Universidad
de La Laguna

GScout
Desarrollo e implantación de una aplicación para la gestión de grupos Scout

Title in English .

José Daniel Juárez Dávila

Dpto. Nombre del Departamento

Escuela Técnica Superior de Ingeniería Informática

Trabajo de Fin de Grado

La Laguna, 29 de mayo de 2013

D. **Nombre Apellido1 Apellido2**, con N.I.F. 12.345.678-X profesor Titular de Universidad adscrito al Departamento de Nombre del Departamento de la Universidad de La Laguna

C E R T I F I C A

Que la presente memoria titulada:

"*Titulo del Trabajo.*"

ha sido realizada bajo su dirección por D. **Nombre Apellido1 Apellido2**, con N.I.F. 12.345.678-X.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 29 de mayo de 2013

Agradecimientos

Equipo de scout Aguere por su colaboración en el proyecto.
XXX
XXX
XXX

Resumen

El objetivo de este trabajo ha sido crear una aplicación web para los scout de Aguere 70, la cual facilite la gestión de los socios de dicha organización.

Para ello usamos como framework Django, pero como tambien vamos a trabajar con la estructura de Google App Engine, utilizamos la version de django-nonrel que nos permite utilizar base de datos no relacionales ya que la que usa App Engine sigue la tecnología BigTable de Google. La ventaja del uso de este tipo de base de datos es que son mas escalable, y gracias al framework de Django se pueden manipular por medio de este con QuerySet de Django, salvo que no podemos usar joins ni many to many, etc ni ninguna relacion entre tablas que no sea de clave foranea.

Aparte del frameworky y de App Engine introducimos en la aplicación APIs de Google para poder iniciar sesión con una cuenta de google y obtener los datos de dicha cuenta sin necesidad de almacenarlo, ademas tambien utilizamos otra API para la exportación de información a Google Drive.

En cuanto a las templates de la aplicacion se utiliza la configuración CSS de nos brinda Bootstraps y alguno que otro codigo en jQuery para crear dinamismo en las paginas y con la iteracion del usuario y la aplicación.

Palabras clave: Palabra reservada1, Palabra reservada2, ...

Abstract

Here should be the abstract in a foreing language...

Keywords: *Keyword1, Keyword2, Keyword2, ...*

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

El objectivo de este proyecto es desarrollar una aplicación para la gestión de un grupo de Scout. La aplicación debera gestionar a los chicos asociados al grupo y mantener su información personal, incluyendo familiares, datos bancarios y medicos.

La tecnologia principal usada se basa en aplicaciones web con un entorno de desarrollo de alto nivel, en nuestro caso don el Framework de Django, y tambien con implantación en la nube, gracias a Google App Engine.

1.1. Antecedentes y estado actual del tema

El Escultismo es un movimiento educativo fundado en el año 1907 por Baden Powell en Inglaterra e instalado en España en 1912. Su musión es dejar este mundo mejor de como lo encontramos, una misión que es posible gracias a una gran labor diaria y educativa que realizan de manera voluntaria jóvenes de todo el mundo.

Hoy en día se pueden encontrar páginas web de organizaciones de grupos Scout, pero la mayoria son simlemente para uso informativo y darse a conocer, como es el caso del grupo Aguere 70, que carece de una aplicación para la gestión de los propios socuts con sus tareas, que es en lo que se ennfocó este proyecto. Existe una implementación, ya desactualizada, basada en tecnología PHP para las funciones básicas en la gestión de grupos Scout: GNU Scout [1]. Esta implementación aunque no fue usada, sirvio para darnos una idea general de como enfocar nuestra aplicación.

De modo que creamos una aplicación en la nube utilizando una versión de Django adaptada a Google App Engine, un servicio de alojamiento web que permite desarrollar aplicaciones online sin necesidad de administrar o mantener servidores dedicados. De esta forma, los usuarios podrán utilizarla lo antes posible, evitando tareas de gestión y administración de sistemas.

1.2. Caracteristicas de la Aplicación

Resumen de los principales recursos utilizados en el desarrollo del proyecto.:

- Framework Django-nonrel

- Despliegue en Google App Engine
- Google APIs (Google+ y Google Drive)
- GitHub
- jQuery
- JavaScript
- Bootstrap CSS

En los próximos capítulos se profundizará en las herramientas empleadas en la elaboración del proyecto.

