

Aplicación móvil “TaskMaster”

Trabajo de Fin de Grado

Índice del Trabajo

1 ESTUDIO PRELIMINAR DEL PROBLEMA	3
1.1 Descripción preliminar (Justificación del proyecto)	
1.2 Objetivos generales.	
2 Casos de Uso.....	8
2.2.1 Diagrama de casos de uso	
2.2.2 Descripción de los casos de uso	
3 DISEÑO DE LA APLICACIÓN.....	12
3.1 Prototipo de la aplicación	
3.1.1 Captura de pantalla de la guía de estilos de la aplicación	
3.1.2 Mapa de Navegabilidad (Captura de pantalla de Figma la navegabilidad)	
4 PLANES DE PRUEBAS.....	14
4.1 Plan de pruebas	
5 Pruebas de Adaptación Visual.....	17
6 Documentación.....	19
5.1 Manual de usuario	
5.2 Manuales Técnicos	
7.Scrip SQL	34
8 Conclusión final	36
9 Bibliografía.....	36

La aplicación que vamos a realizar trata sobre un calendario de tareas, donde el usuario podrá establecer tareas a cualquier hora del día.

Una vez llegada la hora, el usuario recibirá una notificación en su teléfono, avisando que es tiempo de realizar dicha tarea.

Cuando se inicie la aplicación, aparecerá una pantalla de inicio de sesión, donde los usuarios podrán acceder con una cuenta de correo, si el usuario no aparece registrado en la base de datos de la aplicación, este podrá registrarse pulsando en el botón '*Registrarse*', donde tendrá que rellenar un breve formulario con su correo electrónico, su nombre, apellido(s) y contraseña.

En esta pantalla también aparecerá una opción para iniciar en modo administrador.

Funciones del usuario

- Creación y edición de tareas, los usuarios podrán crear nuevas tareas directamente desde el calendario. Solo pulsando en un día específico, podrán ingresar los detalles de la tarea, fecha de inicio, hora exacta y una breve descripción
- Arrastrar y soltar, se añadirá la función de arrastrar y soltar para reorganizar las tareas de los usuarios.
- Notificaciones y recordatorios, los usuarios recibirán una notificación para tareas que están apunto de llegar a su inicio o a su fin. También podrán establecer recordatorios personalizados para cada tarea.
- Colores y etiquetas personalizadas, los usuarios podrán asignar colores y etiquetas a sus tareas para una identificación más rápida y sencilla.
- Vista detallada de las tareas, al pulsar sobre una tarea, el usuario podrá ver detalles adicionales, ya sean notas, subtareas...

Opciones de personalización de usuario.

El usuario podrá cambiar el tema y fondo de pantalla, para ello se ofrecerá una variedad de temas y fondos de pantalla para que los usuarios puedan adaptar la apariencia de la aplicación a sus preferencias personales.

También podrán establecer sus preferencias de notificación, incluyendo la frecuencia y el tipo de recordatorios que deseen recibir.

Funciones del administrador

El administrador poseerá todas las funciones del usuario, además podrá eliminar cuentas a nivel de la base de datos.

Tras iniciar sesión, el usuario podrá ver una pantalla con un calendario, dividido en días, al pulsar en un día, se abrirá un Widget donde el usuario podrá ver todas las actividades que tiene previstas para ese día y podrá añadir nuevas tareas pulsando en el botón de añadir.

En esta pantalla también aparecerá un botón de ajustes, donde al pulsarlo, la aplicación nos redirige a otra pantalla donde veremos nuestros datos de usuario, donde podremos modificarlos pulsando en el botón de modifica y donde el usuario podrá elegir el tema y fondo de pantalla de la aplicación, además de establecer sus preferencias de notificación.

Para personalizar las preferencias de notificación, al usuario se le dará la opción de enviar la notificación de recordatorio 10 minutos antes, 5 minutos antes y 1 minuto antes, al igual que la notificación de finalización de la tarea que podrá enviarla 10 minutos antes de finalizar, 5 minutos antes, 1 minuto antes o simplemente desactivarla.

Stack Tecnológico

- Lenguaje de programación: Java
- Entorno de desarrollo integrado (IDE): Android Studio
- Sistema operativo móvil: Android
- Base de datos: MySQL
- Conector de base de datos: Volley

Este es el Stack tecnológico con el que llevaremos a cabo la aplicación, siempre que no necesitemos cambiar en función de las necesidades de la aplicación a medida que vayamos desarrollándola

Justificación del proyecto

He decidido hacer esta aplicación, ya que encuentro que las personas hoy en día se mantienen muy ocupadas y la mayoría de su tiempo lo malgastan en distracciones como ver televisión o estar con el teléfono.

Con esta aplicación queremos que las personas se organicen su día y que cuando reciban nuestra notificación, paren cualquier actividad para hacer la tarea que tenían prevista para esa hora.

Esta aplicación nace como respuesta a la creciente necesidad de una aplicación de gestión de tareas integrada con un calendario que ofrezca esa simplicidad y funcionalidades avanzadas que piden los usuarios. Mientras que otras aplicaciones del mercado ofrecen muchas funciones similares, TaskMaster se distingue por su enfoque intuitivo y su capacidad de adaptación a las necesidades del usuario.

Alcance del proyecto

TaskMaster tiene como objetivo proporcionar a los usuarios una herramienta eficiente y intuitiva de usar para organizar y llevar un seguimiento de sus tareas diarias. El alcance de este proyecto incluye las siguientes funcionalidades:

- Registro de Usuarios

Los usuarios podrán registrarse utilizando su correo electrónico y una contraseña.

- Creación y Edición de tareas:

Los usuarios podrán crear nuevas tareas, asignándoles un nombre y una descripción detallada. Se permitirá editar tareas ya existentes para modificar detalles como la fecha o la descripción.

- Asignación de Fechas

Cada tarea tendrá una fecha de vencimiento para recordar al usuario de completarla a tiempo.

- Vista de calendario interactivo:

El usuario podrá ver sus tareas en un calendario, permitiendo una representación organizada de sus actividades.

- Notificaciones y recordatorios:

Se enviarán notificaciones para recordar al usuario de tareas a punto de comenzar o de finalizar. Los usuarios podrán modificar la configuración de las notificaciones según sus preferencias.

- Categorización y etiquetado de tareas:

Los usuarios podrán organizar sus tareas utilizando diferentes colores o etiquetas personalizadas. Esto permite al usuario una clasificación más eficiente.

- Almacenamiento en la nube

Las tareas de los usuarios tendrán almacenamiento en la nube gracias a la API de Google Calendar.

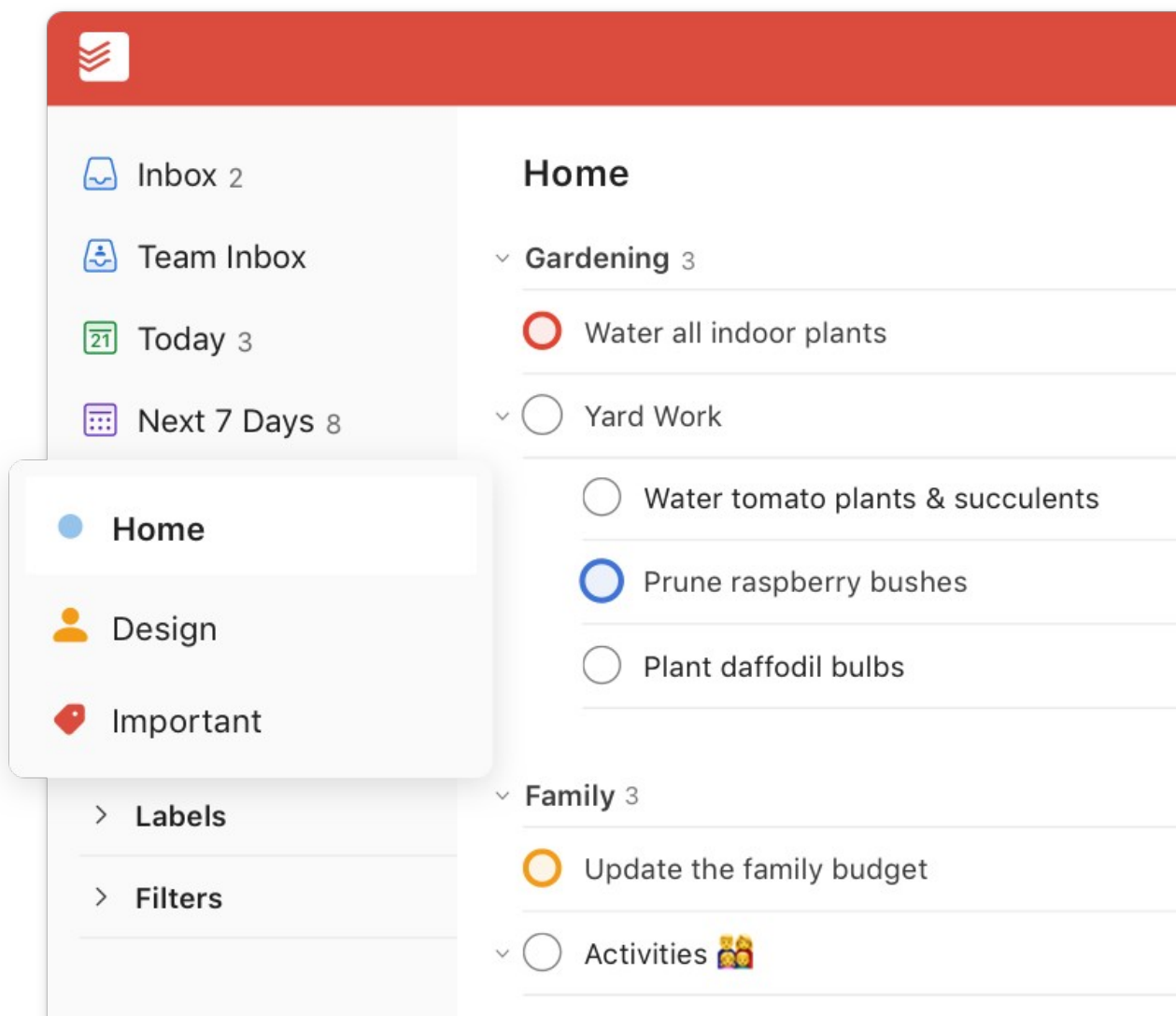
Valoración de Alternativas en el mercado

