

FACULTAD DE INGENIERÍA

UNIVERSIDAD DE BUENOS AIRES

TALLER DE PROGRAMACIÓN III (75.61)

2DO CUATRIMESTRE 2015

EJERCICIO DE APLICACIONES CLOUD

GOOGLE APP ENGINE

“RSVP”

FLORENCIA ALVAREZ ETCHEVERRY

19 Octubre 2015

ÍNDICE

Objetivos.....	3
Desglose de Actividades y Estimación de Esfuerzo	4
Vista de Casos de Uso	5
Vista Lógica	6
Vista de Procesos.....	7
Vista de Implementación.....	9
Vista de Despliegue	10
Pruebas de Carga.....	11

OBJETIVOS

El sistema tiene como objetivo que personas invitadas a un determinado evento confirmen asistencia al mismo. Los invitados accederán al sitio donde pueden registrarse utilizando una dirección de email, nombre y apellido, y compañía para la que trabajan. Se debe controlar que no se puedan registrar más invitados que una cantidad máxima de vacantes que posee el evento y si no hay más disponibles se envía un mensaje indicando tal cosa.

Además, en cualquier momento debe ser posible validar si una determinada persona ha confirmado que asistirá al evento.

Los requerimientos no funcionales se aseguran al utilizar el servicio Cloud de Google App Engine. Es decir, se asegura que el sistema esté siempre disponible y se limitan los costos de despliegue, servidores, hosting, etc.

También se realizarán pruebas de carga al sistema para mostrar cómo se comporta la aplicación bajo diferentes niveles de carga.

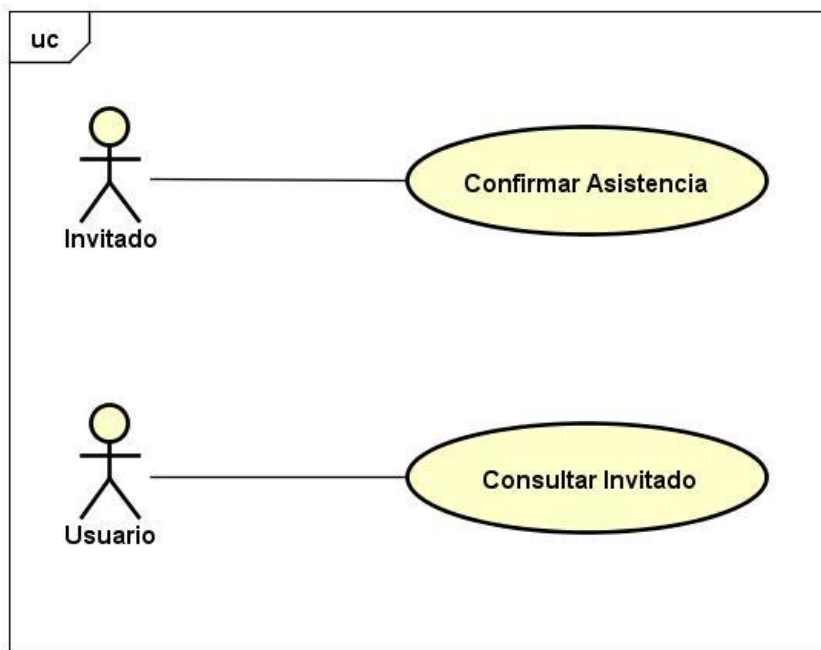
DESGLOSE DE ACTIVIDADES Y ESTIMACIÓN DE ESFUERZO

Tareas a realizar para el desarrollo del proyecto y la estimación de su realización:

1. Hacer diagrama de casos de uso, 5 minutos.
2. Hacer diagrama de clases, 15 minutos.
3. Hacer diagrama de secuencia de primer caso de uso, 10 minutos.
4. Hacer diagrama de secuencia de segundo caso de uso, 10 minutos.
5. Hacer diagrama de componentes, 15 minutos.
6. Hacer diagrama de deployment, 15 minutos.
7. Instalar y configurar Maven, 5 horas.
8. Configurar ambiente (Eclipse, proyecto, Java), 3 horas.
9. Hacer tutorial Google App Engine, 5 horas.
10. Hacer páginas y clases de la aplicación, 7 horas.
11. Probar la aplicación localmente, 3 horas.
12. Subir aplicación a Cloud, 1 minuto.
13. Probar aplicación online, 15 minutos.
14. Investigar herramienta para test de carga, 5 horas.
15. Diseñar protocolo de pruebas de carga, 2 horas.
16. Ejecutar tests de carga, 30 minutos.
17. Escribir informe, 2 horas.
18. Imprimir informe, 10 minutos.