1.3. Actividades

El desarrollo del proyecto se organizó de las siguientes tareas:

Tarea	**Actividad**
Tarea1	Analisis de la aplicación GNU Scout [1] y el modelo de datos utilizado.
Tarea 2	Entrevistas con los responsables de un grupo Scout para analizar las funcionalidades ya previstas según [1] y añadir/eliminar/modificar aquellas de interés.
Tarea 3	Montar un repositorio GIT [2] para alojar los codigos del proyecto. Definir la estrategia de branching.
Tarea 4	Definir u proyecto en Pivotal tracker[3] para el seguimiento del proyecto. Esta herramienta facilita el desarrollo siguiento metodologías ágiles.
Tarea 5	Desarrollo de un proyecto piloto, realizar la implantación en GAE[2] comprobando el funcionamiento básico de esta plataforma.
Tarea 6	Entrevistas de seguimiento. 2º reunión con los responsables del grupo Scout para mostrar el piloto de la aplicación, refinar diseños,etc
Tarea 7	Desarrollo de la aplicación: implementar las funcionalidades requeridas (posibles entrevistas a lo largo del proceso para comprobar si la implementación cumple los requisitos del cliente: se realizarán por lo menos 4 iteraciones completas: análisis, desarrollo, test, implantación)
Tarea 8	Puesta en producción (fase beta), formación de usuario y gestión de errores.

1.4. Período de Desarrollo del Proyecto

El período de elaboración del proyecto abarca desde el 11 de septiembre de 2012 cuando se presentaron las ofertas de los proyectos, hasta mediados de junio de 2013 que es cuando corresponden las respectivas defensas orales, pero lo que es la elaboración en sí de la aplicación del proyecto abarcó del 30 de Enero de 2013 hasta primeros de Junio incluyendo los retoques finales y puesta a punto de la aplicación.

Figura 1.1: Ejemplo

Capítulo 2

Entorno de Desarrollo

En el capítulo anterior se ha introducido los antecedentes como el estado actual del proyecto, nombrado sus herramientas, actividades y periodos de desarrollo. Ahora nos vamos enfocar y describir lo que es el entorno de desarrollo de la aplicación

2.1. Bases de la aplicación

La aplicación GScout esta programada principalmente en lenguaje python, bajo el framework de django-nonrel, ya que como se ha comentado antes se necesitaba este framework especifico para trabajar con bases de datos no relacionales, debido a que como usamos Google App Engine para el despliegue, un requerimiento que tiene esta tecnologia es que solo trabaja con bases de datos no relacionales. En un principio se habia propuesto el uso de pivotaltracker para el seguimiento de la aplicación, pero como solo habia un desarrollador en proceso, pues se descartó la idea y se limito a tener un repositorio gestor de versiones por medio de github, donde se van guardando los cambios oportunos, y segun la información de los "commit" se puede ver lo que se ha hecho.

2.2. Modelo Vista Controlador: Django

Django es un framework de desarrollo web de código abierto, escrito en Python, que cumple en cierta medida el paradigma del Modelo Vista Controlador. Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005; el framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt.

2.3. Google App Engine

Google App Engine te permite ejecutar tus aplicaciones web en la infraestructura de Google. Las aplicaciones App Engine son fáciles de crear, de mantener y de ampliar al ir aumentando el tráfico y las necesidades de almacenamiento de datos. Con App Engine no necesitarás utilizar ningún servidor: solo tendrás que subir tu aplicación para que los

usuarios puedan empezar a utilizarla.

Puedes proporcionar a la aplicación tu propio nombre de dominio (como, por ejemplo, http://www.example.com/) a través de Google Apps. También puedes proporcionarle un nombre que esté disponible en el dominio appspot.com. Podrás compartir tu aplicación con todo el mundo o limitar el acceso a los miembros de tu organización.

Google App Engine admite aplicaciones escritas en varios lenguajes de programación.