Existen aplicaciones en el mercado con un final parecido a este proyecto como puede ser la aplicación de Todoist o Asana.

Nuestra aplicación se diferenciará a las demás aplicaciones del mercado en su enfoque intuitivo y su capacidad para adaptarse a las necesidades del mercado.

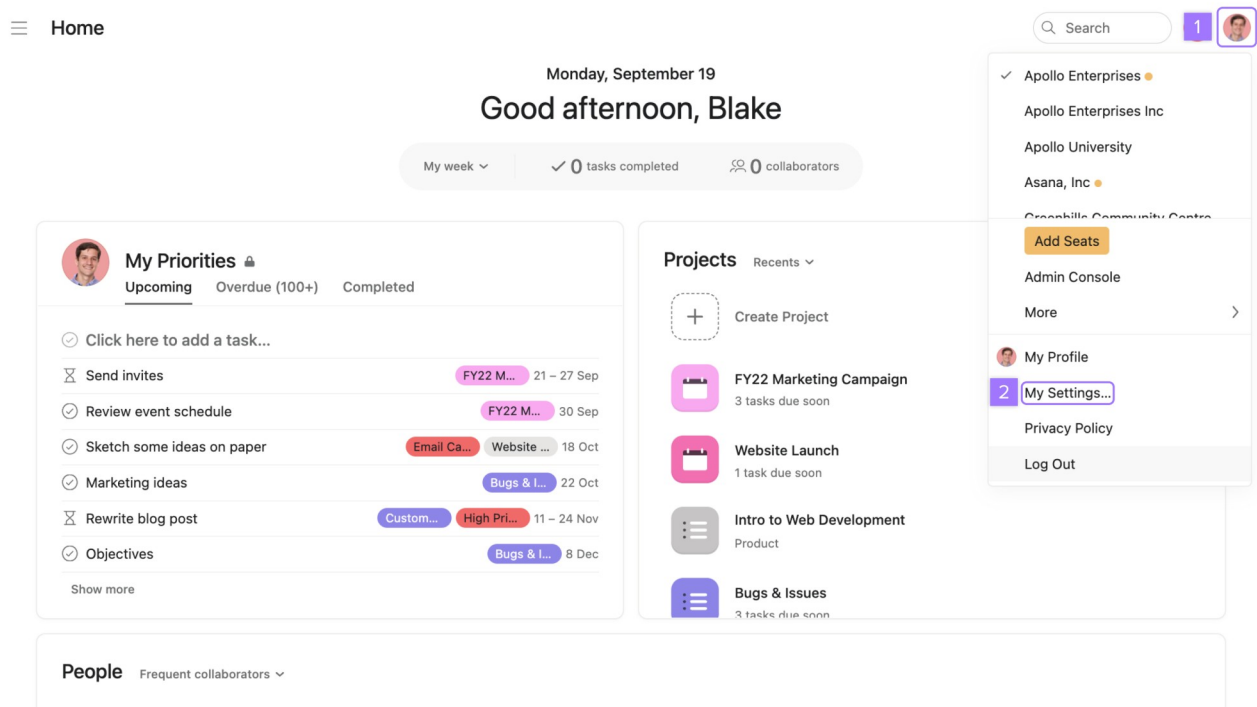
Todoist

Vemos como la aplicación de Todoist tiene una interfaz bastante intuitiva, aun así, nuestra aplicación intentará ser mucho mas intuitiva para aquellas personas que no están tan familiarizadas con las aplicaciones móviles.



Asana

La aplicación de Asana también tiene una interfaz bastante intuitiva, y algo bastante bueno es su forma de mostrar todas las tareas pendientes a lo largo del día que tiene el usuario, esta aplicación puede ser potencialmente mejor para aquellos usuarios que decidan organizar su día completo, en cambio nuestra aplicación se centra en tareas puntuales como puede ser una cita.

