



TECNOLOGICO NACIONAL DE MEXICO

# **Instituto Tecnológico de la Laguna**

Ingeniería en Sistemas Computacionales

## **PROGRAMACION DE DISPOSITIVOS MOVILES CON ANDROID**

SEMESTRE: Ago - Dic / 2018

GRUPO: "A" 10 – 11 Hrs

PRACTICA No. U5P1

### **Proyecto Final: App Perdidos y Encontrados**

ALUMNOS:

15130685 Fernando Alfonso Caldera Olivas  
15131438 Jesús Iván Sifuentes Valle

PROFESOR:

Ing. Luis Fernando Gil Vázquez

**Torreón, Coah. a Miércoles 12 de Diciembre de 2018**

## Situación Didáctica

El Tecnológico de La Laguna es una institución de educación superior que cuenta con una población de mas de 5,000 estudiantes y cerca de 500 trabajadores. Resulta común que entre una comunidad de personas de éste tamaño frecuentemente se extravíen objetos tales como memorias USB, celulares, sueteres, libretas, libros, audifonos y un sinnúmero de cosas y accesorios. Las personas que extravían alguna pertenencia tienen que recurrir a pegar alguna hoja en las paredes de los edificios con un aviso, sin embargo no todas las personas ponen atención a los avisos por lo que resulta inefectivo la mayoría de las ocasiones.

Estudiantes de la materia de Programación de Dispositivos Mviles en Android han visto esta área de oportunidad para crear una aplicación Android que funcione como un sistema de “Perdidos y Encontrados” en la que un usuario pueda registrar los detalles de algun objeto perdido, los demás usuarios de la aplicación podrán ver los objetos reportados como perdidos y en caso de que lo encuentren puedan registrar un aviso de que fue encontrado. La aplicación debería dar opciones de generar una llamada telefonica, enviar un SMS o bien un correo electrónico a quien extravió dependiendo si éste alimento estos datos para contactarlo. De igual manera es deseable que en la app pueda consultarse el estado de cada extravío, si esta perdido, si fue encontrado o si fue recuperado y pudiera ofrecer formas de consultar los perdidos y encontrados por diferentes criterios.

La aplicación debe contar con una pantalla de splash vistosa donde se muestre el nombre de la aplicación. La aplicación debe contar con una función Acerca de con los datos completos de identificación de la institución, carrera, materia, autores, nombre de la app, periodo semestral, foto de los autores y logos del TecNM y el ITL.

Si la aplicación usa una base de datos deberá tener una opción que permita crear las tablas de la base de datos por primera vez.

## Analisis

La aplicación “Perdidos y Encontrados” es una aplicación pensada para ayudar a la gente que encontró un objeto sin dueño y que pretende devolverlo a su legítimo dueño, esto, a través de la publicación del mismo objeto. Por otra parte, también está pensada para ayudar a la gente que perdió un objeto, a encontrarlo. Esto a través del sistema de búsqueda que tiene incluida la aplicación, que te permite hacer una búsqueda por descripción, lugar donde perdiste tu objeto y rango de fechas.

El flujo de la aplicación es el siguiente:

1. Una persona A encuentra un objeto O sin dueño.
2. La persona A, con toda la mejor intención del mundo, quiere regresar dicho objeto O a su legítimo dueño.
3. La persona A usa la aplicación “Perdidos y Encontrados” y se crea una cuenta, si aún no tiene una.
  - a. Si crea una cuenta, debe especificar solamente un nombre de usuario **único**, una contraseña y un teléfono de contacto.
4. La persona A inicia sesión en su cuenta.
5. Inmediatamente después, se desplegará el menú principal, donde se podrán ver las publicaciones existentes ordenadas de manera descendente por la fecha en que se encontraron los objetos publicados, una sección de búsqueda y una barra con botones para ver las publicaciones de la persona A, agregar una publicación o salir de la sesión.
6. La persona A crea una nueva publicación, agregando información que describa al objeto O, el lugar donde lo encontró y en qué fecha lo encontró.
  - a. También puede tomarle fotos al objeto O con una aplicación externa y luego adjuntar las imágenes de dicho objeto.
7. La persona A publica el objeto O como “objeto sin dueño”.
8. Ahora entra en escena la persona B, quien es el dueño legítimo del objeto O.
9. La persona B quiere encontrar su objeto O.
10. Utilizando la aplicación “Perdidos y Encontrados”, va a encontrar su objeto O.
11. La persona B se crea una cuenta, si es que no tiene una.
12. La persona B inicia sesión con su cuenta.
13. Inmediatamente al iniciar sesión, se desplegará el menú principal, desde donde el cual podrá ver todas las publicaciones que hay hasta el momento.
14. Si la persona B no ve una descripción convincente en las primeras publicaciones, puede usar la sección de búsqueda para buscar por descripción, lugar donde perdió el objeto O y rango de fechas en que lo perdió.
15. Una vez localizada una publicación que le parezca a la persona B, es su objeto O, puede presionar el botón “inspeccionar publicación”, el cual se muestra con el icono de una lupa.
16. Al inspeccionar la publicación, se abre una nueva actividad donde se muestran los datos que ya veía el usuario B, además del teléfono de contacto del usuario hizo la publicación, un botón para marcar su número y una galería con las imágenes que adjuntó dicho usuario.
17. En este caso, la persona y usuario A fue quien hizo la publicación, por lo que se muestra su número de contacto.
18. La persona y usuario B puede llamar al usuario A y preguntar por su objeto.

Estos son los alcances de la aplicación. A partir del punto 18, depende totalmente de los usuarios A y B el hecho de que se llegue a un acuerdo o no.

La aplicación se conectará a una base de datos en MS SQL Server utilizando el conector JDBC 4 para el mismo servidor de base de datos, en este caso, utilizaremos el conector JTDS.

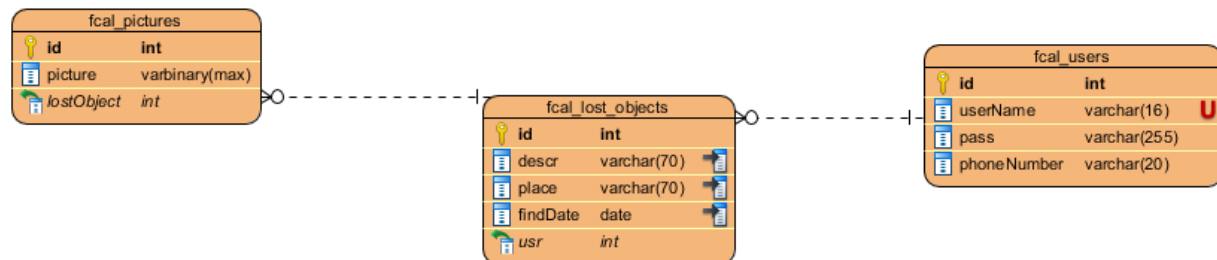
Datos de la Conexión:

- Host: 187.189.201.198
- Puerto: 1433
- Base de datos: pmadb
- Usuario: pma
- Contraseña: android

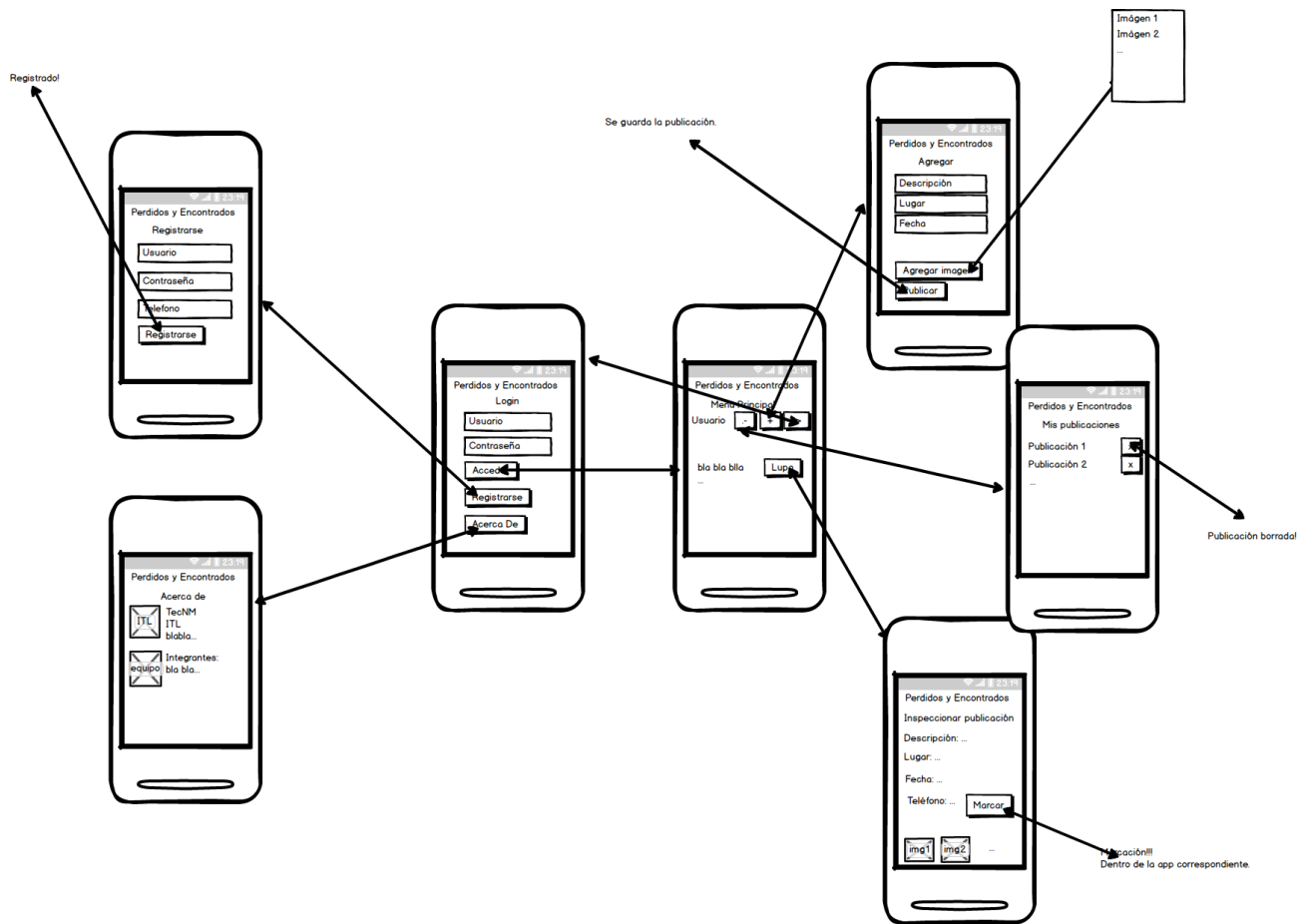
Cadena de conexión: `"jdbc:jtds:sqlserver://" + Host + ":" + Puerto + ";"`  
`+ "databaseName=" + Base de datos + ";user=" + Usuario`  
`+ ";password=" + Contraseña + ";"`

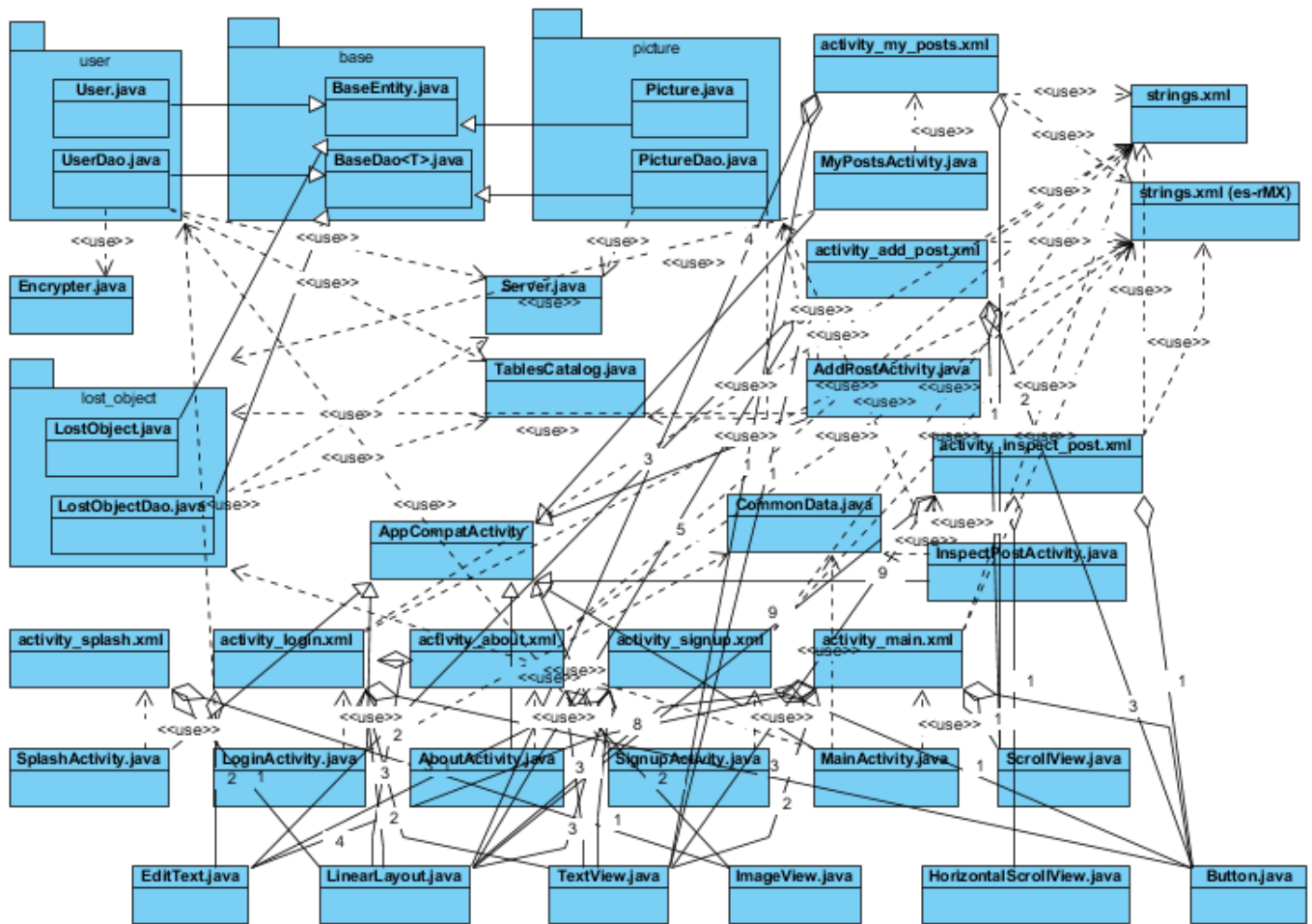
•

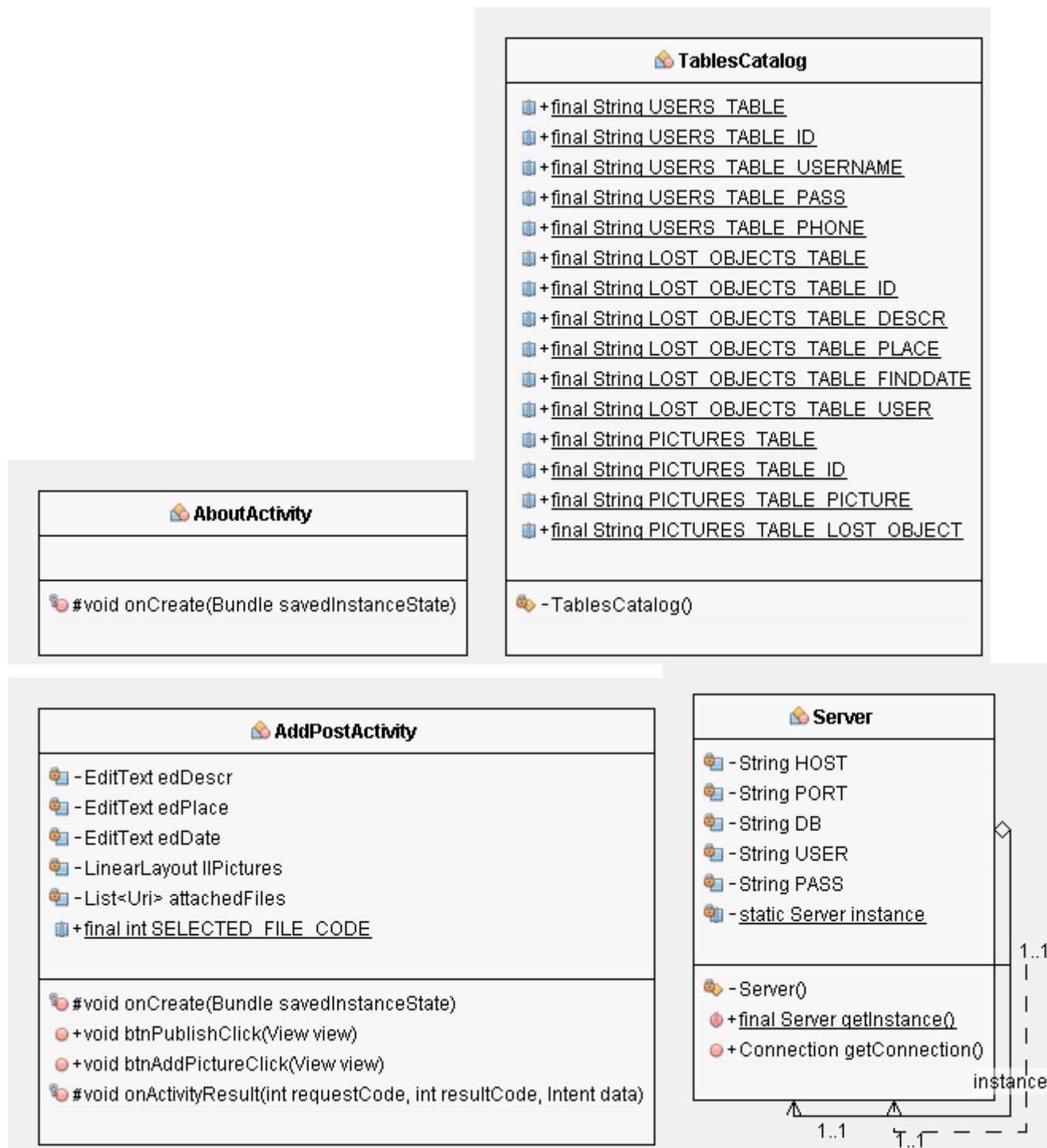
Las tablas a utilizar son las siguientes:

