

CASOS DE PRUEBAS

TC (Test Cases)

Nombre del documento	Plan de Pruebas
Nombre del fichero:	<i>ISST-TC-V19.docx</i>
Versión:	<i>1</i>
Fecha de entrega:	<i>06/05/2024</i>
Caso de estudio:	<i>Ocio con mascotas</i>
Entrega	<i>Sprint 3</i>

Autores/as: Alejandro Muñoz, Marina Lumbreras, Ignacio Diez
--

Revisor/a: Marina Lumbreras

INDICE

1	INTRODUCCIÓN	1
1.1	Propósito del documento.....	1
1.2	Estructura del documento	1
2	PLAN DE PRUEBAS.....	2
2.1	Visión global de las pruebas	2
2.2	Recursos, personal y productos de las pruebas.....	2
2.3	Control y seguimiento de las pruebas.....	2
2.4	Métricas, herramientas y entorno de pruebas	2
2.5	Calendario de pruebas	2
3	ESPECIFICACIÓN CASOS DE PRUEBA	2
3.1	Caso de prueba en Postman.....	2
3.1.1	<i>Descripción</i>	2
3.1.2	<i>Recursos</i>	3
3.1.3	<i>Precondiciones</i>	3
3.1.4	<i>Datos de entrada</i>	3
3.1.5	<i>Postcondiciones</i>	5
3.1.6	<i>Flujo de eventos</i>	6
3.1.7	<i>Resultado</i>	6
3.2	Caso de prueba en Selenium.....	7
3.2.1	<i>Descripción</i>	7
3.2.2	<i>Recursos</i>	7
3.2.3	<i>Precondiciones</i>	7
3.2.4	<i>Datos de entrada</i>	8
3.2.5	<i>Postcondiciones</i>	9
3.2.6	<i>Flujo de eventos</i>	11
3.2.7	<i>Resultado</i>	11

1 INTRODUCCIÓN

1.1 Propósito del documento

Este documento de casos de prueba establece el alcance y los objetivos de las pruebas para validar el correcto funcionamiento de la aplicación. Contiene una lista de casos de prueba diseñados para cubrir diversos escenarios y funcionalidades. Su propósito es garantizar la calidad del software mediante pruebas exhaustivas que abarcan diferentes situaciones de uso. Facilita la detección temprana de errores y su corrección, asegurando la entrega de un producto confiable y libre de fallos.

1.2 Estructura del documento

El documento sigue una estructura que abarca la visión general del proyecto, la descripción detallada de los casos de prueba en Postman y Selenium, así como sus resultados. Incluye secciones como Recursos, Precondiciones, Flujo de eventos, y Resultado, proporcionando un análisis exhaustivo de la calidad y fiabilidad del software. Cada sección aborda aspectos específicos de las pruebas, desde la planificación hasta la ejecución y evaluación.

2 PLAN DE PRUEBAS

2.1 Visión global de las pruebas

El proyecto consiste en una aplicación web diseñada para gestionar reseñas de usuarios sobre tiendas y servicios. La estrategia de pruebas se enfoca en garantizar que todas las funcionalidades del sistema cumplan con los criterios de aceptación de historias de usuario, así como con los requisitos no funcionales y las reglas de dominio establecidas. Se realizarán pruebas de validación y de sistema para asegurar la calidad y fiabilidad del software.

2.2 Recursos, personal y productos de las pruebas

- **Recursos:** Se utilizarán herramientas como Postman para pruebas de servicio y Selenium para pruebas de sistema. Además, se necesitará acceso a una base de datos de prueba que refleje el entorno de producción.
- **Personal:** El equipo de pruebas estará compuesto por miembros del equipo de desarrollo encargados de diseñar y ejecutar los casos de prueba, así como de analizar los resultados obtenidos.
- **Productos:** Los resultados de las pruebas incluirán informes detallados con los casos de prueba ejecutados, los defectos identificados y su estado, así como cualquier observación relevante sobre el comportamiento del sistema.

