getStringExtra()* convierte los datos obtenidos del *getIntent()* a tipo cadena.

## 5.7 AsyncTask

Definido para tareas asíncronas y unificar los aspectos relacionados que se realizarán en segundo plano en forma de hilos para la ejecución asociada a una aplicación. *AsyncTask* libera al programador del uso de hilos, la sincronización entre ellos y la presentación de resultados en el hilo primario.

La clase *AsyncTask* posee los siguientes métodos para su ejecución en segundo plano.

- *onPreExecute()*: En este método van todas aquellas instrucciones que se ejecutarán antes de iniciar la tarea en segundo plano. Normalmente es la inicialización de variables, objetos y la preparación de componentes de la interfaz.
- *doInBackground(Parámetros...)*: Recibe los parámetros de entrada para ejecutar las instrucciones específicas que irán en segundo plano, luego de que ha terminado *onPreExecute()*. Dentro de él se puede invocar un método auxiliar llamado *publishProgress()*, el cual transmitirá unidades de progreso al hilo principal. Estas unidades miden cuanto tiempo falta para terminar la tarea, de acuerdo a la velocidad y prioridad que se está ejecutando.
- *onProgressUpdate(Progreso...)*: Este método se ejecuta en el hilo de UI luego de que *publishProgress()* ha sido llamado. Su ejecución se prolongará lo necesario hasta que la

tarea en segundo plano haya sido terminada. Recibe las unidades de progreso, así que se puede usar algún View para mostrarlas al usuario y que sea consciente de la cantidad de tiempo que debe esperar.

- *onPostExecute*(Resultados...): Aquí se pueden publicar todos los resultados retornados por *doInBackground()* hacia el hilo principal.
- *onCancelled()*: Ejecuta las instrucciones que se desean realizar al cancelar la tarea asíncrona.

## 5.8 Dependencias

Las dependencias forman parte de las configuraciones de compilación personalizada, lo que permite una flexibilidad de compilar archivos locales y desde repositorios remotos sin alguna modificación, con la ventaja de evitar buscar, descargar y copiar manualmente paquetes ejecutables. Con la ayuda de *Gradle*<sup>28</sup> busca y descarga las dependencias permitiendo preconfigurar el soporte para el control remoto, los repositorios pueden ser localizados desde JCenter [36] y Maven Central [35].

## 5.9 Ciclo de vida de una *Activity*

El ciclo de vida de una *Activity* se realiza por medio de la implementación de métodos los cuales siguen a un proceso para la ejecución de la aplicación (ver Figura 2). Existen seis estados esenciales en una *Activity* y son:

- *onCreate()* se invoca al crear la *Activity*, así se encargada de realizar la configuración.
- *onRestart()* se invoca si se reinicia la *Activity*.
- *onStart()* se invoca cuando la *Activity* es visible en pantalla para el usuario.
- *onResume()* se invoca cuando la *Activity* comienza a interactuar con el usuario.
- *onPause()* se invoca al detener la *Activity* o al reclamar CPU u otros recursos. En este método se guarda el estado para que al reiniciar una *Activity* pueda comenzar con el mismo estado con el que se finalizó.
- *onStop()* se invoca para detener la *Activity* y pasar a una fase de invisibilidad y posteriores eventos de ciclo de vida.
- *onDestroy()* se invoca al eliminar una *Activity* de la memoria del sistema. Se produce por invocar directamente *onFinish()* o cuando el sistema decide detener la *Activity* para liberar recursos.

---

<sup>28</sup> Paquete de herramientas de compilación avanzada, para autorizar, administrar, configurar y flexibilidad en la compilación de las aplicaciones.

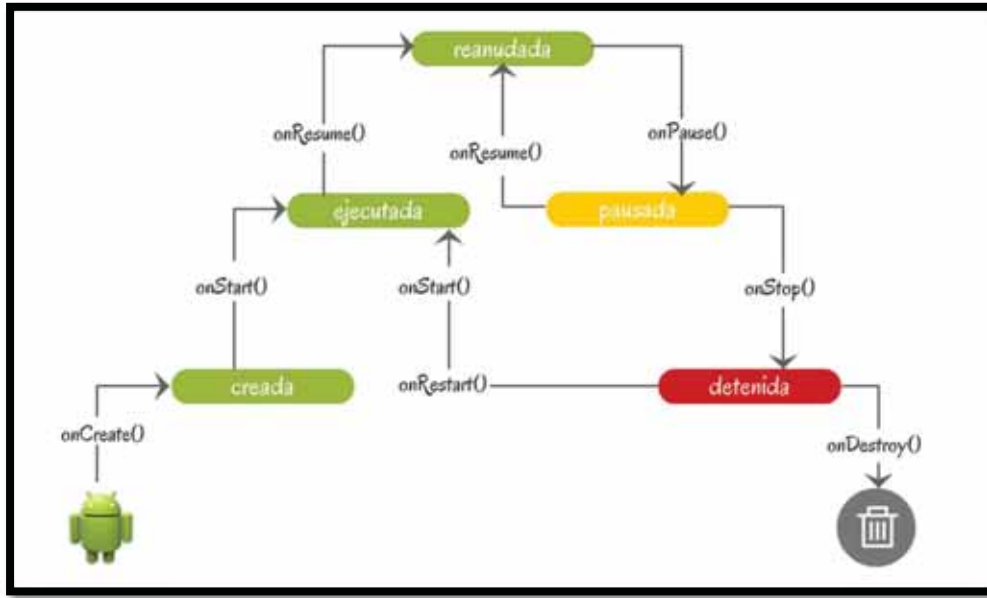


Figura 2. Ciclo de vida de una *Activity* [37] .

## 5.10 Aplicación Web

La aplicación Web tiene este nombre por efectuar una función a través de la internet. Los datos o archivos con los que trabaja son procesados y almacenados dentro de la Web, es decir toda la información se guarda de forma permanente en grandes servidores para que los usuarios los puedan consultar por medio de sus dispositivos móviles.

Desde cualquier lugar, momento y dispositivo móvil se pueden acceder a los servicios, sólo se necesita una conexión a internet o intranet. Algunas aplicaciones Web son el correo electrónico, google docs, Facebook, entre otros.

## 5.11 API REST

Es una arquitectura de servicios distribuidos, se pueden encontrar recursos dada una *URL*<sup>29</sup> y mediante el protocolo *HTTP*<sup>30</sup>. Mediante cabeceras especiales como *Accept* o *Content-Type* se puede especificar que representaciones entiende el servidor y el cliente y que representación se usa en un mensaje para transporte del estado del recurso. Las representaciones a usar se especifican mediante los llamados tipos mime, ya sea *GET* para leer, *DELETE* para borrar, *PUT* para actualizar y *POST* para crear. Algunos de ellos son estándares, como *XML*<sup>31</sup> o *JSON*<sup>32</sup>.

<sup>29</sup> Localizador Uniforme de Recursos (*Uniform Resource Locator*).

<sup>30</sup> Protocolo de Transferencia (*Hypertext Transfer Protocol*).

<sup>31</sup> Lenguaje de Marcas Extensible (*eXtensible Markup Language*).

<sup>32</sup> Objetos de Notación de JavaScript (*JavaScript Object Notation*).

*HTTP* es un protocolo que sigue los principios de *REST*, por lo tanto, hacer servicios *REST* es algo que aprovecha toda la infraestructura de la web ya existente. La ventaja es que no necesita algo como un *WSDL*<sup>33</sup>.

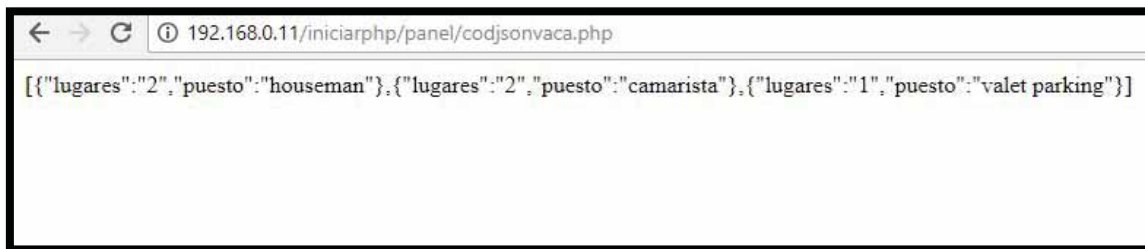
## 5.12 JSON

Es un formato para el intercambio de datos, la sintaxis que usa es para identificar y gestionar datos. JSON está basado en *JavaScript* por lo que usa convenciones que son ampliamente conocidas por lenguajes de programación de alto y medio nivel [38] .

Utiliza dos estructuras universales, las cuales son:

- *Objeto*, que es una colección de pares de nombre/valor y,
- Una lista ordenada de valores como arreglos, vectores, listas o secuencias.

Un objeto se representa con “{“ y “}”, adentro pueden haber miembros y parejas las cuales son representadas por un nombre “:” y un valor, seguido de una “,” de existir más. Un arreglo comienza con “[“ y “]”, de existir más se deberá escribir una “,”, los valores son representados como cadena de caracteres, números, verdadero, falso, nulo así como objetos y arreglos anidados (ver Figura 3).



```
[{"lugares": "2", "puesto": "houseman"}, {"lugares": "2", "puesto": "camarista"}, {"lugares": "1", "puesto": "valet parking"}]
```

Figura 3. Ejemplo de formato Json para Vacantes.

<sup>33</sup> Lenguaje de Descripción de Servicios Web (*Web Services Description Language*).

## 6. Desarrollo del Proyecto

En este capítulo se especifica el proceso de elaboración de la aplicación móvil por lo que se emplean los conocimientos del capítulo 5. Marco Teórico, requerimientos de hardware y software, el modelo arquitectónico, funcionalidades específicas de la aplicación móvil, así como del servidor y una base de datos relacional. Cabe mencionar que se modificó el porcentaje de las actividades que se realizaron en mayor o menor medida de acuerdo a los estándares de la empresa.

### 6.1 Investigación previa

El primer paso fue hacer una revisión de la *API* 19 de Android 4.4 como punto de partida, ya que, por cuestiones de mercadotecnia, la mayoría de los usuarios cuentan con móviles que tienen esta versión, además de los permisos y dependencias para el uso de la *API* de Android y hardware del dispositivo [40].

Cabe destacar que en esta sección se definen las funcionalidades específicas que la aplicación Android debe cumplir para visualizar todos los datos necesarios para estar al día con la información que maneja el centro de convenciones y así como también los servicios de hotelería; sin embargo, debido a las políticas y estándares se modificó el porcentaje de las actividades que se realizaron en mayor o menor medida de acuerdo a lo permitido por el corporativo y la red de la empresa.

A continuación, se enlistan los objetivos específicos que se tuvieron que modificar por cuestiones de seguridad:

Original	Modificado
Diseñar e implementar un módulo para generar y notificar el envío de un reporte al finalizar el proceso de solicitud, también se guardará como historial del trabajador.	Diseñar e implementar un módulo que visualice la resolución al finalizar el proceso de solicitud, también guardará y mostrará como historial del trabajador.
Diseñar e implementar un módulo para generar una firma electrónica avanzada, para su autorización y finalización del trámite interno solicitado por el usuario.	Diseñar e implementar un módulo que genere y decodifique una firma electrónica avanzada, para su autorización y finalización del trámite interno solicitado por el usuario.

### 6.2 Requerimientos

- Laptop Sistema Operativo Windows 10 Home de 64bits, Intel® Celeron® CPU N2830 2.16GHZ
- Laptop Sistema Operativo Linux/Ubuntu 14.04.5 LTS, Intel® Core® Duo T2250
- Dispositivo Móvil con Sistema Operativo Android 4.4 o superior.

- El teléfono móvil debe contar con conexión de datos inalámbrico, *WI-FI*.
- Cada usuario debe contar con un usuario y contraseña para poder solicitar, aprobar o cancelar dichos trámites internos de la empresa.

El sistema implementado es de tipo cliente/servidor, por lo que la aplicación realizada debe conectarse como cliente al servidor para consultar o solicitar algún trámite que requiera el usuario, es por ello que se emuló un servidor con una Laptop Gateway Sistema Operativo GNU/Linux Ubuntu 10.04 configurada de tipo *LAMP*<sup>34</sup> de tal manera que la conexión sea *INTRANET*<sup>35</sup> (ver Figura 4).



Figura 4. Conexión Intranet.

Utilizando las herramientas de programación propuestas para desarrollar una aplicación Web que involucra a un cliente en el caso de “*AuDiFoHHn*” y por el otro lado a un servidor, enseguida se enlistan los siguientes paquetes de softwares utilizados:

<sup>34</sup> *Linux-Apache-MySQL-Php*

<sup>35</sup> Es una red informática que utiliza protocolo de internet para compartir información de manera interna en una organización.

#### Servidor de la Base de Datos para Linux

- Gestor de *BD MariaDB* 10.2.8 derivado de *MySQL*
- *MySQL* 5.7.19
- Administrados de *BD PhpMyAdmin* 4.7.4

#### Servidor de peticiones Web para Linux

- *Apache2* 2.4.27
- *Php* 5.6.31

#### Cliente Android para Windows

- *Java JDK* 8 actualización 131
- *Java JRE* 8 actualización 161
- *IDE* Android Studio 2.3.3
- *SDK* Android *API* 19
  - Librerías o Dependencias:
    - ◆ *'com.loopj.android:android-async-http:1.4.9'*
    - ◆ *'com.github.snowdream.android:smartimageview:0.0.2'*
    - ◆ *'com.squareup.picasso:picasso:2.5.2'* de *hdodenhof:circleimageview:2.1.0'*
    - ◆ *'com.journeyapps:zxing-android-embedded:2.0.1@aar'*  
*'com.journeyapps:zxing-android-legacy:2.0.1@aar'*  
*'com.journeyapps:zxing-android-integration:2.0.1@aar'*  
*'com.google.zxing:core:3.0.1'*

### 6.3 Modelo Arquitectónico de la aplicación

El patrón de arquitectura de software de la aplicación es el Modelo-Vista-Controlador (*MVC*) constituido por una interfaz de usuario y lógica de negocios en tres componentes distintos que se relacionan para al final tener como resultado el sistema “*AuDiFoHHN*”.

La descripción de los componentes es la siguiente:

- **Modelo:** es referido a la representación desarrollada basada en la información con la que operará la aplicación.
- **Vista:** es la interfaz con la que el usuario podrá interactuar con la aplicación. En Android, las interfaces son archivos *XML*.
- **Controlador:** es aquel encargado de crear y depurar la aplicación.



## 6.4 Funcionamiento de la aplicación

Se establece una comunicación cliente – servidor para la aplicación, donde el cliente es *AuDiFoHHn* instalada en el dispositivo móvil, en tanto el servidor llamado “Cerebro” es una computadora independiente a los dispositivos móviles y donde se encuentra la base de datos.

## 6.5 Funcionamiento del servidor

Cerebro es el encargado de realizar la sincronización de la gestión de los trámites y eventos de la empresa. Las operaciones que se realizan es registrar, consultar, eliminar y actualizar.

Toda la información es generada en código *JSON* para poder ser consumida por *AuDiFoHHn* a través del protocolo *HTTP* en una red local(intranet).

## 6.6 Base de datos

En la documentación entregable para este proyecto se anexa el script fuente para su consulta con el nombre de “*phpgaleriafotos.sql*”, dicho archivo esta localizado con la ruta en el explorador de archivos de *Windows E:\BD\phpgaleriafotos.sql*.

La base de datos remota (ver Figura 5) que está en Cerebro consta de las siguientes tablas:

- ∴ area: es una ramificación del departamento especializada en una actividad específica (tabla1).
- ∴ departamento: contiene las secciones en la que está dividida la empresa (tabla2).
- ∴ estacionamiento: se registran todas las peticiones de aquellos trabajadores que necesitan estacionamiento (tabla3).
- ∴ fotos: guarda todos los eventos y actividades que se llevaran a cabo durante el mes (tabla4).
- ∴ manager: se encuentra las cuentas de los gerentes para autorizar las peticiones de los empleados (tabla5).
- ∴ permisosin: se almacena un registro de todos los permisos sin goce de sueldo de cualquier miembro de equipo (tabla6).
- ∴ permisocon: se almacena un registro de todos los días autorizados como permiso con goce de cualquier miembro de equipo (tabla7).
- ∴ registrovacaciones: se guarda el registro de todas las vacaciones que cualquier trabajador pudiera pedir (tabla8).
- ∴ salt: es una medida de calificación la cual es referida al “Servicio en general” o en otras palabras “El servicio al cliente” (tabla9).
- ∴ tabudiasvacaciones: contiene la tabulación de días tomados y su correspondiente descanso (tabla10).

- ∴ tabuvacaciones: contiene la tabulación de los días de vacaciones correspondientes por antigüedad laboral, así como también los días de descansos correspondientes (tabla11).
- ∴ trabajador: se encuentra toda la información necesaria de cada trabajador para las solicitudes (tabla12).
- ∴ vacantes: almacena todas las vacantes (tabla13).

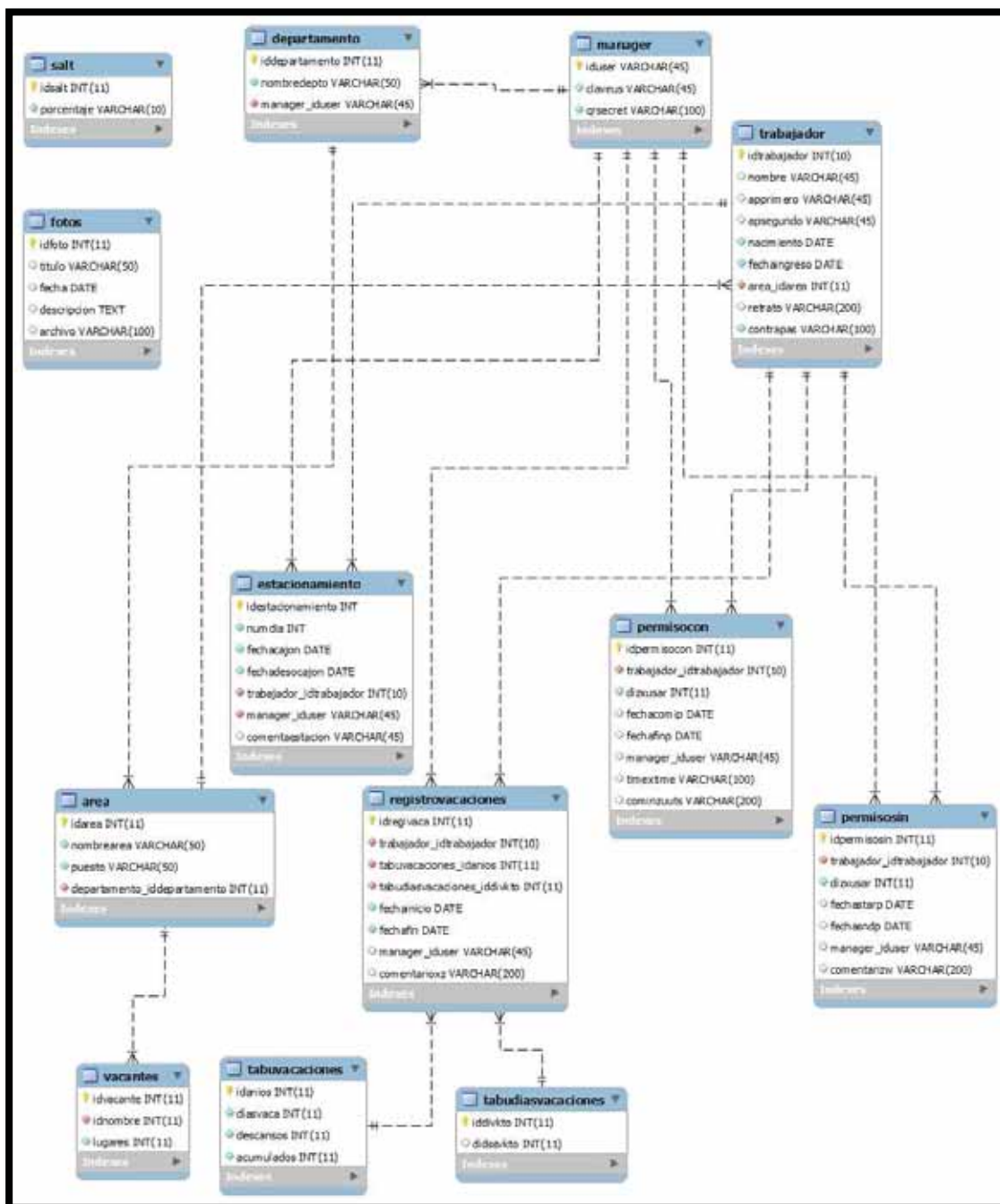


Figura 5. Diagrama da la base de datos “phpgaleriafotos.sql”.

Área								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>Idare</b>	Int(11)			X		X	X	Identificador del área
<b>Nombreaea</b>	Varchar(50)			X				Nombre del área
<b>Puesto</b>	Varchar(50)			X				El puesto que de cada área
<b>departamento_iddepartemento</b>	Int(11)			X				Identificador de cada departamento

Tabla1. Especificaciones de la tabla área.

Departamento								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>Iddepartamento</b>	Int(11)			X		X	X	Identificador de cada departamento
<b>Nombredepto</b>	Varchar(50)			X				Nombre del departamento
<b>manager_iduser</b>	Varchar(45)			X				Identificador de usuario para cada gerente

Tabla2. Especificaciones de la tabla departamento.

<b>Estacionamiento</b>								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>Idestacionamiento</b>	Int(11)			X		X	X	Folio petición estacionamiento
<b>Numdia</b>	int(11)			X				Número de días solicitados
<b>Fechacajon</b>	Date			X				Fecha inicio de solicitud
<b>Fechadesocajon</b>	Date			X				Fecha final de solicitud
<b>Trabajador_ Idtrabajador</b>	Int(11)			X				Identificador del trabajador
<b>Manager_iduser</b>	Varchar(45)			X				Identificador del gerente

Tabla3. Especificaciones de la tabla estacionamiento.

<b>Fotos</b>								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>Idfoto</b>	Int(11)			X		X	X	Identificador de galeria
<b>Titulo</b>	Varchar(50)		X					Título de cada evento o actividad
<b>Fecha</b>	Date			X				Fecha en la que sucederá cada actividad o evento
<b>descripción</b>	Text			X				Descripción de cada evento o actividad
<b>archivo</b>	Varchar(100)		X					Nombre de la imagen.

Tabla4. Especificaciones de la tabla fotos.

Manager								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>Iduser</b>	Varchar(45)			X				Identificador de usuario para los gerentes
<b>claveus</b>	Varchar(45)			X				Clave secreta de cada gerente
<b>qrsecret</b>	Varchar(100)			X				Clave que será en código QR para cada gerente

Tabla5. Especificaciones de la tabla manager.

Permisocon								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>Idpermisocon</b>				X		X	X	Identificador de permiso sin goce de sueldo
<b>trabajador_idtrabajador</b>				X				Identificador único para cada trabajador
<b>Dizxusar</b>				X				Días que pide un trabajador
<b>Fechacomip</b>			X					Fecha de inicio del permiso
<b>Fechafinp</b>			X					Fecha fin del permiso
<b>manager_iduser</b>			X					Identificador del gerente
<b>Timextime</b>			X					Justifica el motivo por el cual se paga el día
<b>Cominzuuts</b>			X					Comentario referente al permiso

Tabla6. Especificaciones de la tabla permisocon.

Permisosin								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>idpermisosin</b>				X		X	X	Identificador de permiso sin goce de sueldo
<b>trabajador_ idtrabajador</b>				X				Identificador único para cada trabajador
<b>dizxusar</b>				X				Días que pide un trabajador
<b>fechastarp</b>			X					Fecha de inicio del permiso
<b>fechasndp</b>			X					Fecha fin del permiso
<b>manager_ iduser</b>			X					Identificador del gerente
<b>comentarizw</b>			X					Comentario referente al permiso

Tabla7. Especificaciones de la tabla permisosin.



Registrovacaciones								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>Idregivaca</b>	Int(11)			X		X	X	Identificador de vacaciones
<b>trabajador_ idtrabajador</b>	Int(11)			X				Identificador de cada trabajador
<b>tabuvacaciones_ idanios</b>	Int(11)			X				Identificador del tabulador de días vacaciones
<b>tabudiasvacaciones_ iddivkto</b>	Int(11)			X				Identificador del tabulador de días con descanso
<b>Fechainicio</b>	Date			X				Fecha de inicio de vacaciones
<b>Fechafin</b>	Date			X				Fecha de fin de vacaciones
<b>manager_iduser</b>	Varchar(45)		X					Identificador de usuario de cada gerente
<b>Comentarioz</b>	Varchar(200)			X				Comentario respecto a las vacaciones

Tabla8. Especificaciones de la tabla registrovacaciones.

Salt								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>Idsalt</b>	Int(11)			X		X	X	Identificador del servicio al cliente
<b>porcentaje</b>	Varchar(10)			X				La cifra de cada dia

Tabla9. Especificaciones de la tabla salt.

Tabudiasvacaciones								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>iddivkto</b>	Int(11)			X		X	X	Identificador de días tomados
<b>didssvkto</b>	Int(11)		X					Días de descanso correspondiente por días tomados

Tabla10. Especificaciones de la tabla tabudiasvacaciones.

<b>Tabuvacaciones</b>								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>idanos</b>	Int(11)			X		X	X	Identificador de tabulaciones por aniversario
<b>diasvaca</b>	Int(11)			X				Días que corresponden por año
<b>descansos</b>	Int(11)			X				Descansos que corresponden por días
<b>acumulados</b>	Int(11)			X				Tabulación de días acumulados

Tabla11. Especificaciones de la tabla tabuvacaciones.

<b>Trabajador</b>								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>idtrabajador</b>	Int(10)			X			X	Identificador único para cada trabajador
<b>nombre</b>	Varchar(45)		X					Nombre del trabajador
<b>appprimero</b>	Varchar(45)		X					Apellido paterno del trabajador
<b>apsegundo</b>	Varchar(45)		X					Apellido materno del trabajador
<b>nacimiento</b>	Date			X				Fecha de nacimiento del trabajador
<b>fechaingreso</b>	Date			X				Fecha de ingreso del trabajador
<b>area_idarea</b>	Int(11)			X				Área en la que trabaja
<b>retrato</b>	Varchar(200)		X					Imagen de perfil del trabajador
<b>contrapas</b>	Varchar(100)			X				Clave de acceso al sistema AuDiFoHHn

Tabla12. Especificaciones de la tabla trabajador.

Vacante								
	Tipo	Unique	Null	Not null	Default	Autoincrement	PK	Description
<b>idvacante</b>	Int(11)			X		X	X	Identificador de la vacante
<b>idnombre</b>	Int(11)			X				Nombre de la vacante
<b>Lugares</b>	Int(11)			X				Plazas disponibles

Tabla13. Especificaciones de la tabla vacante.

## 6.7 Procedimiento

Se describe el proceso de la aplicación móvil “AuDiFoHHN” así como la implementación del servidor “Cerebro” para un futuro seguimiento.

Antes de iniciar es importante montar la base de datos, tener instaladas las herramientas para un servidor de peticiones Web y servidor de bases de datos mencionadas en la sección 6.2 de este mismo capítulo. Los pasos para importar la base de datos son los siguientes:

1. Abrir el navegador Web
2. Escribir en la barra de direcciones la siguiente URL:  
*http://localhost/phpmyadmin/index.php.*
3. Escribir usuario y contraseña en el login (ver Figura 6).



Figura 6. Login del gestor y administrador de bases de datos.

4. Seleccionar la opción de importar y dar clic en el botón Seleccionar archivo (ver Figura 7).



Figura 7. Importar la base de datos

5. En el explorador de Windows acceder a la siguiente ruta *E:\BD\phpgaleriafotos.sql* y dar clic en el botón Abrir (ver Figura 8).

