



TRABAJO FIN DE GRADO

IES LAS SALINAS

ÁLVARO BLÁZQUEZ FERNÁNDEZ
2º DESARROLLO APLICACIONES MULTIPLATAFORMA
Anahí Mula

ÍNDICE / INDEX

RESUMEN / ABSTRACT	2
INTRODUCCIÓN / INTRODUCTION	3
HISTORIA DEL TATUAJE / STORY OF TATTOO	3
ANDROID STUDIO	5
LA APP / THE APP	5
JUSTIFICACIÓN / JUSTIFICATION	6
OBJETIVOS / GOALS.....	6
ANÁLISIS DE LA EMPRESA: DATOS DE LA ORGANIZACIÓN / BUSINESS ANALYSIS: ORGANIZATION DATA	7
ANÁLISIS DE REQUERIMIENTOS	8
DESARROLLO: BASES DE DATOS / DEVELOPMENT: DATABASES	10
Modelo Entidad-Relación / Entity-Relationship Diagram	10
Diagrama de clases / Class Diagram	10
Diagrama de casos de uso / Use Case Diagram	11
SQLite	11
DESARROLLO: CÓDIGO / DEVELOPMENT: CODE.....	13
DESARROLLO: INTERFAZ / DEVELOPMENT: INTERFACE.....	26
DESARROLLO: SISTEMAS DE GESTIÓN EMPRESARIAL / DEVELOPMENT: BUSINESS MANAGEMENT SYSTEMS	27
DESARROLLO: DOCUMENTACIÓN / DEVELOPMENT: DOCUMENTATION.....	27
MANUAL DE INSTALACIÓN / INSTALLATION MANUAL.....	27
MANUAL DEL USUARIO / USER MANUAL	27
MANUAL DE ADMINISTRACIÓN / ADMINISTRATION MANUAL.....	28
DESARROLLO: SEGURIDAD / DEVELOPMENT: SECURITY	28
CONCLUSIÓN / CONCLUSION	29
BIBLIOGRAFÍA / BIBLIOGRAPHY	29

RESUMEN / ABSTRACT

El siguiente trabajo muestra la aplicación ERP realizada para la empresa Ink Art Tattoo Studio. Una aplicación para facilitar la comunicación entre la empresa y el cliente. Para acceder a sus funciones hará falta registrarse en la aplicación para así poder almacenar los datos en la base de datos y luego poder logearte con total normalidad. Las funcionalidades de la aplicación son: con la vista de administrador, puedes consultar los usuarios que hay registrados, y con la vista de usuario normal tienes la capacidad de pedir cita para el estudio.

The following project shows the ERP application made for the company Ink Art Tattoo Studio. An application to facilitate communication between the company and the customer. To access its functions, it will be necessary to register in the application in order to store the data in the database and then be able to log in normally. The functionalities of the application are: with the administrator view, you can check the registered users, and with the normal user view you have the ability to request an appointment for the study.

INTRODUCCIÓN / INTRODUCTION

HISTORIA DEL TATUAJE / STORY OF TATTOO

Se cree que el mundo del tatuaje es algo actual, sin embargo, es una costumbre milenaria y quizá una de las primeras desde que surgió el ser humano moderno. La historia del tatuaje no es lineal. No tiene un único punto de partida ni una evolución cronológica constante. Lo que sí se sabe es que la práctica del tatuaje es común a sociedades humanas muy diversas desde épocas lejanas. Se tiene constancia de Ötzi, el “hombre de hielo”, un cazador neolítico encontrado en un glacial, cuyo cuerpo es el más antiguo que se haya preservado y que además presenta tatuajes en el cuerpo, concretamente en la espalda y en las rodillas. Un examen médico reveló que padecía artritis en las zonas tatuadas, por lo que podría indicar que estos tatuajes cumplían una función mágico-curativa.

El arte de tatuar no solo está documentado desde tiempos inmemoriales, sino que se encuentra en los 5 continentes con distintos significados, tanto positivos como negativos.

En las tribus de Polinesia la tradición del tatuaje está más consolidada y se caracteriza por sus figuras geométricas.

En Japón, la historia del tatuaje pasa por altibajos considerables. Existe un escrito que documenta su existencia con fines decorativos, pero con el paso del tiempo su significado se deteriora: pasa a utilizarse para estigmatizar a criminales, cuya imagen se refuerza cuando se asocia a la Yakuza, la mafia japonesa. En la Antigüedad grecorromana también se usa el tatuaje con fines discriminatorios. Posteriormente, el cristianismo rechazaría el tatuaje por su percepción alterada de la imagen divina a través del cuerpo.

El resurgimiento de la cultura del tatuaje en Occidente se produce a través de los exploradores y marineros que acompañan en las expediciones, en particular durante los viajes del capitán James Cook por la Polinesia en el siglo XVIII. Sin embargo, la práctica del tatuaje se ve asociada al mundo marginal y a la delincuencia.

En Estados Unidos aparecen los primeros profesionales que abren sus tiendas en Nueva York. A partir de los años 60 y 70, la comunidad hippie adopta el tatuaje y lo valoriza como un arte más. A partir de los 80, las distintas culturas difunden el tatuaje entre los jóvenes. Por este boom, el tatuaje se reivindica como una manifestación artística plena.

La historia del tatuaje en España es aún más reciente. Apenas se habla de ello en los años 60 y 70. A finales del siglo XX conoce un importante auge como fenómeno de moda.

Hoy en día, el tatuaje se considera una expresión artística respetable, hasta el punto de formar parte de la vida cotidiana, pero por desgracia aún se sigue percibiendo negativamente en entornos muy concretos.

It is believed that the world of tattooing is something current, however, it is an ancient custom and perhaps one of the first since the modern human being emerged. The history of the tattoo is not linear. It does not have a single starting point or a constant chronological evolution. What is known is that the practice of tattooing has been common to very diverse human societies since ancient times. There is evidence of Ötzi, the "ice man", a Neolithic hunter found on a glacier, whose body is the oldest that has been preserved and who also has tattoos on his body, specifically on his back and knees. A medical examination revealed that he suffered from arthritis in the tattooed areas, which could indicate that these tattoos fulfilled a magical-healing function.

The art of tattooing is not only documented from time immemorial, but it is found in the 5 continents with different meanings, both positive and negative.

In the Polynesian tribes, the tattoo tradition is more consolidated and is characterized by its geometric figures.

In Japan, the history of tattooing goes through considerable ups and downs. There is a document that documents its existence for decorative purposes, but over time its meaning deteriorates: it becomes used to stigmatize criminals, whose image is reinforced when associated with the Yakuza, the Japanese mafia. In Greco-Roman antiquity, tattoos are also used for discriminatory purposes. Subsequently, Christianity would reject the tattoo because of its altered perception of the divine image through the body.

The resurgence of tattoo culture in the West occurs through the explorers and sailors who accompany expeditions, particularly during Captain James Cook's voyages through Polynesia in the 18th century. However, the practice of tattooing is associated with the marginal world and crime.

The first professionals to open their stores in New York appear in the United States. From the 60s and 70s, the hippie community adopted the tattoo and valued it as one more art. Starting

in the 80s, different cultures spread the tattoo among young people. For this boom, the tattoo is claimed as a full artistic manifestation.

The history of tattooing in Spain is even more recent. Little is said about it in the 60s and 70s. At the end of the 20th century, it became known as a fashionable phenomenon.

Today, the tattoo is considered a respectable artistic expression, to the point of being part of everyday life, but unfortunately it is still negatively perceived in very specific environments.

ANDROID STUDIO

Android Studio es el IDE oficial para la plataforma Android. Está basado en el software IntelliJ IDEA de JetBrains.

El lenguaje preferido de Google para el desarrollo en Android es Kotlin, pero permite otros lenguajes de programación como Java y C++.

Android Studio is the official IDE for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android Development.

Kotlin replaced Java as Google's preferred language for Android app Development, but Java is still supported, as is C++.

