

Geolocalización de Sitios de Interés Para Aplicaciones Móviles

G-SIAM



Plan de Pruebas

VERSIÓN

1.1



Contenido

1	<u>INTRODUCCIÓN.....</u>	3
1.1	MISIÓN	3
1.2	OBJETIVO	3
2	<u>TIPOS DE PRUEBAS</u>	3
2.1	PRUEBAS UNITARIAS DE LOS SERVICIOS	4
2.2	PRUEBAS FUNCIONALES.....	4
2.3	PRUEBAS SOBRE LA DOCUMENTACIÓN.....	4
3	<u>ESTRATEGIA DE PRUEBAS.....</u>	4
3.1	METODOLOGÍA DE PRUEBA	4
3.2	PROCESO Y CICLOS DE PRUEBAS DEL PROYECTO	4
3.3	CASOS DE PRUEBA.....	5
4	<u>HERRAMIENTAS</u>	6
4.1	REGISTRO Y SEGUIMIENTO DE DEFECTOS	6



1 Introducción

1.1 Misión

El propósito de las pruebas es corroborar que los requerimientos planteados en el documento de especificación de requerimientos (ERS) son cubiertos por el sistema.

El plan de pruebas tiene como misión definir las pautas para poder cumplir con el propósito de las pruebas de manera exitosa.

1.2 Objetivo

El objetivo de este plan de prueba es documentar la estrategia y el enfoque de todas las fases de prueba correspondientes al proyecto de grado G-SIAM, a fin de garantizar el cumplimiento de la misión especificada anteriormente.

Mediante este plan se especifica el tipo de pruebas que se realizan, para verificar el comportamiento esperado del sistema, y la adecuación a los estándares de calidad definidos tanto para el sistema como para la documentación. Se define también cuando y como se realizarán.

Lo que se busca al ejecutar distintos tipos de pruebas de forma correcta es:

- Identificación temprana de la mayor cantidad de defectos, con el fin de reducir costos.
- Encontrar problemas importantes para minimizar riesgos.
- Verificar que el sistema cumple con el ERS.
- Dejar constancia de errores y problemas mediante documentación.
- Constatar la adecuación a estándares.
- Comprobar la calidad del sistema.

2 Tipos de Pruebas

Dadas las características del proyecto el Equipo define realizar tres tipos de pruebas:

- pruebas unitarias de los servicios (Caja Negra). Mediante JUnit
- pruebas funcionales (Caja negra). Se realizarán pruebas sobre las interfaces finales del sistema para comprobar el funcionamiento esperado
- Se contará con pruebas relacionadas a la documentación y el código.



2.1 Pruebas unitarias de los servicios

Para automatizar las pruebas de los servicios se realizarán pruebas unitarias de estos con el framework Junit junto a una librería para facilitar el testeo de servicios REST llamada REST-assured. Las pruebas se ejecutan desde el código fuente.

2.2 Pruebas Funcionales

El objetivo de las pruebas funcionales o de caja negra es verificar que el sistema satisface todos los requerimientos (ERS) y funciona según lo especificado en los Casos de Uso. Básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida de una funcionalidad en particular.

2.3 Pruebas sobre la documentación

La finalidad de estas pruebas es verificar que la documentación y el código generado cumplan con los estándares definidos.

3 Estrategia de pruebas

3.1 Metodología de Prueba

Las pruebas de Caja Negra las realiza el responsable de testing cuando el programador realiza una entrega (parcial o total) del programa. Dado el tamaño del proyecto estas pruebas serán cruzadas, es decir, al ser solo dos integrantes, la persona que construya un CU testeara el CU de la 2da persona y viceversa.

Las pruebas funcionales o de Caja Negra estarán basadas en casos de pruebas.

Las pruebas sobre la documentación podrán ejecutarse opcionalmente al finalizar cada iteración, pero si será obligatorio realizar una prueba completa antes de finalizar el proyecto.

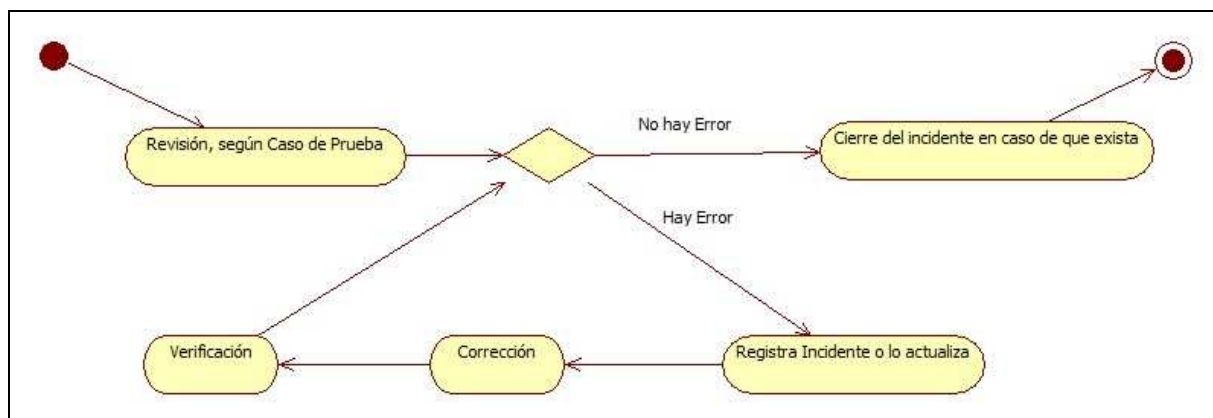
3.2 Proceso y ciclos de pruebas del proyecto

