

Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Reporte Final del Proyecto de Integración
Proyecto Tecnológico
Aplicación móvil para el registro de afectaciones por sismos

Trimestre 2018 Otoño

José Luis Delgadillo Cervantes
2142000942
al2142000942@alumnos.azc.uam.mx

M en C. Josué Figueroa González
Departamento de Sistemas
jfgo@azc.uam.mx

29 de Noviembre de 2018

Declaratoria

Yo, M en C. Josué Figueroa González, 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 en el Repositorio de UAM Azcapotzalco.



M en C. Josué Figueroa González

Yo, José Luis Delgadillo Cervantes, 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 UAM Azcapotzalco.



José Luis Delgadillo Cervantes

Justificación

Debido a los problemas referentes a la consolidación de información durante el registro de la misma se propone una aplicación de registro de afectaciones por sismos para obtener datos consistentes mediante un formulario, captura de imágenes y posición geográfica.

La aplicación móvil permitirá recolectar la información de manera estructurada y organizada lo que permitirá su posterior análisis, clasificación y organización de una manera más eficiente para posteriores revisiones.

Objetivos

Objetivo General

- Diseñar e implementar una aplicación móvil capaz de realizar el registro de incidentes a inmuebles por sismos.

Objetivos Específicos

- Diseñar e implementar el módulo de obtención de posición geográfica
- Diseñar e implementar el módulo de captura de imagen mediante cámara fotográfica.
- Diseñar e implementar el módulo de Registro de Incidentes.

Marco Teórico

En esta sección se indica el uso de las tecnologías aplicadas para la realización de este proyecto.

1.1.Android

1.1.1 Historia

En octubre de 2003, localidad Palo Alto, Andy Rubin, Rich miner, Chris White y Nivk Sears fundan Android Inc. con el objetivo de desarrollar un sistema operativo para móviles basados en Linux.

Android cuenta con una gran comunidad de desarrolladores creando aplicaciones para extender la funcionalidad de los dispositivos. A principios de 2018 se superaban ya los dos millones de aplicaciones disponibles en Google Play, la tienda de aplicaciones oficial de Android; No obstante, no es un sistema operativo libre de malware, aunque la mayoría de ello es descargado de sitios de terceros.[3]

1.1.2 Versiones

Las versiones de Android reciben, en inglés, el nombre de diferentes postres o dulces. En cada versión el postre elegido empieza por una letra que va conforme un orden alfabético, a continuación se muestran en la Tabla 1 las versiones existentes hasta diciembre de 2018.

Versiones Android			
Letra	Nombre	Versión	Traducción
A	Apple Pie	1.0	Tarta de Manzana
B	Banana Bread	1.1	Pan de Platano
C	Cupcake	1.5	Magdalena
D	Donut	1.6	Dona
E	Éclair	2.0/2.1	Palo de Crema
F	Froyo	2.2	Yogur helado
G	Gingerbread	2.3	Pan de Gengibre
H	Honeycomb	3.0/3.1/3.2	Panel
I	Ice Cream Sandwich	4.0	Sandwich de Helado
J	Jelly Bean	4.1/4.2/4.3	Gominola
K	Kit Kat	4.4	Kit Kat
L	Lollipop	5.0/5.0.1/5.0.2/5.1	Piruleta
M	Marshmallow	6.0/6.0.1	Malvavisco
N	Nougat	7.0/7.1/7.1.1/7.1.2	Turrón
O	Oreo	8.0/8.1	Oreo
P	Pie	9.0	Pastel

Tabla 1. Versiones de Android.

Las versiones de Android más utilizadas hasta finales del 2018 son[3]:

- Android Oreo 8.x (21.5%)
- Android Nougat 7.x (28.2%)
- Android Marshmallow 6.x (21.3%)
- Android Lollipop 5.x (17.9%)
- Android Kit Kat 4.4 (7.6%)
- Android Jelly Bean 4.x (3.0%)
- Anteriores (0.5%)

El sistema operativo no es exclusivo de teléfonos inteligentes puesto que también se encuentra en portátiles, netbooks, tabletas, GoogleTV, relojes de pulsera, auriculares receptores decodificadores integrados entre otros dispositivos.

1.1.3 Android Studio

Android Studio es el entorno de Desarrollo Integrado (IDE) oficial para el desarrollo de aplicaciones para Android se basa en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android, como las siguientes:

- Un sistema de compilación basado en Gradle flexible.
- Un emulador rápido con varias funciones.
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android.
- Instant Run para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK.
- Integración de plantillas de código y GitHub para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código.
- Gran cantidad de herramienta y frameworks de prueba.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión etc.
- Compatibilidad con C++ y NDK.
- Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine.

1.1.4 Arquitectura de Android

Android es una pila de software de código abierto basado en Linux creada para una variedad amplia de dispositivos y factores de forma. En la Figura 1 se muestran los componentes principales de la plataforma Android.[10]

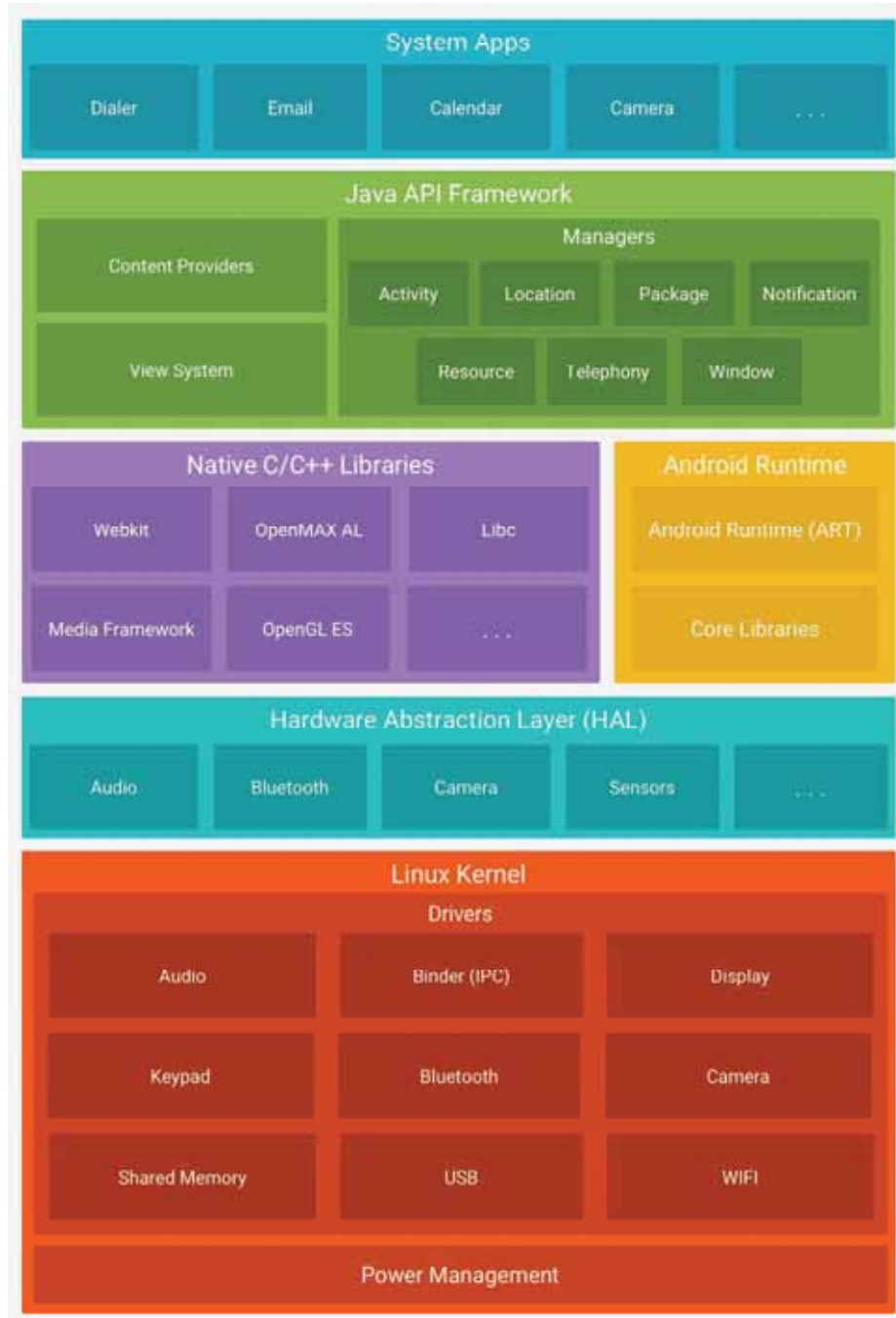


Figura 1. Arquitectura Android.

1.1.5 Ciclo de Vida Aplicación

El ciclo de vida de una activity, fragment, o de manera general de la aplicación se muestra en la Figura 2, para mas detalle se describen los métodos:

`onCreate(Bundle)`

Representa el momento en el que la actividad se crea. Este método normalmente lo generará el asistente al crear una nueva actividad en Android, y es donde crearemos todo lo que vaya a necesitar la actividad. Si antes hemos salvado los datos de la actividad en un objeto Bundle, podremos utilizarlo para regenerarla. Normalmente no lo usaremos.[11]

`onStart()`

La actividad va a pasar a estar en pantalla, aunque no necesariamente visible. Si venimos de una parada, pasaremos antes por `onRestart()`.

`onRestart()`

Anterior a `onStart()` cuando procedemos de una llamada a `onStop()`.

`onResume()`

La actividad va a empezar a responder a la interacción del usuario.

`onPause()`

La actividad va a dejar de responder a la interacción del usuario.

`onStop()`

La actividad ha pasado completamente a segundo plano.

`onDestroy()`

La actividad va a ser destruida y sus recursos liberados.

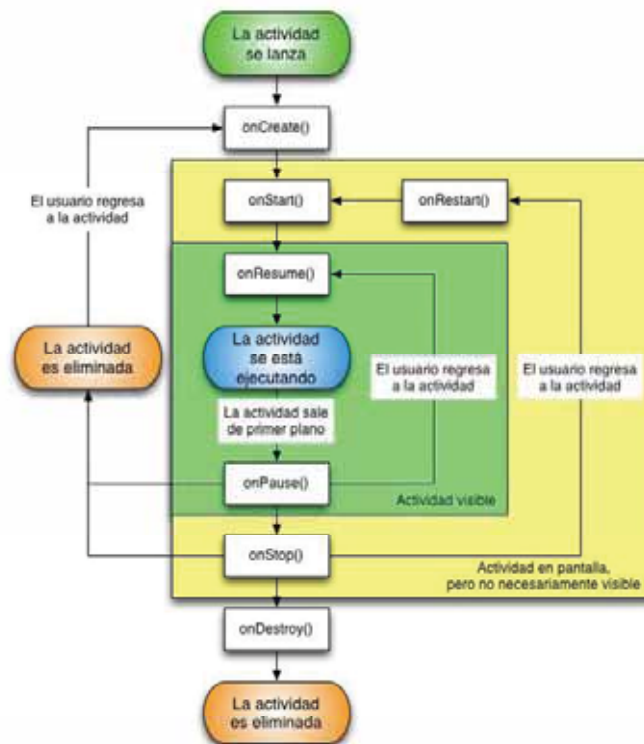


Figura 2. Ciclo de vida aplicación Android.

1.2 MySQL

MySQL es un sistema de gestión de bases de datos SQL (Structured Query Language) de código abierto más popular, está desarrollado, distribuido y respaldado por Oracle Corporation.

MySQL Server se desarrolló originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y se ha utilizado con éxito en entornos de producción altamente exigentes durante varios años. Aunque está en constante desarrollo, MySQL Server ofrece hoy un conjunto de funciones rico y útil. Su conectividad, velocidad y seguridad hacen que MySQL Server sea muy adecuado para acceder a bases de datos en Internet.

El software de base de datos MySQL es un sistema cliente / servidor que consta de un servidor SQL multiproceso que admite diferentes back-end, varios programas y bibliotecas diferentes para el cliente, herramientas administrativas y una amplia gama de interfaces de programación de aplicaciones (API).[16]

1.3 Volley

Volley es una biblioteca HTTP que hace que las redes para aplicaciones de Android sean más fáciles y, lo que es más importante, más rápidas. Volley está disponible en GitHub.

Volley ofrece los siguientes beneficios:

- Programación automática de solicitudes de red.
- Múltiples conexiones de red concurrentes.
- Disco transparente y memoria caché de respuesta con coherencia de caché HTTP estándar.
- Soporte para la priorización de solicitudes.
- API de solicitud de cancelación.
- Facilidad de personalización.
- Pedidos sólidos que facilitan el llenado correcto de su IU con datos obtenidos de forma asíncrona desde la red.
- Herramientas de depuración y rastreo.

Volley es excelente en las operaciones de tipo RPC utilizadas para rellenar una interfaz de usuario, como recuperar una página de resultados de búsqueda como datos estructurados. Se integra fácilmente con cualquier protocolo y sale de la caja con soporte para cadenas en bruto, imágenes y JSON. Al proporcionar soporte integrado para las funciones que necesita, Volley lo libera de escribir código repetitivo y le permite concentrarse en la lógica específica de su aplicación.

Desarrollo del proyecto

2.1 Componentes

Para el desarrollo del proyecto se requirieron componentes tanto de software como hardware los cuales eran necesarios para la conclusión de este proyecto. A continuación se describen dichos componentes.

2.1.1 Software

Para el desarrollo de la aplicación se requirió Android Studio en su versión 3.2.1, junto con el SDK de Android y AVD Manager para que fuera posible la virtualización de un dispositivo Android y/o en su defecto depurar mediante un teléfono inteligente, el motor de base de datos MySQL en su versión 8.0 y su manejador MySQL WorkBench en su versión 6.3, el editor de textos Sublime Text en su Build 4041 y por último el Servidor Apache TomCat en su versión 2.4.29.

2.1.2 Hardware

El hardware que se requirió únicamente de una Laptop capaz de soportar Android Studio y un teléfono inteligente los cuales cumplieran con las siguientes características respectivamente.

