

DOCUMENTO DE PLAN DE PRUEBAS

Plantilla inspirada en el estándar IEEE 829-2008 y adaptada a las necesidades del curso
de Construcción de Software

(Plantilla compilada por Ph.D. Franklin Parrales B.)

Tabla de contenido

1. Introducción	3
1.1. Propósito.....	3
1.2. Objetivos.....	3
2. Alcance	3
3. Definiciones, siglas, y abreviaturas	5
4. Documentos de referencia.....	6
5. Estrategia de pruebas	6
5.1. Pruebas unitarias.....	6
5.2. Pruebas de integración	10
6. Criterios de entrada y de salida	22
6.1. Criterio de entrada del plan de pruebas	23
6.2. Criterio de aceptación del plan de pruebas	23
6.3. Criterio de suspensión y reanudación	23
7. Entregables	23
7.1. Protocolo de pruebas	24
7.2. Informe de pruebas	24
7.3. Reporte de observaciones	24
8. Ambiente de pruebas	24
8.1. Requerimientos base de hardware	24
8.2. Requerimientos base de software en el ambiente de pruebas	24
8.3. Herramientas de apoyo para la ejecución de pruebas.....	25
9. Cronograma.....	25
10. Riesgos, dependencias, suposiciones y restricciones.....	25
10.1. Riesgos.....	25
10.2. Dependencias	26
10.3. Suposiciones	26
10.4. Restricciones.....	26

1. Introducción

1.1. Propósito

El propósito de este documento es garantizar que los requerimientos y especificaciones funcionales definidas en el proyecto Sistema de Gestión para Taller Vehicular se encuentren desarrolladas correctamente y cumplan con las funcionalidades y estándares de calidad esperados, con el objetivo de minimizar los incidentes que puedan ocurrir al momento de realizar las pruebas de aceptación del usuario funcional.

1.2. Objetivos

- Validar y comprobar las entradas de los datos a registrar.
- Identificar y detallar las pruebas más importantes a realizarse para este proyecto.
- Establecer casos y condiciones de prueba basados en los requisitos funcionales.
- Identificar posibles errores o problemas al ejecutar los distintos casos de prueba.
- Dar un seguimiento oportuno a la corrección de incidentes reportados.
- Determinar la existencia de diferencias entre los requisitos funcionales y el desarrollo del producto.

2. Alcance

Tipo de pruebas	Aplica según el proyecto (si/no)
Pruebas unitarias	Si
Pruebas de integración	Si

Los objetivos de las pruebas fueron los siguientes:

Módulo de Clientes:

- **Pruebas Unitarias**
 - Validaciones de datos de clientes: Validación de campos obligatorios (nombre, correo) y tipos de datos correctos.
- **Pruebas de Integración**
 - Registro de clientes: Verificación del correcto almacenamiento en memoria.
 - Actualización de clientes: Verificación de la correcta modificación de registros.
 - Eliminación de clientes: Verificación de la correcta eliminación lógica desde la tabla.

- Consulta de clientes: Verificación de la correcta visualización y recuperación en la tabla.

Módulo de Vehículos:

- **Pruebas Unitarias**
 - Validaciones de datos de vehículos: Verificación de campos como marca, modelo, año, y su asociación obligatoria con un cliente.
- **Pruebas de Integración**
 - Registro de vehículos: Verificación del almacenamiento del vehículo y su correcta asociación a un cliente.
 - Actualización de vehículos: Verificación de los cambios aplicados sobre los datos registrados.
 - Eliminación de vehículos: Confirmación de la eliminación del registro de la vista.
 - Consulta de vehículos: Verificación de la recuperación de los vehículos asociados a un cliente.

Módulo de Inventario:

- **Pruebas Unitarias**
 - Validaciones de productos: Validación de campos obligatorios (nombre, precio, stock) y tipos de datos correctos.
- **Pruebas de Integración**
 - Registro de productos: Verificación del almacenamiento correcto en memoria.
 - Actualización de productos: Validación de la modificación de información de productos.
 - Baja lógica de productos: Confirmación de que los productos se oculten sin ser eliminados del array.
 - Búsqueda de productos: Comprobación de la búsqueda en tiempo real por nombre.