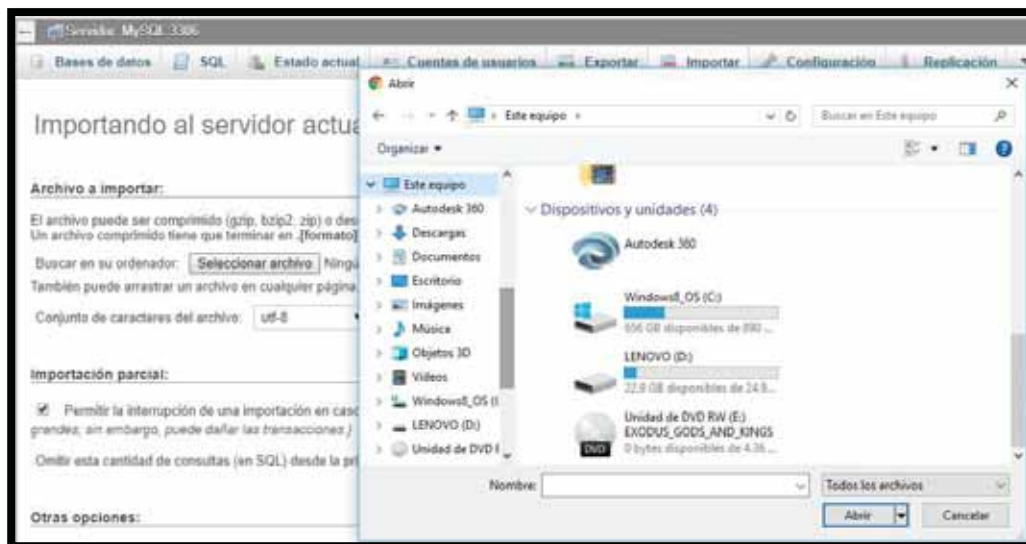


Figura 8. Explorador de Windows.

6. En la categoría de formato desplegar el menú de opciones para seleccionar *SQL*, que es la extensión del script de la base de datos (ver Figura 9).

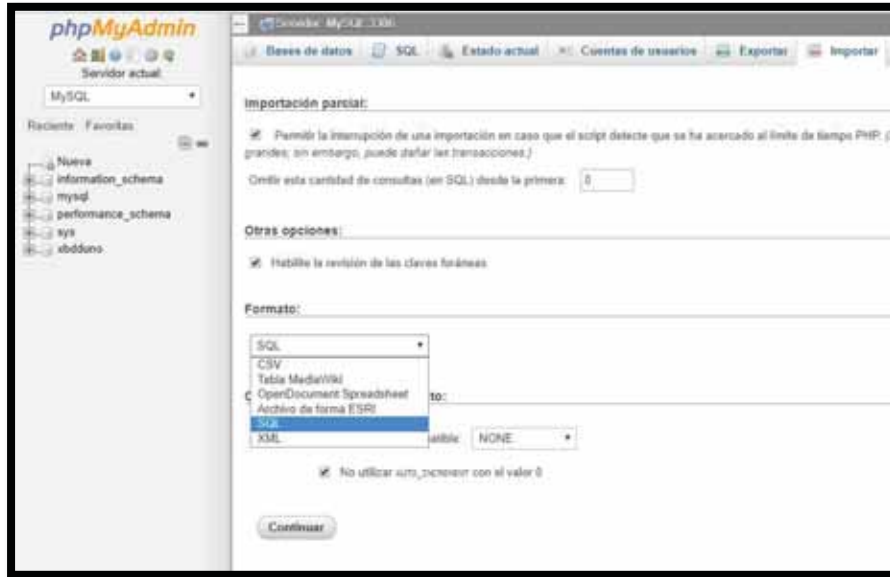


Figura 9. Menú de extensiones.

7. Dar clic al botón Continuar para ejecutar el script, enseguida se mostrará un mensaje “Importación ejecutada exitosamente, 44 consultas ejecutadas. (phpgaleriafotos.sql)” (ver Figura 10).

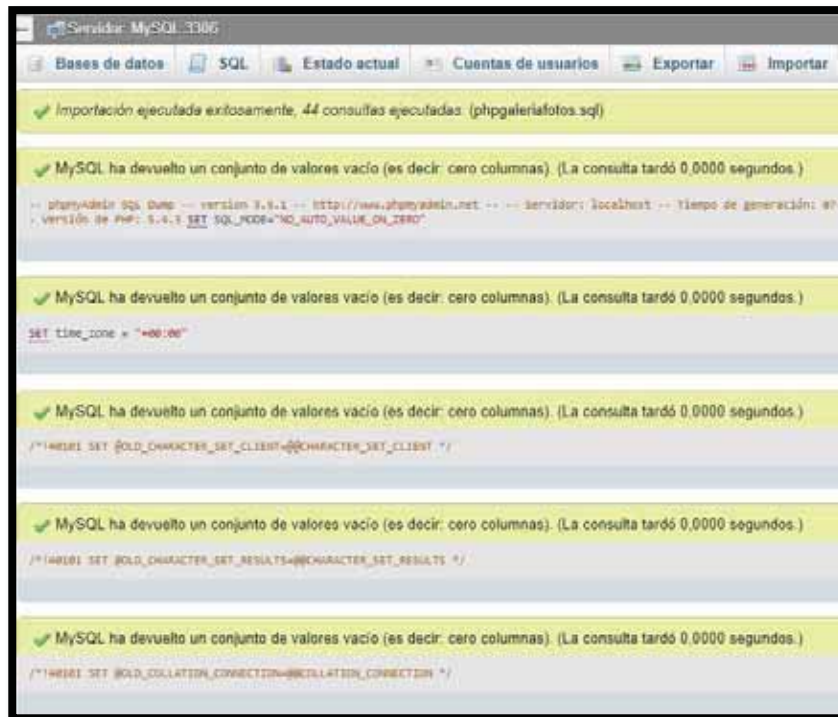


Figura 10. Ejecución exitosa.

Una vez que el *SDK* de Android Studio este en ejecución se tendrá que importar el proyecto desde el menú de herramientas Archivo – Abrir con la siguiente ruta *E:\PI\DocPT*, para entrar a las clases del código fuente con sus respectivos métodos y funciones, los pasos son los siguientes:

1. Expandir la carpeta de la barra de la ventana de herramientas en la pestaña 1: *Project – app – java – roberto.espejel.app.proyecto.docpt.rulaitas.ian*.
2. Dar doble clic a la clase para entrar al código fuente de los métodos y funciones específicas que realiza (ver Figura 11).

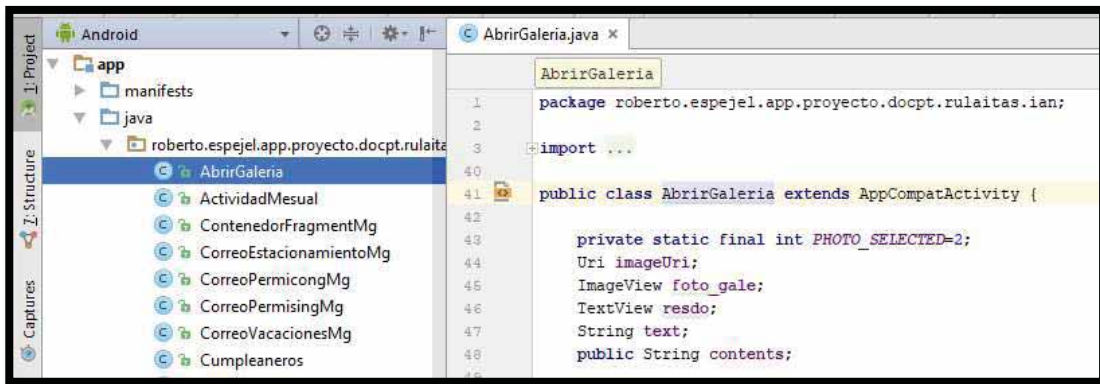


Figura 11. Barra de la ventana de herramienta vista Android.

A cada clase le corresponde un archivo con extensión *XML* según sea el caso que efectúa, y son representados como la interfaz gráfica para el usuario. Para acceder a los archivos *XML* es necesario seguir los siguientes pasos:

1. Expandir la carpeta de la barra de la ventana de herramientas en la pestaña 1: *Project – app – res – layout*.
2. Dar doble clic al archivo con extensión *XML* para mostrar en la ventana del editor la vista final de usuario de forma gráfica o en código *XML*.
3. Dar clic en la opción Text para acceder al código *XML* o *Design* para hacerlo de forma gráfica (ver Figura 12).



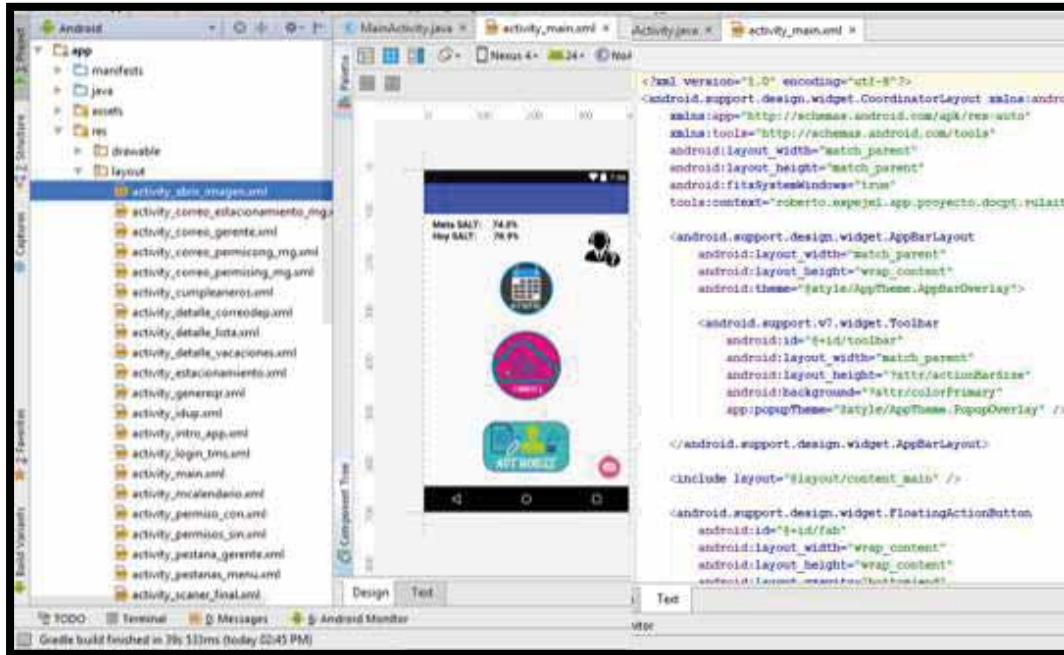


Figura 12. Ventana del editor con vista gráfica (izquierda) y con vista código XML (derecha).

En la carpeta *app - res - drawable* se encuentran las imágenes personalizadas para las pestañas, las animaciones de cambio de color a los botones y el color de fondo personalizado para cada interfaz gráfica, la animación del cambio de color del botón *Connect* para el *login* de trabajadores y de gerentes fue creada en código *XML* al igual que el color de fondo para cada pantalla, es importante mencionar que las imágenes con extensión *BMP*<sup>36</sup>, *GIF*<sup>37</sup>, *JPEG*<sup>38</sup> y *PNG*<sup>39</sup> son válidas para el recurso y fueron creadas con una escala de 30X30 con una resolución de 72 *DPI*<sup>40</sup> dependiendo la resolución de la pantalla del teléfono móvil a la cual sea dirigido (ver Figura 13), en este caso se empleó para una pantalla con la resolución de 540 x 960 *PPI*<sup>41</sup> lo que es asignada como extensión *hdpi* dentro de la carpeta *drawable* (ver Figura 14).

<sup>36</sup> Mapa de Bits (*Bitmap*)

<sup>37</sup> Formato de Intercambio de Gráficos (*Graphics Interchange Format*)

<sup>38</sup> Grupo Conjunto de Expertos en Fotografía (*Join Photographic Experts Group*)

<sup>39</sup> Gráficos de Red Portátiles (*Portable Network Graphics*)

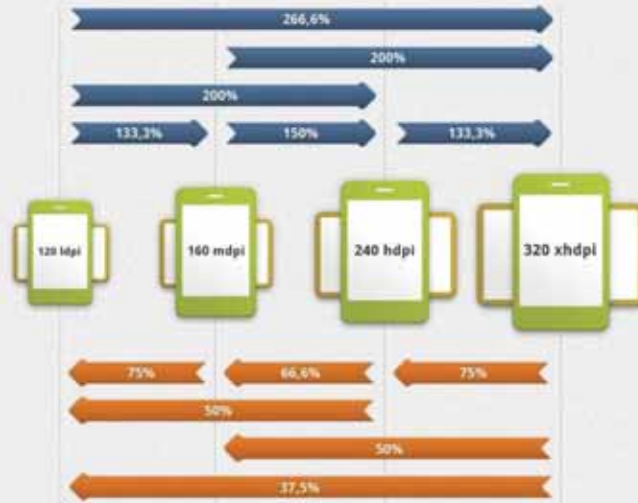
<sup>40</sup> Puntos Por Pulgada (PPP) (*Dots Per Inch*)

<sup>41</sup> Píxeles Por Pulgada (PPI) (*Pixels Per Inch*)



ANDROID

DENSIDADES DE PANTALLA



RESOLUCIONES COMUNES

MÓVIL



240x400 px  
240x432 px



320x480 px



480x800 px  
480x854 px  
600x1024 px

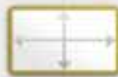


640x960 px

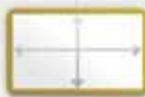
TABLET



1024x600 px



1280x800 px  
1024x768 px  
1280x768 px



1536x1152 px  
1920x1152 px  
1920x1200 px



2048x1536 px  
2560x1536 px  
2560x1600 px

ICONO DE LA APP



36 px



48 px



72 px



96 px

Figura 13. Tabla para el diseño de iconos [39] .

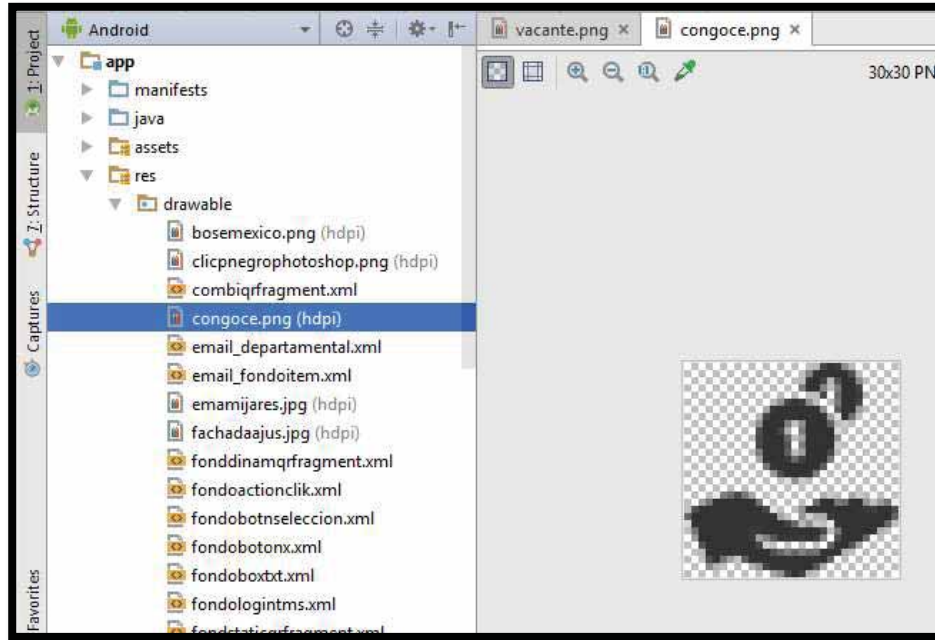


Figura 14. Recurso *drawable*.

En la carpeta *app – res – mipmap* se hallan todas las imágenes que son utilizadas como botón de imagen (ver Figura 15).

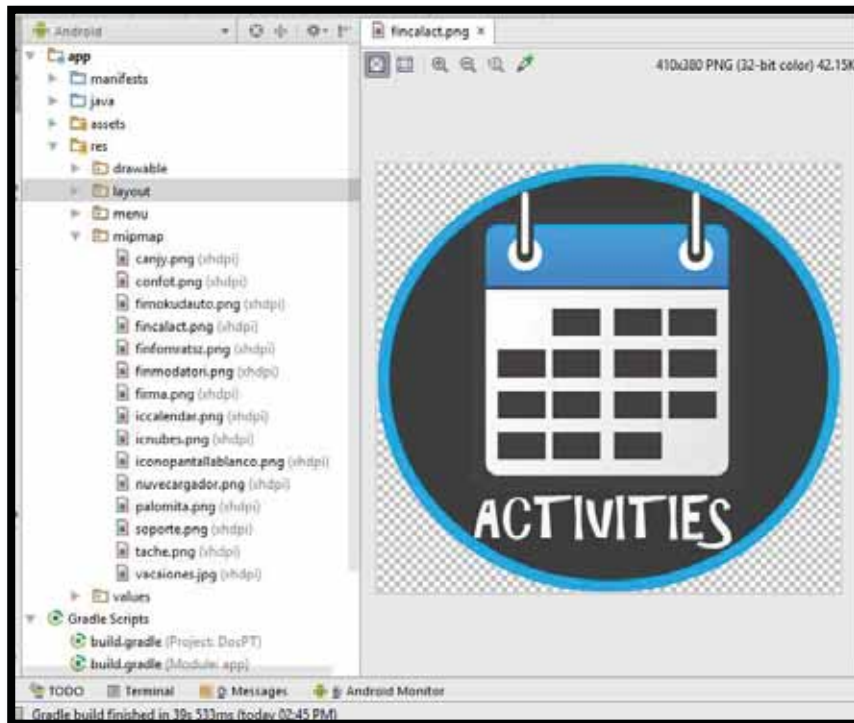


Figura 15. Recurso *mipmap*.

En la carpeta *app – res – values – strings.xml* se encuentran todas las cadenas que se utilizan en la interfaz gráfica para el usuario final, ya sea en títulos, encabezados de las pantallas, botones, etiquetas, menú, etc. (ver Figura 16).

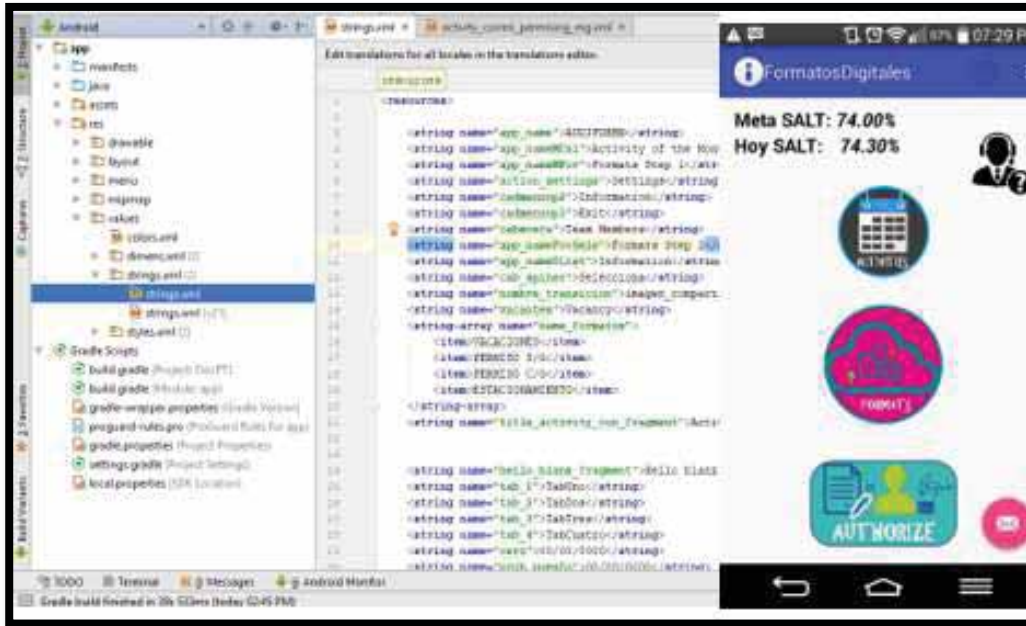


Figura 16. Archivo strings.xml (izquierda) e interfaz de usuario (derecha).

Otro archivo muy importante que es necesario mencionar es *Gradle Scripts – build.gradle (Module: app)*, ya que es lugar donde se encuentran todas las dependencias externas seguidas de la etiqueta “*compile*” (ver Figura 17).

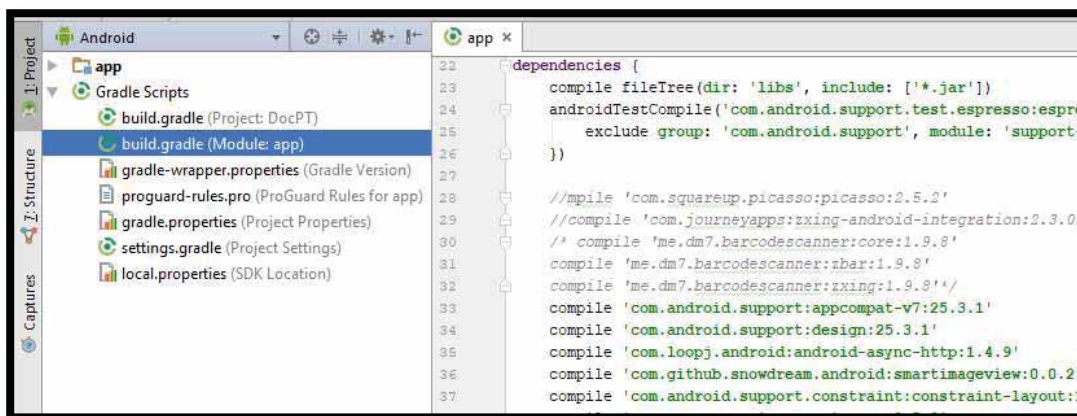


Figura 17. Recurso build.gradle (Module:app).

## **6.8 Uso del Sistema**

El manual de usuario en formato video para su mejor ilustración en el manejo de la aplicación en sus diferentes módulos.

## 7. Resultados

A continuación, se detallan los resultados obtenidos del desarrollo de este proyecto:

La pantalla principal está compuesta por el indicador del SALT, un botón de actividades, un botón de formatos, un botón de autorización de formatos y otro botón para generar la firma electrónica avanzada (ver Figura 18).



Figura 18. Pantalla principal.

El módulo de actividades cuenta con la pestaña de cumpleaños, actividades y vacantes en el mes (ver Figura 19).



Figura 18. Pestaña de cumpleaños, actividades del mes y vacantes.

El módulo de formatos cuenta con un login para los trabajadores (ver Figura 19), después de que el usuario y la contraseña sean correctos se muestra el módulo que llena automáticamente los datos necesarios del trabajador (ver Figura 20) junto con un menú para elegir un formato (ver Figura 21) seguida de un calendario y botón de registro para ver todo el historial de solicitudes hechas (ver Figura 22).



Figura 19. Login trabajadores.



Figura 20. Datos del trabajador

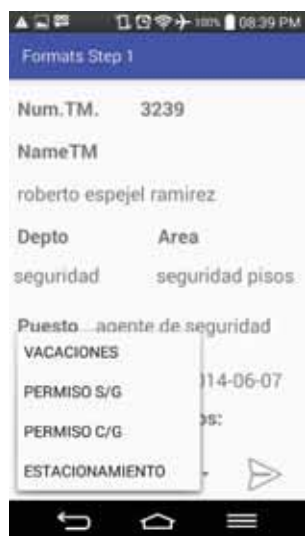


Figura 21. Menú de trámites.

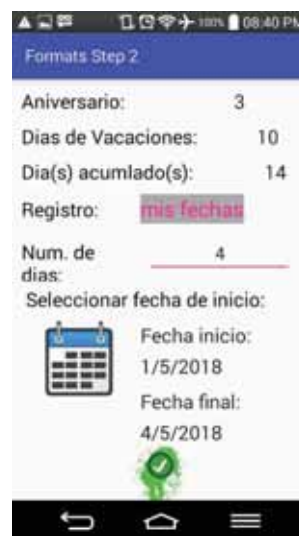


Figura 22. Calendario para elegir los días a solicitar.

El módulo de historial muestra un registro de todas las solicitudes hechas por el miembro de equipo (ver Figura 23) con tres tipos de resolución las cuales son: aceptada (ver Figura 24), cancelada (ver Figura 24) o pendiente (ver Figura 25).

Folio	Inicio	Fin
1	2017-11-25	2017-11-26
2	2017-11-27	2017-11-29
3	2018-01-05	2018-01-10
5	2017-12-26	2017-12-26
6	2017-12-26	2018-00-26
11	2018-05-01	2018-05-04

Figura 23. Historial de solicitudes.



Figura 24. Resolución de tipo cancelada con un breve comentario



Figura 25. Resolución de tipo aceptada.

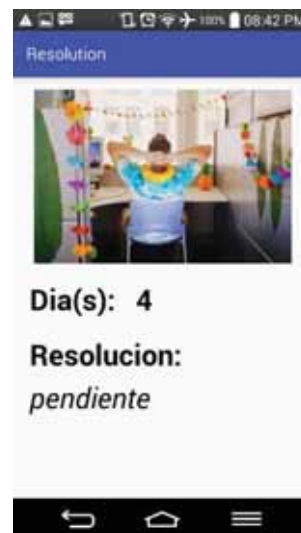


Figura 26. Resolución de tipo pendiente.

El módulo autorización cuenta con un login para los gerentes (ver Figura 27), una vez que el usuario y contraseña son válidos se muestra las solicitudes de todos los trabajadores relacionados al departamento del gerente, cada petición indica si fue aceptada, rechazada o está en proceso (ver Figura 28).





Figura 27. Login de gerente departamental.

Figura 28. Solicitudes con su respectiva resolución.

El módulo de autorización también cuenta con un menú de botones (ver Figura 29) para aceptar o cancelar alguna solicitud, en el caso de cancelar se muestra un cuadro de texto para algún comentario (ver Figura 30) y por el otro lado para aceptar se puede elegir si el código QR es leído por la cámara del teléfono (ver Figura 31) o por medio del explorador de archivo con una imagen de la firma electrónica avanzada (ver Figura 32).

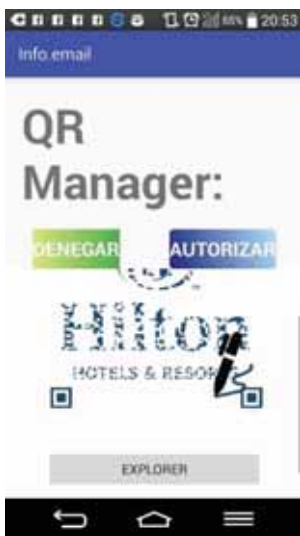


Figura 29. Menú de elección para autorizar solicitud.

Figura 30. Cancelar la solicitud.



Figura 31. Aceptar la solicitud por medio de escanear el código QR con la cámara del teléfono inteligente.

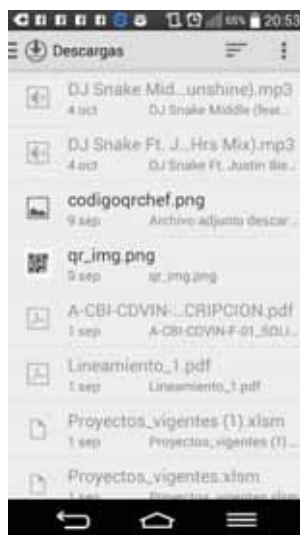


Figura 32. Aceptar la solicitud por medio del explorador de archivo a través de una imagen como firma electrónica avanzada.

Las pruebas realizadas en la aplicación “*AuDiFoHHn*” fueron registrar a grupos de diez trabajadores a cuatro departamentos diferentes. Cada trabajador por medio de la aplicación solicitó un trámite de vacaciones, el segundo un trámite de permiso sin goce de sueldo, otro a permiso con goce de sueldo y uno más solicitó estacionamiento.