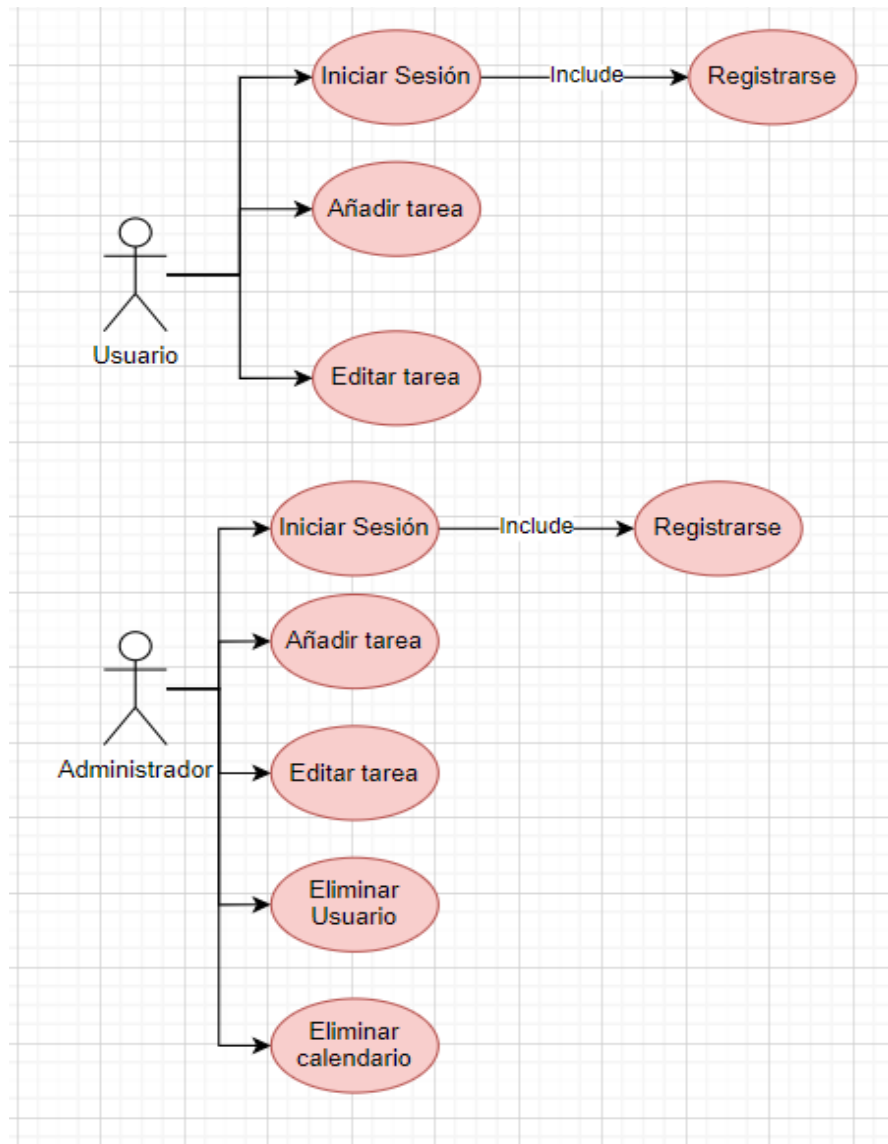


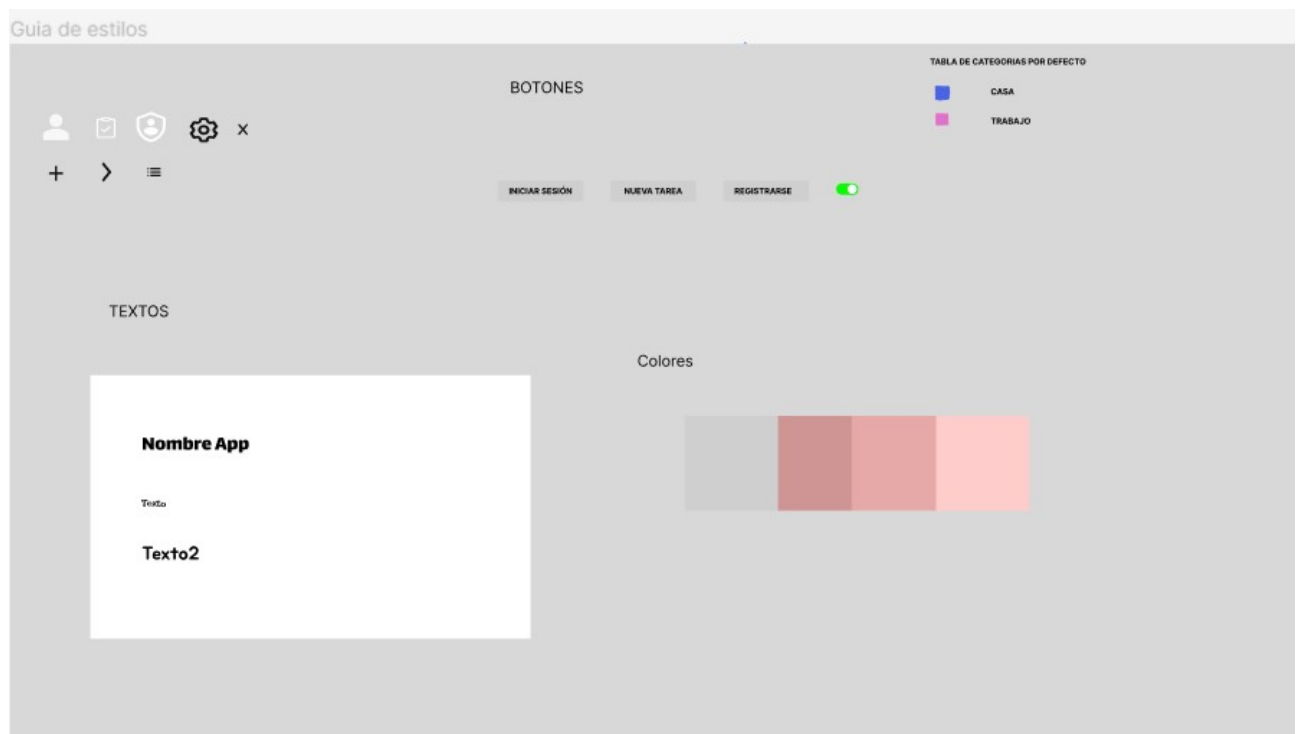
Casos de uso

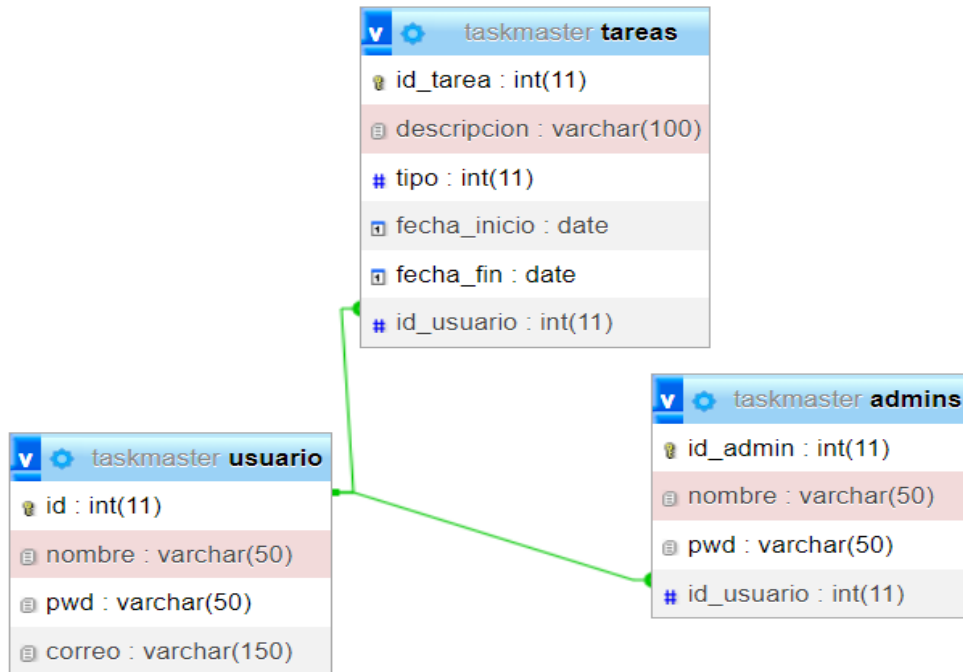
CU-01	Iniciar Sesión		
Versión	1.0		
Dependencias	Gestionar usuarios Validación de datos		
Precondición	El usuario ha introducido su correo y contraseña en el formulario de inicio de sesión		
Descripción	La aplicación deberá comprobar si el usuario esta registrado en la base de datos		
Secuencia normal	Paso	Acción	
	1	El usuario escribe su correo	
	2	El usuario escribe su contraseña	
	3	El usuario pulsa en iniciar sesión	
Postcondición	El usuario accede a la aplicación con éxito		
Excepciones	Paso	Acción	
	1	Si el usuario no esta registrado	
		E.1	El sistema informa de que no está registrada la cuenta

CU-02	Añadir tarea		
Versión	1.0		
Dependencias	Gestión de usuarios Gestión de calendario Gestión de tareas		
Precondición	El usuario esta registrado y ha pulsado en una fecha.		
Descripción	La aplicación deberá almacenar la tarea.		
Secuencia normal	Paso	Acción	
	1	El usuario inicia sesión	
	2	El usuario pulsa sobre una fecha	
	3	El usuario pulsa el botón de añadir tarea	
	4	El usuario añade la tarea	
Postcondición	La tarea aparece en el calendario tal y como el usuario la ha añadido.		
Excepciones	Paso	Acción	
	1	La datos de la tarea son erróneos	
		E.1	El sistema informa al usuario de que algunos datos introducidos son erróneos.

CU-03	Editar tarea		
Versión	1.0		
Dependencias	Gestión usuarios Gestión calendario Gestión tareas		
Precondición	El usuario esta registrado y pulsa en editar tarea		
Descripción	Al editar la tarea, esta debe aparecer con los nuevos datos en el calendario		
Secuencia normal	Paso	Acción	
	1	El usuario inicia sesión	
	2	El usuario pulsa sobre una fecha	
	3	El usuario pulsa en una tarea	
	4	El usuario pulsa en editar tarea	
	5	El usuario modifica los datos de la tarea	
Postcondición	La tarea aparece en el calendario con los nuevos datos.		
Excepciones	Paso	Acción	
	1	La tarea no se modifica	
		E.1	El sistema informa al usuario de que ha ocurrido un error al modificar dicha tarea







4 PLANES DE PRUEBAS

4.1 Plan de pruebas

Comprobar inicio sesión	Código del CP	PP001
	Prueba de despliegue	Sí
Descripción: Comprobar si al pulsar iniciar sesión con un usuario ya registrado se accede a la aplicación		
Prerrequisitos: Botón iniciar sesión Textbox para el usuario Textbox para la contraseña		
Pasos: Iniciar la aplicación Poner el usuario Poner la contraseña Pulsar el botón de iniciar sesión		
Resultado esperado El usuario accede a la aplicación		


Comprobar registrar sesión	Código del CP	PP002
	Prueba de despliegue	Sí
Descripción: Comprobar si al registrar un usuario se almacena en la base de datos		
Prerrequisitos: Textbox usuario Textbox contraseña Botón registrase		
Pasos: Iniciar la aplicación Introducir usuario Introducir contraseña Pulsar registrarse		
Resultado esperado El sistema debe enviar una alerta sobre que el usuario se ha creado correctamente		

Comprobar cerrar sesión	Código del CP	PP003
	Prueba de despliegue	Sí
Descripción: Comprobar si al pulsar en cerrar sesión se cierra la sesión		
Prerrequisitos: Icono usuario Botón cerrar sesión		
Pasos: Iniciar la aplicación Introducir usuario Introducir contraseña Pulsar iniciar Pulsar icono usuario Pulsar cerrar sesión		
Resultado esperado El sistema debe volver a la pantalla de login		

Pruebas de Adaptación

Vista de diferentes pantallas en Tablet Pixel C (1280 x 900 dp, xhdpi)

REGISTRAR
USUARIO



Introduce un ID de usuario (000)


Introduce un nombre de usuario

Introduce una clave

Introduce un correo

REGISTRARSE

This is a registration screen with a blue gradient background. It features a title 'REGISTRAR USUARIO' at the top, a user icon, and four input fields for user ID, username, password, and email. A green 'REGISTRARSE' button is at the bottom.

