

DOCUMENTACIÓN

PROYECTO GO ANALYTICS

16 de junio de 2016

Ivan Leonardo Galeano Saavedra
Networks Lab.
Bogotá, Colombia
ilgaleanos@gmail.com

Índice general

1. Motivaciones, lenguajes y otros.	5
1.1. Lenguajes.	6
1.2. Arquitectura.	6
1.3. Dependencias	7
1.4. Terminología	8
 2. Código: Usuarios	 9
2.1. Necesidad	9
2.2. Funcionalidad	9
2.2.1. index	9
2.2.2. acceder	9
2.2.3. home	9
2.2.4. menu	10
2.2.5. salir	10
2.2.6. recuperar	10
2.2.7. usuariorecuperar	10
2.2.8. empresas	10
2.2.9. empresaeditar	10
2.2.10. empresaeliminar	11
2.2.11. empresanueva	11
2.2.12. proyectonuevo	11
2.2.13. proyectoeditar	11
2.2.14. proyectoeliminar	11
2.2.15. usuarios	12
2.2.16. usuarioeditar	12
2.2.17. usuarioeliminar	12
2.2.18. usuarioreenviar	12
2.2.19. usuarioactivar	12
2.2.20. usuarionuevo	12
2.2.21. licencia	13
2.2.22. cuenta	13
2.2.23. logs	13
2.2.24. reportarerror	13
2.2.25. términos y privacidad	13
2.2.26. e400, e403, e404, e500	13

Capítulo 1

Motivaciones, lenguajes y otros.

Introducción

El presente texto es un intento de exponer de la manera mas clara, sencilla y concreta el código referente a Go Analytics, también llamado talenter o gogo para el equipo de desarrollo. El desarrollo esta realizado casi en su totalidad en código pyhton en su versión 2.7.X y django 1.8.12+; algunas partes como veremos están escritas en código C++ compilado para python. De la misma manera de la parte del usuario se uso javascript con jquery y librerías estándar para diversas funcionalidades.

Esta es la segunda y actual versión del código ya que inicialmente se tenia uno más sencillo, pero funcional, el cual por innumerables cambios de requerimientos se hizo inmanejable. Tratando de mantener una filosofía mas modular se elaboró pensando en un gogo sin versión 360. Esta versión 360 se agregó como módulos adicionales tratando de mantener el código que se había escrito sin modificaciones en miras a la estabilidad.

Inicialmente se expondrá la arquitectura actual, versiones, requerimientos y demás paquetería necesaria. Se dedicará un capítulo para cada carpeta de código (app de django) en la cual se expondrá la necesidad, funcionalidad y errores probables de cada app, para entender la arquitectura se recomineda un estudio juicioso de los models.py de cada una. los errores no reportados a la fecha, significa que los usuarios que probaron la aplicación o no encontraron o no los reportaron

Nota personal: Espero que el desarrollador que trate con el código sea muy comprensivo a la hora de examinarlo ya que el fue desarrollado sin unos requerimientos detallados, sino con cada nueva versión, se solicitaban nuevas modificaciones sobre lo ya elaborado lo que ha conllevado en una no clara consecución de un código completamente estable.

1.1. Lenguajes.

El uso de este manual requiere conocimiento en Django 1.8, Python 2.7, C++11 El código ha sido escrito en lenguaje Python y C++

```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
```

para el framework Django

Django 1.8.13

Javascript con JQuery 2.x.

1.2. Arquitectura.

En AWS poseemos 3 servicios:

1. EC2: 2 servidores, servidor nano para un servicio prestado a Slam Security (no hablaremos de ese proyecto en el manual) y servidor micro T2 con sistema operativo Ubuntu 14.04. 30 gb de almacenamiento y 1 Gb de memoria RAM.
2. RDS: 1 Servidor T2 nano de base de datos en Postgresql de 5 Gb de almacenamiento y 1 Gb de memoria RAM, este servidor se inició por las previsiones de ventas del área comercial.
3. SES: 1 conexión smtp de alto envío de correos.

