



**ParkingBox**  
**Planes de Pruebas Funcionales**

Versión: 1.0  
Fecha: 23/08/2024



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

#### HOJA DE CONTROL

<b>Organismo</b>	Estacionamiento calle 109		
<b>Proyecto</b>	ParkingBox		
<b>Entregable</b>	Planes de Pruebas Funcionales		
<b>Autor</b>	Jhon Sebastián Alvarez, Javier Gil Sierra.		
<b>Versión/Edición</b>	1.0	<b>Fecha Versión</b>	23/08/2024
<b>Aprobado por</b>	Intuición Sena	<b>Fecha Aprobación</b>	DD/MM/AAAA
		<b>Nº Total de Páginas</b>	17

#### REGISTRO DE CAMBIOS

<b>Versión doc</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
1.0	Versión inicial	Jhon Sebastián Alvarez, Javier Gil Sierra.	23/08/2024

#### CONTROL DE DISTRIBUCIÓN

<b>Nombre y Apellidos</b>
Jhon Sebastián Alvarez Medina.
Javier Gil Sierra.



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

## ÍNDICE

1 INTRODUCCIÓN .....	4
1.1 Objeto .....	4
1.2 Alcance .....	4
2 TRAZABILIDAD DE CASOS DE PRUEBAS – REQUISITOS .....	5
3 DEFINICIÓN DE LOS CASOS DE PRUEBAS .....	6
4 ESTRATEGIA DE EJECUCIÓN DE PRUEBAS .....	14
5 ANEXOS .....	15
6 GLOSARIO .....	16
7 BIBLIOGRAFÍA Y REFERENCIAS .....	17



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

## 1 INTRODUCCIÓN

### 1.1 Objetivos

El objetivo de las pruebas funcionales es verificar que el sistema desarrollado cumple con los requisitos especificados, asegurando su correcto funcionamiento en escenarios reales de uso. Este documento detalla los casos de prueba diseñados para evaluar las funcionalidades clave, como el inicio de sesión y el ingreso de vehículos, alineándolos con los requisitos establecidos en la fase de diseño. Además, se incluye una matriz de trazabilidad que vincula cada caso de prueba con su correspondiente requisito, proporcionando una clara visión de la cobertura de pruebas. La estrategia de ejecución sigue un enfoque meticuloso, utilizando Cypress.js para realizar pruebas automáticas E2E, garantizando que cada componente del sistema funcione conforme a lo esperado y cumpla con los estándares de calidad requeridos.

### 1.2 Alcance

El alcance de este documento se dirige principalmente a las unidades de desarrollo y aseguramiento de calidad de la empresa, específicamente a los aprendices de programación Jhon Álvarez Medina y Javier Gil Sierra, quienes son responsables de la generación, documentación y ejecución de las pruebas funcionales del sistema. Bajo la supervisión y aprobación del instructor Albeiro Ramos, este equipo ha validado cada etapa del proceso, asegurando que el sistema cumple con los estándares y requisitos definidos. La empresa Estacionamiento Calle 109 es la unidad organizativa informada, que recibirá los resultados de estas pruebas y el registro detallado de la validación del sistema, garantizando que el producto final esté alineado con sus expectativas y necesidades operativas.



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

## 2 TRAZABILIDAD DE CASOS DE PRUEBAS – REQUISITOS

	RF-001	RF-002	RF-003	RF-004
CP-1	20/08/2024	20/08/2024	20/08/2024	23/08/2024
CP-2	23/08/2024	23/08/2024	23/08/2024	23/08/2024
CP-3	23/08/2024	23/08/2024	23/08/2024	23/08/2024
CP-4	23/08/2024	23/08/2024	23/08/2024	23/08/2024
CP-5	23/08/2024	23/08/2024	23/08/2024	23/08/2024
CP-6	23/08/2024	23/08/2024	23/08/2024	23/08/2024



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

### 3 DEFINICIÓN DE LOS CASOS DE PRUEBAS

Cypress.js es una herramienta utilizada para realizar pruebas end-to-end (E2E) en aplicaciones Node.js y React, centrándose en la funcionalidad desde la perspectiva del usuario final. Este tipo de pruebas incluye pruebas funcionales, que verifican que las funcionalidades de la aplicación funcionen correctamente, pruebas de caja negra, que se centran en las entradas y salidas sin considerar la estructura interna del código, y pruebas estáticas, que revisan la configuración sin ejecutar el código. Estoy integrando Cypress.js para asegurar la estabilidad y funcionalidad de la aplicación mediante estas pruebas detalladas.