LA APP / THE APP

La razón de por qué hacer esta aplicación es para ir agilizando el proceso de pedir cita para un estudio de tatuaje, además de quitar a los tatuadores una carga al no tener que estar pendiente de ir respondiendo los WhatsApp / emails que les lleguen de los clientes interesados. Sin una aplicación como esta, el proceso para pedir cita seguirá siendo igual de pesado para ambas partes, tanto como para los clientes como para los tatuadores, que se tenían que preocupar de ir preguntando al cliente específicamente qué quiere y luego acordar una fecha que le viniese bien a los dos. Con esta aplicación, solo necesitas registrarte y empezar con el proceso de pedir cita rellenando los campos necesarios.

The reason why making this application is to speed up the process of making an appointment for a tattoo studio, in addition to removing a burden from tattoo artists by not having to be aware of answering WhatsApp / emails that come from clients interested. Without an application like this, the process for making an appointment will continue to be equally burdensome for both parties, both for clients and tattoo artists, who had to worry about

asking the client specifically what he wants and then agree on a date that It would be good for both of you. With this application, you just need to register and start the process of making an appointment by filling in the necessary fields.

JUSTIFICACIÓN / JUSTIFICATION

¿Por qué hacer una aplicación como esta, para una empresa como esta? Opino que todos los sectores van a acabar modernizándose y utilizando las nuevas tecnologías, y por mi experiencia, este tipo de empresas no están muy modernizadas en cuanto al uso de nuevas tecnologías. Por este mismo motivo, he decidido empezar esta aplicación, para ir mostrando que es posible actualizar el sector sin cambiarlo demasiado.

La razón de por qué hacer la aplicación en Android Studio es porque es el IDE oficial para el desarrollo de las aplicaciones Android.

Why make an app like this one for a Company like this? I think that every sector will end up modernizing and using the new technologies. For my experience, these kind of companies are not very modernized in terms of the use of new technologies. For this same reason, I have decided to start this Application, to show that it is possible to update the sector without changing it too much.

The reason why making the application in Android Studio is because it is the official IDE for the development of Android applications.

OBJETIVOS / GOALS

Con esta aplicación tengo como objetivo prioritario el agilizar el proceso de pedir cita online para un estudio de tatuajes, actualizando el actual proceso que puede llegar a ser molesto y cansado para ambas partes.

With this application I have as a priority objective to streamline the process of making an appointment online for a tattoo studio, updating the current process that can be annoying and tiresome for both parties.

ANÁLISIS DE LA EMPRESA: DATOS DE LA ORGANIZACIÓN / BUSINESS

ANALYSIS: ORGANIZATION DATA

Ink Art Tattoo Studio es el primer estudio de tatuajes en la zona en el que ofrecemos al cliente el llevar una pieza de arte en su piel. En los últimos años los tatuajes están ganando más fama, y la gente de esta zona que quiere hacerse uno se ve obligada a moverse ya sea a Illescas o Aranjuez dado que en esta zona no hay ningún estudio, por lo que hemos visto una oportunidad para montar el negocio.

Ink Art Tattoo Studio es un estudio de tatuajes que cuenta con jóvenes talentos dispuestos a superarse con cada pieza realizada. Desde simples y pequeños tatuajes hasta extensas piezas, nuestros tatuadores se atreven a diseñar todo aquello que le pidan los clientes y en caso de que no tengas una referencia clara sobre tus tatuajes, nuestros tatuadores pueden elaborarte unos diseños personalizados en base a tus preferencias. Además, próximamente contaremos con una página web totalmente funcional, sencilla y fácilmente reconocible en la que los tatuadores colgarán sus trabajos para que llegue a mucha más gente. Esto mismo también lo haremos en nuestros perfiles de las redes sociales Twitter e Instagram. Nuestros servicios no acaban una vez ha terminado el tatuaje, sino que proporcionamos unos cuidados posteriores al tatuaje. Esta es la organización de la empresa:

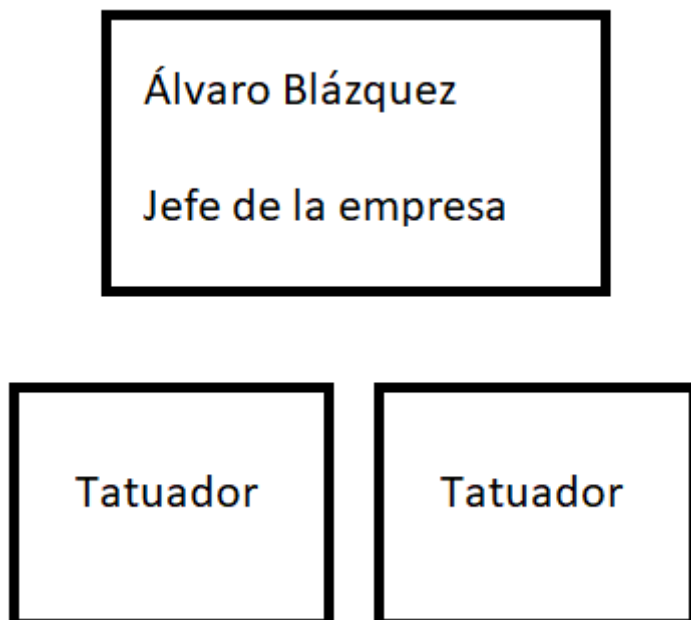


Imagen 1. Organización de la empresa

Ink Art Tattoo Studio is the first tattoo studio in the area where we offer the client to wear a piece of art on their skin. In recent years' tattoos are gaining more fame, and people in this area who want to get one are forced to move to either Illescas or Aranjuez since there is no study in this area, so we have seen an opportunity to start the business.

Ink Art Tattoo Studio is a tattoo studio that has young talents willing to excel with each piece done. From simple and small tattoos to extensive pieces, our tattoo artists dare to design everything that clients ask of them and in case you do not have a clear reference about your tattoos, our tattoo artists can elaborate custom designs based on your preferences. In addition, soon we will have a fully functional, simple and easily recognizable web page where tattoo artists will post their work so that it reaches many more people. We will also do the same on our social media profiles Twitter and Instagram. Our services do not end once the tattoo is finished, but we provide after-tattoo care.

ANÁLISIS DE REQUERIMIENTOS

Para el análisis de requerimientos he visto la necesidad de crear una aplicación móvil para facilitar la comunicación entre empleado y cliente. Esta aplicación será para Android, por lo que será indispensable un dispositivo (móvil, tableta) con este sistema operativo.

Las funciones de la aplicación, para la vista del cliente, será el poder pedir cita de manera online después de registrarse en la aplicación. En cuanto al administrador de la aplicación, podrá consultar los usuarios registrados.

A continuación, presento el diagrama de Gantt del proyecto, dividido en 6 franjas que ocupan de 2 a 6 horas cada una, dependiendo de la tarea.

Next, I present the Gantt chart of the project, divided into 6 strips that take from 2 to 6 hours each, depending on the task.