2.3 Control y seguimiento de las pruebas

Los resultados de las pruebas serán recogidos por el equipo de desarrollo, quienes asignarán tareas para resolver los defectos identificados. El equipo de desarrollo revisará y corregirá los problemas reportados, y se realizarán nuevas ejecuciones de pruebas para verificar la corrección de las soluciones implementadas.

2.4 Métricas, herramientas y entorno de pruebas

Las herramientas de pruebas incluirán Postman y Selenium, mientras que el entorno de pruebas replicará el entorno de producción lo más fielmente posible, utilizando datos de prueba realistas.

2.5 Calendario de pruebas

Las pruebas serán realizadas en los últimos cinco días del proyecto, una vez la aplicación realice todas las funcionalidades requeridas. Si en esos cinco días se descubriese algún error sería rápidamente gestionado por parte del equipo de desarrollo.

3 ESPECIFICACIÓN CASOS DE PRUEBA

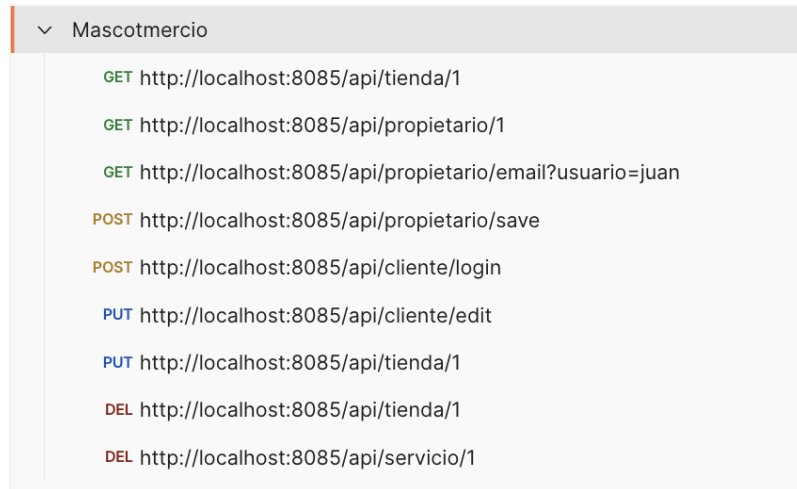
3.1 Caso de prueba en Postman

Se realiza una prueba utilizando Postman para comprobar el funcionamiento de la API. Se comprueban una serie de peticiones http de tipo Get, Put, Delete y Update para asegurar una correcta actualización de la base de datos utilizando estos métodos.

3.1.1 Descripción

La prueba se realiza utilizando la herramienta de Postman en la que se han importado una serie de peticiones http que permiten probar el funcionamiento de la API. Han sido seleccionadas una serie de muestras representativas de la aplicación para comprobar distintas perspectivas tanto de modelos como de métodos http. Dado el gran número de

métodos http se han seleccionado para la prueba algunos de los que hemos considerado más representativos, estos se muestran a continuación.



Algunas de las historias de usuario con relación en la prueba pueden ser:

- Como usuario poder realizar un registro en la web para poder crear establecimientos.
- Como cliente poder registrarte en la web para poder acceder a las funcionalidades de la misma.
- Como propietario tener la capacidad de crear, actualizar y borrar tiendas para publicarlás en la plataforma.

La prueba se realiza en 3 rondas en las que se ejecutan todas las peticiones http que se muestran en la imagen superior. Además cada petición nos devolverá información acerca del estado de la misma y el tiempo que ha tardado en realizarse, el cual tiene que ser menor de 200ms por petición.

3.1.2 Recursos

Personas involucradas, herramientas, recursos hardware o software específicos, etc.

Los recursos utilizados para la prueba son varios, el uso de visual studio para ejecutar Spring Boot con la aplicación en el puerto 8085, al que se realizarán las peticiones. También, se utiliza la herramienta de Postman desde la cual se realizan todas las peticiones en serie. Por último se ha utilizado Swagger para crear las peticiones individuales de cada solicitud http.

3.1.3 Precondiciones

En la prueba se van a verificar dos cosas, por un lado que el estado de la solicitud sea 200 OK, es decir, que ha funcionado correctamente y no ha tenido ningun problema, y por otro lado, que la solicitud no tarda mas de 200ms en procesarse, tiempo más que suficiente para que una solicitud se haya completado con éxito.

3.1.4 Datos de entrada

Se introducen las diferentes solicitudes http mencionadas anteriormente para cada uno de los métodos de la imagen:

GET Mascotmercio / http://localhost:8085/api/tienda/1



Response Headers Request

200 6 ms 212 B

General

URL: http://localhost:8085/api/tienda/1

Method: GET

Status Code: 200 OK

Request Headers

Content-Type: application/json

User-Agent: PostmanRuntime/7.38.0

Accept: */*

Postman-Token: 0d480a65-4085-4027-b43f-cea4577a610e

Host: localhost:8085

Accept-Encoding: gzip, deflate, br

Connection: keep-alive

PUT Mascotmercio / http://localhost:8085/api/cliente/edit



Response Headers Request

200 4 ms 366 B

Pretty



```
1 {
2   "idCliente": 1,
3   "nombre": "Nombre actualizado",
4   "apellido": "Apellido actualizado",
5   "usuario": "usuario_actualizado",
6   "contraseña": "contraseña_actualizada",
7   "email": "email_actualizado@example.com"
8 }
```

POST Mascotmercio / http://localhost:8085/api/cliente/login



Response Headers Request

200 10 ms 299 B

Pretty



```
1 {
2   "usuario": "usuario_a_loguear",
3   "contraseña": "contraseña_a_loguear"
4 }
```

DELETE Mascotmercio / <http://localhost:8085/api/tienda/1> ✕

Response

Headers

Request

200 17 ms 212 B

▼ General

URL: <http://localhost:8085/api/tienda/1>
Method: **DELETE**
Status Code: **200 OK**

▼ Request Headers

User-Agent: [PostmanRuntime/7.38.0](#)
Accept: [*/*](#)
Postman-Token: [2f839ee3-5a55-4f16-b6a1-69898e5ffebf](#)
Host: [localhost:8085](#)
Accept-Encoding: [gzip, deflate, br](#)
Connection: [keep-alive](#)


En las imágenes se muestran distintos ejemplos de métodos Get, Put, Delete y Update y los datos de header o body que se han empleado en Postman.

3.1.5 Postcondiciones

Condiciones a verificar y lista de posibles estados del sistema al concluir la ejecución del caso de prueba.

Estos son los resultados de la prueba realizada, se han realizado 3 iteraciones de la prueba con 2 tests en cada método, en total 54. Todos ellos se han completado con éxito y el tiempo medio de respuesta ha sido de 22ms. En la imagen inferior se muestran los resultados de manera gráfica de cada uno de los tests.

Mascotmercio - Run results

 Ran today at 20:00:09 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	3	14s 927ms	54	22 ms

RUN SUMMARY

		1	2	3
▼ GET	http://localhost:8085/api/tienda/1	6 0		
	Pass Status code is 200			
	Pass Response time is less than 200ms			
▶ GET	http://localhost:8085/api/propietario/1	6 0		
▶ GET	http://localhost:8085/api/propietario/email...	6 0		
▼ POST	http://localhost:8085/api/propietario/save	6 0		
	Pass Status code is 200			
	Pass Response time is less than 200ms			
▶ POST	http://localhost:8085/api/cliente/login	6 0		
▼ PUT	http://localhost:8085/api/cliente/edit	6 0		
	Pass Status code is 200			
	Pass Response time is less than 200ms			
▶ PUT	http://localhost:8085/api/tienda/1	6 0		
▼ DELETE	http://localhost:8085/api/tienda/1	6 0		
	Pass Status code is 200			
	Pass Response time is less than 200ms			
▶ DELETE	http://localhost:8085/api/servicio/1	6 0		

3.1.6 Flujo de eventos

El flujo de eventos de la prueba ha sido el siguiente:

- Consulta en Swagger de cada una de las peticiones http a realizar
- Importación a la herramienta Postman de las peticiones de swagger
- Creación de un entorno de prueba en Postman con todas las peticiones importadas
- Configuración y ejecución de las pruebas de manera individual
- Ejecución de la prueba global con las 3 rondas y 54 tests realizados
- Análisis de resultados y evaluación final.

3.1.7 Resultado

La prueba ha sido completada con un 100% de éxito. Esto se debe a que el funcionamiento de todas las peticiones http funcionan correctamente como ya se había comprobado en el anterior sprint cuando tuvimos que implementar React usando la API. Este test es de utilidad cuando se ha desarrollado el back-end y se quiere implementar un front con otra herramienta diferente que necesita acceder a la API. Las peticiones http nos permiten conocer la configuración de la aplicación para comprobar que el funcionamiento es el adecuado y no se producen problemas con el servidor.

En definitiva, esta prueba es sencilla pero efectiva y muy importante para lograr una base sólida sobre la cual poder programar un front sin problemas de funcionalidad con la base de datos, por lo que se ha completado con éxito la prueba de servicios con Postman.

3.2 Caso de prueba en Selenium

Para esta prueba a nivel de aplicación probaremos utilizando la herramienta Selenium el funcionamiento de nuestra aplicación simulando las cuatro historias de usuario que definimos en el documento de Visión.

3.2.1 Descripción

Requisitos:

- Registro de nuevo usuario.
- Publicación de negocio por parte del propietario.
- Consulta de establecimientos y ofertas por parte del cliente.
- Realización de valoración por parte del cliente.

Historias de usuario:

- Como nuevo usuario interesado en utilizar la plataforma Mascotmercio, deseo poder registrarme para acceder a las funcionalidades disponibles en la aplicación.
- Como propietario de un establecimiento, deseo poder publicar mi negocio en la plataforma Mascotmercio para poder ofrecer mis servicios a los usuarios interesados.
- Como cliente, quiero poder consultar los diferentes establecimientos y sus ofertas para poder elegir el que mejor se adecue a mis intereses
- Como cliente, quiero poder realizar una valoración a uno de los establecimientos registrados en Mascotmercio para poder compartir mi opinión sobre él con otros clientes.

En el caso de prueba se ejecutarán cinco escenarios en los cuales se simularán las cuatro historias de usuario previamente mencionadas.

3.2.2 Recursos

Los recursos utilizados para llevar a cabo esta prueba de uso han sido varios. Hemos utilizado Selenium IDE para la grabación y ejecución de pruebas automatizadas. Para usar correctamente esta herramienta era necesario usar también un navegador web compatible que en este caso ha sido Chrome.

3.2.3 Precondiciones

Para cada una de las cuatro pruebas diferentes que se van a realizar las precondiciones son distintas.

Registro de usuario: En la base de datos no exista ningún cliente ni propietario con el mismo nombre de usuario.

Consulta de establecimientos: Existe una sesión abierta ya sea como propietario o usuario y existen al menos una entrada de un establecimiento con sus respectivos servicios.