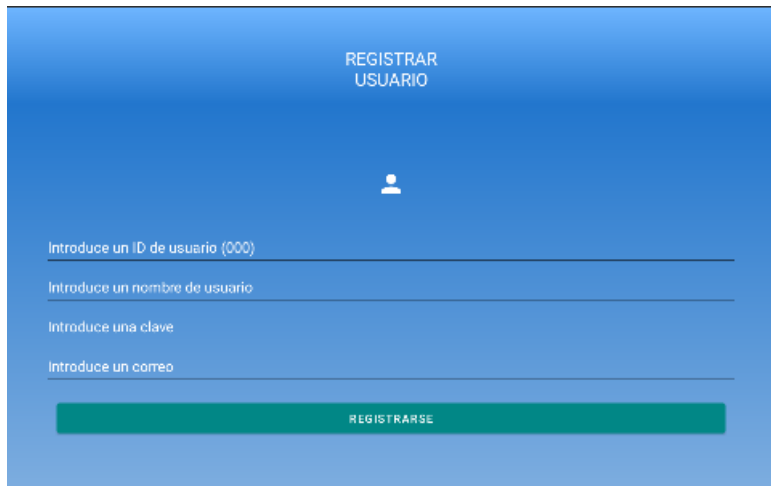


INICIAR

REGISTRAR

This is a login screen with an orange gradient background. It features a calendar icon, two input fields, and two buttons: 'INICIAR' and 'REGISTRAR'.

Vista de diferentes pantallas en Tablet Nexus 7 (600 x 960 dp, xhdpi)



REGISTRAR
USUARIO



Introduce un ID de usuario (000)

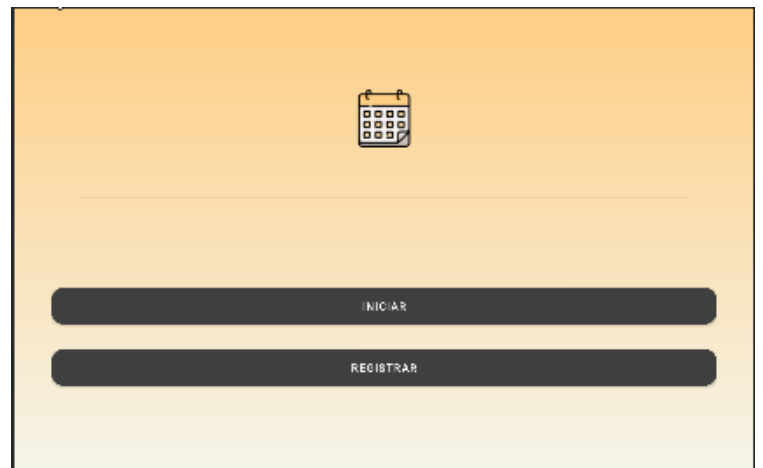
Introduce un nombre de usuario


Introduce una clave

Introduce un correo

REGISTRARSE

This is a user registration screen with a blue gradient background. It features a title 'REGISTRAR USUARIO' at the top, a user icon, and four input fields for user ID, username, password, and email. A green 'REGISTRARSE' button is at the bottom.





INICIAR

REGISTRAR

This is a login and registration screen with an orange-to-yellow gradient background. It features a calendar icon at the top, a horizontal line, and two dark grey buttons labeled 'INICIAR' and 'REGISTRAR' at the bottom.

Manual de usuario

El objetivo de este manual es otorgar ayuda a los usuarios para que puedan aprender sobre la utilización de esta App.

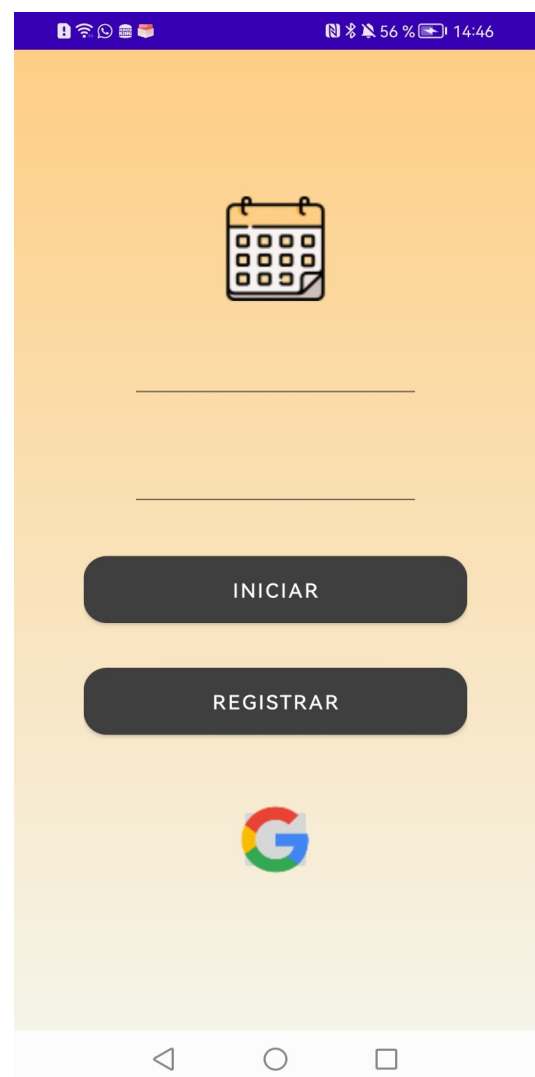
Para poder usar esta aplicación se necesita cualquier dispositivo con el que acceder a Internet.

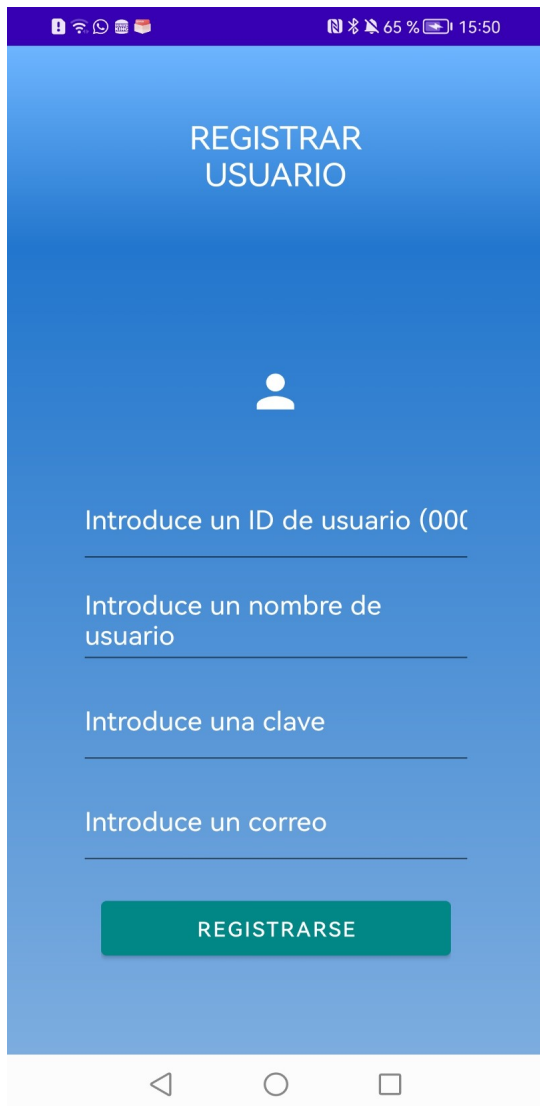
Las funcionalidades que ofrece la aplicación son las siguientes:

En primer lugar la aplicación consta de una pantalla de login que permite tanto crear como iniciar sesión con una cuenta ya existente.

Simplemente se tendrá que introducir el nombre de usuario y la contraseña de este

Cada usuario podrá elegir al iniciar sesión un correo electrónico, en el es donde se guardaran los eventos creados en Google Calendar.