Las conexiones las realicé de la siguiente manera. La aplicación corre en un servidor apache

```
Server version: Apache/2.4.7 (Ubuntu)
Server built:   Jan 14 2016 17:45:23
```

para el cual los certificados ssl los generé por medio de Let's Encrypt <https://letsencrypt.org/getting-started/>. estas claves ssl no estoy muy seguro de trasladarlas de servidor por el modo de generación, es por ello que no están incluidas y es altamente recomendable leer la documentación de este método de obtención y mantenimiento de claves ssl.

La aplicación se conecta localmente a una instancia de pgbouncer (*esto con la finalidad de mantener siempre una conexión abierta y evitar el overhead de estar abriendo conexiones*)

```
pgbouncer version 1.5.4
(compiled by <buildd@roseapple> at 2013-06-18 17:16:50)
```

La clave de la conexión local con pgbouncer es

```
'USER': 'usuariodb_gogo',
'PASSWORD': 'W#y2d@uV4+eSPuwrEc$UTrE4eCruTHas',
```

la clave de conexión de pgbouncer a la base de datos es

```
user=usuariodb_gogo
password=W#y2d@uV4+eSPuwrEc$UTrE4eCruTHas
```

El archivo de configuración apache2.conf, los sites-availables, gogo pgbouncer.ini al igual que el userlist.txt se encuentran en ./gogo/gogo/configs/

La base de datos contiene 3 bases de datos una llamada gogo perteneciente a esta aplicación, otra llamada talenter que no es mas que una copia incompleta de la primera y una tercera perteneciente a slam. Se cuentan con tres Group Roles y tres Login Roles uno para cada aplicación y un rol maestro (no superusuario, RDS no lo proporciona) :

```
user=suidi
password=Networks123*
```

Las claves de administración de dominios están bajo Juan Sebastián Henao P. y Ricardo Montoya Meneses.

1.3. Dependencias

Se deben instalar los siguientes paquetes para que la aplicación se ejecute según es debido:

1. gcc
2. python-crypto
3. python-psycopg2
4. apache2
5. libapache2-mod-wsgi
6. python-scipy
7. python-xlrd
8. python-xlwt
9. python-pip
10. redis-server
11. libpq-dev
12. python-dev
13. ctemplate (librería de template en c++ conocida también como google-template)

y las siguientes con el uso de pip

1. django==1.8.13
2. django-mptt
3. redis
4. django-redis-cache
5. django-pgjsonb

1.4. Terminología

Un **usuario** es un individuo consultor o cliente que tiene acceso a la administración o visualización de resultados en la herramienta por medio de un usuario y una contraseña. Un usuario **consultor** es la persona encargada de gestionar el ingreso de la mayor parte de la información en los proyectos. Un usuario **cliente** es la encargada de visualizar resultados. Un **participante** (denominado colaborador dentro del código) es la persona a la cual le llega el link para llenar las encuestas. Un **evaluado** es la persona o entidad de la cual se realiza la encuesta. Un **evaluador** una persona que contesta encuestas.

Capítulo 2

Código: Usuarios

Es la aplicación columna vertebral del proyecto, se encuentran las tablas que son transversales a la aplicación.

2.1. Necesidad

Se escribió con el objetivo de mantener lo mas centralizado la aplicación con la tabla proyectos. En una versión anterior ésta tabla pertenecía a *cuestionarios* de manera que se puede notar como aun conserva dicho nombre asociado “*cuestionarios_proyectos*”.

2.2. Funcionalidad

2.2.1. index

Controlador de la página index.

errores

Al formulario le hace falta una buena implementación del formulario de contacto, aprovisionado de una mejor seguridad (captcha).

2.2.2. acceder

Controlador de la página de logueo.

errores

Al igual que la página de inicio le hace falta un captcha de seguridad.

2.2.3. home

Controlador encargado de enlistar los proyectos asociados a cada usuario

errores

No se ha reportado ninguno a la fecha.

2.2.4. menu

Controlador encargado de la redirección según tipo de usuario al intentar ingresar a un proyecto.

errores

No se ha reportado ninguno a la fecha.

2.2.5. salir

Controlador encargado de destruir las sesiones.

errores

No se ha reportado ninguno a la fecha.