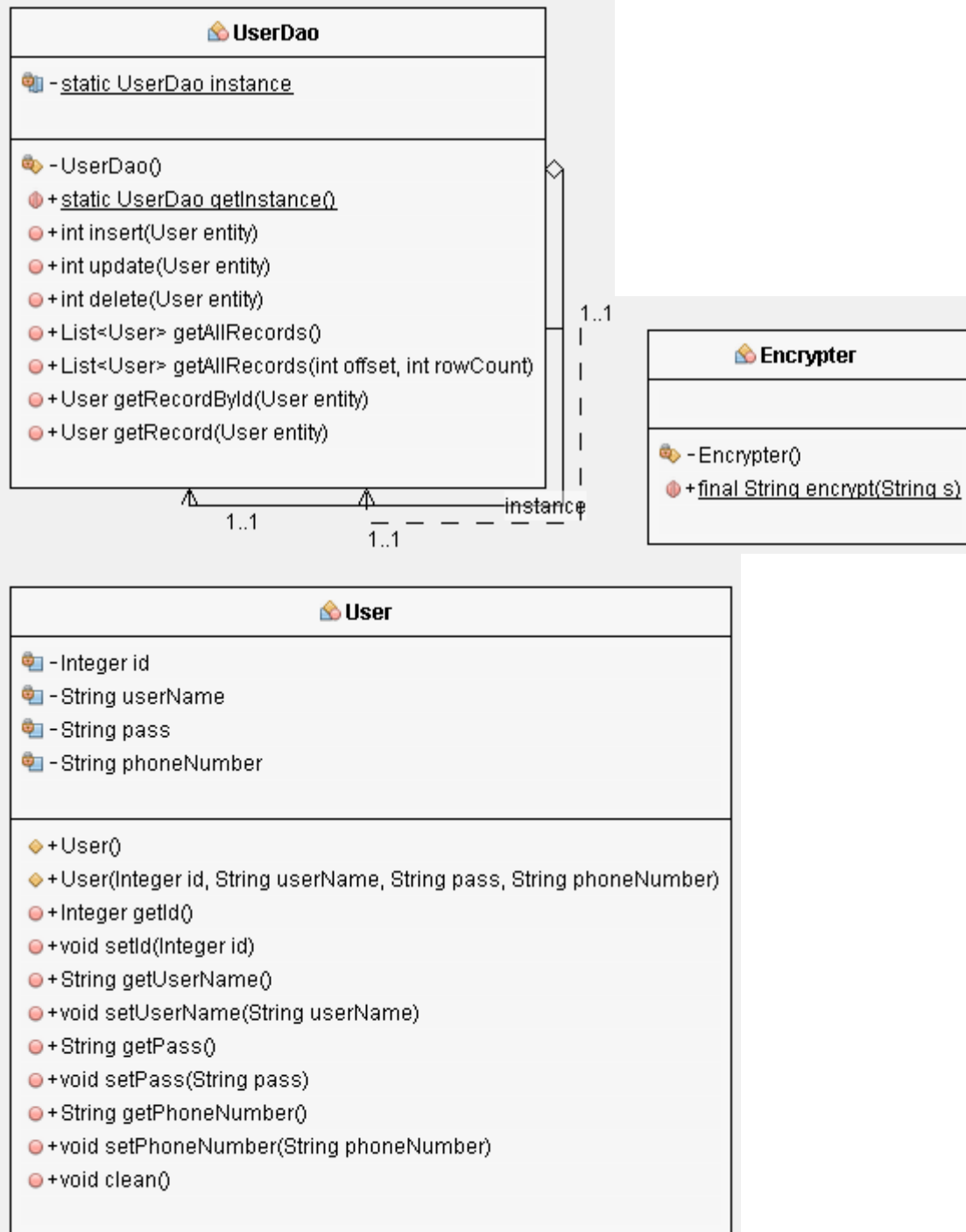


## Diseño

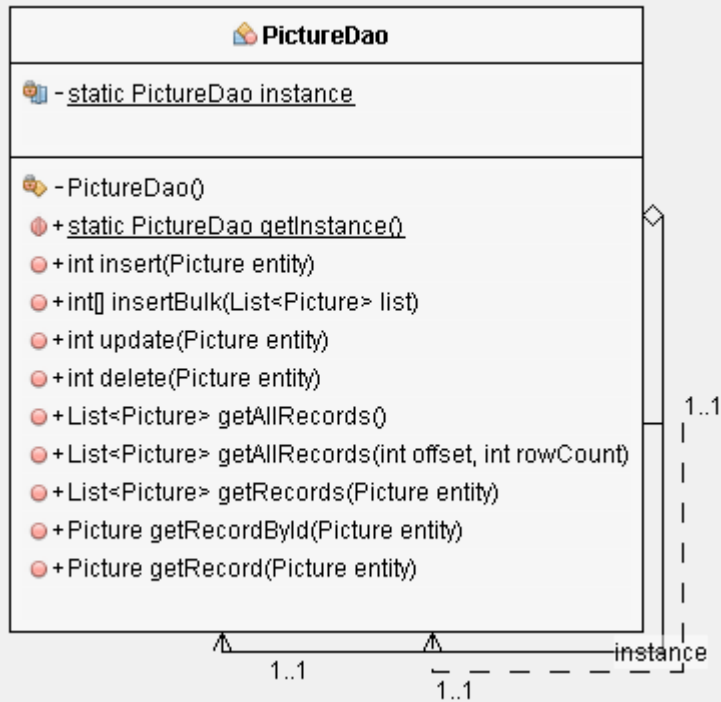




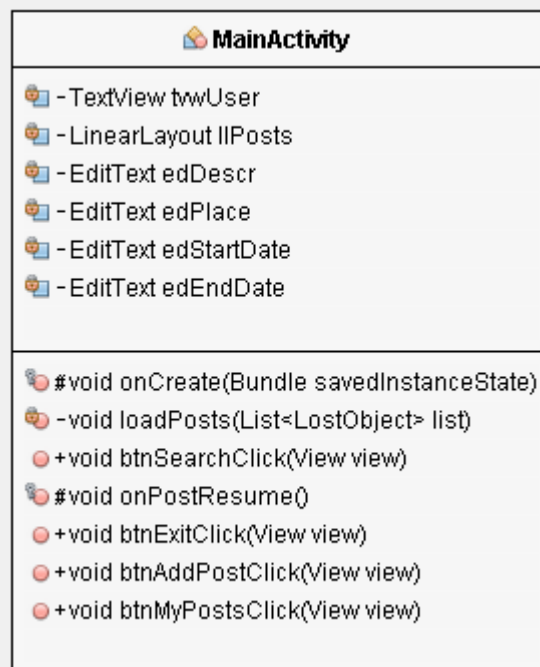
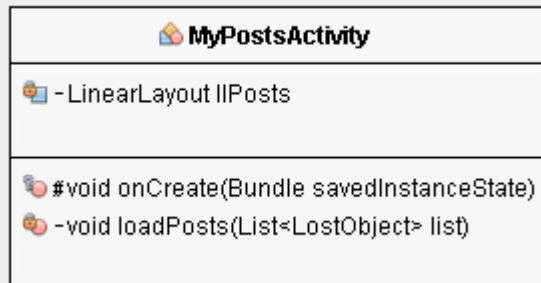
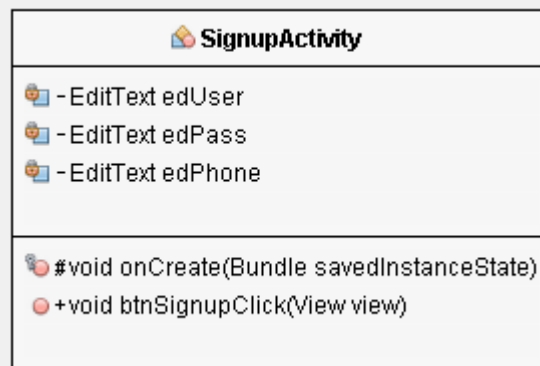
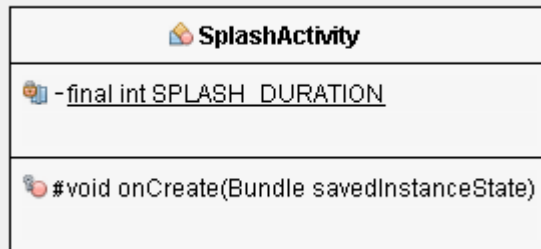
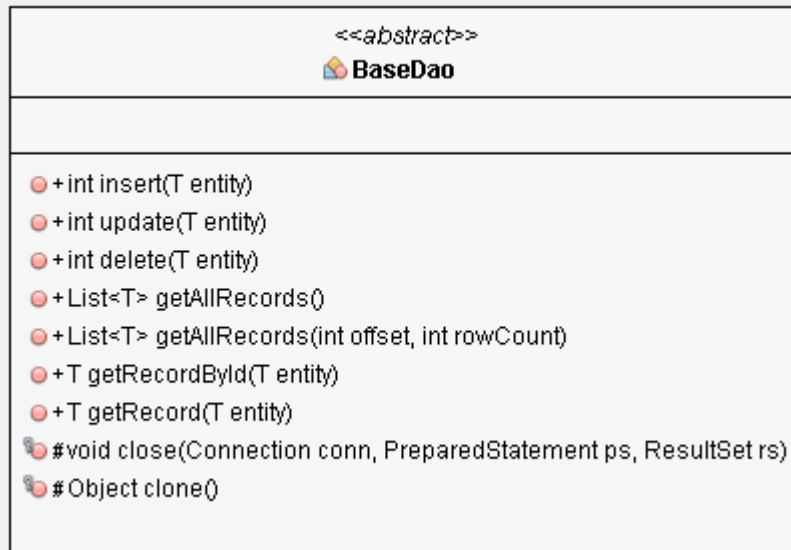


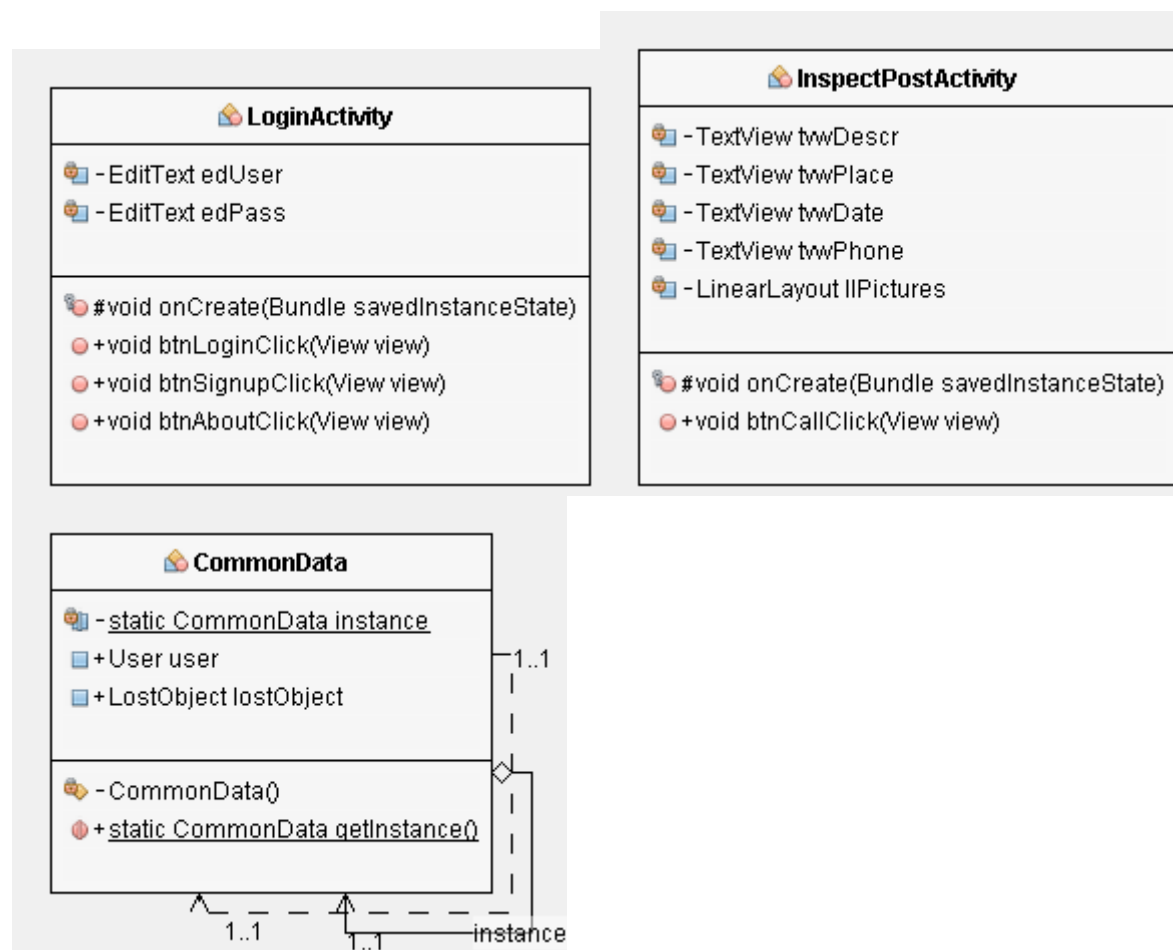












## Código

### BaseEntity.java

```

/*-----
: *
: *          INSTITUTO TECNOLOGICO DE LA LAGUNA
: *          INGENIERIA EN SISTEMAS COMPUTACIONALES
: *          PROGRAMACION MOVILES EN ANDROID "A"
: *
: *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
: *
: *          Entidad básica para tipado.
: *
: * Archivo      : BaseEntity.java
: * Autor        : Fernando Alfonso Caldera Olivas      15130685
: *              : Jesús Iván Sifuentes Valle           15131438
: * Fecha        : 13/Dic/2018
: * Compilador   : Android Studio 3.1.3
: * Descripción  : Esta clase sirve como padre de todas las clases que sean modelos (MVC).
: *              : Es necesario heredar de esta clase para poder implementar el DAO,
: *              : ya que se requiere que el tipado del DAO herede de esta clase.
: * Ultima modif:
: * Fecha        Modificó          Motivo
: *=====
: *
: *-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.base;

public abstract class BaseEntity{

    //-----
    //Método para limpiar el modelo (sus datos).
    public abstract void clean();
    //-----
}

```

**BaseDao.java**

```

/*-----
*:
*:          INSTITUTO TECNOLOGICO DE LA LAGUNA
*:          INGENIERIA EN SISTEMAS COMPUTACIONALES
*:          PROGRAMACION MOVILES EN ANDROID "A"
*:
*:          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
*:
*:          DAO básico
*:
*: Archivo      : BaseDao.java
*: Autor        : Fernando Alfonso Caldera Olivas      15130685
*:              Jesús Iván Sifuentes Valle            15131438
*: Fecha        : 13/Dic/2018
*: Compilador   : Android Studio 3.1.3
*: Descripción  : Esta clase sirve como base para los DAOs, como extensión del modelo (MVC),
*:              que sirve para acceder a los datos.
*:              Requiere un tipado que extienda de BaseEntity para funcionar y tiene varias
*:              firmas de métodos de acceso a los datos que son comunes, como insertar,
actualizar,
*:              etc.
*:              También contiene un método para cerrar la conexión, sentencias preparadas y
conjuntos
*:              resultado.
*:              Las clases que hereden de esta, deben implementar el patrón Singleton, por lo
que queda
*:              deshabilitada la acción de clonar.
*:              Tod0 lo anterior, para lograr una Arquitectura Limpia (Independiente del
Framework,
*:              Manejador de Base de Datos, etc).
*: Ultima modif:
*: Fecha        Modificó          Motivo
*:=====
*:
*:-----*/
package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.base;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;

public abstract class BaseDao<T extends BaseEntity> {

    //-----
    //Guarda una entidad.
    public abstract int insert(T entity) throws Exception;
    //-----

    //-----
    //Actualiza una entidad.
    public abstract int update(T entity) throws Exception;
    //-----

    //-----
    //Borra una entidad
    public abstract int delete(T entity) throws Exception;
    //-----

```

```
//-----  
//Obtiene todas las entidades  
public abstract List<T> getAllRecords() throws Exception;  
//-----  
  
//-----  
//Obtiene todas las entidades con paginación  
public abstract List<T> getAllRecords(int offset, int rowCount) throws Exception;  
//-----  
  
//-----  
//Obtiene entidad por id  
public abstract T getRecordById(T entity) throws Exception;  
//-----  
  
//-----  
//Obtiene entidad por parámetros de búsqueda simples  
public abstract T getRecord(T entity) throws Exception;  
//-----  
  
//-----  
//Método para cerrar conexión, Sentencias preparadas y Conjuntos resultado.  
protected final void close(Connection conn, PreparedStatement ps, ResultSet rs) {  
    if(rs != null) {  
        try {  
            rs.close();  
        } catch (SQLException e) {  
            }  
    }  
    if(ps != null) {  
        try {  
            ps.close();  
        } catch (SQLException e) {  
            }  
    }  
    if(conn != null) {  
        try {  
            conn.close();  
        } catch (SQLException e) {  
            }  
    }  
}  
//-----  
  
//-----  
//Clonación deshabilitada.  
@Override  
protected final Object clone() throws CloneNotSupportedException {  
    throw new CloneNotSupportedException();  
}  
//-----  
}
```





**LostObject.java**

```

/*-----
 *
 *          INSTITUTO TECNOLOGICO DE LA LAGUNA
 *          INGENIERIA EN SISTEMAS COMPUTACIONALES
 *          PROGRAMACION MOVILES EN ANDROID "A"
 *
 *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
 *
 *          Modelo para publicaciones de objetos perdidos
 *
 * Archivo      : LostObject.java
 * Autor        : Fernando Alfonso Caldera Olivas      15130685
 *              : Jesús Iván Sifuentes Valle          15131438
 * Fecha        : 13/Dic/2018
 * Compilador   : Android Studio 3.1.3
 * Descripción  : Implementa el modelo para mapear la tabla fcal_lost_objects.
 * Ultima modif:
 * Fecha        Modificó          Motivo
 *-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.lost_object;

import java.sql.Date;
import java.sql.Time;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.user.User;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.base.BaseEntity;

public class LostObject extends BaseEntity {
    //Atributos para mapear las columnas de la tabla
    private Integer id;
    private String descr;
    private String place;
    private Date findDate;
    private User usr;

    //Constructores
    //-----

    public LostObject() {
    }
    //-----

    //-----

    public LostObject(Integer id, String descr, String place, Date findDate, User usr) {
        this.id = id;
        this.descr = descr;
        this.place = place;
        this.findDate = findDate;
        this.usr = usr;
    }
    //-----

    //Getters y Setters
    //-----

    public Integer getId() {
        return id;
    }
    //-----
}

```

```
//-----  
-----  
public void setId(Integer id) {  
    this.id = id;  
}  
//-----  
-----  
  
//-----  
-----  
public String getDescr() {  
    return descr;  
}  
//-----  
-----  
  
//-----  
-----  
public void setDescr(String descr) {  
    this.descr = descr;  
}  
//-----  
-----  
  
//-----  
-----  
public String getPlace() {  
    return place;  
}  
//-----  
-----  
  
//-----  
-----  
public void setPlace(String place) {  
    this.place = place;  
}  
//-----  
-----  
  
//-----  
-----  
public Date getFindDate() {  
    return findDate;  
}  
//-----  
-----  
  
//-----  
-----  
public void setFindDate(Date findDate) {  
    this.findDate = findDate;  
}  
//-----  
-----  
  
//-----  
-----  
public User getUsr() {  
    if(usr == null){  
        usr = new User();  
    }  
    return usr;  
}  
//-----  
-----
```

```
//-----  
-----  
public void setUsr(User usr) {  
    this.usr = usr;  
}  
//-----  
-----  
  
//Implementación para limpiar modelo  
@Override  
public void clean() {  
    this.id = null;  
    this.descr = null;  
    this.place = null;  
    this.findDate = null;  
    this.usr = null;  
}  
}
```

**LostObjectDao.java**

```

/*-----
 *
 *          INSTITUTO TECNOLÓGICO DE LA LAGUNA
 *          INGENIERIA EN SISTEMAS COMPUTACIONALES
 *          PROGRAMACION MOVILES EN ANDROID "A"
 *
 *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
 *
 *          DAO para publicaciones de objetos perdidos
 *
 * Archivo      : LostObjectDao.java
 * Autor        : Fernando Alfonso Caldera Olivas      15130685
 *              : Jesús Iván Sifuentes Valle          15131438
 * Fecha        : 13/Dic/2018
 * Compilador   : Android Studio 3.1.3
 * Descripción  : Implementa el DAO para las publicaciones de objetos perdidos, heredando
 *              : e implementando los métodos de BaseDao.
 * Última modif:
 * Fecha        Modificó          Motivo
 *=====
 *
 *-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.lost_object;

import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Time;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.List;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.base.BaseDao;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.user.User;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.utils.Server;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.utils.TablesCatalog;

public class LostObjectDao extends BaseDao<LostObject> {

    //-----
    //Implementación del Singleton
    private static LostObjectDao instance;
    //-----

    //-----
    private LostObjectDao() {}
    //-----

    //-----
    public static LostObjectDao getInstance() {
        if(instance == null) {
            instance = new LostObjectDao();
        }
        return instance;
    }
    //-----

    //-----

```

```

-----
@Override
public int insert(LostObject entity) throws Exception {
    int c = 0;
    Connection conn = null;
    PreparedStatement ps = null;
    try{
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " insert into " + TablesCatalog.LOST_OBJECTS_TABLE
            + " ( "
                + TablesCatalog.LOST_OBJECTS_TABLE_DESCR + " , "
                + TablesCatalog.LOST_OBJECTS_TABLE_PLACE + " , "
                + TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE + " , "
                + TablesCatalog.LOST_OBJECTS_TABLE_USER
            + " ) "
            + " values(?, ?, ?, ?) "
        );
        int p = 1;
        ps.setString(p++, entity.getDescr());
        ps.setString(p++, entity.getPlace());
        ps.setDate(p++, entity.getFindDate());
        ps.setInt(p++, entity.getUsr().getId());
        c = ps.executeUpdate();
        LostObject lostObjectAux = getRecord(entity);
        entity.setId(lostObjectAux.getId());
    }catch(Exception e){
        throw e;
    }finally{
        close(conn, ps, null);
    }
    return c;
}
//-----

```

```

-----
@Override
public int update(LostObject entity) throws Exception {
    int c = 0;
    Connection conn = null;
    PreparedStatement ps = null;
    try{
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " update " + TablesCatalog.LOST_OBJECTS_TABLE
            + " set "
                + TablesCatalog.LOST_OBJECTS_TABLE_DESCR + " = ? , "
                + TablesCatalog.LOST_OBJECTS_TABLE_PLACE + " = ? , "
                + TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE + " = ? , "
                + TablesCatalog.LOST_OBJECTS_TABLE_USER + " = ? "
            + " where " + TablesCatalog.LOST_OBJECTS_TABLE_ID + " = ?"
        );
        int p = 1;
        ps.setString(p++, entity.getDescr());
        ps.setString(p++, entity.getPlace());
        ps.setDate(p++, entity.getFindDate());
        ps.setInt(p++, entity.getUsr().getId());
        ps.setInt(p++, entity.getId());

        c = ps.executeUpdate();
    }catch(Exception e){
        throw e;
    }finally{
        close(conn, ps, null);
    }
}

```

```

    }
    return c;
}
//-----

//-----

@Override
public int delete(LostObject entity) throws Exception {
    int c = 0;
    Connection conn = null;
    PreparedStatement ps = null;
    try{
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " delete from " + TablesCatalog.LOST_OBJECTS_TABLE
            + " where " + TablesCatalog.LOST_OBJECTS_TABLE_ID + " = ?"
        );
        int p = 1;
        ps.setInt(p++, entity.getId());

        c = ps.executeUpdate();
    }catch(Exception e){
        throw e;
    }finally{
        close(conn, ps, null);
    }
    return c;
}
//-----

//-----

@Override
public List<LostObject> getAllRecords() throws Exception {
    List<LostObject> list = new ArrayList<>();
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    try{
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " select " + TablesCatalog.LOST_OBJECTS_TABLE + ".*, "
            + TablesCatalog.USERS_TABLE + ".*" + " from " +
TablesCatalog.LOST_OBJECTS_TABLE
            + " join " + TablesCatalog.USERS_TABLE + " on " +
TablesCatalog.LOST_OBJECTS_TABLE + "." + TablesCatalog.LOST_OBJECTS_TABLE_USER
            + " = " + TablesCatalog.USERS_TABLE + "." +
TablesCatalog.USERS_TABLE_ID
            + " order by " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE + " desc "
        );

        rs = ps.executeQuery();

        while(rs.next()){
            LostObject lo = new LostObject();
            lo.setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_ID));
            lo.setDescr(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_DESCR));
            lo.setPlace(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_PLACE));
            lo.setFindDate(rs.getDate(TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE));
            lo.getUsr().setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_USER));
            lo.getUsr().setUserName(rs.getString(TablesCatalog.USERS_TABLE_USERNAME));
            lo.getUsr().setPhoneNumber(rs.getString(TablesCatalog.USERS_TABLE_PHONE));

```

```

        list.add(lo);
    }

    }catch(Exception e){
        throw e;
    }finally{
        close(conn, ps, rs);
    }
    return list;
}

//-----

//-----

@Override
public List<LostObject> getAllRecords(int offset, int rowCount) throws Exception {
    List<LostObject> list = new ArrayList<>();
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    try{
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " select " + TablesCatalog.LOST_OBJECTS_TABLE + ".*, "
            + TablesCatalog.USERS_TABLE + ".*" + " from " +
TablesCatalog.LOST_OBJECTS_TABLE
            + " join " + TablesCatalog.USERS_TABLE + " on " +
TablesCatalog.LOST_OBJECTS_TABLE + "." + TablesCatalog.LOST_OBJECTS_TABLE_USER
            + " = " + TablesCatalog.USERS_TABLE + "." +
TablesCatalog.USERS_TABLE_ID
            + " order by " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE + " desc "
            + " offset " + offset + " rows fetch next " + rowCount + " rows"
        );

        rs = ps.executeQuery();

        while(rs.next()){
            LostObject lo = new LostObject();
            lo.setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_ID));
            lo.setDescr(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_DESCR));
            lo.setPlace(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_PLACE));
            lo.setFindDate(rs.getDate(TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE));
            lo.getUsr().setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_USER));
            lo.getUsr().setUserName(rs.getString(TablesCatalog.USERS_TABLE_USERNAME));
            lo.getUsr().setPhoneNumber(rs.getString(TablesCatalog.USERS_TABLE_PHONE));
            list.add(lo);
        }

    }catch(Exception e){
        throw e;
    }finally{
        close(conn, ps, rs);
    }
    return list;
}

//-----

//-----

//Obtiene las publicaciones en un rango de fechas, que cumplan con un patrón de descripción
y un patrón
//de lugar.
public List<LostObject> getRecordsInRange(String startDate, String endDate, String descr,
String place) throws Exception{

```

```

List<LostObject> list = new ArrayList<>();
Connection conn = null;
PreparedStatement ps = null;
ResultSet rs = null;
try{
    conn = Server.getInstance().getConnection();
    String s = " select " + TablesCatalog.LOST_OBJECTS_TABLE + ".*, "
        + TablesCatalog.USERS_TABLE + ".*" + " from " +
TablesCatalog.LOST_OBJECTS_TABLE
        + " join " + TablesCatalog.USERS_TABLE + " on " +
TablesCatalog.LOST_OBJECTS_TABLE + "." + TablesCatalog.LOST_OBJECTS_TABLE_USER
        + " = " + TablesCatalog.USERS_TABLE + "." + TablesCatalog.USERS_TABLE_ID
        + " where 1 = 1 ";

    if(!startDate.trim().equals("") && !endDate.trim().equals("")){
        s += " and " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE + " between ? and ?";
    }
    if(!place.trim().equals("")){
        s += " and " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_PLACE + " like ? ";
    }
    if(!descr.trim().equals("")){
        s += " and " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_DESCR + " like ? ";
    }

    s += " order by " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE + " desc ";

    ps = conn.prepareStatement(s);

    int p = 1;
    SimpleDateFormat date = new SimpleDateFormat("dd/MM/yyyy");

    long ms = 0;
    if(!startDate.trim().equals("")) {
        ms = date.parse(startDate).getTime();
        ps.setDate(p++, new Date(ms));
    }
    if(!endDate.trim().equals("")) {
        ms = date.parse(endDate).getTime();
        ps.setDate(p++, new Date(ms));
    }
    if(!place.trim().equals("")) {
        ps.setString(p++, "%" + place + "%");
    }
    if(!descr.trim().equals("")) {
        ps.setString(p++, "%" + descr + "%");
    }

    rs = ps.executeQuery();

    while(rs.next()){
        LostObject lo = new LostObject();
        lo.setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_ID));
        lo.setDescr(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_DESCR));
        lo.setPlace(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_PLACE));
        lo.setFindDate(rs.getDate(TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE));
        lo.getUsr().setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_USER));
        lo.getUsr().setUserName(rs.getString(TablesCatalog.USERS_TABLE_USERNAME));
        lo.getUsr().setPhoneNumber(rs.getString(TablesCatalog.USERS_TABLE_PHONE));
        list.add(lo);
    }

} catch (Exception e) {
    throw e;
}

```



```

    }finally{
        close(conn, ps, rs);
    }
    return list;
}
//-----
//-----

//Obtiene las publicaciones en un rango de fechas, que cumplan con un patrón de descripción
y un patrón
//de lugar, usando paginación.
public List<LostObject> getRecordsInRange(String startDate, String endDate, String descr,
String place, int offset, int rowCount) throws Exception{
    List<LostObject> list = new ArrayList<>();
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    try{
        conn = Server.getInstance().getConnection();
        String s = " select " + TablesCatalog.LOST_OBJECTS_TABLE + ".*, "
            + TablesCatalog.USERS_TABLE + ".*" + " from " +
TablesCatalog.LOST_OBJECTS_TABLE
            + " join " + TablesCatalog.USERS_TABLE + " on " +
TablesCatalog.LOST_OBJECTS_TABLE + "." + TablesCatalog.LOST_OBJECTS_TABLE_USER
            + " = " + TablesCatalog.USERS_TABLE + "." + TablesCatalog.USERS_TABLE_ID
            + " where 1 = 1 ";

        if(!startDate.trim().equals("") && !endDate.trim().equals("")){
            s += " and " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE + " between ? and ?";
        }
        if(!place.trim().equals("")){
            s += " and " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_PLACE + " like ? ";
        }
        if(!descr.trim().equals("")){
            s += " and " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_DESCR + " like ? ";
        }

        s += " order by " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE + " desc ";
        s += " offset " + offset + " rows fetch next " + rowCount + " rows";

        ps = conn.prepareStatement(s);

        int p = 1;
        SimpleDateFormat date = new SimpleDateFormat("dd/MM/yyyy");

        long ms = 0;
        if(!startDate.trim().equals("")) {
            ms = date.parse(startDate).getTime();
            ps.setDate(p++, new Date(ms));
        }
        if(!endDate.trim().equals("")) {
            ms = date.parse(endDate).getTime();
            ps.setDate(p++, new Date(ms));
        }
        if(!place.trim().equals("")) {
            ps.setString(p++, "%" + place + "%");
        }
        if(!descr.trim().equals("")) {
            ps.setString(p++, "%" + descr + "%");
        }
    }
}

```

```

rs = ps.executeQuery();

while(rs.next()){
    LostObject lo = new LostObject();
    lo.setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_ID));
    lo.setDescr(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_DESCR));
    lo.setPlace(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_PLACE));
    lo.setFindDate(rs.getDate(TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE));
    lo.getUsr().setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_USER));
    lo.getUsr().setUserName(rs.getString(TablesCatalog.USERS_TABLE_USERNAME));
    lo.getUsr().setPhoneNumber(rs.getString(TablesCatalog.USERS_TABLE_PHONE));
    list.add(lo);
}

} catch (Exception e) {
    throw e;
} finally {
    close(conn, ps, rs);
}
return list;
}
//-----

//-----

//Obtiene todas las publicaciones de un usuario especificado, usando paginación.
public List<LostObject> getRecordsByUser(User u, int offset, int rowCount) throws Exception
{
    List<LostObject> list = new ArrayList<>();
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    try{
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " select " + TablesCatalog.LOST_OBJECTS_TABLE + ".*, "
            + TablesCatalog.USERS_TABLE + ".*" + " from " +
TablesCatalog.LOST_OBJECTS_TABLE
            + " join " + TablesCatalog.USERS_TABLE + " on " +
TablesCatalog.LOST_OBJECTS_TABLE_USER
            + " = " + TablesCatalog.USERS_TABLE_ID
            + " where " + TablesCatalog.LOST_OBJECTS_TABLE + "."
            + TablesCatalog.LOST_OBJECTS_TABLE_USER + " = ? "
            + " order by " + TablesCatalog.LOST_OBJECTS_TABLE + "."
            + TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE + " desc "
            + " offset " + offset + " rows fetch next " + rowCount + " rows"
        );

        int p = 1;
        ps.setInt(p++, u.getId());

        rs = ps.executeQuery();

        while(rs.next()){
            LostObject lo = new LostObject();
            lo.setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_ID));
            lo.setDescr(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_DESCR));
            lo.setPlace(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_PLACE));
            lo.setFindDate(rs.getDate(TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE));
            lo.getUsr().setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_USER));
            lo.getUsr().setUserName(rs.getString(TablesCatalog.USERS_TABLE_USERNAME));
            lo.getUsr().setPhoneNumber(rs.getString(TablesCatalog.USERS_TABLE_PHONE));
            list.add(lo);
        }
    }
}

```

```

    }catch(Exception e){
        throw e;
    }finally{
        close(conn, ps, rs);
    }
    return list;
}
//-----

//-----

//Obtiene todas las publicaciones de un usuario especificado
public List<LostObject> getRecordsByUser(User u) throws Exception {
    List<LostObject> list = new ArrayList<>();
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    try{
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " select " + TablesCatalog.LOST_OBJECTS_TABLE + ".*, "
            + TablesCatalog.USERS_TABLE + ".*" + " from " +
TablesCatalog.LOST_OBJECTS_TABLE
            + " join " + TablesCatalog.USERS_TABLE + " on " +
TablesCatalog.LOST_OBJECTS_TABLE + "." + TablesCatalog.LOST_OBJECTS_TABLE_USER
            + " = " + TablesCatalog.USERS_TABLE + "." +
TablesCatalog.USERS_TABLE_ID
            + " where " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_USER + " = ? "
            + " order by " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE + " desc "
        );

        int p = 1;
        ps.setInt(p++, u.getId());

        rs = ps.executeQuery();

        while(rs.next()){
            LostObject lo = new LostObject();
            lo.setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_ID));
            lo.setDescr(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_DESCR));
            lo.setPlace(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_PLACE));
            lo.setFindDate(rs.getDate(TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE));
            lo.getUsr().setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_USER));
            lo.getUsr().setUserName(rs.getString(TablesCatalog.USERS_TABLE_USERNAME));
            lo.getUsr().setPhoneNumber(rs.getString(TablesCatalog.USERS_TABLE_PHONE));
            list.add(lo);
        }

    }catch(Exception e){
        throw e;
    }finally{
        close(conn, ps, rs);
    }
    return list;
}
//-----

//-----

@Override
public LostObject getRecordById(LostObject entity) throws Exception {
    LostObject lo = new LostObject();

```

```

Connection conn = null;
PreparedStatement ps = null;
ResultSet rs = null;
try{
    conn = Server.getInstance().getConnection();
    ps = conn.prepareStatement(
        " select " + TablesCatalog.LOST_OBJECTS_TABLE + ".*, "
        + TablesCatalog.USERS_TABLE + ".*" + " from " +
TablesCatalog.LOST_OBJECTS_TABLE
        + " join " + TablesCatalog.USERS_TABLE + " on " +
TablesCatalog.LOST_OBJECTS_TABLE + "." + TablesCatalog.LOST_OBJECTS_TABLE_USER
        + " = " + TablesCatalog.USERS_TABLE + "." +
TablesCatalog.USERS_TABLE_ID
        + " where " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_ID + " = ?"

    );
    int p = 1;
    ps.setInt(p++, entity.getId());
    rs = ps.executeQuery();
    if(rs.next()){
        lo.setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_ID));
        lo.setDescr(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_DESCR));
        lo.setPlace(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_PLACE));
        lo.setFindDate(rs.getDate(TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE));
        lo.getUsr().setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_USER));
        lo.getUsr().setUserName(rs.getString(TablesCatalog.USERS_TABLE_USERNAME));
        lo.getUsr().setPhoneNumber(rs.getString(TablesCatalog.USERS_TABLE_PHONE));
    }
} catch (Exception e) {
    throw e;
} finally {
    close(conn, ps, rs);
}
return lo;
}
//-----
//-----
@Override
public LostObject getRecord(LostObject entity) throws Exception {
    LostObject lo = new LostObject();
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    try{
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " select " + TablesCatalog.LOST_OBJECTS_TABLE + ".*, "
            + TablesCatalog.USERS_TABLE + ".*" + " from " +
TablesCatalog.LOST_OBJECTS_TABLE
            + " join " + TablesCatalog.USERS_TABLE + " on " +
TablesCatalog.LOST_OBJECTS_TABLE + "." + TablesCatalog.LOST_OBJECTS_TABLE_USER
            + " = " + TablesCatalog.USERS_TABLE + "." +
TablesCatalog.USERS_TABLE_ID
            + " where " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_DESCR + " = ? "
            + " and " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_PLACE + " = ? "
            + " and " + TablesCatalog.LOST_OBJECTS_TABLE + "." +
TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE + " between ? and ? "

        );
        int p = 1;

```

```
ps.setString(p++, entity.getDescr());
ps.setString(p++, entity.getPlace());
ps.setDate(p++, entity.getFindDate());
ps.setDate(p++, entity.getFindDate());
rs = ps.executeQuery();
if(rs.next()){
    lo.setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_ID));
    lo.setDescr(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_DESCR));
    lo.setPlace(rs.getString(TablesCatalog.LOST_OBJECTS_TABLE_PLACE));
    lo.setFindDate(rs.getDate(TablesCatalog.LOST_OBJECTS_TABLE_FINDDATE));
    lo.getUsr().setId(rs.getInt(TablesCatalog.LOST_OBJECTS_TABLE_USER));
    lo.getUsr().setUserName(rs.getString(TablesCatalog.USERS_TABLE_USERNAME));
    lo.getUsr().setPhoneNumber(rs.getString(TablesCatalog.USERS_TABLE_PHONE));
}
} catch(Exception e){
    throw e;
} finally{
    close(conn, ps, rs);
}
return lo;
}
//-----
}
}
```

**Picture.java**

```

/*-----
 *
 *          INSTITUTO TECNOLOGICO DE LA LAGUNA
 *          INGENIERIA EN SISTEMAS COMPUTACIONALES
 *          PROGRAMACION MOVILES EN ANDROID "A"
 *
 *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
 *
 *          Modelo para imágenes de objetos en publicaciones
 *
 * Archivo      : Picture.java
 * Autor        : Fernando Alfonso Caldera Olivas      15130685
 *               Jesús Iván Sifuentes Valle            15131438
 * Fecha        : 13/Dic/2018
 * Compilador   : Android Studio 3.1.3
 * Descripción  : Implementa el modelo para mapear la tabla fcal_pictures.
 * Última modif:
 * Fecha        Modificó          Motivo
 *=====
 *
 *-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.picture;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.base.BaseEntity;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.lost_object.LostObject;

public class Picture extends BaseEntity {

    //Atributos para mapear columnas de tabla
    private Integer id;
    private byte[] picture;
    private LostObject lostObject;

    //Constructores
    //-----

    public Picture() {
    }
    //-----

    //-----

    public Picture(Integer id, byte[] picture, LostObject lostObject) {
        this.id = id;
        this.picture = picture;
        this.lostObject = lostObject;
    }
    //-----

    //Getters y Setters
    //-----

    public Integer getId() {
        return id;
    }
    //-----

    //-----

    public void setId(Integer id) {
        this.id = id;
    }
}

```

```
//-----  
-----  
  
//-----  
-----  
  
public byte[] getPicture() {  
    return picture;  
}  
//-----  
-----  
  
//-----  
-----  
  
public void setPicture(byte[] picture) {  
    this.picture = picture;  
}  
//-----  
-----  
  
//-----  
-----  
  
public LostObject getLostObject() {  
    if(lostObject == null)  
        lostObject = new LostObject();  
    return lostObject;  
}  
//-----  
-----  
  
//-----  
-----  
  
public void setLostObject(LostObject lostObject) {  
    this.lostObject = lostObject;  
}  
//-----  
-----  
  
//Implementación para limpiar el modelo  
@Override  
public void clean() {  
    this.id = null;  
    this.picture = null;  
    this.lostObject = null;  
}  
}
```

**PictureDao.java**

```

/*-----
 *
 *          INSTITUTO TECNOLÓGICO DE LA LAGUNA
 *          INGENIERIA EN SISTEMAS COMPUTACIONALES
 *          PROGRAMACION MOVILES EN ANDROID "A"
 *
 *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
 *
 *          Implementación de DAO para las imágenes.
 *
 * Archivo      : PictureDao.java
 * Autor        : Fernando Alfonso Caldera Olivas      15130685
 *              : Jesús Iván Sifuentes Valle          15131438
 * Fecha        : 13/Dic/2018
 * Compilador   : Android Studio 3.1.3
 * Descripción  : Hereda de BaseDao e implementa sus métodos para la tabla
 *              : fcal_pictures
 * Ultima modif:
 * Fecha        Modificó          Motivo
 *=====
 *
 *-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.picture;

import android.app.ActionBar;
import android.media.UnsupportedSchemeException;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.base.BaseDao;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.utils.Server;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.utils.TablesCatalog;

public class PictureDao extends BaseDao<Picture> {

    //Implementación del Singleton
    //-----

    private static PictureDao instance;

    private PictureDao() {}

    public static PictureDao getInstance() {
        if(instance == null) {
            instance = new PictureDao();
        }
        return instance;
    }
    //-----

    //-----

    @Override
    public int insert(Picture entity) throws Exception {
        throw new UnsupportedOperationException();
    }
    //-----

    //-----

```



```

-----
//Método para inserción masiva de imágenes.
public int[] insertBulk(List<Picture> list) throws Exception{
    int[] c = new int[0];
    Connection conn = null;
    PreparedStatement ps = null;
    try{
        conn = Server.getInstance().getConnection();
        conn.setAutoCommit(false);
        ps = conn.prepareStatement(
            " insert into " + TablesCatalog.PICTURES_TABLE
            + " ( " + TablesCatalog.PICTURES_TABLE_PICTURE + " , "
            + TablesCatalog.PICTURES_TABLE_LOST_OBJECT + " ) "
            + " values (?, ?) "
        );
        for(Picture pic : list){

            int p = 1;
            ps.setBytes(p++, pic.getPicture());
            ps.setInt(p++, pic.getLostObject().getId());
            ps.addBatch();

        }

        c = ps.executeBatch();
        conn.commit();
    }catch(Exception e){
        conn.rollback();
        throw e;
    }finally{
        close(conn, ps, null);
    }
    return c;
}
//-----
-----

//-----
-----

@Override
public int update(Picture entity) throws Exception {
    throw new UnsupportedOperationException();
}
//-----
-----

//-----
-----

//Borra todas las imágenes de una publicación (la publicación se toma del modelo Picture,
es uno
//de sus atributos)
@Override
public int delete(Picture entity) throws Exception {
    int c = 0;
    Connection conn = null;
    PreparedStatement ps = null;
    try{
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " delete from " + TablesCatalog.PICTURES_TABLE
            + " where " + TablesCatalog.PICTURES_TABLE_LOST_OBJECT + " = ? "
        );
        int p = 1;
        ps.setInt(p++, entity.getLostObject().getId());
        c = ps.executeUpdate();
    }catch(Exception e){
        throw e;
    }finally{

```

```

        close(conn, ps, null);
    }
    return c;
}
//-----

//-----

@Override
public List<Picture> getAllRecords() throws Exception {
    throw new UnsupportedOperationException();
}
//-----

//-----

@Override
public List<Picture> getAllRecords(int offset, int rowCount) throws Exception {
    throw new UnsupportedOperationException();
}
//-----

//-----

//Obtiene todas las imágenes de una publicación especificada (la publicación se toma del
//modelo Picture, es uno de sus atributos).
public List<Picture> getRecords(Picture entity) throws Exception{
    List<Picture> list = new ArrayList<>();
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    try{
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " select * from " + TablesCatalog.PICTURES_TABLE
            + " where " + TablesCatalog.PICTURES_TABLE_LOST_OBJECT + " = ? "
        );
        int p = 1;
        ps.setInt(p++, entity.getLostObject().getId());
        rs = ps.executeQuery();
        while(rs.next()){
            Picture pi = new Picture();
            pi.setId(rs.getInt(TablesCatalog.PICTURES_TABLE_ID));
            pi.setPicture(rs.getBytes(TablesCatalog.PICTURES_TABLE_PICTURE));
            pi.getLostObject().setId(rs.getInt(TablesCatalog.PICTURES_TABLE_LOST_OBJECT));
            list.add(pi);
        }
    } catch (Exception e) {
        throw e;
    } finally {
        close(conn, ps, rs);
    }
    return list;
}
//-----

//-----

@Override
public Picture getRecordById(Picture entity) throws Exception {
    throw new UnsupportedOperationException();
}
//-----

```

```
-----  
    //-----  
-----  
    @Override  
    public Picture getRecord(Picture entity) throws Exception {  
        throw new UnsupportedOperationException();  
    }  
    //-----  
-----  
}
```

**User.java**

```

/*-----
*:
*:          INSTITUTO TECNOLOGICO DE LA LAGUNA
*:          INGENIERIA EN SISTEMAS COMPUTACIONALES
*:          PROGRAMACION MOVILES EN ANDROID "A"
*:
*:          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
*:
*:          Modelo para los usuarios de la aplicación
*:
*: Archivo      : User.java
*: Autor        : Fernando Alfonso Caldera Olivas      15130685
*:              Jesús Iván Sifuentes Valle            15131438
*: Fecha        : 13/Dic/2018
*: Compilador   : Android Studio 3.1.3
*: Descripción  : Implementa el modelo para mapear la tabla fcal_users.
*: Ultima modif:
*: Fecha        Modificó          Motivo
*:=====
*:
*:-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.user;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.base.BaseEntity;

public class User extends BaseEntity {
    //Atributos para mapear las columnas de la tabla
    private Integer id;
    private String userName;
    private String pass;
    private String phoneNumber;

    //Constructores
    //-----
    public User() {
    }
    //-----

    //-----
    public User(Integer id, String userName, String pass, String phoneNumber) {
        this.id = id;
        this.userName = userName;
        this.pass = pass;
        this.phoneNumber = phoneNumber;
    }
    //-----

    //Getters y Setters
    //-----
    public Integer getId() {
        return id;
    }
    //-----

    //-----
    public void setId(Integer id) {
        this.id = id;
    }
}

```

```
//-----  
-----  
  
//-----  
-----  
  
public String getUsername() {  
    return userName;  
}  
//-----  
-----  
  
//-----  
-----  
  
public void setUsername(String userName) {  
    this.userName = userName;  
}  
//-----  
-----  
  
//-----  
-----  
  
public String getPass() {  
    return pass;  
}  
//-----  
-----  
  
//-----  
-----  
  
public void setPass(String pass) {  
    this.pass = pass;  
}  
//-----  
-----  
  
//-----  
-----  
  
public String getPhoneNumber() {  
    return phoneNumber;  
}  
//-----  
-----  
  
//-----  
-----  
  
public void setPhoneNumber(String phoneNumber) {  
    this.phoneNumber = phoneNumber;  
}  
//-----  
-----  
  
//Implementación para limpiar el modelo  
//-----  
-----  
  
@Override  
public void clean() {  
    this.id = null;  
    this.userName = null;  
    this.pass = null;  
    this.phoneNumber = null;  
}  
//-----  
-----  
}
```

**UserDao.java**

```

/*-----
 *
 *          INSTITUTO TECNOLOGICO DE LA LAGUNA
 *          INGENIERIA EN SISTEMAS COMPUTACIONALES
 *          PROGRAMACION MOVILES EN ANDROID "A"
 *
 *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
 *
 *          Implementación del DAO para los usuarios
 *
 * Archivo      : UserDao.java
 * Autor        : Fernando Alfonso Caldera Olivas      15130685
 *              : Jesús Iván Sifuentes Valle          15131438
 * Fecha        : 13/Dic/2018
 * Compilador   : Android Studio 3.1.3
 * Descripción  : Implementa el DAO, herendando de BaseDao e implementando sus métodos.
 * Última modif:
 * Fecha        Modificó          Motivo
 *-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.user;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.List;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.base.BaseDao;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.utils.Encrypter;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.utils.Server;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.utils.TablesCatalog;

public final class UserDao extends BaseDao<User> {

    //Implementación del Singleton
    //-----

    private static UserDao instance;

    private UserDao() {}

    public static UserDao getInstance(){
        if(instance == null){
            instance = new UserDao();
        }
        return instance;
    }
    //-----

    //-----

    @Override
    public int insert(User entity) throws Exception {
        int c = 0;
        Connection conn = null;
        PreparedStatement ps = null;
        try{
            conn = Server.getInstance().getConnection();
            ps = conn.prepareStatement(
                " insert into " + TablesCatalog.USERS_TABLE + " " +
                " ( " +
                TablesCatalog.USERS_TABLE_USERNAME + " , " +
                TablesCatalog.USERS_TABLE_PASS + " , " +