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

<b>Prueba Iniciar Sesión</b>	<b>&lt;Código del CP&gt;</b>	<b>1</b>
	<b>¿Prueba de despliegue?</b>	<b>No</b>

**Descripción:**  
Esta prueba tiene como objetivo indicar como iniciar sesión al usuario para poder interactuar con el sistema software.

**Prerrequisitos**

1. Necesita usuario registrado en la aplicación para poder ingresar.
2. Para poder visualizar la página desde el local hay que ubicarse en esta ruta (<http://localhost:3000/>).
3. En la vista de ingreso a la aplicación hay un botón de control de iniciar de sesión.
4. Luego hacer clics en el control de iniciar sesión.

**En caso de algún desacuerdo:**

1. Si se deja un campos vacio el sistema despliega un anuncio de que faltan información por diligenciar.
2. Si el usuario diligenciado en el formulario no coincide con sus credenciales, El sistema indica el error de autenticación y tiene que validar nuevamente.
3. Cuando el usuario que ingresa credenciales que no existen no permite ingresar y despliega la alerta de usuario no valido o no existente.

**Pasos:**

1. Al hacer clics en el control de inicio sesión se despliega un módulo con un formulario.
2. Este formulario tiene dos campos email y contraseña.
3. Se deben diligenciar ambos campos si no se hace el sistema te indicara que hay campos vacíos.
4. Si las credenciales están correctas te podrás ingresar al módulo del dashboard.

**Resultado esperado:**  
Se espera que si al seguir los pasos para iniciar sesión permita al usuario ir a la página del dashboard.  
**Resultado Obtenido:**  
El resultado obtenido es aprobado en esta prueba funcional.





	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

Prueba Obtener datos de la lista vehículo	<Código del CP>	2
	¿Prueba de despliegue?	No

**Descripción:**  
 Esta prueba tiene como objetivo probar el método de obtener los datos del vehículo en su lista que es de tipo get, en el sistema para visualizar sus parámetros.

**Prerrequisitos**

1. Para poder visualizar en la página de vehículo su lista en la página de dashboard hay una barra de menú.
2. En la vista del dashboard en la barra de menú hay un control de Vehículos Ingreso.
3. Luego hacer clics en el control de Vehículos Ingreso.
4. Haber creado con antelación un vehículo en el sistema.
5. tener la conexión a la base de datos y al servidor backend.

**En caso de algún desacuerdo:**

1. Si no hay conexión a la base de datos, el sistema no podrá obtener los datos de la lista de vehículos y desplegará un mensaje de error indicando la falta de conexión a la base de datos.
2. Si el puerto del backend no está activo, el sistema no podrá comunicarse con el servidor, por lo tanto, no mostrará la lista de vehículos y mostrará un mensaje de error de conexión con el servidor.
3. Si ocurre un error interno en el servidor backend (como un fallo en la API), el sistema no podrá obtener la lista de vehículos y mostrará un mensaje genérico indicando un error al recuperar los datos.

**Pasos:**

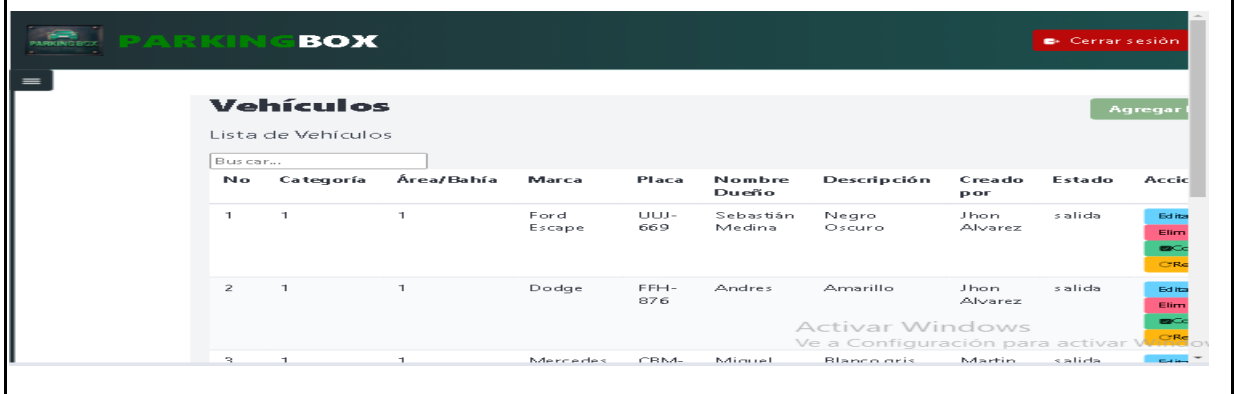
1. Al hacer clics en el control de Vehículo Ingreso se despliega un módulo con una lista y botones de acción.
2. Este módulo para obtener los datos del vehículo hay que tener datos creados y asegurarse que el servicio de base de datos esté bien configurado.
3. Si todo está correctamente y hay datos creados la lista debe obtener los parámetros creados.

**Resultado esperado:**

Se espera que el módulo visualice los parámetros de los vehículos que sean creados.

**Resultado Obtenido:**

El resultado obtenido es aprobado en esta prueba funcional.







	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

<b>Prueba Registro Ingreso de Vehículo</b>	<b>&lt;Código del CP&gt;</b>	<b>3</b>
	<b>¿Prueba de despliegue?</b>	<b>No</b>

**Descripción:**  
Esta prueba tiene como objetivo probar el método crear vehículo que de tipo post y se visualice en la lista del módulo Vehículo Ingreso.

**Prerrequisitos**

1. Para poder crear vehículo en el módulo de Vehículo Ingreso hay un control para agregar nuevo.
2. Al hacer click en el control agregar nuevo se despliega un módulo en la pantalla que tiene un formulario.
3. Los campos a diligenciar en este formulario son estos: Marca, Placa, Nombre del Dueño, Descripción, Categoría y Área/Bahía.

**En caso de algún desacuerdo:**

1. Si se deja un campo obligatorio sin diligenciar, el sistema mostrará una alerta indicando que falta información por completar.
2. Si se ingresa información en un formato incorrecto (por ejemplo, caracteres en un campo numérico), el sistema desplegará un mensaje de alerta informando que la información no es válida.
3. Si ocurre un error al enviar los datos al backend, el sistema mostrará un mensaje de error indicando que no se pudo registrar el vehículo.

**Pasos:**

1. Los campos a diligenciar tiene estas características: Marca (este es de tipo Varchar o alfanumérico), Placa (este es de tipo Varchar o alfanumérico), Nombre del Dueño (este es de tipo Varchar o alfanumérico), Descripción (este es de tipo string o texto), Categoría (este campo es de selección y de tipo entero) y Área/Bahía (este campo es de selección y de tipo entero).
2. Después de diligenciar los campos como corresponde con la información del vehículo hacer clics en el botón de guardar de este formulario esto despliega la impresión de ticket de creación del vehículo.
3. Luego de imprimir el ticket la lista del módulo de Vehículo Ingreso el dato se almacena con sus parámetros.

**Resultado esperado:**

Se espera que el módulo Vehículo Ingreso se cree exitosamente el vehículo y se almacene en la lista.

**Resultado Obtenido:**

El resultado obtenido es aprobado en esta prueba funcional.

5	3	3	Monark 70	SWV-420	Sebastián Alvarez	Verde Oscuro	Martin Muñoz	salida		
6	1	1	BMW	ILK-990	Miguel	Rojo Oscuro	Jhon Alvarez	salida		
7	1	1	Volkswagen	NDS-085	Samuel	Gris Mate	Jhon Alvarez	entrada		
Mostrando 1 a 7 de 7 entradas										



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

Prueba Editar Vehículo	<Código del CP>	4
	¿Prueba de despliegue?	No
<b>Descripción:</b> Esta prueba tiene como objetivo probar el método Editar vehículo que es de tipo patch y se visualice en la lista del módulo Vehículo Ingreso.		
<b>Prerrequisitos</b> <ol style="list-style-type: none"> <li>1. Para poder editar vehículo en el módulo de Vehículo Ingreso hay un control para editar.</li> <li>2. Al hacer clics en el control editar se despliega un módulo en la pantalla que tiene un formulario.</li> <li>3. Los campos a diligenciar en este formulario son estos: Marca, Placa, Nombre del Dueño, Descripción, Categoría y Área/Bahía.</li> </ol> <b>En caso de algún desacuerdo:</b> <ol style="list-style-type: none"> <li>1. Si se deja un campo obligatorio sin diligenciar, el sistema mostrará una alerta indicando que falta información por completar.</li> <li>2. Si se ingresa un formato incorrecto en alguno de los campos, el sistema desplegará un mensaje de error informando que los datos son inválidos.</li> <li>3. Si ocurre un error en la comunicación con el backend al guardar los cambios, el sistema indicará que no se pudo actualizar el vehículo correctamente.</li> </ol>		
<b>Pasos:</b> <ol style="list-style-type: none"> <li>1. Los campos a diligenciar tiene estas características: Marca (este es de tipo Varchar o alfanumérico), Placa (este es de tipo Varchar o alfanumérico), Nombre del Dueño (este es de tipo Varchar o alfanumérico), Descripción (este es de tipo string o texto), Categoría (este campo es de selección y de tipo entero) y Área/Bahía (este campo es de selección y de tipo entero).</li> <li>2. Después de diligenciar los campos como corresponde con la información del vehículo hacer click en el botón de actualizar de este formulario esto despliega la impresión de ticket de creación del vehículo.</li> </ol>		
<b>Resultado esperado:</b> Se espera que el módulo Vehículo Ingreso se edite exitosamente el vehículo y se actualice en la lista.		
<b>Resultado Obtenido:</b> El resultado obtenido es aprobado en esta prueba funcional.		



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

Prueba Salida de Vehículo	<Código del CP>	5
	¿Prueba de despliegue?	No

**Descripción:**  
Esta prueba tiene como objetivo probar el método Salida de vehículo que es de tipo get y patch, se visualice en la lista del módulo Vehículo Ingreso.

**Prerrequisitos**

1. Para poder ejecutar el método salida del vehículo en el módulo de Vehículo Ingreso hay un control para cotizar.
2. Al hacer clics en el control cotizar se despliega un módulo en la pantalla que tiene un formulario.
3. Este módulo tiene un control para calcular el costo de estacionamiento y cambio para escribir el monto de pago.

**Pasos:**

1. Al hacer clics en el control de calcular obtiene el costo de estacionamiento y luego diligenciar el monto de pago.
2. Después de diligenciar el monto de pago en el campo de cambio se muestra el cálculo de la diferencia que se debe devolver .
3. Luego hacer clics en pagar esto despliega la impresión y al terminal este proceso la lista cambiara el estado del vehiculó a salida en la lista de vehículo.

**En caso de algún desacuerdo:**

1. Si el vehículo no tiene un estado de entrada, el sistema no permitirá realizar la cotización y mostrará un mensaje de error.
2. Si no se ingresa el monto de pago, el sistema indicará que es necesario especificar el pago antes de proceder.
3. Si el monto de pago es menor que el costo del estacionamiento, el sistema mostrará en el ticket que el cliente quedo en deuda.

**Resultado esperado:**

Se espera que el módulo Vehículo Ingreso haga la cotización exitosamente el vehículo y se actualice en la lista.

**Resultado Obtenido:**

El resultado obtenido es aprobado en esta prueba funcional.

5	3	3	Monark 70	SWV-420	Sebastián Álvarez	Verde Oscuro	Martin Muñoz	salida	B	E	C
6	1	1	BMW	NLK-990	Miguel	Rojo Oscuro	Jhon Alvarez	salida	B	E	C
7	1	1	Volkswagen Jetta	NDS-085	Miguel Pérez	Gris Oscuro	Jhon Alvarez	salida	B	E	C

Mostrando 1 a 7 de 7 entradas

Activar Windows  
Ve a Configuración para activar Windows

Anterior 1



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

Prueba Reingresar Vehículo	<Código del CP>	6
	¿Prueba de despliegue?	No

**Descripción:**  
 Esta prueba tiene como objetivo probar el método Reingresar de vehículo que es de tipo patch, se visualice en la lista del módulo Vehículo Ingreso.

**Prerrequisitos**

1. Para poder ejecutar el método reingresar vehículo en el módulo de Vehículo Ingreso hay un control para reingresar.
2. Al hacer clics en el control reingresar se despliega un módulo en la pantalla que tiene un formulario.
3. Este formulario tiene un campo para cambiar el Nombre del Dueño si el vehículo ingresa con otra persona.

**Pasos:**

1. Al hacer clics en el control de Ingresar de Nuevo.
2. Esta acción muestra un alerta que indica que se ingresó de nuevo el vehículo con el tiempo de ingreso que se obtiene tras ejecutar esta función.
3. Luego de presionar en la alerta el botón de ok se despliega la impresión del reingreso del vehículo y se actualiza el estado del vehículo a entrada.

**En caso de algún desacuerdo:**

1. Si el vehículo no está en estado de salida, el sistema no permitirá reingresarlo y mostrará un mensaje de error.
2. Si no se actualiza el nombre del dueño cuando el vehículo ingresa con otra persona, el sistema emitirá una advertencia para confirmar si el dueño es el mismo.
3. Si ocurre un error en la base de datos al reingresar el vehículo, el sistema no completará la operación y desplegará un mensaje de fallo en la reingreso del vehículo.

**Resultado esperado:**  
 Se espera que el módulo Vehículo Ingreso haga el reingreso exitosamente el vehículo y se actualice en la lista.

**Resultado Obtenido:**  
 El resultado obtenido es aprobado en esta prueba funcional.

5	3	3	Monark 70	SWV-420	Sebastián Álvarez	Verde Oscuro	Martin Muñoz	salida	6
6	1	1	BMW	ÑLK-990	Miguel	Rojo Oscuro	Jhon Alvarez	salida	6
7	1	1	Volkswagen Jetta	ÑDS-085	Estiven Pérez	Gris Oscuro	Jhon Alvarez	entrada	6

Mostrando 1 a 7 de 7 entradas

Activar Windows  
 Ve a Configuración para activar Windows

Anterior 1



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

Prueba Eliminar Vehículo	<Código del CP>	7
	¿Prueba de despliegue?	No

**Descripción:**  
 Esta prueba tiene como objetivo probar el método Eliminar vehículo que es de tipo delete, se visualice en la lista del módulo Vehículo Ingreso.

**Prerrequisitos**

1. Para poder ejecutar el método eliminar vehículo en el módulo de Vehículo Ingreso hay un control para eliminar.
2. Al hacer clics en el control eliminar.
3. Al hacer clics en el control despliega un alerta donde pregunta que si se quiere eliminar.

**En caso de algún desacierto:**

1. Si el vehículo no está en estado de salida, el sistema no permitirá eliminarlo y mostrará un mensaje de error.
2. Si no se acepta la confirmación en la alerta, el vehículo no se eliminará y el sistema regresará a la lista de vehículos.
3. Si ocurre un error en la comunicación con el backend, el sistema no podrá eliminar el vehículo y mostrará un mensaje de fallo en la operación.