Requisitos mínimos de Android Studio[18]

- Windows 7/8/10 (32 o 64 bits).
- 2 GB de RAM (8 GB de RAM recomendado).
- 2 GB de espacio libre mínimo (4 GB recomendado).
- Resolución mínima de 1.280 x 800.
- Java 8.
- 64 bits y procesador Intel (emulador).

Requisitos mínimos Teléfono

- Android 4.1 o superior
- Wifi
- GPS
- Camara
- 512MB memoria RAM
- 4GB memoria Interna

2.2 Metodología

Debido al tamaño del proyecto se optó por usar la metodología conocida como Desarrollo en Cascada la cual va a bloques fijando un objetivo uno a uno como se ilustra en la Figura 3, al ser este de forma secuencial para poder comenzar una nueva etapa se requiere terminar la anterior y así sucesivamente hasta su despliegue y mantenimiento,[19]

El método consta de 5 etapas que se enlistan a continuación.

- **Requisitos:** En esta fase se analizan las necesidades de los usuarios finales del software para determinar que objetivos debe cubrir.
- **Diseño del Sistema:** Descompone y organiza el sistema en elementos que puedan elaborarse por separado, se realizan los algoritmos necesarios para el cumplimiento de los requerimientos, también es la fase donde se implementa el código fuente,haciendo uso de prototipos así como de pruebas y ensayos para corregir errores.
- **Implementación:** Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente.
- **Verificación:** Es la fase en donde el usuario final o cliente ejecuta el sistema, y se asegura que cubra sus necesidades.
- **Mantenimiento:** Una de las etapas más críticas, ya que se destina un 75% de los recursos, es el mantenimiento del software ya que al utilizarlo como usuario final puede ser que no cumpla con todas las expectativas.

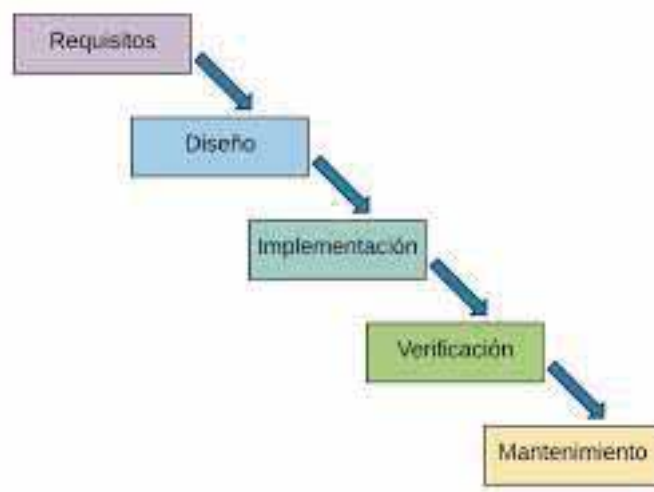


Figura 3.Modelo en cascada.

2.3 Desarrollo

Basándose en los objetivos se describe el proceso para el desarrollo del sistema que a continuación se detalla.

2.3.1 Diseño de plantillas

Se utiliza el formato del CENAPRED el cual esta dividida en secciones como se muestra en las Figuras 4, 5 y 6.

INFORMACIÓN GENERAL		Fecha:	Coordenadas: (_____ N, _____ O, _____ msnm)
Nombre del inmueble:			
Calle y número:		Colonia:	Código postal:
Pueblo o ciudad:		Delegación/Municipio:	Estado:
Referencias:			(entre calles "A" y "B", un sitio notable, etc.)
Contacto: nombre, cargo, correo-e:			Teléfono: +()

Figura 4. Información general.

Uso:	<input type="checkbox"/> Vivienda	<input type="checkbox"/> Hospital	Número total de niveles, n = _____	Año de:	Topografía:	<input type="checkbox"/> Planicie
	<input type="checkbox"/> Oficinas	<input type="checkbox"/> Iglesia	Número de sótanos: _____	de construcción: _____		<input type="checkbox"/> Ladera de cerro
<input type="checkbox"/> Comercio	<input type="checkbox"/> Reunión (cine/estadio/salón)	Pisos para estacionamiento: _____	Número ocupantes: _____	de daño severo: _____	<input type="checkbox"/> Rivera río/lago	<input type="checkbox"/> Fondo de valle
<input type="checkbox"/> Escuela	<input type="checkbox"/> Industrial (fábrica/bodega)	<input type="checkbox"/> Elevador		de rehabilitación: _____	<input type="checkbox"/> Depósitos lacustres	<input type="checkbox"/> Costa
<input type="checkbox"/> Otro: _____		<input type="checkbox"/> Escalera de emergencia		Dimensiones:		
<input type="checkbox"/> Desocupada				Frente X = _____ m		
				Fondo Y = _____ m		
Otras características: Mantenimiento: _____		Zona de seguridad: _____	Sistema contra incendio: _____	Cisterna/tinacos: _____	m ²	

Figura 5. Uso Inmueble.

EVALUACIÓN DE DAÑOS																																																									
Geotécnicos: <input type="checkbox"/> Grietas en el terreno <input type="checkbox"/> Hundimientos: _____ cm <input type="checkbox"/> Inclinación del edificio: _____ % Losas: <input type="checkbox"/> Colapso <input type="checkbox"/> Grietas máx: _____ mm <input type="checkbox"/> Flecha máx: _____ cm Conexiones: <input type="checkbox"/> Falla	<table border="1"> <tr> <td></td> <td>Columnas</td> <td>Colapso</td> <td>Grietas cortante</td> <td>Grietas flexión</td> <td>Aplastamiento</td> <td>Pandeo barras</td> <td>Pandeo placas</td> <td>Falla soldadura</td> <td>Ancho máximo de grieta (mm)</td> <td>Separación de estribos (cm)</td> </tr> <tr> <td></td> <td>Trabes</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td>_____ mm</td> <td>_____ cm</td> </tr> <tr> <td></td> <td>Muro</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td>_____ mm</td> <td>_____ cm</td> </tr> <tr> <td></td> <td>Concreto</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td>_____ mm</td> <td>_____ cm</td> </tr> <tr> <td></td> <td>Mampost.</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td>_____ mm</td> <td>_____ cm</td> </tr> </table>		Columnas	Colapso	Grietas cortante	Grietas flexión	Aplastamiento	Pandeo barras	Pandeo placas	Falla soldadura	Ancho máximo de grieta (mm)	Separación de estribos (cm)		Trabes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ mm	_____ cm		Muro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ mm	_____ cm		Concreto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ mm	_____ cm		Mampost.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ mm	_____ cm	Entrepiso crítico (más débil y/o más dañado): No. de columnas (o muros) daño severo = _____ (colapso, aplastamiento, pandeo, grietas > 2 mm) Total de columnas (muros) en el entrepiso = _____ NIVEL DE DAÑO DE LA ESTRUCTURA <input type="checkbox"/> Colapso total <input type="checkbox"/> Daño severo <input type="checkbox"/> Colapso parcial <input type="checkbox"/> Daño medio <input type="checkbox"/> Daño ligero
	Columnas	Colapso	Grietas cortante	Grietas flexión	Aplastamiento	Pandeo barras	Pandeo placas	Falla soldadura	Ancho máximo de grieta (mm)	Separación de estribos (cm)																																															
	Trabes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ mm	_____ cm																																															
	Muro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ mm	_____ cm																																															
	Concreto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ mm	_____ cm																																															
	Mampost.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____ mm	_____ cm																																															
Otros daños: <input type="checkbox"/> Vidrios <input type="checkbox"/> Acabados <input type="checkbox"/> Plafones <input type="checkbox"/> Fachadas <input type="checkbox"/> Bardas y pretilas <input type="checkbox"/> Cubos (escalera/elevador) <input type="checkbox"/> Instalaciones																																																									

Figura 6. Evaluación de daños.

Realizando el diseño de la aplicación como se muestran en las Figuras 7, 8 y 9 con base en las Figuras antes mencionadas.

Incidente: INC378
Usuario: Admin

Nombre Inmueble

Nombre de la Calle

Numero Ext. _____ Numero Int. _____

Colonia _____ C. P. _____

Delegacion o Municipio _____

Estado:
Ciudad de México

Indicar referencia Ej. Entre calles a y b

Nombre Contacto

55-12345678

correo@example.com

SIGUIENTE

Figura 7. Diseño información general.

Incidente: INC378
Usuario: Admin

Uso Inmueble

Vivienda

Num. Niveles _____ Num. Setanos _____

Plazo Estacionamiento _____ Num. Ocupantes _____

Elevador Escalera Emergencia

Año de Construcción _____ Año de dño de severo _____

Año de Rehabilitación _____

Topografía: Planicie

Mantenimiento Zona de Seguridad

Sistema contra incendio Sistema

*En caso de contar con sistema indique: _____ M3

ANTERIOR SIGUIENTE

Figura 8. Diseño Uso Inmueble.

Incidente: INC378
Usuario: Admin



GEOTÉCNICOS

Grietas en el Terreno Hundimientos

*En caso de haber hundimiento especifique: CM

Inclinación del edificio

*En caso de inclinación especifique: %

LOSAS

Colapso
 Grietas máx Flecha máx

*En caso de grietas (máx) especifique: mm

*En caso de flecha (máx) especifique: cm

Conexiones Falla

DAÑOS ESTRUCTURALES

Columnas

No Aplica

Ancho máximo grieta mm Separación Estribo cm

Trabes

No Aplica

Ancho máximo grieta mm Separación Estribo cm

Concreto

No Aplica

Ancho máximo grieta mm Separación Estribo cm

Mampost.

No Aplica

Ancho máximo grieta mm Separación Estribo cm

No. de columnas (o muros) daño severo=

Total de columnas (muros) en el entrepiso=

NIVEL DE DAÑO DE LA ESTRUCTURA

No Aplica

OTROS DAÑOS

Vidrios Acabados
 Plafones Fachadas
 Bardas y Pretiles Cubos (Escalera/Elevador)
 Instalaciones

REGRESAR

FINALIZAR

Figura 9. Diseño evaluación de daños.

2.3.2 Diseño de la base de datos

Con base en los campos de las Figuras 7, 8 y 9 se crea la base de datos con la diferencia de anexar un identificador que englobe las secciones del formato, además de una tabla para usuarios y otra para la toma de fotografías, a continuación se muestra en la Figura 10 el diagrama de entidad relación, el código se muestra al final de este documento en el Apéndice.

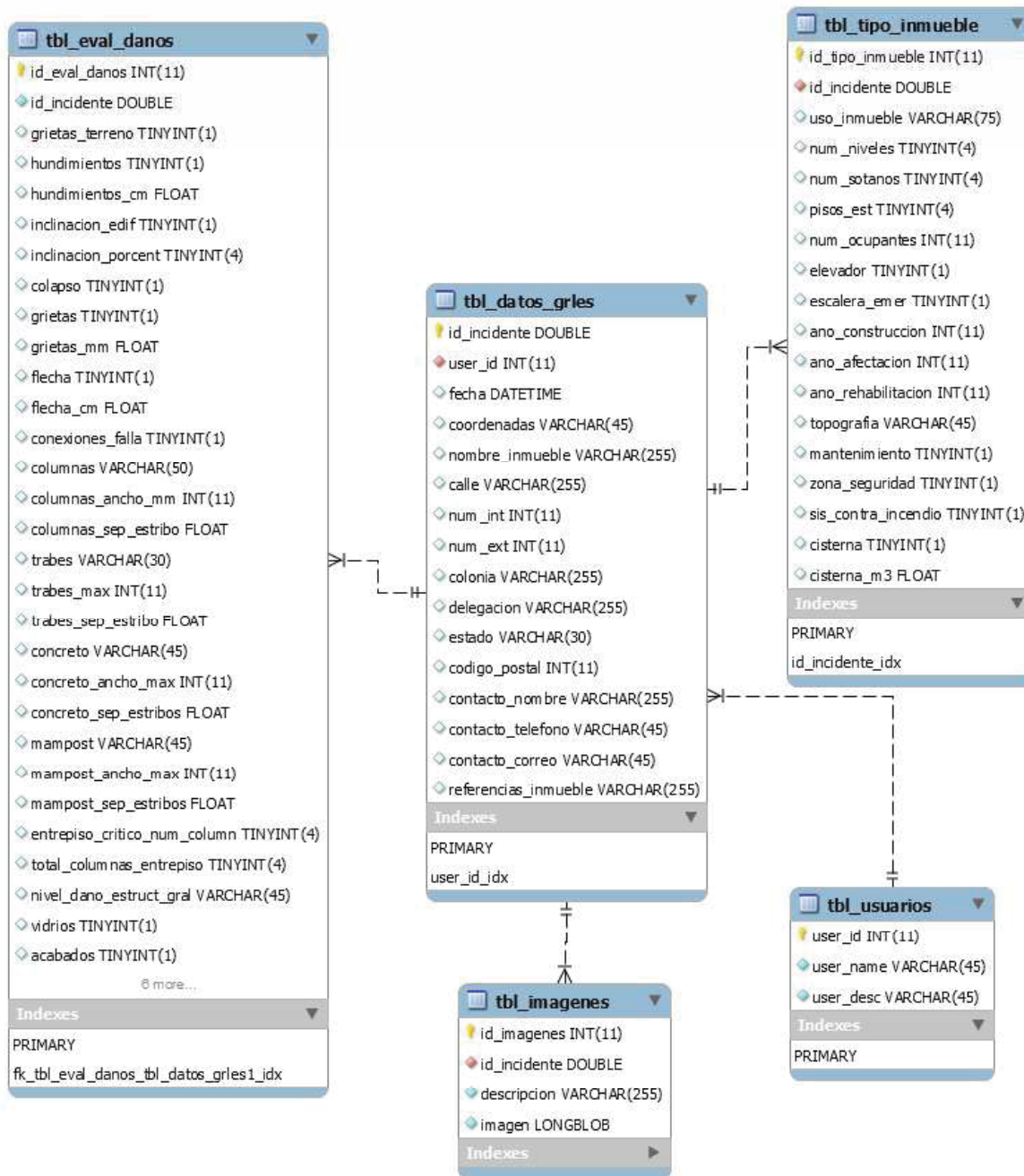


Figura 10. Diagrama entidad relación.