Convencionalmente el gestionar la papelería antes mencionada comienza con el llenado de los datos laborales del empleado, marcar el tipo de trámite, indicar fecha de inicio y fin, comentarios opcionales y, firma del gerente departamental, del miembro de equipo y recursos humanos. El gerente de recursos humanos es el último en firmar después de haber validado la información por sistema. La respuesta es regresada al gerente departamental y por último al trabajador que realizó la petición.

La aplicación móvil consulta los datos actualizados en tiempo real, por lo que reduce el proceso de validación que realiza el departamento de recursos humanos, por tanto, la comunicación ya no es efectuada por intermediarios. Sin presentar problema alguno todas las solicitudes en la sesión de un gerente fueron aprobadas y denegadas con éxito; mostrando en la cuenta de trabajadores la respuesta final de cada trámite.

En las tablas 14 a 16 se muestran los tiempos tomados a la respuesta de cada solicitud sin el uso de la aplicación desarrollada.

Trámites X Factores	Recursos humanos	Gerente con carga de trabajo de 100% al 80%	Tiempo de respuesta
Vacaciones	24 horas	24 a 36 horas	48 a 60 horas
Permiso con goce de sueldo	24 horas	24 a 36 horas	48 a 60 horas
Permiso sin goce de sueldo	24 horas	24 a 36 horas	48 a 60 horas
Estacionamiento	24 horas	6 a 12 horas	30 a 36 horas

Tabla 14. Tiempos sin la aplicación con carga de trabajo gerencial del 100% a 80%.

Trámites X Factores	Recursos humanos	Gerente con carga de trabajo de 80% al 60%	Tiempo de respuesta
Vacaciones	24 horas	12 a 24 horas	36 a 48 horas
Permiso con goce de sueldo	24 horas	12 a 24 horas	36 a 48 horas
Permiso sin goce de sueldo	24 horas	12 a 24 horas	36 a 48 horas
Estacionamiento	24 horas	3 a 6 horas	27 a 30 horas

Tabla 15. Tiempos sin la aplicación con carga de trabajo gerencial del 80% al 60%.

Trámites X Factores	Recursos humanos	Gerente con carga de trabajo de 60% al 50%	Tiempo de respuesta
Vacaciones	24 horas	4 a 12 horas	28 a 36 horas
Permiso con goce de sueldo	24 horas	4 a 12 horas	28 a 36 horas
Permiso sin goce de sueldo	24 horas	4 a 12 horas	28 a 36 horas
Estacionamiento	24 horas	1 a 3 horas	25 a 27 horas

Tabla 16. Tiempos sin la aplicación con carga de trabajo gerencial del 60% al 50%.

En las tablas 17 a 19 indican los tiempos tomados a la respuesta de cada solicitud con el uso de la aplicación propuesta.

Trámites X Factores	Recursos humanos	Gerente con carga de trabajo de 100% a 80%	Tiempo de respuesta
Vacaciones	0 horas	16 a 24 horas	16 a 24 horas
Permiso con goce de sueldo	0 horas	16 a 24 horas	16 a 24 horas
Permiso sin goce de sueldo	0 horas	16 a 24 horas	16 a 24 horas
Estacionamiento	0 horas	4 a 8 horas	4 a 8 horas

Tabla 17. Tiempos con la aplicación con carga de trabajo gerencial del 100% a 80%.

Trámites X Factores	Recursos humanos	Gerente con carga de trabajo de 80% al 60%	Tiempo de respuesta
Vacaciones	0 horas	8 a 16 horas	8 a 16 horas
Permiso con goce de sueldo	0 horas	8 a 16 horas	8 a 16 horas
Permiso sin goce de sueldo	0 horas	8 a 16 horas	8 a 16 horas
Estacionamiento	0 horas	2 a 4 horas	2 a 4 horas

Tabla 18. Tiempos con la aplicación con carga de trabajo gerencial del 80% al 60%.

Trámites X Factores	Recursos humanos	Gerente con carga de trabajo de 60% al 50%	Tiempo de respuesta
Vacaciones	0 horas	4 a 8 horas	4 a 8 horas
Permiso con goce de sueldo	0 horas	4 a 8 horas	4 a 8 horas
Permiso sin goce de sueldo	0 horas	4 a 8 horas	4 a 8 horas
Estacionamiento	0 horas	30 minutos a 2 horas	40 minutos a 2 horas

Tabla 19. Tiempos con la aplicación con carga de trabajo gerencial del 60% al 50%

## 8. Conclusiones

La conexión a la base de datos fue probada de manera intranet en un *Web Service* local utilizando la *IP*<sup>42</sup> privada, lo que garantizó que la consulta de datos fuera exitosa, de igual manera se logró para una conexión a internet con la *IP* pública; lo que significa que la base de datos puede montarse en “la nube”<sup>43</sup> de algún proveedor como *Google Cloud* [43], *Microsoft Azure* [44], *Amazon Web Service* [45] entre otros.

El desarrollo de las interfaces gráficas fueron de lo más intuitivo para un mejor manejo de la aplicación. Desde el menú principal que indica la acción de cada botón, los encabezados de cada pantalla para que el usuario este enterado del proceso en el que se encuentra, también el manejo de mensajes de error en el caso de que un dato no fuese correcto o incompleto, la portabilidad de la firma electrónica avanzada para generar y guardar si el usuario no cuenta con ella de manera física y la implementación de un segundo filtro de seguridad para autorizar los trámites hace segura la autenticación del gerente departamental en su propia sesión.

Con la utilización de la aplicación resultó que los trabajadores se pudieron informar de todas las actividades relevantes como cursos, capacitaciones, avisos del torneo de futbol, horarios de zumba, visitas a fundaciones de niños huérfanos, cumpleaños, *SALT* y vacantes, así como también se simplificaron los tiempos de espera a la respuesta de cualquier trámite solicitado desde la aplicación en comparación de la forma tradicional, es decir, con los tiempos tomados de las tablas 14,15,16,17,18 y 19 del capítulo 7, se observa que el uso de la aplicación reduce los tiempos en una tercera parte a comparación de no usarla. Considerando que la aplicación abarca una tercera parte de los trámites internos de la empresa esto resulta muy significativo para incluir todos los trámites que la empresa utiliza. Si fuera el caso que se tuvieran todos los trámites en la aplicación desarrollada los tiempos estimados de repuesta serían de no máximo 15 min por cada uno; y viéndolo desde el punto de vista ambiental, el consumo de papel sería muy por debajo del 25% al empleado actualmente, esta acción impactaría en los gastos departamentales, así como los de la empresa en general.

Desde el punto de vista de los gerentes, la aplicación les permitió estar más organizados con la papelería sin la angustia de que algún trámite se fuera a traspapelar, perder, ensuciar o romper. Otro punto de ayuda es la planificación del personal en consideración de vacaciones y permisos, que tales solicitudes afectan la logística de la operación en la empresa, este proceso es de suma consideración para autorizar algún trámite antes mencionado, tampoco es necesario que se involucren a terceros,

---

<sup>42</sup> Protocolo de Internet (*Internet Protocol*).

<sup>43</sup> Computación en la nube (*Cloud Computing*)

como es el caso del proceso de validación de datos por el departamento de recursos humanos manualmente.

El tiempo de respuesta que se tomaron con la papelería tradicional para las vacaciones, permiso con goce de sueldo y permiso sin goce de sueldo fue a lo máximo de sesenta horas y mínimo de veintiocho horas dependiendo de la carga de trabajo que tenga cada departamento; en el departamento de recursos humanos suele resolver en un periodo de veinticuatro horas por ser la parte donde se concentran todas las solicitudes para la validación de datos laborales. Es importante mencionar que existen factores como la validación de datos que realiza recursos humanos y la carga de trabajo que el gerente departamental tiene, considerando que tenga que verificar ocupación en habitaciones, centro de convenciones ocupado a su totalidad, vacaciones, incapacidades, cursos y capacitaciones dentro de su equipo de trabajadores.

Con la aplicación el tiempo máximo para las vacaciones, permiso con goce de sueldo y permiso sin goce de sueldo fue de veinticuatro horas como máximo y cuatro horas como mínimo para el tiempo de respuesta, ya que no se considera al departamento de recursos humanos como intermediario, sino que la misma aplicación realiza la consulta de forma rápida y eficiente; lo único que el gerente departamental está obligado a consultar los factores involucrados.

En el caso del trámite para estacionamiento los tiempos fueron más reducidos, ya que solo se toma en cuenta la ocupación que se tenga en el centro de convenciones, es por eso que esta solicitud tiene como máximo un tiempo de respuesta de treinta y seis horas a veinticinco horas como mínimo sin el uso de la aplicación; en el caso de usarla el trámite de estacionamiento obtuvo un tiempo de respuesta de veinticuatro horas como máximo a un mínimo de 40 minutos.

La eficacia de la aplicación es de suma importancia para una respuesta rápida a la solicitud de cada trámite internos de la empresa, los trabajadores conocen de toda información de último momento, la economía de la empresa crece ya que el consumo de papel es reducido contribuyendo con el medio ambiente.

## **8.1 Seguimiento a futuro**

En cuanto a lo que se relaciona al proyecto y los objetivos logrados, queda a diseñar e implementar los siguientes módulos:

- Un módulo que muestre el horario de un trabajador, así como su descanso; tomando en cuenta que trabajé por turnos como mañana, tarde y noche, o trabaje por horas.
- Un módulo que muestre el menú del comedor.
- Un módulo para solicitar la requisición departamental



- Un módulo específico para los informes del sindicato
- Un módulo de chat para soporte técnico interno
- Un módulo para enfermería
- Un módulo de consulta de huéspedes
- Un módulo para el menú del bar
- Un módulo para agendar viajes

## 9. Bibliografía

- [1] “UAM Azcapotzalco”, *Azc.uam.mx*, 2018. [En línea]. Disponible: <https://www.azc.uam.mx/>. [Último acceso: 1- Enero- 2018].
- [2] S. CDMX, “Secretaría de Finanzas de la CDMX”, *Secretaría de Finanzas de la CDMX*, 2018. [En línea]. Disponible: <http://www.finanzas.cdmx.gob.mx/>. [Último acceso: 1- Enero- 2018].
- [3] “Comisión Nacional del Agua | Gobierno | gov.mx”, *Gob.mx*, 2018. [En línea]. Disponible: <https://www.gob.mx/conagua>. [Último acceso: 1- Enero- 2018].
- [4] “Servicio de Administración Tributaria”, *Sat.gob.mx*, 2018. [En línea]. Disponible: <http://www.sat.gob.mx/Paginas/Inicio.aspx>. [Último acceso: 1- Enero- 2018].
- [5] “Inicio”, *Portal.infonavit.org.mx*, 2018. [En línea]. Disponible: <http://portal.infonavit.org.mx/wps/wcm/connect/infonavit/inicio>. [Último acceso: 1- Enero- 2018].
- [6] “Sitio Web del IMSS”, *Imss.gob.mx*, 2018. [En línea]. Disponible: <http://www.imss.gob.mx/>. [Último acceso: 1- Enero- 2018].
- [7] “Teléfono e Internet para tu Hogar, Negocio o Empresa.”, *Telmex – Macronet*, 2018. [En línea]. Disponible: <http://telmex.com/>. [Último acceso: 1- Enero- 2018].
- [8] “Telcel es la Red”, *Telcel.com*, 2018. [En línea]. Disponible: <https://www.telcel.com/>. [Último acceso: 1- Enero- 2018].
- [9] “Izzi - Inicio”, *Izzi.mx*, 2018. [En línea]. Disponible: <https://www.izzi.mx/home>. [Último acceso: 1- Enero- 2018].
- [10] “Bienvenidos a la Banca en Línea | BBVA Bancomer”, *Bancomer.com*, 2018. [En línea]. Disponible: <https://www.bancomer.com/>. [Último acceso: 1- Enero- 2018].
- [11] “Citibanamex | El Banco Nacional de México | Citibanamex.com”, *Banamex.com*, 2018. [En línea]. Disponible: <https://www.banamex.com/>. [Último acceso: 1- Enero- 2018].
- [12] “Banco Santander”, *Santander.com.mx*, 2018. [En línea]. Disponible: <http://www.santander.com.mx/mx/home/>. [Último acceso: 1- Enero- 2018].
- [13] "UAM. Universidad Autónoma Metropolitana. App UAM", *Uam.mx*, 2018. [En línea]. Disponible: <http://www.uam.mx/appuam/>. [Último acceso: 1- Enero- 2018].
- [14] S. CDMX, *Secretaría de Finanzas de la CDMX*, 2018. [En línea]. Disponible: <http://www.finanzas.cdmx.gob.mx/tesoreria-app>. [Último acceso: 1- Enero- 2018].
- [15] *Play.google.com*, 2018. [En línea]. Disponible: [https://play.google.com/store/apps/details?id=com.smn.android.weather&hl=es\\_419](https://play.google.com/store/apps/details?id=com.smn.android.weather&hl=es_419). [Último acceso: 1- Enero- 2018].
- [16] *Play.google.com*, 2018. [En línea]. Disponible: [https://play.google.com/store/apps/details?id=com.sat&hl=es\\_419](https://play.google.com/store/apps/details?id=com.sat&hl=es_419). [Último acceso: 1- Enero- 2018].
- [17] "Dispositivos móviles", *Portal.infonavit.org.mx*, 2018. [En línea]. Disponible: [http://portal.infonavit.org.mx/wps/wcm/connect/infonavit/contactanos/canales\\_de\\_servicio/dispositivos\\_moviles](http://portal.infonavit.org.mx/wps/wcm/connect/infonavit/contactanos/canales_de_servicio/dispositivos_moviles). [Último acceso: 1- Enero- 2018].

- [18] "Aplicaciones", *Imss.gob.mx*, 2018. [En línea]. Disponible: <http://www.imss.gob.mx/apps>. [Último acceso: 1- Enero- 2018].
- [19] "Administra tus servicios con app Telmex, para iOS y Android", *Telmex.com*, 2018. [En línea]. Disponible: <http://telmex.com/web/hogar/apptelmex>. [Último acceso: 1- Enero- 2018].
- [20] "MiTelcel", *Mitelcel.com*, 2018. [En línea]. Disponible: <https://www.mitelcel.com/mitelcel/login>. [Último acceso: 1- Enero- 2018].
- [21] K. Vélez, "izzi-Conoce nuestras apps", *Izzi.mx*, 2018. [En línea]. Disponible: <https://www.izzi.mx/apps/>. [Último acceso: 1- Enero- 2018].
- [22] "Bancomer Móvil | BBVA Bancomer", *Bancomer.com*, 2018. [En línea]. Disponible: <https://www.bancomer.com/personas/apps/bancomer-movil.html#Token>. [Último acceso: 1- Enero- 2018].
- [23] "Citibanamex Móvil | Citibanamex.com", *Banamex.com*, 2018. [En línea]. Disponible: <https://www.banamex.com/es/personas/banca-digital/citibanamex-movil.html>. [Último acceso: 1- Enero- 2018].
- [24] "Apps para móviles Android o iPhone - Banco Santander", *Bancosantander.es*, 2018. [En línea]. Disponible: <https://www.bancosantander.es/es/particulares/banca-En línea/app>. [Último acceso: 1- Enero- 2018].
- [25] *HiltonHonors3.hilton.com*, 2018. [En línea]. Disponible: [http://hiltonhonors3.hilton.com/es\\_XM/hilton-honors-mobile-app/](http://hiltonhonors3.hilton.com/es_XM/hilton-honors-mobile-app/). [Último acceso: 1- Enero- 2018].
- [26] "2.3. Dalvik VM – Software de Comunicación", *Sites.google.com*, 2018. [En línea]. Disponible: <https://sites.google.com/site/swcuc3m/home/android/generalidades/dalvikvm-1>. [Último acceso: 1- Enero- 2018].
- [27] M. Lozano Ortega y A. Gallego Sánchez, *Desarrollo de aplicaciones Android con java*. Paracuellos del Jarama: RA-MA, 2017.
- [28] J. Amaro Soriano, *El gran libro de programación avanzada con Android*. México: Alfaomega, 2013.
- [29] J. Amaro Soriano, *Android: programación de dispositivos móviles a través de ejemplos*. México: Alfaomega Grupo Editor, 2012.
- [30] J. Nolasco Valenzuela, *Desarrollo de aplicaciones móviles con Android*. Paracuellos de Jarama, Madrid: Ra-Ma, 2015.
- [31] "El sistema operativo GNU y el movimiento del software libre", *Gnu.org*, 2018. [En línea]. Disponible: <https://www.gnu.org/home.es.html>. [Último acceso: 4- Enero- 2018].
- [32] C. Studio, "Conoce Android Studio | Android Studio", *Developer.android.com*. 2018. [En línea]. Disponible: <https://developer.android.com/index.html?hl=es-419>. [Último acceso: 4- Enero- 2018].
- [33] *Columna80.files.wordpress.com*. 2018. [En línea]. Disponible: <https://columna80.files.wordpress.com/2011/02/0013-01-pila-software-android.png>. [Último acceso: 11- Enero- 2018].

- [34] “Bases de datos SQL | AWS”, *Amazon Web Services, Inc.*, 2018. [En línea]. Disponible: <https://aws.amazon.com/es/relational-database/>. [Último acceso: 11- Enero- 2018].
- [35] “The Central Repository Search Engine”, *Search.maven.org*, 2018. [En línea]. Disponible: <https://search.maven.org/>. [Último acceso: 15- Enero- 2018].
- [36] *Jcenter.bintray.com*, 2018. [En línea]. Disponible: <http://jcenter.bintray.com/>. [Último acceso: 15- Enero- 2018].
- [37] “Desarrollo Android”, *Pinterest*. 2018. [En línea]. Disponible: <https://www.pinterest.es/pin/535154368195185330/>. [Último acceso: 16- Enero- 2018].
- [38] “JSON”, *Json.org*, 2018. [En línea], Disponible: <https://json.org/json-es.html>. [Último acceso: 16- Enero- 2018].
- [39] “Un Tipo Digital – Diseñador visual de interfaces digitales >> Diseño de móviles & tablets: las diferentes densidades”, *Untipodigital.com*, 2018. [En línea]. Disponible: <http://www.untipodigital.com/disen-de-moviles-tablets-las-diferentes-densidades/>. [Último acceso: 17- Enero- 2018].
- [40] P. control, “Paneles de control | Android Developers”, *Developer.android.com*, 2018. [En línea]. Available: <https://developer.android.com/about/dashboards/index.html?hl=es-419>. [Último acceso: 19- Enero- 2018].
- [41] “Stack Overflow – Where Developers Learn, Share, & Build Careers”, *Stackoverflow.com*, 2018. [En línea]. Disponible: <https://stackoverflow.com>. [Último acceso]: 20- Enero- 2018].
- [42] “Desarrollo Android”, *CódigoFacilito*, 2018. [En línea]. Disponible: <https://codigofacilito.com/categories/desarrollador-android>. [Último acceso: 30- Enero- 2018].
- [43] “Cloud Computing, servicios de alojamiento y APIs de Google Cloud | Google Cloud”, *Google Cloud*, 2018. [En línea]. Disponible: <https://cloud.google.com/?hl=es>. [Último acceso: 2- Febrero- 2018].
- [44] “Microsoft Azure: plataforma y servicios de informática en la nube”, *Azure.microsoft.com*, 2018. [En línea]. Disponible: <https://azure.microsoft.com/es-mx/>. [Último acceso: 3- Febrero- 2018].
- [45] “AWS | Cloud Computing – Servicios de informática en la nube”, *Amazon Web Services, Inc.*, 2018. [En línea]. Disponible: <https://aws.amazon.com/es/>. [Último acceso: 4- Febrero- 2018].

## 10. Apéndices

En este capítulo se encuentran todas las clases del proyecto, así como sus respectivas funciones y métodos que se usaron.

### 10.1 AbrirGaleria.class

Permite abrir un explorador de archivos para seleccionar la firma electrónica avanzada en Código QR con extensión PNG.

```
package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.AsyncTask;
import android.provider.MediaStore;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.google.zxing.BinaryBitmap;
import com.google.zxing.ChecksumException;
import com.google.zxing.FormatException;
import com.google.zxing.LuminanceSource;
import com.google.zxing.MultiFormatReader;
import com.google.zxing.NotFoundException;
import com.google.zxing.RGBLuminanceSource;
import com.google.zxing.Reader;
import com.google.zxing.Result;
import com.google.zxing.common.HybridBinarizer;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class AbrirGaleria extends AppCompatActivity {

    private static final int PHOTO_SELECTED=2;
    Uri imageUri;
    ImageView foto_gale;
    TextView resdo;
    String text;
    public String contents;

    final Context contx=this;
    public static final String FOVLIA = "FOLIAS";
    public static final String NMVJT = "USERMNGT";
    public static final String AUXBAN = "AUXIBAN";
```

```

String dirip= "http://192.168.1.72/iniciarphp/";
String getuno = dirip + "panel/guadaraceptarvaca.php";
String getdos = dirip + "panel/guadaraceptarpsing.php";
String gettres = dirip + "panel/guadaraceptarpcong.php";
String getcuatro = dirip + "panel/guadaraceptarestaciona.php";
private ProgressDialog msjito;
public String gtnusr;
public String folkij;
public String qrpas;
public String clpass;
public String idbandera;
public String dirconsultd;

EnchuxeWebServt hiloexect;

AlertDialog.Builder aletrbuilder;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_abrir_imagen);
    foto_gale=(ImageView) findViewById(R.id.imgabririmagen);
    resdo = (TextView) findViewById(R.id.respucodigoqrgaleria);

    Intent intent = getIntent();
    gtnusr = intent.getStringExtra(FragmentAceptar.USRMG);
    clpass = intent.getStringExtra(FragmentAceptar.CVGT);
    qrpas = intent.getStringExtra(FragmentAceptar.PSSGTQR);
    folkij = intent.getStringExtra(FragmentAceptar.FXIO);
    idbandera = intent.getStringExtra(FragmentAceptar.FLAGMIX);
    opengaleria();
    openvalidacion();
}

//Método para validar la contraseña del Gerente para validar los tramites.
private void openvalidacion() {
    LayoutInflater li=LayoutInflater.from(contx);
    View promptview=li.inflate(R.layout.menspassqr, null);
    aletrbuilder = new AlertDialog.Builder(contx);
    aletrbuilder.setView(promptview);
    final EditText userInput=(EditText)
promptview.findViewById(R.id.edtxmespaasinput);
    aletrbuilder.setCancelable(false).setPositiveButton("OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {

            if(userinput.getText().toString().equals(clpass)){

Toast.makeText(getApplicationContext(),"Exito",Toast.LENGTH_LONG).show();
                hiloexect = new EnchuxeWebServt();
                if(idbandera.equals("vacaciones")) {
                    dirconsultd = getuno + "?getn=" + gtnusr + "&fofz=" +
folkij;
                }
                else if(idbandera.equals("percongo")){
                    dirconsultd = gettres + "?getn=" + gtnusr + "&fofz=" +
folkij;
                }
                else if(idbandera.equals("persingo")){
                    dirconsultd = getdos + "?getn=" + gtnusr + "&fofz=" +
folkij;
                }
            }else{

```

```

        dirconsultd = getcuatro + "?getn=" + gtnusr + "&fofz=" +
folki;
    }
    hiloexec.execute(dirconsultd);
    Intent regresar = new
Intent(getApplicationContext(), ModuloAutorizar.class);
    regresar.putExtra(NMVJT, gtnusr);
    startActivity(regresar);
    finish();

}
else{

Toast.makeText(getApplicationContext(), "Error", Toast.LENGTH_LONG).show();
    Intent regresar = new
Intent(getApplicationContext(), ContenedorFragmentMg.class);
    regresar.putExtra(FOVLIA, folki);
    regresar.putExtra(NMVJT, gtnusr);
    regresar.putExtra(AUXBAN, idbandera);
    startActivity(regresar);
    finish();

}

}
}).setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Intent regresar = new
Intent(getApplicationContext(), ContenedorFragmentMg.class);
        regresar.putExtra(FOVLIA, folki);
        regresar.putExtra(NMVJT, gtnusr);
        regresar.putExtra(AUXBAN, idbandera);
        startActivity(regresar);
        finish();

    }
});
}
//Método que abre un explorador de archivos
private void opengaleria() {
    Intent galeri = new Intent(Intent.ACTION_GET_CONTENT,
MediaStore.Images.Media.INTERNAL_CONTENT_URI);
    galeri.setType("image/*");
    startActivityForResult(Intent.createChooser(galeri, "Selecciona Imagen"),
PHOTO_SELECTED);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data){
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode==RESULT_OK && requestCode==PHOTO_SELECTED){
        imageUri=data.getData();

        try {
            Bitmap
bMap=MediaStore.Images.Media.getBitmap(this.getContentResolver(), imageUri);

            int[] intArray = new int[bMap.getWidth()*bMap.getHeight()];
            //copy pixel data from the Bitmap into the 'intArray' array
            bMap.getPixels(intArray, 0, bMap.getWidth(), 0, 0,
bMap.getWidth(), bMap.getHeight());

            LuminanceSource source = new RGBLuminanceSource(bMap.getWidth(),

```

```

bMap.getHeight(), intArray);
        BinaryBitmap bitmap = new BinaryBitmap(new
HybridBinarizer(source));
        Reader reader = new MultiFormatReader();
        Result result = reader.decode(bitmap);
        contents = result.getText();
        foto_gale.setImageBitmap(bMap);
        resdo.setText(contents);
        if(contents.equals(qrpas)){
            Toast.makeText(getApplicationContext(), "Firma QR
valida", Toast.LENGTH_LONG).show();
            AlertDialog alerdialog = aletrbuilder.create();
            alerdialog.show();
        }else{
            Toast.makeText(getApplicationContext(), "Firma QR no
valida", Toast.LENGTH_LONG).show();
            Intent regresar = new
Intent(getApplicationContext(), ContenedorFragmentMg.class);
            regresar.putExtra(FOVLIA, folki);
            regresar.putExtra(NMVJT, gtnusr);
            regresar.putExtra(AUXBAN, idbandera);
            startActivity(regresar);
            finish();
        }
    }
} catch (IOException e) {
    e.printStackTrace();
} catch (FormatException e) {
    e.printStackTrace();
} catch (ChecksumException e) {
    e.printStackTrace();
} catch (NotFoundException e) {
    e.printStackTrace();
}
}
}

private String getRealPath(Uri imageUri) {
    String path=null;
    String[] filepath = {MediaStore.MediaColumns.DATA};
    Cursor cursor=
getContentResolver().query(imageUri, filepath, null, null, null);
    if(cursor.moveToFirst()){
        int
columnindex=cursor.getColumnIndexOrThrow(MediaStore.MediaColumns.DATA);
        path=cursor.getString(columnindex);
    }

    cursor.close();
    return path;
}

//Método para enviar los datos a la base de datos remota de Cerebro.
private class EnchuxeWebServt extends AsyncTask<String,Void,Void> {
    @Override
    protected Void doInBackground(String... params) {
        String cadena = params[0];
        URL url = null;
        String devuelve = "";

        try {
            url = new URL(cadena);
            HttpURLConnection connexion =

```



```

(HttpURLConnection)url.openConnection();
    connexion.setRequestProperty("Content-Type","text/html");
    connexion.setRequestProperty("Host","192.168.1.72");
    connexion.setRequestProperty("User-
Agent","Mozilla/5.0(X11;Ubuntu;Linux i686;rv:49.0) Gecko/20100101 Firefox/49.0");
    connexion.setRequestMethod("GET");
    int respuesta = connexion.getResponseCode();
    StringBuilder result = new StringBuilder();

    if (respuesta == HttpURLConnection.HTTP_OK){

        InputStream in = new
BufferedInputStream(connexion.getInputStream());

        BufferedReader reader = new BufferedReader(new
InputStreamReader(in));

        String line=null;
        while ((line = reader.readLine()) != null) {
            result.append(line);
            break;
        }

        reader.close();

    }
} catch (IOException e) {
    e.printStackTrace();
}
return null;
}

@Override
protected void onPreExecute() {
    super.onPreExecute();
    msjito = new ProgressDialog(AbrirGaleria.this);
    msjito.setMessage("Sending data...");
    msjito.show();
}
}
}
}