VISTA DE CASOS DE USO

Se identifican dos casos de uso en el sistema que resuelven los requerimientos funcionales. Se muestran en el siguiente diagrama.



powered by Astah

Figura 1: Diagrama de Casos de Uso.

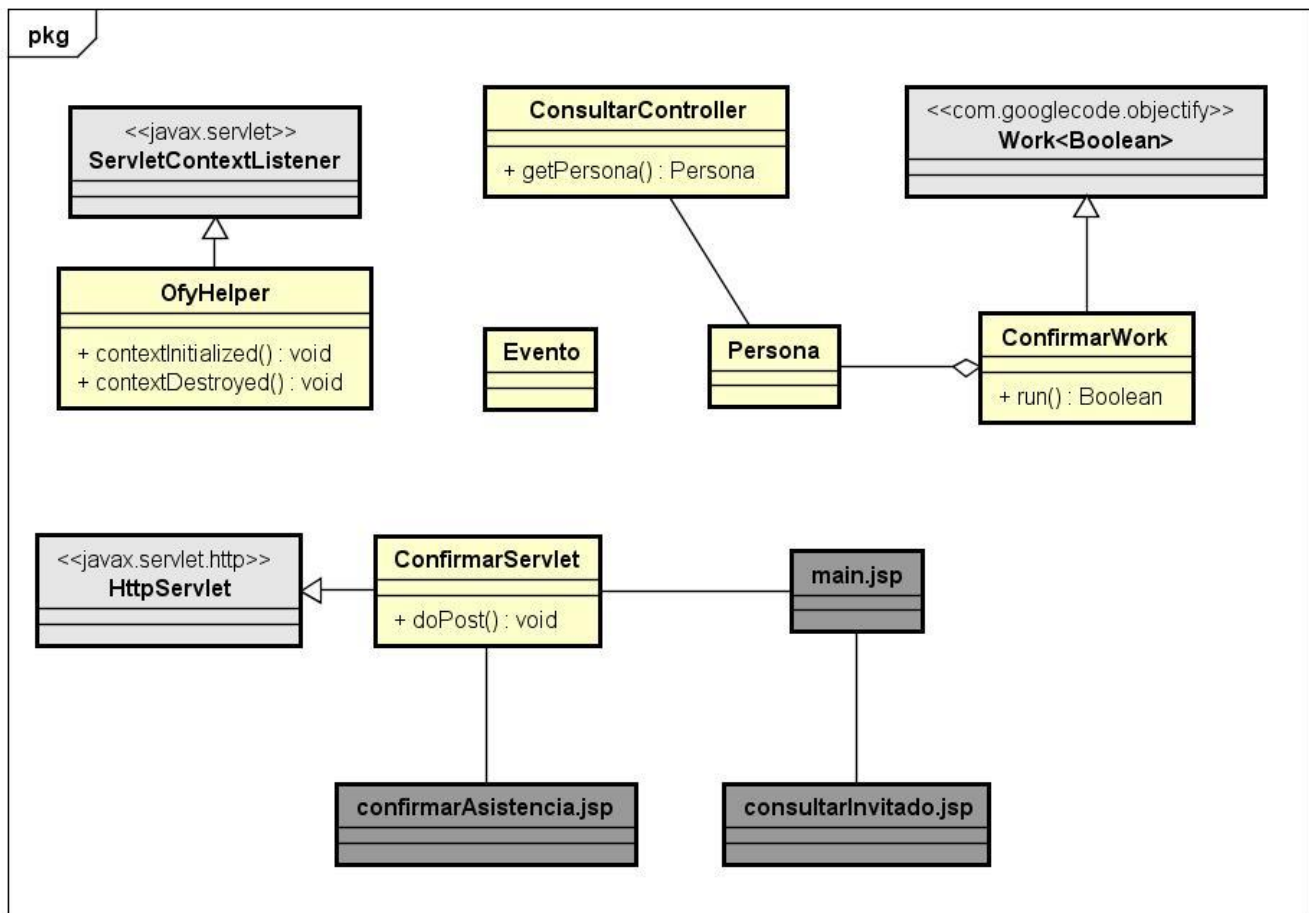
Se describen también los casos de uso modelados para este sistema.