2.3.3 Generación de código PHP

Se generan las sentencias SQL para la validación de usuario, insertar y actualizar en la o las tablas, según corresponda en la base de datos.

2.3.4 Obtención de posición geográfica.

Este módulo será el encargado de obtener la posición geográfica del dispositivo. Este módulo será el encargado de obtener la posición geográfica del dispositivo al iniciar un incidente, ya que se considera que al iniciar el incidente el usuario se encuentra en el sitio del levantamiento. Por otra parte en ese momento se estará asignando el número de reporte correspondiente a dicho incidente.

2.3.5 Implementación captura de imagen

Este módulo será el responsable de tomar fotografías, el usuario será capaz de tomar una fotografía para enviarla hacia la base de datos con posibilidad de agregar una breve descripción como se muestra en la Figura 11.



Figura 11. Captura de imágenes.

2.3.6 Implementación registro incidente

Este módulo será el encargado de registrar los campos de las figuras 7, 8 y 9 en la base de datos, haciendo el uso de la biblioteca volley replicándose en cada una de las 3 secciones que se compone dicho incidente.

2.3.7 Implementación validación de usuario

Este módulo es el encargado de validar al usuario para iniciar un incidente nuevo, el diseño se muestra en la Figura 12.



Figura 12. Diseño inicio.

2.3.8 Diagramas

A continuación se muestra en la Figura 13 el diagrama de casos de uso y en la Figura 14 el diagrama de flujo, el cual tiene como fin el ilustrar la interacción del usuario con la aplicación.

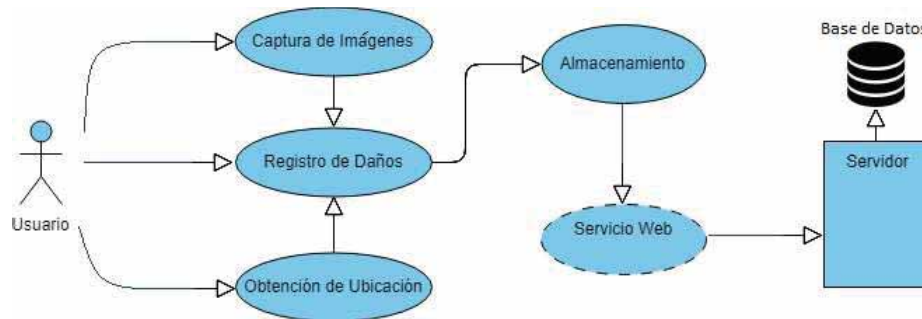


Figura 13. Diagrama casos de uso.

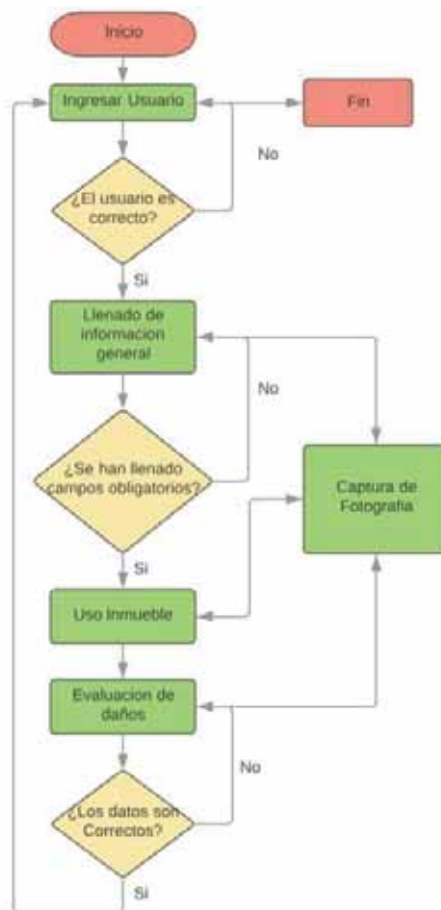


Figura 14. Diagrama de flujo.

Resultados

Concluidos e integrados los módulos descritos anteriormente, la aplicación es capaz de registrar incidentes en una base de datos con posición geográfica así como la toma de fotografías, todo debidamente integrado y vinculado a un identificador único para su distinción entre un incidente y otro.

A continuación se muestran los módulos funcionando debidamente en cada sección.

3.1 Módulo de inicio.

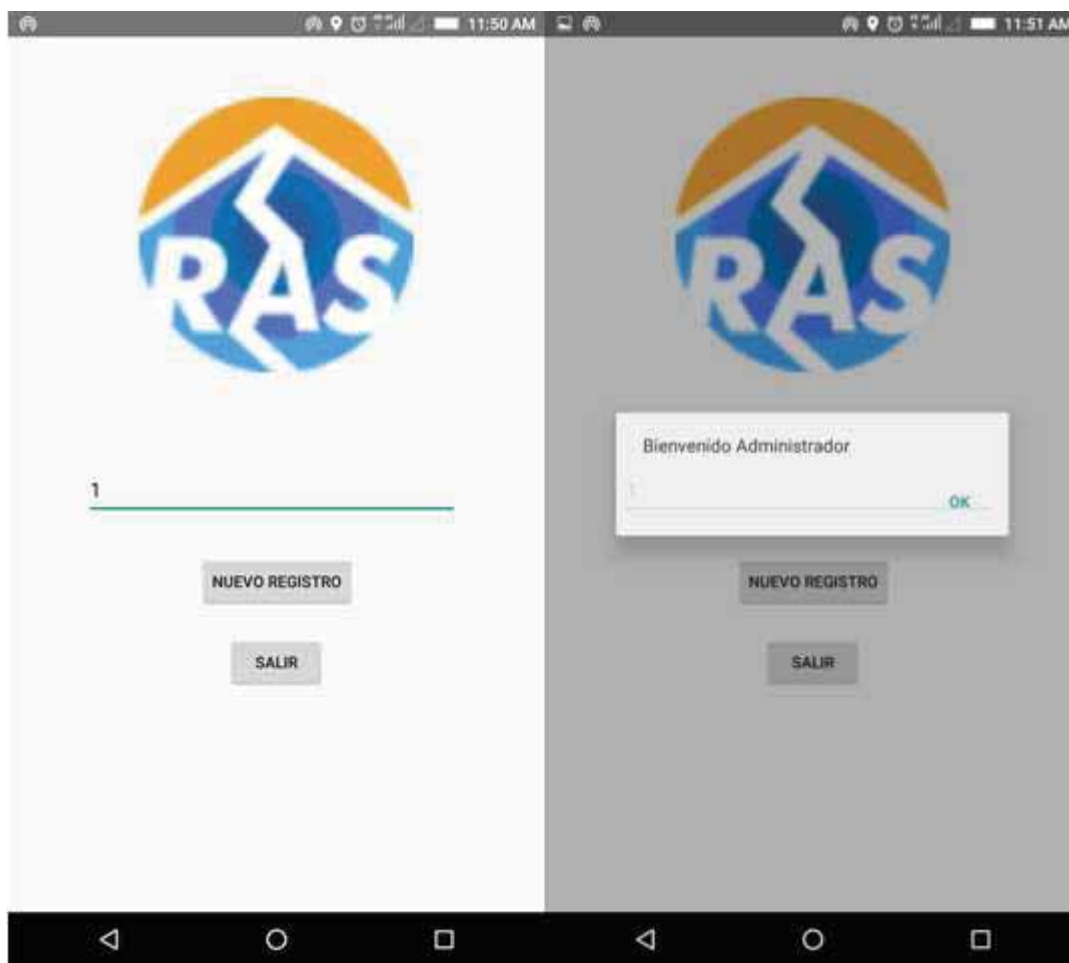


Figura 15.Login Exitoso.

3.2 Módulo de información general.

Incidente: INC384
Usuario: Administrador

SEPOMEX

Av. Ceylan

Numero Ext. 468 Numero Int. 0

Cosmopolita 02523

Azcapotzalco

Estado:
Ciudad de México

Detrás de la estación de carga Pantaco

Jose Luis Delgadillo Cervantes

5572742129

jodelgadillo@sepomex.gob.mx

SIGUIENTE

Figura 16. Módulo de Información general.

3.3 Módulo uso de inmueble.

Uso Inmueble

Vivienda

Num. Niveles 3 Num. Suites 0

Flores 1 Num. Ocupantes 500

Estado: Elevador Escalera Emergencia

Año de Construcción 1952 Año de daño severo 2017

Año de Rehabilitación 2018

Topografía: Planicie

Mantenimiento Zona de Seguridad

Sistema contra incendio Cisterna

*En caso de contar con sistema indique: 200 M3

ANTERIOR SIGUIENTE

Figura 17. Módulo Uso inmueble.

3.4 Módulo evaluación de daños.

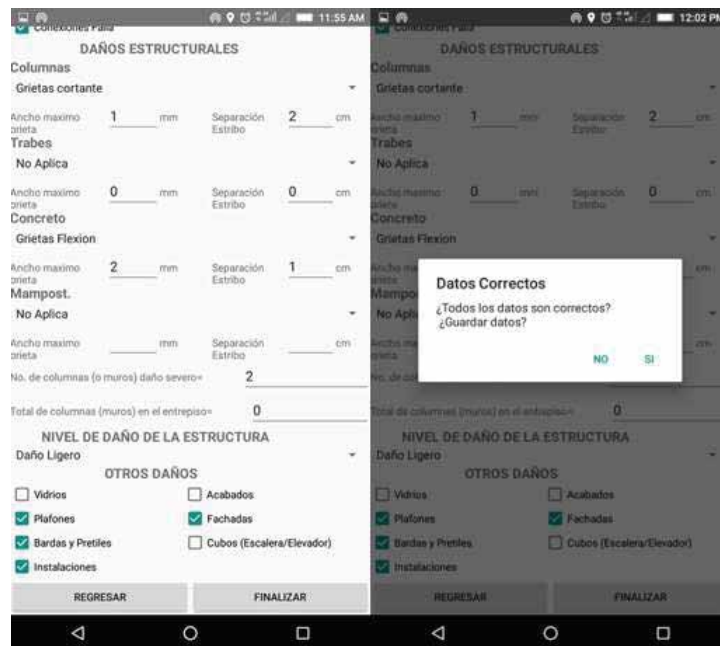


Figura 18. Módulo Evaluación de daños y termino de captura de incidente.

3.5 Módulo captura fotografías.

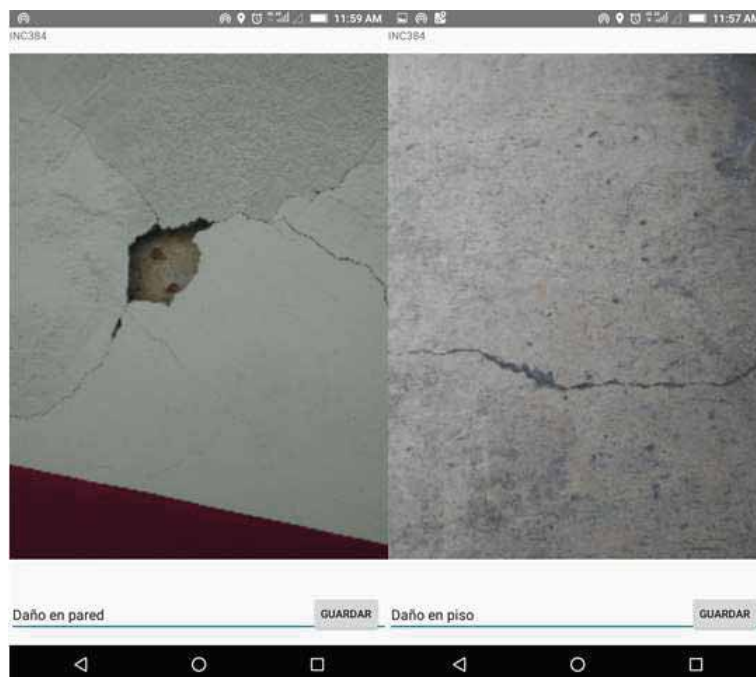


Figura 19. Módulo captura de fotografías.

3.6 Información en base de datos.

A continuación se muestra en la Figura 20 los datos obtenidos desde el dispositivo móvil a la base de datos el cual con fines de ejemplo se toma el incidente 384.

The screenshot displays four SQL queries and their corresponding result grids in a database management interface. Each query is preceded by a '1 •' icon.

Query 1: `SELECT * FROM db_ras.tbl_datos_grles where id_incidente=384;`

id_incidente	user_id	fecha	coordenadas	nombre_inmueble	calle	num	num_colonia
384	1	2018-11-24 ...	19.478489623492795,-99.16470796515209	SEPOMEX	Av. Cevlan	0	468 Cosmoopolita

Query 2: `SELECT * FROM db_ras.tbl_tipo_inmueble where id_incidente=384;`

id_tipo_inmueble	id_incidente	uso_inmueble	num_niveles	num_sotanos	pisos_est	num_ocupantes	elevador	escalera_emer	ano
210	384	Oficinas	3	0	1	500	1	0	1952

Query 3: `SELECT * FROM db_ras.tbl_eval_danos where id_incidente=384;`

id_eval_danos	id_incidente	grietas_terreno	hundimientos	hundimientos_cm	inclinacion_edif	inclinacion_porcent	colapso	grietas
210	384	1	0	0	0	0	0	0

Query 4: `SELECT * FROM db_ras.tbl_imagenes where id_incidente=384;`

id_imagenes	id_incidente	descripcion	imagen
27	384	Daño en piso	BLOB
28	384	Daño en pared	BLOB

Figura 20. Resultados base de datos.

Análisis y discusión de resultados

El resultado obtenido al finalizar el proyecto cumple con los objetivos establecidos ya que la aplicación es capaz de:

- Capturar la posición geográfica del dispositivo y almacenarla en una base de datos.
- Permitir la captura de imágenes a partir de la cámara del dispositivo móvil y almacenarla en una base de datos.
- Permitir la captura de un formulario a una base de datos.
- Realiza la integración de toda la información capturada descrita en los puntos anteriores.

Si bien no se puede asegurar que la aplicación se encuentra óptimas condiciones para un uso en una situación real, debido a la omisión de campos así como la falta de implementación de un croquis del formato original propuesto por el CENAPRED, es completamente funcional en cuanto a sus objetivos iniciales se refiere, de tal forma que se puede considerar como un proyecto terminado.

Conclusiones

En este proyecto se cumple con los objetivos establecidos en un principio ya que se ha logrado obtener una aplicación Android capaz de registrar un incidente causado por sismo a la vez que la integra y relaciona mediante tablas en una base de datos.

La aplicación desarrollada funge como una muestra del formulario original pues no se muestran todos los campos del formulario propuesto por el CENAPRED, además de no contar con un módulo que pueda interactuar con el usuario para poder dibujar un croquis que a diferencia del formato en papel del original esto ultimo es posible.

El uso de aplicaciones móviles va en aumento por lo que es de suma importancia este tipo de trabajos pues hace mas eficiente las labores de recolección de la información.

Referencias Bibliográficas

- [1] Inegi.org.mx, 2018. [Online]. Available: http://www.inegi.org.mx/saladeprensa/boletines/2017/afectaciones/afectaciones2017_09.pdf.
- [2] "Android ya potencia 2,000 millones de dispositivos en todo el mundo", CNET en Español, 2018. [Online]. Available: <https://www.cnet.com/es/noticias/android-2000-millones-de-dispositivos/>.
- [3] "Android", Es.wikipedia.org, 2018. [Online]. Available: <https://es.wikipedia.org/wiki/Android>.
- [4] J. Ricaño Quintero "Desarrollo e Implementación de un Sistema para Pre-Registro a Posgrados CBI de la Universidad Autónoma Metropolitana" Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México 2016.
- [5] E. Montelongo González "Aplicación Móvil para visualizar el expediente médico de un paciente utilizando servicios web" Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2017.
- [6] F. García Maldonado "Prototipo de una aplicación móvil para la gestión de síntomas de un paciente" Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2014.
- [7] "Semáforo Delictivo", Semaforo.com.mx, 2018. [Online]. Available: <http://www.semaforo.com.mx/>.
- [8] "Juega Limpio", Juega Limpio, 2018. [Online]. Available: <http://juegalimpio.mx/>.
- [9] Play.google.com, 2018. [Online]. Available: https://play.google.com/store/apps/details?id=com.pbksoft.pacecontrolhl=es_MX.
- [10] "Arquitectura de la plataforma — Android Developers", Android Developers, 2018. [Online]. Available: <https://developer.android.com/guide/platform/?hl=es-419>.
- [11] V. González, "El ciclo de vida de una aplicación de Android", Androidsis, 2018. [Online]. Available: <https://www.androidsis.com/el-ciclo-de-vida-de-una-aplicacion-de-android/>.
- [12] "¿Qué es Java?", Java.com, 2018. [Online]. Available: https://www.java.com/es/about/whatis_java.jsp.
- [13] "Java (lenguaje de programación)", Es.wikipedia.org, 2018. [Online]. Available: [https://es.wikipedia.org/wiki/Java\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java(lenguaje_de_programaci%C3%B3n)).

[14] "PHP: Prefacio - Manual", Php.net, 2018. [Online]. Available: <http://php.net/manual/es/preface.php>.

[15] "PHP: ¿Qué puede hacer PHP? - Manual", Php.net, 2018. [Online]. Available: <http://php.net/manual/es/intro-whatcando.php>.

[16] "MySQL :: MySQL 8.0 Reference Manual :: 1.3.1 What is MySQL?", Dev.mysql.com, 2018. [Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>.

[17] "Volley overview — Android Developers", Android Developers, 2018. [Online]. Available: <https://developer.android.com/training/volley/>.

[18] "Android Studio: Requisitos mínimos", Android Studio FAQs, 2018. [Online]. Available: <https://androidstudiofaqs.com/conceptos/android-studio-requisitos-minimos>.

[19] "Desarrollo en cascada", Es.wikipedia.org, 2018. [Online]. Available: https://es.wikipedia.org/wiki/Desarrollo_en_cascada.

Apéndice

Código Fuente

SQL

—Tabla datos generales

```
CREATE TABLE `tbl_datos_grles` (  
  `id_incidente` double NOT NULL AUTOINCREMENT,  
  `user_id` int(11) NOT NULL,  
  `fecha` datetime DEFAULT NULL,  
  `coordenadas` varchar(45) DEFAULT NULL,  
  `nombre_inmueble` varchar(255) DEFAULT NULL,  
  `calle` varchar(255) DEFAULT NULL,  
  `num_int` int(11) DEFAULT NULL,  
  `num_ext` int(11) DEFAULT NULL,  
  `colonia` varchar(255) DEFAULT NULL,  
  `delegacion` varchar(255) DEFAULT NULL,  
  `estado` varchar(30) DEFAULT NULL,  
  `codigo_postal` int(11) DEFAULT NULL,  
  `contacto_nombre` varchar(255) DEFAULT NULL,  
  `contacto_telefono` varchar(45) DEFAULT NULL,  
  `contacto_correo` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`id_incidente`),  
  KEY `user_id_idx` (`user_id`),  
  CONSTRAINT `user_id` FOREIGN KEY (`user_id`) REFERENCES `tbl_usuarios` (`user_id`) ON  
    DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB AUTOINCREMENT=148 DEFAULT CHARSET=utf8;
```

—Tabla evaluacion

```
CREATE TABLE `tbl_eval_danos` (  
  `id_eval_danos` int(11) NOT NULL AUTOINCREMENT,  
  `id_incidente` double NOT NULL,  
  `grietas_terreno` tinyint(1) DEFAULT NULL,  
  `hundimientos` tinyint(1) DEFAULT NULL,  
  `hundimientos_cm` float DEFAULT NULL,  
  `inclinacion_edif` tinyint(1) DEFAULT NULL,  
  `inclinacion_porcent` tinyint(4) DEFAULT NULL,  
  `colapso` tinyint(1) DEFAULT NULL,  
  `grietas` tinyint(1) DEFAULT NULL,  
  `grietas_mm` float DEFAULT NULL,  
  `flecha` tinyint(1) DEFAULT NULL,  
  `flecha_cm` float DEFAULT NULL,  
  `conexiones_falla` tinyint(1) DEFAULT NULL,  
  `columnas` varchar(50) DEFAULT NULL,  
  `columnas_ancho_mm` int(11) DEFAULT NULL,  
  `columnas_sep_estribo` float DEFAULT NULL,  
  `trabes` varchar(30) DEFAULT NULL,  
  `trabes_max` int(11) DEFAULT NULL,  
  `trabes_sep_estribo` float DEFAULT NULL,  
  `concreto` varchar(45) DEFAULT NULL,  
  `concreto_ancho_max` int(11) DEFAULT NULL,  
  `concreto_sep_estribos` float DEFAULT NULL,  
  `mampost` varchar(45) DEFAULT NULL,  
  `mampost_ancho_max` int(11) DEFAULT NULL,
```

```

'mampost_sep_estribos' float DEFAULT NULL,
'entrepiso_critico_num_column' tinyint(4) DEFAULT NULL,
'total_columnas_entrepiso' tinyint(4) DEFAULT NULL,
'nivel_dano_estruct_gral' varchar(45) DEFAULT NULL,
'vidrios' tinyint(1) DEFAULT NULL,
'acabados' tinyint(1) DEFAULT NULL,
'plafones' tinyint(1) DEFAULT NULL,
'fachadas' tinyint(1) DEFAULT NULL,
'bardas_pretilas' tinyint(1) DEFAULT NULL,
'cubos' tinyint(1) DEFAULT NULL,
'instalaciones' tinyint(1) DEFAULT NULL,
PRIMARY KEY ('id_eval_danos')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

—Tabla imagenes
CREATE TABLE 'tbl_imagenes' (
'id_imagenes' int(11) NOT NULL AUTOINCREMENT,
'id_incidente' double NOT NULL,
'descripcion' varchar(255) NOT NULL,
'imagen' longblob NOT NULL,
PRIMARY KEY ('id_imagenes'),
KEY 'id_incidente_idx' ('id_incidente'),
CONSTRAINT 'incidente' FOREIGN KEY ('id_incidente') REFERENCES 'tbl_datos_grles' ('
id_incidente') ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

—Tabla tipo inmuebles
CREATE TABLE 'tbl_tipo_inmueble' (
'id_tipo_inmueble' int(11) NOT NULL AUTOINCREMENT,
'id_incidente' double NOT NULL,
'uso_inmueble' varchar(75) DEFAULT NULL,
'num_niveles' tinyint(4) DEFAULT NULL,
'num_sotanos' tinyint(4) DEFAULT NULL,
'pisos_est' tinyint(4) DEFAULT NULL,
'num_ocupantes' int(11) DEFAULT NULL,
'elevador' tinyint(1) DEFAULT NULL,
'escalera_emer' tinyint(1) DEFAULT NULL,
'ano_construccion' int(11) DEFAULT NULL,
'ano_afectacion' int(11) DEFAULT NULL,
'ano_rehabilitacion' int(11) DEFAULT NULL,
'topografia' varchar(45) DEFAULT NULL,
'mantenimiento' tinyint(1) DEFAULT NULL,
'zona_seguridad' tinyint(1) DEFAULT NULL,
'sis_contra_incendio' tinyint(1) DEFAULT NULL,
'cisterna' tinyint(1) DEFAULT NULL,
'cisterna_m3' float DEFAULT NULL,
PRIMARY KEY ('id_tipo_inmueble'),
KEY 'id_incidente_idx' ('id_incidente'),
CONSTRAINT 'id_incidente' FOREIGN KEY ('id_incidente') REFERENCES 'tbl_datos_grles' ('
id_incidente') ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

—Tabla Usuarios
CREATE TABLE 'tbl_usuarios' (
'user_id' int(11) NOT NULL,
'user_name' varchar(45) NOT NULL,
'user_desc' varchar(45) NOT NULL,
PRIMARY KEY ('user_id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

PHP

Validar.php

```
<?php
//Validar Usuario
$con = mysqli_connect("localhost", "root", "", "db_ras");

$userid = $_POST["user"];
$coordenadas=$_POST["coordenadas"];

$stmt = mysqli_prepare($con, "SELECT * FROM tbl-usuarios WHERE user_id = ?");
mysqli_stmt_bind_param($stmt, "i", $userid);
mysqli_stmt_execute($stmt);

mysqli_stmt_store_result($stmt);
mysqli_stmt_bind_result($stmt, $user_id, $user_name, $user_desc);

$response = array();
$response["success"] = false;
while(mysqli_stmt_fetch($stmt)){
    $response["success"] = true;
    $response["name"] = $user_name;
    $response["desc"] = $user_desc;
}
if($response["success"]==true)
{
    $stmt = mysqli_prepare($con, "INSERT INTO tbl_datos_grles (user_id, fecha,
coordenadas) VALUES (?,now(),?)");
    mysqli_stmt_bind_param($stmt, "is", $userid, $coordenadas);
    mysqli_stmt_execute($stmt);

    $stmt = mysqli_prepare($con, "SELECT 'AUTOINCREMENT'-1
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'db_ras'
AND TABLE_NAME = 'tbl_datos_grles'");
    mysqli_stmt_execute($stmt);
    mysqli_stmt_store_result($stmt);
    mysqli_stmt_bind_result($stmt, $id_incidente);

    while(mysqli_stmt_fetch($stmt)){
        $response["id_incidente"] = $id_incidente;
    }
    //Insertar en la tabla tbl_tipo_inmueble
    $stmt = mysqli_prepare($con, "INSERT INTO tbl_tipo_inmueble (id_incidente) VALUES
(?,?)");
    mysqli_stmt_bind_param($stmt, "i", $id_incidente);
    mysqli_stmt_execute($stmt);
    //Insertar en la tabla tbl_eval_danos
    $stmt = mysqli_prepare($con, "INSERT INTO tbl_eval_danos (id_incidente) VALUES (?);
");
    mysqli_stmt_bind_param($stmt, "i", $id_incidente);
    mysqli_stmt_execute($stmt);
}
echo json_encode($response);
?>
```

Datosgenerales.php

```
<?php
$con = mysqli_connect("localhost", "root", "", "db_ras");
$id_incidente = $_POST["id_incidente"];
$nombre_inmueble=$_POST["nombreInmueble"];
$nombre_calle=$_POST["nombreCalle"];
$numExt=$_POST["numExt"];
$numInt=$_POST["numInt"];
$colonia=$_POST["colonia"];
$cp=$_POST["cp"];
$delegacion=$_POST["delegacion"];
$estado=$_POST["estado"];
$referencia=$_POST["referencias"];
$nombreContacto=$_POST["nomContacto"];
$telefonoContacto=$_POST["telContacto"];
$correoContacto=$_POST["correoContacto"];
$stmt = mysqli_prepare($con, "UPDATE tbl_datos_grles SET nombre_inmueble=?, calle
=?, num_int=?, num_ext=?, colonia=?, delegacion=?, estado=?, codigo_postal=?, contacto_nombre
=?, contacto_telefono=?, contacto_correo=?, referencias_inmueble=? WHERE id_incidente=?");
mysqli_stmt_bind_param($stmt, "ssiiisssisissi", $nombre_inmueble, $nombre_calle,
$numInt, $numExt, $colonia, $delegacion, $estado, $cp, $nombreContacto, $telefonoContacto,
$correoContacto, $referencia, $id_incidente);
mysqli_stmt_execute($stmt);
$response = array();
$response["success"] = true;
echo json_encode($response);
?>
```

TipoInmueble.php