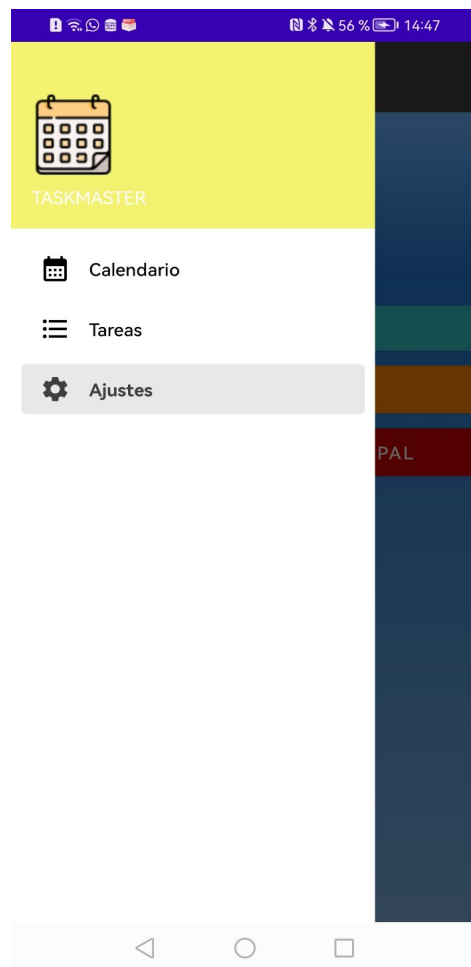




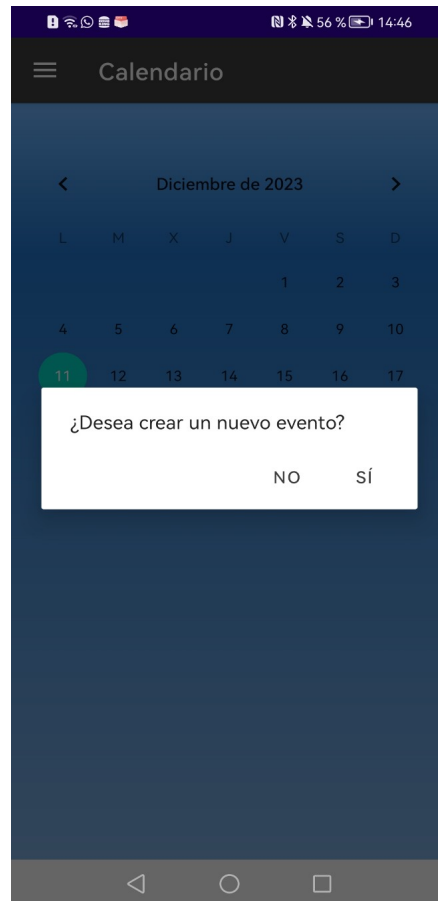
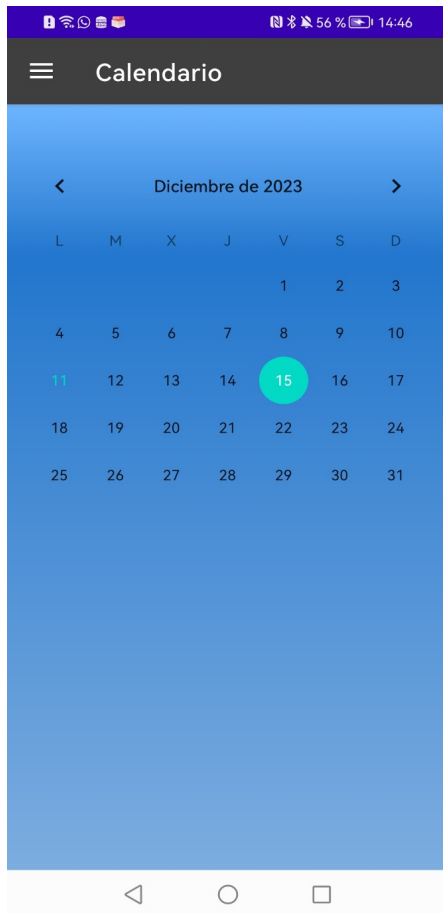
The screenshot shows a mobile application interface for user registration. At the top, the status bar displays various icons and the time 15:50. The app's title bar is blue with the text 'REGISTRAR USUARIO' in white. Below the title is a white user icon. The main content area has a blue gradient background and contains four input fields with labels: 'Introduce un ID de usuario (000', 'Introduce un nombre de usuario', 'Introduce una clave', and 'Introduce un correo'. Each field has a horizontal line for text entry. At the bottom of the form is a green button with the text 'REGISTRARSE' in white. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Al pulsar el botón registrar, la app le llevará a este formulario donde podrá dar de alta a un usuario. Para dar de alta al usuario deberá introducir un ID, siendo un valor número de 0 a 999, un nombre de usuario y una contraseña con la que se accederá posteriormente

Esta pantalla se muestra tras iniciar sesión, en ella vemos que tenemos un menú desplegable en la parte superior izquierda, este menú se divide en 3.



La primera pantalla, es un calendario, donde al pulsar sobre un día concreto, nos aparecerá un mensaje preguntando si queremos añadir un evento para ese día.
Si pulsamos en el 'Sí' pasaremos a la siguiente pantalla



Una vez pulsado el botón de ‘Sí’ la aplicación nos pasará a un tipo de formulario donde introduciremos los datos del evento que queremos crear.

Entre estos datos encontramos con un Switch que podemos activar para decir que el evento durará todo el día.

También introduciremos la fecha de inicio con su respectiva hora, al igual que su fecha final, la fecha inicial por defecto será el día seleccionado en el calendario anteriormente.

También podemos añadir el tipo de evento manejándolos por colores.

Por último podemos elegir cuando queremos que se nos mande una notificación de que el evento va a empezar.

The screenshot shows a form titled "Nueva tarea" on an orange background. It contains the following fields:

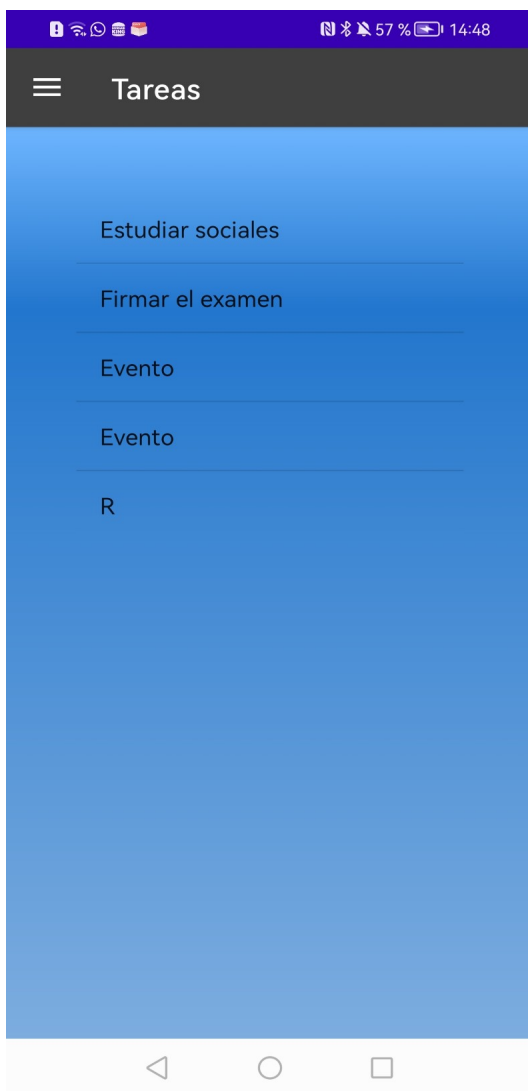
- Titulo**: A text input field with the placeholder "titulo".
- Descripcion**: A text input field with the placeholder "Descripcion".
- Fecha Inicio**: A date selection field with a calendar icon and the placeholder "Fecha inicial".
- Hora Inicio**: A time selection field with a clock icon and the placeholder "Hora inicial".

The screenshot shows the configuration part of the task form on an orange background. It contains the following sections:

- Hora Fin**: A date selection field with a calendar icon and the placeholder "Hora final".
- Tipo**: A section with five colored circles (grey, blue, green, yellow, red) for selecting the event type.
- Notificacion**: A section with three radio button options: "30 min antes", "1 hora antes", and "2 horas antes".
- GUARDAR TAREA**: A green button to save the task.

At the bottom, there is a white navigation bar with three icons: a back arrow, a circle, and a square.

En la segunda opción del menú, nos encontramos con la pantalla de tareas, esta pantalla nos refleja las siguientes tareas que tiene el usuario programadas, al pulsar sobre una de ellas, la aplicación nos mostrara los detalles de dicho evento seleccionado.



