

Documentación Técnica

Doc Ambiente de Desarrollo

Android Studio 1.1.0
Build #AI-135.1740770, built on February 18, 2015
JRE: 1.7.0_80-b15 amd64
JVM: Java HotSpot(TM) 64-Bit Server VM by Oracle Corporation

SDK MANAGER

Packages			
SDK Path: /home/fernando/Android/Sdk			
Packages			
Name	API	Rev.	Status
Tools			
<input checked="" type="checkbox"/> Android SDK Tools		24.1.2	Update available: rev. 24.3.1
<input type="checkbox"/> Android SDK Platform-tools		22	Installed
<input type="checkbox"/> Android SDK Build-tools		22.0.1	Installed
<input type="checkbox"/> Android SDK Build-tools		21.1.2	Installed
<input type="checkbox"/> Android SDK Build-tools		20	Not installed
<input type="checkbox"/> Android SDK Build-tools		19.1	Not installed
Tools (Preview Channel)			
<input type="checkbox"/> Android SDK Build-tools		23 rc2	Not installed
Android M (API 22, MNC preview)			
<input type="checkbox"/> Documentation for Android 'MNC' Preview	MNC	1	Not installed
<input checked="" type="checkbox"/> SDK Platform Android M Preview	MNC	1	Not installed
<input type="checkbox"/> Samples for SDK API MNC Preview	MNC	2	Not installed
<input type="checkbox"/> Android TV Intel x86 Atom System Image	MNC	1	Not installed
<input type="checkbox"/> ARM 64 v8a System Image	MNC	1	Not installed
<input type="checkbox"/> ARM EABI v7a System Image	MNC	1	Not installed
<input type="checkbox"/> Intel x86 Atom_64 System Image	MNC	1	Not installed
<input type="checkbox"/> Intel x86 Atom System Image	MNC	1	Not installed
<input type="checkbox"/> MIPS System Image	MNC	1	Not installed
<input type="checkbox"/> Sources for Android 'MNC' Preview SDK	MNC	1	Not installed
Android 5.1.1 (API 22)			
<input type="checkbox"/> Documentation for Android SDK	22	1	Installed
<input type="checkbox"/> SDK Platform	22	2	Installed
<input checked="" type="checkbox"/> Samples for SDK	22	5	Update available: rev. 6
<input type="checkbox"/> Android TV ARM EABI v7a System Image	22	1	Installed
<input type="checkbox"/> Android TV Intel x86 Atom System Image	22	1	Installed
<input type="checkbox"/> Android Wear ARM EABI v7a System Image	22	2	Not installed
<input type="checkbox"/> Android Wear Intel x86 Atom System Image	22	2	Not installed
<input type="checkbox"/> ARM EABI v7a System Image	22	1	Installed
<input type="checkbox"/> Intel x86 Atom_64 System Image	22	1	Installed
<input checked="" type="checkbox"/> Intel x86 Atom System Image	22	1	Installed
<input type="checkbox"/> Google APIs	22	1	Installed
<input type="checkbox"/> Google APIs ARM EABI v7a System Image	22	1	Installed
<input type="checkbox"/> Google APIs Intel x86 Atom_64 System Image	22	1	Installed
<input type="checkbox"/> Google APIs Intel x86 Atom System Image	22	1	Installed
<input type="checkbox"/> Sources for Android SDK	22	1	Installed

Doc de librerías y compilación

Ninguna librería adicional.

Agregar permisos en AndroidManifest.xml:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

para acceso al GPS del device y para conexión HTTP.

Doc de Arquitectura/Diseño

La aplicación tiene siete actividades que le sirven para cumplir con las funcionalidades propuestas. Se encuentran en la carpeta Activities.

Tiene la carpeta Threads, que contiene todas las clases que tiran hilos para ejecutar tareas en background como las llamadas al servidor y el refreshing de la pantalla de chats y la de usuarios.

La carpeta Entities contiene las clases propias del modelo de la aplicación: User, Message, Chat.