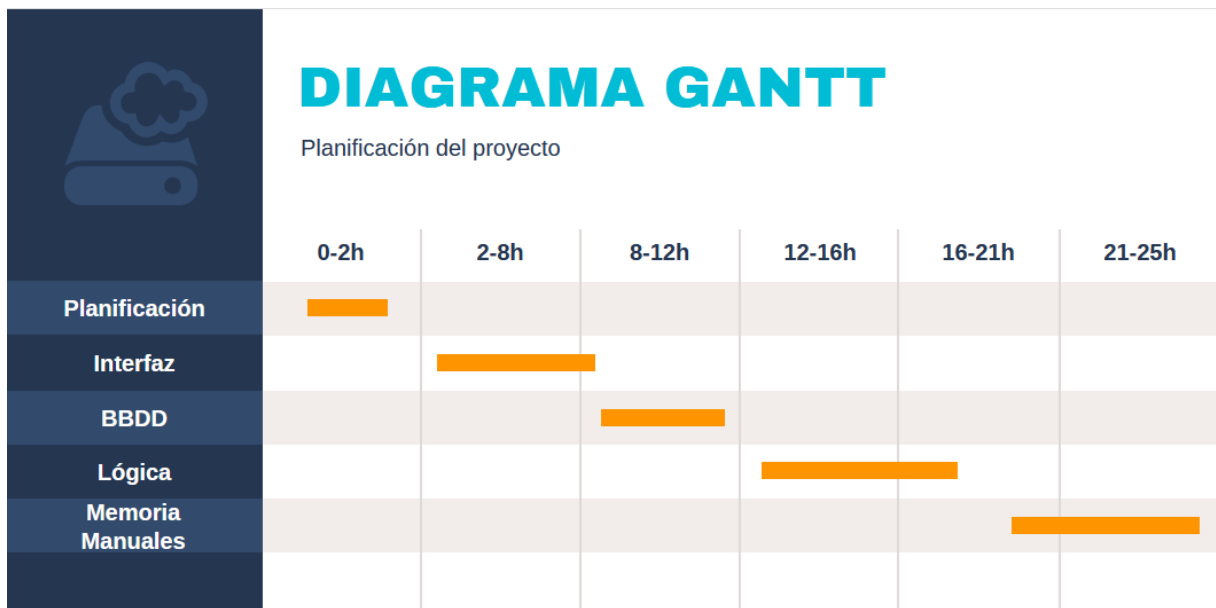


Imagen 2. Diagrama de Gantt

En cuanto a gastos, la empresa supone un gasto de 22€ para la licencia de Google y poder subir la aplicación a la Store. También habrá un gasto de 800€ para un ordenador que estará montado en la oficina y que servirá como servidor para que, en caso de fallos, no se pierdan los datos almacenados en la base de datos, y otro gasto de 1400€, que se desglosa en 700€ para dos móviles (uno para cada tatuador).

La empresa contará como hardware con un ordenador de sobremesa situado en la tienda y dos móviles Android, todos los dispositivos con conexión a Internet. En el caso de los dos teléfonos móviles, bastará que tengan acceso a la aplicación de la empresa para ver las citas pendientes.

For the requirements analysis I have seen the need to create a mobile application to facilitate communication between employee and customer. This application will be for Android, so a device (mobile, tablet) with this operating system will be essential.

The functions of the application, for the customer's view, will be to be able to make an appointment online after registering in the application. As for the application administrator, you can consult the registered users.

As for expenses, the company assumes an expense of € 22 for the Google license and being able to upload the application to the Store. There will also be an expense of € 800 for a computer that will be mounted in the office, and another expense of € 1,400, which is broken down into € 700 for two phones (one for each tattoo artist).

The company will have as a hardware a desktop computer located in the store and two Android mobiles, all devices with an Internet connection. In the case of the two mobile phones, it will be enough that they have access to the company application to see the pending appointments.

DESARROLLO: BASES DE DATOS / DEVELOPMENT: DATABASES

Modelo Entidad-Relación / Entity-Relationship Diagram

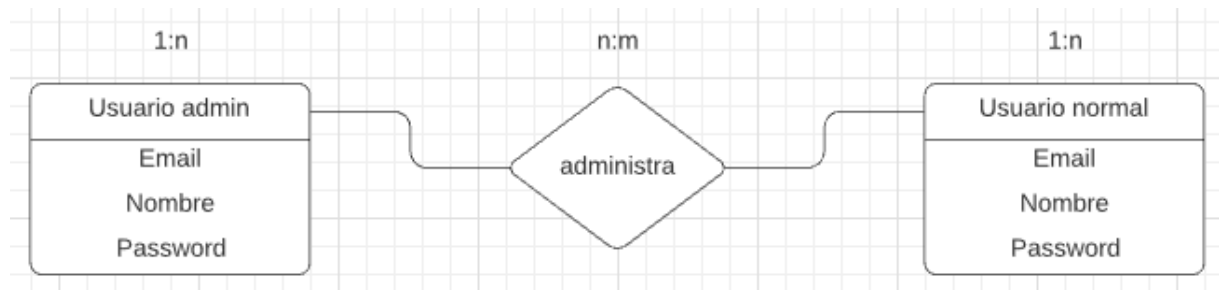


Imagen 3. MER

En el diagrama entidad-relación, en este caso hay 2 entidades: el usuario administrador y el usuario normal. Un usuario administrador administra a 0 o varios usuarios normales, y un usuario normal puede estar administrado por uno o varios usuarios administradores.

In the entity-relationship diagram, in this case there are 2 entities: the administrator user and the normal user. An administrator user manages 0 or more normal users, and a normal user can be administered by one or more administrator users.

Diagrama de clases / Class Diagram

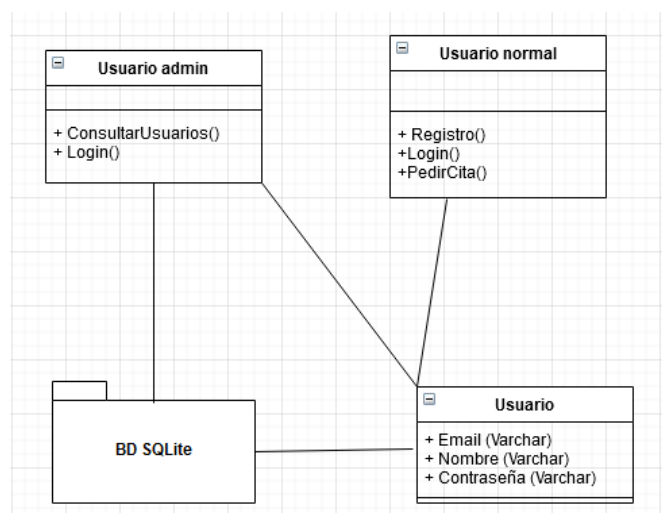


Imagen 4. Diagrama de clases

Diagrama de casos de uso / Use Case Diagram

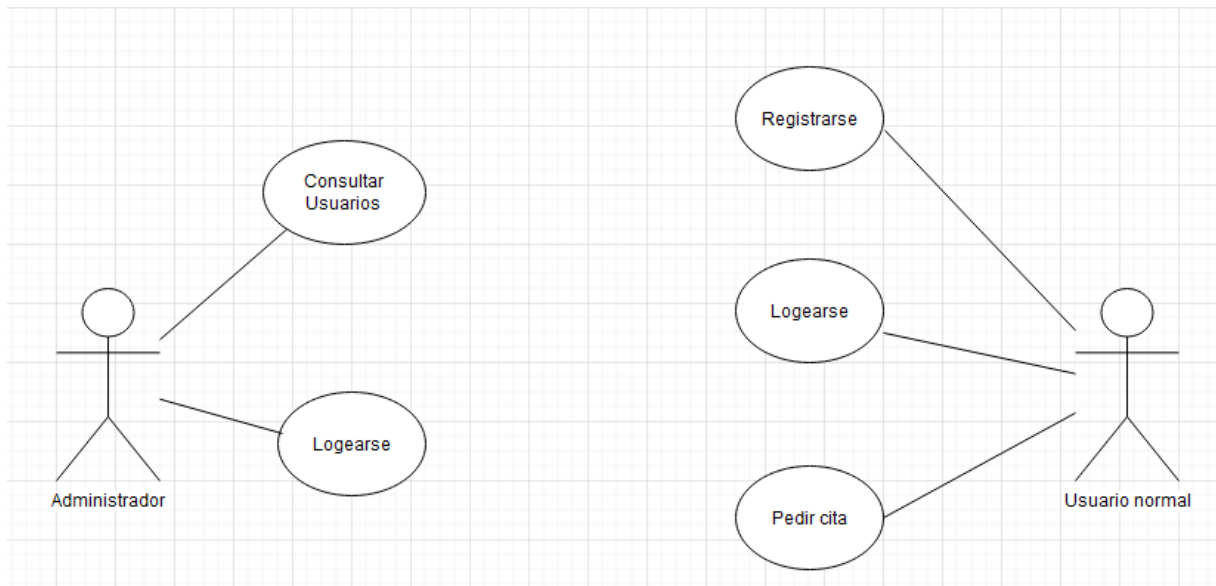


Imagen 5. Diagrama de casos de uso

En el diagrama de casos de uso se puede ver como un usuario administrador es capaz de consultar usuarios o logearse, mientras que el usuario normal es capaz de registrarse, logearse y pedir citas.

In the use case diagram you can see how an administrator user is able to consult users or log in, while the normal user is able to register, log in and request appointments.

SQLite

SQLite es un Sistema de Gestión de Bases de Datos (SGBD) relacional compatible con ACID, contenida en una biblioteca escrita en C. A diferencia de los Sistemas de Gestión de Bases de Datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. El conjunto de la base de datos (definiciones, tablas, índices y los propios datos), son guardados como un solo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

Por su pequeño tamaño, SQLite es muy adecuado para Android, por lo que he decidido utilizarlo en este proyecto.

En la siguiente imagen se puede ver cómo se ha creado la tabla con la que se va a trabajar usando SQLite:

SQLite is a relational Database Management System contained in a C library.

Unlike client-server Database Management Systems, the SQLite engine is not a separate process with which the main program communicates. Instead, the SQLite library is linked to the program, becoming an integral part of it. The program uses the SQLite functionality through simple calls to subroutines and functions. The set of the database (definitions, tables, indexes and the data itself), are saved as a single standard file on the host machine. This simple design is accomplished by locking the entire database file at the beginning of each transaction.

Due to its small size, SQLite is very suitable for Android, so I have decided to use it in this project.

In the following image you can see how the table to work with using SQLite has been created:

```
public class AdminSQLiteOpenHelper extends SQLiteOpenHelper {
    public AdminSQLiteOpenHelper(@Nullable Context context, @Nullable SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    private static final String tablaUsuarios = "usuarios";

    private static final String cEmail = "email";
    private static final String cNombre = "nombre_usuario";
    private static final String cPassword = "password";

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + tablaUsuarios + "("
            + cEmail + " VARCHAR(50) PRIMARY KEY," + cNombre + " varchar(20),"
            + cPassword + " varchar(20) + ")");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    }
}
```

Imagen 6. Creación de la tabla

Para lograr esto, solo he utilizado el constructor por defecto de SQLite y después he configurado como he querido llamar a la tabla (en este caso, "usuarios") y a los distintos campos de la misma (email, nombre_usuario y password). Una vez hecho esto, solo queda crear la propia tabla en el método por defecto onCreate, desde el que se inicializará la sentencia execSQL y así poder crear la tabla. En mi caso, de los campos creados anteriormente, he utilizado el campo "email" como clave primaria de la tabla, y además es un campo alfanumérico de hasta 50 bytes. Las columnas de nombre y password son alfanuméricos también con un tamaño de 20 bytes cada una.

To achieve this, I have only used the default SQLite constructor and then I have configured how I wanted to call the table (in this case, "usuarios") and the different fields of the table (email, nombre_usuario and password). Once this is done, it only remains to create the table itself in the default method onCreate, from which the execSQL statement will be initialized and thus be able to create the table. In my case, of the fields created previously, I have used the field "email" as the primary key of the table, and it is also an alphanumeric field of up to 50 bytes. The name and password columns are alphanumeric also with a size of 20 bytes each.

DESARROLLO: CÓDIGO / DEVELOPMENT: CODE

La primera cosa que vemos al iniciar la aplicación es la siguiente pantalla:

The first thing we see when starting the application is the following screen:

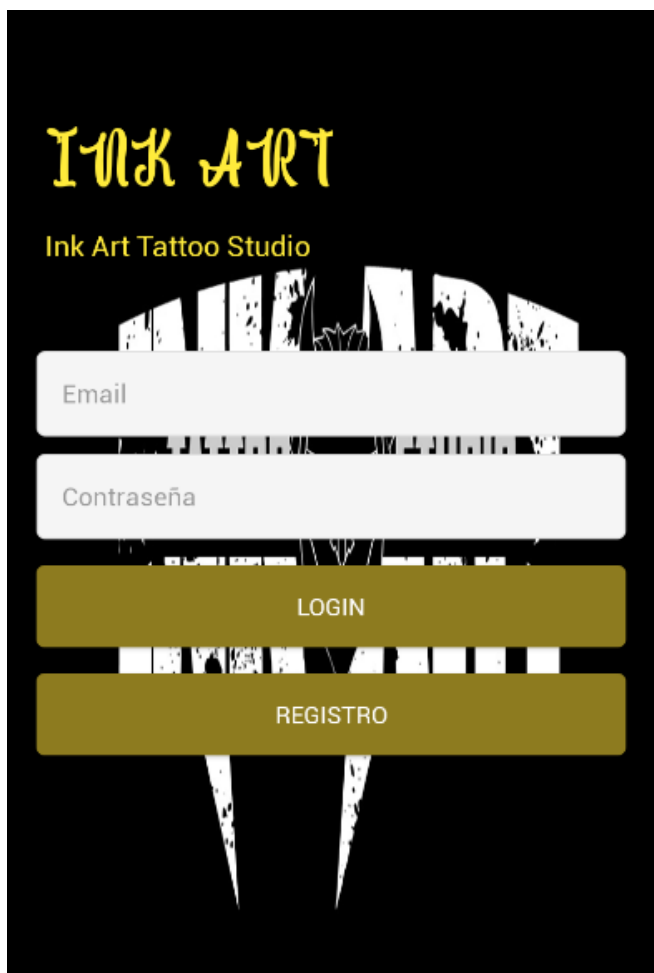


Imagen 7. Pantalla de Login

Podemos ver que cuenta con dos TextView, cuya función son mostrar el título y el subtítulo, dos EditText, en los cuales introduciremos nuestros datos para poder logearnos en la aplicación, y dos botones. El primer botón sirve para confirmar los datos introducidos

previamente y poder avanzar a la siguiente pantalla. El segundo botón es, por si no se tuviese una cuenta registrada en la aplicación, te lleva a otra pantalla distinta para poder crear tu cuenta. La lógica detrás de esta pantalla es la siguiente:

We can see that it has two TextViews, whose function is to show the title and the subtitle, two EditText, in which we will introduce our data to be able to log into the application, and two buttons. The first button is used to confirm previously entered data and to advance to the next screen. The second button is, in case you don't have an account registered in the application, it takes you to a different screen to create your account. The logic behind this screen is as follows:

```
public void logear(View v){
    Intent i;
    AdminSQLiteOpenHelper adminSQLiteOpenHelper = new AdminSQLiteOpenHelper( context: this, name: "usuarios", factory: null, version: 1);
    SQLiteDatabase sqLiteDatabase = adminSQLiteOpenHelper.getWritableDatabase();
    String email = log_correo.getText().toString();
    String clave = log_clave.getText().toString();
    Cursor fila = sqLiteDatabase.rawQuery( sql: "SELECT password, nombre_usuario FROM usuarios WHERE email = '" + email + "'", selectionArgs: null);
    String nombreUsuario = "";
    String contraUsuario = "";
    if(fila.moveToFirst()){
        contraUsuario = fila.getString( columnIndex: 0);
        nombreUsuario = fila.getString( columnIndex: 1);
    }
    if(email.equals("admin@inkart.com") && clave.equals(contraUsuario)){
        i = new Intent( packageContext: this, AdminUser.class);
        startActivity(i);
    }
    if(!email.equals("admin@inkart.com") && !email.equals("") && clave.equals(contraUsuario)){
        i = new Intent( packageContext: this, NormalUser.class);
        i.putExtra( name: "nombre_usuario", nombreUsuario);
        startActivity(i);
    }
    sqLiteDatabase.close();
}
```