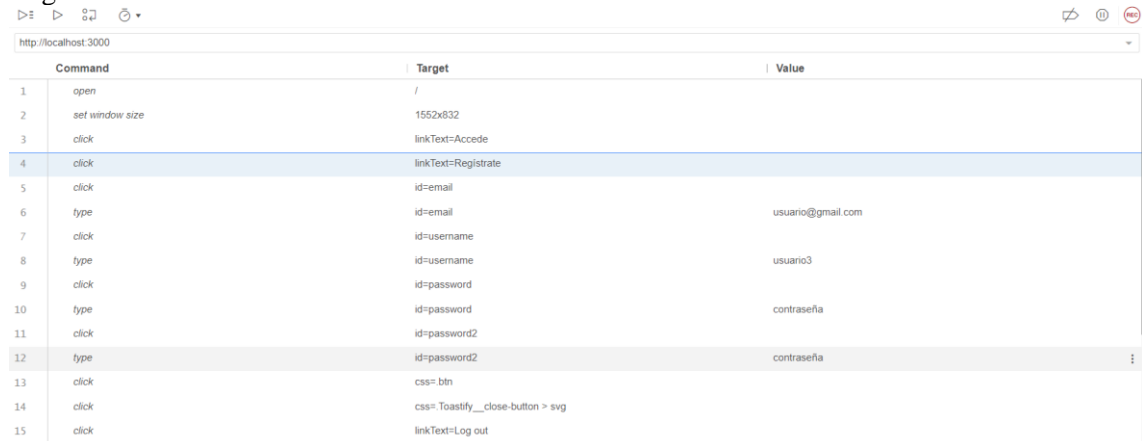
Creación de negocio: Existe una sesión abierta como propietario.

Publicación de reseña: Tiene que existir una sesión abierta como usuario y un establecimiento al que vincular la reseña

3.2.4 Datos de entrada

Para cada uno de los casos de prueba los valores de entrada son diferentes:

Registro de cliente:

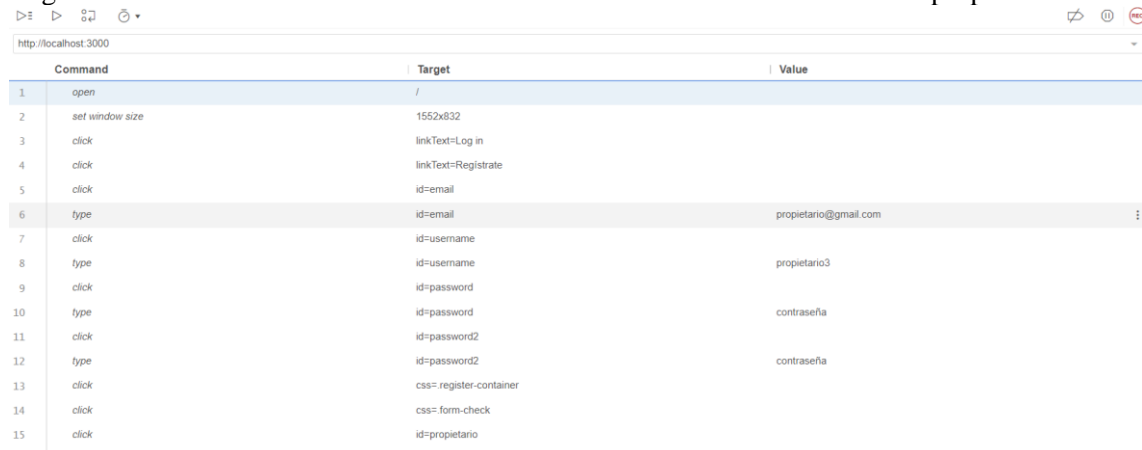


	Command	Target	Value
1	open	/	
2	set window size	1552x832	
3	click	linkText=Accede	
4	click	linkText=Registrate	
5	click	id=email	
6	type	id=email	usuario@gmail.com
7	click	id=username	
8	type	id=username	usuario3
9	click	id=password	
10	type	id=password	contraseña
11	click	id=password2	
12	type	id=password2	contraseña
13	click	css= btn	
14	click	css= Toastify__close-button > svg	
15	click	linkText=Log out	

