Jobify - Taller de Programación II Documentation

Publicación 1.0

Luis Arancibia, Tomás Mussi, Alfredo, Ezequiel Dufau

nov. 30, 2016

Contents:

1. Introducción **1**

2. Instalación **3**

2.1. Heroku-test . . . 3
2.2. Coverage . . . 3
2.3. Curl-Curlpp . . . 4
2.4. Gtest . . . 4
2.5. Heroku-Node-Postgresql . . . 4
2.6. Jsoncpp-Mongoose-LevelDB . . . 5
2.7. Log4cpp . . . 6

CAPÍTULO 1

Introducción

El trabajo práctico consta de 4 aplicaciones que funcionan para dar servicio a Jobify, una red social de profesionales.

- **AppAndroid**: aplicación android que permite a un usuario registrarse, editar su perfil en la red social, chatear, votar, encontrar profesionales cercanos y agregar contactos a la lista de amigos. Desarrollado en Android
- **AppServer**: servidor responsable de toda la lógica de Jobify, da servicios a la aplicación android. Además persiste todos los datos del usuario. Desarrollado en c++ y levelDB como base de datos.
- **SharedServer**: servidor que contiene datos de uso común de la aplicación. Utilizando Heroku para hacer un deploy de la aplicación. Desarrollado en node.js y postgreSQL como base de datos
- **WebAdmin**: web que permite editar los datos utilizado por el SharedServer.

CAPÍTULO 2

Instalación

Contenido:

Heroku-test

- Instalar python

  sudo apt-get install python

- Instalar pip

  sudo easy_install pip

  Aclaración: si no funciona instalar sudo apt-get install python-setuptools

- Instalar nose

  pip install nose

  sudo pip install requests sudo pip install -U mock

Prueba de test:

Primero asegurarse de que este corriendo el servidor localmente, si no es asi ejecutar

heroku local web

Luego ejecutar los test con el siguiente comando

heroku local test

Coverage

#Intalar lcov

sudo apt-get install lcov

#Uso Hay que compilar el server como siempre y una vez finalizado, ejecutar el script ./coverage.sh en la carpeta server. Se genera la documentación y automaticamente se abre el firefox con un informe de coverage. Se probó con los 17 test que están implementados.

Curl-Curlpp

Bajar paquete Para ubuntu 14.04 el paquete es: libcurl4-openssl-dev

sudo apt-get install libcurl4-openssl-dev libboost-all-dev

Descargar y destarear curl

wget https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/curlpp/curlpp-0.7.3.tar.gz
tar zxf curlpp-0.7.3.tar.gz

Instalar

cd curlpp-0.7.3 ./configure make sudo make install

Gtest

1. Descargar el repositorio git clone https://github.com/google/googletest.git

2. Compilar la libreria

cd googletest

cmake -DBUILD_SHARED_LIBS=ON .
 make
 cd googletest

sudo cp -a include/gtest /usr/include

cmake -DBUILD_SHARED_LIBS=ON .
 make
 sudo cp -a libgtest_main.so libgtest.so /usr/lib/

3. Updatear la cache del linker
 $ sudo ldconfig -v | grep gtest

Deberiamos ver esto
 libgtest.so.0 -> libgtest.so.0.0.0
 libgtest_main.so.0 -> libgtest_main.so.0.0.0

Heroku-Node-Postgresql

1- Descargar el proyecto del GitHub:

git clone -b [branch] [remote_repo] en nuestro caso seria:

    git clone -b sharedServer https://github.com/tomasmussi/taller2.git

2- Instalar Node.js:

    curl -sL https://deb.nodesource.com/setup_4.x | sudo -E bash -

    sudo apt-get install -y nodejs

Ejecutar:

    node -version

y verificar que sea V4.5.0

3- Instalar Heroku:

Chequear que ruby este instalado con:

    ruby -version

y si esta instalado ejecutar:

    wget -O- https://toolbelt.heroku.com/install-ubuntu.sh | sh

4- Instalar Postgresql:

sudo apt-get install postgresql

para chequear:

sudo su postgres

cambio de usuario -> postgres@alfredo-GFAST:/home/alfredo/Jobify/taller2/SharedServerHeroku$

ejecutar:

exit

para volver al usuario -> alfredo@alfredo-GFAST:/home/alfredo/Jobify/taller2/SharedServerHeroku$

5- Ingresar a la carpeta taller2/SharedServer y ejecutar:

heroku local database

6- Luego ejecutar:

heroku local web

7- Ingresar a:

http://localhost:5000/

http://localhost:5000/job_positions

Jsoncpp-Mongoose-LevelDB

1- Guia de instalacion leveldb

Instalar git-core and libsnappy-dev

sudo apt-get install git-core libsnappy-dev

Clonar leveldb del repositorio de git

git clone https://github.com/google/leveldb.git

Compilación

cd leveldb make

Instalación

cd out-shared sudo cp –preserve=links libleveldb.* /usr/local/lib cd ../include sudo cp -R leveldb /usr/local/include/ sudo ldconfig

2- Instalacion mongoose-cpp y jsoncpp

Para instalar mongoose-cpp y jsoncpp y poder compilar el server, se deben seguir estos pasos

1. Hacer un checkout del proyecto [jsoncpp](https://github.com/open-source-parsers/jsoncpp)

   git clone https://github.com/open-source-parsers/jsoncpp.git

2. **dentro de jsoncpp ejecutar comandos:** mkdir build cd build cmake .. sudo make install

Con esto logramos instalar jsoncpp en linux, creo que lo que hace es copiar todo el codigo a /usr/local/include y cuando compilas, se buscan todos los includes en $PATH que contiene a /usr/local/include. Y listo! Porque la carpeta de mongoose-cpp la tenemos dentro de nuestro proyecto, asi que los pasos son:

cd server mkdir build && cd build cmake .. make ./src/server

Y ya está el server andando escuchando en localhost, puerto 8080. Abrir en el navegador:

localhost:8080/

Log4cpp

Bajar ultima version de: http://log4cpp.sourceforge.net/

sudo ./configure

sudo make

sudo make check

sudo make install

sudo ldconfig