Módulo de Órdenes:

- **Pruebas Unitarias**
 - Validaciones de órdenes: Validación de selección de cliente, vehículo, y productos antes de registrar una orden.
- **Pruebas de Integración**
 - Creación de órdenes: Verificación de que se almacenen correctamente en memoria.
 - Cálculo automático: Comprobación del cálculo correcto de subtotal y total.
 - Consulta de órdenes: Validación de visualización y seguimiento del estado de las órdenes.

- Asociación de datos: Confirmar que una orden esté vinculada correctamente a un cliente y a su vehículo.

Módulo de Base de Datos:

- **Pruebas de Integración**
 - Conexión: Verificación de la conexión correcta a la base de datos (actualmente trabaja en memoria, pero puede ser adaptable).

2.2. Limitaciones

Este plan de pruebas no contempla los siguientes módulos:

- Integración con APIs de pago/facturación.
- Módulo de inventario
- Módulo de reportes

3. Definiciones, siglas, y abreviaturas

Sigla / Término	Definición
.NET	Un marco de trabajo de desarrollo de software creado por Microsoft. Proporciona herramientas y bibliotecas para desarrollar aplicaciones de escritorio, web y móviles.
Autenticación	Proceso de verificar la identidad de un usuario o sistema, generalmente comprobando credenciales como nombre de usuario y contraseña.
Validación	Proceso de verificar que los datos ingresados cumplen con los requisitos y restricciones especificadas, como formato, longitud o tipo de dato.
MessageBox	Ventana emergente en aplicaciones de Windows que muestra un mensaje al usuario, y puede incluir botones para permitir respuestas.
Campo / Input	Área en una interfaz de usuario donde el usuario puede ingresar datos; también conocido como campo de entrada o input.
Excepción	Evento que ocurre durante la ejecución de un programa y que interrumpe su flujo normal. Las excepciones se manejan para evitar fallos inesperados.
Parámetro	Valor que se pasa a una función, procedimiento o método para ser utilizado durante su ejecución.

4. Documentos de referencia

DOCUMENTO	VERSIÓN
Documento de diseño detallado final	

5. Estrategia de pruebas

5.1. Pruebas unitarias

Las pruebas unitarias son fundamentales para garantizar la calidad y fiabilidad del software, especialmente en aplicaciones como un sistema de gestión para un taller vehicular. Estas pruebas se centran en validar la funcionalidad de pequeñas unidades de código, asegurando que cada función, método o componente opere de forma correcta y consistente bajo diversas condiciones. En el contexto de nuestra aplicación de gestión de taller, las pruebas unitarias están orientadas a verificar que los datos ingresados por el usuario, como información de clientes, vehículos, productos en inventario u órdenes de servicio, sean precisos, válidos y coherentes. Estas pruebas también buscan detectar errores lógicos en los cálculos de totales en órdenes, validaciones de formularios y navegación entre módulos.

Código	CP-001
Caso de prueba	Validar datos correctos de cliente
Responsable	Grupo D
Descripción de la prueba	Esta prueba tiene como objetivo verificar que los datos proporcionados para un cliente estén completos y sean del tipo esperado. Se instancia la clase Cliente con los siguientes valores: <ul style="list-style-type: none">• nombres: "Ana López"• apellidos: "Pérez"• cedula: "0923456789"• correo: "analopez@gmail.com"• telefono: "0987654321"• direccion: "Quito, Ecuador"
Requisito previo	Haber accedido correctamente al módulo de clientes y tener acceso a la clase Cliente.
Resultado esperado	Al ejecutar el método "validarCliente()" de la clase "Cliente", este devuelve true, indicando que los datos son válidos.
Resultado obtenido	True

Estado	Exitoso
Observaciones	Ninguna. La prueba se realizó con datos controlados.

Código	CP-002
Caso de prueba	Validar cédula incorrecta (menos de 10 dígitos)
Responsable	Grupo D
Descripción de la prueba	Verificar que el sistema rechace cédulas con formato inválido. Se instancia la clase Cliente con: <ul style="list-style-type: none"> cedula: "12345" (inválida) Se ejecuta el método validarCliente().
Requisito previo	Haber accedido al sistema y estar en el módulo de registro de clientes.
Resultado esperado	El método validarCliente() devuelve false y muestra el mensaje: "La cédula debe tener 10 dígitos."
Resultado obtenido	False + mensaje de error: "La cédula debe tener 10 dígitos."
Estado	Exitoso
Observaciones	El sistema valida correctamente que la cédula contenga exactamente 10 dígitos.