Caso de uso: Confirmar Asistencia	
Descripción:	El invitado se registra al evento con su mail, nombre, apellido y compañía en la que trabaja.
Actores:	Invitado
Precondiciones:	-
Flujo Principal:	<ol style="list-style-type: none">1. El invitado ingresa su mail, nombre, apellido y compañía.2. Se obtiene la cantidad de registrados al evento.3. Se verifica si hay vacantes disponibles.4. Si no se agotaron las vacantes, se registra al invitado al evento.5. Se muestra el mensaje de retorno al invitado de acuerdo se haya podido registrar o no.
Postcondiciones:	El invitado está confirmado en el evento.

Caso de uso: Consultar Invitado	
Actores:	Usuario
Precondiciones:	-
Flujo Principal:	<ol style="list-style-type: none"> 1. Un usuario ingresa el mail del invitado que desea saber si está registrado en el evento. 2. Se consulta por ese mail en la base de datos. 3. Se devuelve al usuario si ese mail se encuentra registrado o no.
Postcondiciones:	El usuario obtuvo la información sobre la confirmación del invitado ingresado.

VISTA LÓGICA

En la aplicación se tiene la vista representada en los archivos .jsp (entidades en color gris oscuro) y el modelo que incluye clases propias y de la API de Google App Engine.