```

```

        TablesCatalog.USERS_TABLE_PHONE +
        " ) values (?, ?, ?)");
    int p = 1;
    ps.setString(p++, entity.getUserName());
    //Encriptamos el password
    entity.setPass(Encrypter.encrypt(entity.getPass()));
    ps.setString(p++, entity.getPass());
    ps.setString(p++, entity.getPhoneNumber());
    c = ps.executeUpdate();
    User u = getRecord(entity);
    entity.setId(u.getId());
} catch (Exception e) {
    throw e;
} finally {
    close(conn, ps, null);
}
return c;
}
//-----

//-----

@Override
public int update(User entity) throws Exception {
    int c = 0;
    Connection conn = null;
    PreparedStatement ps = null;
    try {
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " update " + TablesCatalog.USERS_TABLE
            + " set "
            + TablesCatalog.USERS_TABLE_USERNAME + " = ?, "
            + TablesCatalog.USERS_TABLE_PASS + " = ?, "
            + TablesCatalog.USERS_TABLE_PHONE + " = ? "
            + " where " + TablesCatalog.USERS_TABLE_ID + " = ?"
        );
        int p = 1;
        ps.setString(p++, entity.getUserName());
        //Encriptamos el password
        entity.setPass(Encrypter.encrypt(entity.getPass()));
        ps.setString(p++, entity.getPass());
        ps.setString(p++, entity.getPhoneNumber());
        ps.setInt(p++, entity.getId());
        c = ps.executeUpdate();
    } catch (Exception e) {
        throw e;
    } finally {
        close(conn, ps, null);
    }
    return c;
}
//-----

//-----

@Override
public int delete(User entity) throws Exception {
    int c = 0;
    Connection conn = null;
    PreparedStatement ps = null;
    try {
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " delete from " + TablesCatalog.USERS_TABLE

```

```

        + " where " + TablesCatalog.USERS_TABLE_ID + " = ?"
    );
    int p = 1;
    ps.setInt(p++, entity.getId());
    c = ps.executeUpdate();
    entity.clean();
} catch (Exception e) {
    throw e;
} finally {
    close(conn, ps, null);
}
return c;
}
//-----

//-----

@Override
public List<User> getAllRecords() throws Exception {
    throw new UnsupportedOperationException();
}
//-----

//-----

@Override
public List<User> getAllRecords(int offset, int rowCount) throws Exception {
    throw new UnsupportedOperationException();
}
//-----

//-----

@Override
public User getRecordById(User entity) throws Exception {
    User u = new User();
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;
    try {
        conn = Server.getInstance().getConnection();
        ps = conn.prepareStatement(
            " select * from " + TablesCatalog.USERS_TABLE
            + " where " + TablesCatalog.USERS_TABLE_ID + " = ?"
        );
        int p = 1;
        ps.setInt(p++, entity.getId());
        rs = ps.executeQuery();

        if (rs.next()) {
            u.setId(rs.getInt(TablesCatalog.USERS_TABLE_ID));
            u.setUsername(rs.getString(TablesCatalog.USERS_TABLE_USERNAME));
            u.setPass(rs.getString(TablesCatalog.USERS_TABLE_PASS));
            u.setPhoneNumber(rs.getString(TablesCatalog.USERS_TABLE_PHONE));
        }
    } catch (Exception e) {
        throw e;
    } finally {
        close(conn, ps, rs);
    }
    return u;
}
//-----

```



```
-----  
//-----  
-----  
@Override  
public User getRecord(User entity) throws Exception {  
  
    User u = new User();  
    Connection conn = null;  
    PreparedStatement ps = null;  
    ResultSet rs = null;  
  
    try{  
        conn = Server.getInstance().getConnection();  
        ps = conn.prepareStatement(  
            " select * from " + TablesCatalog.USERS_TABLE  
            + " where " + TablesCatalog.USERS_TABLE_USERNAME + " = ?"  
        );  
  
        int p = 1;  
        ps.setString(p++, entity.getUserName());  
        rs = ps.executeQuery();  
  
        if(rs.next()){  
            u.setId(rs.getInt(TablesCatalog.USERS_TABLE_ID));  
            u.setUserName(rs.getString(TablesCatalog.USERS_TABLE_USERNAME));  
            u.setPass(rs.getString(TablesCatalog.USERS_TABLE_PASS));  
            u.setPhoneNumber(rs.getString(TablesCatalog.USERS_TABLE_PHONE));  
        }  
    } catch (Exception e) {  
        throw e;  
    } finally {  
        close(conn, ps, rs);  
    }  
    return u;  
}  
//-----  
-----  
}
```

**Encrypter.java**

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERIA EN SISTEMAS COMPUTACIONALES
*:          PROGRAMACION MOVILES EN ANDROID "A"
*:
*:          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
*:
*:          Encriptador de cadenas
*:
*: Archivo      : Encrypter.java
*: Autor        : Fernando Alfonso Caldera Olivas      15130685
*:              Jesús Iván Sifuentes Valle            15131438
*: Fecha        : 13/Dic/2018
*: Compilador   : Android Studio 3.1.3
*: Descripción  : Encriptador propio de cadenas. Se usa para encriptar las contraseñas
*:              de los usuarios antes de insertarlos en la tabla.
*: Ultima modif:
*: Fecha        Modificó          Motivo
*:=====
*:
*:-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.utils;

public final class Encrypter {
    //Implementación de clase estática.
    private Encrypter() {
    }

    //Método estático para encriptar.
    public static final String encrypt(String s){
        String encryptedString = "";
        for(int i = 0, j = s.length() - 1; i < s.length() / 2; i++, j--){
            encryptedString += (char)((s.charAt(i) ^ Integer.MAX_VALUE) % 256);
            encryptedString += (char)((s.charAt(j) ^ Integer.MAX_VALUE) % 256);
        }
        return encryptedString;
    }
}

```

**Server.java**

```

/*-----
: *
: *          INSTITUTO TECNOLÓGICO DE LA LAGUNA
: *          INGENIERIA EN SISTEMAS COMPUTACIONALES
: *          PROGRAMACION MOVILES EN ANDROID "A"
: *
: *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
: *
: *          Despachador de Conexiones
: *
: *
: * Archivo      : Server.java
: * Autor        : Fernando Alfonso Caldera Olivas      15130685
: *              : Jesús Iván Sifuentes Valle           15131438
: * Fecha        : 13/Dic/2018
: * Compilador   : Android Studio 3.1.3
: * Descripción  : Clase estática que implementa singleton, no es heredable.
: *              : Tiene método para devolver conexión.
: * Última modif:
: * Fecha        Modificó          Motivo
: *=====
: *
: *-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.utils;

import android.os.StrictMode;

import net.sourceforge.jtds.jdbc.Driver;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public final class Server {

    //Datos de conexión
    private final String HOST;
    private final String PORT;
    private final String DB;
    private final String USER;
    private final String PASS;

    //Implementación del singleton
    //-----
    private static Server instance;

    //Constructor
    private Server() throws SQLException {
        //Inicializamos las Constantes.
        HOST = "187.189.201.198";
        PORT = "1433";
        DB = "pmadb";
        USER = "pma";
        PASS = "android";

        //Registramos solo una vez el driver
        try{
            DriverManager.registerDriver(new Driver());
        }catch(SQLException e){
            throw e;
        }
    }

    public static final Server getInstance() throws SQLException {
        if(instance == null){

```

```
        instance = new Server();
    }
    return instance;
}
//-----

//-----

//Obtenemos conexión con el DriverManager, usando los valores de las constantes y la cadena
//de conexión.
public final Connection getConnection() throws SQLException{
    Connection conn = null;
    try{
        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
        conn = DriverManager.getConnection("jdbc:jtds:sqlserver://" + HOST + ":" + PORT +
";"
        + "databaseName=" + DB + ";user=" + USER
        + ";password=" + PASS + ";");
    }catch(SQLException e){
        throw e;
    }
    return conn;
}
//-----

}
```

## TablesCatalog.java

```

/*-----
 *
 *          INSTITUTO TECNOLOGICO DE LA LAGUNA
 *          INGENIERIA EN SISTEMAS COMPUTACIONALES
 *          PROGRAMACION MOVILES EN ANDROID "A"
 *
 *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
 *
 *          Catalogo de tablas
 *
 * Archivo      : TablesCatalog.java
 * Autor        : Fernando Alfonso Caldera Olivas      15130685
 *              : Jesús Iván Sifuentes Valle          15131438
 * Fecha        : 13/Dic/2018
 * Compilador   : Android Studio 3.1.3
 * Descripción  : Clase estática que solo da acceso a constantes para nombrar tablas y sus
 *              : columnas. Se utiliza para aumentar considerablemente la escalabilidad y
 *              : facilidad de mantenimiento.
 * Última modif:
 * Fecha        Modificó          Motivo
 * =====
 *
 *-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.utils;

public final class TablesCatalog {
    //Implementación de clase estática.
    private TablesCatalog(){
    }

    //Mapeo de metadatos de tabla "users"
    public static final String USERS_TABLE = "fcal_users";
    public static final String USERS_TABLE_ID = "id";
    public static final String USERS_TABLE_USERNAME = "userName";
    public static final String USERS_TABLE_PASS = "pass";
    public static final String USERS_TABLE_PHONE = "phoneNumber";

    //Mapeo de metadatos de tabla "lost_objects"
    public static final String LOST_OBJECTS_TABLE = "fcal_lost_objects";
    public static final String LOST_OBJECTS_TABLE_ID = "id";
    public static final String LOST_OBJECTS_TABLE_DESCR = "descr";
    public static final String LOST_OBJECTS_TABLE_PLACE = "place";
    public static final String LOST_OBJECTS_TABLE_FINDDATE = "findDate";
    public static final String LOST_OBJECTS_TABLE_USER = "usr";

    //Mapeo de metadatos de tabla "pictures"
    public static final String PICTURES_TABLE = "fcal_pictures";
    public static final String PICTURES_TABLE_ID = "id";
    public static final String PICTURES_TABLE_PICTURE = "picture";
    public static final String PICTURES_TABLE_LOST_OBJECT = "lostObject";
}

```

**CommonData.java**

```

/*-----
 *
 *          INSTITUTO TECNOLÓGICO DE LA LAGUNA
 *          INGENIERIA EN SISTEMAS COMPUTACIONALES
 *          PROGRAMACION MOVILES EN ANDROID "A"
 *
 *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
 *
 *          Singleton de sesión
 *
 * Archivo      : CommonData.java
 * Autor        : Fernando Alfonso Caldera Olivas      15130685
 *               Jesús Iván Sifuentes Valle            15131438
 * Fecha        : 13/Dic/2018
 * Compilador   : Android Studio 3.1.3
 * Descripción  : Clase que implementa singleton para guardar datos de sesión, o que se van a
 *               pasar entre actividades.
 * Última modif:
 * Fecha        Modificó          Motivo
 *=====
 *
 *-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.lost_object.LostObject;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.user.User;

public final class CommonData {
    //Implementación del Singleton
    //-----
    -----

    private static CommonData instance;
    private CommonData() {}

    public static CommonData getInstance() {
        if(instance == null) {
            instance = new CommonData();
        }
        return instance;
    }
    //-----
    -----

    //Datos de la sesión
    public User user;

    //Datos de inspeccionar
    public LostObject lostObject;
}

```

**AboutActivity.java**

```

/*-----
*:
*:          INSTITUTO TECNOLOGICO DE LA LAGUNA
*:          INGENIERIA EN SISTEMAS COMPUTACIONALES
*:          PROGRAMACION MOVILES EN ANDROID "A"
*:
*:          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
*:
*:          Actividad Acerca de...
*:
*: Archivo      : AboutActivity.java
*: Autor       : Fernando Alfonso Caldera Olivas      15130685
*:             Jesús Iván Sifuentes Valle            15131438
*: Fecha        : 13/Dic/2018
*: Compilador   : Android Studio 3.1.3
*: Descripción  : Actividad que muestra información acerca de los autores de la aplicación,
donde
*:              estudiaron, bajo que materia hicieron la app y el nombre del catedrático.
*: Ultima modif:
*: Fecha      Modificó          Motivo
*:=====
*:
*:-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class AboutActivity extends AppCompatActivity {

    //-----
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about);
    }
    //-----
}

```

**activity\_about.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".AboutActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:orientation="vertical"
        android:weightSum="3">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1.1"
            android:orientation="horizontal"
            android:weightSum="2">

            <ImageView
                android:id="@+id/imageView3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                app:srcCompat="@drawable/itl" />

            <TextView
                android:id="@+id/textView3"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="@string/about_head" />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="0.7"
            android:orientation="horizontal"
            android:weightSum="2">

            <ImageView
                android:id="@+id/imageView4"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                app:srcCompat="@drawable/equipo" />

            <TextView
                android:id="@+id/textView4"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="@string/about_body" />
        </LinearLayout>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
```



```
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView5"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="@string/about_foot" />
    </LinearLayout>
</LinearLayout>
</LinearLayout>
```

**AddPostActivity.java**

```

/*-----
 *
 *          INSTITUTO TECNOLÓGICO DE LA LAGUNA
 *          INGENIERIA EN SISTEMAS COMPUTACIONALES
 *          PROGRAMACION MOVILES EN ANDROID "A"
 *
 *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
 *
 *          Actividad para agregar publicación
 *
 * Archivo      : AddPostActivity.java
 * Autor        : Fernando Alfonso Caldera Olivas      15130685
 *              : Jesús Iván Sifuentes Valle           15131438
 * Fecha        : 13/Dic/2018
 * Compilador   : Android Studio 3.1.3
 * Descripción  : Actividad que permite que el usuario agregue publicaciones, añadiendo
información
 *              para la misma como descripción, lugar y fecha y, además, le da la posibilidad
de
 *              adjuntar imágenes.
 * Última modif:
 * Fecha        Modificó          Motivo
 *=====
 *
 *-----*/

```

```
package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp;
```

```

import android.app.AlertDialog;
import android.content.ActivityNotFoundException;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import java.sql.Date;
import java.sql.Time;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.List;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.lost_object.LostObject;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.lost_object.LostObjectDao;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.picture.Picture;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.picture.PictureDao;

```

```

public class AddPostActivity extends AppCompatActivity {

    //Campos para la publicación
    private EditText edDescr;
    private EditText edPlace;

```

```

private EditText edDate;
//Campos para capturar las imágenes adjuntas
private LinearLayout llPictures;
private List<Uri> attachedFiles;
//Código para selección de archivo
public static final int SELECTED_FILE_CODE = 1;

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

    //Obtenemos las referencias
    edDescr = findViewById(R.id.edDescr);
    edPlace = findViewById(R.id.edPlace);
    edDate = findViewById(R.id.edDate);
    llPictures = findViewById(R.id.llPictures);

    //Inicializamos el arreglo de archivos adjuntos
    attachedFiles = new ArrayList<>();
}

public void btnPublishClick(View view) {

    //Validamos la longitud de la cadena de entrada.
    if(edDate.getText().toString().length() != 10 ){
        edDate.setError("Ingrese una fecha con el formato dd/mm/aaaa");
        return;
    }

    //Tras pasar la validación, obtenemos los valores y los almacenamos
    //en un objeto LostObject
    LostObject lo = new LostObject();
    lo.setDescr(edDescr.getText().toString());
    lo.setPlace(edPlace.getText().toString());
    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
    long ms = 0;
    try {
        ms = sdf.parse(edDate.getText().toString()).getTime();
    } catch (ParseException e) {
        edDate.setError("Ingrese una fecha con el formato dd/mm/aaaa");
        return;
    }
    lo.setFindDate(new Date(ms));
    //Le asignamos como usuario dueño de esta publicación, al usuario que está
    //en sesión.
    lo.setUsr(CommonData.getInstance().user);

    try {
        //Insertamos la publicación.
        LostObjectDao.getInstance().insert(lo);

        ByteArrayOutputStream baos = null;
        InputStream is = null;
        try{
            List<Picture> pics = new ArrayList<>();
            //Iteramos los archivos adjuntos y los convertimos
            //a arreglo de bytes, para poder insertarlos como BLOB en
            //la tabla de imágenes, los agregamos a una lista de Pictures.
            for(Uri u : attachedFiles){
                is = getContentResolver().openInputStream(u);
                baos = new ByteArrayOutputStream();
                byte[] buffer = new byte[1024];
                int n;
                while(-1 != (n = is.read(buffer))){
                    baos.write(buffer, 0, n);
                }
            }
        }
    }
}

```

```

    }
    try{
        is.close();
    }catch(Exception e){
        Log.e("GuardarImagenes", e.getMessage());
    }
    Picture pic = new Picture();
    pic.setLostObject(lo);
    pic.setPicture(baos.toByteArray());
    pics.add(pic);

    try{
        baos.close();
    }catch(Exception e){

    }
}
//Insertamos masivamente las imágenes
PictureDao.getInstance().insertBulk(pics);
}catch(Exception e){
    //Si hay error al guardar las imágenes, borramos la publicación, mandamos error
y salimos.
    Toast.makeText(this, "Ocurrió un error al guardar las imágenes.",
Toast.LENGTH_LONG).show();
    try{
        LostObjectDao.getInstance().delete(lo);
    }catch(Exception e1){

    }
    return;
}

//Si tod0 salió bien, Mandamos alert dialog.
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle("Publicación exitosa.")
    .setMessage("La publicación se ha guardado con éxito.")
    .setPositiveButton("Aceptar", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
            AddPostActivity.this.finish();
        }
    })
    .setCancelable(false)
    .create().show();
} catch (Exception e) {
    //Si hay error desde la inserción de la publicación, mandamos el error.
    Toast.makeText(this, "Ocurrió un error al guardar la publicación.",
Toast.LENGTH_LONG).show();
}
}

public void btnAddPictureClick(View view){
    //Creamos el intent para abrir el explorador de archivos.
    Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
    //Agregamos el filtro para para que solo permita
    //abrir imágenes.
    intent.setType("image/*");
    //Agregamos la categoría de la aplicación, default.
    intent.addCategory(Intent.CATEGORY_DEFAULT);

    //Tratamos de abrir el explorador de archivos. Si no se puede, lanzamos un toast
    //Con la excepción.
    try {
        startActivityForResult(Intent.createChooser(intent, "Seleccione una opción."),
SELECTED_FILE_CODE);
    }catch (ActivityNotFoundException e){

```

```

        Toast.makeText(this, "Explorador de archivos no encontrado.\nPor favor instale un
explorador de archivos.", Toast.LENGTH_LONG).show();
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    //Si el código de petición es SELECTED_FILE_CODE
    if(requestCode == SELECTED_FILE_CODE){
        //Si el código de resultado es igual a RESULT_OK
        if(resultCode == RESULT_OK){
            //Verificamos que el archivo no se encuentre adjuntado
            boolean isAttached = false;
            for(Uri u : attachedFiles){
                if(u.getPath().equals(data.getData().getPath())){
                    isAttached = true;
                    break;
                }
            }
            if(!isAttached) {
                //Agregamos un nuevo hijo al layout que sea otro layout linear, horizontal,
                //Con 2 views, uno que sea un textview para visualizar el nombre del
                archivo y el
                layout.
                //otro un boton para remover el adjunto que estará adentro de otro linear

                LinearLayout linearLayout = new LinearLayout(this);
                linearLayout.setOrientation(LinearLayout.HORIZONTAL);
                linearLayout.setWeightSum(2f);

                TextView textView = new TextView(this);
                String[] pathSegments = data.getData().getPath().split("/");
                String fileName = pathSegments[pathSegments.length - 1];
                textView.setText(fileName);
                textView.setLayoutParams(new LinearLayout.LayoutParams(
                    LinearLayout.LayoutParams.MATCH_PARENT,
                    LinearLayout.LayoutParams.WRAP_CONTENT,
                    1f
                ));

                LinearLayout llButton = new LinearLayout(this);
                llButton.setOrientation(LinearLayout.VERTICAL);
                llButton.setLayoutParams(new LinearLayout.LayoutParams(
                    LinearLayout.LayoutParams.MATCH_PARENT,
                    LinearLayout.LayoutParams.WRAP_CONTENT,
                    1
                ));
                llButton.setGravity(Gravity.CENTER_VERTICAL | Gravity.END);

                Button btnDelete = new Button(this);
                btnDelete.setBackground(getDrawable(R.drawable.tacha));
                btnDelete.setLayoutParams(new LinearLayout.LayoutParams(
                    50,
                    50
                ));

                btnDelete.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        int index = llPictures.indexOfChild((View)
v.getParent().getParent());
                        attachedFiles.remove(index);
                        llPictures.removeView((View) v.getParent().getParent());
                    }
                });

                llButton.addView(btnDelete);
            }

```

```
//Agregamos el botón y el textview al layout creado (horizontal).  
//Esto simula la adición de un nuevo renglón a una tabla.  
linearLayout.addView(textView);  
linearLayout.addView(llButton);  
//Agregamos el nuevo "renglón" al layout que lista dichos  
//Elementos adjuntos.  
llPictures.addView(linearLayout);  
//Agregamos la ruta del archivo en la lista de archivos adjuntos.  
attachedFiles.add(data.getData());  
    }  
}  
}
```

**activity\_add\_post.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".AddPostActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:gravity="center"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView12"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/add_post_screen"
            android:textSize="24sp" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:orientation="vertical">

        <EditText
            android:id="@+id/edDescr"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="@string/description"
            android:inputType="textMultiLine" />

        <EditText
            android:id="@+id/edPlace"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="@string/place"
            android:inputType="textMultiLine" />

        <EditText
            android:id="@+id/edDate"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="@string/date"
            android:inputType="date" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/layout_bottom"
            android:orientation="vertical">
```

```
<Button
    android:id="@+id/btnAddPicture"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="btnAddPictureClick"
    android:text="@string/add_photo" />

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="200dp">

    <LinearLayout
        android:id="@+id/llPictures"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" />
</ScrollView>

</LinearLayout>

<Button
    android:id="@+id/btnPublish"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="btnPublishClick"
    android:text="@string/publish" />
</LinearLayout>
</LinearLayout>
```



**InspectPostActivity.java**

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERIA EN SISTEMAS COMPUTACIONALES
*:          PROGRAMACION MOVILES EN ANDROID "A"
*:
*:          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
*:
*:          Actividad para inspeccionar publicaciones.
*:
*: Archivo      : InspectPostActivity.java
*: Autor        : Fernando Alfonso Caldera Olivas      15130685
*:              Jesús Iván Sifuentes Valle            15131438
*: Fecha        : 13/Dic/2018
*: Compilador   : Android Studio 3.1.3
*: Descripción  : Actividad que permite inspeccionar una publicación, mostrando los datos de
contacto
*:              y su galería.
*: Última modif:
*: Fecha        Modificó          Motivo
*:=====
*:
*:-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp;

import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.drawable.Drawable;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import java.util.List;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.picture.Picture;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.picture.PictureDao;

public class InspectPostActivity extends AppCompatActivity {

    //Campos que visualizan los datos de la publicación
    private TextView tvwDescr;
    private TextView tvwPlace;
    private TextView tvwDate;
    //Campo que visualiza el dato de usuario
    private TextView tvwPhone;
    //Linear layout de las imágenes en el carrusel (Horizontal Scroll View)
    private LinearLayout llPictures;

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

        //Obtenemos las referencias
        tvwDescr = findViewById(R.id.tvwDescr);
        tvwPlace = findViewById(R.id.tvwPlace);
        tvwDate = findViewById(R.id.tvwDate);