```

## 10.2 ActividadMensual.class

Muestra todos las actividades que se llevarán a cabo durante todo un mes.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.graphics.Rect;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;

```

```

import android.widget.ListView;
import android.widget.TextView;
import com.github.snowdream.android.widget.SmartImageView;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;
import org.json.JSONArray;
import org.json.JSONException;
import java.util.ArrayList;
import cz.msebera.android.httpclient.Header;

/**
 * Created by Roberto Espejel Ramirez 207205750 on 26/01/2017.
 */

public class ActividadMesual extends AppCompatActivity {

    ListView lv;

    ArrayList title = new ArrayList();
    ArrayList descrip = new ArrayList();
    ArrayList foto = new ArrayList();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.info_calendario);

        lv=(ListView) findViewById(R.id.listainfo);
        descargarImagen();
        lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
                Intent intent = new
Intent(ActividadMesual.this,DetalleEvento.class);
                intent.putExtra("contenido",title.get(position).toString());
                intent.putExtra("pixis",foto.get(position).toString());
                intent.putExtra("detalle",descrip.get(position).toString());
                startActivity(intent);
            }
        });
    }

    //Método que solicita la conexión a Cerebro para descargar todos las actividades.
    private void descargarImagen(){
        title.clear();
        descrip.clear();
        foto.clear();

        final ProgressDialog progressDialog = new
ProgressDialog(ActividadMesual.this);
        progressDialog.setMessage("Cargando Eventos...");
        progressDialog.show();

        AsyncHttpClient client = new AsyncHttpClient();
        client.get("http://192.168.1.72/iniciarpHP/panel/codigojson.php", new
AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[] headers, byte[]
responseBody) {

```

```

        if(statusCode == 200){
            progressDialog.dismiss();
            try {
                JSONArray jasonarray = new JSONArray(new
String(responseBody));
                for (int i=0;i<jasonarray.length();i++){

title.add(jasonarray.getJSONObject(i).getString("fecha"));

descrip.add(jasonarray.getJSONObject(i).getString("descripcion"));

foto.add(jasonarray.getJSONObject(i).getString("archivo"));
                }
                lv.setAdapter(new ListViewAdapter(getApplicationContext()));
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }

    @Override
    public void onFailure(int statusCode, Header[] headers, byte[]
responseBody, Throwable error) {

    }
}

//Método que despliega la información de todas las actividades en forma de lista.
private class ListViewAdapter extends BaseAdapter {
    Context context;
    LayoutInflater inflater;
    SmartImageView smartImageView;

    public ListViewAdapter(Context applicationContext) {
        this.context=applicationContext;

inflater=(LayoutInflater) context.getSystemService(LAYOUT_INFLATER_SERVICE);

    }

    @Override
    public int getCount() { return foto.size();}

    @Override
    public Object getItem(int position) {
        return position;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    public View getView(int position, View convertView, ViewGroup parent) {

        TextView txtTitulo;
        TextView txtSubtitulo;

        ViewGroup viewgroup=
(ViewGroup) inflater.inflate(R.layout.activity_mcalendario,null);

```

```

        smartImageView=(SmartImageView)
viewgroup.findViewById(R.id.imv_list);
        txtTitulo = (TextView)
viewgroup.findViewById(R.id.title_list_calendario);
        txtSubtitulo = (TextView)
viewgroup.findViewById(R.id.subti_list_calendario);
        String
urlfinal="http://192.168.1.72/iniciarphp/fotos/"+foto.get(position).toString();
        Rect rect = new
Rect(smartImageView.getLeft(), smartImageView.getTop(), smartImageView.getRight(), s
martImageView.getBottom());

        smartImageView.setImageUrl(urlfinal,rect);
        txtTitulo.setText(title.get(position).toString());
        txtSubtitulo.setText(descrip.get(position).toString());

        return viewgroup;
    }
}
}

```

### 10.3 ContenedorFragmentMg.class

Este fragment divide la pantalla en 2, por la parte superior están los botones de aceptar y denegar, y por la parte inferior muestra las opciones correspondientes de cada botón.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.content.Intent;
import android.net.Uri;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.FrameLayout;
import android.widget.LinearLayout;
import android.widget.Toast;

public class ContenedorFragmentMg extends AppCompatActivity implements
FragmentDenegar.OnFragmentInteractionListener,FragmentAceptar.OnFragmentInteracti
onListener{

    Button acept;
    Button canl;
    FrameLayout flayou;
    public static final String FOLIA = "FOLIAS";
    public static final String NMJT = "USERMNGT";
    public static final String BANDMIX = "FLAGMIX";
    public String usrgt;
    public String frgis;
    public String bansenal;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detalle_correodep);
        acept = (Button) findViewById(R.id.btndcdautorizar);
        canl = (Button) findViewById(R.id.btndcpdenegar);
    }
}

```

```

flayou = (FrameLayout) findViewById(R.id.fralayoutcon);

Intent intent = getIntent();
usrgt = intent.getStringExtra(CorreoVacacionesMg.NMGT);
frgis = intent.getStringExtra(CorreoVacacionesMg.FOLRGZ);
bansenal=intent.getStringExtra(CorreoVacacionesMg.BANDERA);
if(intent.getStringExtra(AbrirGaleria.NMVJT)!=null){
    usrgt = intent.getStringExtra(AbrirGaleria.NMVJT);
    frgis = intent.getStringExtra(AbrirGaleria.FOVLIA);
    bansenal = intent.getStringExtra(AbrirGaleria.AUXBAN);
}
if(intent.getStringExtra(ScannerFinal.NMVJT)!=null){
    usrgt=intent.getStringExtra(ScannerFinal.NMVJT);
    frgis=intent.getStringExtra(ScannerFinal.FOVLIA);

}
else if(intent.getStringExtra(CorreoPermisingMg.BANDERA)!=null){
    usrgt=intent.getStringExtra(CorreoPermisingMg.NMGT);
    frgis=intent.getStringExtra(CorreoPermisingMg.FOLRGZ);
    bansenal=intent.getStringExtra(CorreoPermisingMg.BANDERA);
}
else if(intent.getStringExtra(CorreoPermicongMg.BANDERA)!=null){
    usrgt=intent.getStringExtra(CorreoPermicongMg.NMGT);
    frgis=intent.getStringExtra(CorreoPermicongMg.FOLRGZ);
    bansenal=intent.getStringExtra(CorreoPermicongMg.BANDERA);
}
else if(intent.getStringExtra(CorreoEstacionamientoMg.BANDERA)!=null){
    usrgt=intent.getStringExtra(CorreoEstacionamientoMg.NMGT);
    frgis=intent.getStringExtra(CorreoEstacionamientoMg.FOLRGZ);
    bansenal=intent.getStringExtra(CorreoEstacionamientoMg.BANDERA);
}
}
//Método que da función al botón cancelar para denegar un tramite
canl.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Bundle bund = new Bundle();
        bund.putString(FOLIA,frgis);
        bund.putString(NMJT,usrgt);
        bund.putString(BANDMIX,bansenal);

        FragmentDenegar fraA = new FragmentDenegar();
        fraA.setArguments(bund);

        getSupportFragmentManager().beginTransaction().replace(R.id.fralayoutcon, fraA).disallowAddToBackStack().commit();
        Toast.makeText(getApplicationContext(), "Desea agregar un comentario?", Toast.LENGTH_LONG).show();
    }
});

//Método para el botón aceptar el cual podrá aprobar los tramites.
accept.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Bundle bunddos = new Bundle();
        bunddos.putString(FOLIA,frgis);
        bunddos.putString(NMJT,usrgt);
        bunddos.putString(BANDMIX,bansenal);

        FragmentAceptar fraB = new FragmentAceptar();
        fraB.setArguments(bunddos);

        getSupportFragmentManager().beginTransaction().replace(R.id.fralayoutcon, fraB).addToBackStack(null).commit();
        Toast.makeText(getApplicationContext(), "Firma

```

```

QR", Toast.LENGTH_LONG).show();
        }
    });

}

@Override
public void onFragmentInteraction(Uri uri) {

}
}

```

## 10.4 CorreoEstacionamientoMg.class

Despliega en forma de lista todas las solicitudes de estacionamiento relacionadas con el Gerente de departamento.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Handler;
import android.support.v4.widget.SwipeRefreshLayout;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ImageButton;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.HashMap;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.ClientProtocolException;
import cz.msebera.android.httpclient.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.params.BasicHttpParams;

public class CorreoEstacionamientoMg extends AppCompatActivity {

    String username;
    String myJSON;
    ArrayList firmnger = new ArrayList();
    ArrayList folio = new ArrayList();

```

```

public static final String NMGT = "NMMGR";
public static final String FOLRGZ = "FOLIOREGZ";
public static final String BANDERA = "FLAGCPS";

private static final String TAG_FOL="idestacionamiento";
private static final String TAG_RESULTS="mailestaciona";
private static final String TAG_ID =
"estacionamiento.trabajador_idtrabajador";
private static final String TAG_NAME = "nombre";
private static final String TAG_PAT = "aprimero";
private static final String TAG_MAT = "asegundo";
private static final String TAG_DIA ="numdia";
private static final String TAG_FIN = "fechacajon";
private static final String TAG_FFI = "fechadesocajon";
private static final String TAG_MNG = "estacionamiento.manager_iduser";

JSONArray peoples = null;

ArrayList<HashMap<String, String>> personList;

ListView listapeticiones;
TextView idgerente;
ImageButton valid;
public static ListAdapter adapter;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_correo_estacionamiento_mg);
    getSupportActionBar().hide();

    listapeticiones=(ListView) findViewById(R.id.lstcorreogerenteestaciona);
    idgerente=(TextView) findViewById(R.id.nombestacionatxtvw);
    valid=(ImageButton) findViewById(R.id.btnsaliextestaciona);

    Intent intent = getIntent();
    username = intent.getStringExtra(ModuloAutorizar.USER_LOG);
    idgerente.setText(username);
    if(intent.getStringExtra(FragmentDenegar.FGYTG) !=null) {
        username=intent.getStringExtra(FragmentDenegar.FGYTG);
        idgerente.setText(username);
    }else if(intent.getStringExtra(ScannerFinal.NMVJT) !=null) {
        username=intent.getStringExtra(ScannerFinal.NMVJT);
        idgerente.setText(username);
    }else if(intent.getStringExtra(AbrirGaleria.NMVJT) !=null) {
        username=intent.getStringExtra(AbrirGaleria.NMVJT);
        idgerente.setText(username);
    }

    personList = new ArrayList<HashMap<String, String>>();
    iniciomail();

    listapeticiones.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
            if(firmnger.get(position).toString().equals("vacio")){
                Intent intent = new
Intent(CorreoEstacionamientoMg.this, ContenedorFragmentMg.class);
                intent.putExtra(NMGT, username);
            }
        }
    });
}

```

```

        intent.putExtra(FOLRGZ, folio.get(position).toString());
        intent.putExtra(BANDERA, "estacionamiento");
        startActivity(intent);
    }
    else{
        Toast.makeText(getApplicationContext(), "Ya fue aprobado",
Toast.LENGTH_SHORT).show();
    }

    });

    final SwipeRefreshLayout swiprefre = (SwipeRefreshLayout)
findViewById(R.id.refrehlist);

    swiprefre.setColorSchemeResources(android.R.color.holo_blue_bright,
android.R.color.holo_green_light,
android.R.color.holo_orange_light,
android.R.color.holo_red_light);

    swiprefre.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener()
{
    @Override
    public void onRefresh() {
        swiprefre.setRefreshing(true);
        new Handler().postDelayed(new Runnable() {
            @Override public void run() {
                username=idgerente.getText().toString();
                swiprefre.setRefreshing(false);
            }
        }, 5000);
    }
});

    btsalida();
}

//Método para descargar los datos en relación a los tramites de estacionamiento
relacionado con el Gerente de departamento.
    private void iniciomail() {
        class obtpeticion extends AsyncTask<String, Void, String > {
            private Dialog windDialog;

            @Override
            protected void onPreExecute () {
                super.onPreExecute();
                windDialog = ProgressDialog.show(CorreoEstacionamientoMg.this,
"Please wait", "Loading...");
            }

            @Override
            protected String doInBackground (String...params){
                DefaultHttpClient httpclient = new DefaultHttpClient(new
BasicHttpParams());
                HttpPost httppost = new
HttpPost("http://192.168.1.72/iniciarpHP/panel/codjsonestaciondepartamental.php?i
dger="+username);

                httppost.setHeader("Content-type", "application/json");

                InputStream inputStream = null;

```



```

String result = null;
try {
    HttpResponse response = httpClient.execute(httpPost);
    HttpEntity entity = response.getEntity();

    inputStream = entity.getContent();
    BufferedReader reader = new BufferedReader(new
InputStreamReader(inputStream, "UTF-8"), 8);
    StringBuilder sb = new StringBuilder();

    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n" );
    }
    result = sb.toString();
    reader.close();//--

} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
} catch (ClientProtocolException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} finally {
    try {
        if (inputStream != null) inputStream.close();
    } catch (Exception squish) {
    }
}
return result;
}
protected void onPostExecute(String result){
    windDialog.dismiss();
    myJSON=result;
    despleList();
}
obtpeticion op = new obtpeticion();
op.execute();

}

//Método que muestra en forma de lista todas las solicitudes de estacionamiento
private void despleList(){
    firmnger.clear();
    folio.clear();

    try {
        JSONObject jsonObj = new JSONObject(myJSON);
        peoples = jsonObj.getJSONArray(TAG_RESULTS);

        for(int i=0;i<peoples.length();i++){
            JSONObject c = peoples.getJSONObject(i);
            folio.add(c.getString(TAG_FOL));
            String id = c.getString(TAG_ID);
            String name = c.getString(TAG_NAME);
            String apu = c.getString(TAG_PAT);
            String aps = c.getString(TAG_MAT);
            String dizp = c.getString(TAG_DIA);
            String fstar = c.getString(TAG_FIN);
            String fend = c.getString(TAG_FFI);
            String mgfre = c.getString(TAG_MNG);
            firmnger.add(c.getString(TAG_MNG));
        }
    }
}

```

```

        HashMap<String,String> persons = new HashMap<String,String> ();

        persons.put (TAG_NAME,name);
        persons.put (TAG_PAT,apu);
        persons.put (TAG_MAT,aps);
        persons.put (TAG_DIA,dizp);
        persons.put (TAG_FIN,fstar);
        persons.put (TAG_FFI,fend);
        persons.put (TAG_MNG,mgfre);

        personList.add(persons);
    }

    adapter = new SimpleAdapter (CorreoEstacionamientoMg.this, personList,
R.layout.info_maildepartvaca,
        new
String[] {TAG_NAME, TAG_PAT, TAG_MAT, TAG_DIA, TAG_FIN, TAG_FFI, TAG_MNG},
        new int[] {R.id.txvwlisemailname, R.id.txvwlisemailapp,
R.id.txvwlisemailaps, R.id.txemaildepdias, R.id.txemaildepfinicio, R.id.txemaildepff
in, R.id.txemailmngde}
    );

    listapeticiones.setAdapter (adapter);

} catch (JSONException e) {
    e.printStackTrace();
}
}

//Método para la función del botón salida la cual finalizara la sesion del
Gerente.
private void btsalida() {
    valid.setOnClickListener (new View.OnClickListener() {
        @Override
        public void onClick (View v) {

            Intent intexir = new
Intent (CorreoEstacionamientoMg.this, MainActivity.class);
            startActivity (intexir);
            finish();
        }
    });
}
}
}

```

## 10.5 CorreoPermicongMg.class

Despliega en forma de lista todas las solicitudes de permiso con goce relacionadas con el Gerente de departamento.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Handler;
import android.support.v4.widget.SwipeRefreshLayout;
import android.support.v7.app.AppCompatActivity;

```

```

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ImageButton;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.HashMap;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.ClientProtocolException;
import cz.msebera.android.httpclient.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.params.BasicHttpParams;

public class CorreoPermiconMg extends AppCompatActivity {

    String username;
    String myJSON;
    ArrayList firmnger = new ArrayList();
    ArrayList folio = new ArrayList();

    public static final String NMGT = "NMMGR";
    public static final String FOLRGZ = "FOLIOREGZ";
    public static final String BANDERA = "FLAGCPS";

    private static final String TAG_FOL="idpermisocon";
    private static final String TAG_RESULTS="maildepcon";
    private static final String TAG_ID = "permisocon.trabajador_idtrabajador";
    private static final String TAG_NAME = "nombre";
    private static final String TAG_PAT = "aprimero";
    private static final String TAG_MAT = "apsegundo";
    private static final String TAG_DIA ="permisocon.dizxusar";
    private static final String TAG_FIN = "fechacomip";
    private static final String TAG_FFI = "fechafinp";
    private static final String TAG_MNG = "permisocon.manager_iduser";
    private static final String TAG_TXMT = "timexetime";

    JSONArray peoples = null;

    ArrayList<HashMap<String, String>> personList;

    ListView listapeticiones;
    TextView idgerente;
    ImageButton solid;
    public static ListAdapter adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_correo_permicong_mg);
getSupportActionBar().hide();

listapeticiones=(ListView) findViewById(R.id.lstcorreogerentepong);
idgerente=(TextView) findViewById(R.id.nombpcontxtvw);
salid=(ImageButton) findViewById(R.id.btnsaliextpong);

Intent intent = getIntent();
username = intent.getStringExtra(ModuloAutorizar.USER_LOG);
idgerente.setText(username);
if(intent.getStringExtra(FragmentDenegar.FGYTG) !=null) {
    username=intent.getStringExtra(FragmentDenegar.FGYTG);
    idgerente.setText(username);
} else if(intent.getStringExtra(ScannerFinal.NMVJT) !=null) {
    username=intent.getStringExtra(ScannerFinal.NMVJT);
    idgerente.setText(username);
} else if(intent.getStringExtra(AbrirGaleria.NMVJT) !=null) {
    username=intent.getStringExtra(AbrirGaleria.NMVJT);
    idgerente.setText(username);
}

personList = new ArrayList<HashMap<String,String>>();
iniciomail();

listapeticiones.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
        if(firmnger.get(position).toString().equals("vacio")){
            Intent intent = new
Intent(CorreoPermicongMg.this, ContenedorFragmentMg.class);
            intent.putExtra(NMGT, username);
            intent.putExtra(FOLRGZ, folio.get(position).toString());
            intent.putExtra(BANDERA, "percongo");
            startActivity(intent);
        }
        else{
            Toast.makeText(getApplicationContext(), "Ya fue aprobado",
Toast.LENGTH_SHORT).show();
        }
    }
});

final SwipeRefreshLayout swiprefre = (SwipeRefreshLayout)
findViewById(R.id.refrehlist);

swiprefre.setColorSchemeResources(android.R.color.holo_blue_bright,
android.R.color.holo_green_light,
android.R.color.holo_orange_light,
android.R.color.holo_red_light);

swiprefre.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener()
{
    @Override
    public void onRefresh() {
        swiprefre.setRefreshing(true);
        new Handler().postDelayed(new Runnable() {
            @Override public void run() {
                username=idgerente.getText().toString();
            }
        }, 2000);
    }
});

```

```

        swiprefre.setRefreshing(false);
    }
    }, 5000);
    });
    btsalida();
}

//Método descarga todos los datos a través de Cerebro.
private void iniciomail() {
    class obtpeticion extends AsyncTask<String, Void, String > {
        private Dialog windDialog;

        @Override
        protected void onPreExecute () {
            super.onPreExecute();
            windDialog = ProgressDialog.show(CorreoPermicongMg.this, "Please
wait", "Loading...");
        }

        @Override
        protected String doInBackground (String...params){

            DefaultHttpClient httpclient = new DefaultHttpClient(new
BasicHttpParams());
            HttpPost httppost = new
HttpPost("http://192.168.1.72/iniciarphtml/panel/codjsonpcongdepartamental.php?idge
r="+username);

            httppost.setHeader("Content-type", "application/json");

            InputStream inputStream = null;
            String result = null;
            try {
                HttpResponse response = httpclient.execute(httppost);
                HttpEntity entity = response.getEntity();

                inputStream = entity.getContent();
                BufferedReader reader = new BufferedReader(new
InputStreamReader(inputStream, "UTF-8"), 8);
                StringBuilder sb = new StringBuilder();

                String line = null;
                while ((line = reader.readLine()) != null) {
                    sb.append(line + "\n" );
                }
                result = sb.toString();
                reader.close();//--

            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            } catch (ClientProtocolException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            } finally {
                try {
                    if (inputStream != null) inputStream.close();
                } catch (Exception squish) {

```

```

        }
    }
    return result;
}
protected void onPostExecute(String result){
    windDialog.dismiss();
    myJSON=result;
    despleList();
}
}
obtpeticion op = new obtpeticion();
op.execute();
}
//Método para darle funcionalidad de terminar sesión al botón salida
private void btsalida() {
    valid.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent intexir = new
Intent(CorreoPermicongMg.this,MainActivity.class);
            startActivity(intexir);
            finish();

        }
    });
}
//Método para desplegar toda la información en forma de lista
private void despleList(){
    firmnger.clear();
    folio.clear();

    try {
        JSONObject jsonObj = new JSONObject(myJSON);
        peoples = jsonObj.getJSONArray(TAG_RESULTS);

        for(int i=0;i<peoples.length();i++){
            JSONObject c = peoples.getJSONObject(i);
            folio.add(c.getString(TAG_FOL));
            String id = c.getString(TAG_ID);
            String name = c.getString(TAG_NAME);
            String apu = c.getString(TAG_PAT);
            String aps = c.getString(TAG_MAT);
            String dizp = c.getString(TAG_DIA);
            String fstar = c.getString(TAG_FIN);
            String fend = c.getString(TAG_FFI);
            String mgfre = c.getString(TAG_MNG);
            String tiemtiem = c.getString(TAG_TXMT);
            firmnger.add(c.getString(TAG_MNG));

            HashMap<String,String> persons = new HashMap<String,String>();

            //persons.put(TAG_ID,id);
            persons.put(TAG_NAME,name);
            persons.put(TAG_PAT,apu);
            persons.put(TAG_MAT,aps);
            persons.put(TAG_DIA,dizp);
            persons.put(TAG_FIN,fstar);
            persons.put(TAG_FFI,fend);
            persons.put(TAG_MNG,mgfre);
            persons.put(TAG_TXMT,tiemtiem);

```

```

        personList.add(persons);
    }

    adapter = new SimpleAdapter(CorreoPermicongMg.this, personList,
R.layout.info_maildepartvaca,

        new
String[] {TAG_NAME, TAG PAT, TAG MAT, TAG DIA, TAG FIN, TAG FFI, TAG MNG, TAG TXMT},
        new int[] {R.id.txvwlisemailname, R.id.txvwlisemailapp,
R.id.txvwlisemailaps, R.id.txemaildepdias, R.id.txemaildepfinicio, R.id.txemaildepff
in, R.id.txemailmngde, R.id.txtvwrazon}
    );

    listapeticiones.setAdapter(adapter);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}
}

```

## 10.6 CorreoPermisngMg.class

Despliega en forma de lista todas las solicitudes de permiso sin goce relacionadas con el Gerente de departamento.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Handler;
import android.support.v4.widget.SwipeRefreshLayout;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.Button;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.HashMap;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.ClientProtocolException;

```

```

import cz.msebera.android.httpclient.client.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.params.BasicHttpParams;

public class CorreoPermisngMg extends AppCompatActivity {

    String username;
    String myJSON;
    ArrayList firmnger = new ArrayList();
    ArrayList folio = new ArrayList();

    public static final String NMGT = "NMMGR";
    public static final String FOLRGZ = "FOLIOREGZ";
    public static final String BANDERA = "FLAGCPS";

    private static final String TAG_FOL="idpermisosin";
    private static final String TAG_RESULTS="maildepsin";
    private static final String TAG_ID = "permisosin.trabajador_idtrabajador";
    private static final String TAG_NAME = "nombre";
    private static final String TAG_PAT = "aprimero";
    private static final String TAG_MAT = "asegundo";
    private static final String TAG_DIA = "dizxusar";
    private static final String TAG_FIN = "fechastarp";
    private static final String TAG_FFI = "fechasndp";
    private static final String TAG_MNG = "permisosin.manager_iduser";

    JSONArray peoples = null;

    ArrayList<HashMap<String, String>> personList;

    ListView listapeticiones;
    TextView idgerente;
    ImageButton solid;
    public static ListAdapter adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_correo_permising_mg);
        getSupportActionBar().hide();

        listapeticiones=(ListView) findViewById(R.id.lstcorreogerentepsing);
        idgerente=(TextView) findViewById(R.id.nombpsintxtvw);
        solid=(ImageButton) findViewById(R.id.btnsaliextpsing);

        Intent intent = getIntent();
        username = intent.getStringExtra(ModuloAutorizar.USER_LOG);
        idgerente.setText(username);
        if(intent.getStringExtra(FragmentDenegar.FGYTG)!=null) {
            username=intent.getStringExtra(FragmentDenegar.FGYTG);
            idgerente.setText(username);
        }else if(intent.getStringExtra(ScannerFinal.NMVJT)!=null) {
            username=intent.getStringExtra(ScannerFinal.NMVJT);
            idgerente.setText(username);
        }else if(intent.getStringExtra(AbrirGaleria.NMVJT)!=null) {
            username=intent.getStringExtra(AbrirGaleria.NMVJT);
            idgerente.setText(username);
        }
    }

    personList = new ArrayList<HashMap<String, String>>();
    iniciomail();
}

```



```

        listapeticiones.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
        if(firmnger.get(position).toString().equals("vacio")){
            Intent intent = new
Intent(CorreoPermisngMg.this, ContenedorFragmentMg.class);
            intent.putExtra(NMGT, username);
            intent.putExtra(FOLRGZ, folio.get(position).toString());
            intent.putExtra(BANDERA, "persingo");
            startActivity(intent);
        }
        else{
            Toast.makeText(getApplicationContext(), "Ya fue aprobado",
Toast.LENGTH_SHORT).show();
        }
    }
});

    final SwipeRefreshLayout swiprefre = (SwipeRefreshLayout)
findViewById(R.id.refrehlist);

    swiprefre.setColorSchemeResources(android.R.color.holo_blue_bright,
android.R.color.holo_green_light,
android.R.color.holo_orange_light,
android.R.color.holo_red_light);

    swiprefre.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener()
{
        @Override
        public void onRefresh() {
            swiprefre.setRefreshing(true);
            new Handler().postDelayed(new Runnable() {
                @Override public void run() {
                    username=idgerente.getText().toString();
                    swiprefre.setRefreshing(false);
                }
            }, 5000);
        }
    });

    btsalida();

}

//Método para poder finalizar sesión a través del botón salida.
private void btsalida() {
    salid.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent intexir = new
Intent(CorreoPermisngMg.this, MainActivity.class);
            startActivity(intexir);
            finish();

        }
    });
}
}

```

```

// Descarga todos los datos desde la base de datos remota.
private void iniciomail() {

    class obtpeticion extends AsyncTask<String, Void, String > {
        private Dialog windDialog;

        @Override
        protected void onPreExecute () {
            super.onPreExecute();
            windDialog = ProgressDialog.show(CorreoPermisngMg.this, "Please
wait", "Loading...");
        }

        @Override
        protected String doInBackground (String...params){
            DefaultHttpClient httpclient = new DefaultHttpClient(new
BasicHttpParams());
            HttpPost httppost = new
HttpPost("http://192.168.1.72/iniciarphtml/panel/codjsonpsingdepartamental.php?idge
r="+username);

            httppost.setHeader("Content-type", "application/json");