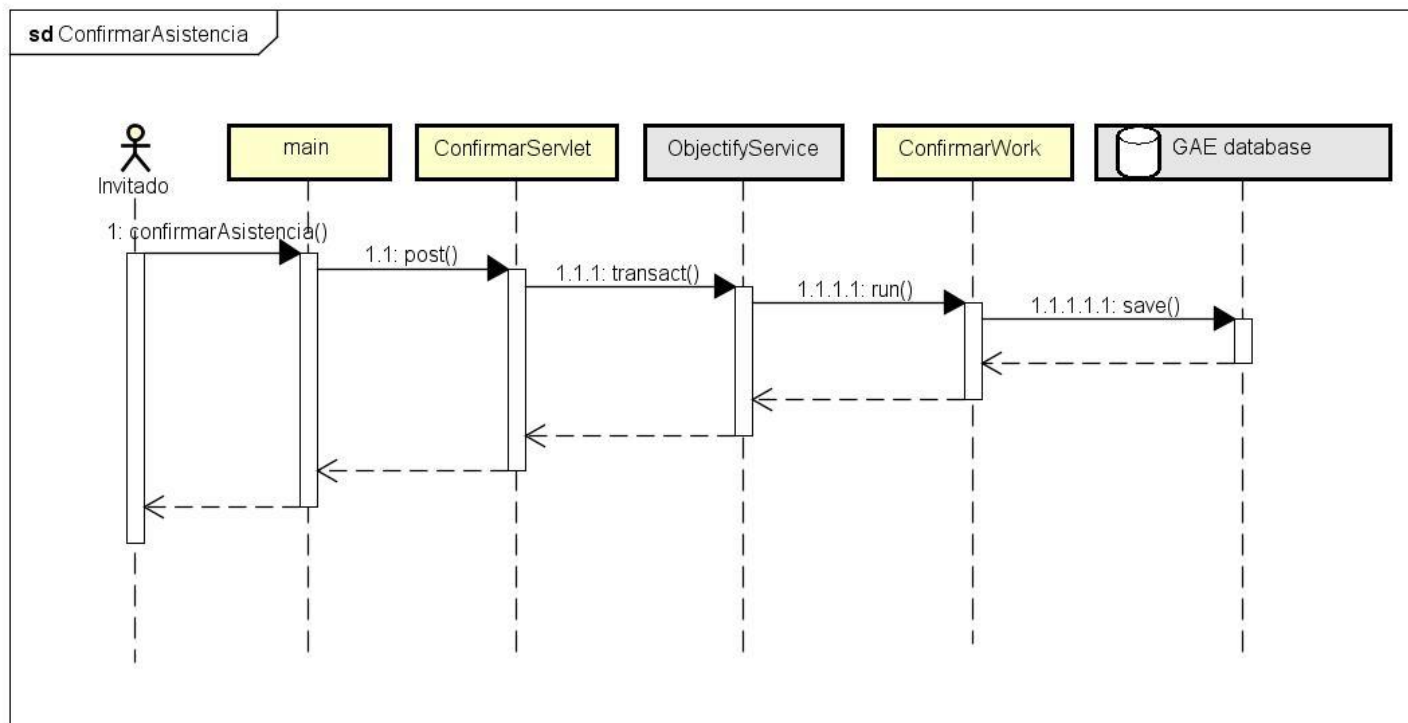
powered by Astah

Figura 2: Diagrama de Clases.

La clase OfyHelper registra las entidades Evento y Persona en la base de datos. “Evento” se utiliza como clave padre de todas las entidades Persona que se van a registrar en la base de datos. Las clases ConfirmarServlet y ConsultarController se comunican con la base de datos para ejecutar las transacciones de alta y consulta.

VISTA DE PROCESOS

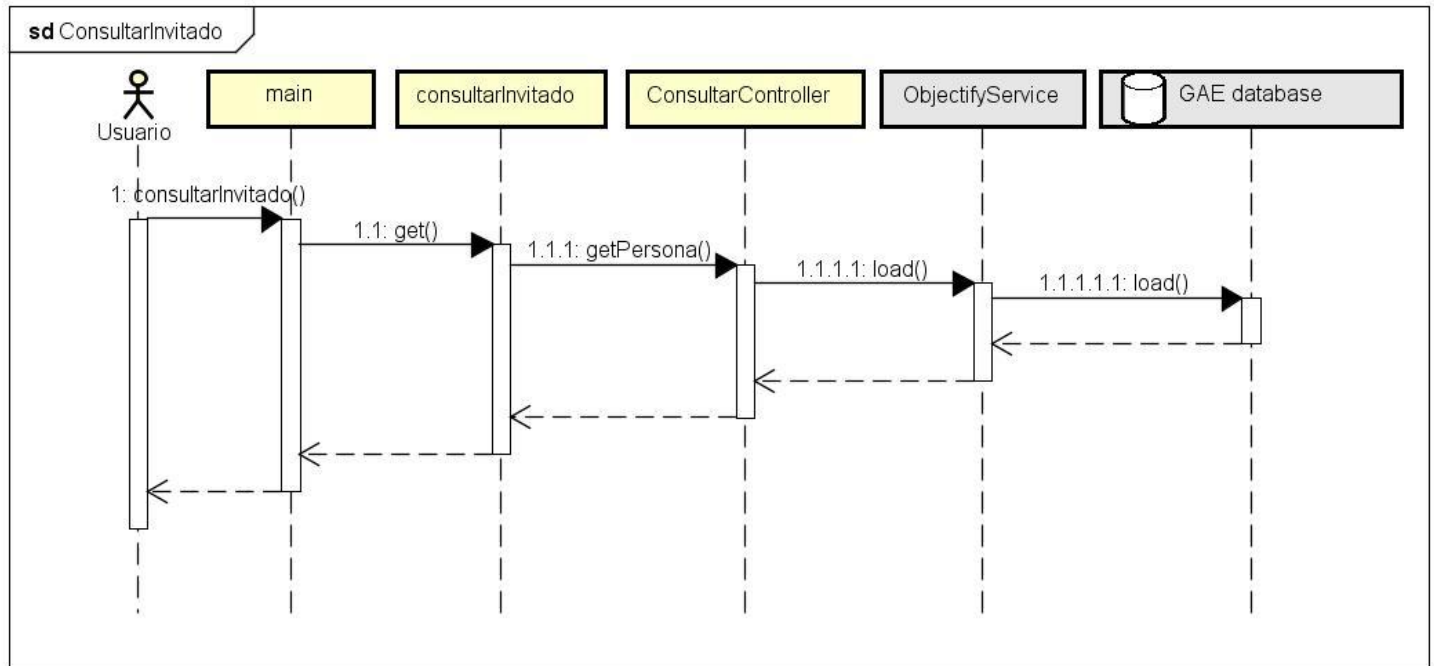
A través de diagramas de secuencia, se puede modelar el flujo de la información y los datos en el sistema. En la figura 3 se muestra la secuencia cuando un invitado se registra al evento. En gris claro se indican las entidades no propias.



powered by Astah

Figura 3: Diagrama de Secuencias, Confirmar Asistencia.

