**package** com.example.jorgeperezcong.prueba;

**import** android.content.ContentValues;

**import** android.content.Context;

**import** android.database.Cursor;

**import** android.database.sqlite.SQLiteDatabase;

**import** android.database.sqlite.SQLiteOpenHelper;

**import** java.util.ArrayList;

**public class** AdminSQLLiteOpenHelper **extends** SQLiteOpenHelper{

**public** AdminSQLLiteOpenHelper(Context context) {

**super**(context, **"Login.db"**, **null**, 1);

}

@Override

**public void** onCreate(SQLiteDatabase db) {

db.execSQL(**"create table usuarios(usuario text primary key, contra text)"**);

db.execSQL(**"create table estampas(idEstampa integer primary key, numEstampa text, usuario text references usuarios )"**);

}

@Override

**public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {

db.execSQL(**"drop table if exists usuarios"**);

db.execSQL(**"create table usuarios(usuario text primary key, contra text)"**);

db.execSQL(**"drop table if exists estampas"**);

db.execSQL(**"create table estampas(idEstampa integer primary key AUTOINCREMENT, numEstampa text, usuario text references usuarios )"**);

}

*//método para crear un nuevo usuario y una nueva contraseña e insertarlo en la base de datos usuarios*

**public boolean** insert(String usuario, String contra){

SQLiteDatabase db= **this**.getWritableDatabase();

ContentValues contentValues=**new** ContentValues();

contentValues.put(**"usuario"**, usuario);

contentValues.put(**"contra"**, contra);

**long** ins=db.insert(**"usuarios"**,**null**, contentValues);

**if**(ins==1)**return false**;

**else return true**;

}

*//inserta una nueva estampa por usuario*

**public boolean** insertStamp(String usuario, String numEstampa){

SQLiteDatabase db = **this**.getWritableDatabase();

ContentValues contentValues=**new** ContentValues();

contentValues.put(**"numEstampa"**, numEstampa);

contentValues.put(**"usuario"**, usuario);

**long** ins=db.insert(**"estampas"**,**null**, contentValues);

**if** (ins==-1) **return false**;

**else return true**;

}

*//verifica que el nuevo nombre de usuario no esté tomado*

**public boolean** checkUsuario(String usuario){

SQLiteDatabase db=**this**.getReadableDatabase();

Cursor cursor=db.rawQuery(**"Select \* from usuarios where usuario =?"**, **new** String[]{usuario});

**if**(cursor.getCount()>0)**return false**;

**else return true**;

}

*//verifica que el usuario y contraseña coincidan*

**public boolean** usuarioContra(String usuario, String contra){

SQLiteDatabase db=**this**.getReadableDatabase();

Cursor cursor=db.rawQuery(**"Select \* from usuarios where usuario=? and contra=?"**, **new** String[]{usuario, contra});

**if**(cursor.getCount()>0) **return true**;

**else return false**;

}

*//Eliminar una estampa determinada*

**public void** deleteStamp(String stamp, String usuario){

**this**.getWritableDatabase().delete(**"estampas"**,**"numEstampa='"**+stamp+**"'and usuario='"**+usuario+**"'"**,**null**);

}

*//Regresa un cursor (la tabla completa) de todas las estampas que le corresponden a un usuario*

**public** Cursor getData(String usuario){

SQLiteDatabase db=**this**.getReadableDatabase();

Cursor cursor=db.rawQuery(**"Select numEstampa from estampas where usuario=?"**, **new** String []{usuario});

**return** cursor;

}

*//Regresa un cursor de la tabla con las estampas repetidas de un usuario*

**public** Cursor getDataRep(String usuario){

SQLiteDatabase db=**this**.getReadableDatabase();

Cursor cursor=db.rawQuery(**"Select numEstampa,COUNT(\*) from estampas where usuario=? GROUP BY numEstampa HAVING COUNT(\*)>1 "**, **new** String []{usuario});

**return** cursor;

}

}