```
<?php
$con = mysqli_connect("localhost", "root", "", "db_ras");
$id_incidente = $_POST["id_incidente"];
$usoInm=$_POST["usoInmueble"];
$numNiv=$_POST["numNiveles"];
$numSot=$_POST["numSotanos"];
$pisosE=$_POST["pisosEst"];
$numOcup=$_POST["numOcupantes"];
$elev=$_POST["elevador"];
$escaleraE=$_POST["escaleraEmer"];
$anoConst=$_POST["anoConstruccion"];
$anoAfect=$_POST["anoDanoSevero"];
$anoRehabilita=$_POST["anoRehabilitacion"];
$topo=$_POST["topografia"];
$mant=$_POST["mantenimiento"];
$zonaSeg=$_POST["zonaSeguridad"];
$sisContInc=$_POST["sisContraInc"];
$cist=$_POST["cisterna"];
$cistM3=$_POST["cisternaM3"];
$stmt = mysqli_prepare($con, "UPDATE tbl_tipo_inmueble SET uso_inmueble=?,
num_niveles=?, num_sotanos=?, pisos_est=?, num_ocupantes=?, elevador=?, escalera_emer=?,
ano_construccion=?, ano_afectacion=?, ano_rehabilitacion=?, topografia=?, mantenimiento=?,
zona_seguridad=?, sis_contra_incendio=?, cisterna=?, cisterna_m3=? WHERE id_incidente=?");
mysqli_stmt_bind_param($stmt, "siiiiiiiiisiiiiidi", $usoInm, $numNiv, $numSot, $pisosE,
$numOcup, $elev, $escaleraE, $anoConst, $anoAfect, $anoRehabilita, $topo, $mant, $zonaSeg,
$sisContInc, $cist, $cistM3, $id_incidente);
mysqli_stmt_execute($stmt);
$response = array();
$response["success"] = true;
echo json_encode($response);
?>
```

Evaluaciondanos.php

```
<?php
$con = mysqli_connect("localhost", "root", "", "db_ras");

$id_incidente = $_POST["id_incidente"];
$grietasT = $_POST["grietasTerreno"];
$hundim=$_POST["hundimiento"];
$hundimcm=$_POST["hundimentocm"];
$inclinacionE=$_POST["inclinacionEdif"];
$inclinacionEPorcent=$_POST["inclinacionEdifPorcent"];
$losasC=$_POST["losasColapso"];
$losasGriMax=$_POST["losasGrietasMax"];
$losasFleMax=$_POST["losasFlechaMax"];
$grietaMaxmm=$_POST["grietasMaxmm"];
$fleMaxcm=$_POST["flechaMaxcm"];
$conFalla=$_POST["conexionFalla"];
$column=$_POST["columnas"];
$colAnchoMaxG=$_POST["colAnchoMaxGrieta"];
$colSeparacionE=$_POST["colSeparacionEstribo"];
$trabe=$_POST["trabes"];
$traAnchoMaxG=$_POST["traAnchoMaxGrieta"];
$traSeparacionE=$_POST["traSeparacionEstribo"];
$concre=$_POST["concreto"];
$conAnchoMaxG=$_POST["conAnchoMaxGrieta"];
$conSeparacionE=$_POST["conSeparacionEstribo"];
$mampos=$_POST["mampost"];
$mamAnchoMaxG=$_POST["mamAnchoMaxGrieta"];
$mamSeparacionE=$_POST["mamSeparacionEstribo"];
$numColumnasDanoS=$_POST["numColumnasDanoSevero"];
$totalColumnasE=$_POST["totalColumnasEntrepiso"];
$danoEstruct=$_POST["danoEstructura"];
$vidrio=$_POST["vidrios"];
$acabado=$_POST["acabados"];
$plafon=$_POST["plafones"];
$fachada=$_POST["fachadas"];
$bardasP=$_POST["bardasPretilas"];
$cucho=$_POST["cubos"];
$instalacion=$_POST["instalaciones"];

$stmt = mysqli_prepare($con, "UPDATE tbl_eval_danos SET grietas_terreno=?,
hundimientos=?,hundimientos_cm=?,inclinacion_edif=?,inclinacion_porcent=?,colapso=?,
grietas=?,grietas_mm=?,flecha=?,flecha_cm=?,conexiones_falla=?,columnas=?,
columnas_anchomm=?,columnas_sep_estribo=?,trabes=?,trabes_max=?,trabes_sep_estribo=?,
concreto=?,concreto_anchomm=?,concreto_sep_estribos=?,mampost=?,mampost_anchomm=?,
mampost_sep_estribos=?,entrepiso_critico_num_column=?,total_columnas_entrepiso=?,
nivel_dano_estruct_gral=?,vidrios=?,acabados=?,plafones=?,fachadas=?,bardas_pretilas=?,
cubos=?,instalaciones=? WHERE id_incidente=?");

mysqli_stmt_bind_param($stmt, "iiiiididisidsidsidsidiisiiiiiii", $grietasT,
$hundim, $hundimcm, $inclinacionE, $inclinacionEPorcent, $losasC, $losasGriMax, $grietaMaxmm,
$losasFleMax, $fleMaxcm, $conFalla, $column, $colAnchoMaxG, $colSeparacionE, $trabe,
$traAnchoMaxG, $traSeparacionE, $concre, $conAnchoMaxG, $conSeparacionE, $mampos,
$mamAnchoMaxG, $mamSeparacionE, $numColumnasDanoS, $totalColumnasE, $danoEstruct, $vidrio,
$acabado, $plafon, $fachada, $bardasP, $cucho, $instalacion, $id_incidente);

mysqli_stmt_execute($stmt);

$response = array();
$response["success"] = true;

echo json_encode($response);
?>
```

SavePhoto.php

```
<?php
$con = mysqli_connect("localhost", "root", "", "db_ras");
$response = array();
$idIncidente = $_POST["idIncidente"];
$descripcion=$_POST["descripcion"];
$imagen=$_POST["imagen"];

$date=new DateTime();
$nombre=date_format($date, 'dMyHms');
$path="ImagenesRAS/$nombre.jpg";

$url="http://localhost/$path";

file_put_contents($path, base64_decode($imagen));
$bytesArchivo=file_get_contents($path);

$stmt = mysqli_prepare($con, "insert into tbl_imagenes values (null,?,?,?);");
mysqli_stmt_bind_param($stmt, "sss", $idIncidente, $descripcion, $bytesArchivo);

if(mysqli_stmt_execute($stmt)){
    $response["success"] = true;
}
else{
    $response["success"] = false;
}
echo json_encode($response);
?>
```


Java

Variables.java

```
package com.example.luisd.ras;

import android.app.Application;
import android.os.Bundle;

import java.util.HashMap;

public class Variables extends Application {

    private String ID_INCIDENTE=null;
    private String USERNAME=null;
    private int USERID=0;
    private String URL="192.168.1.66";

    private HashMap<String, String> STATE2=null;
    private HashMap<String, String> STATE3=null;

    public void setSTATE2(HashMap<String, String> state2){
        STATE2=state2;
    }
    public HashMap<String, String> getState2(){
        return STATE2;
    }
    public void setSTATE3(HashMap<String, String> state3){
        STATE3=state3;
    }
    public HashMap<String, String> getState3(){
        return STATE3;
    }
    public String getID_INCIDENTE() {
        return ID_INCIDENTE;
    }
    public void setID_INCIDENTE(String id_incidente){
        ID_INCIDENTE=id_incidente;
    }
    public String getUsername(){
        return USERNAME;
    }
    public void setUsername(String username){
        USERNAME=username;
    }
    public int getUserID(){
        return USERID;
    }
    public void setUserID(int userid){
        USERID=userid;
    }
    public String getURL(){return URL;}
}
```

LoginActivity.java

```
package com.example.luisd.ras;
import android.Manifest;
import android.annotation.SuppressLint;
import android.app.Application;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Build;
import android.os.Looper;
import android.provider.Settings;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AlertDialog;
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.EditText;
import android.widget.Toast;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.IOException;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import pub.devrel.easypermissions.AfterPermissionGranted;
import pub.devrel.easypermissions.EasyPermissions;
public class LoginActivity extends AppCompatActivity implements View.OnClickListener {

    private static final int REQUEST_LOCATION_PERMISSION = 1;
    Button salir , initReg;
    EditText txtUser;
    private String coordenadas = "Sin coordenadas!";
    private LocationManager locationManager;
    private LocationListener locationListener = new LocationListener() {
        @Override
        public void onLocationChanged(Location location) {

            // Este metodo se ejecuta cada vez que el GPS recibe nuevas coordenadas
            // debido a la deteccion de un cambio de ubicacion

            String Text = "Mi ubicacion actual es: " + "\n Lat = "
                + location.getLatitude() + "\n Long = " + location.getLongitude();
            coordenadas=Double.toString(location.getLatitude()+","+Double.toString(location
                .getLongitude());
            System.out.println(Text);
        }
        @Override
        public void onStatusChanged(String provider , int status , Bundle extras) {

        }
    }
}
```

```

@Override
public void onProviderEnabled(String provider) {

}

@Override
public void onProviderDisabled(String provider) {
    Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
    startActivity(intent);
}
};
private ProgressDialog progressDialog;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    requestLocationPermission();
    getCoordinate();

    salir = (Button) findViewById(R.id.btnSalir);
    initReg = (Button) findViewById(R.id.btnLogin);
    txtUser = (EditText) findViewById(R.id.txtUser);

    initReg.setOnClickListener(this);
    salir.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showMessage();
        }
    });
}
public void showMessage() {

    AlertDialog.Builder builder = new AlertDialog.Builder(this);

    builder.setTitle("Salir");
    builder.setMessage("Realmente deseas salir?");

    builder.setPositiveButton("SI", new DialogInterface.OnClickListener() {

        public void onClick(DialogInterface dialog, int which) {

            int p = android.os.Process.myPid();
            dialog.dismiss();
            android.os.Process.killProcess(p);

        }
    });

    builder.setNegativeButton("NO", new DialogInterface.OnClickListener() {

        @Override
        public void onClick(DialogInterface dialog, int which) {

            dialog.dismiss();

        }
    });

    AlertDialog alert = builder.create();
    alert.show();
}
@Override
public void onClick(View v) {
    final String user = txtUser.getText().toString();
    getCoordinate();
}

```

```

    if (user.equals("")) {
        AlertDialog.Builder builder = new AlertDialog.Builder(LoginActivity.this);
        builder.setMessage("Ingreso capturista!").setNegativeButton("Error", null)
            .create().show();
        return;
    }
    Cargando();
    final Response.Listener<String> responseListener = new Response.Listener<String>() {

        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonResponse = new JSONObject(response);
                boolean success = jsonResponse.getBoolean("success");
                if (success) {
                    progressDialog.dismiss();
                    String username = jsonResponse.getString("name");
                    String id_incidente = jsonResponse.getString("id_incidente");
                    //Se guardan las variables que estaran presentes en toda la
ejecucion de la aplicacion
                    ((Variables) getApplicationContext()).setID_INCIDENTE(id_incidente);
                    ((Variables) getApplicationContext()).setUSERNAME(username);
                    System.out.println("NOMBRE DE USUARIO: " + username);
                    System.out.println("ID DEL INCIDENTE: " + id_incidente);
                    AlertDialog.Builder builder = new AlertDialog.Builder(LoginActivity
this);
                    builder.setMessage("Bienvenido " + username).setNegativeButton("OK",
null)
                        .create().show();
                    Intent intent = new Intent(LoginActivity.this, GralActivity.class);
                    LoginActivity.this.startActivity(intent);

                } else {
                    progressDialog.dismiss();
                    AlertDialog.Builder builder = new AlertDialog.Builder(LoginActivity
this);
                    builder.setMessage("Capturista Erroneo").setNegativeButton("Error",
null)
                        .create().show();
                }
            } catch (JSONException e) {
                progressDialog.dismiss();
                e.printStackTrace();
                AlertDialog.Builder builder = new AlertDialog.Builder(LoginActivity.this
);
                builder.setMessage("No hay conexi n con la BD").setNegativeButton("
Error", null)
                    .create().show();
            }
        }
    };
    LoginRequest loginRequest = new LoginRequest(user, coordenadas, responseListener);
    RequestQueue queue = Volley.newRequestQueue(LoginActivity.this);
    queue.add(loginRequest);
}

public void getCoordinate() {
    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.
ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED && ActivityCompat.
checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.
PERMISSION_GRANTED) {
        Toast.makeText(this, "No cuento con permisos!", Toast.LENGTH_SHORT).show();
        return;
    }
    locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, (

```

```

LocationListener) locationListener);
    locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, (
LocationListener) locationListener);
}
private void Cargando(){
    progressDialog=new ProgressDialog(this);
    progressDialog.setMessage("Cargando");
    progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
    progressDialog.setIndeterminate(true);
    progressDialog.setProgress(0);
    progressDialog.show();
}
@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    // Forward results to EasyPermissions
    EasyPermissions.onRequestPermissionsResult(requestCode, permissions, grantResults,
this);
}
@AfterPermissionGranted(REQUEST_LOCATION_PERMISSION)
public void requestLocationPermission() {
    String[] perms = {Manifest.permission.ACCESS_FINE_LOCATION};
    if(EasyPermissions.hasPermissions(this, perms)) {
        Toast.makeText(this, "Permisos Correctos!", Toast.LENGTH_SHORT).show();
    }
    else {
        EasyPermissions.requestPermissions(this, "Please grant the location permission",
REQUEST_LOCATION_PERMISSION, perms);
    }
}
}
}

```

LoginRequest.java

```
package com.example.luisd.ras;

import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;
import java.util.HashMap;
import java.util.Map;

public class LoginRequest extends StringRequest {
    private static final Variables var=new Variables();
    private static final String REGISTER_REQUEST_URL="http://"+var.getURL()+"/Validar.php";
    private Map<String,String> params;

    public LoginRequest(String user,String coordenadas,Response.Listener<String> listener) {
        super(Method.POST,REGISTER_REQUEST_URL, listener, null);
        params=new HashMap<>();
        String x=String.valueOf(user);
        System.out.println("Esto es x="+x);
        params.put("user",x);
        params.put("coordenadas",coordenadas);
    }
    @Override
    public Map<String,String> getParams(){return params;}
}
```

GralActivity.java