En el siguiente diagrama de secuencias se describen las acciones en el sistema realizadas cuando un usuario consulta por la confirmación de un invitado a través de su mail. Nuevamente en gris las entidades utilizadas para el servicio de Google.

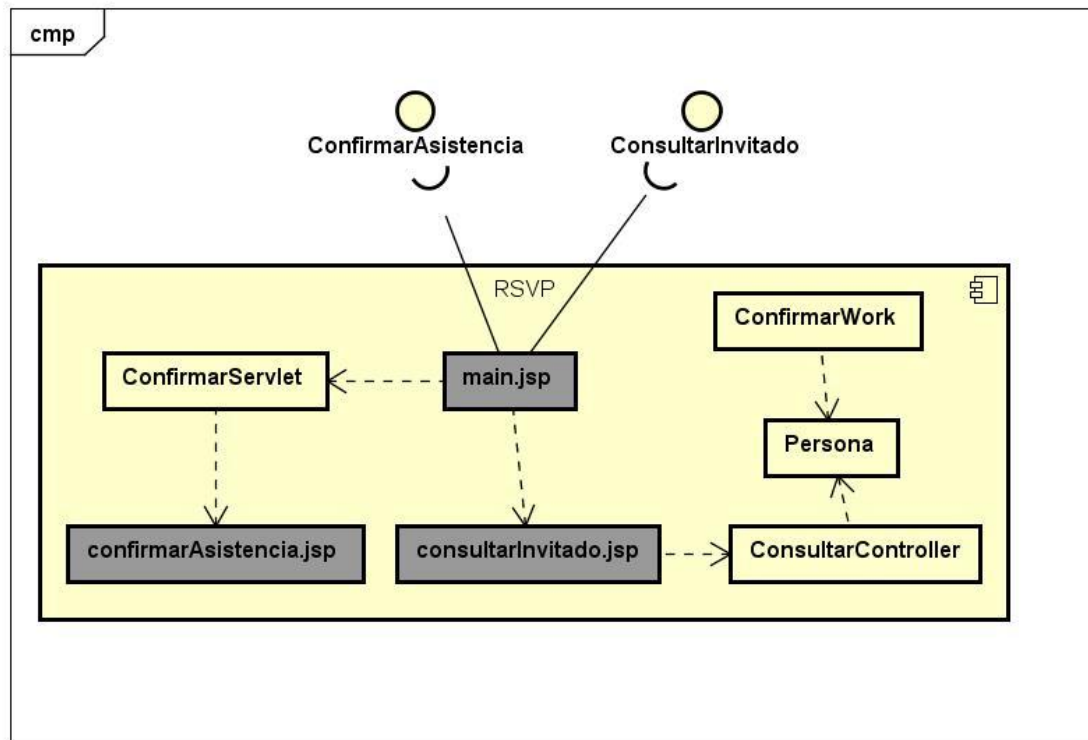


powered by Astah

Figura 4: Diagrama de Secuencia, consulta estado.