Código	CP-003
Caso de prueba	Validar registro exitoso de vehículo
Responsable	Grupo D
Descripción de la prueba	Esta prueba tiene como objetivo verificar que un vehículo con datos válidos sea registrado correctamente en el sistema. Se ingresan los siguientes datos: marca: "Toyota" <ul style="list-style-type: none"> modelo: "Corolla" año: "2022" placa: "ABC-123" (única) dueño: "Juan Pérez" (existente) Se ejecuta el proceso de registro de vehículo y se guarda en LocalStorage.
Requisito previo	El cliente debe existir previamente en el sistema.
Resultado esperado	Vehículo registrado en LocalStorage. Mensaje: "Vehículo registrado exitosamente".
Resultado obtenido	Registro exitoso.

Estado	Exitoso
Observaciones	La placa se valida como única.

Código	CP-004
Caso de prueba	Validar placa duplicada
Responsable	Grupo D
Descripción de la prueba	<p>Esta prueba tiene como objetivo verificar que el sistema rechace el registro de un vehículo cuando la placa ya existe en el sistema. Se intenta registrar un nuevo vehículo con los siguientes datos:</p> <ul style="list-style-type: none"> • marca: "Hyundai" • modelo: "Elantra" • año: "2021" • placa: "ABC-123" (ya registrada previamente) • dueño: "Juan Pérez" (existente) <p>Se ejecuta el método registrarVehiculo().</p>
Requisito previo	El vehículo con placa "ABC-123" ya debe estar registrado en el sistema.
Resultado esperado	El sistema no permite el registro y muestra el mensaje de error: "La placa ya está registrada."
Resultado obtenido	Mensaje de error mostrado.
Estado	Exitoso
Observaciones	El sistema cumple con el Requisito Funcional RF-06, que establece que la placa debe ser única por vehículo registrado.

Código	CP-005
Caso de prueba	Calcular total de orden con descuento
Responsable	Grupo D
Descripción de la prueba	<p>Esta prueba tiene como objetivo verificar que el sistema calcule correctamente el total de una orden de servicio aplicando un descuento del 10%. Se establecen los siguientes valores para la orden:</p> <ul style="list-style-type: none"> • subtotal: \$100 • descuento: 10%

	Se ejecuta el método calcularTotal() del módulo de órdenes de servicio.
Requisito previo	El módulo de órdenes de servicio debe estar cargado correctamente y la orden debe tener los campos de subtotal y descuento establecidos.
Resultado esperado	El método calcularTotal() aplica correctamente el descuento del 10% sobre el subtotal de \$100, dando como resultado un total de \$90.
Resultado obtenido	\$90.
Estado	Exitoso
Observaciones	El sistema realiza correctamente la operación matemática y aplica el porcentaje de descuento correspondiente. Esta funcionalidad está alineada con el RF-11, que establece el cálculo automático del total con descuentos aplicables.

Código	CP-006
Caso de prueba	Validar inicio de sesión exitoso
Responsable	Grupo D
Descripción de la prueba	<p>Esta prueba tiene como objetivo verificar que un usuario con credenciales correctas pueda autenticarse correctamente en el sistema y acceder a su panel de control.</p> <p>Se utilizan las siguientes credenciales:</p> <ul style="list-style-type: none"> • correo: "admin@taller.com" • contraseña: "admin123" <p>Se ejecuta el proceso de autenticación a través del formulario de inicio de sesión.</p>
Requisito previo	Las credenciales del usuario deben estar previamente almacenadas en LocalStorage.
Resultado esperado	El sistema valida exitosamente las credenciales, concede acceso al usuario y redirige automáticamente al dashboard principal del sistema.
Resultado obtenido	Autenticación exitosa.
Estado	Exitoso
Observaciones	El sistema cumple con el RNF-01, relacionado con la seguridad en el proceso de autenticación, al validar correctamente las credenciales ingresadas.