La carpeta Common contiene clases con funcionalidades comunes que sirven a todas las actividades de la aplicación:

- Constants que sirve para guardar datos durante la ejecución. La comunicación entre las actividades es simplificada con el uso de esta clase "PUBLICA".
- ChatEditor: se encarga de renderizar un Chat en ChatActivity completo y de agregarle nuevos mensajes.
- Log: se encarga de dejar registro de todos los eventos que suceden en la app.
- MyLocationListener: se encarga de obtener la localización a partir del GPS del celular.
- Packager: se encarga de empaquetar y desempaquetar los paquetes enviados y recibidos del servidor en base 64.

La carpeta Mocks contiene el Server mock que sirve para testear la aplicación, el cual replica todas las funcionalidades del servidor real.

Doc de Clases

Diagrama de clases de la estructura. (Voy a agregar)

Descripción de cada una de las clases: (pongo la descripción de una sola a modo de ejemplo)

LogInActivity

SignUpActivity

Es la actividad donde se puede registrar un nuevo user.

El usuario debe completar los campos que Username y Password y hacer touch en el botón Confirm.

- En caso de que las credenciales sean correctas se pasará a la pantalla de configuración del perfil ConfigurationActivity.
- En caso de que no lo sean se mostrará un Toast indicando el problema.
- Al hacer touch en el botón Confirm se hace un request al servidor. En caso de haber

problemas de conexión se mostrara un Toast informando del error.
Tambien se puede hacer touch en el boton BackToLogin para volver a la pantalla LoginActivity y acceder a la aplicación con un usuario ya registrado.

ChatsHallActivity

...

ChatEditor

Constants

...

Chat

Message

User

ConfigPostAsyncTask

CurrentChatsPostAsyncTask

...

Doc de Código

copy paste del codigo fuente ?

Doc de REST API

Del lado del cliente se usa Httppost y Httpget en una simple llamada en un hilo aparte que no bloquee el UI thread para no quitarle performance a la app ni bloquearla a la espera del response del server.

Se hacen paquetes (strings con todos los datos que se quieran mandar en formato Json), se encodean en base64 y se antepone "package=". Luego esto es lo qe se envia al servidor en los requests.

Para los posts uso la estructura propia de los posts que provee Android.

POST

```
HttpClient httpClient = new DefaultHttpClient();
```

```
HttpResponse response;
```

```
HttpPost httpPost = new HttpPost(url);
```

```
List<NameValuePair> nameValuePairs = new ArrayList<>(1);
```

```
nameValuePairs.add(new BasicNameValuePair("package", package_));
```

```
httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
```

Execute HTTP Post Request

```
response = httpClient.execute(httpPost);
```

Para los gets armo la url a mano porque solo mando la variable "paquete".

GET

```
HttpClient httpClient = new DefaultHttpClient();
```

```
HttpResponse response;
```

```
url = url.concat("?"+ package_);  
response = httpClient.execute(new HttpGet(url));
```

donde package_ es el json encodeado en base 64.

Doc de Pruebas

Las pruebas muestran la logica detras de la aplicación y el manejo correcto de los datos, haciendo uso de los layouts e iniciando las activities.

Esta forma permite ver que el modelo funciona correctamente y a la vez comprueba el correcto flow de la app y los layouts.

Se testea cada activity por separado (carpeta activities), se hacen pruebas de integracion que incumben a varios recursos (carpeta integration), y se hacen tambien tests de comportamiento de la app completa, como test de persistencia en caso de cerrarla y abrirla (carpeta app).

Codigo fuente de los tests ?

Manual de Usuario del Cliente

Intro general de la App

MensajerO es la aplicación Android de hoy para chatear con tus amigos. Es muy facil de usar. Sin complicaciones.

Features incluídos

- Podes ver a todos los usuarios conectados y chatear con ellos.
- Podes setear tu estado y tu foto de perfil.
- Si tu estado es offline puedes enviar y recibir mensajes pero no vas a aparecer en la lista de usuarios para que te vean los demas.
- Podes mandar un mensaje de difusion a todos los usuarios.
- Podes ver donde se encuentran tus amigos y cual fue la ultima vez que se conectaron.