```
package com.example.luisd.ras;

import android.content.DialogInterface;
import android.content.Intent;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;

public class GralActivity extends AppCompatActivity {
    EditText nombreInmueble;
    EditText nombreCalle;
    EditText numExt;
    EditText numInt;
    EditText colonia;
    EditText cp;
    EditText delegacion;
    Spinner estado;
    EditText referencias;
    EditText nomContacto;
    EditText telContacto;
    EditText correoContacto;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_gral);
        if (((Variables) getApplicationContext()).getUSERNAME() == null) { finish(); }
        Spinner spinner = (Spinner) findViewById(R.id.spEstado);
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
            R.array.Estados, android.R.layout.simple_spinner_item);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinner.setAdapter(adapter);
        //Llenar lista con los datos del usuario e incidente
        TextView tvIncidente;
        TextView tvUsername;
        tvUsername=(TextView) findViewById(R.id.tvUsername);
        tvIncidente=(TextView) findViewById(R.id.tvIdIncidente);
        String inc=((Variables) getApplicationContext()).getID.INCIDENTE();
        tvIncidente.setText(" Incidente: INC"+inc);
        tvUsername.setText(" Usuario: "+((Variables) getApplicationContext()).getUSERNAME());
        //Crear Vista CameraActivity
        Button btnCamara=(Button) findViewById(R.id.btnCamara);
        Button btnSigTpo=(Button) findViewById(R.id.btnSigTpo);
        btnCamara.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showActivityCamera();
            }
        });
    }
};
```

```

        btnSiguiente.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                registrar();
            }
        });
    }
    @Override
    public void onBackPressed() {

        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Salir");
        builder.setMessage("No haz concluido el reporte!\n Realmente deseas salir?");
        builder.setPositiveButton("SI", new DialogInterface.OnClickListener() {

            public void onClick(DialogInterface dialog, int which) {
                int p = android.os.Process.myPid();
                android.os.Process.killProcess(p);
                dialog.dismiss();
            }
        });

        builder.setNegativeButton("NO", new DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
            }
        });

        AlertDialog alert = builder.create();
        alert.show();
    }
    private void showActivityCamera(){Intent intent = new Intent(GralActivity.this,
    CameraActivity.class);
    GralActivity.this.startActivity(intent);
    }
    private void registrar(){
        if(!validarCampos()){
            Toast.makeText(this, "Faltan datos por capturar!", Toast.LENGTHSHORT).show();
            return;}
        nombreInmueble=findViewById(R.id.etNombre);
        nombreCalle=findViewById(R.id.etCalle);
        numExt=findViewById(R.id.etNumExt);
        numInt=findViewById(R.id.etNumInt);
        colonia=findViewById(R.id.etColonia);
        cp=findViewById(R.id.etCP);
        delegacion=findViewById(R.id.etDelegacion);
        estado=findViewById(R.id.spEstado);
        referencias=findViewById(R.id.etReferencias);
        nomContacto=findViewById(R.id.etNombreContacto);
        telContacto=findViewById(R.id.etTelefonoContacto);
        correoContacto=findViewById(R.id.etCorreoContacto);

        final Response.Listener<String> responseListener= new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {

```



```

        try {
            JSONObject jsonResponse = new JSONObject(response);
            boolean success = jsonResponse.getBoolean("success");
            if(success){
                System.out.println("Registro gral exitoso");
                Intent intentTipo;
                intentTipo= new Intent(GralActivity.this, TipoActivity.class);
                GralActivity.this.startActivity(intentTipo);
            }
            else{
                AlertDialog.Builder build;
                build=new AlertDialog.Builder(GralActivity.this);
                build.setMessage("No se pudo registrar en la BD!");
                build.setPositiveButton("OK", new DialogInterface.OnClickListener()
{
                    public void onClick(DialogInterface dialog, int which) {
                        dialog.dismiss();
                    }
                });
                AlertDialog alert = build.create();
                alert.show();
            }
        }catch (JSONException e) {
            e.printStackTrace();
            AlertDialog.Builder builder = new AlertDialog.Builder(GralActivity.this)
;
            builder.setMessage("No hay conexi n con la BD").setNegativeButton("
Error", null)
                .create().show();
        }
    };
    GralRequest gralRequest = new GralRequest(((Variables) getApplicationContext()).
getID.INCIDENTE(), nombreInmueble.getText().toString(), nombreCalle.getText().toString()
, numExt.getText().toString(), numInt.getText().toString(), colonia.getText().
toString(), cp.getText().toString(),
    delegacion.getText().toString(), estado.getSelectedItem().toString(),
    referencias.getText().toString(),
    nomContacto.getText().toString(), telContacto.getText().toString(),
    correoContacto.getText().toString(), responseListener);
    RequestQueue queue = Volley.newRequestQueue(GralActivity.this);
    queue.add(gralRequest);
}
private boolean validarCampos(){
    nombreInmueble=findViewById(R.id.etNombre);
    nombreCalle=findViewById(R.id.etCalle);
    numExt=findViewById(R.id.etNumExt);
    colonia=findViewById(R.id.etColonia);
    delegacion=findViewById(R.id.etDelegacion);
    nomContacto=findViewById(R.id.etNombreContacto);
    telContacto=findViewById(R.id.etTelefonoContacto);
    correoContacto=findViewById(R.id.etCorreoContacto);
    if(nombreInmueble.getText().toString().equals("") || nombreCalle.getText().toString().
equals("") ||
        numExt.getText().toString().equals("") || colonia.getText().toString().equals(
"") || delegacion.getText().toString().equals("") ||
        nomContacto.getText().toString().equals("") || telContacto.getText().toString
().equals("") || correoContacto.getText().toString().equals(""))
        {return false;}
    else return true;
}
}
}

```

GralRequest.java

```
package com.example.luisd.ras;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;
import java.util.HashMap;
import java.util.Map;

public class GralRequest extends StringRequest {
    private static final Variables var=new Variables();
    private static final String REGISTER_REQUEST_URL="http://"+var.getURL()+"/Datosgenerales.php";
    private Map<String,String> params;

    public GralRequest(String id_incidente,String nombreInmueble,String nombreCalle,String numExt,String numInt,String colonia,String cp,String delegacion,String estado,String referencias,String nomContacto,String telContacto,String correoContacto,Response.Listener<String> listener) {
        super(Request.Method.POST,REGISTER_REQUEST_URL, listener, null);
        params=new HashMap<>();
        params.put("id_incidente",id_incidente);
        params.put("nombreInmueble",nombreInmueble);
        params.put("nombreCalle",nombreCalle);
        params.put("numExt", numExt);
        params.put("numInt", numInt);
        params.put("colonia", colonia);
        params.put("cp", cp);
        params.put("delegacion", delegacion);
        params.put("estado", estado);
        params.put("referencias", referencias);
        params.put("nomContacto", nomContacto);
        params.put("telContacto", telContacto);
        params.put("correoContacto", correoContacto);
    }
    @Override
    public Map<String,String> getParams(){return params;}
}
```

TipoActivity.java

```
package com.example.luisd.ras;

import android.content.DialogInterface;
import android.content.Intent;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;

public class TipoActivity extends AppCompatActivity {
    HashMap<String, String> state2;
    String usoInm, numNiv, numSot, pisosEst, numOcu, elev, escaleraEmer, anoConstruct, anoSev,
    anoReha, topo, mante, zonaSeg, sisInce, cister, cisternm3;
    Spinner usoInmueble;
    EditText numNiveles;
    EditText numSotanos;
    EditText numPisosEst;
    EditText numOcupantes;
    CheckBox elevador;
    CheckBox escalera;
    EditText anoConstruccion;
    EditText anoDanoSevero;
    EditText anoRehabilitacion;
    Spinner tipoTopografia;
    CheckBox mantenimiento;
    CheckBox zonaSeguridad;
    CheckBox sistemaIncendio;
    CheckBox cisterna;
    EditText cisternaM3;
    @Override
    protected void onCreate(final Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tipo);
        if(((Variables) getApplicationContext()).getUSERNAME()==null){finish();}
        TextView tvIncidente;
        TextView tvUsername;
        tvUsername=findViewById(R.id.tvUsername);
        tvIncidente=findViewById(R.id.tvIdIncidente);
        String inc=((Variables) getApplicationContext()).getID.INCIDENTE();
        tvIncidente.setText(" Incidente: INC"+inc);
        tvUsername.setText(" Usuario: "+((Variables) getApplicationContext()).getUSERNAME());
        Button btnAnterior=findViewById(R.id.btnBackGral);
        Button btnSiguiete=findViewById(R.id.btnSigEval);
        Button btnCamera=findViewById(R.id.btnCamara);
        state2=((Variables) getApplicationContext()).getState2();
        usoInmueble=findViewById(R.id.spUsoInm);
        numNiveles=findViewById(R.id.etNumNiveles);
        numSotanos=findViewById(R.id.etNumSotanos);
        numPisosEst=findViewById(R.id.etPisosEstaciona);
        numOcupantes=findViewById(R.id.etNumOcupantes);
    }
}
```

```

elevator=findViewById(R.id.cbElevador);
escalera=findViewById(R.id.cbEscaleraEmer);
anoConstruccion=findViewById(R.id.etAnoConstruc);
anoDanoSevero=findViewById(R.id.etAnoDanoSevero);
anoRehabilitacion=findViewById(R.id.etAnoRehabilita);
tipoTopografia=findViewById(R.id.spTopografia);
mantenimiento=findViewById(R.id.cbMantenimiento);
zonaSeguridad=findViewById(R.id.cbZonaSeguridad);
sistemaIncendio=findViewById(R.id.cbSistemaIncendio);
cisterna=findViewById(R.id.cbCisterna);
cisternaM3=findViewById(R.id.etCisternaM3);
if(state2!=null)
{
    CargarValores();
}
btnAnterior.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        onBackPressed();
    }
});
btnSiguiente.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        registrar();
    }
});
btnAnterior.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Regresar();
    }
});
btnCamera.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showActivityCamera();
    }
});
}
public void onBackPressed() {
    Regresar();
}
public void Regresar(){

    usoInm=usoInmueble.getSelectedItem().toString();
    numNiv=numNiveles.getText().toString();
    numSot=numSotanos.getText().toString();
    pisosEst=numPisosEst.getText().toString();
    numOcu=numOcupantes.getText().toString();
    if(elevator.isChecked()){elev="1";} else{elev="0";}
    if(escalera.isChecked()){escaleraEmer="1";} else{escaleraEmer="0";}
    anoConstruct=anoConstruccion.getText().toString();
    anoSev=anoDanoSevero.getText().toString();
    anoReha=anoRehabilitacion.getText().toString();
    topo=tipoTopografia.getSelectedItem().toString();
    if(mantenimiento.isChecked()){mante="1";} else{mante="0";}
    if(zonaSeguridad.isChecked()){zonaSeg="1";} else{zonaSeg="0";}
    if(sistemaIncendio.isChecked()){sisInce="1";} else{sisInce="0";}
    if(cisterna.isChecked()){cister="1";} else{cister="0";}
    cisternm3=cisternaM3.getText().toString();
    state2= new HashMap<>();
    state2.put("usoInm",usoInm);
    state2.put("numNiv",numNiv);

```

```

        state2.put("numSot", numSot);
        state2.put("pisosEst", pisosEst);
        state2.put("numOcu", numOcu);
        state2.put("elev", elev);
        state2.put("escaleraEmer", escaleraEmer);
        state2.put("anoConstruct", anoConstruct);
        state2.put("anoSev", anoSev);
        state2.put("anoReha", anoReha);
        state2.put("topo", topo);
        state2.put("mante", mante);
        state2.put("zonaSeg", zonaSeg);
        state2.put("sisInce", sisInce);
        state2.put("cister", cister);
        state2.put("cistern3", cistern3);

        ((Variables) getApplicationContext()).setState2(state2);
        finish();
    }
    public void CargarValores(){
        state2=((Variables) getApplicationContext()).getState2();
        String encontrarValue = state2.get("usoinm");
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.usosinm, android.R.layout.simple_spinner_item);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        usoInmueble.setAdapter(adapter);
        if (encontrarValue != null) {
            int spinnerPosition = adapter.getPosition(encontrarValue);
            usoInmueble.setSelection(spinnerPosition);
        }
        numNiveles.setText(state2.get("numNiv"));
        numSotanos.setText(state2.get("numSot"));
        numPisosEst.setText(state2.get("pisosEst"));
        numOcupantes.setText(state2.get("numOcu"));
        if (state2.get("elev").equals("1")){elevador.setChecked(true);}
        if (state2.get("escaleraEmer").equals("1")){escalera.setChecked(true);}
        anoConstruccion.setText(state2.get("anoConstruct"));
        anoDanoSevero.setText(state2.get("anoSev"));
        anoRehabilitacion.setText(state2.get("anoReha"));
        encontrarValue=state2.get("topo");
        adapter=ArrayAdapter.createFromResource(this, R.array.topografia, android.R.layout.simple_spinner_item);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        tipoTopografia.setAdapter(adapter);
        if (encontrarValue!=null){
            int spinnerPosition=adapter.getPosition(encontrarValue);
            tipoTopografia.setSelection(spinnerPosition);
        }
        if (state2.get("mante").equals("1")){mantenimiento.setChecked(true);}
        if (state2.get("zonaSeg").equals("1")){zonaSeguridad.setChecked(true);}
        if (state2.get("sisInce").equals("1")){sistemaIncendio.setChecked(true);}
        if (state2.get("cister").equals("1")){cisterna.setChecked(true);}
        cisternaM3.setText(state2.get("cistern3"));
    }
    private void showActivityCamera(){Intent intent = new Intent(TipoActivity.this, CameraActivity.class);
        TipoActivity.this.startActivity(intent);
    }
    private void registrar(){
        usoInmueble=findViewById(R.id.spUsoInm);
        numNiveles=findViewById(R.id.etNumNiveles);
        numSotanos=findViewById(R.id.etNumSotanos);
        numPisosEst=findViewById(R.id.etPisosEstaciona);
        numOcupantes=findViewById(R.id.etNumOcupantes);
        elevador=findViewById(R.id.cbElevador);
    }

```