Registro

de

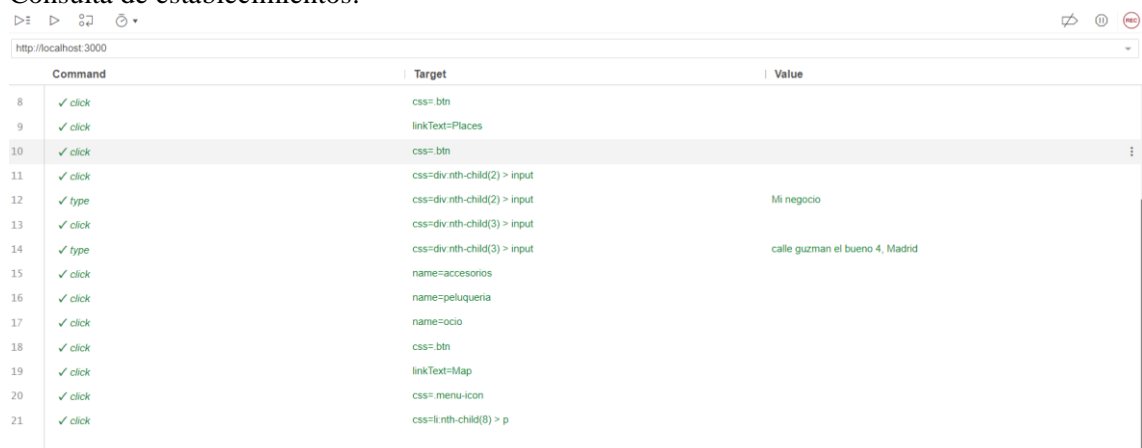
propietario:



	Command	Target	Value
1	open	/	
2	set window size	1552x832	
3	click	linkText=Log in	
4	click	linkText=Registrate	
5	click	id=email	
6	type	id=email	propietario@gmail.com
7	click	id=username	
8	type	id=username	propietario3
9	click	id=password	
10	type	id=password	contraseña
11	click	id=password2	
12	type	id=password2	contraseña
13	click	css= register-container	
14	click	css= form-check	
15	click	id=propietario	

La diferencia entre los dos registros es el comando en el que se indica que la casilla con id "propietario" ha sido marcada.

Consulta de establecimientos:



	Command	Target	Value
8	✓ click	css= btn	
9	✓ click	linkText=Places	
10	✓ click	css= btn	
11	✓ click	css=div:nth-child(2) > input	
12	✓ type	css=div:nth-child(2) > input	Mi negocio
13	✓ click	css=div:nth-child(3) > input	
14	✓ type	css=div:nth-child(3) > input	calle guzman el bueno 4, Madrid
15	✓ click	name=accesorios	
16	✓ click	name=peluqueria	
17	✓ click	name=ocio	
18	✓ click	css= btn	
19	✓ click	linkText=Map	
20	✓ click	css= menu-icon	
21	✓ click	css=li:nth-child(8) > p	

Después de iniciar sesión como propietario, el escenario simula la creación de un establecimiento que tenga como nombre "Mi negocio" y que se encuentre en la dirección "Guzmán el Bueno 4, Madrid" y ofrezca tres servicios.

http://localhost:3000			
	Command	Target	Value
7	type	id=password	contraseña
8	click	css= btn	
9	click	css= marker:nth-child(8)	
10	mouse down at	css= mapboxgl-canvas	703,329 1500015258789
11	mouse move at	css= mapboxgl-canvas	703,329 1500015258789
12	mouse up at	css= mapboxgl-canvas	703,329 1500015258789
13	click	css= busqueda-container > input	
14	type	css= busqueda-container > input	Tienda 23
15	send keys	css= busqueda-container > input	\$(KEY_ENTER)
16	click	css=button:nth-child(2) > img	
17	click	css= option-container:nth-child(1) > label	
18	click	css= menu-icon	
19	click	css=li:nth-child(3) > strong	
20	click	css=li:nth-child(3) .btn	

Los comandos para este escenario ponen a prueba todas las funciones que contiene el mapa. Filtrado por servicio, búsqueda por nombre, despliegue de menú y búsqueda a través del menú.