Código	CP-007
Caso de prueba	Bloqueo tras 3 intentos fallidos
Responsable	Grupo D
Descripción de la prueba	<p>Esta prueba tiene como objetivo verificar que el sistema aplique correctamente una política de seguridad que bloquee la cuenta de un usuario tras tres intentos fallidos de inicio de sesión consecutivos. Se utiliza el nombre de usuario registrado:</p> <ul style="list-style-type: none"> usuario: "administrador1" <p>Se intenta iniciar sesión tres veces utilizando una contraseña incorrecta.</p>
Requisito previo	El usuario "administrador1" debe estar registrado previamente en el sistema.
Resultado esperado	Luego de tres intentos fallidos de autenticación, el sistema bloquea temporalmente la cuenta y muestra el siguiente mensaje: "Cuenta temporalmente desactivada."
Resultado obtenido	Después del tercer intento fallido, el sistema bloqueó el acceso y mostró el mensaje esperado.
Estado	Exitoso
Observaciones	Esta prueba confirma la implementación de la política de seguridad correspondiente al RNF-01, garantizando la protección contra accesos no autorizados mediante bloqueo de cuenta.

5.2. Pruebas de integración

Las pruebas de integración garantizan que los módulos del Sistema de Gestión de Taller Automotriz funcionen de forma conjunta. Validan la correcta ejecución de operaciones CRUD y la interacción entre Clientes, Vehículos y Órdenes. Aseguran la consistencia de datos en LocalStorage y el cumplimiento de reglas de negocio transversales. El enfoque está en integrar interfaz HTML/JS, lógica en JavaScript y almacenamiento temporal. Estas pruebas detectan errores en la comunicación entre capas del sistema. Permiten verificar la estabilidad de los flujos de trabajo. Aseguran la integridad de los datos en todo momento. Además, el sistema mantiene su funcionalidad incluso sin conexión a una base de datos externa.

Código	CP-008
Caso de prueba	Conexión exitosa a la base de datos
Responsable	Grupo D
Descripción de la prueba	<p>La prueba tiene como objetivo verificar que la conexión a la base de datos PostgreSQL se realice exitosamente utilizando Npgsql.</p> <p>Se utiliza la siguiente cadena de conexión: Host=localhost;Port=5432;Database=dbProyectoCS; Username=postgres;Password=admin123;SSL Mode=Prefer;Trust Server Certificate=true</p> <p>Se intenta abrir una conexión con esta cadena utilizando el método Open() del objeto NpgsqlConnection, y se verifica el estado de la conexión.</p>
Requisito previo	El servidor PostgreSQL debe estar en ejecución en el host y puerto especificados, la base de datos dbProyectoCS debe existir, y las credenciales proporcionadas deben ser válidas.
Resultado esperado	La conexión a la base de datos se abre correctamente y el estado de la conexión es "Open".
Resultado obtenido	El estado de la conexión es "Open".
Estado	Exitoso
Observaciones	Conexión establecida exitosamente utilizando Npgsql para PostgreSQL.

Código	CP-009
Caso de prueba	Inicio de sesión exitoso
Responsable	Grupo D
Descripción de la prueba	<p>Esta prueba tiene como objetivo verificar que, al ingresar un nombre de usuario y contraseña válidos, el sistema permita el acceso correctamente. Se hace uso del método "AutenticarUsuario" del "UsuarioRepository" al cual se le pasan los siguientes parámetros:</p> <ul style="list-style-type: none"> • correo: "admin@taller.com" • contraseña: "admin123"

Requisito previo	Tener un usuario registrado previamente en la base de datos con credenciales válidas.
Resultado esperado	El método "AutenticarUsuario" del "UsuarioRepository" devuelve un usuario si las credenciales son correctas.
Resultado obtenido	Acceso concedido.
Estado	Exitoso
Observaciones	Ninguna observación adicional.

Código	CP-010
Caso de prueba	Inicio de sesión fallido por credenciales incorrectas
Responsable	Grupo D
Descripción de la prueba	Esta prueba tiene como objetivo verificar que, al ingresar un nombre de usuario o contraseña incorrectos, el sistema no permita el acceso y responda con un mensaje de error apropiado. Se hace uso del método AutenticarUsuario del UsuarioRepository al cual se le pasan los siguientes parámetros: <ul style="list-style-type: none"> • correo: "admin@taller.com" • contraseña: "claveIncorrecta"
Requisito previo	Tener un usuario registrado previamente en la base de datos con el correo "admin@taller.com". El usuario debe estar activo.
Resultado esperado	El método AutenticarUsuario del UsuarioRepository devuelve null o lanza una excepción controlada indicando que las credenciales son inválidas. El sistema no permite el acceso.
Resultado obtenido	Autenticación fallida. Acceso denegado.
Estado	Exitoso
Observaciones	El sistema respondió correctamente con un mensaje de error al usuario indicando que las credenciales no son válidas. No se permitió el acceso al sistema.