```

        escalera=findViewById(R.id.cbEscaleraEmer);
        anoConstruccion=findViewById(R.id.etAnoConstruc);
        anoDanoSevero=findViewById(R.id.etAnoDanoSevero);
        anoRehabilitacion=findViewById(R.id.etAnoRehabilita);
        tipoTopografia=findViewById(R.id.spTopografia);
        mantenimiento=findViewById(R.id.cbMantenimiento);
        zonaSeguridad=findViewById(R.id.cbZonaSeguridad);
        sistemaIncendio=findViewById(R.id.cbSistemaIncendio);
        cisterna=findViewById(R.id.cbCisterna);
        cisternaM3=findViewById(R.id.etCisternaM3);

        final Response.Listener<String> responseListener= new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonResponse = new JSONObject(response);
                    boolean success = jsonResponse.getBoolean("success");
                    if(success){
                        System.out.println("Registro tipo exitoso");
                        Intent intentEval=new Intent(TipoActivity.this, EvaluacionActivity.
class);
                        TipoActivity.this.startActivity(intentEval);
                    }else{
                        AlertDialog.Builder build;
                        build=new AlertDialog.Builder(TipoActivity.this);
                        build.setMessage("No se pudo registrar en la BD!");
                        build.setPositiveButton("OK", new DialogInterface.OnClickListener()
{
                            public void onClick(DialogInterface dialog, int which) {
                                dialog.dismiss();
                            }
                        });
                        AlertDialog alert = build.create();
                        alert.show();
                    }
                }catch (JSONException e) {
                    e.printStackTrace();
                    AlertDialog.Builder builder = new AlertDialog.Builder(TipoActivity.this)
;
                    builder.setMessage("No hay conexi n con la BD").setNegativeButton("
Error", null)
                        .create().show();
                }
            }
        };
        if (elevador.isChecked()){elev="1";} else{elev="0";}
        if (escalera.isChecked()){escaleraEmer="1";} else{escaleraEmer="0";}
        if (mantenimiento.isChecked()){mante="1";} else{mante="0";}
        if (zonaSeguridad.isChecked()){zonaSeg="1";} else{zonaSeg="0";}
        if (sistemaIncendio.isChecked()){sisInce="1";} else{sisInce="0";}
        if (cisterna.isChecked()){cister="1";} else{cister="0";}
        TipoRequest tipoRequest = new TipoRequest(((Variables) getApplicationContext()).
getID.INCIDENTE(), usoInmueble.getSelectedItem().toString(),
            numNiveles.getText().toString(), numSotanos.getText().toString(), numPisosEst.
getText().toString(), numOcupantes.getText().toString(),
            elev, escaleraEmer, anoConstruccion.getText().toString(), anoDanoSevero.getText
().toString(), anoRehabilitacion.getText().toString(),
            tipoTopografia.getSelectedItem().toString(), mante, zonaSeg, sisInce, cister,
            cisternaM3.getText().toString(), responseListener);
        RequestQueue queue = Volley.newRequestQueue(TipoActivity.this);
        queue.add(tipoRequest);
    }
}

```

TipoRequest.java

```
package com.example.luisd.ras;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;

import java.util.HashMap;
import java.util.Map;

public class TipoRequest extends StringRequest {
    private static final Variables var=new Variables();
    private static final String REGISTER_REQUEST_URL="http://"+var.getURL()+"/TipoInmueble.
    php";
    private Map<String,String> params;

    public TipoRequest(String id_incidente, String usoInmueble, String numNiveles, String
    numSotanos, String pisosEst,
        String numOcupantes, String elevador, String escaleraEmergencia,
    String anoConstruccion, String anoDanoSevero, String anoRehabilitacion,
        String topografia, String mantenimiento,String zonaSeguridad,String
    sisContraInc,String cisterna,String cisternaM3
        , Response.Listener<String> listener) {
        super(Request.Method.POST,REGISTER_REQUEST_URL, listener, null);
        params=new HashMap<>();

        params.put("id_incidente",id_incidente);
        params.put("usoInmueble",usoInmueble);
        params.put("numNiveles",numNiveles);
        params.put("numSotanos",numSotanos);
        params.put("pisosEst",pisosEst);
        params.put("numOcupantes",numOcupantes);
        params.put("elevador",elevador);
        params.put("escaleraEmer",escaleraEmergencia);
        params.put("anoConstruccion",anoConstruccion);
        params.put("anoDanoSevero",anoDanoSevero);
        params.put("anoRehabilitacion",anoRehabilitacion);
        params.put("topografia",topografia);
        params.put("mantenimiento",mantenimiento);
        params.put("zonaSeguridad",zonaSeguridad);
        params.put("sisContraInc",sisContraInc);
        params.put("cisterna",cisterna);
        params.put("cisternaM3",cisternaM3);

    }
    @Override
    public Map<String,String> getParams(){return params;}
}
```

EvaluacionActivity.java

```
package com.example.luisd.ras;

import android.content.DialogInterface;
import android.content.Intent;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.HashMap;
import java.util.Map;

public class EvaluacionActivity extends AppCompatActivity {
    //Strings de los CB
    String gri ,hun ,incEdif , losasCol , losasGri , losasFle , conFalla , vid , aca , pla , facha , bardasPre
        , cub , instala ;
    //Hashmap
    HashMap<String , String> state ;

    CheckBox grietas ;
    CheckBox hundimiento ;
    EditText hundimientocm ;
    CheckBox inclinacionEdif ;
    EditText inclinacionEdifPorcent ;
    CheckBox losasColapso ;
    CheckBox losasGrietasMax ;
    CheckBox losasFlechaMax ;
    EditText grietasMaxmm ;
    EditText flechaMaxcm ;
    CheckBox conexionFalla ;
    Spinner columnas ;
    EditText colAnchoMaxGrieta ;
    EditText colSeparacionEstribo ;
    Spinner trabes ;
    EditText traAnchoMaxGrieta ;
    EditText traSeparacionEstribo ;
    Spinner concreto ;
    EditText conAnchoMaxGrieta ;
    EditText conSeparacionEstribo ;
    Spinner mampost ;
    EditText mamAnchoMaxGrieta ;
    EditText mamSeparacionEstribo ;
    EditText numColumnasDanoSevero ;
    EditText totalColumnasEntrepiso ;
    Spinner danoEstructura ;
    CheckBox vidrios ;
    CheckBox acabados ;
    CheckBox plafones ;
    CheckBox fachadas ;
    CheckBox bardasPretilas ;
    CheckBox cubos ;
    CheckBox instalaciones ;
```



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_evaluacion);
    if(((Variables) getApplicationContext()).getUSERNAME()==null){finish();}
    TextView tvIncidente=(TextView)findViewById(R.id.tvIdIncidente);
    TextView tvUsername=findViewById(R.id.tvUsername);
    String inc=((Variables) getApplicationContext()).getID_INCIDENTE();
    tvIncidente.setText(" Incidente: INC"+inc);
    tvUsername.setText(" Usuario: "+((Variables) getApplicationContext()).getUSERNAME());
    //Asignar by id
    UpdateData();
    state=((Variables) getApplicationContext()).getState3();
    //Recuperar los datos en caso de regresar despues de un return
    if(state!=null)
    {
        System.out.println("Se cargan valores");
        CargarValores();
    }else System.out.println("No hay valores");
    Button btnCamera=(Button)findViewById(R.id.btnCamara);
    Button back=findViewById(R.id.btnBack);
    Button btnFinal=findViewById(R.id.btnFinish);
    btnFinal.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Terminar();
        }
    });
    back.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Regresar();
        }
    });
    btnCamera.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showActivityCamera();
        }
    });
}
private void Terminar(){
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Datos Correctos");
    builder.setMessage(" Todos los datos son correctos?\n Guardar datos?");
    builder.setPositiveButton("SI", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            //Guardar antes en la base de datos
            dialog.dismiss();
            Registrar();
        }
    });
    builder.setNegativeButton("NO", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
        }
    });
    AlertDialog alert = builder.create();
    alert.show();
}
}

```

```

private void CargarValores(){
    ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.
danoestruct , android.R.layout.simple_spinner_item);
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

    //Spinners
    String encontrarValue = state.get("columnas");
    columnas.setAdapter(adapter);
    if (encontrarValue != null) {
        int spinnerPosition = adapter.getPosition(encontrarValue);
        columnas.setSelection(spinnerPosition);
    }
    encontrarValue = state.get("trabes");
    trabes.setAdapter(adapter);
    if (encontrarValue != null) {
        int spinnerPosition = adapter.getPosition(encontrarValue);
        trabes.setSelection(spinnerPosition);
    }
    encontrarValue = state.get("concreto");
    concreto.setAdapter(adapter);
    if (encontrarValue != null) {
        int spinnerPosition = adapter.getPosition(encontrarValue);
        concreto.setSelection(spinnerPosition);
    }
    encontrarValue = state.get("mampost");
    mampost.setAdapter(adapter);
    if (encontrarValue != null) {
        int spinnerPosition = adapter.getPosition(encontrarValue);
        mampost.setSelection(spinnerPosition);
    }
    adapter = ArrayAdapter.createFromResource(this, R.array.niveldano , android.R.layout.
simple_spinner_item);
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    encontrarValue = state.get("danoEstructura");
    danoEstructura.setAdapter(adapter);
    if (encontrarValue != null) {
        int spinnerPosition = adapter.getPosition(encontrarValue);
        danoEstructura.setSelection(spinnerPosition);
    }
}
//Fin Spinners
//Checkbox
if (state.get("grietas").equals("1")){grietas.setChecked(true);}
if (state.get("hundimiento").equals("1")){hundimiento.setChecked(true);}
if (state.get("inclinacionEdif").equals("1")){inclinacionEdif.setChecked(true);}
if (state.get("losasColapso").equals("1")){losasColapso.setChecked(true);}
if (state.get("losasGrietasMax").equals("1")){losasGrietasMax.setChecked(true);}
if (state.get("losasFlechaMax").equals("1")){losasFlechaMax.setChecked(true);}
if (state.get("conexionFalla").equals("1")){conexionFalla.setChecked(true);}
if (state.get("vidrios").equals("1")){vidrios.setChecked(true);}
if (state.get("acabados").equals("1")){acabados.setChecked(true);}
if (state.get("plafones").equals("1")){plafones.setChecked(true);}
if (state.get("fachadas").equals("1")){fachadas.setChecked(true);}
if (state.get("bardasPretilas").equals("1")){bardasPretilas.setChecked(true);}
if (state.get("cubos").equals("1")){cubos.setChecked(true);}
if (state.get("instalaciones").equals("1")){instalaciones.setChecked(true);}
//Fin Checkboxes
//EditText
hundimientocm.setText(state.get("hundimientocm"));
inclinacionEdifPorcent.setText(state.get("inclinacionEdifPorcent"));
grietasMaxmm.setText(state.get("grietasMaxmm"));
flechaMaxcm.setText(state.get("flechaMaxcm"));
colAnchoMaxGrieta.setText(state.get("colAnchoMaxGrieta"));
colSeparacionEstribo.setText(state.get("colSeparacionEstribo"));
traAnchoMaxGrieta.setText(state.get("traAnchoMaxGrieta"));

```

```

traSeparacionEstribo.setText(state.get("traSeparacionEstribo"));
conAnchoMaxGrieta.setText(state.get("conAnchoMaxGrieta"));
conSeparacionEstribo.setText(state.get("conSeparacionEstribo"));
mamAnchoMaxGrieta.setText(state.get("mamAnchoMaxGrieta"));
mamSeparacionEstribo.setText(state.get("mamSeparacionEstribo"));
numColumnasDanoSevero.setText(state.get("numColumnasDanoSevero"));
totalColumnasEntrepiso.setText(state.get("totalColumnasEntrepiso"));
//Fin EditText
}
public void Regresar(){
    UpdateData();
    state =new HashMap<>();
    //Guardar los valores en el map;
    state.put("grietas", gri);
    state.put("hundimiento", hun);
    state.put("hundimentocm", hundimentocm.getText().toString());
    state.put("inclinacionEdif", incEdif);
    state.put("inclinacionEdifPorcent", inclinacionEdifPorcent.getText().toString());
    state.put("grietasMaxmm", grietasMaxmm.getText().toString());
    state.put("losasColapso", losasCol);
    state.put("losasGrietasMax", losasGri);
    state.put("losasFlechaMax", losasFle);
    state.put("flechaMaxcm", flechaMaxcm.getText().toString());
    state.put("conexionFalla", conFalla);
    state.put("columnas", columnas.getSelectedItem().toString());
    state.put("colAnchoMaxGrieta", colAnchoMaxGrieta.getText().toString());
    state.put("colSeparacionEstribo", colSeparacionEstribo.getText().toString());
    state.put("trabes", trabes.getSelectedItem().toString());
    state.put("traAnchoMaxGrieta", traAnchoMaxGrieta.getText().toString());
    state.put("traSeparacionEstribo", traSeparacionEstribo.getText().toString());
    state.put("concreto", concreto.getSelectedItem().toString());
    state.put("conAnchoMaxGrieta", conAnchoMaxGrieta.getText().toString());
    state.put("conSeparacionEstribo", conSeparacionEstribo.getText().toString());
    state.put("mampost", mampost.getSelectedItem().toString());
    state.put("mamAnchoMaxGrieta", mamAnchoMaxGrieta.getText().toString());
    state.put("mamSeparacionEstribo", mamSeparacionEstribo.getText().toString());
    state.put("numColumnasDanoSevero", numColumnasDanoSevero.getText().toString());
    state.put("totalColumnasEntrepiso", totalColumnasEntrepiso.getText().toString());
    state.put("danoEstructura", danoEstructura.getSelectedItem().toString());
    state.put("vidrios", vid);
    state.put("acabados", aca);
    state.put("plafones", pla);
    state.put("fachadas", facha);
    state.put("bardasPretilas", bardasPre);
    state.put("cubos", cub);
    state.put("instalaciones", instala);
    ((Variables)getApplicationContext()).setState3(state);
    System.out.println("Se guardan valores");
    finish();
}
private void showActivityCamera(){Intent intent = new Intent(EvaluacionActivity.this,
CameraActivity.class);
    EvaluacionActivity.this.startActivity(intent);
}
private void UpdateData(){
    grietas=findViewById(R.id.cbGrietasTerreno);
    hundimiento=findViewById(R.id.cbHundimiento);
    hundimentocm=findViewById(R.id.etHundimiento);
    inclinacionEdif=findViewById(R.id.cbInclinacion);
    inclinacionEdifPorcent=findViewById(R.id.etInclinacion);
    losasColapso=findViewById(R.id.cbColapso);
    losasGrietasMax=findViewById(R.id.cbGrietasMax);
    losasFlechaMax=findViewById(R.id.cbFlechaMax);
    grietasMaxmm=findViewById(R.id.etGrietasMax);
}

