

Informe final de Testing

Glamp House

Grupo 1:

Stancic Victoria.

Velazquez Laura.

Peralta Laura.

Vargas Rocio.

Garzón Deisy.

Piazzze Rocio.

Contenido

Introducción:

Resumen de las actividades de prueba.

Alcance:

Alcance.

Tipos de pruebas ejecutadas.

Enfoque de la Prueba.

Exit Criteria

Resumen de Resultados:

Diseño de Pruebas.

Ejecución de Pruebas.

Ejecución Manual.

Ejecución Automática.

Reporte de Defectos.

Todos los defectos.

Defecto por Estado.

Defectos Creados y Resueltos.

Defectos Abiertos.

Lecciones Aprendidas / Conclusión.

Introducción:

Este documento es el informe final de testing de “Glamp House” grupo uno. El propósito es proporcionar evidencia de que el Exit Criteria para el proceso de Testing se cumplió y por lo tanto, se concluye la fase de pruebas y puede cerrarse. Se demuestra que los issues de GitLab relacionados con Testing/QA fueron implementados desde el sprint 1 al sprint 4. Este documento va a ser utilizado como entrada para la revisión general de las actividades de prueba y para tomar la decisión si el sistema cumple con las expectativas.

Resumen de las actividades de prueba

Resumen del sistema:

Glamp House es una aplicación web especializada en la búsqueda y reserva de alojamientos que combina la experiencia de acampar al aire libre con el lujo y las condiciones propias de los mejores hoteles en Argentina y Colombia en diferentes zonas. Es por ello, que consta con varios tipos de servicios obtenidos a través de pestañas que ayudarán a los diferentes roles ya sean administrador, anónimo o registrado; a buscar diferentes tipos de categorías, filtrar los resultados, crear una cuenta e ingresar a la misma, ver el detalle de cada producto y reservar alguno en determinada fecha o así mismo, cargarlo al sistema. Todo esto proporcionado gracias a una API conectada a una base de datos, que a la vez pertenece en conjunto con toda la pieza de software a una infraestructura cloud en AWS.

Los invitamos a navegar por nuestra página <http://glamphouse.com.ar/>.

Link de bucket:<http://glamphouse.s3-website.us-east-2.amazonaws.com/>

Link de Planilla de Casos de Prueba y Planilla de Defectos:

 **Glamp House - Testing Casos de prueba.xlsx**

Link Testing Exploratorio:

Sprint 2:  **Sesiones de testing exploratorio - Sprint 2**

Sprint 3:  **Sesiones de testing exploratorio - Sprint 3**

Sprint 4:  **Sesiones de testing exploratorio Sprint 4**

Sprint 1:

Testear la API #13 · created 1 month ago by Santiago Vallazza · Sprint 1 · Obligatoria · Testing / QA	CLOSED · 0 updated 1 month ago
Planificación y ejecución de los tests #12 · created 1 month ago by Santiago Vallazza · Sprint 1 · Obligatoria · Testing / QA	CLOSED · 1 updated 1 month ago

Sprint 2:

Testeo Exploratorio - Manual - Estático #49 · created 1 month ago by Santiago Vallazza · Sprint 2 · Obligatoria · Testing / QA	CLOSED · 10 updated 2 weeks ago
Testeo Automatizado (Postman) #48 · created 1 month ago by Santiago Vallazza · Sprint 2 · Obligatoria · Testing / QA	CLOSED · 0 updated 3 weeks ago
Testeo Automatizado (Selenium) #47 · created 1 month ago by Santiago Vallazza · Sprint 2 · Electiva · Testing / QA	CLOSED · 0 updated 2 weeks ago

Sprint 3:

Implementar tests automatizados	CLOSED	1
#74 · created 3 weeks ago by Santiago Vallazza	🕒 Sprint 3	👤 1
Obligatoria	Testing / QA	updated 1 week ago
Implementar tests manuales	CLOSED	10
#73 · created 3 weeks ago by Santiago Vallazza	🕒 Sprint 3	👤 10
Obligatoria	Testing / QA	updated 1 week ago

Sprint 4:

Confeccionar el reporte final de testing		0
#83 · created 1 week ago by Santiago Vallazza	🕒 Sprint 4	👤 0
En progreso	Obligatoria	Testing / QA
Implementar testing manual		1
#82 · created 1 week ago by Santiago Vallazza	🕒 Sprint 4	👤 1
En progreso	Obligatoria	Testing / QA
Implementar testing automatizado		0
#81 · created 1 week ago by Santiago Vallazza	🕒 Sprint 4	👤 0
En progreso	Obligatoria	Testing / QA

Alcance

Dentro del Alcance

En cuanto al testeo manual se realizaron casos de prueba de todos los issues de los sprints. Se crearon tests unitarios utilizando Selenium IDE y se desarrollaron los scripts de pruebas automatizados utilizando Postman para las API desarrolladas.

Al finalizar con los requerimientos de la página se realizó un testeo exploratorio y reporte de defectos.

A medida que se fue cumpliendo con los requerimientos el encargado del mismo verificó el correcto funcionamiento como por ejemplo en cuanto a la seguridad se verificó que funcione correctamente el almacenado con las contraseñas encriptadas, con el calendario se verificó que se marque el rango de fechas, etc.