**Pasos:**

1. Se debe presionar en el alerta si, eliminarlo.
2. Después de presionar el botón de la alerta eliminara en campo de la lista de vehículo.

**Resultado esperado:**

Se espera que el módulo Vehículo Ingreso se elimine exitosamente el vehículo y se actualice en la lista.

**Resultado Obtenido:**

El resultado obtenido es aprobado en esta prueba funcional.

4	2	2	Yamaha	EFR-765	Steven Muñoz	Azul oscuro	Martin Muñoz	salida	Editar	Eliminar	Revisar
5	3	3	Monark 70	SWV-420	Sebastián Álvarez	Verde Oscuro	Martin Muñoz	salida	Editar	Eliminar	Revisar
6	1	1	BMW	NLK-990	Miguel	Rojo Oscuro	Jhon Alvarez	salida	Editar	Eliminar	Revisar

Mostrando 1 a 6 de 6 entradas



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

## 4 ESTRATEGIA DE EJECUCIÓN DE PRUEBAS

	<b>Ciclo 1</b>	<b>Ciclo 2</b>
<b>CP-1</b>	22/08/2024	23/08/2024
<b>CP-2</b>	22/08/2024	23/08/2024
<b>CP-3</b>	22/08/2024	23/08/2024
<b>CP-4</b>	22/08/2024	23/08/2024
<b>CP-5</b>	22/08/2024	23/08/2024
<b>CP-6</b>	22/08/2024	23/08/2024
<b>CP-7</b>	22/08/2024	23/08/2024



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

## 5 ANEXOS

En el anexo de este documento se incluye información clave recopilada durante la elaboración y validación del Plan de Pruebas Funcionales, como la selección y configuración de la herramienta Cypress.js, utilizada para la ejecución de pruebas automatizadas en el entorno JavaScript y React.js. También se detallan los procedimientos seguidos para garantizar que las pruebas cubran todos los componentes críticos del sistema, como los módulos de inicio de sesión y gestión de vehículos. Además, se adjunta la documentación relacionada con la planificación, ejecución y análisis de los resultados de las pruebas, incluyendo el plan maestro de pruebas, las matrices de trazabilidad, y los informes de resultados. Esta información es fundamental para comprender el proceso de validación, asegurar la calidad del software, y facilitar la continuidad del proyecto en futuras fases de desarrollo o mantenimiento.



	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

## 6 GLOSARIO

Término	Descripción
Pruebas Funcionales	Pruebas que verifican que el sistema cumple con los requisitos especificados, evaluando las funciones del software en base a entradas y salidas esperadas.
Pruebas Estáticas	Evaluaciones del software que se realizan sin ejecutarlo, revisando el código, documentos o diagramas para detectar posibles errores o mejoras.
Caja Negra	Técnica de prueba donde los testers evalúan la funcionalidad del software sin conocer su estructura interna, centrándose en las entradas y salidas.
Cypress.js	Herramienta de testing para aplicaciones web, especialmente diseñada para realizar pruebas automatizadas end-to-end (E2E) en entornos JavaScript.
Node.js	Entorno de ejecución para JavaScript construido con el motor V8 de Google Chrome, que permite el desarrollo del lado del servidor de aplicaciones web.
React.js	Biblioteca de JavaScript utilizada para construir interfaces de usuario dinámicas y eficientes, especialmente en aplicaciones de una sola página (SPA).





	<b>ParkingBox</b> <b>Planes de Pruebas Funcionales</b>	<b>Estacionamiento calle 109</b>
--	---	----------------------------------

## 7 BIBLIOGRAFÍA Y REFERENCIAS

Referencia	Título
Ref. 1	Cypress Documentation: End-to-End Testing Framework
Ref. 2	Master Test Plan: Guidelines and Best Practices
Ref. 3	JavaScript Testing Best Practices: A Comprehensive Guide
Ref. 4	Node.js and React.js Development: Essential Tools and Techniques