Instalación y configuración

1. En tu telefono Android: Ajustes → Seguridad → Origenes Desconocidos (Permitir la instalación de aplicaciones desde orígenes distintos de PlayStore). Tilda la opción.
2. Coloca el archivo [mensajerO.apk](#) en una carpeta que puedas encontrar despues con el file browser de tu cel
3. Busca el archivo en tu cel. Hacele un touch. Acepta la instalacion.
4. Listo! Te comunicas y disfrutas la solides de nuestra app.

Cómo usarla paso a paso

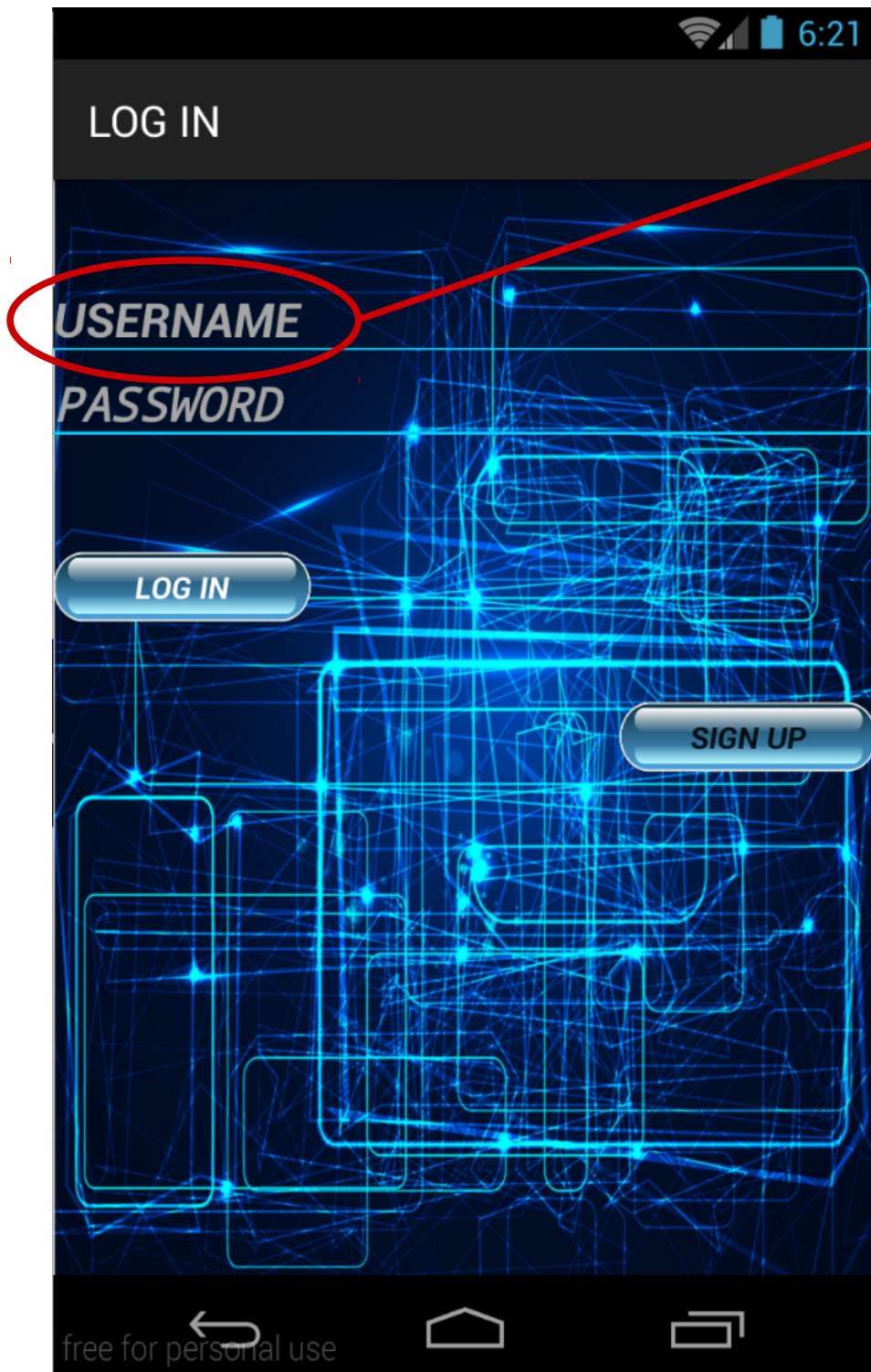
1. Te registras con un nombre de usuario y una password. Cada usuario tiene un nombre de usuario distinto en mensajerO asi que vas a intentar hasta encontrar tu nombre! Y una password que no sea vacia.
2. Configuras tu perfil.
3. Ya estas adentro! Comunícate.