Creación de negocio:

Publicación de reseña:

http://localhost:3000			
	Command	Target	Value
3	click	linkText=Log in	
4	click	id=username	
5	type	id=username	usuario3
6	click	id=password	
7	type	id=password	contraseña
8	click	css= btn	
9	click	linkText=Places	
10	click	linkText=Reseñas	
11	run script	window.scrollTo(0,0)	
12	click	id=título	
13	type	id=título	reseña4
14	click	id=contenido	
15	type	id=contenido	Una gran experiencia
16	click	css=button	

Una vez iniciada sesión como usuario, y accediendo a la página de reseñas de un comercio, se añade una reseña que tiene como título “reseña1” y descripción “Una gran experiencia”.

3.2.5 Postcondiciones

Los resultados de los comandos introducidos para cada uno de los casos que han sido mencionados previamente en el apartado son los siguientes:

Registro

cliente:

Log	Reference	
Running 'registro cliente'		
1. open on / OK		18:34:14
2. setWindowSize on 1552x832 OK		18:34:14
3. click on linkText=Accede OK		18:34:14
4. click on linkText=Registrate OK		18:34:15
5. click on id=email OK		18:34:15
6. type on id=email with value usuario@gmail.com OK		18:34:15
7. click on id=username OK		18:34:16
8. type on id=username with value usuario3 OK		18:34:16
9. click on id=password OK		18:34:16
10. type on id=password with value contraseña OK		18:34:16
11. click on id=password2 OK		18:34:16
12. type on id=password2 with value contraseña OK		18:34:16
13. click on css= btn OK		18:34:17
14. Trying to find css= Toastify__close-button > svg... OK		18:34:17
15. click on linkText=Log out OK		18:34:17
16. click on linkText=Log in OK		18:34:18
17. click on id=username OK		18:34:18
18. type on id=username with value usuario3 OK		18:34:18
19. click on id=password OK		18:34:18
20. type on id=password with value contraseña OK		18:34:18
21. click on css= btn OK		18:34:19
'registro cliente' completed successfully		

Registro propietario:

Log	Reference	
Running 'registro propietario'		
1. open on / OK		18:35:25
2. setWindowSize on 1552x832 OK		18:35:25
3. click on linkText=Log in OK		18:35:25
4. click on linkText=Registrate OK		18:35:25
5. click on id=email OK		18:35:26
6. type on id=email with value propietario@gmail.com OK		18:35:26
7. click on id=username OK		18:35:26
8. type on id=username with value propietario3 OK		18:35:27
9. click on id=password OK		18:35:27
10. type on id=password with value contraseña OK		18:35:27
11. click on id=password2 OK		18:35:27
12. type on id=password2 with value contraseña OK		18:35:27
13. click on css= register-container OK		18:35:27
14. click on css= form-check OK		18:35:27
15. click on id=propietario OK		18:35:27
16. click on css= btn OK		18:35:27
17. click on id=username OK		18:35:27
18. type on id=username with value propietario3 OK		18:35:27
19. click on id=password OK		18:35:27
20. type on id=password with value contraseña OK		18:35:27
21. click on css= btn OK		18:35:27
22. click on linkText=Places OK		18:35:27
'registro propietario' completed successfully		

Consulta de establecimientos:

Log	Reference	
11. mouseMoveAt on css= mapboxgl-canvas with value 703,329,1500015258789 OK		18:36:29
12. mouseUpAt on css= mapboxgl-canvas with value 703,329,1500015258789 OK		18:36:30
13. click on css= busqueda-container > input OK		18:36:30
14. type on css= busqueda-container > input with value Tienda 23 OK		18:36:30
15. sendKeys on css= busqueda-container > input with value \$[KEY_ENTER] OK		18:36:30
16. click on css=button:nth-child(2) > img OK		18:36:31
17. click on css= option-container:nth-child(1) > label OK		18:36:31
18. click on css= menu-icon OK		18:36:32
19. click on css=li:nth-child(3) > strong OK		18:36:32
20. click on css=li:nth-child(3) > btn OK		18:36:32
'consulta establecimiento' completed successfully		