Código	CP-011
Caso de prueba	Registro exitoso de un cliente
Responsable	Grupo D

Descripción de la prueba	<p>Esta prueba tiene como objetivo verificar que, al registrar un nuevo cliente en el sistema, los datos se almacenen correctamente en la base de datos. Se utiliza el método “guardar” del “ClienteRepository”, al cual se le pasa un objeto de tipo “Cliente” con los siguientes datos:</p> <ul style="list-style-type: none"> • apellidos: "Vera" • nombres: "Lucía" • cedula: "1324567890" • correo: lucia.vera@gmail.com • telefono: "0998765432" • direccion: "Av. Colón y 6 de Diciembre"
Requisito previo	Haber iniciado sesión como usuario con permisos administrativos y que la base de datos esté operativa.
Resultado esperado	El método “guardar” del “ClienteRepository” devuelve “True”, confirmando que el cliente fue registrado con éxito.
Resultado obtenido	True
Estado	Exitoso
Observaciones	Los datos ingresados fueron validados previamente por el formulario.

Código	CP-012
Caso de prueba	Actualización exitosa de los datos de un cliente
Responsable	Grupo D
Descripción de la prueba	<p>Esta prueba tiene como objetivo verificar que, al actualizar los datos de un cliente, almacenados en la base de datos, estos se modifiquen correctamente. Se intenta actualizar los datos del cliente utilizando el método “editar” del “ClienteRepository”, el cual recibe un objeto de tipo “Cliente” con los siguientes campos:</p> <ul style="list-style-type: none"> • idCliente: 1 • apellidos: "Páez" • nombres: "Carlos" • cedula: "0947034287" • correo: cpaez@gmail.com • telefono: "0987843874"

	<ul style="list-style-type: none"> direccion: "Av. Quito"
Requisito previo	Haber ingresado correctamente al sistema y que la base de datos esté disponible y accesible.
Resultado esperado	El método "editar" del "ClienteRepository" devuelve "True", indicando que el registro se actualizó correctamente.
Resultado obtenido	True
Estado	Exitoso
Observaciones	Ninguna observación adicional. La prueba se realizó con datos controlados.

Código	CP-013
Caso de prueba	Consulta exitosa de un cliente
Responsable	Grupo D
Descripción de la prueba	<p>Esta prueba tiene como objetivo verificar que el sistema recupere correctamente la información de un cliente registrado al realizar una búsqueda por su cédula.</p> <p>Se utiliza el método "buscarPorCedula" del "ClienteRepository", enviando como parámetro el valor:</p> <ul style="list-style-type: none"> cedula: "0924567890"
Requisito previo	El cliente debe estar previamente registrado en la base de datos y el usuario debe haber iniciado sesión correctamente.
Resultado esperado	El método "buscarPorCedula" devuelve un objeto de tipo "Cliente" con todos los campos completos del registro asociado a la cédula ingresada.
Resultado obtenido	<p>Objeto "Cliente" con datos completos:</p> <ul style="list-style-type: none"> nombres: "Lucía" apellidos: "Vera" correo: lucia.vera@gmail.com telefono: "0998765432" direccion: "Av. Colón y 6 de Diciembre"
Estado	Exitoso
Observaciones	Se comprobó que el resultado es de solo lectura y no permite modificación directa.

Código	CP-014
Caso de prueba	Eliminación lógica exitosa de un cliente
Responsable	Grupo D
Descripción de la prueba	Esta prueba tiene como objetivo verificar que el sistema aplique correctamente una eliminación lógica al cliente registrado, de forma que no se eliminen físicamente sus datos de la base de datos. Se utiliza el método “eliminar” del “ClienteRepository”, enviando el parámetro: <ul style="list-style-type: none"> cedula: "1324567890"
Requisito previo	El cliente debe estar previamente registrado y no debe tener órdenes activas asociadas. El usuario debe tener permisos administrativos.
Resultado esperado	El método “eliminar” devuelve “True”, y el estado del cliente cambia a “inactivo” o “eliminado” sin borrar el registro de la base de datos.
Resultado obtenido	True
Estado	Exitoso
Observaciones	Se verificó que el cliente ya no aparece en las consultas normales, pero sigue presente en la base de datos.