Cuando cierres mensajerO y no hayas echo Log Out, tu sesion quedara registrada y la proxima vez que entres no va a ser necesario ni que te registres de nuevo ni que tengas que loguearte con tu username y tu password. Directamente entras al chathall y tus chats anteriores van a estar ahí.

Si quieres entrar con otro nombre de usuario haces Log Out y te registras con el otro usuario, o te Logueas con las credenciales del otro usuario (si ya esta registrado).

Cuando quieras volver a entrar con un usuario ya registrado lo haces desde la pantalla de Log In con las credenciales (username y password) con las que te registraste en mensajerO.

Screenshots



Pone aca tu nombre de usuario si ya estas registrado (si ya entraste alguna vez)

Screen de cada pantalla con círculos rojos e indicadores de que hace cada elemento.

Known Issues

No se que poner aca.

Troubleshooting

- Si la aplicación no se puede conectar con el servidor te va a informar. Todos tus mensajes van a estar a salvo siempre.

- Si no puedes mandar un mensaje al apretar el botón SEND es porque el servidor no está respondiendo. En cuanto tengas respuesta del servidor vas a poder mandarlo.

Cuales mas podría agregar?

Documentación de Administrador

Pablo cual es la diferencia entre la documentacion de administrador y la documentacion tecnica? A quien estan dirigidas cada una?

Get Started

- division de carpetas: explicacion de como es la estructura del codigo de la app.
- clases y que hacen: explicacion similar a la documentacion tecnica sobre las clases que componen la aplicación y que hace cada una.
- layouts: explicacion de la estructura de los layouts.
- archivos drawables: donde estan, cuales son, donde se usan. Peso recomendado de los archivos.
- testing: cuales son los tests, que hacen, en que se enfocan.
- conexión con el servidor: como se realiza, que pasa si no hay conexión posible.
- logueo de errores: donde se encuentra el log, que formato tiene y que informa.

Instalación y configuración

entorno

- donde descargar el IDE
- configurar el jre
- descargar el sdk

aplicación

- en device Android: Ajustes → Seguridad → Origenes Desconocidos (Permitir la instalación de aplicaciones desde orígenes distintos de PlayStore). Tildar la opción.
- colocar el archivo mensajerO.apk en una la carpeta Downloads.
- Abri el archivo. Acepta la instalacion.
- Listo.

Forma de uso

1. Registrarse con un nombre de usuario y una password. Cada usuario tiene un nombre de usuario distinto en mensajerO asi que intentar hasta encontrar un nombre de usuario valido! La password no debe ser vacia.
2. Configurar perfil.
3. Una vez en chats hall se puede pasar a la pantalla Users donde estan todos los usuarios o hacer Log Out para salir. En caso de tener algun chat se puede abrilo haciendo touch en el.

Cuando se cierra mensajerO y no se haya echo Log Out, la sesion quedara registrada y la proxima vez que se entra a la app no va a ser necesario ni registrarse de nuevo ni ingresar con tu username y tu password. Directamente se entra al chathall donde estaran tus chats anteriores.

Para entrar con otro nombre de usuario se hace Log Out y se registra el otro usuario, o se loguea con las credenciales del otro usuario (si ya esta registrado).

Cuando se quiere volver a entrar con un usuario ya registrado se hace desde la pantalla de Log In con las credenciales (username y password) con las que se ha antes registrado en

mensajerO.

Mantenimiento

Se podria mejorar el codigo.

Se podria hacer mas robusta la app cubriendo mas casos de error.

Se podria agregar mas features.

Known Issues

No se que poner.

Troubleshooting

Causas de problemas?