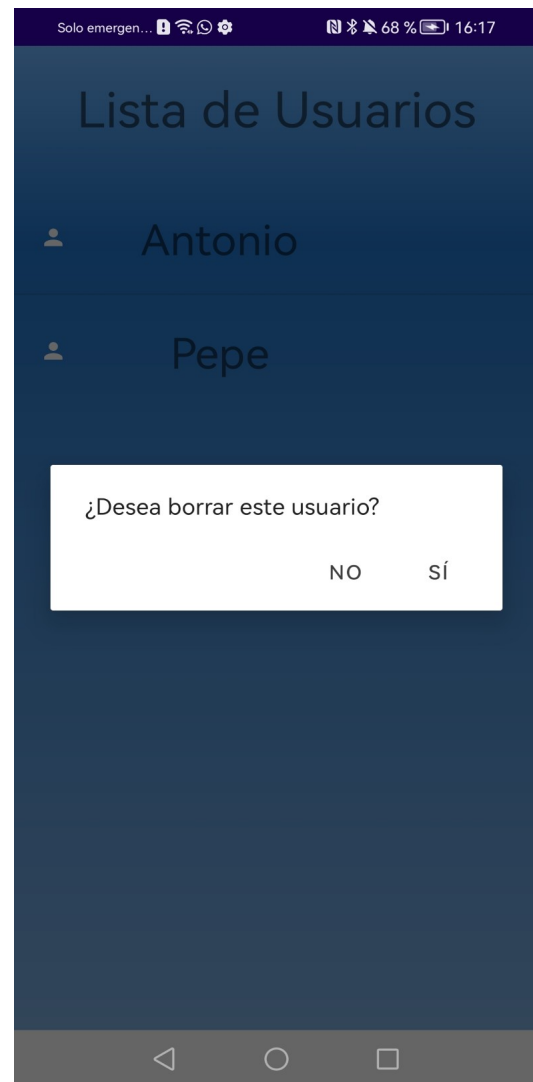
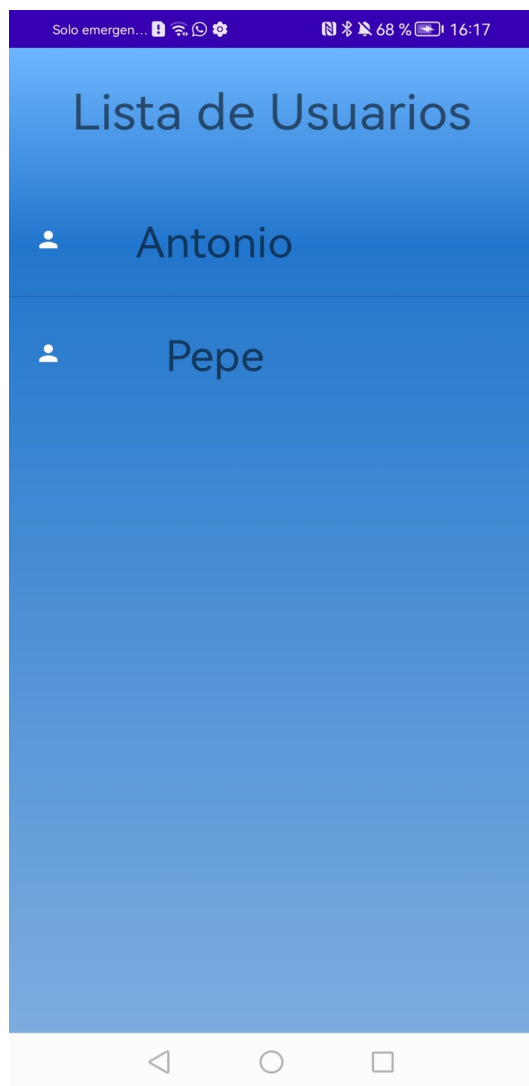
Por último, la ultima opción del menú es la pantalla de ajustes, en esta tendremos una opción para poner la aplicación en modo oscuro, cambiando así su interfaz.

El botón que aparece como CERRAR SESIÓN sirve para desloguearnos de la aplicación y volver a la pantalla de inicio de sesión. Si entramos en modo administrador, además aparecerán dos botones adicionales, uno para borrar usuario y otro para eliminar el calendario principal.



Al pulsar en el botón borrar usuario, la aplicación nos envía a una nueva pantalla donde vemos un listado de todos los usuarios guardados en la base de datos.

Al pulsar sobre uno, el sistema pregunta si estamos seguros de borrar dicho usuario y al pulsar en sí, este usuario será borrado de la base de datos.



Manual Técnico Conexión a Base de Datos desde Android Studio

Para desplegar la base de datos utilizada en este proyecto utilizaremos el sistema de gestión de base de datos llamado XAMPP.

Se trata de un software libre que nos permite hacer conexiones locales a una base de datos. A continuación explicaré los pasos detalladamente para instalar XAMPP en nuestro equipo

Paso 1. Descargar XAMPP.

Descarga XAMPP desde el sitio web oficial.

<https://www.apachefriends.org/index.html>

Haz clic en el botón de descarga para iniciar la descarga del instalador de XAMPP. El instalador es un archivo ejecutable (.exe).

Paso 2. Instalar XAMPP.

- Ejecuta el instalador.

Una vez descargado el archivo, ejecutalo y acepte los permisos de administrador para continuar.

- Seleccionar los componentes.

Durante la instalación de XAMPP se pedirá al usuario que elija los componentes que desea instalar. En nuestro caso instalaremos Apache y MySQL.



- Agregar ruta de instalación.

En este caso instalaremos XAMPP en la ruta por defecto que sería C:\xampp. Pero puedes cambiar esta ruta según tus necesidades.

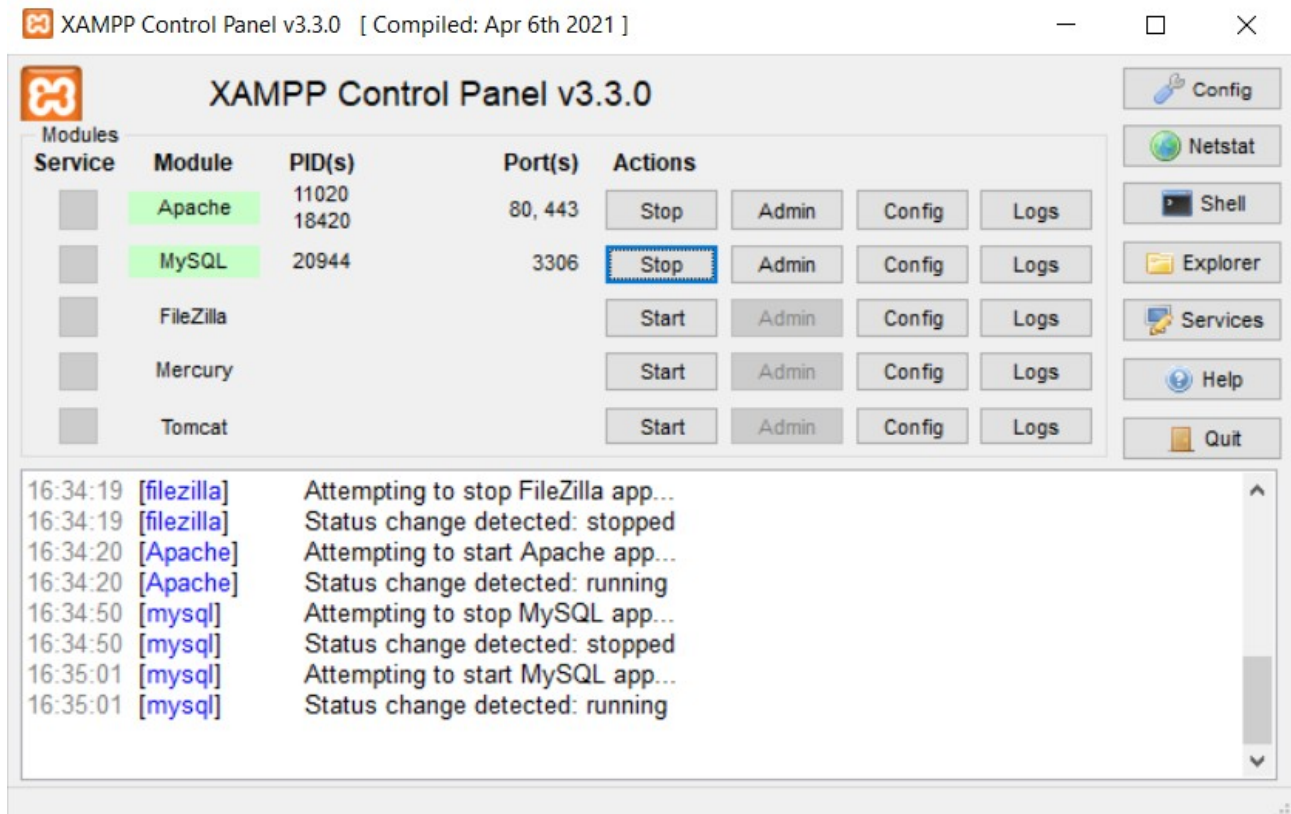
-Completar la instalación.

