



Facultad de Ingeniería Universidad de Buenos Aires

75.61 Taller de Programación III

TP N°3 - Event Managment (Google App
Engine)

Profesor: Andrés Veiga
JTP: Pablo Roca

Integrantes:

Padrón	Nombre	Email
89579	Torres Feyuk, Nicolás R. Ezequiel	ezequiel.torresfeyuk@gmail.com

Índice

1. Introducción	3
2. Desarrollo	3
3. Arquitectura 4 + 1	4
3.1. Casos de Uso	5
3.2. Vista Lógica	6
3.3. Vista de Actividad	7
3.4. Vista de Despliegue	8
4. Resultados	9
4.1. Pruebas de Carga	10
5. Conclusión	11

1. Introducción

El presente trabajo práctico consiste en diseñar e implementar un sistema de recepción de eventos a través de la plataforma de cloud computing de google App Engine. El presente sistema debe permitir el ingreso de personas invitadas en diferentes eventos, como así también tener la posibilidad de poder consultar quienes son los invitados anotados en cada uno de ellos.

2. Desarrollo

Google App Engine permite realizar el desarrollo de las aplicaciones web a través de los siguientes lenguajes de programación:

- Java
- Python
- Go
- PHP

Debido a la simplicidad del lenguaje y la falta de conocimientos en las otras opciones, se decidió implementar el sistema en Python. App Engine permite realizar el desarrollo de la parte Servidor a través de cualquier web framework de Python que implemente la interfaz WSGI(Web Service Gateway Interface). Se lista a continuación algunos de los frameworks que cumplen con esta especificación:

- Django
- CherryPy
- Pylon
- web.py
- web2pyi
- webapp2

Para realizar la presente aplicación se decidió utilizar webapp2. El motivo de esta elección es que la complejidad del presente proyecto era baja y los tutoriales de google respecto a las aplicaciones de Python para App Engine están desarrolladas con este framework.

Para el desarrollo y diseño de la página web se utilizó HTML, scripts en Javascript y JQuery y el template language Jinja2. Este último fue utilizado la generación estática del página principal del proyecto, la cual nuevamente se realizó en función del tutorial de App Engine.

3. Arquitectura 4 + 1

3.1. Casos de Uso

3.2. Vista Lógica

3.3. Vista de Actividad

3.4. Vista de Despliegue

4. Resultados

TODO:

4.1. Pruebas de Carga

5. Conclusión

TODO: