

# Plan de Pruebas Caso de Uso "Nuevo Estacionamiento"

Este documento define las especificaciones de pruebas para el caso de uso "Nuevo Estacionamiento" del sistema UCPark, abarcando pruebas de aceptación y pruebas unitarias seleccionadas.

## 1. Especificación de Pruebas de Aceptación

**Caso de Uso:** Nuevo Estacionamiento **Actor:** Usuario Registrado **Precondiciones:** El usuario debe estar registrado y tener al menos un vehículo asociado.

ID Prueba	Escenario	Pasos	Datos de Entrada	Resultado Esperado
ACP-01	Estacionamiento Exitoso	<ol style="list-style-type: none"><li>El usuario selecciona la opción "Nuevo Estacionamiento".</li><li>El sistema solicita seleccionar un vehículo.</li><li>El usuario selecciona un vehículo válido.</li><li>El sistema registra el inicio del estacionamiento.</li></ol>	Usuario: <code>usuario@test.com</code> Vehículo: ABC-1234	<ul style="list-style-type: none"><li>- Se crea un registro de estacionamiento.</li><li>- El sistema muestra confirmación de inicio.</li><li>- La hora de inicio es la actual.</li></ul>
ACP-02	Vehículo no seleccionado	<ol style="list-style-type: none"><li>El usuario intenta iniciar estacionamiento sin seleccionar vehículo.</li></ol>	Usuario: <code>usuario@test.com</code> Vehículo: <code>null</code>	<ul style="list-style-type: none"><li>- El sistema muestra un mensaje de error indicando que debe seleccionar un vehículo.</li></ul>
ACP-03	Usuario no válido (Sesión expirada)	<ol style="list-style-type: none"><li>El usuario intenta iniciar estacionamiento con una sesión inválida.</li></ol>	Usuario: <code>null</code> Vehículo: ABC-1234	<ul style="list-style-type: none"><li>- El sistema redirige al login o muestra error de autenticación.</li></ul>
ACP-04	Estacionamiento Duplicado (Vehículo Activo)		Usuario: <code>usuario@test.com</code> Vehículo: ABC-1234 (ya activo)	<ul style="list-style-type: none"><li>- El sistema muestra un mensaje de error.</li><li>- El mensaje indica que el vehículo ya tiene un estacionamiento activo.</li><li>- No se crea un nuevo registro.</li></ul>
ACP-05	Vehículo No Pertenece al Usuario		Usuario: <code>usuario@test.com</code> Vehículo: XYZ-789 (de otro usuario)	<ul style="list-style-type: none"><li>- El sistema rechaza la operación.</li><li>- Se muestra error de autorización/validación.</li><li>- No se crea registro de estacionamiento.</li></ul>
ACP-06	Usuario No Tiene Vehículos Registrados		Usuario: <code>usuario-sin-vehiculos@test.com</code> Vehículo: N/A	<ul style="list-style-type: none"><li>- El sistema muestra mensaje informativo indicando que debe registrar un vehículo primero.</li></ul>

ID Prueba	Escenario	Pasos	Datos de Entrada	Resultado Esperado
				- Opcionalmente, redirige a registro de vehículos.

## 2. Especificación de Pruebas Unitarias

### A. Capa de Negocio

Componente:

CarManagement Método a Probar:

newParking(User user, Vehicle vehicle)

Casos de Prueba:

ID	Descripción	Entradas	Comportamiento Esperado (Asserts)
UT-BUS-01	Inicialización Correcta de Estacionamiento	user : Instancia válida vehicle : Instancia válida ( plate: "XYZ-999" )	<ol style="list-style-type: none"> <li>Se debe invocar <code>vehicleDao.createParking()</code>.</li> <li>El objeto Parking pasado al DAO debe tener:           <ul style="list-style-type: none"> <li><code>vehicle</code> igual al vehículo de entrada.</li> <li><code>minutes</code> igual a <code>0</code>.</li> <li><code>amount</code> igual a <code>0.0</code>.</li> <li><code>startTime</code> no nulo (aproximadamente <code>now</code>).</li> </ul> </li> <li>El método debe retornar <code>true</code>.</li> </ol>
UT-BUS-02	Fallo por Entradas Nulas	user : null O vehicle : null	<ol style="list-style-type: none"> <li>El método debe retornar <code>false</code>.</li> <li>No se debe invocar <code>vehicleDao.createParking()</code>.</li> </ol>
UT-BUS-03	Fallo en Persistencia DAO	user : Válido vehicle : Válido <code>vehicleDao.createParking()</code> : retorna <code>null</code> o lanza excepción (mock)	<ol style="list-style-type: none"> <li>El método debe retornar <code>false</code>.</li> <li>Verificar que se maneje adecuadamente el error en el DAO.</li> <li>No se debe propagar excepción no controlada.</li> </ol>
UT-BUS-04	Verificación de Timestamp Correcto	user : Válido vehicle : Válido	<ol style="list-style-type: none"> <li>Capturar el objeto Parking antes de <code>createParking()</code>.</li> <li>Verificar: <code>assertNotNull(parking.getStartTime())</code>.</li> <li>Verificar: <code>parking.getStartTime()</code> esté dentro de <math>\pm 1</math> segundo del momento actual.</li> </ol>
UT-BUS-05	Validación de Vehículo Ya Estacionado	vehicle : Vehículo con estacionamiento activo (mock)	<ol style="list-style-type: none"> <li>El método debe verificar si existe estacionamiento activo.</li> <li>Si existe, retornar <code>false</code>.</li> <li>No invocar <code>vehicleDao.createParking()</code>.</li> </ol>

### B. Capa DAO

Componente:

VehicleDao (Anteriormente ParkingDao , refactorizado) Método a Probar:

createParking(Parking parking)

## Casos de Prueba:

ID	Descripción	Entradas	Comportamiento Esperado (Asserts)
UT-DAO-01	Persistencia Exitosa	<p>parking : Objeto</p> <p>Parking válido (con vehículo y hora definidos)</p>	<p>1. Se debe invocar EntityManager.persist(parking) .</p> <p>2. El método debe retornar el mismo objeto</p> <p>Parking persistido.</p>
UT-DAO-02	Persistencia Fallida (Duplicado/Error)	<p>parking : Objeto</p> <p>Parking que viola restricciones (ej. ID existente si aplica)</p>	<p>1. EntityManager.persist lanza excepción (simulado en mock o prueba de integración).</p> <p>2. Verificar manejo de excepción si existe, o fallo esperado.</p>
UT-DAO-03	Verificar Transaccionalidad	<p>parking : Válido</p> <p>Simular fallo después de persist() (mock EntityManager)</p>	<p>1. Verificar que si persist() falla, no se commitea la transacción.</p> <p>2. Verificar rollback automático o manejo de excepción.</p>
UT-DAO-04	Parking con Datos Nulos Parciales	<p>parking : Objeto con vehicle = null pero otros campos válidos</p>	<p>1. Verificar que persist() lance excepción o retorne error.</p> <p>2. O bien, validar que exista pre-validación antes de persistir.</p>
UT-DAO-05	Verificar ID Autogenerado	<p>parking : Objeto válido sin ID asignado</p>	<p>1. Después de persist() , verificar que parking.getId() no sea null.</p> <p>2. Verificar que el ID sea un valor válido autogenerado por la BD.</p>