MensajerO

Taller de Programación II

Grupo 7

Integrantes:

- Meller, Diego
- Franco, Tomas
- Leguizamon, Cesar

Ayudante:

- Fusca, Gabriel

Repositorio:

https://github.com/mellerster/7552Grupo7

Changelog

- Servidor
  - Servicio de Autentificación (#20)
  - Servicio de registración de usuarios (#37)
  - Servicio para delivery de conversaciones (#66)
  - Servicio para delivery de difusión (#64)
  - Almacenamiento de conversaciones (#65)
  - Servicio de consulta de usuarios disponibles
  - Checkin de usuarios (#67)
  - Servicio de administración de perfíl de usuario. (#41)
- Cliente
  - Registrar un nuevo usuario (#3)
  - Acceder con un usuario ya creado (#2)
  - Ver el listado de usuarios conectados (#4)
  - Configurar Mi Perfil (#7)
  - Ver el perfil de otro usuario (#9)
  - Realizar Checkin para que todos sepan mi ubicación (#54)
  - Ver el listado de conversaciones (#53)
  - Enviar Mensajes con otros usuarios (#55)
  - Enviar un mensaje a todos los usuarios conectados (Broadcast) (#57)

División de Tareas:

Cesar Leguizamon

Encargado de la aplicación Servidor, esto implica:

- Diseño general del servidor.
- Diseño de la base de datos.
- Seteo del sistema de compilación.
- Seteo del sistema de testeo.
- Configuración de la documentación del servidor (Doxygen).
- Configuración de la generación de code coverage.
- Configuración de la librerias utilizadas.
- Implemetación del servidor.
- Implemetación de la base de datos.
- Implemetación de los unit tests.
- Documentación del servidor.

Diego Meller

Aplicación Cliente, se detalla a continuación

- Registrar Usuario.
- Ver estado de usuario.
- Configurar Perfil.
- Ver listado de usuarios conectados.
- Enviar mensaje de broadcast.
- Configurar Ajustes de la aplicación.
- Conexión de la aplicación cliente con la RestAPI (todas las funcionalidades entregadas).
- Documentación

Aplicación Servidor:

- Diseño inicial de la base de datos.
- Request y Handlers de Conversacion y Mensaje.
- Documentación de RestAPI.

Tomas Franco

Aplicación Cliente, se detalla a continuación

- Pantalla de Autentificación de Usuario
- Pantalla de Realizar Checkin
- Pantalla de Listado de Conversaciones
- Pantalla de Conversación

Gestión de Tareas:

Todas las tareas que aparecen como Closed en el sistema de ticketing estan implentadas y en funcionamiento.

Sistema de ticketing: https://github.com/mellerster/7552Grupo7/issues

Aplicación Cliente del Mensajero

El cliente es una aplicación nativa Android, la misma va a estar generada en un archivo apk.

Versión minima del sistema operativo

La aplicación esta compilada para correr en Android a partir de la versión Lollipop 5.0 (API 21)

Instalación

Copiar el archivo apk al dispositivo movil y abrirlo, al hacerlo se instalara la aplicación.

El instalador esta aquí:
https://github.com/mellerster/7552Grupo7/blob/Entrega_Final/documentacion/DocumentacionGenerada/MensajerO.apk

Configuración

En la pantalla inicial, se puede presionar el boton de AJUSTES, desde allí se podra cambiar la IP del servidor, el Puerto de conexión y las demoras, por defecto si no se configura ninguno es la ip http://10.0.2.2 y el puerto 8080, dado que la ip es la de la maquina que aloja el emulador del android. Las demoras son en milisegundos.

Código Fuente

El código fuente se encuentra documentado con doxygen aquí:
https://github.com/mellerster/7552Grupo7/tree/Entrega_Final/documentacion/DocumentacionGenerada/Documentacion_Codigo_Cliente

Administrador del Servidor

Get Started

Para poder compilar y/o correr la aplicación servidor necesitarán:

1. [RocksDB] (https://github.com/facebook/rocksdb “Repositorio github de rocksDB”)
2. [CMake] (http://www.cmake.org “Sitio oficial de CMake”)
3. [Doxygen] (http://www.stack.nl/~dimitri/doxygen/index.html “Sitio oficial de Doxygen”) - Opcional

RocksDB

Es necesario compilar e instalar esta libreria, para esto hay que:

1. Bajarse el codigo fuente de su repositorio
2. Compilar mediante los comandos:

```
$ PORTABLE=1 make static_lib
$ PORTABLE=1 make shared_lib
```

3. Instalar la librerias en el sistema ejecutando:

```
$ make install
```

CMake

Es necesario para compilar la aplicación, se puede instalar ejecutando en la linea de comando:

```
$ sudo apt-get install cmake
```

Doxygen

Es necesario para extraer la documentación del código fuente del servidor, para instalarlo se puede ejecutar el comando:

```
$ sudo apt-get install doxygen
```

Instalación y configuración

Una vez que las dependencias esten cubiertas hay que seguir los siguientes

pasos para compilar la aplicación:

1. Dentro del directorio de la aplicación servidor existe una carpeta vacia llamada bin, entrar en la misma.
2. Ejecutar CMake mediante el comando

```
$ cmake ../src
```

3. Compilar la aplicación ejecutando el comando make

Listo! La aplicación del servidor deberia compilar sin errores.

Para correr la aplicación ejecutar en la terminal:

```
$ ./serverMensajerO
```

Forma de uso

Para iniciar el servidor dentro de la aplicación debe presionar Y y luego enter.

Para finalizar el servidor debe presionar X y luego enter.

Warning Ambos comandos (X e Y) son en Mayuscula

Testing

La aplicación servidor posee una cantidad de tests que pueden ser corridos, existen dos formas de llevar esto a cabo:

- Ejecutar desde dentro de la carpeta bin el comando

```
$ make test
```

O

- Dentro de la carpeta bin existe otra carpeta llamada tests esta contiene, entre otras cosas, varios archivos ejecutables; Estos son los tests a correr.

La diferencia entre las dos formas es que en la segunda se obtiene mas información sobre los tests que fallaron, pero ambas formas corren los mismos tests.

Code coverage

Para obtener los datos de code coverage se debe compilar la aplicación y luego ejecutar el comando

```
$ make test coverage
```

, esto hara un par de cosas:

1. Correrá todos los tests de la aplicación.
2. Capturará toda la información de code coverage de los tests sobre la aplicación.
3. Generará un reporte en formato html dentro de la carpeta CoverageReport.

Se debe abrir el archivo index.html para acceder al reporte.

Mantenimiento

Documentación

El código fuente del servidor esta documentado con Doxygen, por lo tanto es necesario correr el comando:

```
$ doxygen
```

Third-party libs

El servidor utiliza un número de librerias externas, estas son:

- [Mongoose] (https://github.com/cesanta/mongoose "Mongoose git repository"): Utilizado como web-server.
- [JsonCpp] (https://github.com/open-source-parsers/jsoncpp "JsonCpp git repository"): Utilizado para parsear los datos en formato JSON.
- [HumbleLogging] (http://humblelogging.insanefactory.com/ "Página oficial de HumbleLogging"): Framework de loggeo.
- [Catch] (https://github.com/philsquared/Catch "Catch git repository"): Framework de testeo.
- [Hippomocks] (https://github.com/dascandy/hippomocks "Hippomocks git repository"): Framework de mockeo para los tests.