Imagen 8. Método Login

Con este método comprobamos que existe la cuenta. Hay dos usuarios dummy ya existentes, uno que funciona como usuario normal y otro que sirve como administrador único. Esto se refiere a que, si se introducen los datos de este usuario, te lleva a una pantalla de administración en lugar de a una pantalla que vería el usuario normal. Estos usuarios por defecto se crean con el siguiente método:

With this method we verify that the account exists. There are two existing dummy users, one that works as a normal user and the other that serves as a single administrator. This refers to the fact that, if the data of this user is entered, it takes you to an administration screen instead of to a screen that the normal user would see. These users by default are created with the following method:

```

public void datosPorDefecto(String[] datos){
    AdminSQLiteOpenHelper adminSQLiteOpenHelper = new AdminSQLiteOpenHelper( context: this, name: "usuarios", factory: null, version: 1);
    SQLiteDatabase sqLiteDatabase = adminSQLiteOpenHelper.getWritableDatabase();

    ContentValues contenido = new ContentValues();
    contenido.put("email", datos[0]);
    contenido.put("nombre_usuario", datos[1]);
    contenido.put("password", datos[2]);

    sqLiteDatabase.insert( table: "usuarios", nullColumnHack: null, contenido);
    sqLiteDatabase.close();
}

```

Imagen 9. Método datos por defecto

```

String[] datosAdmin = {"admin@inkart.com", "admin", "admin"};
String[] datosNormal = {"pleb@pleb.com", "pleb", "1234"};

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

    log_correo = findViewById(R.id.et_email);
    log_clave = findViewById(R.id.et_password);
    btn_log = findViewById(R.id.btn_login);

    datosPorDefecto(datosAdmin);
    datosPorDefecto(datosNormal);
}

```

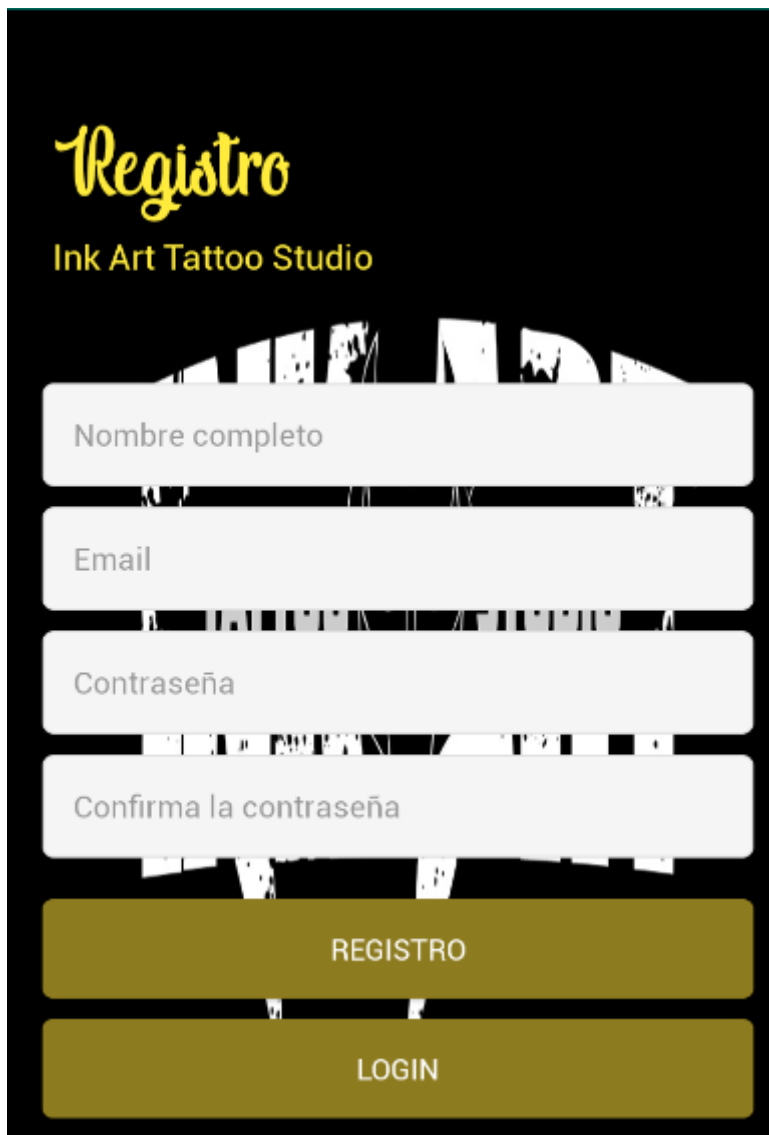
Imagen 10. Inicializar métodos

Esta imagen de arriba muestra la creación de los arrays con los datos de los usuarios, y su inicialización en el método OnCreate.

This image above shows the creation of the arrays with the user data, and their initialization in the OnCreate method.

Si pulsásemos el botón de Registro, nos llevaría a la siguiente pantalla:

If we pressed the Registro button, it would take us to the following screen:



Registro

Ink Art Tattoo Studio

Nombre completo

Email

Contraseña

Confirma la contraseña

REGISTRO

LOGIN

Imagen 11. Pantalla Registro

Se puede ver que en esta interfaz hay dos TextView con la misma función que los que había en la interfaz de Login, cuatro EditText en los que habrá que poner tus datos para registrarte y dos botones: uno para confirmar estos datos y registrarte y otro para ir a la pantalla de Login. La lógica detrás es la siguiente:

It can be seen that in this interface there are two TextViews with the same function as those in the Login interface, four EditText in which you will have to put your data to register and two buttons: one to confirm these data and register and another to go to the Login screen. The logic behind is as follows:

```

public void irLogin(View v){
    Intent irLog = new Intent( packageContext: this, MainActivity.class);
    startActivity(irLog);
}

public void insertar(View v){
    Intent irUser;
    AdminSQLiteOpenHelper adminSQLiteOpenHelper = new AdminSQLiteOpenHelper( context: this, name: "usuarios", factory: null, version: 1);
    SQLiteDatabase sqLiteDatabase = adminSQLiteOpenHelper.getWritableDatabase();
    String email = correo.getText().toString();
    String name = nombre.getText().toString();
    String clave = pass.getText().toString();
    String re_clave = repass.getText().toString();

    if(email.length() <= 0){
        Toast.makeText( context: this, text: "Por favor, rellene todos los campos", Toast.LENGTH_LONG).show();
    } else if (name.length() <= 0){
        Toast.makeText( context: this, text: "Por favor, rellene todos los campos", Toast.LENGTH_LONG).show();
    } else if (clave.length() <= 0){
        Toast.makeText( context: this, text: "Por favor, rellene todos los campos", Toast.LENGTH_LONG).show();
    } else if (re_clave.length() <= 0){
        Toast.makeText( context: this, text: "Por favor, rellene todos los campos", Toast.LENGTH_LONG).show();
    } else {
        if(clave.equals(re_clave) == true){
            ContentValues contenido = new ContentValues();
            contenido.put("email", email);
            contenido.put("nombre_usuario", name);
            contenido.put("password", clave);

            sqLiteDatabase.insert( table: "usuarios", nullColumnHack: null, contenido);
            sqLiteDatabase.close();

            irUser = new Intent( packageContext: this, NormalUser.class);
            startActivity(irUser);
        } else {
            Toast.makeText( context: this, text: "Las contraseñas no coinciden", Toast.LENGTH_LONG).show();
        }
    }
}

```

Imagen 12. Método insertar

El método irLogin corresponde al segundo botón, que es el que te lleva a la pantalla de Login. El segundo método hace la inserción en la base de datos y te lleva a la pantalla de usuario normal al terminar el registro.

The irLogin method corresponds to the second button, which is the one that takes you to the Login screen. The second method does the insertion in the database and takes you to the normal user screen at the end of the registration.

Para hacer una prueba, vamos a logearnos con uno de los dummy que ya tenemos creados.

To do a test, we are going to log in with one of the dummy that we already have created.