El primer paso de este proceso será diseñar los casos de pruebas definidos para el proyecto (Ver punto 3.3), luego se tendrán que preparar los datos de prueba, con esto ya se podrá ejecutar la aplicación para posteriormente comparar los resultados obtenidos con los esperados escritos en los casos de prueba. Al terminar la ejecución se debe documentar apropiadamente en la ficha si la prueba se da como exitosa o no, en caso de que se registre un error, se deberá



crear una incidencia en el sistema de tickets (Issue Tracking de GoogleCode). Una vez corregidos los errores, se debe ejecutar nuevamente la prueba y registrar los resultados. Si hay éxito, se ejecutan los cambios, en caso negativo nuevamente deben documentarse los errores.

A continuación se representa gráficamente el proceso:



3.3 Casos de Prueba

A continuación se muestra un caso de prueba a modo de ejemplo:

ID CU	Nombre	ID CP	Requisitos	Descripción del caso de prueba	Datos de prueba	Valor esperado	Resultado (ok, error)	Observaciones	Responsable	Validación
CU01	Ingreso del usuario	CP01-01	Ya existe el usuario gsiam.test@gmail.com en el sistema (CP02-01).	El usuario ingresa datos correctos.	e-mail: gsiam.test@gmail.com contraseña: 123	El usuario ingresa a la aplicación y se muestra la pantalla principal junto a su nombre (Gsiam Test).	OK		Martin Loureiro	

ID CU: Identificador de caso de uso al cual se le ejecutara el caso de prueba

Nombre: Nombre del caso de prueba.

ID CP: Identificador único del caso de prueba.

Requisitos: Precondiciones para la correcta ejecución del caso de prueba.

Descripción del caso de prueba: Breve comentario de lo que debe hacer el caso de prueba.

Datos de prueba: Datos para poder ejecutar el caso de prueba.

Valor esperado: Valor correcto esperado después de la ejecución del caso de prueba.

Resultado: Resultado correcto (ok) no correcto (error).

Observaciones: Comentarios u observaciones del caso de prueba.

Responsable: Ejecutor del caso de prueba.

Validación: Validación de caso de prueba.



4 Herramientas

4.1 Registro y Seguimiento de defectos

Como se mencionó previamente se deben crear incidentes con los defectos encontrados. Para esto se utiliza el sistema de manejo de tickets que viene incorporado al Google Code llamado Issue Tracking. El mismo contiene un título donde debe describirse el problema en pocas palabras. Luego se detalla el problema lo mejor posible, se pueden adjuntar archivos. Al ticket se le asigna una prioridad, un estado, el tipo de defecto, un mail de copia y se indica a que módulo del sistema afecta.

Los posibles estados son: New, Accepted, Started, Fixed, Verified, Invalid, Duplicate, WontFix y Done.

Las posibles prioridades son: Critical, High, Medium, Low.

A continuación se muestran imágenes de la herramienta:

ID	Type	Status	Priority	Milestone	Owner	Summary + Labels
2	Defect	Accepted	Medium	----	Porcelli,Antonio	Error en el login cuando no encuentra el servicio
3	Defect	Accepted	Medium	----	Porcelli,Antonio	Error en icono al crear sitio
4	Defect	Accepted	Medium	----	Porcelli,Antonio	Error al acceder a la cámara de android
5	Defect	Accepted	Medium	----	mloure	Error al redireccionar despues de crear un usuario
7	Defect	Accepted	Critical	----	Porcelli,Antonio	Error al mostrar sitios
8	Defect	Accepted	Critical	----	mloure	Error al crear un sitio
9	Defect	Accepted	Critical	----	Porcelli,Antonio	Error al publicar



geolocalizacion-uas
G-SIAM

Project Home Downloads Wiki **Issues** Source Administer

New issue Search Open issues for Search Advanced search Search tips

★ **Issue 2: Error en el login cuando no encuentra el servicio**
1 person starred this issue and may be notified of changes.

Status: Accepted
Owner: Porcelli Antonio
Cc: mloure
Type: Defect
Priority: Medium

Reported by project member Porcelli Antonio, Oct 8, 2011

Al ingresar a la aplicación y si el servicio esta levantado pero esta mal formada la url esto da un mulpointerexception por que el parametro error del login servicio no puede ser recuperado en el loginActivity.

Add a comment and make changes below

Add a comment and make changes

Enter your comments and make changes

El ingreso a la herramienta es mediante la siguiente url:

<http://code.google.com/p/geolocalizacion-uas/issues/list>