Código	CP-015
Caso de prueba	Registro exitoso de un vehículo
Responsable	Grupo D
Descripción de la prueba	Esta prueba tiene como objetivo verificar que, al registrar un nuevo vehículo vinculado a un cliente existente, el sistema almacene correctamente la información en la base de datos. Se utiliza el método “registrarVehiculo” del “VehiculoRepository”, con un objeto “Vehiculo” que contiene los siguientes datos: <ul style="list-style-type: none"> idCliente: 1 marca: "Chevrolet" modelo: "Aveo Family" año: 2013 placa: "ABC1234" dueño: "Juan Pérez"

Requisito previo	El cliente asociado debe estar registrado y activo en la base de datos. El usuario debe estar autenticado con permisos para registrar vehículos.
Resultado esperado	El método "registrarVehiculo" devuelve "True", indicando que el vehículo fue almacenado exitosamente en la base de datos.
Resultado obtenido	True
Estado	Exitoso
Observaciones	La placa fue validada como única antes del registro.

Código	CP-016
Caso de prueba	Consulta exitosa de un vehículo
Responsable	Grupo D
Descripción de la prueba	Esta prueba tiene como objetivo verificar que el sistema recupere correctamente la información de un vehículo previamente registrado, al realizar una búsqueda por su placa. Se utiliza el método "buscarPorPlaca" del "VehiculoRepository" con el siguiente parámetro: <ul style="list-style-type: none"> placa: "ABC1234"
Requisito previo	El vehículo debe estar registrado y asociado a un cliente existente. El usuario debe haber iniciado sesión correctamente.
Resultado esperado	El método "buscarPorPlaca" devuelve un objeto "Vehiculo" con todos sus campos, incluyendo la relación con el cliente.
Resultado obtenido	Objeto "Vehiculo" con datos completos: <ul style="list-style-type: none"> marca: "Chevrolet" modelo: "Aveo Family" año: 2013 color: "Plata" dueño: "Lucía Vera"
Estado	Exitoso
Observaciones	El sistema solo permite visualizar los datos, sin opción de edición directa en la consulta.

Código	CP-017
Caso de prueba	Actualización exitosa de los datos de un vehículo

Responsable	Grupo D
Descripción de la prueba	<p>Esta prueba tiene como objetivo verificar que, al actualizar los datos de un vehículo existente (excepto la placa), los cambios se reflejen correctamente en la base de datos. Se utiliza el método “editar” del “VehiculoRepository” enviando un objeto “Vehiculo” con los siguientes valores modificados:</p> <ul style="list-style-type: none"> • idVehiculo: 12 • marca: "Chevrolet" • modelo: "Aveo GT" • año: 2014 • placa: "ABC1234" (no modificable) • color: "Gris oscuro"
Requisito previo	El vehículo debe estar previamente registrado y el usuario debe tener permisos de edición.
Resultado esperado	El método “editar” del “VehiculoRepository” devuelve “True”, y los nuevos datos se reflejan correctamente al consultarlos.
Resultado obtenido	True
Estado	Exitoso
Observaciones	La placa se mantuvo inmutable según las restricciones del sistema.

Código	CP-018
Caso de prueba	Eliminación lógica exitosa de un vehículo
Responsable	Grupo D
Descripción de la prueba	<p>Esta prueba tiene como objetivo verificar que el sistema elimine lógicamente un vehículo registrado, siempre que no esté vinculado a órdenes de servicio activas.</p> <p>Se utiliza el método “eliminar” del “VehiculoRepository” con el siguiente parámetro:</p> <ul style="list-style-type: none"> • placa: "ABC1234"
Requisito previo	El vehículo debe estar registrado y no estar vinculado a una orden de servicio activa. El usuario debe estar autenticado con rol administrativo.