Pulsaremos en continuar y esperaremos a que se complete el proceso de instalación, este proceso puede llevar algún tiempo.

Paso 3. Iniciar los servidores

Una vez finalizada la instalación, se te preguntará si deseas iniciar los servicios de Apache y MySQL. Asegúrate de marcar esas opciones y haz clic en "Finish".

Accederemos a XAMPP Control Panel e iniciaremos los módulos Apache y MySQL.



Paso 4. Acceder a PHPMyAdmin

Por último, Abre tu navegador web y ve a <http://localhost/phpmyadmin/>. Aquí podrás administrar tus bases de datos MySQL utilizando la interfaz gráfica de PHPMyAdmin.

Para la conexión con la base de datos desde Android Studio, utilizaremos la biblioteca Volley.

Volley es una biblioteca HTTP que facilita y agiliza el uso de redes en apps para Android.

Como usar la biblioteca Volley

Para utilizar la biblioteca Volley en nuestra aplicación seguiremos los siguientes pasos.

Paso 1. Agregar dependencia Volley

En el archivo build.gradle (Module: app) añadiremos la siguiente dependencia.

```
implementation 'com.android.volley:volley:1.2.1'
```

Paso 2. Agregar permisos de Internet.

Abre tu archivo “AndroidManifest.xml” y agrega el siguiente permiso.

```
<uses-permission android:name="android.permission.INTERNET" />
```

Paso 3. Crear los métodos para la conexión

Para la conexión con la base de datos realizaremos un método GET a la URL específica de nuestra base de datos.

Para esto antes tendremos que crear unos archivos php en el directorio “htdocs” de XAMPP (C:\xampp\htdocs).

En este caso para iniciar sesión utilizaremos el siguiente archivo php.

```

iniciar.php
C: > xampp > htdocs > proyecto > iniciar.php
1  <?PHP
2  $hostname="localhost";
3  $database="taskmaster";
4  $username="root";
5  $password="";
6
7  $json=array();
8  if(isset($_GET["nombre"]) && isset($_GET["pwd"])){
9      $user=$_GET['nombre'];
10     $pwd=sha1($_GET['pwd']);
11
12
13     $conexion=mysqli_connect($hostname,$username,$password,$database);
14
15     $consulta="SELECT id, nombre, pwd FROM usuario WHERE nombre= '{$user}' AND pwd = '{$pwd}'";
16     $resultado=mysqli_query($conexion,$consulta);
17
18     if($consulta){
19
20         if($reg=mysqli_fetch_array($resultado)){
21             $json['datos'][]=$reg;
22         }
23         mysqli_close($conexion);
24         echo json_encode($json);
25     }
26
27
28
29     else{
30         $results["id"]='';
31         $results["nombre"]='';
32         $results["pwd"]='';
33         $json['datos'][]=$results;
34         echo json_encode($json);
35     }
36

```

En el parámetro \$hostname introduciremos el valor de “localhost”

En \$database introducimos el valor del nombre de nuestra base de datos creada en phpmyadmin, en este caso “taskmaster”.

El parámetro \$username sera por defecto “root”.

En este archivo comprobaremos que existe un usuario con la contraseña introducida en nuestra base de datos y nos devolverá un Json.

Para hacer la petición a este archivo introduciremos el siguiente código en nuestro Activity.

En primer lugar implementaremos las siguientes clases.

```

4 usages
public class Inicio extends AppCompatActivity implements View.OnClickListener, Response.Listener<JSONObject>, Response.ErrorListener{

```

Como segundo tendremos que instanciar los parámetros RequestQueue y JsonRequest en nuestra clase.

```

2 usages
private RequestQueue queue;
2 usages
private JsonRequest jrqr;
2 usages

```

Posteriormente en el método onCreate de la clase iniciaremos el parámetro “queue”

```
//Hacer consulta
queue = Volley.newRequestQueue(getApplicationContext());
```

Por
último

añadiremos los métodos onErrorResponse y onResponse

El método onErrorResponse hace referencia a cuando la petición tiene algún tipo de error ya sea que el usuario no exista en la base de datos o que la consulta este mal escrita.

El método onResponse hace referencia a que la consulta ha sido efectuada correctamente y el usuario existe en la base de datos, por lo que podremos pasar a otra pantalla.

Para hacer la petición utilizaremos la siguiente URL:

“<http://ipWifi/archivophp.php?nombre=USUARIO&pwd=CONTRASEÑA>”

donde ipWifi hace refencia a la IP de la red wifi a la que estamos conectados tanto en el dispositivo como en nuestro equipo.

Para acceder a esta IP podemos abrir la consola de Windows y ejecutar el comando “ipconfig”

En Usuario y Contraseña podemos recoger los parámetros de un EditText donde el usuario podrá escribir su usuario y contraseña.

```
1 usage
private void iniciarSesion() {
    String url = "http://192.168.1.136/proyecto/iniciar.php?nombre=" + usuario.getText().toString() + "&pwd=" + password.getText().toString();
    jrqr = new JsonObjectRequest(Request.Method.GET, url, jsonRequest: null, listener: this, errorListener: this);
    queue.add(jrqr);
}

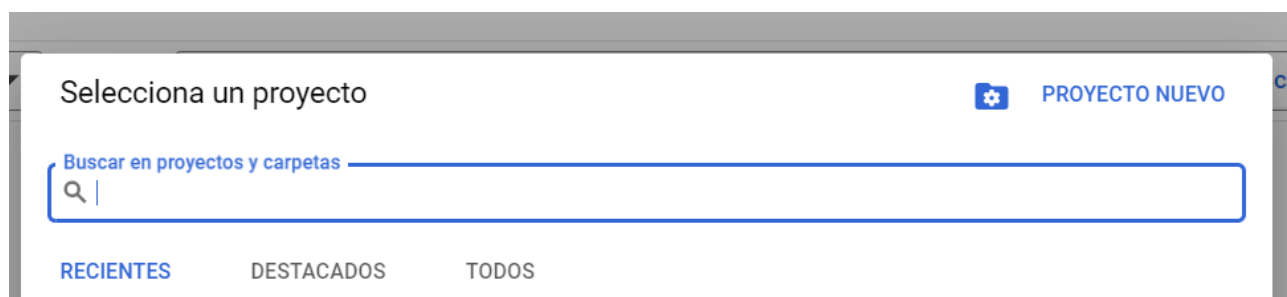
2 usages
@Override
public void onErrorResponse(VolleyError error) {
    System.out.println(error.toString());
    Toast.makeText(context: this, text: "User not found", Toast.LENGTH_SHORT).show();
}

@Override
public void onResponse(JSONObject response) {
    Intent i = new Intent(packageContext: this, Menu.class);
    this.startActivity(i);
}
```