```

```

llPictures = findViewById(R.id.llPictures);
tvwPhone = findViewById(R.id.tvwPhone);

//Establecemos los valores que están en memoria
tvwDescr.setText(CommonData.getInstance().lostObject.getDescr());
tvwPlace.setText(CommonData.getInstance().lostObject.getPlace());
tvwDate.setText(CommonData.getInstance().lostObject.getFindDate().toString());
tvwPhone.setText(CommonData.getInstance().lostObject.getUsr().getPhoneNumber());

//Obtenemos las imágenes de la publicación y las insertamos
//En el scrollablehorizontal.
try{
    Picture pi = new Picture();
    pi.setLostObject(CommonData.getInstance().lostObject);
    List<Picture> list = PictureDao.getInstance().getRecords(pi);
    llPictures.removeAllViews();
    llPictures.setWeightSum(list.size());
    for(Picture pic : list){
        ImageView imgView = new ImageView(this);
        imgView.setLayoutParams(new LinearLayout.LayoutParams(400,
LinearLayout.LayoutParams.MATCH_PARENT, 1));
        Bitmap bitmap = BitmapFactory.decodeByteArray(pic.getPicture(), 0,
pic.getPicture().length);
        imgView.setImageBitmap(bitmap);
        llPictures.addView(imgView);
    }
} catch (Exception e) {
    Toast.makeText(this, "Ocurrió un error al recuperar las imágenes.",
Toast.LENGTH_LONG).show();
}

}

//-----
//-----
//-----

public void btnCallClick(View view){
    //Creamos un intent y realizamos una llamada.
    Intent intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:" +
CommonData.getInstance().lostObject.getUsr().getPhoneNumber()));
    startActivity(intent);
}

//-----
}

```

**activity\_inspect\_post.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".InspectPostActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:gravity="center"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView13"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/inspect_post_screen"
            android:textSize="24sp" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/textView6"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/description" />

            <TextView
                android:id="@+id/tvwDescr"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginStart="@dimen/layout_start_end"
                android:text="Descr" />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/textView8"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/place" />
```

```
<TextView
    android:id="@+id/tvwPlace"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="@dimen/layout_start_end"
    android:text="Place" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/textView10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/date" />

    <TextView
        android:id="@+id/tvwDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/layout_start_end"
        android:text="Date" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="@dimen/layout_bottom"
    android:gravity="center_vertical"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/phone" />

    <TextView
        android:id="@+id/tvwPhone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/layout_start_end"
        android:text="Phone" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_vertical|end"
        android:orientation="vertical">

        <Button
            android:id="@+id/btnCall"
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:layout_marginEnd="@dimen/layout_start_end"
            android:background="@drawable/llamar"
            android:onClick="btnCallClick" />
        </LinearLayout>
    </LinearLayout>

</LinearLayout>

<HorizontalScrollView
    android:layout_width="match_parent"
```

```
        android:layout_height="match_parent">

        <LinearLayout
            android:id="@+id/llPictures"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:orientation="horizontal">

            </LinearLayout>
        </HorizontalScrollView>

    </LinearLayout>
</LinearLayout>
```

```

LoginActivity.java/*-----
-----
: *          INSTITUTO TECNOLÓGICO DE LA LAGUNA
: *          INGENIERIA EN SISTEMAS COMPUTACIONALES
: *          PROGRAMACION MOVILES EN ANDROID "A"
: *
: *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
: *
: *          Actividad para iniciar sesión
: *
: * Archivo      : LoginActivity.java
: * Autor        : Fernando Alfonso Caldera Olivas      15130685
: *              : Jesús Iván Sifuentes Valle          15131438
: * Fecha        : 13/Dic/2018
: * Compilador   : Android Studio 3.1.3
: * Descripción  : Actividad que permite iniciar sesión en una cuenta registrada.
: *              : Lo anterior lo realiza buscando si existe el usuario y comparando las
: *              : contraseñas. Si el usuario y contraseña son correctos, se guarda en sesión
: *              : el usuario.
: * Ultima modif:
: * Fecha        Modificó          Motivo
: *=====
: *
: *-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp;

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

import java.sql.Connection;
import java.sql.PreparedStatement;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.user.User;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.user.UserDao;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.utils.Encrypter;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.utils.Server;

public class LoginActivity extends AppCompatActivity {

    //Campos de logeo.
    private EditText edUser;
    private EditText edPass;

    //-----

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

        //Obtenemos la referencia de los edits
        edUser = findViewById(R.id.edUser);
        edPass = findViewById(R.id.edPass);
    }

    //-----

    //-----

    public void btnLoginClick(View view){

```

```

//Verificamos que los campos no estén vacíos.
boolean fieldsOk = true;
if(edUser.getText() == null || edUser.getText().toString().trim().equals("")){
    edUser.setError("Especifique el usuario.");
    fieldsOk = false;
}
if(edPass.getText() == null || edPass.getText().toString().trim().equals("")){
    edPass.setError("Especifique la contraseña.");
    fieldsOk = false;
}

//Si hay un campo que está vacío, detenemos la ejecución.
if(!fieldsOk){
    return;
}

//Obtenemos usuario de la base de datos
User u = new User();
u.setUserName(edUser.getText().toString());
u.setPass(Encrypter.encrypt(edPass.getText().toString()));

User udb = new User();
udb.setUserName(u.getUserName());
try{
    udb = UserDao.getInstance().getRecord(udb);
}catch(Exception e){
    Toast.makeText(this, "Ocurrió un error al recuperar los datos del usuario.",
Toast.LENGTH_LONG).show();
}

//Verificamos que el usuario exista, si no, mandamos error.
if(udb.getId() == null){
    Toast.makeText(this, "El usuario o la contraseña son incorrectos.",
Toast.LENGTH_LONG).show();
    return;
}

//Los password no coinciden, mandamos error.
if(!u.getPass().equals(udb.getPass())){
    Toast.makeText(this, "El usuario o la contraseña son incorrectos.",
Toast.LENGTH_LONG).show();
    return;
}

//Tod0 está bien, continuamos con la siguiente actividad y guardamos
//al usuario en sesión en memoria.
CommonData.getInstance().user = udb;
Intent intent = new Intent(this, MainActivity.class);
startActivity(intent);
finish();
}
//-----

//-----

public void btnSignupClick(View view){
    Intent intent = new Intent(this, SignupActivity.class);
    startActivity(intent);
}
//-----

//-----

public void btnAboutClick(View view){
    Intent intent = new Intent(this, AboutActivity.class);

```

```
        startActivity(intent);  
    }  
    //-----  
-----  
}
```



**activity\_login.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".LoginActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:gravity="center"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/login_screen"
            android:textSize="24sp" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:gravity="center"
        android:orientation="vertical">

        <EditText
            android:id="@+id/edUser"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/layout_bottom"
            android:ems="10"
            android:hint="@string/usr"
            android:inputType="textPersonName" />

        <EditText
            android:id="@+id/edPass"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/layout_bottom"
            android:ems="10"
            android:hint="@string/pass"
            android:inputType="textPassword" />

        <Button
            android:id="@+id/btnLogin"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/layout_bottom"
            android:onClick="btnLoginClick"
            android:text="@string/login" />

        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:text="@string/signup_ad" />

    <Button
        android:id="@+id/btnSignup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:onClick="btnSignupClick"
        android:text="@string/signup" />

    <Button
        android:id="@+id/btnAbout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:onClick="btnAboutClick"
        android:text="@string/about" />

</LinearLayout>

</LinearLayout>
```

**MainActivity.java**

```

/*-----
 *
 *          INSTITUTO TECNOLOGICO DE LA LAGUNA
 *          INGENIERIA EN SISTEMAS COMPUTACIONALES
 *          PROGRAMACION MOVILES EN ANDROID "A"
 *
 *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
 *
 *          Actividad principal. Menú principal de la aplicación.
 *
 * Archivo      : MainActivity.java
 * Autor        : Fernando Alfonso Caldera Olivas      15130685
 *              : Jesús Iván Sifuentes Valle          15131438
 * Fecha        : 13/Dic/2018
 * Compilador   : Android Studio 3.1.3
 * Descripción  : Actividad que sirve como punto de partida para las demás actividades.
 *              : Este menú principal permite, desde aquí, agregar nuevas publicaciones, ver
 *              : lo que has publicado y borrar tus publicaciones, buscar publicaciones usando
 *              : diferentes filtros e inspeccionar las publicaciones.
 * Ultima modif:
 * Fecha        Modificó          Motivo
 *=====
 *
 */-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import org.w3c.dom.Text;

import java.util.List;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.lost_object.LostObject;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.lost_object.LostObjectDao;

public class MainActivity extends AppCompatActivity {

    //Campos del layout.
    private TextView tvwUser;
    private LinearLayout llPosts;
    private EditText edDescr;
    private EditText edPlace;
    private EditText edStartDate;
    private EditText edEndDate;

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

        //Obtenemos las referencias
        tvwUser = findViewById(R.id.tvwUser);
        llPosts = findViewById(R.id.llPosts);
        edDescr = findViewById(R.id.edDescr);

```

```

edPlace = findViewById(R.id.edPlace);
edStartDate = findViewById(R.id.edStartDate);
edEndDate = findViewById(R.id.edEndDate);

//Establecemos el nombre de usuario en sesión, para que se muestre.
tvwUser.setText(CommonData.getInstance().user.getUserName());

//Cargamos todos los registros (Se va a cambiar en el futuro para usar paginación.)
try{
    List<LostObject> list = LostObjectDao.getInstance().getAllRecords();
    loadPosts(list);
} catch (Exception e){
    Toast.makeText(this, "Ocurrió un error al obtener las publicaciones.",
Toast.LENGTH_LONG).show();
}

}

//-----

//-----

private void loadPosts(List<LostObject> list){
    //Limpiamos la tabla
    llPosts.removeAllViews();

    //Ingresamos los renglones
    for (final LostObject lo : list){
        //Agregamos un linea layout horizontal, el cual va a funcionar como
        //un renglón, como si fuéramos a llenar una tabla.
        //El renglón tendrá 4 elementos.
        LinearLayout ll = new LinearLayout(this);
        ll.setOrientation(LinearLayout.HORIZONTAL);
        ll.setWeightSum(4.0f);

        //Agregamos la fecha cuando se encontró
        TextView tvwDate = new TextView(this);
        tvwDate.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT, 1));
        tvwDate.setText(lo.getFindDate().toString());

        //Agregamos la descripción del objeto
        TextView tvwDescr = new TextView(this);
        tvwDescr.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT, 1));
        tvwDescr.setText(lo.getDescr());

        //Agregamos el lugar donde se encontró.
        TextView tvwPlace = new TextView(this);
        tvwPlace.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT, 1));
        tvwPlace.setText(lo.getPlace());

        //Agregamos botón de inspeccionar
        //Agregamos este layout que va a contener el botón, esto para poder centrar
        //el botón y mandarlo al final (que esté hasta la orilla derecha).
        LinearLayout llButton = new LinearLayout(this);
        llButton.setOrientation(LinearLayout.VERTICAL);
        llButton.setGravity(Gravity.CENTER_VERTICAL | Gravity.END);
        llButton.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT, 1));
        Button btnInspect = new Button(this);

```

```

        btnInspect.setLayoutParams(new LinearLayout.LayoutParams(50,
            50));
        btnInspect.setBackground(getDrawable(R.drawable.lupa));
        btnInspect.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //Al hacer click, establece el lostObject en memoria como el lostObject
                //de esta iteración.
                CommonData.getInstance().lostObject = lo;
                //Inicia la actividad de inspección.
                Intent intent = new Intent(MainActivity.this, InspectPostActivity.class);
                startActivity(intent);
            }
        });
        //Agregamos el botón en el layout que lo va a contener.
        llButton.addView(btnInspect);

        //Agregamos todos los elementos anteriores al "row"
        ll.addView(tvwDate);
        ll.addView(tvwDescr);
        ll.addView(tvwPlace);
        ll.addView(llButton);

        //Agregamos el nuevo row al LayoutPadre.
        llPosts.addView(ll);
    }
}
//-----
//-----

public void btnSearchClick(View view){
    //Validamos que si se busca por fecha, ambos campos de fecha estén llenos
    if(!edStartDate.getText().toString().equals("")){
        if(edEndDate.getText().toString().equals("")){
            edEndDate.setError("Para buscar por fecha, debe llenar ambos campos.");
            return;
        }
    }
    if(!edEndDate.getText().toString().equals("")){
        if(edStartDate.getText().toString().equals("")){
            edStartDate.setError("Para buscar por fecha, debe llenar ambos campos.");
        }
    }
}

//Obtenemos los valores si pasan la validación
String descr = edDescr.getText().toString();
String place = edPlace.getText().toString();
String startDate = edStartDate.getText().toString();
String endDate = edEndDate.getText().toString();

//Buscamos según los parámetros.
try{
    List<LostObject> list = LostObjectDao.getInstance().getRecordsInRange(startDate,
endDate, descr, place);
    loadPosts(list);
} catch (Exception e){
    Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG).show();
}
}
//-----
//-----

```

```
-----
    @Override
    protected void onResume() {
        super.onResume();
        //Cargamos todos los registros (Se va a cambiar en el futuro por una implementación con
        paginación.)
        try{
            List<LostObject> list = LostObjectDao.getInstance().getAllRecords();
            loadPosts(list);
        } catch (Exception e) {
            Toast.makeText(this, "Ocurrió un error al obtener las publicaciones.",
            Toast.LENGTH_LONG).show();
        }
    }
    //-----
    -----
    //-----
    -----
    public void btnExitClick(View view){
        CommonsData.getInstance().user = null;
        Intent intent = new Intent(this, LoginActivity.class);
        startActivity(intent);
        finish();
    }
    //-----
    -----
    //-----
    -----
    public void btnAddPostClick(View view){
        Intent intent = new Intent(this, AddPostActivity.class);
        startActivity(intent);
    }
    //-----
    -----
    //-----
    -----
    public void btnMyPostsClick(View view){
        Intent intent = new Intent(this, MyPostsActivity.class);
        startActivity(intent);
    }
    //-----
    -----
}
```

**activity\_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:gravity="center"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView11"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/main_screen"
            android:textSize="24sp" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:gravity="center"
        android:orientation="horizontal"
        android:weightSum="2">

        <TextView
            android:id="@+id/tvwUser"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="usuario"
            android:textSize="20sp" />

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:gravity="end"
            android:orientation="horizontal">

            <Button
                android:id="@+id/btnMyPosts"
                android:layout_width="30dp"
                android:layout_height="match_parent"
                android:layout_marginEnd="@dimen/layout_start_end"
                android:background="@drawable/lista"
                android:onClick="btnMyPostsClick" />

            <Button
                android:id="@+id/btnAddPost"
                android:layout_width="30dp"
                android:layout_height="match_parent"
                android:layout_marginEnd="@dimen/layout_start_end"
```

```
        android:background="@drawable/agregar"
        android:onClick="btnAddPostClick" />

        <Button
            android:id="@+id/btnExit"
            android:layout_width="30dp"
            android:layout_height="match_parent"
            android:layout_marginEnd="@dimen/layout_start_end"
            android:background="@drawable/salir"
            android:onClick="btnExitClick" />
    </LinearLayout>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="@dimen/layout_start_end"
    android:layout_marginEnd="@dimen/layout_start_end"
    android:layout_marginBottom="@dimen/layout_bottom"
    android:orientation="vertical">

    <EditText
        android:id="@+id/edDescr"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="@string/description"
        android:inputType="textMultiLine" />

    <EditText
        android:id="@+id/edPlace"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="@string/place"
        android:inputType="textMultiLine" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <EditText
            android:id="@+id/edStartDate"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:hint="@string/between_date"
            android:inputType="date" />

        <EditText
            android:id="@+id/edEndDate"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:hint="@string/and_date"
            android:inputType="date" />
    </LinearLayout>

    <Button
        android:id="@+id/btnSearch"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```



```
        android:onClick="btnSearchClick"
        android:text="@string/search" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="@dimen/layout_start_end"
    android:layout_marginEnd="@dimen/layout_start_end"
    android:layout_marginBottom="@dimen/layout_bottom"
    android:orientation="vertical">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:id="@+id/llPosts"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical" />

    </ScrollView>
</LinearLayout>
</LinearLayout>
```

**MyPostsActivity.java**

```

/*-----
 *
 *          INSTITUTO TECNOLÓGICO DE LA LAGUNA
 *          INGENIERIA EN SISTEMAS COMPUTACIONALES
 *          PROGRAMACION MOVILES EN ANDROID "A"
 *
 *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
 *
 *          Actividad para ver tus publicaciones.
 *
 * Archivo      : MyPostsActivity.java
 * Autor        : Fernando Alfonso Caldera Olivas      15130685
 *              : Jesús Iván Sifuentes Valle          15131438
 * Fecha        : 13/Dic/2018
 * Compilador   : Android Studio 3.1.3
 * Descripción  : Actividad que sirve para ver las cosas que has publicado.
 *              : Desde esta actividad, puedes borrar tus publicaciones, lo cual
 *              : logra borrando primero las imágenes asociadas, si es que las hay,
 *              : y luego borra la publicación.
 * Ultima modif:
 * Fecha        Modificó          Motivo
 *=====
 *
 */-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp;

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

import java.util.List;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.lost_object.LostObject;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.lost_object.LostObjectDao;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.picture.Picture;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.picture.PictureDao;

public class MyPostsActivity extends AppCompatActivity {

    //Tabla para agregar publicaciones.
    private LinearLayout llPosts;

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

        //Obtenemos las referencias
        llPosts = findViewById(R.id.llPosts);

        //Buscamos todas las publicaciones del usuario (Cambiará en el futuro por una
        //implementación con paginación).
        try{
            List<LostObject> list =
            LostObjectDao.getInstance().getRecordsByUser(CommonData.getInstance().user);
            loadPosts(list);
        } catch (Exception e){
            Toast.makeText(this, e.getMessage(), Toast.LENGTH_LONG).show();
        }
    }
}

```

```

    }
}
//-----

//-----

private void loadPosts(List<LostObject> list){
    //Limpiamos la tabla.
    llPosts.removeAllViews();

    //Iteramos las publicaciones en la lista.
    for(final LostObject lo : list){
        //Creamos un linear layout que va a actuar como "row" de la tabla.
        //Contendrá 4 elementos.
        final LinearLayout ll = new LinearLayout(this);
        ll.setOrientation(LinearLayout.HORIZONTAL);
        ll.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT));
        ll.setWeightSum(4.0f);

        //Agregamos la fecha con que se registró la publicación.
        TextView tvwDate = new TextView(this);
        tvwDate.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT, 1));
        tvwDate.setText(lo.getFindDate().toString());

        //Agregamos la descripción.
        TextView tvwDescr = new TextView(this);
        tvwDescr.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT, 1));
        tvwDescr.setText(lo.getDescr());

        //Agregamos el lugar.
        TextView tvwPlace = new TextView(this);
        tvwPlace.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT, 1));
        tvwPlace.setText(lo.getPlace());

        //Agregamos el botón con la acción de eliminar publicación.
        //Lo encerramos dentro de un linear layout vertical, para
        //poder centrar el botón verticalmente y mandarlo al final
        //(orilla derecha).
        LinearLayout llButton = new LinearLayout(this);
        llButton.setOrientation(LinearLayout.VERTICAL);
        llButton.setGravity(Gravity.CENTER_VERTICAL | Gravity.END);
        llButton.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT, 1));
        Button btnDelete = new Button(this);
        btnDelete.setLayoutParams(new LinearLayout.LayoutParams(50,
        50));
        btnDelete.setBackground(getDrawable(R.drawable.tacha));
        btnDelete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try{
                    //Eliminamos todas las imágenes asociadas a la publicación.
                    Picture pic = new Picture();
                    pic.setLostObject(lo);
                    PictureDao.getInstance().delete(pic);
                    //Eliminamos la publicación en cuestión, la cual es la

```

```
//publicación de la iteración actual.
LostObjectDao.getInstance().delete(lo);
//Removemos el elemento actual (row actual) del layout,
//eliminandolo del padre.
llPosts.removeView(ll);
Toast.makeText(MyPostsActivity.this, "Publicación eliminada.",
Toast.LENGTH_SHORT).show();
    } catch (Exception e) {
        Toast.makeText(MyPostsActivity.this, "Ocurrió un error al eliminar la
publicación.", Toast.LENGTH_LONG).show();
    }
}
});
//Agregamos el botón a su layout contenedor.
llButton.addView(btnDelete);

//Agregamos todo lo demás al nuevo "row"
ll.addView(tvwDate);
ll.addView(tvwDescr);
ll.addView(tvwPlace);
ll.addView(llButton);

//Agregamos el row a la "tabla".
llPosts.addView(ll);
    }
}
//-----
-----
}
```

**activity\_my\_posts.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MyPostsActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:gravity="center"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView14"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/my_posts_screen"
            android:textSize="24sp" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:orientation="vertical">

        <ScrollView
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <LinearLayout
                android:id="@+id/llPosts"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical" />
            </ScrollView>
        </LinearLayout>
    </LinearLayout>
```

**SignupActivity.java**

```

/*-----
 *
 *          INSTITUTO TECNOLÓGICO DE LA LAGUNA
 *          INGENIERIA EN SISTEMAS COMPUTACIONALES
 *          PROGRAMACION MOVILES EN ANDROID "A"
 *
 *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
 *
 *          Actividad para registrar una cuenta.
 *
 * Archivo      : SignupActivity.java
 * Autor        : Fernando Alfonso Caldera Olivas      15130685
 *              : Jesús Iván Sifuentes Valle          15131438
 * Fecha        : 13/Dic/2018
 * Compilador   : Android Studio 3.1.3
 * Descripción  : Actividad que sirve para registrar una cuenta nueva.
 *              : Para registrar una cuenta nueva, tienes que agregar un nombre
 *              : de usuario que no exista, una contraseña y un número de telefono.
 *
 * Ultima modif:
 * Fecha        Modificó          Motivo
 *-----
 */-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp;

import android.content.DialogInterface;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.user.User;
import mx.edu.itl2018b.a15130685.u5perdidosencontradosapp.user.UserDao;

public class SignupActivity extends AppCompatActivity {

    //Campos para registrarse
    private EditText edUser;
    private EditText edPass;
    private EditText edPhone;

    //-----

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

        //Obtenemos las referencias
        edUser = findViewById(R.id.edUser);
        edPass = findViewById(R.id.edPass);
        edPhone = findViewById(R.id.edPhone);

    }
    //-----

    //-----

    public void btnSignupClick(View view){
        //Verificamos que los campos no estén vacíos
        boolean fieldsOk = true;

```

```

if(edUser.getText() == null || edUser.getText().toString().trim().equals("")){
    edUser.setError("Especifique el usuario.");
    fieldsOk = false;
}
if(edPass.getText() == null || edPass.getText().toString().trim().equals("")){
    edPass.setError("Especifique la contraseña.");
    fieldsOk = false;
}

//Si hay un error en los campos, termina
if(!fieldsOk){
    return;
}

//Guardamos los datos del layout en un objeto usuario.
User u = new User();
u.setUserName(edUser.getText().toString());
u.setPass(edPass.getText().toString());
u.setPhoneNumber(edPhone.getText().toString());