Google App Engine permite desarrollar fácilmente aplicaciones que se ejecuten de forma fiable, incluso con pesadas cargas de trabajo y grandes cantidades de datos. App Engine incluye las siguientes funciones:

- Servidor web dinámico, totalmente compatible con las tecnologías web más comunes,
- Almacenamiento permanente con funciones de consulta, clasificación y transacciones,
- Escalado automático y distribución de carga,
- API para autenticar usuarios y enviar correo electrónico a través de Google Accounts,
- Un completo entorno de desarrollo local que simula Google App Engine en tu equipo,
- Colas de tareas que realizan trabajos fuera del ámbito de una solicitud web,
- Tareas programadas para activar eventos en momentos determinados y en intervalos regulares.

2.4. Google APIs

En la aplicación se usaron dos APIs de Google para aumentar y mejorar su funcionalidad.

2.4.1. Google Plus

Como GAE ya posee un modulo de autenticación por medio de Google Auth, en el proyecto se modifico para que se hiciera por medio de Google Plus, generando unas credenciales que se guardaran en el usuarios para que la aplicaccion pueda tener determinados permisos y poder obtener información que proporciona Google Plus. El objetivo de la incorporación de esta API a nuestra aplicación es evitar almacenar datos personales de los integrantes de la organización que usa la aplicación en una base de datos, sino usar directamente los que nos proporciona Google Plus, como nombre, apellidos, foto de perfíl, dirección, etc.

2.4.2. Google Drive

Uno de los objetivos de la aplicación en cuanto a funcionalidad era poder exportar una tabla filtrada o no con los datos de los socios, como estamos utilizando tecnologia Google, decidimos que la mejor forma era utilizar la API de Google Drive, para crear documentos de hoja de calculo con la información que le pasara la aplicación, para ello fue necesario modificar las credenciales que se almacenaban en los usuarios, para darle permiso y poder utilizar las funciones que nos proporciona dicha API.

2.5. GitHub

GitHub es un software para alojar proyectos utilizando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

Nuestra aplicación esta alojada en un repertorio público, cuyo repertorio se facilitara en la sección de enlances.

2.6. JavaScript y JQuery

2.6.1. JavaScript

JavaScript es un lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador, etc.

2.6.2. JQuery

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web. Cuyas caracteristicas son:

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.

- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.

2.7. CSS y Bootstrap

Para generar el estilo de nuestra aplicación utilizamos las tecnologías que se descriran a continuación.

2.7.1. CSS (Hojas de estilo)

Las hojas de estilo en cascada (Cascading Style Sheets, o sus siglas CSS) hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL.

2.7.2. Bootstrap

Para agilizar el maquetado de las páginas web de la aplicación utilizamos Twitter Bootstrap, que es una colección de herramientas de software libre para la creación de sitios y aplicaciones web. Contiene plantillas de diseño basadas en HTML y CSS con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz, así como extensiones opcionales de JavaScript.

Capítulo 3

Descripción de la aplicación

Bla, Bla, Bla, ...

3.1. Primer apartado de este capitulo

```
# -*- coding: utf-8 -*-

from django.conf.urls.defaults import *
from socios.views import *

urlpatterns = patterns('',
    ('^new/', 'django.views.generic.simple.direct_to_template',
      {'template': 'socios/f_asociado.html'}),
    ('^search/', 'django.views.generic.simple.direct_to_template',
      {'template': 'socios/search.html'}),
    (r'^create_personal/', newPersonal),
    (r'^search_socio/', search),
    (r'([0-9]+)/personales/?', personales_socio),
    (r'([0-9]+)/economicos/?', economicos_socio),
    (r'([0-9]+)/medicos/?', medicos_socio),
    (r'([0-9]+)/familiares/?', familiares_socio),
    (r'([0-9]+)/edit_personales/?', edit_personales),
    (r'^modify_personales/?', modify_personales),
    (r'([0-9]+)/edit_economicos/?', edit_economicos),
    (r'^modify_economicos/?', modify_economicos),
    (r'([0-9]+)/edit_medicos/?', edit_medicos),
    (r'^modify_medicos/?', modify_medicos),
    (r'^listado/', listado),
    (r'^listado_del/', listado_del),
    (r'^listado_economicos/', listado_economicos),
    (r'^del_socios/', del_socios),
    (r'^export_economicos/?', export_economicos),
     (r'^export/?', export),
    ('^familiares_new/', 'django.views.generic.simple.direct_to_template',
      {'template': 'socios/f_familiares.html'}),
```

```
    (r'search_familia', search_familia),
    (r'([A-Z]?[0-9]+[A-Z]?)/edit_familia/?', edit_familia),
    (r'modify_familia/?', modify_familia),
    (r'post_change_familia/?', post_change_familia),
    (r'([0-9]+)/change_familia/?', change_familia),
    (r'cambio_unidad/?', cambio_unidad),


)
```