```

```

flechaMaxcm=findViewById(R.id.etFlechaMax);
conexionFalla=findViewById(R.id.cbFalla);
columnas=findViewById(R.id.spColumnas);
colAnchoMaxGrieta=findViewById(R.id.etColumnAnchoMM);
colSeparacionEstribo=findViewById(R.id.etColumnSepEstribo);
trabes=findViewById(R.id.spTrabe);
traAnchoMaxGrieta=findViewById(R.id.etTrabeAnchoMM);
traSeparacionEstribo=findViewById(R.id.etTrabeSepEstribo);
concreto=findViewById(R.id.spConcreto);
conAnchoMaxGrieta=findViewById(R.id.etConcretAnchoMM);
conSeparacionEstribo=findViewById(R.id.etConcretSepEstribo);
mampost=findViewById(R.id.spMampost);
mamAnchoMaxGrieta=findViewById(R.id.etMampostAnchoMM);
mamSeparacionEstribo=findViewById(R.id.etMampostSepEstribo);
numColumnasDanoSevero=findViewById(R.id.etColumnDanoSevero);
totalColumnasEntrepiso=findViewById(R.id.etColumnDanoEntrepiso);
danoEstructura=findViewById(R.id.spDanoEstructural);
vidrios=findViewById(R.id.cbVidrios);
acabados=findViewById(R.id.cbAcabados);
plafones=findViewById(R.id.cbPlafones);
fachadas=findViewById(R.id.cbFachadas);
bardasPretilas=findViewById(R.id.cbBardasPretilas);
cubos=findViewById(R.id.cbCubos);
instalaciones=findViewById(R.id.cbInstalaciones);
if(grietas.isChecked()){gri="1";}else{gri="0";}
if(hundimiento.isChecked()){hun="1";}else{hun="0";}
if(inclinacionEdif.isChecked()){incEdif="1";}else{incEdif="0";}
if(losasColapso.isChecked()){losasCol="1";}else{losasCol="0";}
if(losasGrietasMax.isChecked()){losasGri="1";}else{losasGri="0";}
if(losasFlechaMax.isChecked()){losasFle="1";}else{losasFle="0";}
if(conexionFalla.isChecked()){conFalla="1";}else{conFalla="0";}
if(vidrios.isChecked()){vid="1";}else{vid="0";}
if(acabados.isChecked()){aca="1";}else{aca="0";}
if(plafones.isChecked()){pla="1";}else{pla="0";}
if(fachadas.isChecked()){facha="1";}else{facha="0";}
if(bardasPretilas.isChecked()){bardasPre="1";}else{bardasPre="0";}
if(cubos.isChecked()){cub="1";}else{cub="0";}
if(instalaciones.isChecked()){instala="1";}else{instala="0";}
}
public void onBackPressed() {
    Regresar();
}
private void Registrar(){
    final Response.Listener<String> responseListener= new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonResponse = new JSONObject(response);
                boolean success = jsonResponse.getBoolean("success");
                if(success){
                    System.out.println("Registro evaluacion exitoso");
                    AlertDialog.Builder builder = new AlertDialog.Builder(
EvaluacionActivity.this);
                    builder.setMessage("Reporte guardado con exito!").setNegativeButton(
"OK", null)
                        .create().show();
                    int p = android.os.Process.myPid();
                    android.os.Process.killProcess(p);
                }
            }
        }
    };
}

```

```

else{
    AlertDialog.Builder build;
    build=new AlertDialog.Builder(EvaluacionActivity.this);
    build.setMessage("No se pudo registrar en la BD!");
    build.setPositiveButton("OK", new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int which) {
        dialog.dismiss();
    }
});
    AlertDialog alert = build.create();
    alert.show();
}
} catch (JSONException e) {
    System.out.println("FAILED BD!!!!!!!!!!!!!!!!!!!!!!!!!!!!");
    e.printStackTrace();
    AlertDialog.Builder builder = new AlertDialog.Builder(EvaluacionActivity
.this);
    builder.setMessage("No hay conexi n con la BD").setNegativeButton("
Error", null)
        .create().show();
}
}
};
UpdateData();
EvaluacionRequest evaluacionRequest = new EvaluacionRequest(((Variables)
getApplicationContext().getID_INCIDENTE(), gri, hun, hundimientocm.getText().toString(),
incEdif, inclinacionEdifPorcent.getText().toString()
, losasCol, losasGri, losasFle, grietasMaxmm.getText().toString(), flechaMaxcm.
getText().toString(), conFalla, columnas.getSelectedItemAt().toString()
, colAnchoMaxGrieta.getText().toString(), colSeparacionEstribo.getText().
toString(), trabes.getSelectedItemAt().toString(), traAnchoMaxGrieta.getText().toString()
, traSeparacionEstribo.getText().toString(), concreto.getSelectedItemAt().
toString(), conAnchoMaxGrieta.getText().toString(), conSeparacionEstribo.getText().
toString()
, mampost.getSelectedItemAt().toString(), mamAnchoMaxGrieta.getText().toString()
, mamSeparacionEstribo.getText().toString(), numColumnasDanoSevero.getText().toString()
, totalColumnasEntrepiso.getText().toString(), danoEstructura.getSelectedItem
().toString(), vid, aca, pla, facha, bardasPre, cub, instala, responseListener);
RequestQueue queue = Volley.newRequestQueue(EvaluacionActivity.this);
queue.add(evaluacionRequest);
}
}
}

```

EvaluacionRequest.java

```
package com.example.luisd.ras;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;

import java.util.HashMap;
import java.util.Map;

public class EvaluacionRequest extends StringRequest {
    private static Variables var=new Variables();
    private static final String REGISTER_REQUEST_URL="http://"+var.getURL()+" /
Evaluaciondanos.php";
    private Map<String,String> params;

    public EvaluacionRequest(String id_incidente,String grietasTerreno,String hundimiento,
String hundimientocm,String inclinacionEdif,
String inclinacionEdifPorcent,String losasColapso,String
losasGrietasMax,String losasFlechaMax,
String grietasMaxmm,String flechaMaxcm,String conexionFalla,
String columnas,String colAnchoMaxGrieta,
String colSeparacionEstribo,String trabes,String
traAnchoMaxGrieta,String traSeparacionEstribo,
String concreto,String conAnchoMaxGrieta,String
conSeparacionEstribo,String mampost,
String mamAnchoMaxGrieta,String mamSeparacionEstribo,String
numColumnasDanoSevero,String totalColumnasEntrepiso,
String danoEstructura,String vidrios,String acabados,String
plafones,String fachadas,String bardasPretilas,
String cubos,String instalaciones,Response.Listener<String>
listener) {
    super(Request.Method.POST,REGISTER_REQUEST_URL, listener, null);
    params=new HashMap<>();

    params.put("id_incidente",id_incidente);
    params.put("grietasTerreno",grietasTerreno);
    params.put("hundimiento",hundimiento);
    params.put("hundimientocm",hundimientocm);
    params.put("inclinacionEdif",inclinacionEdif);
    params.put("inclinacionEdifPorcent",inclinacionEdifPorcent);
    params.put("losasColapso",losasColapso);
    params.put("losasGrietasMax",losasGrietasMax);
    params.put("losasFlechaMax",losasFlechaMax);
    params.put("grietasMaxmm",grietasMaxmm);
    params.put("flechaMaxcm",flechaMaxcm);
    params.put("conexionFalla",conexionFalla);
    params.put("columnas",columnas);
    params.put("colAnchoMaxGrieta",colAnchoMaxGrieta);
    params.put("colSeparacionEstribo",colSeparacionEstribo);
    params.put("trabes",trabes);
    params.put("traAnchoMaxGrieta",traAnchoMaxGrieta);
    params.put("traSeparacionEstribo",traSeparacionEstribo);
    params.put("concreto",concreto);
    params.put("conAnchoMaxGrieta",conAnchoMaxGrieta);
    params.put("conSeparacionEstribo",conSeparacionEstribo);
    params.put("mampost",mampost);
    params.put("mamAnchoMaxGrieta",mamAnchoMaxGrieta);
    params.put("mamSeparacionEstribo",mamSeparacionEstribo);
    params.put("numColumnasDanoSevero",numColumnasDanoSevero);
```

```
        params.put("totalColumnasEntrepiso", totalColumnasEntrepiso);
        params.put("danoEstructura", danoEstructura);
        params.put("vidrios", vidrios);
        params.put("acabados", acabados);
        params.put("plafones", plafones);
        params.put("fachadas", fachadas);
        params.put("baldasPretilas", bardasPretilas);
        params.put("cubos", cubos);
        params.put("instalaciones", instalaciones);
    }
    @Override
    public Map<String, String> getParams() { return params; }
}
```

CameraActivity.java

```
package com.example.luisd.ras;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.BitmapDrawable;
import android.provider.MediaStore;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Base64;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

public class CameraActivity extends AppCompatActivity {
    private static final int CAMERA_REQUEST_CODE=1;
    ImageView imagen;
    boolean take=false;
    Bitmap bitmap;
    EditText textDesc;
    private ProgressDialog progressDialog;
    int intentos=0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_camera);
        Button save;
        textDesc=findViewById(R.id.etDescripcion);
        TextView textView=(TextView)findViewById(R.id.tvIncidente);
        textView.setText("INC"+((Variables) getApplicationContext()).getID.INCIDENTE());
        imagen=(ImageView)findViewById(R.id.ivPhoto);
        save=(Button)findViewById(R.id.btnSavePhoto);
        save.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {
                    CargandoC();
                    validatePhoto();
                } catch (IOException e) {
                    e.printStackTrace();
                    progressDialog.dismiss();
                }
            }
        });
    }
}
```



```

        progressDialogC.dismiss();
    }
});
imagen.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        openCamera();
    }
});
}
private void openCamera() {
    Intent cameraIntent=new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if(cameraIntent.resolveActivity(getPackageManager())!=null){
        startActivityForResult(cameraIntent,CAMERA_REQUEST_CODE);
    }
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode==CAMERA_REQUEST_CODE&&resultCode==RESULT_OK){
        take=true;
        Bundle extras=data.getExtras();
        bitmap= (Bitmap) extras.get("data");
        imagen.setImageBitmap(bitmap);
    }
}
@Override
public void onBackPressed() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Regresar");
    builder.setMessage("Deseas Regresar?");
    builder.setPositiveButton("SI", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            finish();
            dialog.dismiss();
        }
    });
    builder.setNegativeButton("NO", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
        }
    });
    AlertDialog alert = builder.create();
    alert.show();
}
public void validatePhoto() throws IOException {
    AlertDialog.Builder builder;
    if(bitmap==null){
        builder=new AlertDialog.Builder(this);
        builder.setMessage("No hay imagen que capturar!");

        builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {

            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
            }
        });
        AlertDialog alert = builder.create();
        alert.show();
        return;
    }
}

```

```

else{
    if(textDesc.getText().toString().equals(""))
    {
        builder=new AlertDialog.Builder(this);
        builder.setMessage("Falta agregar una descripción!");
        builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
            }
        });
        AlertDialog alert = builder.create();
        alert.show();
        return;
    }
    CargandoC();
    savePhoto();
}
return;
}
public void savePhoto() throws IOException {
    //Convertir imagen a String
    ByteArrayOutputStream array=new ByteArrayOutputStream();
    Bitmap bitmap2=Bitmap.createScaledBitmap(bitmap,bitmap.getWidth()*4,bitmap.getHeight()*3,true);
    bitmap2.compress(Bitmap.CompressFormat.PNG,100,array);
    byte[] imagenByte=array.toByteArray();
    String imageString= Base64.encodeToString(imagenByte,Base64.DEFAULT);
    String id_Inc=((Variables) getApplicationContext()).getID_INCIDENTE();
    String desc=textDesc.getText().toString();

    final Response.Listener<String> responseListener=new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try
            {
                JSONObject jsonResponse = new JSONObject(response);
                boolean success = jsonResponse.getBoolean("success");
                if(success){
                    progressDialogC.dismiss();
                    Toast.makeText(CameraActivity.this, "La imagen ha sido guardada con éxito!", Toast.LENGTHLONG).show();
                    imagen.setImageResource(R.drawable.take);
                    Intent intent = new Intent(CameraActivity.this, CameraActivity.class);

                );

                    CameraActivity.this.startActivity(intent);
                    finish();
                }
                else{
                    AlertDialog.Builder build;
                    build=new AlertDialog.Builder(CameraActivity.this);
                    build.setMessage("No se pudo guardar la imagen!");
                    build.setPositiveButton("OK", new DialogInterface.OnClickListener()

                    {

                        public void onClick(DialogInterface dialog, int which) {
                            dialog.dismiss();
                        }

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

```

```

        } catch (JSONException e) {
            System.out.println("FAILED BD!!!!!!!!!!!!!!!!!!!!!!!!!!!!");
            e.printStackTrace();
            AlertDialog.Builder builder = new AlertDialog.Builder(CameraActivity.
this);
            builder.setMessage("No hay conexión con la BD").setNegativeButton("
Error", null)
                .create().show();
        }
    };
    CameraRequest cameraRequest = new CameraRequest(id_Inc, desc, imageString,
responseListener);
    RequestQueue queue = Volley.newRequestQueue(CameraActivity.this);
    queue.add(cameraRequest);
}
private void CargandoC() {
    progressDialog=new ProgressDialog(this);
    progressDialog.setMessage("Cargando");
    progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
    progressDialog.setIndeterminate(true);
    progressDialog.setProgress(0);
    progressDialog.show();
}
}
}

```

CameraRequest.java

```
package com.example.luisd.ras;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;

import java.util.HashMap;
import java.util.Map;

public class CameraRequest extends StringRequest {
    private static final Variables var=new Variables();
    private static final String REGISTER_REQUEST_URL="http://"+var.getURL()+"/SavePhoto.php"
    ;
    private Map<String,String> params;

    public CameraRequest(String id_incidente,String descripcion,String image, Response.
    Listener<String> listener) {
        super(Request.Method.POST,REGISTER_REQUEST_URL, listener , null);
        params=new HashMap<>();

        params.put("idIncidente",id_incidente);
        params.put("descripcion",descripcion);
        params.put("imagen",image);

    }
    @Override
    public Map<String,String> getParams(){return params;}
}
```