//Verificamos que el nombre de usuario este disponible
User udb = new User();
udb.setUserName(u.getUserName());

try{
    udb = UserDao.getInstance().getRecord(udb);
} catch (Exception e){
    Toast.makeText(this, "Ocurrió un error de conexión.", Toast.LENGTH_LONG).show();
}

//Si el usuario existe, mandamos error
if(udb.getId() != null){
    edUser.setError("El nombre de usuario ya existe.");
}
//si no existe, lo insertamos.
else{
    try{
        UserDao.getInstance().insert(u);
        //Tod0 bien, notificamos al usuario y concluimos el activity
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Usuario registrado.")
            .setMessage("El usuario se ha registrado con éxito!")
            .setCancelable(false)
            .setPositiveButton("Aceptar", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    dialog.dismiss();
                    SignupActivity.this.finish();
                }
            }).create().show();
    } catch (Exception e){
        Toast.makeText(this, "Ocurrió un error al registrar al usuario.",
Toast.LENGTH_LONG).show();
    }
}
}
//-----
}

```

**activiy\_signup.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    tools:context=".SignupActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:gravity="center"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView9"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/signup_screen"
            android:textSize="24sp" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="@dimen/layout_start_end"
        android:layout_marginEnd="@dimen/layout_start_end"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:gravity="center"
        android:orientation="vertical">

        <EditText
            android:id="@+id/edUser"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/layout_bottom"
            android:ems="10"
            android:hint="@string/usr"
            android:inputType="textPersonName" />

        <EditText
            android:id="@+id/edPass"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/layout_bottom"
            android:ems="10"
            android:hint="@string/pass"
            android:inputType="textPassword" />

        <EditText
            android:id="@+id/edPhone"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="@dimen/layout_bottom"
            android:ems="10"
            android:hint="@string/phone"
            android:inputType="phone" />

        <Button
```



```
        android:id="@+id/btnSignup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="@dimen/layout_bottom"
        android:onClick="btnSignupClick"
        android:text="@string/signup" />
    </LinearLayout>
</LinearLayout>
```

**SplashActivity.java**

```

/*-----
 *
 *          INSTITUTO TECNOLÓGICO DE LA LAGUNA
 *          INGENIERIA EN SISTEMAS COMPUTACIONALES
 *          PROGRAMACION MOVILES EN ANDROID "A"
 *
 *          SEMESTRE: AGO-DIC/2018      HORA: 10-11 HRS
 *
 *          Actividad de splash
 *
 * Archivo      : SplashActivity.java
 * Autor        : Fernando Alfonso Caldera Olivas      15130685
 *              : Jesús Iván Sifuentes Valle          15131438
 * Fecha        : 13/Dic/2018
 * Compilador   : Android Studio 3.1.3
 * Descripción  : Actividad que sirve de splash, para introducir al usuario a la
 *              : pantalla de login.
 *
 *
 *
 * Última modif:
 * Fecha      Modificó      Motivo
 *=====
 *
 *-----*/

package mx.edu.itl2018b.a15130685.u5perdidosencontradosapp;

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

public class SplashActivity extends AppCompatActivity {

    //Duración del splash.
    private static final int SPLASH_DURATION = 2000;

    //-----
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);
        //Iniciamos un nuevo manejador, que va a iniciar la actividad de login y cerrar
        //este splash después de la duración especificada en la constante.
        new Handler().postDelayed(
            new Runnable() {
                @Override
                public void run() {
                    Intent intent = new Intent(SplashActivity.this, LoginActivity.class);
                    startActivity(intent);
                    finish();
                }
            }, SPLASH_DURATION
        );
    }
    //-----
}

```

**activity\_splash.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".SplashActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:srcCompat="@drawable/perdidos_encontrados_logo" />
</LinearLayout>
```

**dimens.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <dimen name="layout_start_end">15sp</dimen>
  <dimen name="layout_bottom">10sp</dimen>
</resources>
```

**strings.xml (Default)**

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Perdidos y Encontrados</string>
    <string name="usr">Usuario:</string>
    <string name="pass">Contraseña:</string>
    <string name="phone">Teléfono:</string>
    <string name="login">Acceder</string>
    <string name="signup">Registrarse</string>
    <string name="signup_ad">¿No tiene cuenta?</string>
    <string name="my_posts">Mis publicaciones</string>
    <string name="add_post">Agregar publicación</string>
    <string name="exit">Salir</string>
    <string name="between_date">Entre fecha...</string>
    <string name="and_date">Y fecha...</string>
    <string name="between_time">Entre hora...</string>
    <string name="and_time">Y hora...</string>
    <string name="description">Descripción:</string>
    <string name="place">Lugar:</string>
    <string name="search">Buscar</string>
    <string name="date">Fecha:</string>
    <string name="time">Tiempo:</string>
    <string name="add_photo">Añadir fotografía</string>
    <string name="publish">Publicar</string>
    <string name="about_head">
        TECNOLOGICO NACIONAL DE MEXICO\n
        Instituto Tecnológico de la Laguna\n
        Ingenieria en Sistemas Computacionales\n\n

        PROGRAMACION DE DISPOSITIVOS MOVILES CON ANDROID\n\n
    </string>
    <string name="about_body">
        SEMESTRE: Ago - Dic / 2018\t\t\tGRUPO: "A" 10 - 11 Hrs\n\n

        PRACTICA No.    U5P1\n

        Proyecto Final: App Perdidos y Encontrados\n\n

        ALUMNOS:\n\n

        15130685    Fernando Alfonso Caldera Olivas\n\n
        15131438    Jesús Iván Sifuentes Valle\n\n
    </string>
    <string name="about_foot">
        PROFESOR:\n
        Ing. Luis Fernando Gil Vázquez\n\n\n
        Torreón, Coah. a Miércoles 12 de Diciembre de 2018\n
    </string>
    <string name="about">Acerca de</string>
    <string name="login_screen">Acceso</string>
    <string name="signup_screen">Registro</string>
    <string name="main_screen">Menú Principal</string>
    <string name="add_post_screen">Agregar Publicación</string>
    <string name="inspect_post_screen">Inspección</string>

```

```
<string name="my_posts_screen">Mis Publicaciones</string>  
</resources>
```

**strings.xml (es-MX)**

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Perdidos y Encontrados</string>
    <string name="usr">Usuario:</string>
    <string name="pass">Contraseña:</string>
    <string name="phone">Teléfono:</string>
    <string name="login">Acceder</string>
    <string name="signup">Registrarse</string>
    <string name="signup_ad">¿No tiene cuenta?</string>
    <string name="my_posts">Mis publicaciones</string>
    <string name="add_post">Agregar publicación</string>
    <string name="exit">Salir</string>
    <string name="between_date">Entre fecha...</string>
    <string name="and_date">Y fecha...</string>
    <string name="between_time">Entre hora...</string>
    <string name="and_time">Y hora...</string>
    <string name="description">Descripción:</string>
    <string name="place">Lugar:</string>
    <string name="search">Buscar</string>
    <string name="date">Fecha:</string>
    <string name="time">Tiempo:</string>
    <string name="add_photo">Añadir fotografía</string>
    <string name="publish">Publicar</string>
    <string name="about_head">
        TECNOLÓGICO NACIONAL DE MÉXICO\n
        Instituto Tecnológico de la Laguna\n
        Ingeniería en Sistemas Computacionales\n\n

        PROGRAMACIÓN DE DISPOSITIVOS MÓVILES CON ANDROID\n\n
    </string>
    <string name="about_body">
        SEMESTRE: Ago - Dic / 2018\t\t\tGRUPO: "A" 10 - 11 Hrs\n\n

        PRACTICA No.    U5P1\n

        Proyecto Final: App Perdidos y Encontrados\n\n

        ALUMNOS:\n\n

        15130685    Fernando Alfonso Caldera Olivas\n\n
        15131438    Jesús Iván Sifuentes Valle\n\n
    </string>
    <string name="about_foot">
        PROFESOR:\n
        Ing. Luis Fernando Gil Vázquez\n\n\n
        Torreón, Coah. a Miércoles 12 de Diciembre de 2018\n
    </string>
    <string name="about">Acerca de</string>
    <string name="login_screen">Acceso</string>
    <string name="signup_screen">Registro</string>
    <string name="main_screen">Menú Principal</string>
    <string name="add_post_screen">Agregar Publicación</string>
    <string name="inspect_post_screen">Inspección</string>

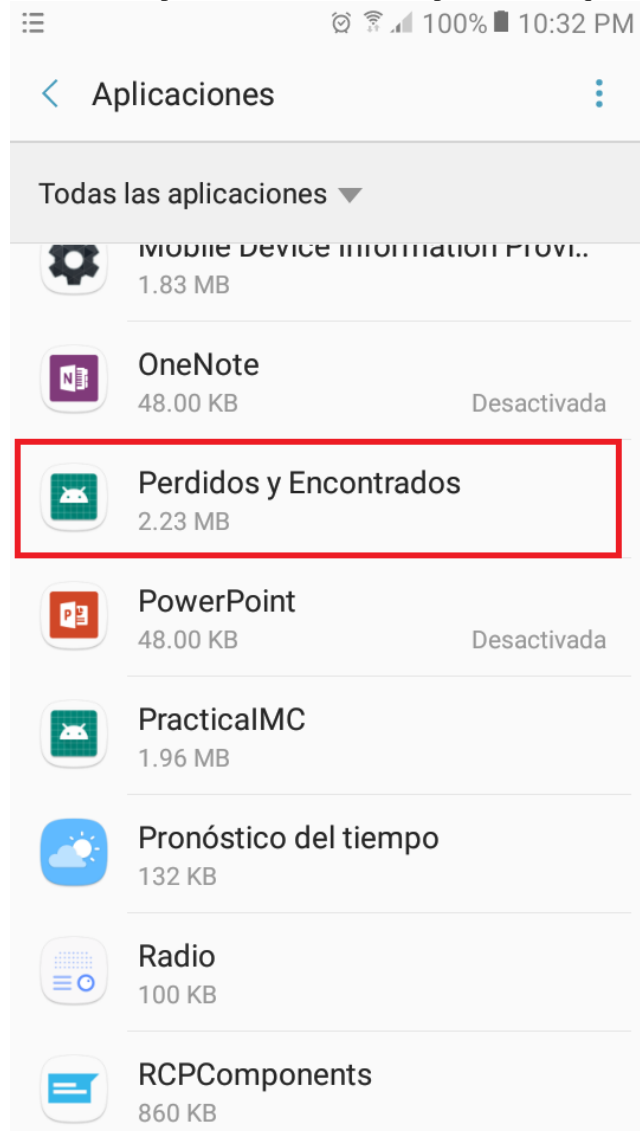
```

```
<string name="my_posts_screen">Mis Publicaciones</string>  
</resources>
```

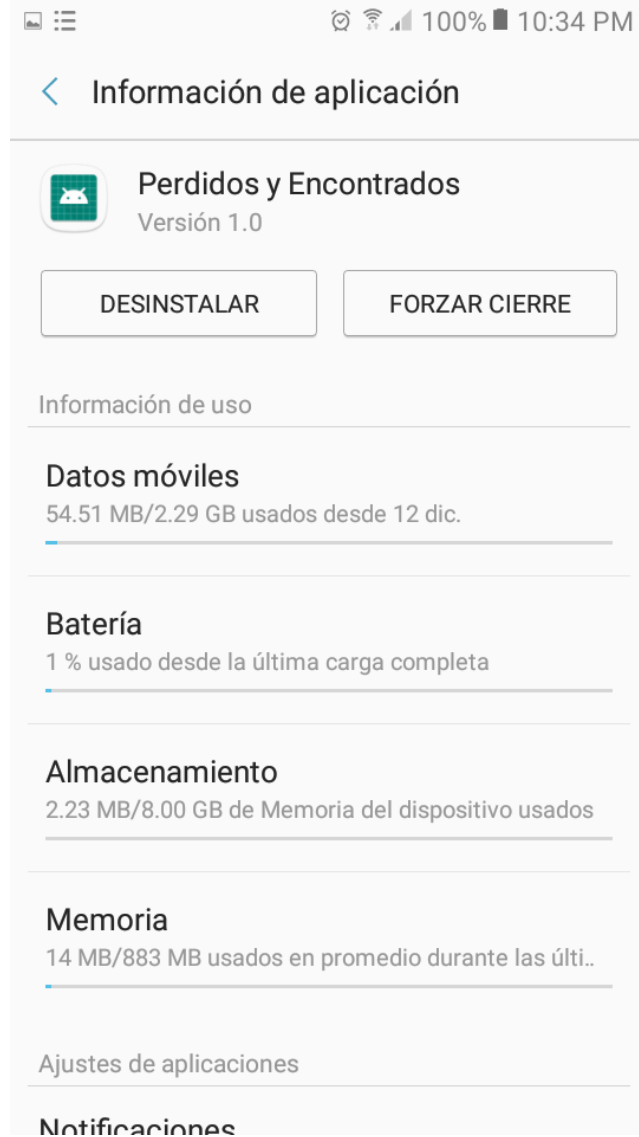


## Prueba de Ejecución

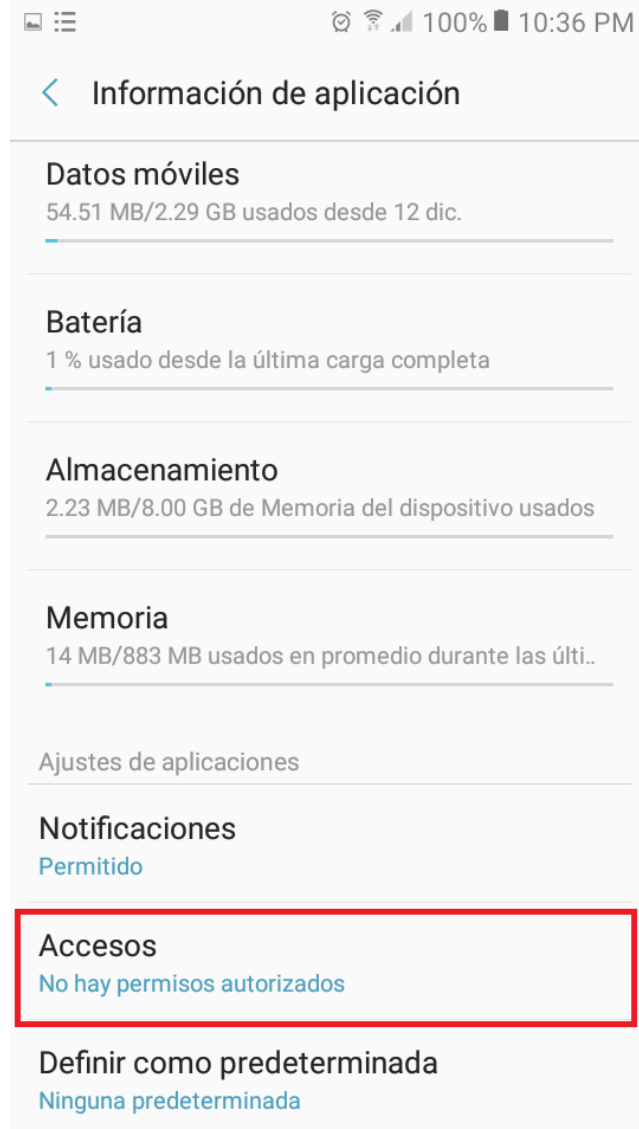
Para poder usar la aplicación **“Perdidos y Encontrados”**, necesita tener una conexión a internet ya sea estando conectado a una red Wi-Fi o usando datos móviles, y darle permiso a la aplicación para acceder al almacenamiento externo y para hacer llamadas. Para darle los permisos necesarios a la aplicación, busque la aplicación en su lista de aplicaciones y presione sobre ella.



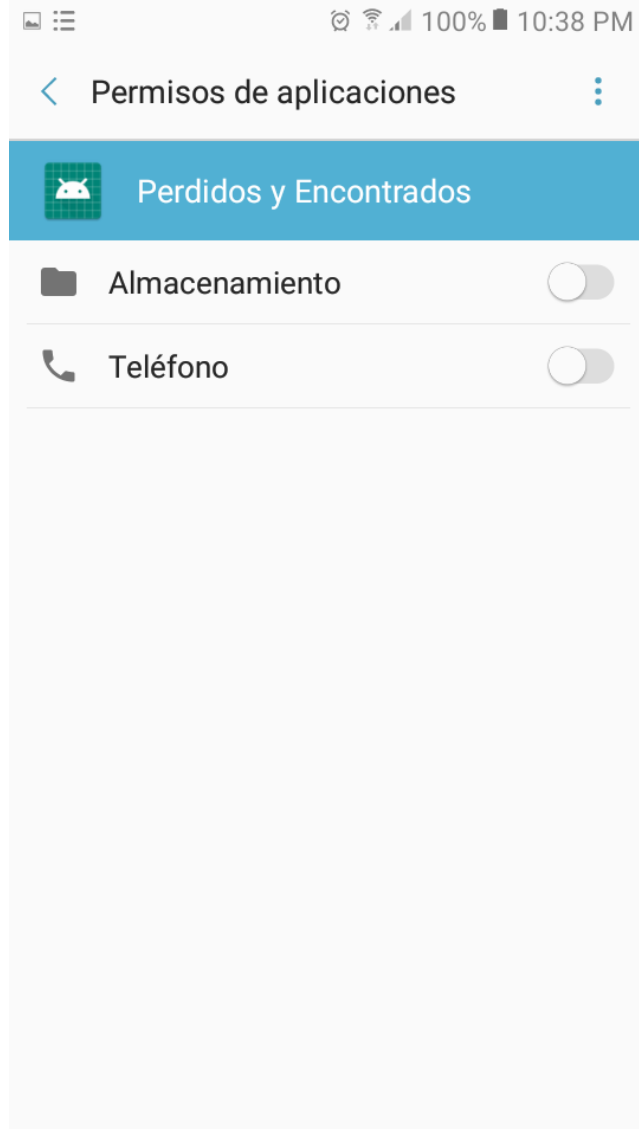
Aparecerá una pantalla similar a la siguiente, varía de dispositivo a dispositivo.



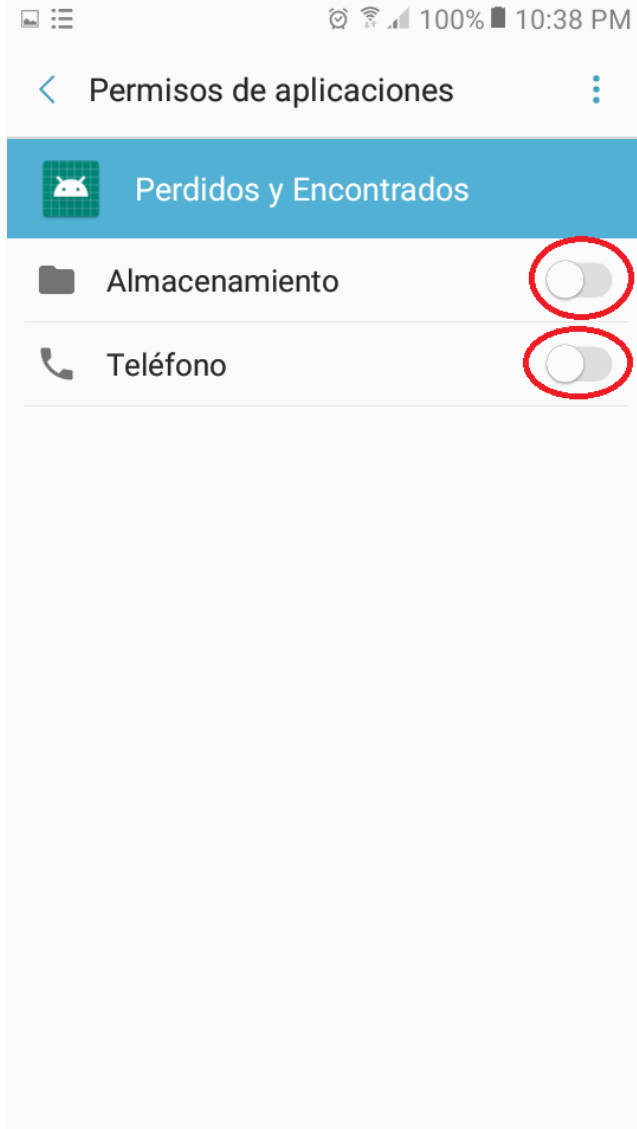
Ahora, deslice hacia abajo, hasta encontrar la opción **Accesos**. Presione la opción **Accesos**.



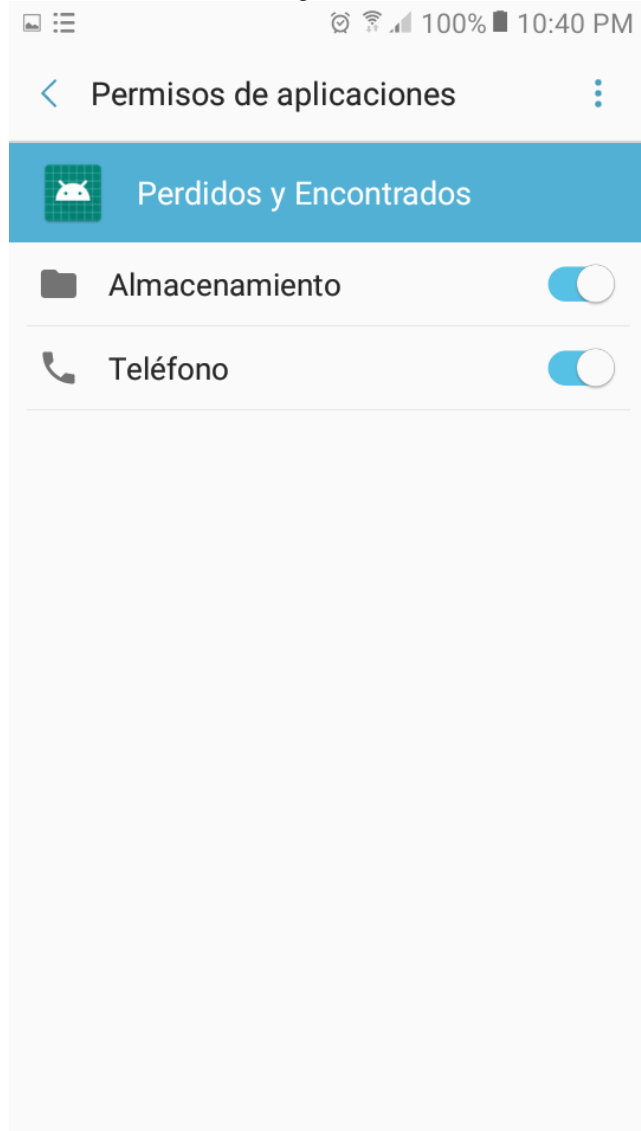
Se mostrará una pantalla similar a la siguiente, de nuevo, varía de dispositivo a dispositivo.