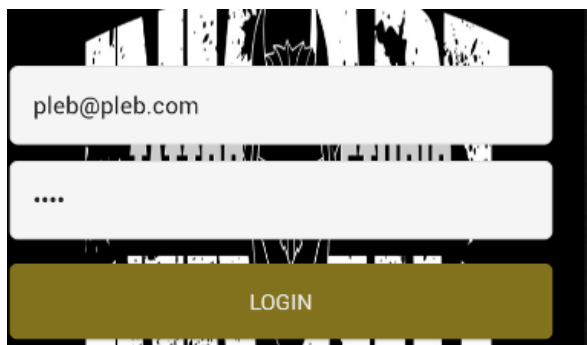


Imagen 13. Prueba Login

Una vez se ha logeado, aparecerá la siguiente pantalla:

Once you have logged in, the following screen will appear:

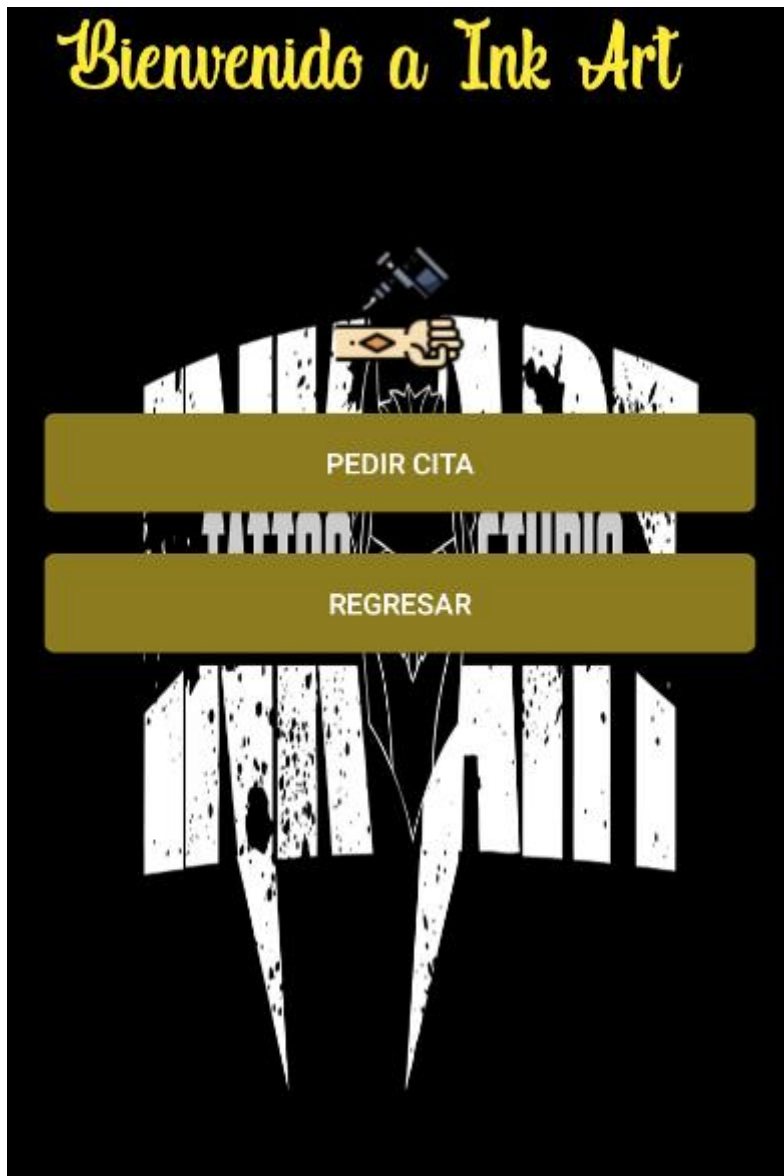


Imagen 14. Menú usuario

En esta interfaz hay un TextView a modo de título de bienvenida, un ImageView cuya imagen es un brazo siendo tatuado, y dos botones: el botón de Regresar te lleva de vuelta al menú de Login, y el botón de Pedir Cita te moverá a otra pantalla en la que podrás llevar a cabo el proceso de pedir cita rellenando antes unos datos. La lógica es la siguiente:

In this interface there is a TextView as a welcome title, an ImageView whose image is an arm being tattooed, and two buttons: the Return button takes you back to the Login menu, and the Request Appointment button will move you to another screen in which you can carry out

the process of requesting an appointment by filling in some information first. The logic is as follows:

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

    bienvenidopleb = findViewById(R.id.textView_PlebUser);
    tattoo = findViewById(R.id.iv_tattoo);
    pideCita = findViewById(R.id.btn_pedirCitaMenu);
    volver = findViewById(R.id.bt_volverMenu);
}

public void irMenuCita(View v){
    Intent i = new Intent( packageContext: this, MenuPedirCita.class);
    startActivity(i);
}

public void volver(View v){
    Intent irAtras = new Intent( packageContext: this, MainActivity.class);
    startActivity(irAtras);
}
```

Imagen 15. Métodos menú usuario

En este código simplemente hay dos métodos para los dos botones que te llevan a distintas pantallas: el menú para pedir cita o el menú de Login.

In this code there are simply two methods for the two buttons that take you to different screens: the menu to make an appointment or the Login menu.



Imagen 16. Menú pedir cita

El menú para pedir cita está compuesto por dos TextView que funcionan a modo de mensajes de título y subtítulo, cuatro EditText para rellenar información como el teléfono, la zona, el tamaño y una breve descripción, y un botón para confirmar estos datos y que te lleve a una pantalla nueva en la que se mostrará un resumen de estos datos introducidos. La lógica es esta:

The menu to request an appointment is made up of two TextViews that work as title and subtitle messages, four EditText to fill in information such as the phone, the area, the size and a brief description, and a button to confirm this information and that you take you to a new screen where a summary of these entered data will be displayed. The logic is this:

```

public void confirmar(View v){
    String telef = tlfn.getText().toString();
    String zona_Confirmado = zona.getText().toString();
    String size_Confirmado = size.getText().toString();
    String design_Confirmado = design.getText().toString();

    if(telef.length()<=0){
        Toast.makeText( context: this, text: "Por favor, rellene todos los campos", Toast.LENGTH_LONG).show();
    }
    else if(zona_Confirmado.length()<=0){
        Toast.makeText( context: this, text: "Por favor, rellene todos los campos", Toast.LENGTH_LONG).show();
    }
    else if(size_Confirmado.length()<=0){
        Toast.makeText( context: this, text: "Por favor, rellene todos los campos", Toast.LENGTH_LONG).show();
    }
    else if(design_Confirmado.length()<=0){
        Toast.makeText( context: this, text: "Por favor, rellene todos los campos", Toast.LENGTH_LONG).show();
    }
    else {
        Intent i = new Intent( packageContext: this, ResumenDatos.class);
        i.putExtra( name: "tel", telef);
        i.putExtra( name: "zona", zona_Confirmado);
        i.putExtra( name: "tamano", size_Confirmado);
        i.putExtra( name: "diseno", design_Confirmado);
        startActivity(i);
    }
}
}

```

Imagen 17. Confirmar cita

En este método, se recogen todos los datos introducidos en los EditText para pasarlos a la última pantalla. Además, se asegura de que no se deje ningún campo sin rellenar.

In this method, all the data entered in the EditText is collected to pass it to the last screen. In addition, it ensures that no field is left unfilled.

Imagen 18. Ejemplo cita 1

Se va a hacer la prueba con estos datos introducidos.

The test will be done with these data entered.



Imagen 18. Ejemplo cita 2

Esta última pantalla sirve como resumen de los datos que se han introducido previamente. Cuenta con un TextView en el que se muestra el mensaje y los datos y un botón para volver al menú de Login. Esta es la lógica:

This last screen serves as a summary of the data that has been previously entered. It has a TextView in which the message and data are displayed and a button to return to the Login menu. This is the logic:

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_resumen_datos);  
  
    resumen = findViewById(R.id.textView_ResumenDatos);  
  
    datos = getIntent().getExtras();  
  
    resumen.setText("Tu reserva ha sido completada. Este es el resumen de los datos introducidos: " + System.getProperty("line.separator") + "Telefono: " + datos.getString( key "tel") +  
        System.getProperty("line.separator") + "Zona: " + datos.getString( key "zona") + System.getProperty("line.separator")  
        + "Tamano: " + datos.getString( key "tamano") + System.getProperty("line.separator") + "Diseño: " + datos.getString( key "diseno") + System.getProperty("line.separator") +  
        "Nos pondremos en contacto contigo para concretar la fecha con la mayor brevedad posible.");  
}  
  
public void volver (View v){  
    Intent i = new Intent( packageContext this, MainActivity.class);  
    startActivity(i);  
}
```

Imagen 19. Mensaje datos

Este método recoge los datos que se envían desde la pantalla anterior y muestra el mensaje en el TextView.