            InputStream inputStream = null;
            String result = null;
            try {
                HttpResponse response = httpclient.execute(httppost);
                HttpEntity entity = response.getEntity();

                inputStream = entity.getContent();
                BufferedReader reader = new BufferedReader(new
InputStreamReader(inputStream, "UTF-8"), 8);
                StringBuilder sb = new StringBuilder();

                String line = null;
                while ((line = reader.readLine()) != null) {
                    sb.append(line + "\n" );
                }
                result = sb.toString();
                reader.close();//--
            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            } catch (ClientProtocolException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            } finally {
                try {
                    if (inputStream != null) inputStream.close();
                } catch (Exception squish) {
                }
            }
            return result;
        }

        protected void onPostExecute(String result){
            windDialog.dismiss();
            myJSON=result;
            despleList();
        }
    }

    obtpeticion op = new obtpeticion();
    op.execute();
}

```

```

    }
    //Método para visualizar los datos en forma de lista
    private void despleList(){
        firmnger.clear();
        folio.clear();

        try {
            JSONObject jsonObj = new JSONObject(myJSON);
            peoples = jsonObj.getJSONArray(TAG_RESULTS);

            for(int i=0;i<peoples.length();i++){
                JSONObject c = peoples.getJSONObject(i);
                folio.add(c.getString(TAG_FOL));
                String id = c.getString(TAG_ID);
                String name = c.getString(TAG_NAME);
                String apu = c.getString(TAG_PAT);
                String aps = c.getString(TAG_MAT);
                String dizp = c.getString(TAG_DIA);
                String fstar = c.getString(TAG_FIN);
                String fend = c.getString(TAG_FFI);
                String mgfre = c.getString(TAG_MNG);
                firmnger.add(c.getString(TAG_MNG));

                HashMap<String,String> persons = new HashMap<String,String>();

                persons.put(TAG_NAME,name);
                persons.put(TAG_PAT,apu);
                persons.put(TAG_MAT,aps);
                persons.put(TAG_DIA,dizp);
                persons.put(TAG_FIN,fstar);
                persons.put(TAG_FFI,fend);
                persons.put(TAG_MNG,mgfre);

                personList.add(persons);
            }

            adapter = new SimpleAdapter(CorreoPermisngMg.this, personList,
R.layout.info_maildepartvaca,
                new
String[] {TAG_NAME, TAG_PAT, TAG_MAT, TAG_DIA, TAG_FIN, TAG_FFI, TAG_MNG},
                new int[] {R.id.txvwlisemailname, R.id.txvwlisemailapp,
R.id.txvwlisemailaps, R.id.txemaildepdias, R.id.txemaildepfinicio, R.id.txemaildepff
in, R.id.txemailmngde}
            );

            listapeticiones.setAdapter(adapter);

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

```

## 10.7 CorreoVacacionesMg.class

Despliega en forma de lista todas las solicitudes de vacaciones relacionadas con el Gerente de departamento.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.Dialog;

```

```

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Handler;
import android.support.v4.widget.SwipeRefreshLayout;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.HashMap;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.ClientProtocolException;
import cz.msebera.android.httpclient.client.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.params.BasicHttpParams;

public class CorreoVacacionesMg extends AppCompatActivity {

    String username;
    String myJSON;
    ArrayList firmnger = new ArrayList();
    ArrayList folio = new ArrayList();

    public static final String NMGT = "NMMGR";
    public static final String FOLRGZ = "FOLIOREGZ";
    public static final String BANDERA = "FLAGCG";

    private static final String TAG_FOL="idregivaca";
    private static final String TAG_RESULTS="maildep";
    private static final String TAG_ID = "trabajador_idtrabajador";
    private static final String TAG_NAME = "nombre";
    private static final String TAG_PAT = "aprimero";
    private static final String TAG_MAT = "asegundo";
    private static final String TAG_DIA = "tabudiasvacaciones_iddivkto";
    private static final String TAG_FIN = "fechainicio";
    private static final String TAG_FFI = "fechafin";
    private static final String TAG_MNG = "registrovacaciones.manager_iduser";

    JSONArray peoples = null;

    ArrayList<HashMap<String, String>> personList;

    ListView listapeticiones;

```

```

TextView idgerente;
String firdepto;
ImageButton solid;
public static ListAdapter adapter;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_correo_gerente);
    getSupportActionBar().hide();

    listapeticiones=(ListView) findViewById(R.id.lstcorreogerente);
    idgerente=(TextView) findViewById(R.id.textView16);
    solid=(ImageButton) findViewById(R.id.btnsaliext);

    Intent intent = getIntent();
    username = intent.getStringExtra(ModuloAutorizar.USER_LOG);
    idgerente.setText(username);
    if(intent.getStringExtra(FragmentDenegar.FGYTG) !=null) {
        username=intent.getStringExtra(FragmentDenegar.FGYTG);
        idgerente.setText(username);
    }else if(intent.getStringExtra(ScannerFinal.NMVJT) !=null) {
        username=intent.getStringExtra(ScannerFinal.NMVJT);
        idgerente.setText(username);
    }else if(intent.getStringExtra(AbrirGaleria.NMVJT) !=null) {
        username=intent.getStringExtra(AbrirGaleria.NMVJT);
        idgerente.setText(username);
    }

    personList = new ArrayList<HashMap<String,String>>();
    iniciomail();

    listapeticiones.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
            if(firmnger.get(position).toString().equals("vacio")){
                Intent intent = new
Intent(CorreoVacacionesMg.this,ContenedorFragmentMg.class);
                intent.putExtra(NMGT,username);
                intent.putExtra(FOLRGZ,folio.get(position).toString());
                intent.putExtra(BANDERA,"vacaciones");
                startActivity(intent);
            }
            else{
                Toast.makeText(getApplicationContext(),"Ya fue aprobado",
Toast.LENGTH_SHORT).show();
            }
        }
    });

    final SwipeRefreshLayout swiprefre = (SwipeRefreshLayout)
findViewById(R.id.refrehlist);

    swiprefre.setColorSchemeResources(android.R.color.holo_blue_bright,
android.R.color.holo_green_light,
android.R.color.holo_orange_light,
android.R.color.holo_red_light);

    swiprefre.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener()
{

```

```

        @Override
        public void onRefresh() {
            swiprefre.setRefreshing(true);
            new Handler().postDelayed(new Runnable() {
                @Override public void run() {
                    username=idgerente.getText().toString();
                    swiprefre.setRefreshing(false);
                }
            }, 5000);
        }
    });

    btsalida();

}

//Método para el botón salida el cual se pueda finalizar la sesión
private void btsalida() {
    valid.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent intexir = new
Intent(CorreoVacacionesMg.this,MainActivity.class);
            startActivity(intexir);
            finish();

        }
    });
}

//Método para descargar los datos a través de la base de datos de Cerebro.
private void iniciomail() {

    class obtpeticion extends AsyncTask<String, Void, String > {
        private Dialog windDialog;

        @Override
        protected void onPreExecute () {
            super.onPreExecute();
            windDialog = ProgressDialog.show(CorreoVacacionesMg.this, "Please
wait", "Loading...");
        }

        @Override
        protected String doInBackground (String...params){

            DefaultHttpClient httpclient = new DefaultHttpClient(new
BasicHttpParams());
            HttpPost httppost = new
HttpPost("http://192.168.1.72/iniciarphtml/panel/codjsoncorreodepartamental.php?idg
er="+username);

            httppost.setHeader("Content-type", "application/json");

            InputStream inputStream = null;
            String result = null;
            try {
                HttpResponse response = httpclient.execute(httppost);
                HttpEntity entity = response.getEntity();

```

```

        inputStream = entity.getContent();
        BufferedReader reader = new BufferedReader(new
InputStreamReader(inputStream, "UTF-8"), 8);
        StringBuilder sb = new StringBuilder();

        String line = null;
        while ((line = reader.readLine()) != null) {
            sb.append(line + "\n" );
        }
        result = sb.toString();
        reader.close();//--

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        try {
            if (inputStream != null) inputStream.close();
        } catch (Exception squish) {
        }
    }
    return result;
}

protected void onPostExecute(String result){
    windDialog.dismiss();
    myJSON=result;
    despleList();
}

obtpeticion op = new obtpeticion();
op.execute();
}

//Método para visualizar los datos en forma de lista
private void despleList(){
    firmnger.clear();
    folio.clear();

    try {
        JSONObject jsonObj = new JSONObject(myJSON);
        peoples = jsonObj.getJSONArray(TAG_RESULTS);

        for(int i=0;i<peoples.length();i++){
            JSONObject c = peoples.getJSONObject(i);
            folio.add(c.getString(TAG_FOL));
            String id = c.getString(TAG_ID);
            String name = c.getString(TAG_NAME);
            String apu = c.getString(TAG_PAT);
            String aps = c.getString(TAG_MAT);
            String dizp = c.getString(TAG_DIA);
            String fstar = c.getString(TAG_FIN);
            String fend = c.getString(TAG_FFI);
            String mgfre = c.getString(TAG_MNG);
            firmnger.add(c.getString(TAG_MNG));

            HashMap<String,String> persons = new HashMap<String,String>();

            persons.put(TAG_NAME,name);
            persons.put(TAG_PAT,apu);
            persons.put(TAG_MAT,aps);
            persons.put(TAG_DIA,dizp);

```

```

        persons.put (TAG_FIN, fstar);
        persons.put (TAG_FFI, fend);
        persons.put (TAG_MNG, mgfre);

        personList.add(persons);
    }

    adapter = new SimpleAdapter(CorreoVacacionesMg.this, personList,
R.layout.info_maildepartvaca,

        new
String[] {TAG_NAME, TAG_PAT, TAG_MAT, TAG_DIA, TAG_FIN, TAG_FFI, TAG_MNG},
        new int[] {R.id.txvwlisemailname, R.id.txvwlisemailapp,
R.id.txvwlisemailaps, R.id.txemaildepdias, R.id.txemaildepfinicio, R.id.txemaildepff
in, R.id.txemailmngde}
        );

    listapeticiones.setAdapter(adapter);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}
}

```

## 10.8 Cumpleaños.class

Muestra a todos los trabajadores que cumplen años en el mes que esta transcurriendo.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.ProgressDialog;
import android.content.Context;
import android.graphics.Rect;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.GridView;
import android.widget.ListView;
import android.widget.TextView;

import com.github.snowdream.android.widget.SmartImageView;
import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;

import org.json.JSONArray;
import org.json.JSONException;

import java.util.ArrayList;

import cz.msebera.android.httpclient.Header;

public class Cumpleaños extends AppCompatActivity {

    GridView grid;

    ArrayList fechasms = new ArrayList();

```



```

ArrayList nombretm = new ArrayList();
ArrayList appaterno = new ArrayList();
ArrayList perfil = new ArrayList();

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.info_cumple);
    grid = (GridView) findViewById(R.id.gridvw);

    descargarinformacion();
}

//Método que realiza la conexión de datos a Cerebro y se puedan leer.
private void descargarinformacion() {
    fechatms.clear();
    nombretm.clear();
    perfil.clear();
    appaterno.clear();

    final ProgressDialog mensajito = new ProgressDialog(Cumpleaños.this);
    mensajito.setMessage("Cargando cumpleaños...");
    mensajito.show();

    AsyncHttpClient clienttms = new AsyncHttpClient();

    clienttms.get("http://192.168.1.72/iniciarphtml/panel/happytms.php", new
    AsyncHttpResponseHandler() {
        @Override
        public void onSuccess(int statusCode, Header[] headers, byte[]
    responseBody) {
            if(statusCode == 200){
                mensajito.dismiss();
                try {
                    JSONArray jasonarraycum = new JSONArray(new
String(responseBody));
                    for (int i=0;i<jasonarraycum.length();i++){

nombretm.add(jasonarraycum.getJSONObject(i).getString("nombre"));

appaterno.add(jasonarraycum.getJSONObject(i).getString("appprimero"));

fechatms.add(jasonarraycum.getJSONObject(i).getString("vida"));

perfil.add(jasonarraycum.getJSONObject(i).getString("retrato"));
                }

                grid.setAdapter(new
    ListViewAdapter(getApplicationContext()));
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }

    @Override
    public void onFailure(int statusCode, Header[] headers, byte[]
    responseBody, Throwable error) {
    }
}

```

```

    });
}

//Método para visualizar los datos en forma de lista
private class ListViewAdapter extends BaseAdapter{

    Context contexto;
    LayoutInflater inflantado;
    SmartImageView smartfotografia;

    public ListViewAdapter(Context applicationContext){
        this.contexto=applicationContext;

inflantado=(LayoutInflater) contexto.getSystemService(LAYOUT_INFLATER_SERVICE);
    };

    @Override
    public int getCount() {
        return perfil.size();
    }

    @Override
    public Object getItem(int position) {
        return position;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        TextView nacimientoz;
        TextView nombrecompletms;

        ViewGroup vistagrupo =
        (ViewGroup) inflantado.inflate(R.layout.activity_cumpleaños, null);

        smartfotografia = (SmartImageView)
        vistagrupo.findViewById(R.id.imv_perfil);
        nacimientoz = (TextView) vistagrupo.findViewById(R.id.fechacumple);
        nombrecompletms = (TextView)
        vistagrupo.findViewById(R.id.nombrecumple);

        String urlperrfil =
        "http://192.168.1.72/iniciarpHP/perfiles/"+perfil.get(position).toString();

        Rect contorno = new
        Rect(smartfotografia.getLeft(), smartfotografia.getTop(), smartfotografia.getRight(
        ), smartfotografia.getBottom());
        smartfotografia.setImageUrl(urlperrfil, contorno);
        nacimientoz.setText(fechatms.get(position).toString());
        nombrecompletms.setText(nombrecompletms.get(position).toString()+"
        "+apaterno.get(position).toString());

        return vistagrupo;
    }
}
}

```

## 10.9 DetalleEvento.class

Muestra con detalle las actividades que se llevaran a cabo durante el mes.

```
package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.method.ScrollingMovementMethod;
import android.widget.ImageView;
import android.widget.TextView;

import com.squareup.picasso.Picasso;

public class DetalleEvento extends AppCompatActivity {

    TextView fecha,infor;
    ImageView poster;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detalle_lista);
        fecha= (TextView) findViewById(R.id.txfecha);
        poster= (ImageView) findViewById(R.id.ivfotografia);
        infor = (TextView) findViewById(R.id.txdescripcion);
        infor.setMovementMethod(new ScrollingMovementMethod());

        Intent intent = getIntent();
        Bundle extra = intent.getExtras();

        if(extra!=null){
            String dato = extra.getString("contenido");
            fecha.setText(dato);
            String retrato = extra.getString("pixis");
            String urlfotoz="http://192.168.1.72/iniciarphp/fotos/"+retrato;
            String contenido = extra.getString("detalle");
            infor.setText(contenido);
            cargarimagen(urlfotoz);
        }
    }
    //Método para visualizar las imágenes en pantalla
    private void cargarimagen(String urlfotoz) {
        Picasso.with(this).load(urlfotoz).placeholder(R.mipmap.nuvecargador)
            .error(R.mipmap.nuvecargador)
            .into(poster, new com.squareup.picasso.Callback() {
                @Override
                public void onSuccess() {

                }

                @Override
                public void onError() {

                }
            });
    }
}
```

```
}
```

## 10.10 DetalleVacaciones.class

Muestra con detalle los días de vacaciones, así como también la respuesta a la petición solicitada al Gerente.

```
package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

import org.w3c.dom.Text;

public class DetalleVacaciones extends AppCompatActivity {

    TextView numdia, comtxt;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detalle_vacaciones);
        numdia = (TextView) findViewById(R.id.numtxtdetvakaview);
        comtxt = (TextView) findViewById(R.id.comentxtdetavakaview);

        Intent intent = getIntent();
        Bundle extra = intent.getExtras();

        if(extra!=null){
            String dato = extra.getString("dikx");
            numdia.setText(dato);

            String contenido = extra.getString("comfrt");
            comtxt.setText(contenido);
        }
    }
}
```

## 10.11 Estacionamiento.class

La función que tiene es solicitar estacionamiento.

```
package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.AlertDialog;
import android.app.DatePickerDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
```

```

import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.Calendar;

public class Estacionamiento extends AppCompatActivity {

    TextView feiniv;
    TextView fefinal;
    ImageButton btncalendar;
    ImageButton btacep;
    public int anio,mes,dia,entdivk,ntmhil,senala;
    DatePickerDialog datePickerDialog;
    Button btnver;
    String dirip= "http://192.168.1.72/iniciarphp/";
    String getuno = dirip + "panel/guarparking.php";
    String empleado,fstar,fend,jefe;
    LigarEstacionamiento Hiloestacionamiento;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_estacionamiento);

        feiniv = (TextView) findViewById(R.id.fchincialestacion);
        fefinal = (TextView) findViewById(R.id.fchfincalestacion);

        btacep = (ImageButton) findViewById(R.id.imgbtconfirestacion);
        btnver = (Button) findViewById(R.id.btnidverregistacion);

        Intent intanciado = getIntent();
        Bundle extrainta = intanciado.getExtras();
        if(extrainta!=null) {
            empleado = extrainta.getString("numtm");
            ntmhil = Integer.valueOf(empleado);
            jefe=extrainta.getString("gerente");
        }else{
            Toast.makeText(Estacionamiento.this,"No hay datos",
Toast.LENGTH_LONG).show();
        }
        final Calendar cal = Calendar.getInstance();
        dia=cal.get(Calendar.DAY_OF_MONTH);
        mes=cal.get(Calendar.MONTH);
        anio=cal.get(Calendar.YEAR);
        ponerfechador();
        entdivk=1;

        showDialogOnButtonClickdos();

        metodoacep();
    }
}

```

```

        metoregist();
    }
    //Método que brinda funcionalidad al botón aceptar, y se puedan enviar los datos
    a la base de datos remota
    private void metodoacep() {
        btacep.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                if(entdivk>=1) {
                    Hiloestacionamiento = new LigarEstacionamiento();
                    String dirconsultd = getuno + "?ntm=" + ntmhil + "&dzhave=" +
entdivk + "&fini=" + fstar + "&ffin=" + fend;
                    Hiloestacionamiento.execute(dirconsultd);
                    AlertDialog.Builder bldz = new
AlertDialog.Builder(Estacionamiento.this);
                    bldz.setMessage("Los datos fueron enviados")
                        .setTitle("Ok")
                        .setCancelable(false)
                        .setNeutralButton("Ok", new
DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog, int
which) {

                                }
                            });
                    AlertDialog delaf = bldz.create();
                    delaf.show();

                }
                else{
                    AlertDialog.Builder builx = new
AlertDialog.Builder(Estacionamiento.this);
                    builx.setMessage("Faltan campos")
                        .setTitle("Error")
                        .setCancelable(false)
                        .setNeutralButton("OK", new
DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog, int
which) {

                                dialog.cancel();

                            }
                        });
                    AlertDialog alermj = builx.create();
                    alermj.show();
                }
            }
        });
    }
    //Método que da función al botón ver registros
    private void metoregist() {

        btnver.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent verre = new
Intent(Estacionamiento.this, VerRegistroestacionamiento.class);

```

```

        verre.putExtra("tmnum", empleado);
        startActivity(verre);
    }
});
}
//Método para mostrar el calendario
private void showDialogOnButtonClickdos() {
    btncalendar =(ImageButton) findViewById(R.id.btncalestacionamiento);
    btncalendar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            datePickerDialog = new DatePickerDialog(Estacionamiento.this, new
            DatePickerDialog.OnDateSetListener() {
                @Override
                public void onDateSet(DatePicker view, int year, int month,
int dayOfMonth) {
                    month=month+1;
                    feiniv.setText(dayOfMonth+"/"+month+"/"+year);
                    fstar=year+"/"+month+"/"+dayOfMonth;
                    final Calendar ck = Calendar.getInstance();
                    entdivk=1;
                    senala=1;

                    ck.set(year, month, dayOfMonth);

                    fefinal.setText(dayOfMonth + "/" + month + "/" +
year);
                    fend=year+"/"+month+"/"+dayOfMonth;

                }
            }, anio, mes, dia);

            datePickerDialog.show();
        }
    });
}

private void ponerfechador() {
    feiniv.setText(dia+"/"+(mes+1)+"/"+anio);
    fstar=anio+"/"+(mes+1)+"/"+dia;
}

//Método para establecer la conexión a Cerebro y se pedan enviar los datos
private class LigarEstacionamiento extends AsyncTask<String,Void,Void> {
    @Override
    protected Void doInBackground(String... params) {
        String cadena = params[0];
        URL url = null;

        try {
            url = new URL(cadena);
            HttpURLConnection connexion =
(HttpURLConnection)url.openConnection();
            connexion.setRequestProperty("Content-Type", "text/html");
            connexion.setRequestProperty("Host", "192.168.1.72");
            connexion.setRequestProperty("User-

```

```

Agent", "Mozilla/5.0(X11;Ubuntu;Linux i686;rv:49.0) Gecko/20100101 Firefox/49.0");
    connexion.setRequestMethod("GET");
    int respuesta = connexion.getResponseCode();
    StringBuilder result = new StringBuilder();

    if (respuesta == HttpURLConnection.HTTP_OK) {

        InputStream in = new
BufferedInputStream(connexion.getInputStream());

        BufferedReader reader = new BufferedReader(new
InputStreamReader(in));

        String line=null;
        while ((line = reader.readLine()) != null) {
            result.append(line);
            break;
        }

        reader.close();

    }
} catch (IOException e) {
    e.printStackTrace();
}

return null;
}
}
}

```

## 10.12 FragmentAceptar.class

Contiene dos métodos para autorizar un trámite. El primer método con la apertura de la cámara del Smartphone y el segundo método con un explorador de archivos.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageButton;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedInputStream;
import java.io.BufferedReader;

```



```

import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

/**
 * A simple {@link Fragment} subclass.
 * Activities that contain this fragment must implement the
 * {@link FragmentAceptar.OnFragmentInteractionListener} interface
 * to handle interaction events.
 */
public class FragmentAceptar extends Fragment{

    private OnFragmentInteractionListener mListener;
    public static final String USRMG = "USRNAGRE";
    public static final String FXIO = "FOLKIUL";
    public static final String CVGT = "CLAVEGENENTE";
    public static final String PSSGTQR = "CLAVEQRGT";
    public static final String FLAGMIX = "BANDERAMIXZ";

    ImageButton fqr;
    Button xporeer;
    public String paramac1,paramac2,parambandera,clav,passqr;
    ConnxWbSvss hilosevert;
    String dirip= "http://192.168.1.72/iniciarpq/";
    String getuno = dirip + "panel/codjsoncodigoqr.php";

    public FragmentAceptar() {
        // Required empty public constructor
    }
    //Método para obtener valores de otra activity
    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            paramac1= getArguments().getString(ContenedorFragmentMg.FOLIA);
            paramac2= getArguments().getString(ContenedorFragmentMg.NMJJ);
            parambandera =
getArguments().getString(ContenedorFragmentMg.BANDMIX);

        }
    }
    //Método que contiene al botón para escanear el Código QR o abrir el explorador
    de archivos
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View
view=inflater.inflate(R.layout.fragment_blank_fragment_acep_email_dep, container,
false);

        fqr=(ImageButton)view.findViewById(R.id.imgqrbtn);
        xporeer=(Button)view.findViewById(R.id.btnexploracep);

        hilosevert = new ConnxWbSvss();
        String diradres=getuno + "?mn=" + paramac2;
        hilosevert.execute(diradres);

```

```

        fqr.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent segundascane=new
Intent(getContext(),ScannerFinal.class);
                segundascane.putExtra(USRMG,paramac2);
                segundascane.putExtra(FXIO,paramac1);
                segundascane.putExtra(CVGT,clav);
                segundascane.putExtra(PSSGTQR,passqr);
                segundascane.putExtra(FLAGMIX,parambandera);

                startActivity(segundascane);
                getActivity().finish();
                try {
                    FragmentAceptar.this.finalize();
                } catch (Throwable throwable) {
                    throwable.printStackTrace();
                }

            }
        });

        xporeer.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent tercer = new Intent(getContext(),AbrirGaleria.class);
                tercer.putExtra(USRMG,paramac2);
                tercer.putExtra(FXIO,paramac1);
                tercer.putExtra(CVGT,clav);
                tercer.putExtra(PSSGTQR,passqr);
                tercer.putExtra(FLAGMIX,parambandera);
                startActivity(tercer);
                getActivity().finish();
                try {
                    FragmentAceptar.this.finalize();
                } catch (Throwable throwable) {
                    throwable.printStackTrace();
                }

            }
        });

        return view;
    }

    // TODO: Rename method, update argument and hook method into UI event
    public void onPressed(Uri uri) {
        if (mListener != null) {
            mListener.onFragmentInteraction(uri);
        }
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof OnFragmentInteractionListener) {
            mListener = (OnFragmentInteractionListener) context;
        } else {

```

```

        throw new RuntimeException(context.toString()
            + " must implement OnFragmentInteractionListener");
    }
}

@Override
public void onDetach() {
    super.onDetach();
    mListener = null;
}

/**
 * This interface must be implemented by activities that contain this
 * fragment to allow an interaction in this fragment to be communicated
 * to the activity and potentially other fragments contained in that
 * activity.
 * <p>
 * See the Android Training lesson <a href=
 *
"http://developer.android.com/training/basics/fragments/communicating.html"
 * >Communicating with Other Fragments</a> for more information.
 */
public interface OnFragmentInteractionListener {
    // TODO: Update argument type and name
    void onFragmentInteraction(Uri uri);
}

//Método encargado de conectarse con la base de datos y obtener contraseña y
firma electrónica vanzada para posteriormente ser comparadas para validar los
tramites.
private class ConnxWbSvss extends AsyncTask<String,Void,String> {
    @Override
    protected String doInBackground(String... params) {
        String cadena = params[0];
        URL url = null;

        try {
            url = new URL(cadena);
            HttpURLConnection connexion =
(HttpURLConnection)url.openConnection();
            connexion.setRequestProperty("Content-Type","text/html");
            connexion.setRequestProperty("Host","192.168.1.72");
            connexion.setRequestProperty("User-
Agent","Mozilla/5.0(X11;Ubuntu;Linux i686;rv:49.0) Gecko/20100101 Firefox/49.0");
            connexion.setRequestMethod("GET");
            int respuesta = connexion.getResponseCode();
            StringBuilder result = new StringBuilder();

            if (respuesta == HttpURLConnection.HTTP_OK) {

                InputStream in = new
BufferedInputStream(connexion.getInputStream()); // preparo la cadena de
entrada

                BufferedReader reader = new BufferedReader(new
InputStreamReader(in));

                String line=null;
                while ((line = reader.readLine()) != null) {
                    result.append(line);
                    break;
                }
            }
        }
    }
}

```

```

        reader.close();

        JSONObject respuestaJSON = new JSONObject(result.toString());
        JSONArray resultaJSON =
respuestaJSON.getJSONArray("mngqrsec");
        JSONObject c = resultaJSON.getJSONObject(0);
        clav = c.getString("claveus");
        passqr = c.getString("qrsecret");

    }

    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (JSONException e) {
        e.printStackTrace();
    }

    return null;
}
}
}
}
}

```

### 10.13 FragmentDenegar.class

Su función es denegar la solicitud, así como comentar la razón o motivo por la cual no procede.

```
package roberto.espejel.app.proyecto.docpt.rulaitas.ian;
```

```

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.BufferedInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;