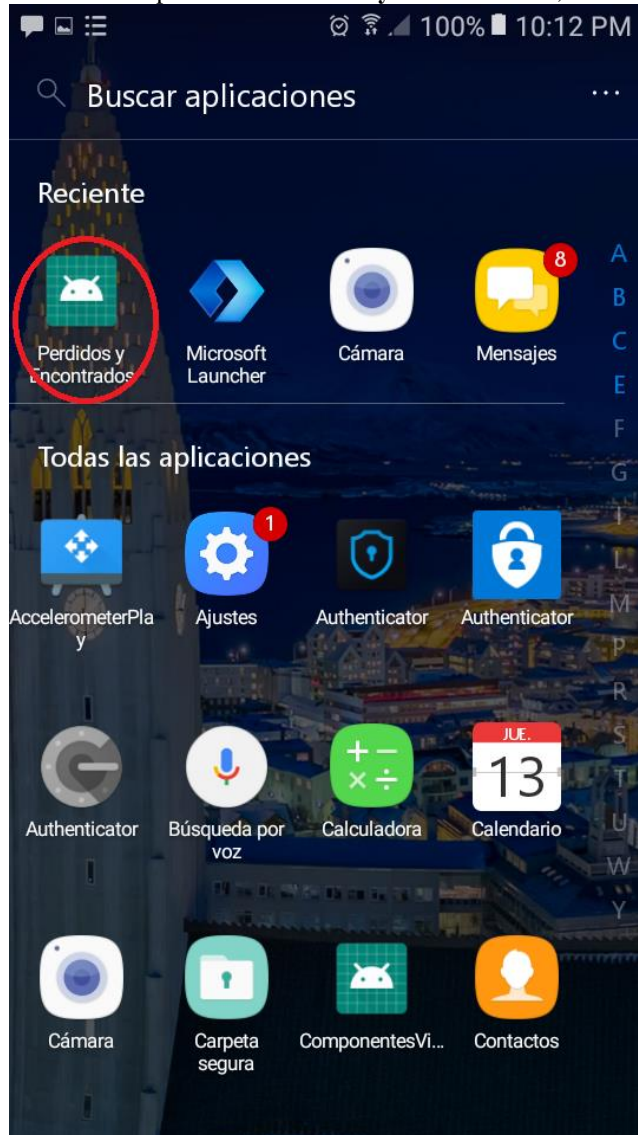
Active los los permisos presionando donde se indica en la siguiente imagen.



Ahora se mostrarán de la siguiente manera.



Para abrir la aplicación **“Perdidos y Encontrados”**, debemos hacer click en el ícono de la aplicación.



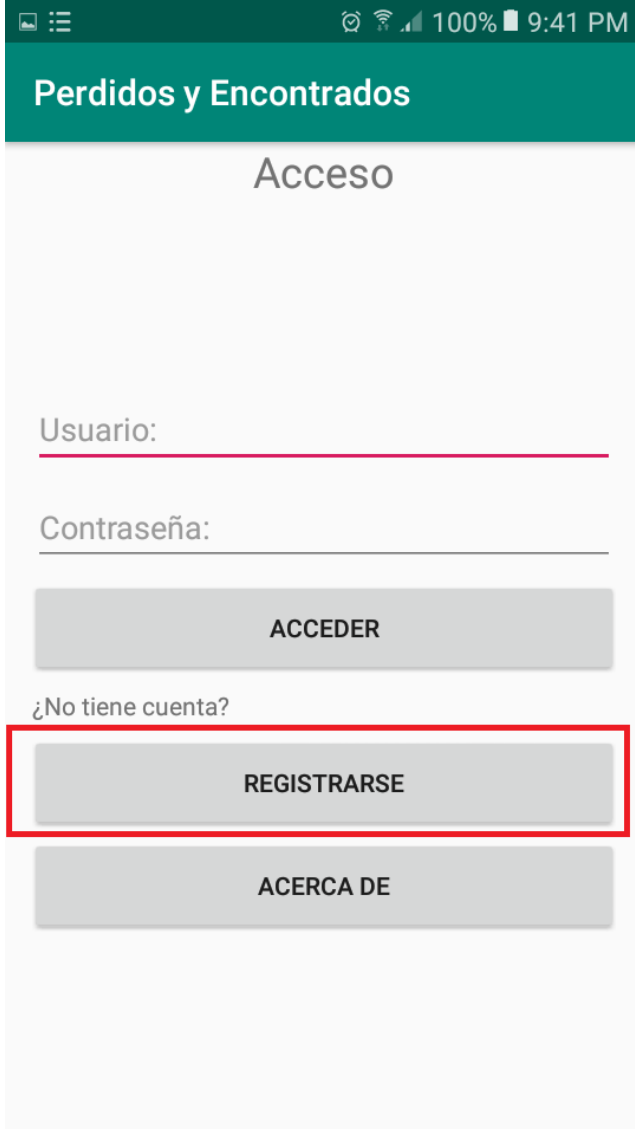
Inmediatamente después de abrir la aplicación, hay que esperar a que cargue. Al abrirla se mostrará el logo de la app.





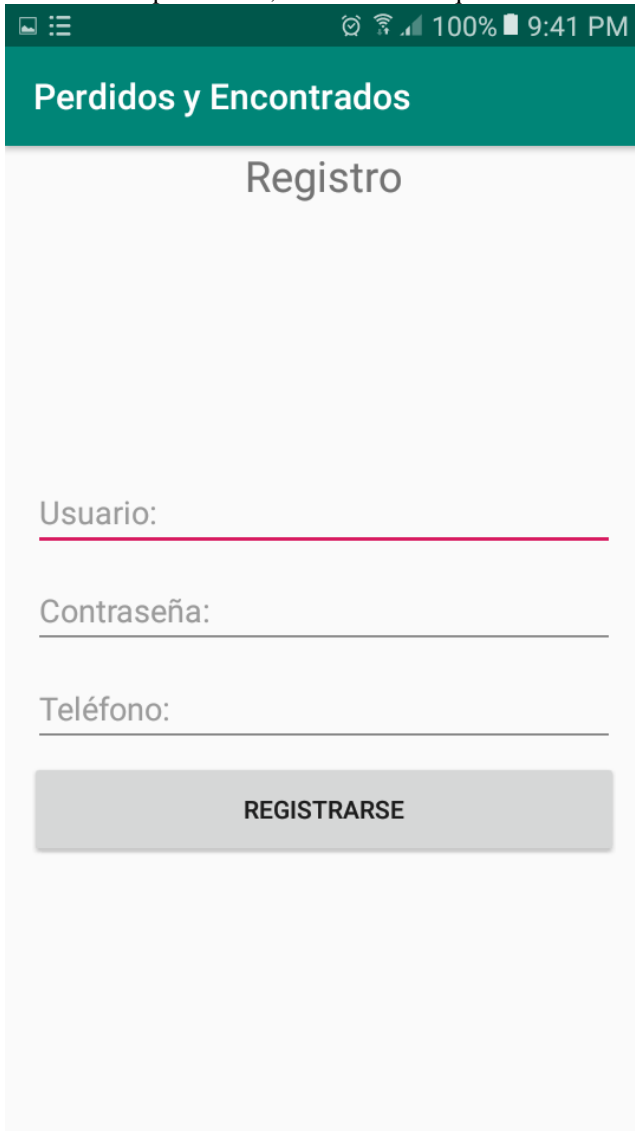
Cuando termine de cargar, se mostrará una pantalla de inicio de sesión, donde el usuario podrá iniciar sesión con su cuenta, si es que tiene una.

Supongamos que no tiene cuenta, hay que presionar el botón que dice **Registrarse**.



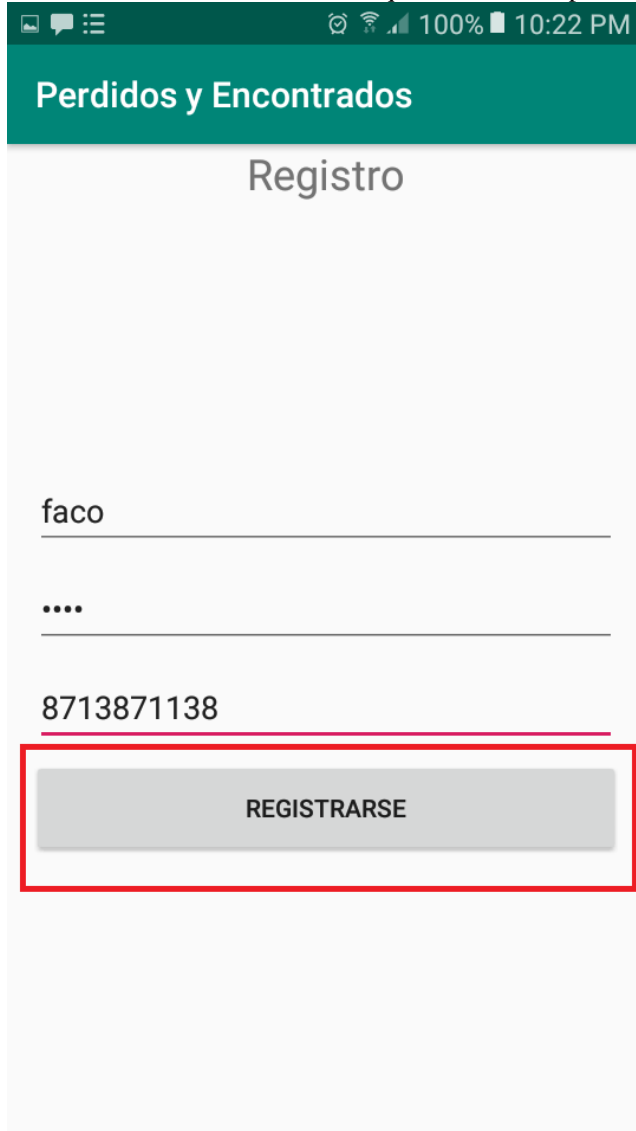
The screenshot shows a mobile application interface. At the top, there is a dark green header with the text "Perdidos y Encontrados" in white. Below the header, the word "Acceso" is centered in a large, light gray font. There are two input fields: "Usuario:" and "Contraseña:", each followed by a horizontal line for text entry. Below these fields is a gray button labeled "ACCEDER". Underneath the button is the text "¿No tiene cuenta?". Below this text is a gray button labeled "REGISTRARSE", which is highlighted with a red rectangular border. At the bottom, there is another gray button labeled "ACERCA DE". The status bar at the very top shows icons for signal, Wi-Fi, and battery, along with the text "100% 9:41 PM".

Al darle click en registrarse, se mostrará una pantalla para registrar una nueva cuenta de usuario. Aquí, debe elegir un nombre de usuario de su preferencia, una contraseña que solo usted conozca y un número de teléfono de contacto.



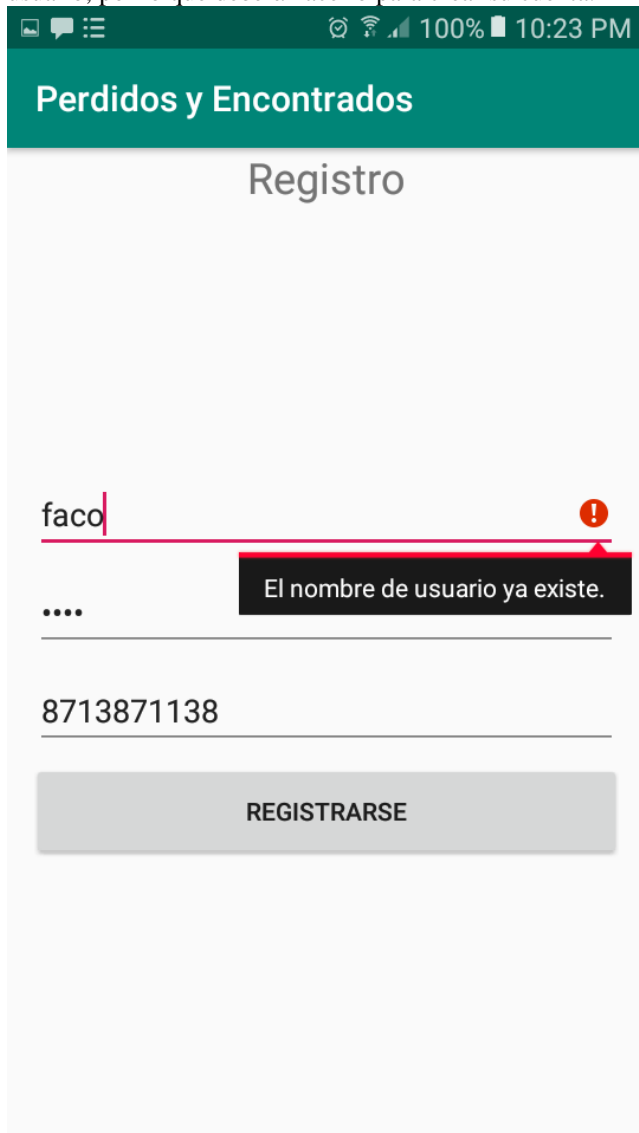
The screenshot shows an Android application interface. At the top, there is a dark green header bar with the text "Perdidos y Encontrados" in white. Below the header, the word "Registro" is centered in a large, dark font. The main area of the screen is light gray and contains three input fields, each with a label to its left: "Usuario:", "Contraseña:", and "Teléfono:". Each label is followed by a horizontal line representing the input field. Below these fields is a gray rectangular button with the text "REGISTRARSE" in all caps.

Cuando termine de llenar los datos que se le solicita, presione en el botón **Registrarse**.



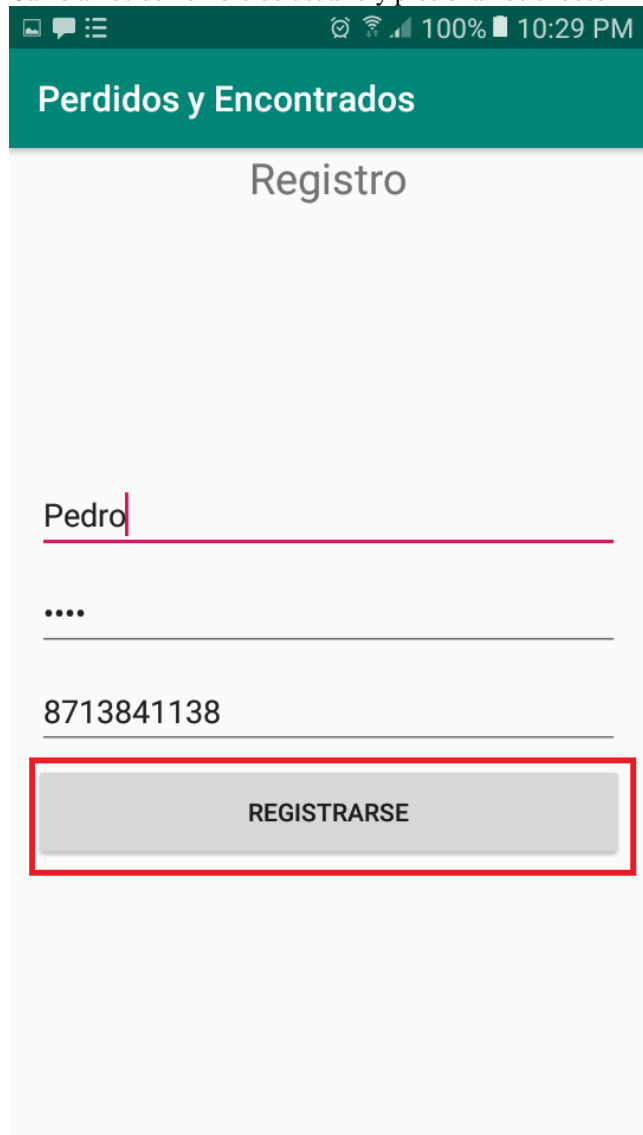
The screenshot shows an Android application interface. At the top, there is a green header bar with the text "Perdidos y Encontrados". Below this, the word "Registro" is centered. The form consists of three input fields: the first contains "faco", the second contains four dots "....", and the third contains the phone number "8713871138". Below the third field is a grey button with the text "REGISTRARSE" in all caps. This button is highlighted with a red rectangular border.

Si el nombre de usuario ya está siendo ocupado por alguien más, le aparecerá un mensaje diciéndole que cambie de nombre de usuario, por lo que deberá hacerlo para crear su cuenta.



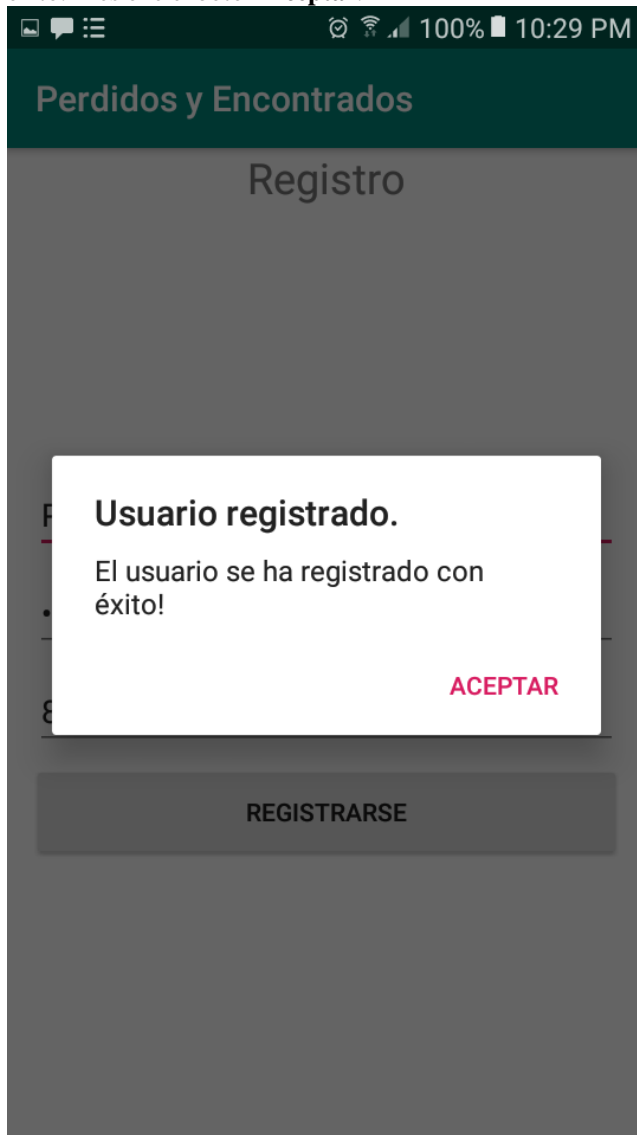
The screenshot shows an Android application interface. At the top is a dark green header bar with the text "Perdidos y Encontrados" in white. Below the header is a light gray area with the title "Registro" in a large, dark font. The registration form consists of two input fields. The first field contains the text "faco" and has a red error message box below it that says "El nombre de usuario ya existe." with a red exclamation mark icon. The second field contains the text "8713871138". Below the input fields is a gray button with the text "REGISTRARSE" in all caps. The status bar at the top of the screen shows various icons including a camera, speech bubble, menu, alarm, Wi-Fi, signal strength, 100% battery, and the time 10:23 PM.

Cambiamos de nombre de usuario y presionamos el botón **Registrarse**.

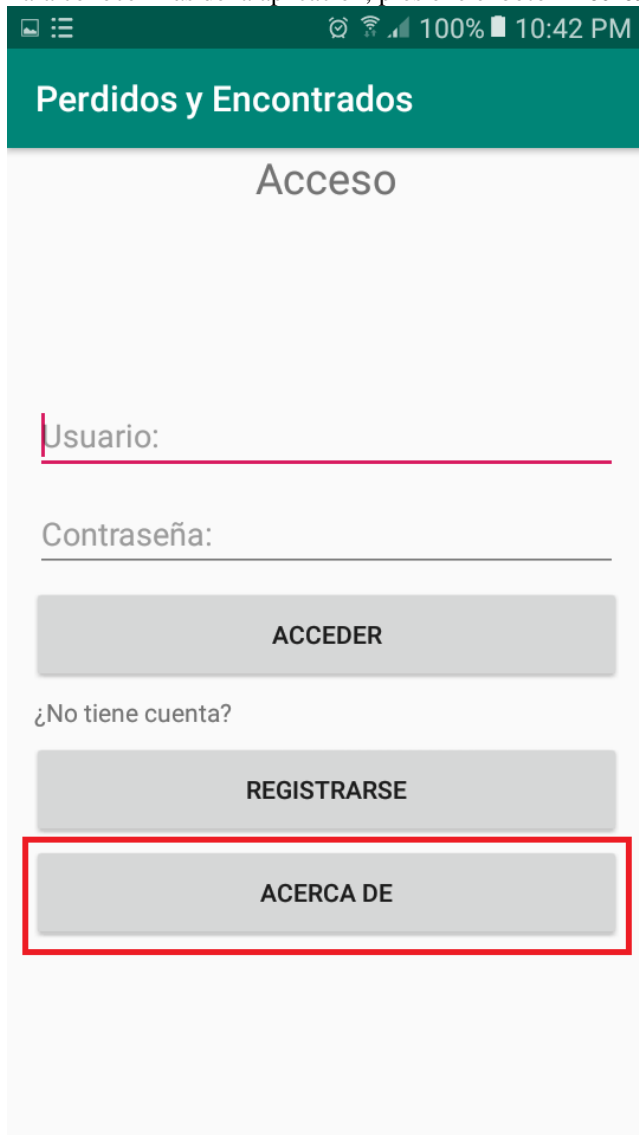


The screenshot shows an Android application interface. At the top, there is a dark green header bar with the text "Perdidos y Encontrados" in white. Below the header, the word "Registro" is centered in a large, gray font. The main area of the screen is light gray and contains three input fields. The first field has the text "Pedro" and a red vertical line at the end. The second field contains four dots "....". The third field contains the number "8713841138". Below these fields is a gray button with the text "REGISTRARSE" in black. The button is highlighted with a red rectangular border.

Si ha elegido un nombre de usuario que esté disponible, le mostrará un dialogo diciéndole que su usuario ha sido registrado con éxito. Presione el botón **Aceptar**.



La aplicación regresará automáticamente a la pantalla de inicio de sesión.  
Para conocer más de la aplicación, presione el botón **Acerca de**.



Perdidos y Encontrados

Acceso

Usuario:

Contraseña:

ACCEDER

¿No tiene cuenta?