Una vez comprobada la funcionalidad que tiene el usuario normal, se va a probar al usuario administrador.

This method collects the data that is sent from the previous screen and displays the message in the TextView.

Once the functionality of the normal user has been verified, the administrator user will be tested.

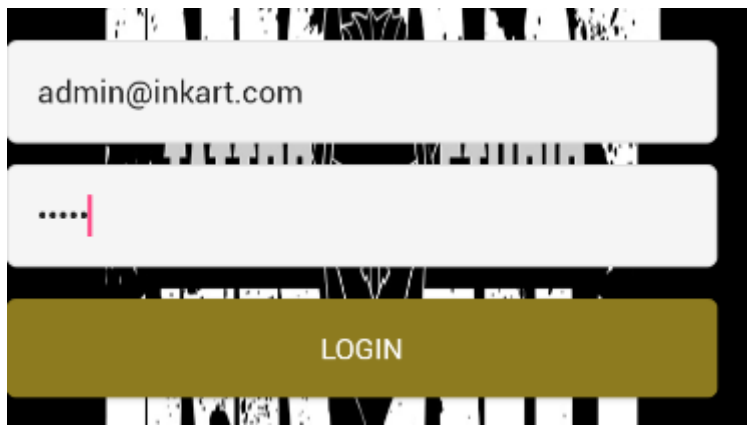


Imagen 20. Ejemplo login admin

Desde el menú de Login se introducen los datos de administrador. Cabe decir que, en caso de que no se introduzcan los datos por defecto para la cuenta de administrador, el sistema no detectará que es administrador, pues solo existirá uno para la aplicación.

From the Login menu enter the administrator data. It should be said that, in case the default data for the administrator account is not entered, the system will not detect that you are an administrator, since there will only be one for the application.

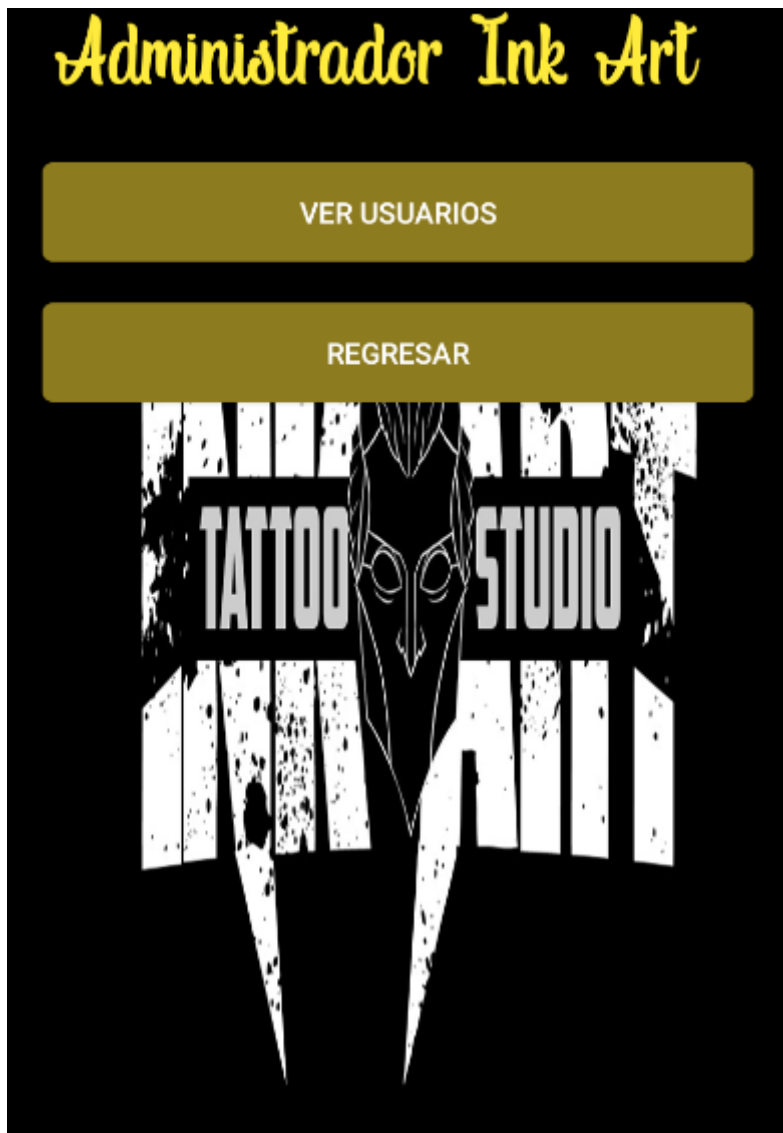


Imagen 21. Menú admin 1

La pantalla de administrador es distinta de la del usuario normal, pues cuenta con un TextView a modo de título y dos botones: uno que hará la consulta y luego lo mostrará en un ListView oculto y el segundo botón que te devolverá al menú de Login.

The administrator screen is different from that of the normal user, since it has a TextView as a title and two buttons: one that will query and then show it in a hidden ListView and the second button that will return you to the Login menu.

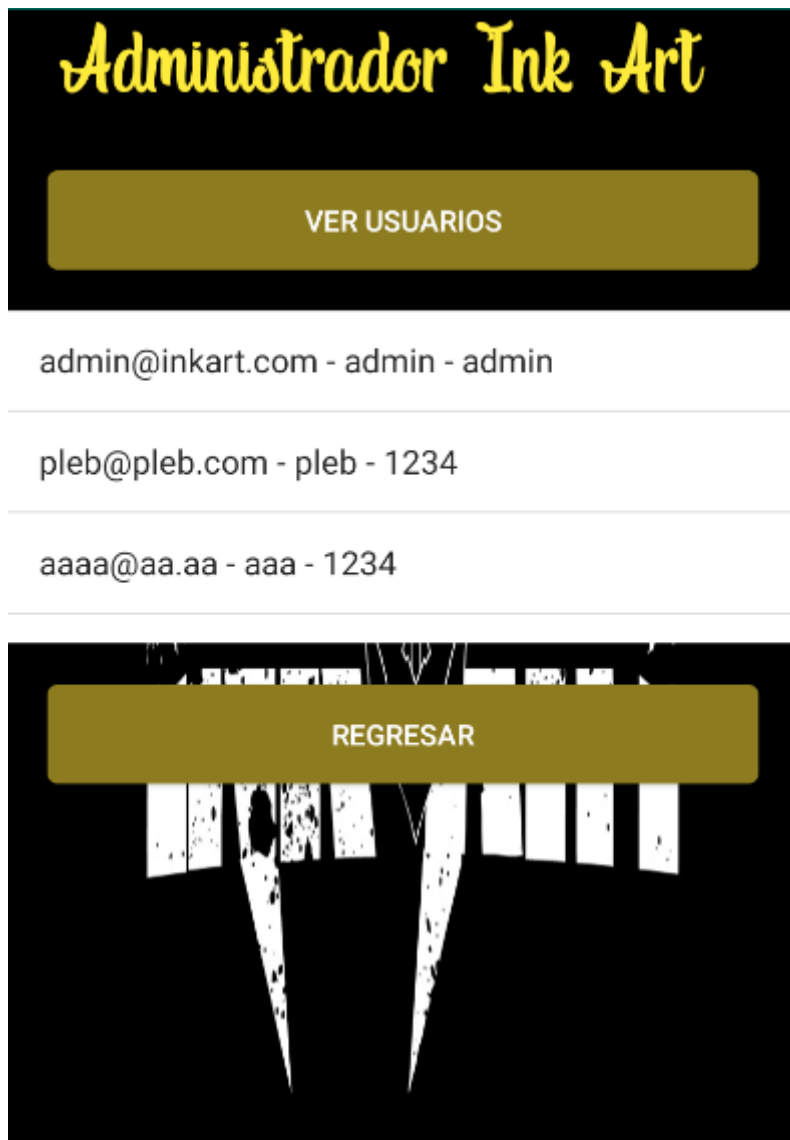


Imagen 21. Menú admin 2

Al presionar el botón de Ver usuarios aparecerá el ListView previamente oculto y mostrará todos los usuarios registrados en la base de datos. La lógica detrás de esto es la siguiente:

Pressing the View Users button will bring up the previously hidden ListView and show all the users registered in the database. The logic behind this is as follows:

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_admin_user);

    titulo = findViewById(R.id.tv_adminUser);
    bt_consulta = findViewById(R.id.bt_consultarUser);
    bt_volver = findViewById(R.id.bt_volverMenu);
    lista = findViewById(R.id.listaConsulta);

    lista.setVisibility(View.GONE);
}

public void consultaBD (View v){
    lista.setVisibility(View.VISIBLE);
    AdminSQLiteOpenHelper adminSQLiteOpenHelper = new AdminSQLiteOpenHelper( context: this, name: "usuarios", factory: null, version: 1);
    SQLiteDatabase sqLiteDatabase = adminSQLiteOpenHelper.getReadableDatabase();

    Cursor cursor = sqLiteDatabase.rawQuery( sql: "SELECT * FROM usuarios", selectionArgs: null);
    if (cursor.moveToFirst()){
        do{
            resultados.add(cursor.getString( columnIndex: 0) + " - " + cursor.getString( columnIndex: 1) + " - " + cursor.getString( columnIndex: 2));
        } while (cursor.moveToNext());
    }

    sqLiteDatabase.close();
    ArrayAdapter<String> adapter = new ArrayAdapter<>( context: this, android.R.layout.simple_list_item_1, resultados);
    lista.setAdapter(adapter);
}

public void volver (View v){
    Intent volver = new Intent( packageContext: this, MainActivity.class);
    startActivity(volver);
}

```

Imagen 22. Método consulta

En el método onCreate se pone el ListView en oculto con el método View.GONE. En el método consultaBD se publica el ListView y se muestran los datos almacenados en la base de datos por orden (email, nombre de usuario y contraseña). El método volver te regresa al menú de Login.

In the onCreate method, the ListView is hidden with the View.GONE method. In the queryBD method, the ListView is published and the data stored in the database is displayed in order (email, username and password). The return method returns you to the Login menu.

DESARROLLO: INTERFAZ / DEVELOPMENT: INTERFACE

Para esta aplicación he utilizado un diseño de interfaz que contiene el logo de la empresa Ink Art como fondo de pantalla que aparece en todas las pantallas. Para el texto de los TextView he usado un amarillo que contrastase con el fondo negro. Para los botones he utilizado un color amarillo más apagado para que se pudiese leer mejor su contenido.

For this application I have used an interface design that contains the Ink Art company logo as the wallpaper that appears on all screens. For the TextView text I have used a yellow that contrasts with the black background. For the buttons I have used a duller yellow color so that their content could be read better.

DESARROLLO: SISTEMAS DE GESTIÓN EMPRESARIAL / DEVELOPMENT: BUSINESS MANAGEMENT SYSTEMS

Para este apartado, voy a enumerar los beneficios que tiene usar un ERP como esta aplicación.

For this section, I am going to list the benefits of using an ERP like this application.

- 1.- Optimización de los procesos de gestión / Optimization of management processes
- 2.- Mejora el proceso de toma de decisiones / Improves the decision-making process
- 3.- Seguridad de datos / Data security
- 4.- Modularidad y escalabilidad / Modularity and scalability
- 5.- Adaptación a las necesidades reales / Adaptation to real needs
- 6.- Mayor control y trazabilidad / Greater control and traceability
- 7.- Automatización de tareas / Task automation

DESARROLLO: DOCUMENTACIÓN / DEVELOPMENT: DOCUMENTATION

MANUAL DE INSTALACIÓN / INSTALLATION MANUAL

Para instalarlo, tenemos dos opciones: una es escanear el código QR de las tarjetas de visita de la empresa, que te redigirá a la Google Store, o bien puedes buscarlo directamente en la Store en caso de no tener una tarjeta de la empresa.

To install it, we have two options: one is to scan the QR code of the company's business cards, which will redirect you to the Google Store, or you can search for it directly in the Store in case you don't have a company card.

MANUAL DEL USUARIO / USER MANUAL

El usuario normal, una vez tenga instalada la aplicación, deberá registrarse en la misma desde el menú de Registro. Una vez tenga su cuenta, podrá logearse en la aplicación. Al estar ya dentro, podrá pedir cita para el estudio pulsando el botón de Pedir cita. Desde el menú al que te redirige, introduces los datos que te piden y los confirmas, entonces te llevará a otra pantalla y tu cita habrá sido registrada.

The normal user, once the application is installed, must register in it from the Registration menu. Once you have your account, you can login in the application. Once you are inside, you

can request an appointment for the study by pressing the Request appointment button. From the menu to which you are redirected, you enter the information they ask for and confirm it, then it will take you to another screen and your appointment will have been registered.

MANUAL DE ADMINISTRACIÓN / ADMINISTRATION MANUAL

El usuario administrador deberá logearse en la aplicación con sus datos de administrador, ya predefinidos en los métodos del código, entonces será redirigido a la pantalla de administración, desde la cual puede consultar todos los usuarios registrados en la base de datos.

The administrator user must log into the application with their administrator data, already predefined in the code methods, then they will be redirected to the administration screen, from which they can consult all the users registered in the database.

DESARROLLO: SEGURIDAD / DEVELOPMENT: SECURITY

La aplicación cuenta con medidas de seguridad básicas como pedir un correo y una contraseña al logearte y al registrarte, para evitar que usuarios ajenos a la aplicación puedan tener acceso a la misma. Además, la empresa cuenta con un ordenador de sobremesa que actúa a modo de servidor en el que se almacenarán los datos de la base de datos por si hubiese algún error como un corte de electricidad.

The application has basic security measures such as requesting an email and a password when logging in and registering, to prevent users outside the application from having access to it. In addition, the company has a desktop computer that acts as a server where data from the database will be stored in case there is an error such as a power outage.

CONCLUSIÓN / CONCLUSION

Para acabar, debo decir que a pesar de tener un proyecto de corta duración y haber desechado ideas que implementar a la aplicación, creo que he cumplido con el objetivo principal de crear un ERP para tu empresa, y es agilizar un poco ciertos trámites, además de que una empresa del sector de los tatuajes utilice las nuevas tecnologías para el negocio.

To finish, I must say that despite having a short project and having discarded ideas to implement the application, I believe that I have fulfilled the main objective of creating an ERP for your company, and that is to speed up certain procedures a bit, in addition for a company in the tattoo industry to use new technologies for business.

BIBLIOGRAFÍA / BIBLIOGRAPHY

- Draw.io
- Lucidchart.com
- <https://juanpetattoo.es/blog/historia-del-tatuaje/>
- <https://developer.android.com/studio/intro>
- <https://es.venngage.com/features/hacer-diagrama-de-gantt>
- <https://es.wikipedia.org/wiki/SQLite>
- <https://www.wordreference.com/>