```

```
/**
```

```

* A simple {@link Fragment} subclass.
* Activities that contain this fragment must implement the
* {@link FragmentDenegar.OnFragmentInteractionListener} interface
* to handle interaction events.
*/
public class FragmentDenegar extends Fragment {

    private String param1,param2,parabanderpsing;
    public static final String FGYTG = "GTHY";
    private OnFragmentInteractionListener mListener;
    Button enviar;
    EditText comenatario;
    ConnWebServidor hiloservidor;
    String dirip= "http://192.168.1.72/iniciarphp/";
    String getuno = dirip + "panel/guarcancelarvaca.php";
    String getdos = dirip + "panel/guarcancelarpsing.php";
    String gettres = dirip + "panel/guarcancelarpcong.php";
    String getcuatro = dirip + "panel/guarcancelarestaciona.php";
    public String dirconsultd;
    public FragmentDenegar() {
        // Required empty public constructor
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if(getArguments() !=null){

            param1= getArguments().getString(ContenedorFragmentMg.FOLIA);
            param2= getArguments().getString(ContenedorFragmentMg.NMJT);
            parabanderpsing =
getArguments().getString(ContenedorFragmentMg.BANDMIX);

        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_blank_fragment_email_dep,
container, false);
        enviar = (Button)view.findViewById(R.id.btnfragsen);
        comenatario = (EditText)view.findViewById(R.id.edtxmultifragment);

        enviar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                hiloservidor = new ConnWebServidor();
                String monto= null;
                try {
                    monto =
URLDecoder.decode(String.valueOf(comenatario.getText()),"UTF-8");
                } catch (UnsupportedEncodingException e) {
                    e.printStackTrace();
                }
                if(parabanderpsing.equals("percongo")){
                    dirconsultd = gettres + "?ntm=" + param1 + "&come=cancelada-"
+ monto;
                }
                else if(parabanderpsing.equals("persingo")){
                    dirconsultd = getdos + "?ntm=" + param1 + "&come=cancelada-"
+ monto;
                }
            }
        });
    }
}

```

```

        }else if(parabanderpsing.equals("vacaciones")){
            dirconsultd = getuno + "?ntm=" + param1 + "&come=cancelada-"
+ monto;
        }else{
            dirconsultd = getcuatro + "?ntm=" + param1 +
"&come=cancelada-" + monto;
        }
        hiloservidor.execute(dirconsultd);
        Toast.makeText(getActivity(),"Solicitud
cancelada",Toast.LENGTH_LONG).show();
        Intent intenfrat = new Intent(getContext(),
ModuloAutorizar.class);
        intenfrat.putExtra(FGYTG, param2);

        startActivity(intenfrat);
        getActivity().finish();
    }
});
return view;
}

// TODO: Rename method, update argument and hook method into UI event
public void onPressed(Uri uri) {
    if (mListener != null) {
        mListener.onFragmentInteraction(uri);
    }
}

@Override
public void onAttach(Context context) {
    super.onAttach(context);
    if (context instanceof OnFragmentInteractionListener) {
        mListener = (OnFragmentInteractionListener) context;
    } else {
        throw new RuntimeException(context.toString()
+ " must implement OnFragmentInteractionListener");
    }
}

@Override
public void onDetach() {
    super.onDetach();
    mListener = null;
}

/**
 * This interface must be implemented by activities that contain this
 * fragment to allow an interaction in this fragment to be communicated
 * to the activity and potentially other fragments contained in that
 * activity.
 * <p>
 * See the Android Training lesson <a href=
"http://developer.android.com/training/basics/fragments/communicating.html"
 * >Communicating with Other Fragments</a> for more information.
 */
public interface OnFragmentInteractionListener {
    // TODO: Update argument type and name
    void onFragmentInteraction(Uri uri);
}

private class ConnWebServidor extends AsyncTask<String,Void,Void> {

```

```

@Override
protected Void doInBackground(String... params) {
    String cadena = params[0];
    URL url = null;
    String devuelve = "";

    try {
        url = new URL(cadena);
        HttpURLConnection connexion =
(HttpURLConnection)url.openConnection();
        connexion.setRequestProperty("Content-Type", "text/html");
        connexion.setRequestProperty("Host", "192.168.1.72");
        connexion.setRequestProperty("User-
Agent", "Mozilla/5.0(X11;Ubuntu;Linux i686;rv:49.0) Gecko/20100101 Firefox/49.0");
        connexion.setRequestMethod("GET");
        int respuesta = connexion.getResponseCode();
        StringBuilder result = new StringBuilder();

        if (respuesta == HttpURLConnection.HTTP_OK) {

            InputStream in = new
BufferedInputStream(connexion.getInputStream()); // preparo la cadena de
entrada

            BufferedReader reader = new BufferedReader(new
InputStreamReader(in)); // la introduzco en un BufferedReader

            // El siguiente proceso lo hago porque el JSONObject necesita
un String y tengo
            // que transformar el BufferedReader a String. Esto lo hago a
traves de un
            // StringBuilder.

            String line=null;
            while ((line = reader.readLine()) != null) {
                result.append(line); // Paso toda la entrada al
StringBuilder

                break;
            }

            reader.close();

        }
    } catch (IOException e) {
        e.printStackTrace();
    }

    return null;
}
}
}

```

## 10.14 Genereqr.class

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.os.Environment;

```

```

import android.provider.MediaStore;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import com.google.zxing.BarcodeFormat;
import com.google.zxing.MultiFormatWriter;
import com.google.zxing.WriterException;
import com.google.zxing.common.BitMatrix;
import com.journeyapps.barcodescanner.BarcodeEncoder;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

public class Genereqr extends AppCompatActivity {

    EditText escribir;
    Button generar;
    Button saveimage;
    ImageView imgqr;
    String cadenaqr;
    Bitmap bitmap;
    int bandera=0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_genereqr);
        escribir = (EditText) findViewById(R.id.editxtcodeqr);
        generar = (Button) findViewById(R.id.btncodeqr);
        imgqr = (ImageView) findViewById(R.id.imgcodegeneqr);
        saveimage = (Button) findViewById(R.id.btnqrsaveasqr);

        generar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                cadenaqr = escribir.getText().toString().trim();
                MultiFormatWriter multiFormatWriter = new MultiFormatWriter();
                try{
                    BitMatrix bitMatrix = multiFormatWriter.encode(cadenaqr,
BarcodeFormat.QR_CODE,200,200);
                    BarcodeEncoder barcodeEncoder = new BarcodeEncoder();
                    bitmap = barcodeEncoder.createBitmap(bitMatrix);
                    imgqr.setImageBitmap(bitmap);
                    bandera=1;
                }
                catch (WriterException e){
                    e.printStackTrace();
                }
            }
        });

        saveimage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Context constesto;
                constesto=getApplicationContext();
                if(bandera==1) {

```





```

import java.io.IOException;
import java.io.InputStream;

public class IntroApp extends AppCompatActivity {

    private GifImageView gifvista;
    private ProgressBar procesobarra;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_intro_app);

        gifvista = (GifImageView) findViewById(R.id.gifmarco);
        procesobarra = (ProgressBar) findViewById(R.id.barraproceso);
        procesobarra.setVisibility(procesobarra.VISIBLE);

        try {
            InputStream ips = getAssets().open("hammock.gif");
            byte[] bites = IOUtils.toByteArray(ips);
            gifvista.setBytes(bites);
            gifvista.startAnimation();
        } catch (IOException e) {
            e.printStackTrace();
        }

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                IntroApp.this.startActivity(new
Intent(IntroApp.this, MainActivity.class));
                IntroApp.this.finish();
            }
        }, 8500);
    }
}

```

## 10.16 LoginGM.class

Funciona para iniciar sesión como Gerente.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;

```

```

import java.util.ArrayList;
import java.util.List;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.NameValuePair;
import cz.msebera.android.httpclient.client.ClientProtocolException;
import cz.msebera.android.httpclient.client.HttpClient;
import cz.msebera.android.httpclient.client.entity.UrlEncodedFormEntity;
import cz.msebera.android.httpclient.client.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.message.BasicNameValuePair;

public class LoginGM extends AppCompatActivity {

    EditText usuario;
    EditText contra;
    public static final String USER_LOG = "USERNAME";

    String username;
    String password;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_idup);
        usuario = (EditText) findViewById(R.id.iduseretmn);
        contra = (EditText) findViewById(R.id.idpassetmn);
    }
    //Método para obtener usuario y contraseña
    public void bdmanager(View view){
        username=usuario.getText().toString();
        password=contra.getText().toString();

        methodologear(username,password);
    }
    //Método que realiza la conexión a Cerebro para validar el usuario y contraseña
    private void methodologear(final String username, String password) {
        class hiloconexmg extends AsyncTask<String,Void,String> {

            private Dialog ventanaDialog;

            @Override
            protected void onPreExecute() {
                super.onPreExecute();
                ventanaDialog = ProgressDialog.show(LoginGM.this, "Please
wait", "Loading...");
            }

            @Override
            protected String doInBackground(String... params) {
                String uname = params[0];
                String pass = params[1];

                InputStream is = null;
                List<NameValuePair> nameValuePairs = new
ArrayList<NameValuePair>();
                nameValuePairs.add(new BasicNameValuePair("username",
uname));
                nameValuePairs.add(new BasicNameValuePair("password", pass));
                String result = null;

                try{

```

```

        HttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost(
"http://192.168.1.72/iniciarpHP/panel/codjsonlogingerente.php");
        httpPost.setEntity(new
UrlEncodedFormEntity(nameValuePairs));

        HttpResponse response = httpClient.execute(httpPost);

        HttpEntity entity = response.getEntity();

        is = entity.getContent();

        BufferedReader reader = new BufferedReader(new
InputStreamReader(is, "UTF-8"), 8);
        StringBuilder sb = new StringBuilder();

        String line = null;
        while ((line = reader.readLine()) != null)
        {
            sb.append(line);
        }
        result = sb.toString();
        reader.close();//--
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    }
    return result;
}

@Override
protected void onPostExecute(String result){
    String s = result.trim();
    ventanaDialog.dismiss();
    if(s.equalsIgnoreCase("success")){
        Intent intent = new
Intent(LoginGM.this,ModuloAutorizar.class);
        intent.putExtra(USER_LOG, username);
        finish();
        startActivity(intent);
    }else {
        Toast.makeText(getApplicationContext(), "Invalid User
Name or Password", Toast.LENGTH_LONG).show();
    }
}

}

hiloconexmg la = new hiloconexmg();
la.execute(username, password);
}
}

```

## 10.17 LoginTms.class

Funciona para iniciar sesión como trabajador.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.List;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.NameValuePair;
import cz.msebera.android.httpclient.client.ClientProtocolException;
import cz.msebera.android.httpclient.client.HttpClient;
import cz.msebera.android.httpclient.client.entity.UrlEncodedFormEntity;
import cz.msebera.android.httpclient.client.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.message.BasicNameValuePair;

public class LoginTms extends AppCompatActivity {

    private EditText campo1;
    private EditText campo2;

    public static final String USER_NAME = "USERNAME";

    String username;
    String password;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login_tms);
        campo1=(EditText) findViewById(R.id.iduntms);
        campo2=(EditText) findViewById(R.id.idpwtms);
    }
    //Método que captura la contraseña y usuario
    public void llamadalink(View view){
        username=campo1.getText().toString();
        password=campo2.getText().toString();
        funlogin(username,password);
    }

    //Método que se conecta al Cerebro para validar usuario y contraseña
    private void funlogin(final String username, String password) {

        class LoginAsync extends AsyncTask<String, Void, String> {

```

```

private Dialog loadingDialog;

@Override
protected void onPreExecute() {
    super.onPreExecute();
    loadingDialog = ProgressDialog.show(LoginTms.this, "Please wait",
>Loading...");
}

@Override
protected String doInBackground(String... params) {
    String uname = params[0];
    String pass = params[1];

    InputStream is = null;
    List<NameValuePair> nameValuePairs = new
ArrayList<NameValuePair>();
    nameValuePairs.add(new BasicNameValuePair("username", uname));
    nameValuePairs.add(new BasicNameValuePair("password", pass));
    String result = null;

    try{
        HttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost(
"http://192.168.1.72/iniciarpHP/panel/codigojsonuserpass.php");
        httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));

        HttpResponse response = httpClient.execute(httpPost);

        HttpEntity entity = response.getEntity();

        is = entity.getContent();

        BufferedReader reader = new BufferedReader(new
InputStreamReader(is, "UTF-8"), 8);
        StringBuilder sb = new StringBuilder();

        String line = null;
        while ((line = reader.readLine()) != null)
        {
            sb.append(line);
        }
        result = sb.toString();
        reader.close();//--

    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return result;
}

@Override
protected void onPostExecute(String result){
    String s = result.trim();
    loadingDialog.dismiss();
    if(s.equalsIgnoreCase("success")){
        Intent intent = new Intent(LoginTms.this,
ModuloFormatos.class);

```

```

        intent.putExtra(USER_NAME, username);
        startActivity(intent);
        finish();
    }else {
        Toast.makeText(getApplicationContext(), "Invalid User Name or Password", Toast.LENGTH_LONG).show();
    }
}

LoginAsync la = new LoginAsync();
la.execute(username, password);
}
}

```

## 10.18 MainActivity.class

Es la contenedora de todos los módulos relacionados al proyecto planteado, así como el salt.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.BufferedInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener{

    ImageButton btncalendar;
    ImageButton btnformatos;
    ImageButton btnfirmaz;
    TextView saltinicial;
    TextView saltfinal;
    String dirip= "http://192.168.1.72/iniciarpHP/";

```

```

String getall= dirip + "panel/codjsonsalt.php";
RecaWebServicio hilologin;
public String salthoy;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    getSupportActionBar().setTitle("FormatosDigitales");
    toolbar.setLogo(android.R.drawable.ic_dialog_info);

    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                .setAction("Action", null).show();
        }
    });
    btncalendar=(ImageButton) findViewById(R.id.imbtnCalendario);
    btncalendar.setOnClickListener(this);
    btnformatos=(ImageButton) findViewById(R.id.imbtnFormatos);
    btnformatos.setOnClickListener(this);
    btnfirmaz=(ImageButton) findViewById(R.id.imbtnFirma);
    btnfirmaz.setOnClickListener(this);
    saltinicial = (TextView) findViewById(R.id.saltmetatxtview);
    saltfinal = (TextView) findViewById(R.id.saltdiatxtview);

    hilologin = new RecaWebServicio();
    hilologin.execute(getall);

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}
//Método que le da función a la barra de herramientas
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    Context context = getApplicationContext();
    CharSequence text = null;
    int duration = Toast.LENGTH_SHORT;

    if(id == R.id.menuop3) {
        text="bye";
        finish();
    }
}

```



```

        Toast toast = Toast.makeText(context, text, duration);
        toast.show();

        return super.onOptionsItemSelected(item);
    }
}
//Método encargado de que cada botón tenga su propia funcionalidad.
@Override
public void onClick(View v) {
    Context context = getApplicationContext();
    CharSequence texto = null;
    int duration = Toast.LENGTH_SHORT;
    switch(v.getId()){
        case R.id.imbtnCalendario:
            Intent intent = new Intent(this, ModuloCalendario.class);
            startActivity(intent);

            break;

        case R.id.imbtnFormatos:
            Intent intent1 = new Intent(this, LoginTms.class);
            startActivity(intent1);
            break;

        case R.id.imbtnFirma:
            Intent inten2 = new Intent(this, LoginGM.class);
            startActivity(inten2);
            break;

        default:
            texto="Opcion sin funcion";
            Toast toast = Toast.makeText(context, texto, duration);
            toast.show();
            break;
    }
}

}

//Método para conectarse a Cerebro y descargue los datos correspondientes al SALT
private class RecaWebServicio extends AsyncTask<String,Void,String>{

    @Override
    protected String doInBackground(String... params) {
        String cadena= params[0];//url para la obtencion de datos remoto
        String devuelve = "";
        URL url = null;
        try{
            url = new URL(cadena);
            HttpURLConnection conexion =
(HttpURLConnection)url.openConnection(); //se abre la conexion

            conexion.setRequestProperty("Content-Type", "text/html");
            conexion.setRequestProperty("Host", "192.168.1.72");
            conexion.setRequestProperty("User-
Agent", "Mozilla/5.0(X11;Ubuntu;Linux i686;rv:49.0) Gecko/20100101 Firefox/49.0");

            int respuesta = conexion.getResponseCode();
            StringBuilder resul = new StringBuilder();

            if(respuesta==HttpURLConnection.HTTP_OK){
                InputStream in = new
BufferedInputStream(conexion.getInputStream()); //prepara la cadena de entrada

```

```

        BufferedReader leer = new BufferedReader(new
InputStreamReader(in)); //

        String renglon;
        while((renglon=leer.readLine())!=null){
            resul.append(renglon);
        }
        JSONObject respcodjson = new JSONObject(resul.toString());
        JSONArray resuljson = respcodjson.getJSONArray("saltt");
        String direcciona = "SIN DATOS";
        devuelve=resuljson.getJSONObject(0).getString("porcentaje");
        salthoy = resuljson.getJSONObject(1).getString("porcentaje");
    }

    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (JSONException e) {
        e.printStackTrace();
    }
    }
    return devuelve;
}

@Override
protected void onPostExecute(String s) {
    super.onPostExecute(s);
    saltinicial.setText(s+"%");
    saltfinal.setText(salthoy+"%");
}
}
}

```

## 10.19 ModuloAutorizar.class

Contiene en divisiones de pestañas las listas de vacaciones, permiso sin goce de sueldo, permiso con goce de sueldo y estacionamiento.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.TabActivity;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.widget.TabHost;

public class ModuloAutorizar extends TabActivity {
    String username;
    public static final String USER_LOG = "USERNAME";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pestana_gerente);

        Intent intent = getIntent();
        username = intent.getStringExtra(LoginGM.USER_LOG);
        if(intent.getStringExtra(FragmentDenegar.FGYTG)!=null){
            username=intent.getStringExtra(FragmentDenegar.FGYTG);

```

```

} else if (intent.getStringExtra(ScannerFinal.NMVJT) != null) {
    username = intent.getStringExtra(ScannerFinal.NMVJT);
} else if (intent.getStringExtra(AbrirGaleria.NMVJT) != null) {
    username = intent.getStringExtra(AbrirGaleria.NMVJT);
}

TabHost tabhostmg = (TabHost) findViewById(android.R.id.tabhost);
TabHost.TabSpec tab6 = tabhostmg.newTabSpec("mitab6");
TabHost.TabSpec tab7 = tabhostmg.newTabSpec("mitab7");
TabHost.TabSpec tab8 = tabhostmg.newTabSpec("Mitab8");
TabHost.TabSpec tab9 = tabhostmg.newTabSpec("Mitab9");

tab6.setIndicator("", getResources().getDrawable(android.R.drawable.ic_menu_mapmode));
Intent vacanns = new Intent(this, CorreoVacacionesMg.class);
vacanns.putExtra(USER_LOG, username);
tab6.setContent(vacanns);

tab7.setIndicator("", getResources().getDrawable(android.R.drawable.ic_menu_day));
Intent persings = new Intent(this, CorreoPermisingMg.class);
persings.putExtra(USER_LOG, username);
tab7.setContent(persings);

tab8.setIndicator("", getResources().getDrawable(android.R.drawable.ic_menu_recent_history));
Intent percong = new Intent(this, CorreoPermicongMg.class);
percong.putExtra(USER_LOG, username);
tab8.setContent(percong);

//tab9.setIndicator("", getResources().getDrawable(android.R.drawable.ic_menu_myplaces));

tab9.setIndicator("", getResources().getDrawable(R.drawable.tabcamionetapng));
Intent parkimetro = new Intent(this, CorreoEstacionamientoMg.class);
parkimetro.putExtra(USER_LOG, username);
tab9.setContent(parkimetro);

tabhostmg.addTab(tab6);
tabhostmg.addTab(tab7);
tabhostmg.addTab(tab8);
tabhostmg.addTab(tab9);
}
}

```

## 10.20 ModuloCalendario.class

Contiene en divisiones de pestañas las listas de actividades, permiso sin goce de sueldo, permiso con goce de sueldo y estacionamiento.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.TabActivity;
import android.content.Intent;

```

```

import android.os.Bundle;
import android.widget.TabHost;

@SuppressWarnings("deprecation")
public class ModuloCalendario extends TabActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pestanas_menu);

        TabHost tabhost = (TabHost) findViewById(android.R.id.tabhost);
        TabHost.TabSpec tab1 = tabhost.newTabSpec("mitab1");
        TabHost.TabSpec tab2 = tabhost.newTabSpec("mitab2");
        TabHost.TabSpec tab3 = tabhost.newTabSpec("mitab3");

        tab1.setIndicator("",getResources().getDrawable(android.R.drawable.ic_menu_recent
        _history));
        tab1.setContent(new Intent(this,ActividadMesual.class));

        tab2.setIndicator("",getResources().getDrawable(android.R.drawable.ic_menu_my_cal
        endar));
        tab2.setContent(new Intent(this,Cumpleaños.class));

        tab3.setIndicator("",getResources().getDrawable(android.R.drawable.ic_menu_myplac
        es));
        tab3.setContent(new Intent(this,Vacantes.class));

        tabhost.addTab(tab1);
        tabhost.addTab(tab2);
        tabhost.addTab(tab3);
    }
}

```

## 10.21 ModuloFormatos.class

Contiene en divisiones de lista los tramites de vacaciones, permiso sin goce de sueldo, permiso con goce de sueldo y estacionamiento.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
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.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;

import org.json.JSONArray;

```

```

import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.Date;

/**
 * Created by Roberto Espejel Ramirez 207205750 on 14/03/2017.
 */

public class ModuloFormatos extends AppCompatActivity implements
View.OnClickListener {

    TextView tnombre, tarea, tdepto, tpuesto, tingresodate, tnumertm;
    ImageButton exitsec, continuar;
    Spinner opcionex;
    public int pasar;
    private ProgressDialog pDialog;

    String dirip= "http://192.168.1.72/iniciarpHP/";
    String getuno = dirip + "panel/codjsontmidhil.php";

    public String aniv, dvak, acumu, toditoma, manager;
    public int paso2;

    String username;
    ObtenerWebService hiloconexion;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.info_formatos);

        exitsec = (ImageButton) findViewById(R.id.btnsalida);
        tnombre = (TextView) findViewById(R.id.tvnombre);
        tarea = (TextView) findViewById(R.id.tvarea);
        tdepto = (TextView) findViewById(R.id.tvdepto);
        tpuesto = (TextView) findViewById(R.id.tvpuesto);

        continuar = (ImageButton) findViewById(R.id.ibtnsiguiente);
        opcionex = (Spinner) findViewById(R.id.spformatz);
        tingresodate = (TextView) findViewById(R.id.tvfechaingreso);
        tnumertm = (TextView) findViewById(R.id.txtidmtsnumb);

        ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(this, R.array.name_formatox, android.R.layout.simple_spinner_item);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
opcionex.setAdapter(adapter);

        Intent intent = getIntent();

```

```

        username = intent.getStringExtra(LoginTms.USER_NAME);
        tnumertm.setText(username);
    if(intent.getStringExtra(PermisosSin.NMTMS)!=null){
        username=intent.getStringExtra(PermisosSin.NMTMS);
        tnumertm.setText(username);
    }else if(intent.getStringExtra(Vacaciones.NMTYM)!=null){
        username=intent.getStringExtra(Vacaciones.NMTYM);
        tnumertm.setText(username);
    }

    metodoconsulta();
    exitsec.setOnClickListener(this);
    continuar.setOnClickListener(this);
    pasar =0;
}

//Método que envía el numero de trabajador para consultar sus datos
private void metodoconsulta() {
    hiloconexion = new ObtenerWebService();
    String dirconsultd = getuno + "?eq=" + username;
    hiloconexion.execute(dirconsultd);
}

//Método para darle función al Spinner y cuando de clic se translade a una
activity especifica
@Override
public void onClick(View v) {
    switch (v.getId()){

        case R.id.ibtnsiguiente:
            String textospiner=opcionex.getSelectedItem().toString();
            paso2=tdepto.length();
            if(textospiner.equals("VACACIONES")){

                if(aniv.equals("-1")){
                    AlertDialog.Builder builtt = new
AlertDialog.Builder(ModuloFormatos.this);
                    builtt.setMessage("Debe tener minimo 1 año")
                        .setTitle("Error")
                        .setCancelable(false)
                        .setNeutralButton("OK", new
DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog, int
which) {
                                dialog.cancel();
                            }
                        });
                    AlertDialog alermjcc = builtt.create();
                    alermjcc.show();

                }
            }else {
                Intent intent2 = new Intent(this, Vacaciones.class);

                intent2.putExtra("numtm", username);
                intent2.putExtra("aniver", aniv);
                intent2.putExtra("dzvk", dvak);
                intent2.putExtra("acu", acumu);
                intent2.putExtra("tdima", toditoma);
                intent2.putExtra("gerente", manager);
                startActivity(intent2);
            }
        }
    }
}

```

```

    }
} else if(textospiner.equals("PERMISO S/G")){

    Intent intent3 = new Intent(this, PermisosSin.class);
    intent3.putExtra("numtm", username);
    intent3.putExtra("gerente", manager);
    startActivity(intent3);

} else if(textospiner.equals("PERMISO C/G")){
    Intent intent4 = new Intent(this, PermisoCon.class);
    intent4.putExtra("numtm", username);
    intent4.putExtra("gerente", manager);
    startActivity(intent4);
}
else{

    Intent intent5 = new Intent(this, Estacionamiento.class);
    intent5.putExtra("numtm", username);
    intent5.putExtra("gerente", manager);
    startActivity(intent5);
}
break;

case R.id.btnsalida:
    Intent intent6 = new Intent(this, MainActivity.class);
    startActivity(intent6);
    finish();
    break;

default:

    break;
}
}

//Método para conectarse al Cerebro y descargar los datos principales del
trabajador
public class ObtenerWebService extends AsyncTask<String,Void,String>{

    private String ar;
    private String de;
    private String pu;
    private String ing;
    Date fe;
    @Override
    protected String doInBackground(String... params) {
        String cadena = params[0];
        URL url = null;
        String devuelve = "";
        pasar=1;

        try {
            url = new URL(cadena);
            HttpURLConnection connexion =
(HttpURLConnection)url.openConnection();
            connexion.setRequestProperty("Content-Type", "text/html");
            connexion.setRequestProperty("Host", "192.168.1.72");
            connexion.setRequestProperty("User-
Agent", "Mozilla/5.0(X11;Ubuntu;Linux i686;rv:49.0) Gecko/20100101 Firefox/49.0");
            connexion.setRequestMethod("GET");
            int respuesta = connexion.getResponseCode();
            StringBuilder result = new StringBuilder();

            if (respuesta == HttpURLConnection.HTTP_OK){

```

```

        InputStream in = new
BufferedInputStream(connection.getInputStream()); // preparo la cadena de
entrada

        BufferedReader reader = new BufferedReader(new
InputStreamReader(in));

        String line=null;
        while ((line = reader.readLine()) != null) {
            result.append(line);
break;
        }

        reader.close();

        JSONObject respuestaJSON = new
JSONObject(result.toString());
        JSONArray resultaJSON =
respuestaJSON.getJSONArray("empleado");

        JSONObject c = resultaJSON.getJSONObject(0);
        devuelve = c.getString("nombre");
        devuelve = devuelve + " " + c.getString("aprimero");
        devuelve = devuelve + " " + c.getString("apsegundo");
        de = c.getString("nombredepto");
        ar = c.getString("nombrearea");
        pu = c.getString("puesto");
        ing = c.getString("fechaingreso");
        aniv = c.getString("aniversario");
        dvak = c.getString("diasvaca");
        acumu=c.getString("acumulados");
        toditoma=c.getString("ttdvkto");
    }

    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

return devuelve;
}

@Override
protected void onPreExecute() {
    super.onPreExecute();
    pDialog = new ProgressDialog(ModuloFormatos.this);
    pDialog.setMessage("Loading tm details. Please wait...");
    pDialog.show();
    tnombre.setText("");
    tingresodate.setText("");
    tpuesto.setText("");
    tarea.setText("");
    tdepto.setText("");
}

@Override

```



```

protected void onPostExecute(String s) {
    super.onPostExecute(s);
    tnombre.setText(s);
    tpuesto.setText(pu);
    tdepto.setText(de);
    tarea.setText(ar);
    tingresodate.setText(ing);
    pDialog.cancel();
}