VISTA DE IMPLEMENTACIÓN

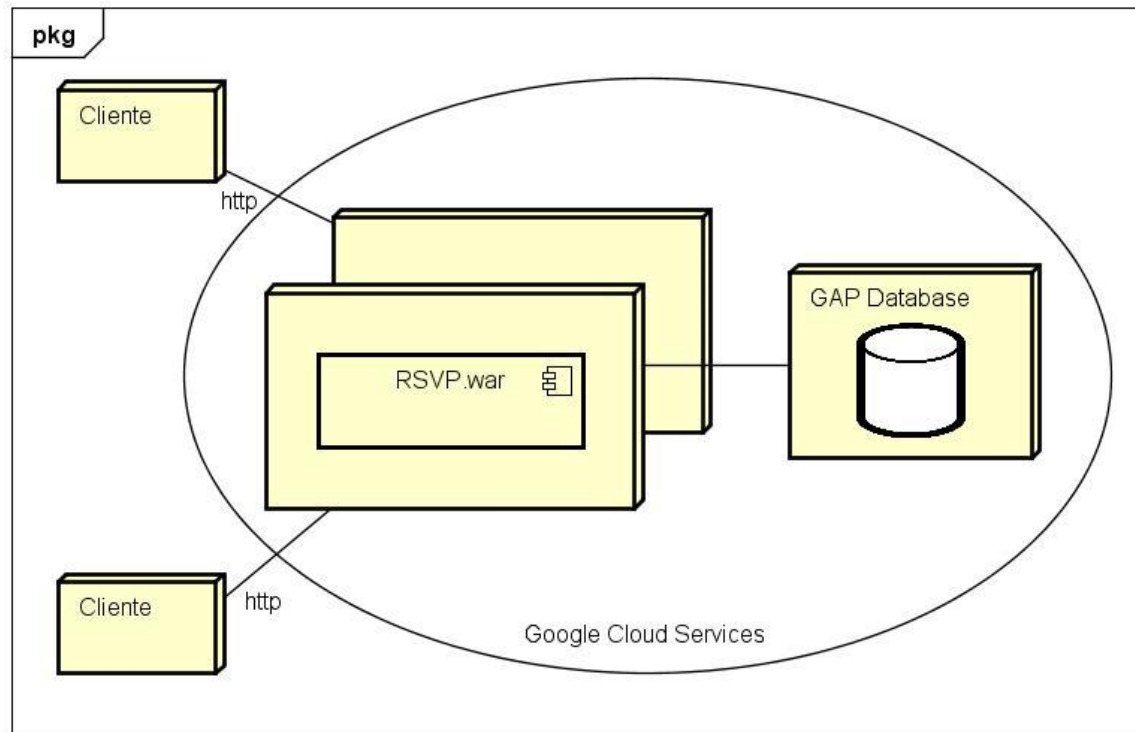
Mediante el diagrama de componentes se puede visualizar las interfaces que provee el sistema, en qué componente se realiza y a través de qué entidad.



powered by Astah

VISTA DE DESPLIEGUE

Mediante el siguiente diagrama de despliegue se puede visualizar la arquitectura y el rol de los servicios Cloud para el acceso y utilización de distintos clientes de la aplicación.



powered by Astah

Figura 10: Diagrama de Despliegue de la aplicación

PRUEBAS DE CARGA

Para las pruebas de carga se utilizó la herramienta “ab - Apache HTTP server benchmarking tool” (<https://httpd.apache.org/docs/2.2/programs/ab.html>), que permite realizar POST requests, indicando los parámetros del POST, la cantidad de requests a efectuar, cuántos realizar concurrentemente, e imprimir un archivo .csv con los porcentajes de los tiempos que tomó realizar esos requests.

Entre las herramientas conocidas (wget, JMeter), ésta es una de las más sencillas de usar, y de éstas, la más completa.

Ejemplo de corrida de la prueba: (1 request)

```
Server Software:      Google
Server Hostname:      www.rsvp-1092.appspot.com
Server Port:          80

Document Path:        /
Document Length:      762 bytes

Concurrency Level:    1
Time taken for tests:  0.325 seconds
Complete requests:    1
Failed requests:       0
Total transferred:    904 bytes
HTML transferred:     762 bytes
Requests per second:  3.08 [#/sec] (mean)
Time per request:     324.584 [ms] (mean)
Time per request:     324.584 [ms] (mean, across all concurrent requests)
Transfer rate:        2.72 [Kbytes/sec] received

Connection Times (ms)
      min mean[+/-sd] median max
Connect:  43  43  0.0   43   43
Processing: 281 281  0.0  281  281
Waiting:   277 277  0.0  277  277
Total:     324 324  0.0  324  324
```

Para correr las pruebas, ejecutar el comando prueba_carga.sh con los parámetros necesarios.