2.2.6. recuperar

Controlador encargado de enviar email con los pasos para recobrar contraseñas.

errores

Al igual que la página de inicio le hace falta un captcha de seguridad.

2.2.7. usuariorecuperar

Controlador encargado de researle el password a los usuarios que realizaron el proceso del controlador recuperar 2.2.6

errores

No se ha reportado ninguno a la fecha.

2.2.8. empresas

Controlador encargado de consultar las empresas que cada usuario ha registrado

errores

No se ha reportado ninguno a la fecha.

2.2.9. empresaeditar

Controlador encargado de proveer los datos para la edición de la empresa

errores

No se ha reportado ninguno a la fecha. Se podría fusionar con el template de creación.

2.2.10. empresaeliminar

Controlador encargado de eliminar las empresas.

errores

No se ha reportado ninguno a la fecha.

2.2.11. empresanueva

Controlador encargado de crear empresas. Estas empresas están asociadas de muchas a uno con User (usuarios).

errores

Ninguna conocido.

2.2.12. proyectonuevo

Controlador encargado de la creación de proyectos. Los User están asociados de muchos a muchos con los Proyectos.

errores

En versiones anteriores se ocultaba el botón de creación cuando subían un archivo no imagen, desembocó en errores. Esta característica fue eliminada. No se ha reportado ninguno a la fecha.

2.2.13. proyectoeditar

Controlador encargado de hacer modificaciones al proyecto. Por motivos de tipos de encuesta y demás muchos los campos no son alterables y se aconseja se mantengan así. Se puede fusionar con el template de renderizado de 2.2.12.

errores

No se ha reportado ninguno a la fecha.

2.2.14. proyectoeliminar

Controlador encargado de eliminar proyectos, la verdadera eliminación se ejecuta como tarea externa programada.

errores

No se ha reportado ninguno a la fecha.

2.2.15. usuarios

Controlador encargado de mostrar los usuarios creados por cada participante, de manera recursiva (se pueden ver los usuarios creados por los usuarios que yo haya creado recursivamente). Se usa una biblioteca de queries transversales para este propósito.

errores

No se ha reportado ninguno a la fecha.

2.2.16. usuarioeditar

Controlador encargado de editar al usuario accedido.

errores

No se ha reportado ninguno a la fecha.

2.2.17. usuarioeliminar

Controlador encargado de eliminar usuarios (logueables) de la aplicación. Esta eliminación se ejecuta en el acto.

errores

No se ha reportado ninguno a la fecha.

2.2.18. usuarioreenviar

Controlador encargado de reenviar correos de bienvenida a los usuarios creados que manifiesten la necesidad de ello.

errores

No se ha reportado ninguno a la fecha.

2.2.19. usuarioactivar

Controlador encargado de darle la bienvenida a un nuevo usuario.

errores

No se ha reportado ninguno a la fecha.

2.2.20. usuarionuevo

Controlador encargado de ingresar los datos de un nuevo usuario.

errores

No se ha reportado ninguno a la fecha.

2.2.21. licencia

Controlador "decorativo" que muestra el número de proyectos que ha adquirido el número de proyectos que ha activado.

errores

No se ha reportado ninguno a la fecha.

2.2.22. cuenta

Controlador encargado del cambio de contraseña por parte de los usuarios.

errores

No se ha reportado ninguno a la fecha.

2.2.23. logs

Controlador encargado de mostrar los registros importantes que han ocurrido en la aplicación.

errores

No se ha reportado ninguno a la fecha.

2.2.24. reportarerror

Controlador encargado de recibir las solicitudes de soporte por parte de los usuarios.

errores

No se ha reportado ninguno a la fecha.

2.2.25. términos y privacidad

Controladores encargados de renderizar las paginas correspondientes a términos de servicio y políticas de privacidad.

errores

No se ha reportado ninguno a la fecha.

2.2.26. e400, e403, e404, e500

Controladores de renderizar los errores correspondientes. Debido a la "satanización" de la palabra *error* se ha cambiado por bloqueo gran parte de los textos de estos templates.

errores

No se ha reportado ninguno a la fecha.