@Override
protected void onProgressUpdate(Void... values) {
    super.onProgressUpdate(values);
}

@Override
protected void onCancelled(String s) {
    super.onCancelled(s);
}

public void execute(String dirip, String s) {
}
}
}

```

## 10.22 PermisoCon.class

Permite hacer la solicitud de algún día o días con permiso con goce de sueldo.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.AlertDialog;
import android.app.DatePickerDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.BufferedInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.Calendar;

```

```

public class PermisoCon extends AppCompatActivity {

    public static final String NMTMS = "MIEQUI";
    TextView feiniv;
    TextView fefinal;
    EditText dizpedidos;
    EditText motivo;
    ImageButton btncalendar;
    ImageButton btacep;
    public int anio,mes,dia,totald,entdivk,ntmhil,senala,mes1,mes2;
    DatePickerDialog datePickerDialog;

    LigarWebServiceCon liowebconex;
    Button btnver;
    String dirip= "http://192.168.1.72/iniciarpHP/";
    String getuno = dirip + "panel/guarpermisocon.php";
    public String empleado,fstar,fend,jefe,razon;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_permiso_con);

        feiniv = (TextView) findViewById(R.id.fchincalpersgcon);
        fefinal = (TextView) findViewById(R.id.fchfincalpersgcon);
        motivo = (EditText) findViewById(R.id.edittxtperconreslpaldo);
        dizpedidos = (EditText) findViewById(R.id.edittexpersigcon);
        btacep = (ImageButton) findViewById(R.id.imgbtconfirpa);
        btnver = (Button) findViewById(R.id.btnidverregpercon);

        Intent intanciado = getIntent();
        Bundle extrainta = intanciado.getExtras();
        if(extrainta!=null) {
            empleado = extrainta.getString("numtm");
            ntmhil = Integer.valueOf(empleado);
            jefe=extrainta.getString("gerente");
            dizpedidos.setText("1");

        }else{
            Toast.makeText(PermisoCon.this,"No hay datos",
            Toast.LENGTH_LONG).show();
        }
        final Calendar cal = Calendar.getInstance();
        dia=cal.get(Calendar.DAY_OF_MONTH);
        mes=cal.get(Calendar.MONTH);
        anio=cal.get(Calendar.YEAR);
        ponerfechador();
        entdivk=Integer.valueOf(dizpedidos.getText().toString());

        showDialogOnButtonClickdos();

        metodoacep();
        metoregist();

    }
    //Método para que el botón registro pueda funcionar y muestre el registro de todos
    los permisos con goce de sueldo
    private void metoregist() {
        btnver.setOnClickListener(new View.OnClickListener() {

```

```

        @Override
        public void onClick(View v) {
            Intent verre = new Intent(PermisoCon.this, VerRegistrocon.class);
            verre.putExtra("tmnum", empleado);
            startActivity(verre);
        }
    });
}
//Método para el botón aceptar y este pueda enviar los datos a la base de datos
private void metodoacep() {
    btacep.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String monto= null;
            try {
                monto =
                URLEncoder.encode(String.valueOf(motivo.getText()), "UTF-8");
            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            }
            razon=motivo.getText().toString();
            senala=razon.length();
            if(entdivk>=1 && senala>5) {
                liowebconex = new LigarWebServiceCon();
                String dirconsultd = getuno + "?ntm=" + ntmhil + "&dzhave=" +
                entdivk + "&justi=" + monto + "&fini=" + fstar + "&ffin=" + fend;
                liowebconex.execute(dirconsultd);
                AlertDialog.Builder bldz = new
                AlertDialog.Builder(PermisoCon.this);
                bldz.setMessage("Los datos fueron enviados")
                    .setTitle("Ok")
                    .setCancelable(false)
                    .setNeutralButton("Ok", new
                DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
                which) {
                    }
                });
                AlertDialog delaf = bldz.create();
                delaf.show();
            }
            else{
                AlertDialog.Builder builx = new
                AlertDialog.Builder(PermisoCon.this);
                builx.setMessage("Faltan campos")
                    .setTitle("Error")
                    .setCancelable(false)
                    .setNeutralButton("OK", new
                DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int
                which) {
                    dialog.cancel();
                    }
                });
                AlertDialog alermj = builx.create();
                alermj.show();
            }
        }
    }
}

```

```

    });
}
//Método que muestra el calendario y se pueda asignar la fecha de inicio y fin.
private void showDialogOnButtonClickdos() {
    btncalendar = (ImageButton) findViewById(R.id.btncalpersigcon);
    btncalendar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            datePickerDialog = new DatePickerDialog(PermisoCon.this, new
DatePickerDialog.OnDateSetListener() {
                @Override
                public void onDateSet(DatePicker view, int year, int month,
int dayOfMonth) {

                    month=month+1;
                    feiniv.setText(dayOfMonth+"/"+month+"/"+year);
                    fstar=year+"/"+month+"/"+dayOfMonth;
                    final Calendar ck = Calendar.getInstance();
                    entdivk=Integer.valueOf(dizpedidos.getText().toString());
                    senala=1;
                    totald=entdivk;

                    ck.set(year, month, dayOfMonth);
                    mes1=month;
                    if(totald==1){

                        fefinal.setText(dayOfMonth + "/" + month + "/" +
year);

                        fend=year+"/"+month+"/"+dayOfMonth;

                    }
                    else {
                        Log.d("mymesfin", "ixz dia
antes="+ck.get(Calendar.DAY_OF_MONTH));
                        totald=totald-1;
                        ck.add(Calendar.DAY_OF_MONTH, totald);
                        mes2=ck.get(Calendar.MONTH);
                        Log.d("mymesfin", "ixz mes 1 antes="+mes1);
                        Log.d("mymesfin", "ixz mes 2 antes="+mes2);
                        if(mes1==mes2) {
                            Log.d("mymesfin", "ixz mes==");
                            Log.d("mymesfin", "ixz dia
despues="+ck.get(Calendar.DAY_OF_MONTH));
                            fefinal.setText(ck.get(Calendar.DAY_OF_MONTH) +
"/" + ck.get(Calendar.MONTH) + "/" + ck.get(Calendar.YEAR));
                            fend = ck.get(Calendar.YEAR) + "/" +
ck.get(Calendar.MONTH) + "/" + ck.get(Calendar.DAY_OF_MONTH);
                            if(mes1==0 && mes2==0){
                                ck.add(Calendar.YEAR, -1);
                                fefinal.setText(ck.get(Calendar.DAY_OF_MONTH)
+ "/" + 12 + "/" + ck.get(Calendar.YEAR));
                                fend = ck.get(Calendar.YEAR) + "/" + 12 + "/"
+ ck.get(Calendar.DAY_OF_MONTH);
                            }

                        }
                    }
                    else{

                        if(mes2==0 && mes1!=mes2){
                            Log.d("mymesfin", "ixz mes=0");
                            Log.d("mymesfin", "ixz dia
despues="+ck.get(Calendar.DAY_OF_MONTH));

```

```

        ck.add(Calendar.YEAR, -1);
        fefinal.setText(ck.get(Calendar.DAY_OF_MONTH)
+ "/" + 12 + "/" + ck.get(Calendar.YEAR));
        fend = ck.get(Calendar.YEAR) + "/" + 12 + "/"
+ ck.get(Calendar.DAY_OF_MONTH);
    }
    else {
        Log.d("mymesfin", "ixz mes =!" + mes1);
        Log.d("mymesfin", "ixz dia
despues=" + ck.get(Calendar.DAY_OF_MONTH));

        fefinal.setText(ck.get(Calendar.DAY_OF_MONTH)
+ "/" + ck.get(Calendar.MONTH) + "/" + ck.get(Calendar.YEAR));
        fend = ck.get(Calendar.YEAR) + "/" +
ck.get(Calendar.MONTH) + "/" + ck.get(Calendar.DAY_OF_MONTH);

    }
}

Log.d("mymesfin", "mes 2 antes" + mes2);

Log.d("myinfo", "Checar fecha" + fend);
Log.d("mymesinici", "mes 1" + mes1);
Log.d("mymesfin", "mes 2 des" + mes2);
}
}, anio, mes, dia);

entdivk = Integer.valueOf(dizpedidos.getText().toString());
if (entdivk > 0 && entdivk < 30) {
    datePickerDialog.show();
}
else {
    AlertDialog.Builder builder = new
AlertDialog.Builder(PermisoCon.this);
    builder.setMessage("Solicite otro folio")
        .setTitle("Error")
        .setCancelable(false)
        .setNeutralButton("OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int
which) {
            dialog.cancel();
        }
    });
    AlertDialog alert = builder.create();
    alert.show();
}

}
});
}

//Metodo para actualizar la fecha actual
private void ponerfechador() {
    feiniv.setText(dia + "/" + (mes + 1) + "/" + anio);
    fstar = anio + "/" + (mes + 1) + "/" + dia;
}

//Métod para conectarse a la base de datos y se puedan enviar los datos de
solicitud
private class LigarWebServiceCon extends AsyncTask<String, Void, Void> {

    @Override

```

```

protected Void doInBackground(String... params) {
    String cadena = params[0];
    URL url = null;

    try {
        url = new URL(cadena);
        HttpURLConnection connexion =
(HttpURLConnection)url.openConnection();
        connexion.setRequestProperty("Content-Type", "text/html");
        connexion.setRequestProperty("Host", "192.168.1.72");
        connexion.setRequestProperty("User-
Agent", "Mozilla/5.0(X11;Ubuntu;Linux i686;rv:49.0) Gecko/20100101 Firefox/49.0");
        connexion.setRequestMethod("GET");
        int respuesta = connexion.getResponseCode();
        StringBuilder result = new StringBuilder();

        if (respuesta == HttpURLConnection.HTTP_OK) {

            InputStream in = new
BufferedInputStream(connexion.getInputStream());
            BufferedReader reader = new BufferedReader(new
InputStreamReader(in));

            String line=null;
            while ((line = reader.readLine()) != null) {
                result.append(line);
break;
            }

            reader.close();

        }
    } catch (IOException e) {
        e.printStackTrace();
    }

    return null;
}
}
}

```

## 10.23 PermisoSin.class

Permite hacer la solicitud de algún día o días con permiso sin goce de sueldo.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.AlertDialog;
import android.app.DatePickerDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;

```

```

import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.ProtocolException;
import java.net.URL;
import java.util.Calendar;

public class PermisosSin extends AppCompatActivity {

    public static final String NMTMS = "MIEQUI";
    TextView feiniv;
    TextView fefinal;
    EditText dizpedidos;
    ImageButton btncalendar;
    ImageButton btcancelar;
    ImageButton btacep;
    public int anio,mes,dia,totald,entdivk,ntmhil,senala,mes1,mes2;
    DatePickerDialog datePickerDialog;
    LigarWebService liowebconex;
    Button btnver;
    String dirip= "http://192.168.1.72/iniciarpHP/";
    String getuno = dirip + "panel/guarpermisosin.php";
    String empleado,fstar,fend,jefe;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_permisos_sin);

        feiniv = (TextView) findViewById(R.id.fchincicalpersg);
        fefinal = (TextView) findViewById(R.id.fchfincalpersg);

        dizpedidos = (EditText) findViewById(R.id.edittexpersig);
        btacep = (ImageButton) findViewById(R.id.imgbtconfirpa);
        btnver = (Button) findViewById(R.id.btnidverregpersin);

        Intent intanciado = getIntent();
        Bundle extrainta = intanciado.getExtras();
        if(extrainta!=null) {
            empleado = extrainta.getString("numtm");
            ntmhil = Integer.valueOf(empleado);
            jefe=extrainta.getString("gerente");
            dizpedidos.setText("1");
        }else{
            Toast.makeText(PermisosSin.this,"No hay datos",
Toast.LENGTH_LONG).show();
        }
        final Calendar cal = Calendar.getInstance();
        dia=cal.get(Calendar.DAY_OF_MONTH);
        mes=cal.get(Calendar.MONTH);
        anio=cal.get(Calendar.YEAR);
        ponerfechador();
    }
}

```

```

        entdivk=Integer.valueOf(dizpedidos.getText().toString());

        showDialogOnButtonClickdos();

        metodoacep();
        metoregist();
    }
    //Método para ver el registro de todos los permisos sin goce de sueldo
    solicitados por el trabajador
    private void metoregist() {
        btnver.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent verre = new
                Intent(PermisosSin.this,VerRegistropsin.class);
                verre.putExtra("tmnum",empleado);
                startActivity(verre);

            }
        });

    }

    //Método para cancelar el envio de datos y regresar a la pantalla anterior
    private void metocancel() {
        btcancelar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent pasar = new Intent(PermisosSin.this,ModuloFormatos.class);
                startActivity(pasar);
                finish();

            }
        });

    }

    //Método para aceptar en envio de datos a la base de datos
    private void metodoacep() {
        btacep.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                if(entdivk>=1) {
                    liowebconex = new LigarWebService();
                    String dirconsultd = getuno + "?ntm=" + ntmhil + "&dzhave=" +
entdivk + "&fini=" + fstar + "&ffin=" + fend;
                    liowebconex.execute(dirconsultd);
                    AlertDialog.Builder bldz = new
                    AlertDialog.Builder(PermisosSin.this);
                    bldz.setMessage("Los datos fueron enviados")
                    .setTitle("Ok")
                    .setCancelable(false)
                    .setNeutralButton("Ok", new
                    DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int
                    which) {

                            }
                        });
                    AlertDialog delaf = bldz.create();

```



```

        delaf.show();
    }
    else{
        AlertDialog.Builder builx = new
AlertDialog.Builder(PermisosSin.this);
        builx.setMessage("Faltan campos")
            .setTitle("Error")
            .setCancelable(false)
            .setNeutralButton("OK", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int
which) {
                    dialog.cancel();
                }
            });
        AlertDialog alermj = builx.create();
        alermj.show();
    }
});
}
//Método para mostrar el calendario y seleccionar la fecha de inicio y fin
private void showDialogOnButtonClickdos() {
    btncalendar = (ImageButton) findViewById(R.id.btncalpersig);
    btncalendar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            datePickerDialog = new DatePickerDialog(PermisosSin.this, new
DatePickerDialog.OnDateSetListener() {
                @Override
                public void onDateSet(DatePicker view, int year, int month,
int dayOfMonth) {
                    month=month+1;
                    feiniv.setText(dayOfMonth+"/"+month+"/"+year);
                    fstar=year+"/"+month+"/"+dayOfMonth;
                    final Calendar ck = Calendar.getInstance();
                    entdivk=Integer.valueOf(dizpedidos.getText().toString());
                    senala=1;
                    totald=entdivk;

                    ck.set(year, month, dayOfMonth);
                    mes1=month;
                    if(totald==1){
                        fefinal.setText(dayOfMonth + "/" + month + "/" +
year);
                        fend=year+"/"+month+"/"+dayOfMonth;
                    }
                    else {
                        Log.d("mymesfin", "ixz dia
antes="+ck.get(Calendar.DAY_OF_MONTH));
                        totald=totald-1;
                        ck.add(Calendar.DAY_OF_MONTH, totald);
                        mes2=ck.get(Calendar.MONTH);
                        Log.d("mymesfin", "ixz mes 1 antes="+mes1);
                        Log.d("mymesfin", "ixz mes 2 antes="+mes2);
                        if(mes1==mes2) {
                            Log.d("mymesfin", "ixz mes==");
                            Log.d("mymesfin", "ixz dia
despues="+ck.get(Calendar.DAY_OF_MONTH));

```



```

        }
        });
        AlertDialog alert = builder.create();
        alert.show();
    }

    });
}

//Método para actualizar la fecha actual
private void ponerfechador() {
    feiniv.setText(dia+"/"+(mes+1)+"/"+anio);
    fstar=anio+"/"+(mes+1)+"/"+dia;
}

//Método para enviar los datos a Cerebro
private class LigarWebService extends AsyncTask<String,Void,Void> {
    @Override
    protected Void doInBackground(String... params) {

        String cadena = params[0];
        URL url = null;

        try {
            url = new URL(cadena);
            HttpURLConnection connexion =
(HttpURLConnection)url.openConnection();
            connexion.setRequestProperty("Content-Type","text/html");
            connexion.setRequestProperty("Host","192.168.1.72");
            connexion.setRequestProperty("User-
Agent","Mozilla/5.0(X11;Ubuntu;Linux i686;rv:49.0) Gecko/20100101 Firefox/49.0");
            connexion.setRequestMethod("GET");
            int respuesta = connexion.getResponseCode();
            StringBuilder result = new StringBuilder();

            if (respuesta == HttpURLConnection.HTTP_OK) {

                InputStream in = new
BufferedInputStream(connexion.getInputStream());

                BufferedReader reader = new BufferedReader(new
InputStreamReader(in));

                String line=null;
                while ((line = reader.readLine()) != null) {
                    result.append(line);
break;
                }

                reader.close();

            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        return null;
    }
}
}
}

```

## 10.24 ScannerFinal.class

Permite abrir la cámara para poder escanear el Código QR.

```
package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.zxing.integration.android.IntentIntegrator;
import com.google.zxing.integration.android.IntentResult;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class ScannerFinal extends AppCompatActivity {

    TextView nomb, entr, secreto, contdo, fortoz;
    final Context contx=this;
    public static final String FOVLIA = "FOLIAS";
    public static final String NMVJT = "USERMNGT";
    String dirip= "http://192.168.1.72/iniciarphp/";
    String getuno = dirip + "panel/guadaracceptarvaca.php";
    String getdos = dirip + "panel/guadaracceptarpsing.php";
    String gettres = dirip + "panel/guadaracceptarpcong.php";
    String getcuatro = dirip + "panel/guadaracceptarestaciona.php";
    private ProgressDialog msjito;
    public String gtnusr;
    public String folkij;
    public String qrpas;
    public String auxbanderinz;
    String dirconsultd;
    EnchufeWebServ hilowebinter;

    AlertDialog.Builder aletrbuilder;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scanner_final);

        nomb=(TextView) findViewById(R.id.mngscannerfina);
        entr=(TextView) findViewById(R.id.clavscannerfinal);
        secreto=(TextView) findViewById(R.id.qrsusercanerfinal);
        contdo=(TextView) findViewById(R.id.contenisnanerfinal);
    }
}
```

```

fortoz=(TextView) findViewById(R.id.formatscanerfinal);

Intent intent = getIntent();
gtnusr = intent.getStringExtra(FragmentAceptar.USRMG);
final String clpass = intent.getStringExtra(FragmentAceptar.CVGT);
qrpas = intent.getStringExtra(FragmentAceptar.PSSGTQR);
folkij = intent.getStringExtra(FragmentAceptar.FXIO);
auxbanderinz = intent.getStringExtra(FragmentAceptar.FLAGMIX);

LayoutInflater li=LayoutInflater.from(contx);
View promptview=li.inflate(R.layout.menspassqr, null);
aletrbuilder = new AlertDialog.Builder(contx);
aletrbuilder.setView(promptview);
final EditText userinput=(EditText)
promptview.findViewById(R.id.edtxmespaasinput);
aletrbuilder.setCancelable(false)
    .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        if(userinput.getText().toString().equals(clpass)){

Toast.makeText(getApplicationContext(), "Exito", Toast.LENGTH_LONG).show();
        hilowebinter = new EnchufeWebServ();
        if(auxbanderinz.equals("vacaciones")) {
            dirconsultd = getuno + "?getn=" + gtnusr +
"&fofz=" + folkij;
        }else if(auxbanderinz.equals("percongo")){
            dirconsultd = gettres + "?getn=" + gtnusr +
"&fofz=" + folkij;
        }
        else if(auxbanderinz.equals("persingo")){
            dirconsultd = getdos + "?getn=" + gtnusr +
"&fofz=" + folkij;
        }else{
            dirconsultd = getcuatro + "?getn=" + gtnusr +
"&fofz=" + folkij;
        }
        hilowebinter.execute(dirconsultd);
        Intent regresar = new
Intent(getApplicationContext(),ModuloAutorizar.class);
        regresar.putExtra(NMVJT,gtnusr);
        startActivity(regresar);
        finish();
    }
    else{

Toast.makeText(getApplicationContext(), "Error", Toast.LENGTH_LONG).show();
        Intent regresar = new
Intent(getApplicationContext(),ContenedorFragmentMg.class);
        regresar.putExtra(FOVLIA,folkij);
        regresar.putExtra(NMVJT,gtnusr);
        startActivity(regresar);
        finish();
    }
    }
    }).setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
//Método para realizar la apertura de la cámara y se pueda scanear el Código QR
    @Override
    public void onClick(DialogInterface dialog, int which) {

```

```

        Intent regresar = new
Intent(getApplicationContext(), ContenedorFragmentMg.class);
        regresar.putExtra(FOVLIA, folkij);
        regresar.putExtra(NMVJT, gtnusr);
        startActivity(regresar);
        finish();
    }
});

nomb.setText(gtnusr);
entr.setText("TG"+clpass);
secreto.setText("TG"+qrpas);

IntentIntegrator inntengra = new IntentIntegrator(ScannerFinal.this);
inntengra.setDesiredBarcodeFormats(IntentIntegrator.QR_CODE_TYPES);
inntengra.setPrompt("Scanear");
inntengra.setCameraId(0);
inntengra.initiateScan();

}
//Método que valida la firma electrónica avanzada correspondiente a cada Gerente
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    final IntentResult result =
IntentIntegrator.parseActivityResult(requestCode, resultCode, data);

    if(result != null){
        if(result.getContents()==null){
            Toast.makeText(getApplicationContext(), "You cancelled the
scanning", Toast.LENGTH_LONG).show();
        }
        else {

            if(qrpas.equals(result.getContents())) {
                Toast.makeText(getApplicationContext(), "Firma QR
valida", Toast.LENGTH_LONG).show();
                contdo.setText("QR:" + result.getContents());
                fortoz.setText("QR:" + result.getFormatName());
                AlertDialog alerdalog = aletrbuilder.create();
                alerdalog.show();
            }
            else{
                Toast.makeText(getApplicationContext(), "Firma QR no
valida", Toast.LENGTH_LONG).show();
                Intent regresar = new
Intent(getApplicationContext(), ContenedorFragmentMg.class);
                regresar.putExtra(FOVLIA, folkij);
                regresar.putExtra(NMVJT, gtnusr);
                startActivity(regresar);
                finish();
            }
        }
    }
    else {
        super.onActivityResult(requestCode, resultCode, data);
    }
}
}

```

```

//Método para conectarse al servidor y poder enviar los datos
private class EnchufeWebServ extends AsyncTask<String,Void,Void> {
    @Override
    protected Void doInBackground(String... params) {
        String cadena = params[0];
        URL url = null;

        try {
            url = new URL(cadena);
            HttpURLConnection connexion =
(HttpURLConnection)url.openConnection();
            connexion.setRequestProperty("Content-Type","text/html");
            connexion.setRequestProperty("Host","192.168.1.72");
            connexion.setRequestProperty("User-
Agent","Mozilla/5.0(X11;Ubuntu;Linux i686;rv:49.0) Gecko/20100101 Firefox/49.0");
            connexion.setRequestMethod("GET");
            int respuesta = connexion.getResponseCode();
            StringBuilder result = new StringBuilder();

            if (respuesta == HttpURLConnection.HTTP_OK) {

                InputStream in = new
BufferedInputStream(connexion.getInputStream()); // preparo la cadena de
entrada

                BufferedReader reader = new BufferedReader(new
InputStreamReader(in));

                String line=null;
                while ((line = reader.readLine()) != null) {
                    result.append(line);
                    break;
                }

                reader.close();

            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        msjito = new ProgressDialog(ScannerFinal.this);
        msjito.setMessage("Sending data...");
        msjito.show();
    }
}
}

```

## 10.25 Vacaciones.class

Permite hacer la solicitud de algún día o días de vacaciones.