Creación

de

negocio:

3. click on linkText=Log in OK	17:25:20	
4. click on id=username OK	17:25:21	
5. type on id=username with value propietario3 OK	17:25:22	
6. click on id=password OK	17:25:23	
7. type on id=password with value contraseña OK	17:25:24	
8. click on css= btn OK	17:25:25	
9. click on linkText=Places OK	17:25:26	
10. click on css= btn OK	17:25:27	
11. click on css=div:nth-child(2) > input OK	17:25:28	
12. type on css=div:nth-child(2) > input with value Mi negocio OK	17:25:29	
13. click on css=div:nth-child(3) > input OK	17:25:30	
14. type on css=div:nth-child(3) > input with value calle guzman el bueno 4, Madrid OK	17:25:31	
15. click on name=accesorios OK	17:25:32	
16. click on name=peluqueria OK	17:25:33	
17. click on name=ocio OK	17:25:34	
18. click on css= btn OK	17:25:35	
19. click on linkText=Map OK	17:25:36	
20. click on css= menu-icon OK	17:25:37	
21. click on css=li:nth-child(8) > p OK	17:25:38	
"Publicacion comercio" completed successfully		17:25:39

Publicación

de

reseña:

Log	Reference	
8. click on css= btn OK		17:23:07
9. click on linkText=Places OK		17:23:08
10. click on linkText=Reseñas OK		17:23:09
11. runScript on window.scrollTo(0,0) OK		17:23:10
12. click on id=título OK		17:23:11
13. type on id=título with value reseña4 OK		17:23:12
14. click on id=contenido OK		17:23:13
15. type on id=contenido with value Una gran experiencia OK		17:23:14
16. click on css=button OK		17:23:15
'Añadir reseña' completed successfully		

Cada una de las simulaciones se realizaron tres veces y en todas el resultado fue el esperado y es el que se observa en las capturas.

3.2.6 Flujo de eventos

Pasos del caso de prueba: acciones de usuario o interacciones usuario-sistema.

Registro de nuevo usuario:

1. Hacer click en el botón de Log in situado en la parte superior izquierda
2. Acceder al enlace de registro
3. Rellenar el formulario de nuevo usuario y pulsar el botón de registro

Consulta establecimiento:

1. Hacer click en el botón de Log in situado en la parte superior izquierda
2. Introducir el nombre de usuario y contraseña
3. Acceder a Map ubicado en la barra de navegación
4. Consultar el establecimiento a través de la barra de búsqueda o el menú desplegable

Creación de negocio:

1. Hacer click en el botón de Log in situado en la parte superior izquierda
2. Introducir nombre de usuario y contraseña de una cuenta registrada como propietario
3. Acceder a Places ubicado en la barra de navegación
4. Pulsar el botón de crear negocio
5. Rellenar el formulario de creación de negocio con la información del negocio
6. Pulsar el botón de confirmación

Publicación de reseña:

1. Hacer click en el botón de Log in situado en la parte superior izquierda
2. Introducir nombre de usuario y contraseña asociados a una cuenta de cliente
3. Acceder a Map/Places y encontrar el negocio al que se quiera realizar la reseña
4. Pulsar el botón reseña
5. Rellenar el formulario de reseña con el título y descripción
6. Pulsar el botón de confirmación

3.2.7 Resultado

Resultado de ejecución del caso de prueba (pasa/no pasa) y comentarios. Puede ser más práctico recoger los resultados en una hoja de cálculo que resuma para cada caso de prueba la información básica: nombre del caso, datos de prueba, resultado. Ese documento puede referenciarse aquí o incluirse como anexo a este documento.

Los resultados del caso de prueba de Selenium han sido positivos. Se han realizado las pruebas de los diferentes escenarios en varias ocasiones y en todas el resultado ha sido el mismo. Por lo que podemos decir que la fiabilidad de nuestra aplicación con respecto a estas interacciones con el usuario es alta.