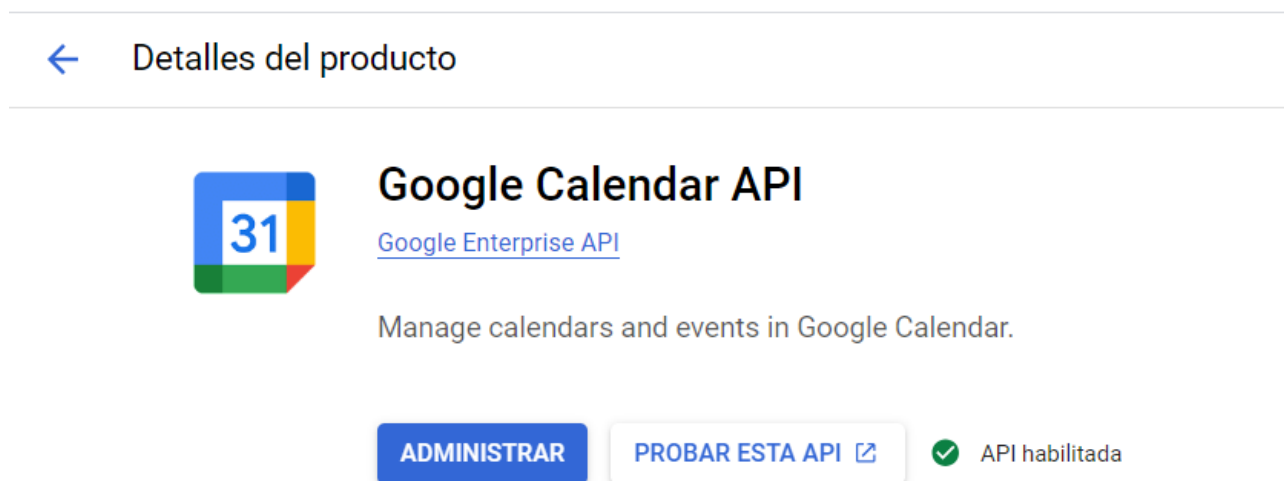

Manual Técnico Uso de la API de Google Calendar

Paso 1: Configuración en la Consola de Desarrolladores de Google

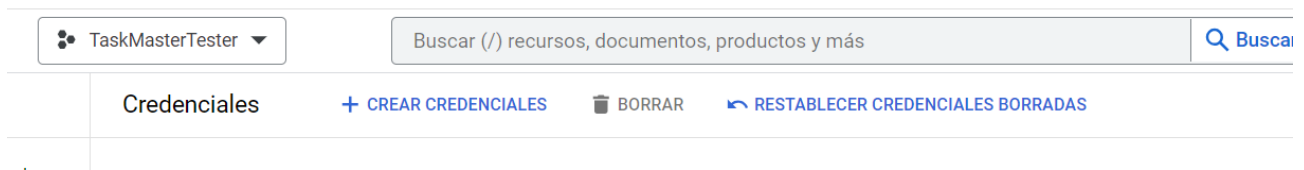
1. Ve a Google Cloud Console.
2. Crea un nuevo proyecto o selecciona uno existente.



3. En la biblioteca del proyecto, habilita la API de Google Calendar.



4. Configura las credenciales para tu proyecto. Selecciona “Crear credenciales” y elige “ID de cliente de OAuth” con tipo de aplicación “Android”. Especifica los detalles de la aplicación, como nombre del paquete y el certificado SHA-1.



5. Descarga el archivo JSON con las credenciales.

Paso 2: Configuración del proyecto de Android Studio

1. Agrega las dependencias necesarias en tu archivo 'build.gradle' (Module:App)

```
implementation 'com.google.api-client:google-api-client-android:1.31.5'
implementation 'com.google.oauth-client:google-oauth-client-jetty:1.31.5'
implementation 'com.google.apis:google-api-services-calendar:v3-rev305-1.25.0'
implementation 'com.google.http-client:google-http-client-gson:1.31.5'
implementation 'com.google.android.gms:play-services-auth:19.2.0'
```

2. Agrega a tu Android Manifest los siguientes permisos

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
```

3. Configuración del Cliente de Google Calendar

En tu actividad principal, necesitas configurar el cliente de Google Calendar. Este cliente se encargará de la autenticación y de realizar las operaciones en la API. Aquí, utilizaras un archivo JSON que descargaste en la Consola de Desarrolladores.

```
// Configuración de Google Sign-In
GoogleSignInOptions signInOptions = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestScopes(new Scope(CalendarScopes.CALENDAR))
    .requestEmail()
    .build();

// Crear el cliente de inicio de sesión de Google
final GoogleSignInClient googleSignInClient = GoogleSignIn.getClient( activity: this, signInOptions);

startActivityForResult(googleSignInClient.getSignInIntent(), REQUEST_AUTHORIZATION);
```

```
1 usage
private void handleSignInResult(Intent data) {
    GoogleSignIn.getSignedInAccountFromIntent(data)
        .addOnSuccessListener(googleSignInAccount -> {
        // Obtener las credenciales de la cuenta de Google seleccionada
        credential = GoogleAccountCredential.usingOAuth2(
            getApplicationContext(), Arrays.asList(CalendarScopes.CALENDAR));
        credential.setSelectedAccount(googleSignInAccount.getAccount());
    });
}
```

4. Añadir Eventos en Google Calendar

```
try {  
  
    JsonFactory jsonFactory = GsonFactory.getDefaultInstance();  
    HttpTransport httpTransport = GoogleNetHttpTransport.newTrustedTransport();  
  
    Calendar service = new Calendar.Builder(httpTransport, jsonFactory, Menu.credential)  
        .setApplicationName("TaskMaster")  
        .build();
```

```
// Crear un evento  
Event event = new Event()  
    .setSummary(txtTitulo.getText().toString())  
    .setDescription(txtDescripcion.getText().toString())  
    .setColorId(color)  
    ;
```

```
event = service.events().insert(calendarId, event).execute();
```

De esta manera estaríamos creando un evento, a este evento podemos añadirle una fecha de inicio y una fecha final.

SCRIP SQL BASE DE DATOS

```
-- phpMyAdmin SQL Dump
-- version 5.2.0
-- https://www.phpmyadmin.net/
--
-- Servidor: 127.0.0.1
-- Tiempo de generación: 11-12-2023 a las 16:26:34
-- Versión del servidor: 10.4.25-MariaDB
-- Versión de PHP: 8.1.10

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de datos: `taskmaster`
--

--
-- Estructura de tabla para la tabla `admins`
--

CREATE TABLE `admins` (
  `id_admin` int(11) NOT NULL,
  `nombre` varchar(50) NOT NULL,
  `pwd` varchar(50) NOT NULL,
  `id_usuario` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Estructura de tabla para la tabla `tareas`
--

CREATE TABLE `tareas` (
  `id_tarea` int(11) NOT NULL,
  `descripcion` varchar(100) NOT NULL,
  `tipo` int(11) NOT NULL,
  `fecha_inicio` date NOT NULL,
  `fecha_fin` date NOT NULL,
  `id_usuario` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Estructura de tabla para la tabla `usuario`
--

CREATE TABLE `usuario` (
  `id` int(11) NOT NULL,
  `nombre` varchar(50) NOT NULL,
  `pwd` varchar(50) NOT NULL,
  `correo` varchar(150) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Volcado de datos para la tabla `usuario`
--

INSERT INTO `usuario` (`id`, `nombre`, `pwd`, `correo`) VALUES
(1, 'Antonio', 'A08F08BAC39AEAE6C9580ADE7AA8387B5A0E7428', ''),
(2, 'A', 'a', ''),
(3, 'Re', 'c387c982a132d05cbd5f88840aef2c8157740049', ''),
(23, 'Aro', 'c8398ae2e4edf10206348b780f79d9877c7941a1', ''),
(25, 'q', 'q', 'antonioreyesortega03@gmail.com'),
(42, 'Prueba', '711383a59fda05336fd2ccf70c8059d1523eb41a', ''),
(66, 'R', '4dc7c9ec434ed06502767136789763ec11d2c4b7', ''),
(234, 'Tu', 'a3da4c6307d230e1f1c8ad62e00d05ff1f1f5b52', 'Areyort827@gmail.com');

--
-- Índices para tablas volcadas
--

--
-- Indices de la tabla `admins`
--
ALTER TABLE `admins`
  ADD PRIMARY KEY (`id_admin`),
  ADD KEY `id_usuario` (`id_usuario`);

--
-- Indices de la tabla `tareas`
--
ALTER TABLE `tareas`
  ADD PRIMARY KEY (`id_tarea`),
  ADD KEY `id_usuario` (`id_usuario`);

--
-- Indices de la tabla `usuario`
--
ALTER TABLE `usuario`
  ADD PRIMARY KEY (`id`);

--
-- Restricciones para tablas volcadas
--

--
-- Filtros para la tabla `admins`
--
ALTER TABLE `admins`
  ADD CONSTRAINT `admins_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES `usuario` (`id`);

--
-- Filtros para la tabla `tareas`
--
ALTER TABLE `tareas`
  ADD CONSTRAINT `tareas_ibfk_1` FOREIGN KEY (`id_usuario`) REFERENCES `usuario` (`id`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Conclusión final

Como conclusión final de este proyecto, me gustaría aportar que cuando comencé con esta aplicación, no tenía muchas esperanza de poder conseguir mi objetivo final, pero tras esforzarme por hacer de esta aplicación lo mejor posible, me di cuenta que investigando en los sitios correctos, se puede obtener toda la información que necesitas y gracias a esto pude conseguir sacar adelante este proyecto.

Con muchas mejoras posibles, pero con el tiempo de realización del proyecto no pude mejorar la aplicación a grandes escalas.

A nivel personal he quedado bastante satisfecho con el resultado, he aprendido muchas cosas de las que no tenía ni idea y esto es lo importante de realizar estos proyectos, el aprendizaje.

Bibliografía

<https://developers.google.com/calendar?hl=es-419>

<https://developers.google.com/workspace/guides/create-project?hl=es-419>

<https://www.youtube.com/watch?v=T-f-3RxAfZ0>

<https://www.youtube.com/watch?v=x66x9tltEX0&t=351s>