Listing 3.1: Ejemplo de listado desde archivo

3.2. Segundo apartado de este capitulo

```
class D_Personales(models.Model):
    nombre=models.CharField(max_length=100)
    apellidos=models.CharField(max_length=100)
    dni=models.CharField(max_length=100)
    sexo=models.CharField(max_length=20)
    f_nacimiento=models.DateTimeField()
    direccion=models.CharField(max_length=200)
    c_postal=models.CharField(max_length=100)
    localidad=models.CharField(max_length=100)
    provincia=models.CharField(max_length=100)
    fijo=models.CharField(max_length=100)
    movil=models.CharField(max_length=100)
    f_ingreso=models.DateTimeField()
    f_baja=models.DateTimeField(blank=True, null=True)
    seccion=models.CharField(max_length=100)
    estudios=models.TextField()
    profesion=models.TextField()
    deportes=models.TextField()
    aficiones=models.TextField()
    socio_id=models.ForeignKey(Socio)
```

Listing 3.2: Ejemplo de listado puesto aquÃ

3.3. Tercer apartado de este capitulo

Capítulo 4

Desarrollo de la aplicación

En el capitulo 1 se describio bla, bla, bla.....

Capítulo 5

Conclusiones y trabajos futuros

Este capítulo es obligatorio. Toda memoria de Trabajo de Fin de Grado debe incluir unas conclusiones y unas líneas de trabajo futuro

Capítulo 6

Summary and Conclusions

This chapter is compulsory. The memory should include an extended summary and conclusions in english.

6.1. First Section

Capítulo 7

Presupuesto

Este capítulo es obligatorio. Toda memoria de Trabajo de Fin de Grado debe incluir un presupuesto.

7.1. Sección Uno

Tipos	**Descripcion**
AAAA	BBBB
CCCC	DDDD
EEEE	FFFF
GGGG	HHHH

Tabla 7.1: Tabla resumen de los Tipos

Apéndice A

Título del Apéndice 1

A.1. Algoritmo XXX

```
*******************************************************************************
*
* Fichero .h
*
*******************************************************************************
*
* AUTORES
*
*
* FECHA
*
*
* DESCRIPCION
*
*
*******************************************************************************/
```

A.2. Algoritmo YYY

```
/*******************************************************************************
 *
 * Fichero .h
 *
 *******************************************************************************
 *
 * AUTORES
 *
 * FECHA
 *
 * DESCRIPCION
```

```
 *
 *
 ********************************************************************************/
```

Apéndice B

Título del Apéndice 2

B.1. Otro apendice: Seccion 1

Texto

B.2. Otro apendice: Seccion 2

Texto

Bibliografía

[1] ACM LaTeX Style. http://www.acm.org/publications/latex_style/.

[2] GitHub. http://www.github.com.

[3] FACOM OS IV SSL II USER'S GUIDE, 99SP0050E5. Technical report, 1990.

[4] D. H. Bailey and P. Swarztrauber. The fractional Fourier transform and applications. *SIAM Rev.*, 33(3):389–404, 1991.

[5] A. Bayliss, C. I. Goldstein, and E. Turkel. An iterative method for the Helmholtz equation. *J. Comp. Phys.*, 49:443–457, 1983.

[6] C. Darwin. *The Origin Of Species.* November 1859.

[7] C. Goldstein. Multigrid methods for elliptic problems in unbounded domains. *SIAM J. Numer. Anal.*, 30:159–183, 1993.

[8] P. Swarztrauber. *Vectorizing the FFTs.* Academic Press, New York, 1982.

[9] S. Taásan. *Multigrid Methods for Highly Oscillatory Problems.* PhD thesis, Weizmann Institute of Science, Rehovot, Israel, 1984.