REGISTRARSE

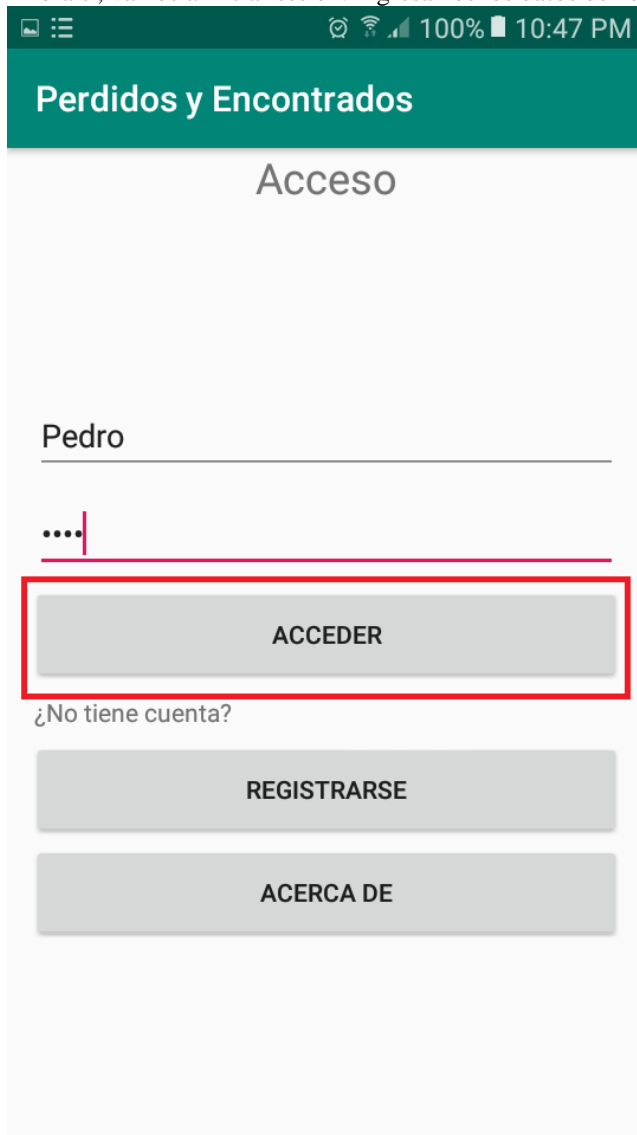
ACERCA DE

Se mostrará la siguiente pantalla, con información de la aplicación.





Para regresar a la pantalla de inicio de sesión, simplemente presione el botón designado en su celular para ir hacia atrás. Ahora sí, vamos a iniciar sesión. Ingresamos los datos de nuestra cuenta y presionamos el botón **Acceder**.



The screenshot shows a mobile application interface. At the top, there is a green header bar with the text "Perdidos y Encontrados". Below this, the word "Acceso" is centered. There are two input fields: the first contains the name "Pedro", and the second contains four dots, indicating a password field. A red rectangular box highlights the "ACCEDER" button located below the password field. Below the "ACCEDER" button, there is a link that says "¿No tiene cuenta?". At the bottom, there are two more buttons: "REGISTRARSE" and "ACERCA DE". The status bar at the very top shows icons for signal, Wi-Fi, and battery, along with the time "10:47 PM" and "100%" battery level.

Una vez dentro, nos aparece inmediatamente una pantalla de menú principal. Desde aquí podemos hacer muchas cosas, como revisar las publicaciones de otros usuarios, filtrar las publicaciones por diferentes parámetros, agregar una publicación nosotros mismos, revisar las publicaciones que hemos hecho y salir de la sesión.



The screenshot shows the main menu of the 'Perdidos y Encontrados' app. At the top is a teal header with the title 'Perdidos y Encontrados'. Below it is a section titled 'Menú Principal' for a user named 'Pedro'. This section contains three icons: a list icon, a green plus icon, and a share icon. There are three input fields: 'Descripción:', 'Lugar:', and a date range selector 'Entre fecha...' and 'Y fecha...'. A grey 'BUSCAR' button is positioned below the date fields. At the bottom is a list of six items, each with a date, a title, and a location. To the right of this list is a vertical stack of six magnifying glass icons.

Fecha	Titulo	Ubicacion
2018-12-25	mi dignidad	con ella
2018-12-25	mi dignidad	en su corazon
2018-12-25	mi dignidad	en su corazon
2018-12-13	Laptop HP Roja	ITL 19M
2018-12-13	Bicho raro	En ITL LECA
2018-12-12	Gorro rojo	Centro de Torreon

A continuación se explica la pantalla anterior.

1. Botón para ver tus publicaciones: Muestra las publicaciones que haz hecho y te da la posibilidad de borrarlas.
2. Botón para agregar publicación: Te permite agregar una publicación, especificando los datos y te da la posibilidad de adjuntar tantas imágenes quieras a la publicación.
3. Botón salir sesión: De saca de la sesión. Te devuelve a la pantalla de inicio de sesión.
4. Controles de búsqueda: Estos controles te permiten filtrar las publicaciones en base en la descripción, lugar o un rango de fechas.
5. Últimas publicaciones: Aquí puedes ver cuales son las últimas publicaciones que se han hecho, en base a las fechas en que se encontraron los “objetos sin dueño”.

Perdidos y Encontrados

Menú Principal

Pedro

1 2 3

4 Descripción:

Lugar:

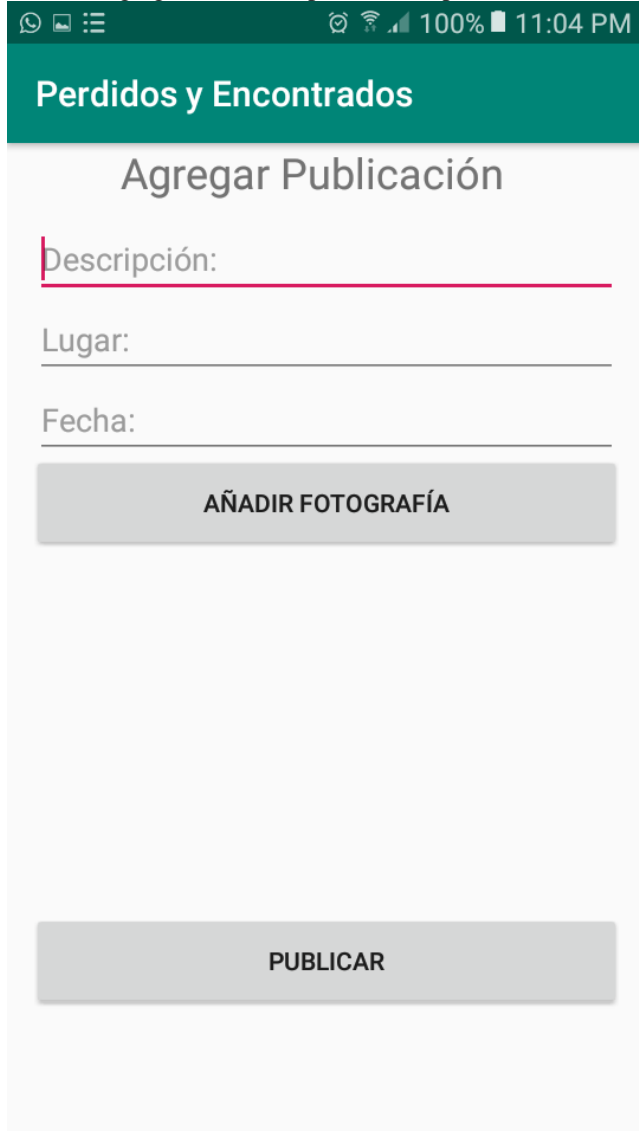
Entre fecha... Y fecha...

BUSCAR

5

2018-12-25	mi dignidad	con ella
2018-12-25	mi dignidad	en su corazon
2018-12-25	mi dignidad	en su corazon
2018-12-13	Laptop HP Roja	ITL 19M
2018-12-13	Bicho raro	En ITL LECA
2018-12-12	Gorro rojo	Centro de Torreon

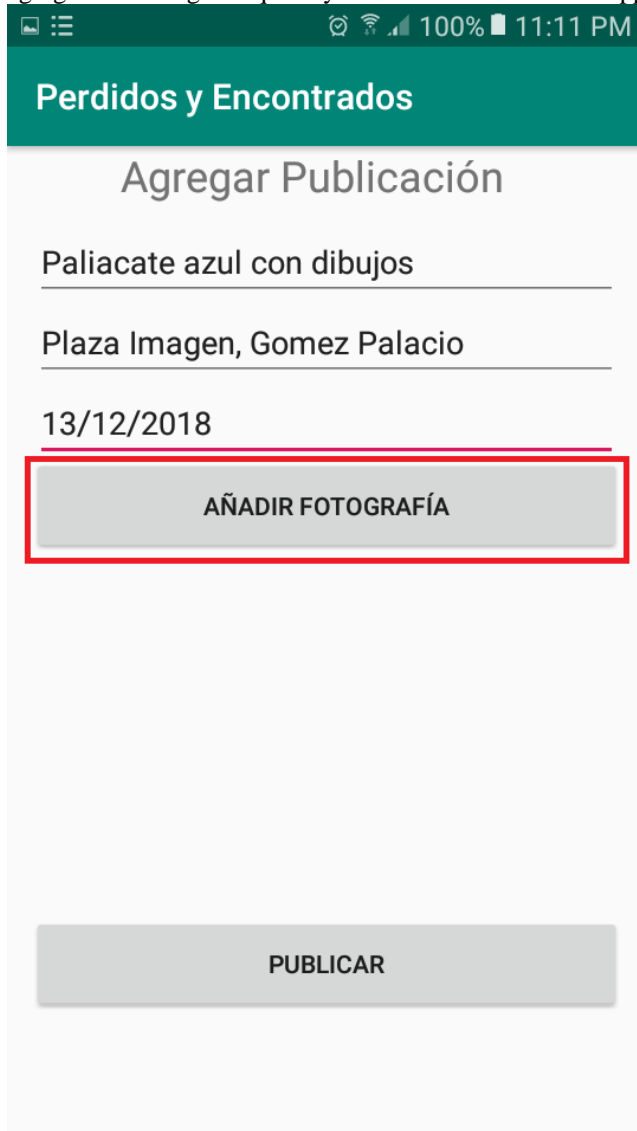
Vamos a agregar una nueva publicación, para eso, vamos a presionar el botón correspondiente, y nos mostrará esta pantalla.



The screenshot shows a mobile application interface with a teal header bar containing the text "Perdidos y Encontrados". Below the header, the title "Agregar Publicación" is centered. The form consists of three text input fields labeled "Descripción:", "Lugar:", and "Fecha:". Below these fields is a grey button labeled "AÑADIR FOTOGRAFÍA". At the bottom of the form is another grey button labeled "PUBLICAR". The status bar at the top of the phone shows the time as 11:04 PM and 100% battery.

Aquí podemos especificar los datos que nosotros queramos.

Digamos que queremos agregar un paliacate extraviado. Llenamos los campos correspondientes. Una vez hecho esto, podemos agregar unas imágenes que hayan sido tomadas con otra app. Para hacer esto, presionamos el botón **Añadir Fotografía**.



Perdidos y Encontrados

Agregar Publicación

Paliacate azul con dibujos

Plaza Imagen, Gomez Palacio

13/12/2018

AÑADIR FOTOGRAFÍA

PUBLICAR

Seleccionamos las fotografías que querramos añadir a nuestra publicación y, cuando todo este listo, presionamos el botón **Publicar**.

Perdidos y Encontrados

Agregar Publicación

Paliacate azul con dibujos

Plaza Imagen, Gomez Palacio

13/12/2018

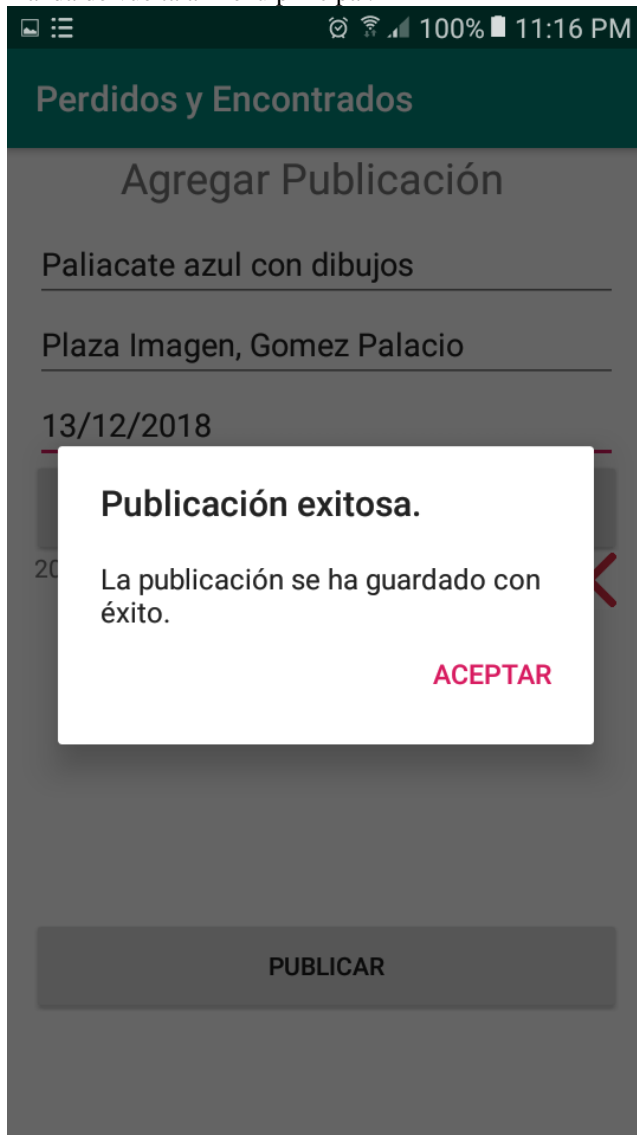
AÑADIR FOTOGRAFÍA

20181213\_231309.jpg



PUBLICAR





OJO: Si nos equivocamos de fotografía, o simplemente queremos quitar alguna fotografía, podemos presionar el botón con la tacha “X” para remover la foto.

Al presionar el botón, nos saldrá el siguiente mensaje, si la publicación fue exitosa. Presionamos en donde dice **Aceptar**, y nos manda de vuelta al menú principal.



Podemos ver que nuestra publicación ahora se ha agregado a las últimas publicaciones, en base a la fecha que establecimos al momento de publicarla.


 

   100%  11:18 PM

## Perdidos y Encontrados

### Menú Principal

Pedro





Descripción:

Lugar:

Entre fecha...

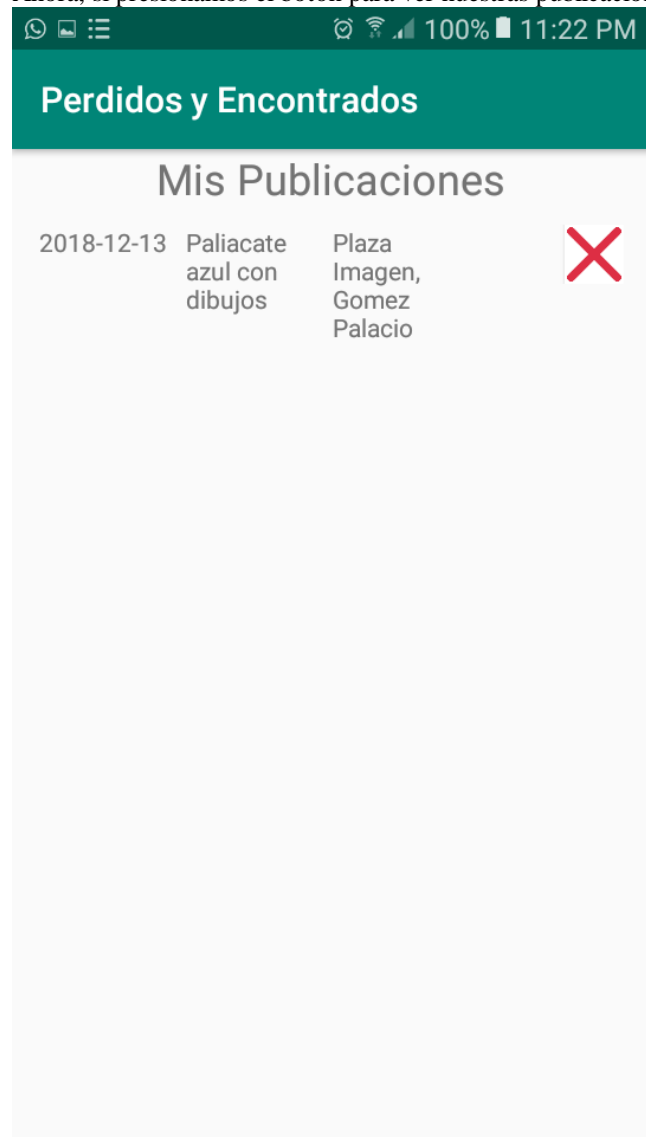
Y fecha...

BUSCAR

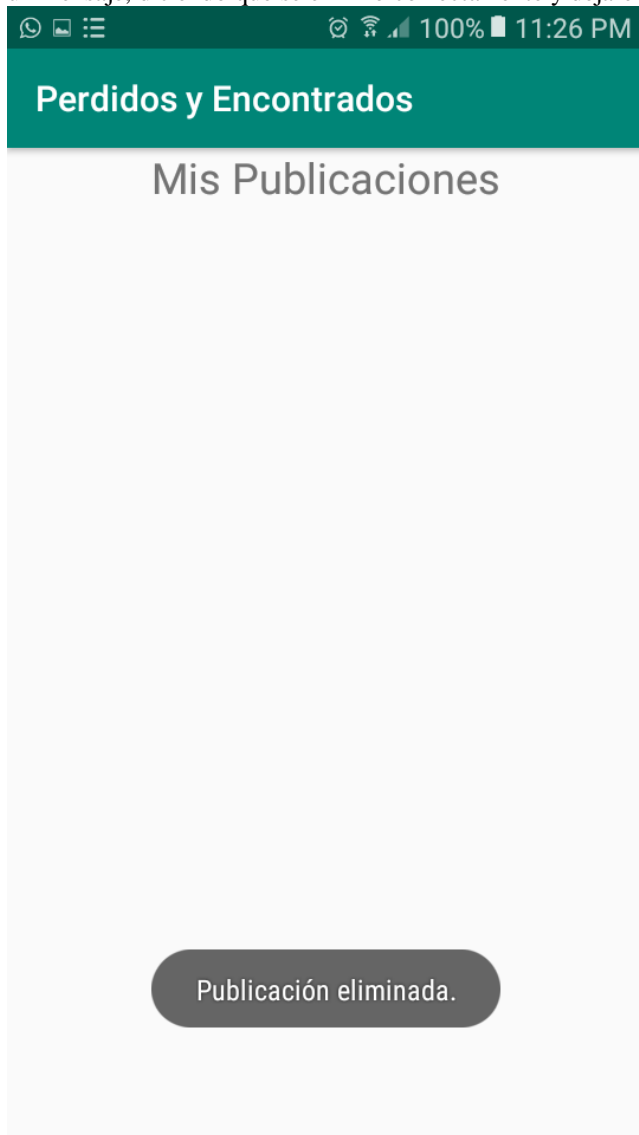
2018-12-25	mi dignidad	con ella	
2018-12-25	mi dignidad	en su corazon	
2018-12-25	mi dignidad	en su corazon	
2018-12-13	Laptop HP Roja	ITL 19M	
2018-12-13	Paliacate azul con dibujos	Plaza Imagen, Gomez Palacio	
2018-12-13	Bicho raro	En ITL LECA	
2018-12-12	Gorro rojo	Centro de Torreon	



Ahora, si presionamos el botón para ver nuestras publicaciones, vamos a ver que aparece solo una (la que acabamos de hacer).



Si presionamos el botón de la tachita, la publicación se borrará. Vamos a hacerlo. Presionamos el botón de la tachita y nos aparecerá un mensaje, diciendo que se eliminó correctamente y dejaremos de ver la publicación en ese preciso momento.



Si nos regresamos al menú principal (presionado el botón de regresar designado por el celular), veremos que nuestra publicación ya no está.



**Perdidos y Encontrados**

**Menú Principal**

Pedro   

Descripción: \_\_\_\_\_

Lugar: \_\_\_\_\_

Entre fecha... Y fecha... \_\_\_\_\_

**BUSCAR**

2018-12-25	mi dignidad	con ella	
2018-12-25	mi dignidad	en su corazon	
2018-12-25	mi dignidad	en su corazon	
2018-12-13	Laptop HP Roja	ITL 19M	
2018-12-13	Bicho raro	En ITL LECA	
2018-12-12	Gorro rojo	Centro de Torreon	

Podemos usar los controles de búsqueda para filtrar las publicaciones que se muestran.

Por ejemplo, si nosotros hubiésemos perdido nuestra laptop, pero no sabemos donde pudiera estar ni sabemos cuando la perdimos, podemos simplemente escribir “laptop”, y el sistema de búsqueda nos ayudará a localizar las publicaciones que nos podrían interesar y que potencialmente, pudieran ser nuestro objeto perdido.

Escribimos laptop y presionamos el botón de **Buscar**.

The screenshot shows the 'Perdidos y Encontrados' app interface. At the top is a green header with the title 'Perdidos y Encontrados'. Below it is a 'Menú Principal' section. Under the menu, the name 'Pedro' is displayed next to three icons: a list icon, a green plus icon, and a right arrow icon. Below the name is a search input field containing the text 'laptop'. Underneath the search field is a 'Lugar:' label followed by an empty input field. Below that are two date selection fields labeled 'Entre fecha...' and 'Y fecha...'. A large grey button labeled 'BUSCAR' is highlighted with a red border. Below the button is a list of search results. The first result is highlighted with a blue border and contains the text '2018-12-13 Laptop HP Roja ITL 19M' followed by a magnifying glass icon.

Como vemos, nos devuelve esa única publicación que tiene la palabra “laptop” en su descripción.

Quizás, esta sea nuestra laptop, pero no estamos seguros.

Si nos interesa saber más de esta publicación, presionamos el **botón de la lupa** para “inspeccionar” la publicación.

The screenshot shows an Android application interface. At the top is a status bar with icons for notifications, Wi-Fi, cellular signal, and battery at 100%, along with the time 11:32 PM. Below the status bar is a green header with the text "Perdidos y Encontrados". Underneath is a section titled "Menú Principal". In this section, the name "Pedro" is displayed next to three icons: a list icon, a green plus icon, and a share icon. Below the name is a text input field containing the word "laptop". Under the input field is a label "Lugar:" followed by another input field. Below these are two date input fields labeled "Entre fecha..." and "Y fecha...". A grey button with the text "BUSCAR" is positioned below the date fields. At the bottom of the screen, there is a list item with the text "2018-12-13 Laptop HP ITL 19M Roja". To the right of this list item is a magnifying glass icon, which is highlighted with a red square border.

Al presionar el botón de la lupa, podemos ver la misma información que veíamos anteriormente (descripción, lugar y fecha) pero, además, podemos ver la información de contacto de la persona que hizo la publicación (su número de teléfono) y un botón para llamar a dicho número. Sin embargo, antes de llamar a la persona, debemos asegurarnos de que realmente se trata de nuestro objeto perdido. Para esto, esta pantalla te permite visualizar las imágenes que la persona subió al momento de hacer la publicación. Puedes verlas todas deslizándote a la derecha.



Así concluye este tutorial para usar la app **“Perdidos y Encontrados”**.