Planificación y ejecución de los tests

En esta issue se solicitó “Planear casos de test basados en las historias de usuarios que existen en el sprint, y luego ejecutarlos” se trabajó en un archivo excel con distintas hojas, la primera hoja “Casos de prueba - S1” en el cual detallamos cada caso de prueba con un “ID” (que tiene el siguiente formato “FE-01/BE-01”, en el cual, FE/BE equivale al tipo de prueba Frontend o Backend, “01” número de prueba; también cuenta con columnas como “Sprint” equivale al N.º de sprint correspondiente, “Fecha de creación” que es cuando se crea el caso de prueba, “Título”, en el cual damos un título al caso de prueba; “Descripción”, en el que damos una breve descripción de lo que consiste el caso de prueba; “Precondición” donde detallamos las precondiciones para llevar a cabo el caso de prueba; “Acción”, en la que enumeramos los pasos a seguir para llevar a cabo la prueba; “Resultado esperado”, que muestra lo que se espera al ejecutar cada paso de la columna Acción; “Reportado por”, que indica la persona que desarrolló el caso de prueba; “Tipo de Prueba” detalla el tipo de prueba; “Estado”, que especifica el estado en el que se encuentra el diseño del caso de prueba (puede ser, “TO DO”, “In progress”, “Reviewed”, “In review”).

En este sprint se llevaron a cabo las pruebas manuales completando este archivo.

Sprint	ID	Fecha de creación	Título	Descripción	Área dev	Precondición	Pasos			Estado	Tipo de prueba	Reportado por
							#	Acción	Resultado esperado			
1	FE-01	5/8/2022	Header	Verificar la posición y ancho del header que se ve en el HTML y coincidan con las que pide en el Figma y peticiones.	FRONT	Tener acceso a la página	1	Ingresar a la página.	El header debe estar en posición superior y tener un ancho 100% en todos los dispositivos.	Reviewed	Aceptación	Victoria Stancic
1	FE-02	5/8/2022	Header con logo y lema.	Presencia del header con logo y lema.	FRONT	Tener acceso a la página	1	Ingresar a la página.	El logo y lema tiene que estar en el header en la parte izquierda y al clickarlo vaya al home.	Reviewed	Aceptación	Victoria Stancic

Además, en este sprint se solicitó “Realizar también una prueba exploratoria sobre lo que se desarrolló durante el Sprint” para eso, se exploró la página observando lo que fue pedido, las cosas que funcionaban y lo que había que mejorar, reportando defectos en la planilla.

Sprint	Fecha	Datos de prueba	Estado de Ejecución
1	11/08/2022	Implementar template general responsive.	Passed
1	11/08/2022	API: Agregar endpoint de categorías.	Passed
1	11/08/2022	Implementación template bloque buscador.	Failed
1	11/08/2022	Implementar el template del bloque listado.	Passed
1	11/08/2022	Implementar el template del bloque categorías.	Failed
1	11/08/2022	Implementar el template de login y registro.	Passed
1	11/08/2022	Evento de login de usuario.	Passed
1	11/08/2022	Implementar evento de header.	Passed

Fuera de Alcance

El testeo se limitó a lo requerido en GitLab por lo tanto quedó sin realizar la implementación de framework JEST.

Tipos de Pruebas Ejecutadas

Tipos de pruebas Ejecutadas	Sprint 1	Sprint 2	Sprint 3	Sprint 4
Prueba Estática	SI	SI	SI	SI
Prueba Exploratoria	SI	SI	SI	SI
Prueba de humo	SI	SI	SI	SI
Prueba de regresión	SI	SI	SI	SI
Prueba automatizada con Selenium	NO	SI	SI	SI
Prueba automatizada con Postman	SI	SI	SI	SI

Enfoque de la Prueba

Se crearon test funcionales tanto positivos cómo negativos para cada funcionalidad requerida. En cada sprint al recibir requerimientos nuevos desde desarrollo se ejecutaban las pruebas de humo y luego las pruebas correspondientes a las nuevas funcionalidades. En todos los sprints se crearon los scripts necesarios para automatizar el testing de la API en Postman.

Se ejecutó la suite de pruebas de regresión completa, se reportaron los defectos encontrados en la planilla correspondiente y se realizó las sesiones de prueba exploratoria basándose en las issues de GitLab.

Exit Criteria

Se definió los siguientes criterios de aceptación para finalizar las pruebas:

- No se debe tener defectos en estado abierto de severidad crítica.
- La ejecución de los Test de Postman deben tener un 100% de reviewed.
- Los casos de prueba deben tener un 90% de reviewed.
- Los testeos de Selenium IDE deben tener una cobertura mínima del 40%.

Resumen de Resultados

Diseño de Pruebas

Tests realizados por sprint:

	Test Manuales	Test Automáticos (Selenium)	Test de Integración (Postman)	Test Total
Sprint 1	91	0	5	96
Sprint 2	38	10	23	71
Sprint 3	33	9	32	74
Sprint 4	30	5	38	73

Reporte de Defectos

Todos los defectos

La siguiente sección muestra información con respecto al número total de defectos que se han presentado durante la duración de la fase de prueba.

Defectos Abiertos

La siguiente sección muestra información con respecto al número total de defectos que permanecen abiertos al final de la fase de pruebas.

Defectos creados y resueltos

DEFECTOS	
Estado	Cantidad
No resuelto	17
Resuelto	21
TOTAL	38

Al momento de entregar este informe quedan 11 defectos de severidad baja, 4 de severidad media y 1 de severidad alta sin resolver.

Lecciones Aprendidas / Conclusión

El proceso de testing nos permitió mejorar en cada una de las áreas de trabajo, analizar defectos para mejorar en lo estético y funcional, además de aprender a comunicarnos y escucharnos para poder lograr una aplicación de calidad.