```
package roberto.espejel.app.proyecto.docpt.rulaitas.ian;
```

```

import android.app.AlertDialog;
import android.app.DatePickerDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.Calendar;

/**
 * Created by cocodrilito on 22/05/2017.
 */

public class Vacaciones extends AppCompatActivity {

    TextView aniersa;
    TextView dizvaka;
    TextView feiniv;
    TextView fefinal;
    TextView tvdiacumulados;
    EditText dizpedidos;
    ImageButton btcancelar;
    ImageButton btacep;
    Button btnver;

    ImageButton btncalendar;
    public int anio,mes,dia,totald,entdivk,acumula,tdita,ntmhil,senal,mes1,mes2;
    DatePickerDialog datePickerDialog;
    ConexWebService hilowebconex;
    String dirip= "http://192.168.1.72/iniciarphp/";
    String getuno = dirip + "panel/guaregisvaca.php";
    String empleado,fstar,fend,jefe;
    public static final String NMTYM = "MYTMSR";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.select_formatos);
        aniersa=(TextView)findViewById(R.id.tvaniver);
        dizvaka = (TextView)findViewById(R.id.dyaz);
        feiniv = (TextView)findViewById(R.id.fchincalvaka);
        fefinal = (TextView)findViewById(R.id.fchfincalvaka);
        tvdiacumulados=(TextView)findViewById(R.id.tvdizacumulados);
        dizpedidos = (EditText)findViewById(R.id.editdiasxtomar);
    }

```



```

btacep = (ImageButton)findViewById(R.id.imgbtconfirm);
btnver = (Button)findViewById(R.id.bttverregis);

Intent intanciado = getIntent();
Bundle extrainta = intanciado.getExtras();

if(extrainta!=null){

    empleado= extrainta.getString("numtm");
    ntmhil=Integer.valueOf(empleado);
    String cump = extrainta.getString("aniver");
    aniersa.setText(cump);
    String dias = extrainta.getString("dzvk");
    dizvaka.setText(dias);
    String acumulativo=extrainta.getString("acu");
    jefe=extrainta.getString("gerente");
    acumula=Integer.valueOf(acumulativo);
    String diasusados=extrainta.getString("tdima");
    tdita=Integer.valueOf(diasusados);
    acumula=acumula-tdita;
    tvdiacumulados.setText(String.valueOf(acumula));
    dizpedidos.setText("1");
    entdivk=1;
    senal=0;

}
else{
    Toast.makeText(Vacaciones.this, "No hay datos",
    Toast.LENGTH_LONG).show();
}

final Calendar cal = Calendar.getInstance();
dia=cal.get(Calendar.DAY_OF_MONTH);
mes=cal.get(Calendar.MONTH);
anio=cal.get(Calendar.YEAR);
ponerfechador();
entdivk=Integer.valueOf(dizpedidos.getText().toString());

    showDialogOnClick();

    metodoacep();
    metoregist();
}
//Método para ver los registros de las vacaciones solicitadas
private void metoregist() {
    btnver.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent verre = new
Intent(Vacaciones.this,VerRegistrovacaciones.class);
            verre.putExtra("tmnum",empleado);
            startActivity(verre);
        }
    });
}

//Método para enviar los datos aceptados a Cerebro
private void metodoacep() {
    btacep.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(senal>=1) {
                hilowebconex = new ConexWebService();
            }
        }
    });
}

```

```

        String dirconsultd = getuno + "?ntm=" + ntmhil + "&anivez=" +
aniersa.getText() + "&dzhave=" + entdivk + "&fini=" + fstar + "&ffin=" + fend;
        hilowebconex.execute(dirconsultd);
        AlertDialog.Builder bldz = new
AlertDialog.Builder(Vacaciones.this);
        bldz.setMessage("Los datos fueron enviados")
            .setTitle("Ok")
            .setCancelable(false)
            .setNeutralButton("Ok", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int
which) {
                    acumula=acumula-entdivk;
                    tvdiacumulados.setText(String.valueOf(acumula));
                }
            });
        AlertDialog delaf = bldz.create();
        delaf.show();
    }
    else{
        AlertDialog.Builder builx = new
AlertDialog.Builder(Vacaciones.this);
        builx.setMessage("Faltan campos")
            .setTitle("Error")
            .setCancelable(false)
            .setNeutralButton("OK", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int
which) {
                    dialog.cancel();
                }
            });
        AlertDialog alermj = builx.create();
        alermj.show();
    }
}
});
}
//Método para actualizar la fecha actual
private void ponerfechador() {
    feiniv.setText(dia+"/"+(mes+1)+"/"+anio);
    fstar=anio+"/"+(mes+1)+"/"+dia;
}

//Método para mostrar el calendario y poder hacer el calculo de los días
correspondientes a tomar considerando el descanso.
public void showDialogOnButtonClick(){
    btncalendar =(ImageButton) findViewById(R.id.btncalvaka);
    btncalendar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // showDialog(DIALOG_ID);

            datePickerDialog = new DatePickerDialog(Vacaciones.this, new
DatePickerDialog.OnDateSetListener() {
                @Override
                public void onDateSet(DatePicker view, int year, int month,
int dayOfMonth) {
                    month=month+1;

```

```

        feiniv.setText(dayOfMonth+"/"+month+"/"+year);
        fstar=year+"/"+month+"/"+dayOfMonth;
        final Calendar ck = Calendar.getInstance();

entdivk=Integer.valueOf(dizpedidos.getText().toString());
        if(entdivk==1){
            Toast.makeText(Vacaciones.this,"0 descanso",
Toast.LENGTH_SHORT).show();
                totald=entdivk;
            }
        else if(entdivk>1 && entdivk<5){
            Toast.makeText(Vacaciones.this,"0 descanso",
Toast.LENGTH_SHORT).show();
                totald=entdivk;
            }
        else if(entdivk>=5 && entdivk<7){
            totald=entdivk+1;
            Toast.makeText(Vacaciones.this,"1 descansos",
Toast.LENGTH_SHORT).show();
                }
        else if(entdivk>=7 && entdivk<13){
            totald=entdivk+2;
            Toast.makeText(Vacaciones.this,"2 descansos",
Toast.LENGTH_SHORT).show();
                }
        else if(entdivk>=13 && entdivk<19){
            totald=entdivk+3;
            Toast.makeText(Vacaciones.this,"3 descansos",
Toast.LENGTH_SHORT).show();
                }
        else if(entdivk>=19 && entdivk<25){
            totald=entdivk+4;
            Toast.makeText(Vacaciones.this,"4 descansos",
Toast.LENGTH_SHORT).show();
                }
        else{
            totald=entdivk+5;
            Toast.makeText(Vacaciones.this,"5 descansos",
Toast.LENGTH_SHORT).show();
                }
        senala=1;

        ck.set(year, month, dayOfMonth);
        mes1=ck.get(Calendar.MONTH);
        if(totald==1){
            fefinal.setText(dayOfMonth + "/" + month + "/" +
year);
                fend=year+"/"+month+"/"+dayOfMonth;
            }
        else {
            Log.d("mymesfin","ixz dia
antes="+ck.get(Calendar.DAY_OF_MONTH));
                totald=totald-1;
                ck.add(Calendar.DAY_OF_MONTH, totald);
                mes2=ck.get(Calendar.MONTH);
                Log.d("mymesfin","ixz mes 1 antes="+mes1);
                Log.d("mymesfin","ixz mes 2 antes="+mes2);
                if(mes1==mes2) {
                    Log.d("mymesfin","ixz mes==");
                    Log.d("mymesfin","ixz dia
despues="+ck.get(Calendar.DAY_OF_MONTH));

```

```

fefinal.setText(ck.get(Calendar.DAY_OF_MONTH) + "/" + ck.get(Calendar.MONTH) +
"/" + ck.get(Calendar.YEAR));
                                fend = ck.get(Calendar.YEAR) + "/" +
ck.get(Calendar.MONTH) + "/" + ck.get(Calendar.DAY_OF_MONTH);
                                if(mes1==0 && mes2==0){
                                    ck.add(Calendar.YEAR,-1);

fefinal.setText(ck.get(Calendar.DAY_OF_MONTH) + "/" + 12 + "/" +
ck.get(Calendar.YEAR));
                                fend = ck.get(Calendar.YEAR) +
"/" + 12 + "/" + ck.get(Calendar.DAY_OF_MONTH);
                                    }
                                }
                                else{

                                    if(mes2==0 && mes1!=mes2){
                                        Log.d("mymesfin","ixz mes=0");
                                        Log.d("mymesfin","ixz dia

despues="+ck.get(Calendar.DAY_OF_MONTH));

                                        ck.add(Calendar.YEAR,-1);

fefinal.setText(ck.get(Calendar.DAY_OF_MONTH) + "/" + 12 + "/" +
ck.get(Calendar.YEAR));
                                fend = ck.get(Calendar.YEAR) +
"/" + 12 + "/" + ck.get(Calendar.DAY_OF_MONTH);
                                    }
                                    else {
                                        Log.d("mymesfin","ixz mes

="!"+mes1);
                                        Log.d("mymesfin","ixz dia

despues="+ck.get(Calendar.DAY_OF_MONTH));

fefinal.setText(ck.get(Calendar.DAY_OF_MONTH) + "/" + ck.get(Calendar.MONTH) +
"/" + ck.get(Calendar.YEAR));
                                fend = ck.get(Calendar.YEAR) +
"/" + ck.get(Calendar.MONTH) + "/" + ck.get(Calendar.DAY_OF_MONTH);
                                    }
                                }
                                }

                                Log.d("myinfo","ixz Checar fecha final="+fend);
                                Log.d("mymesinici","ixz mes 1 des="+mes1);
                                Log.d("mymesfin","ixz mes 2 des="+mes2);
                            }
                        }, anio, mes, dia);

entdivk=Integer.valueOf(dizpedidos.getText().toString());
if(entdivk>0 && entdivk<=acumula && entdivk<30) {
    datePickerDialog.show();
}
else{
    AlertDialog.Builder builder = new
AlertDialog.Builder(Vacaciones.this);
    builder.setMessage("No hay mas dias acumulados")
        .setTitle("Error")
        .setCancelable(false)
        .setNeutralButton("OK", new
DialogInterface.OnClickListener() {
            @Override

```

```

        public void onClick(DialogInterface dialog, int
which) {
        dialog.cancel();
    }
    });
    AlertDialog alert = builder.create();
    alert.show();
}
});
}
//Método para la conexión a Cerebro y se puedan guardar los datos a la base de
datos
private class ConexWebService extends AsyncTask<String,Void,Void>{
    @Override
    protected Void doInBackground(String... params) {
        String cadena = params[0];
        URL url = null;

        try {
            url = new URL(cadena);
            HttpURLConnection connexion =
(HttpURLConnection)url.openConnection();
            connexion.setRequestProperty("Content-Type", "text/html");
            connexion.setRequestProperty("Host", "192.168.1.72");
            connexion.setRequestProperty("User-
Agent", "Mozilla/5.0(X11;Ubuntu;Linux i686;rv:49.0) Gecko/20100101 Firefox/49.0");
            connexion.setRequestMethod("GET");
            int respuesta = connexion.getResponseCode();
            StringBuilder result = new StringBuilder();

            if (respuesta == HttpURLConnection.HTTP_OK) {

                InputStream in = new
BufferedInputStream(connexion.getInputStream());

                BufferedReader reader = new BufferedReader(new
InputStreamReader(in));

                String line=null;
                while ((line = reader.readLine()) != null) {
                    result.append(line) break;
                }

                reader.close();

            }
        } catch (IOException e) {
            e.printStackTrace();
        }

        return null;
    }
}
}
}

```

## 10.26 Vacantesz.class

Muestra las vacantes disponibles en algún departamento.

```
package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.app.ProgressDialog;
import android.content.Context;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ListView;
import android.widget.TextView;

import com.loopj.android.http.AsyncHttpClient;
import com.loopj.android.http.AsyncHttpResponseHandler;

import org.json.JSONArray;
import org.json.JSONException;

import java.util.ArrayList;

import cz.msebera.android.httpclient.Header;

public class Vacantesz extends AppCompatActivity {

    ListView listvak;

    ArrayList puesto = new ArrayList();
    ArrayList lugares = new ArrayList();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_vacantesz);
        listvak = (ListView) findViewById(R.id.listvakax);
        cargardata();
    }
    //Método para conectarse al servidor y se puedan descargar dichos datos
    private void cargardata() {
        puesto.clear();
        lugares.clear();

        final ProgressDialog mensajizz = new ProgressDialog(Vacantesz.this);
        mensajizz.setMessage("Cargando las vacantes...");
        mensajizz.show();

        AsyncHttpClient clientvx = new AsyncHttpClient();

        clientvx.get("http://192.168.1.72/iniciarphtml/panel/codjsonvaca.php", new
        AsyncHttpResponseHandler() {
            @Override
            public void onSuccess(int statusCode, Header[] headers, byte[]
            responseBody) {
                if(statusCode == 200){
                    mensajizz.dismiss();
                    try {
                        JSONArray jasonarrayvaca = new JSONArray(new
                        String(responseBody));
                        for (int i=0;i<jasonarrayvaca.length();i++){
```

```

lugares.add(jasonarrayvaca.getJSONObject(i).getString("lugares"));
puesto.add(jasonarrayvaca.getJSONObject(i).getString("puesto"));
    }
    lisvak.setAdapter(new
ListViewAdapter(getApplicationContext()));
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

@Override
public void onFailure(int statusCode, Header[] headers, byte[]
responseBody, Throwable error) {

}
});
}

//Método para poder visualizar los datos en forma de lista
private class ListViewAdapter extends BaseAdapter {
    Context context;
    LayoutInflater inflater;

    public ListViewAdapter(Context applicationContext) {
        this.context=applicationContext;

inflater=(LayoutInflater) context.getSystemService(LAYOUT_INFLATER_SERVICE);
    }

    @Override
    public int getCount() {
        return puesto.size();
    }

    @Override
    public Object getItem(int position) {
        return position;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        TextView ps;
        TextView lug;
        ViewGroup viewgroup=
(ViewGroup) inflater.inflate(R.layout.activity_vacantes,null);

        ps= (TextView) viewgroup.findViewById(R.id.txtpuesto);
        lug=(TextView) viewgroup.findViewById(R.id.txtlugares);

        ps.setText(puesto.get(position).toString());
        lug.setText(lugares.get(position).toString());

        return viewgroup;
    }
}
}

```

## 10.27 VerRegistroestacionamiento.class

Permite visualizar en forma de lista, el historial de todas las solicitudes de estacionamiento hechas por el trabajador.

```
package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.HashMap;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.ClientProtocolException;
import cz.msebera.android.httpclient.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.params.BasicHttpParams;

public class VerRegistroestacionamiento extends AppCompatActivity {

    String myJSON;

    private static final String TAG_RESULTS="registroparking";
    private static final String TAG_ID = "idestacionamiento";
    private static final String TAG_REGISDIA="numdia";
    private static final String TAG_NAME = "fechacajon";
    private static final String TAG_ADD ="fechadesocajon";
    private static final String TAG_COMEN="comentaestacion";

    JSONArray peoples = null;

    ArrayList<HashMap<String, String>> personList;

    ListView listaregistro;
    ArrayList<String> dizx = new ArrayList();
    ArrayList<String> comntx = new ArrayList();
    ArrayList<String> fend = new ArrayList();
    String numemple;

    @Override
```



```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ver_registroestacionamiento);
    listregistro = (ListView) findViewById(R.id.listregistacionamneto);
    Intent intanciax = getIntent();
    Bundle extravar = intanciax.getExtras();

    if(extravar!=null) {

        numemple = extravar.getString("tmnum");

    }
    else{
        Toast.makeText(VerRegistroestacionamiento.this,"No hay datos",
        Toast.LENGTH_LONG).show();
    }

    personList = new ArrayList<HashMap<String,String>>();

    metodobusqpsin();

    listregistro.setOnItemClickListener(new
    AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
        position, long id) {

            Intent intent = new
            Intent(VerRegistroestacionamiento.this,DetalleVacaciones.class);
            intent.putExtra("dikx",dizx.get(position).toString());

            intent.putExtra("comfrt",comntx.get(position).toString());

            startActivity(intent);

        }
    });
}
//Método para buscar todo el hitorial del trabajador con relación a las
peticiones de estacionamiento
private void metodobusqpsin() {
    class GetDataJSON extends AsyncTask<String, Void, String> {

        @Override
        protected String doInBackground(String... params) {
            DefaultHttpClient httpclient = new DefaultHttpClient(new
            BasicHttpParams());
            HttpPost httppost = new
            HttpPost("http://192.168.1.72/iniciarpHP/panel/codjsonregistacionamiento.php?cod
            =" + numemple);

            // Depends on your web service
            httppost.setHeader("Content-type", "application/json");

            InputStream inputStream = null;
            String result = null;
            try {
                HttpResponse response = httpclient.execute(httppost);
                HttpEntity entity = response.getEntity();

                inputStream = entity.getContent();
                // json is UTF-8 by default

```

```

        BufferedReader reader = new BufferedReader(new
InputStreamReader(inputStream, "UTF-8"), 8);
        StringBuilder sb = new StringBuilder();

        String line = null;
        while ((line = reader.readLine()) != null) {
            sb.append(line + "\n");
        }
        result = sb.toString();

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        try {
            if (inputStream != null) inputStream.close();
        } catch (Exception squish) {
        }
    }
    }
    return result;
}
protected void onPostExecute(String result){
    myJSON=result;
    showListrps();
}
}
GetDataJSON g = new GetDataJSON();
g.execute();
}
//Método para desplegar en forma de lista los datos obtenidos
protected void showListrps() {
    try {
        JSONObject jsonObj = new JSONObject(myJSON);
        peoples = jsonObj.getJSONArray(TAG_RESULTS);

        for(int i=0;i<peoples.length();i++){
            JSONObject c = peoples.getJSONObject(i);
            String id = c.getString(TAG_ID);
            String name = c.getString(TAG_NAME);
            String address = c.getString(TAG_ADD);
            dixx.add(c.getString(TAG_REGISDIA));
            comntx.add(c.getString(TAG_COMEN));

            HashMap<String,String> persons = new HashMap<String,String>();

            persons.put(TAG_ID, id);
            persons.put(TAG_NAME, name);
            persons.put(TAG_ADD, address);

            personList.add(persons);
        }

        ListAdapter adapter = new
SimpleAdapter(VerRegistroestacionamiento.this, personList,
R.layout.info_registro,
            new String[]{TAG_ID, TAG_NAME, TAG_ADD},
            new int[]{R.id.txtfolix, R.id.txtinix, R.id.txtfinx}
        );

        listaregistro.setAdapter(adapter);
    }
}

```

```

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}

```

## 10.28 VerRegistrocon.class

Permite visualizar en forma de lista, el historial de todos los permisos con goce de sueldo hechas por el trabajador.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.HashMap;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.ClientProtocolException;
import cz.msebera.android.httpclient.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.params.BasicHttpParams;

public class VerRegistrocon extends AppCompatActivity {

    String myJSON;

    private static final String TAG_RESULTS="registrocg";
    private static final String TAG_ID = "idpermisocon";
    private static final String TAG_REGISDIA="dizxusar";
    private static final String TAG_NAME = "fechacomip";
    private static final String TAG_ADD = "fechafinp";
    private static final String TAG_TXT = "timexime";
    private static final String TAG_COMEN="cominzuuts";

    JSONArray peoples = null;

```

```

ArrayList<HashMap<String, String>> personList;

ListView listaregistro;
ArrayList<String> dizx = new ArrayList();
ArrayList<String> commtx = new ArrayList();
ArrayList<String> fend = new ArrayList();
String numemple;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ver_registrocon);
    listaregistro = (ListView) findViewById(R.id.listregispcong);
    Intent intancix = getIntent();
    Bundle extravar = intancix.getExtras();

    if(extravar!=null) {
        numemple = extravar.getString("tmnum");
    }
    else{
        Toast.makeText(VerRegistrocon.this, "No hay datos",
Toast.LENGTH_LONG).show();
    }

    personList = new ArrayList<HashMap<String,String>>();

    metodobusqpsin();

    listaregistro.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {

            Intent intent = new
Intent(VerRegistrocon.this,DetalleVacaciones.class);
            intent.putExtra("dikx",dizx.get(position).toString());
            intent.putExtra("comfrt",commtx.get(position).toString());
            startActivity(intent);

        }
    });
}

//Método para buscar todo el historial del trabajador con relación a las
peticiones de permisos con goce de sueldo
private void metodobusqpsin() {
    class GetDataJSON extends AsyncTask<String, Void, String> {

        @Override
        protected String doInBackground(String... params) {
            DefaultHttpClient httpclient = new DefaultHttpClient(new
BasicHttpParams());
            HttpPost httppost = new
HttpPost("http://192.168.1.72/iniciarphtml/panel/codjsonregipcontm.php?cod=" +
numemple);

```

```

// Depends on your web service
httppost.setHeader("Content-type", "application/json");

InputStream inputStream = null;
String result = null;
try {
    HttpResponse response = httpClient.execute(httppost);
    HttpEntity entity = response.getEntity();

    inputStream = entity.getContent();
    // json is UTF-8 by default
    BufferedReader reader = new BufferedReader(new
InputStreamReader(inputStream, "UTF-8"), 8);
    StringBuilder sb = new StringBuilder();

    String line = null;
    while ((line = reader.readLine()) != null) {
        sb.append(line + "\n");
    }
    result = sb.toString();

} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
} catch (ClientProtocolException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} finally {
    try {
        if (inputStream != null) inputStream.close();
    } catch (Exception squish) {
    }
}
return result;
}
protected void onPostExecute(String result){
    myJSON=result;
    showListrps();
}
}
GetDataJSON g = new GetDataJSON();
g.execute();
}

```

//Método para visualizar los datos descargador en forma de lista

```

private void showListrps() {
    try {
        JSONObject jsonObj = new JSONObject(myJSON);
        peoples = jsonObj.getJSONArray(TAG_RESULTS);

        for(int i=0;i<peoples.length();i++){
            JSONObject c = peoples.getJSONObject(i);
            String id = c.getString(TAG_ID);
            String name = c.getString(TAG_NAME);
            String address = c.getString(TAG_ADD);
            String timext=c.getString(TAG_TXT);
            dixx.add(c.getString(TAG_REGISDIA));
            comntx.add(c.getString(TAG_COMEN));
        }
    }
}

```

```

        HashMap<String,String> persons = new HashMap<String,String>();

        persons.put(TAG_ID,id);
        persons.put(TAG_NAME,name);
        persons.put(TAG_ADD,address);
        persons.put(TAG_TXT,timext);

        personList.add(persons);
    }

    ListAdapter adapter = new SimpleAdapter(VerRegistropcon.this,
personList, R.layout.info_registro,
        new String[]{TAG_ID,TAG_NAME,TAG_ADD,TAG_TXT},
        new int[]{R.id.txtfolix, R.id.txtinix, R.id.txtfinx,
R.id.timxtimpcon}
    );

    listaregistro.setAdapter(adapter);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}
}

```

## 10.29 VerRegistropsin.class

Permite visualizar en forma de lista, el historial de todos los permisos con goce de suleodo hechas por el trabajador.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.HashMap;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.ClientProtocolException;
import cz.msebera.android.httpclient.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;

```

```

import cz.msebera.android.httpclient.params.BasicHttpParams;

public class VerRegistropsin extends AppCompatActivity {

    String myJSON;

    private static final String TAG_RESULTS="registrops";
    private static final String TAG_ID = "idpermisosin";
    private static final String TAG_REGISDIA="dizxusar";
    private static final String TAG_NAME = "fechastarp";
    private static final String TAG_ADD ="fechasndp";
    private static final String TAG_COMEN="comentariozw";

    JSONArray peoples = null;

    ArrayList<HashMap<String, String>> personList;

    ListView listaregistro;
    ArrayList<String> dizx = new ArrayList();
    ArrayList<String> comntx = new ArrayList();
    ArrayList<String> fend = new ArrayList();
    String numemple;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ver_registropsin);
        listaregistro = (ListView) findViewById(R.id.listregispsing);
        Intent intanciax = getIntent();
        Bundle extravar = intanciax.getExtras();

        if(extravar!=null) {
            //String dato = extrainta.getString("fecha");
            //fedatetm.setText(dato);
            numemple = extravar.getString("tmnum");
            //tvwfolio.setText(numemple);
        }
        else{
            Toast.makeText(VerRegistropsin.this,"No hay datos",
            Toast.LENGTH_LONG).show();
        }

        personList = new ArrayList<HashMap<String,String>>();

        metodobusqpsin();

        listaregistro.setOnItemClickListener(new
        AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int
            position, long id) {

                Intent intent = new
                Intent(VerRegistropsin.this,DetalleVacaciones.class);
                intent.putExtra("dikx",dizx.get(position).toString());
                //String
                urlfotoz="http://192.168.0.18/iniciarphp/fotos/"+foto.get(position).toString();
                intent.putExtra("comfrt",comntx.get(position).toString());
                //intent.putExtra("detalle",descrip.get(position).toString());
                startActivity(intent);
            }
        }
    }
}

```

```

    });
}
//Método para buscar todo el historial del trabajador con relación a las
//peticiones de permiso sin goce de sueldo
private void metodobusqpsin() {
    class GetDataJSON extends AsyncTask<String, Void, String> {

        @Override
        protected String doInBackground(String... params) {
            DefaultHttpClient httpclient = new DefaultHttpClient(new
            BasicHttpParams());
            HttpPost httppost = new
            HttpPost("http://192.168.1.72/iniciarpHP/panel/codjsonregipsintm.php?cod=" +
            numemple);

            // Depends on your web service
            httppost.setHeader("Content-type", "application/json");

            InputStream inputStream = null;
            String result = null;
            try {
                HttpResponse response = httpclient.execute(httppost);
                HttpEntity entity = response.getEntity();

                inputStream = entity.getContent();
                // json is UTF-8 by default
                BufferedReader reader = new BufferedReader(new
                InputStreamReader(inputStream, "UTF-8"), 8);
                StringBuilder sb = new StringBuilder();

                String line = null;
                while ((line = reader.readLine()) != null) {
                    sb.append(line + "\n");
                }
                result = sb.toString();

            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            } catch (ClientProtocolException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            } finally {
                try {
                    if (inputStream != null) inputStream.close();
                } catch (Exception squish) {
                }
            }
            return result;
        }
        protected void onPostExecute(String result){
            myJSON=result;
            showListrps();
        }
    }
    GetDataJSON g = new GetDataJSON();
    g.execute();
}
//Método para visualizar los datos descargados en forma de lista
protected void showListrps() {
    try {
        JSONObject jsonObj = new JSONObject(myJSON);
        peoples = jsonObj.getJSONArray(TAG_RESULTS);
    }
}

```



```

        for(int i=0;i<peoples.length();i++){
            JSONObject c = peoples.getJSONObject(i);
            String id = c.getString(TAG_ID);
            String name = c.getString(TAG_NAME);
            String address = c.getString(TAG_ADD);
            dizx.add(c.getString(TAG_REGISDIA));
            comntx.add(c.getString(TAG_COMEN));

            HashMap<String,String> persons = new HashMap<String,String>();

            persons.put(TAG_ID,id);
            persons.put(TAG_NAME,name);
            persons.put(TAG_ADD,address);

            personList.add(persons);
        }

        ListAdapter adapter = new SimpleAdapter(VerRegistropsin.this,
personList, R.layout.info_registro,
            new String[]{TAG_ID,TAG_NAME,TAG_ADD},
            new int[]{R.id.txtfolix, R.id.txtinix, R.id.txtfinx}
        );

        listaregistro.setAdapter(adapter);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}
}

```

### 10.30 VerRegistrovacaciones.class

Permite visualizar en forma de lista, el historial de todas las solicitudes de vacaciones hechas por el trabajador.

```

package roberto.espejel.app.proyecto.docpt.rulaitas.ian;

import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

```

```

import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.HashMap;

import cz.msebera.android.httpclient.HttpEntity;
import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.ClientProtocolException;
import cz.msebera.android.httpclient.methods.HttpPost;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.params.BasicHttpParams;

public class VerRegistrovacaciones extends AppCompatActivity {

    String myJSON;

    private static final String TAG_RESULTS="registro";
    private static final String TAG_ID = "idregivaca";
    private static final String TAG_REGISDIA="tabudiasvacaciones_iddivkto";
    private static final String TAG_NAME = "fechainicio";
    private static final String TAG_ADD ="fechafin";
    private static final String TAG_COMEN="comentarioxz";

    JSONArray peoples = null;

    ArrayList<HashMap<String, String>> personList;

    ListView listaregistro;
    ArrayList<String> dizx = new ArrayList();
    ArrayList<String> comntx = new ArrayList();
    String numemple;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ver_registro);
        listaregistro = (ListView) findViewById(R.id.regislist);
        Intent intanciax = getIntent();
        Bundle extravar = intanciax.getExtras();

        if(extravar!=null) {
            numemple = extravar.getString("tmnum");
        }
        else{
            Toast.makeText(VerRegistrovacaciones.this, "No hay datos",
                Toast.LENGTH_LONG).show();
        }

        personList = new ArrayList<HashMap<String, String>>();
        metodobusq();
        listaregistro.setOnItemClickListener(new
        AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int
            position, long id) {

                Intent intent = new
                Intent(VerRegistrovacaciones.this, DetalleVacaciones.class);
                intent.putExtra("dikx", dizx.get(position).toString());
                intent.putExtra("comfrt", comntx.get(position).toString());
                startActivity(intent);
            }
        });
    }
}

```

```

    });
}

//Método para buscar todo el historial del trabajador con relación a las
//peticiones de vacaciones
private void metodobusq() {
    class GetDataJSON extends AsyncTask<String, Void, String> {

        @Override
        protected String doInBackground(String... params) {
            DefaultHttpClient httpClient = new DefaultHttpClient(new
            BasicHttpParams());
            HttpPost httpPost = new
            HttpPost("http://192.168.1.72/iniciarpHP/panel/codjsonregivaka.php?cod=" +
            numemple);

            // Depends on your web service
            httpPost.setHeader("Content-type", "application/json");

            InputStream inputStream = null;
            String result = null;
            try {
                HttpResponse response = httpClient.execute(httpPost);
                HttpEntity entity = response.getEntity();

                inputStream = entity.getContent();
                // json is UTF-8 by default
                BufferedReader reader = new BufferedReader(new
                InputStreamReader(inputStream, "UTF-8"), 8);
                StringBuilder sb = new StringBuilder();

                String line = null;
                while ((line = reader.readLine()) != null) {
                    sb.append(line + "\n");
                }
                result = sb.toString();

            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            } catch (ClientProtocolException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            } finally {
                try {
                    if (inputStream != null) inputStream.close();
                } catch (Exception squish) {
                }
            }
            return result;
        }

        protected void onPostExecute(String result){
            myJSON=result;
            showList();
        }
    }
    GetDataJSON g = new GetDataJSON();
    g.execute();
}
//Método para visualizar los datos en forma de lista

```

```

protected void showList() {
    try {
        JSONObject jsonObj = new JSONObject(myJSON);
        peoples = jsonObj.getJSONArray(TAG_RESULTS);

        for(int i=0;i<peoples.length();i++){
            JSONObject c = peoples.getJSONObject(i);
            String id = c.getString(TAG_ID);
            String name = c.getString(TAG_NAME);
            String address = c.getString(TAG_ADD);
            dixx.add(c.getString(TAG_REGISDIÁ));
            comntx.add(c.getString(TAG_COMEN));

            HashMap<String,String> persons = new HashMap<String,String>();

            persons.put(TAG_ID,id);
            persons.put(TAG_NAME,name);
            persons.put(TAG_ADD,address);

            personList.add(persons);
        }

        ListAdapter adapter = new SimpleAdapter(VerRegistrovacaciones.this,
        personList, R.layout.info_registro,
            new String[]{TAG_ID,TAG_NAME,TAG_ADD},
            new int[]{R.id.txtfolix, R.id.txtinix, R.id.txtfinx}
        );

        listaregistro.setAdapter(adapter);

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}

```