Resultado esperado	El método "eliminar" devuelve "True", y el estado del vehículo se actualiza a "inactivo".
Resultado obtenido	True
Estado	Exitoso
Observaciones	El vehículo ya no aparece en los listados activos, pero sigue disponible en la base de datos para auditoría.

Código	CP-019
Caso de prueba	Creación exitosa de una orden de servicio
Responsable	Grupo D
Descripción de la prueba	<p>Esta prueba tiene como objetivo verificar que el sistema pueda crear correctamente una nueva orden de servicio en el sistema. Se utiliza el método crearOrden del OrdenRepository con los siguientes parámetros:</p> <ul style="list-style-type: none"> • cliente: "Juan Pérez" • Vehículo: "Kia" • fecha: "2025-04-30" • descripcion: "Cambio de aceite" • estado: "Pendiente" • ítems de trabajo: "Cantidad", "Precio" "Total"
Requisito previo	El vehículo debe estar registrado y disponible en el sistema. El usuario debe estar autenticado con rol administrativo.
Resultado esperado	El método crearOrden devuelve un ID de orden generado, y la nueva orden de servicio se guarda en el sistema con el estado "Pendiente".
Resultado obtenido	ID generado: 10
Estado	Exitoso
Observaciones	Ninguna observación.

Código	CP-020
Caso de prueba	Consulta del estado de una orden de servicio
Responsable	Grupo D
Descripción de la prueba	Esta prueba tiene como objetivo verificar que el sistema pueda mostrar correctamente el estado actual de una orden de servicio. Se consulta el estado de una orden de servicio con ID 1.

Requisito previo	Haber creado una orden de servicio con ID 1 y asignado un estado.
Resultado esperado	El sistema devuelve el estado de la orden de servicio correctamente (por ejemplo, "En progreso").
Resultado obtenido	"En progreso"
Estado	Exitoso
Observaciones	Es importante que el sistema actualice el estado de la orden a medida que cambia durante su ciclo de vida (pendiente, en progreso, completada).

Código	CP-021
Caso de prueba	Actualización de estado de la orden de servicio
Responsable	Grupo D
Descripción de la prueba	Esta prueba tiene como objetivo verificar que el sistema actualice correctamente el estado de una orden de servicio. Se utiliza el método actualizarEstado del OrdenRepository con los siguientes parámetros: <ul style="list-style-type: none"> ordenId: 10 nuevoEstado: "Completada"
Requisito previo	La orden de servicio con ordenId 201 debe existir y estar en estado "Pendiente". El usuario debe estar autenticado con rol administrativo.
Resultado esperado	El estado de la orden se actualiza correctamente a "Completada".
Resultado obtenido	Estado actualizado: "Completada"
Estado	Exitoso
Observaciones	El estado de la orden se refleja correctamente en el sistema y ya no aparece en los listados de órdenes pendientes.

Código	CP-022
Caso de prueba	Eliminación lógica de una orden de servicio
Responsable	Grupo D
Descripción de la prueba	Esta prueba tiene como objetivo verificar que el sistema elimine lógicamente una orden de servicio en estado "Completada". Se utiliza el método

	eliminarOrden del OrdenRepository con el siguiente parámetro: <ul style="list-style-type: none"> ordenId: 10
Requisito previo	La orden de servicio con ordenId 10 debe existir y estar en estado "Completada". El usuario debe estar autenticado con rol administrativo.
Resultado esperado	El método eliminarOrden devuelve "True", y el estado de la orden se actualiza a "Eliminada".
Resultado obtenido	True
Estado	Exitoso
Observaciones	La orden ya no aparece en los listados de órdenes activas, pero sigue disponible en la base de datos para auditoría.

Código	CP-023
Caso de prueba	Registro de un nuevo repuesto en el inventario
Responsable	Grupo D
Descripción de la prueba	Esta prueba tiene como objetivo verificar que el sistema permita crear un nuevo repuesto en el inventario. Se utiliza el método crearRepuesto del StockRepository con los siguientes parámetros: <ul style="list-style-type: none"> nombre del repuesto: "Filtro de aceite" Categoría: "Motor" codigo: "R001" stock actual: 50 stock mínimo: 20 proveedor: "Carlos Gómez" notas: "Realizado con éxito" precioUnitario: \$15.00
Requisito previo	El usuario debe estar autenticado con rol administrativo y tener acceso al módulo de inventario.
Resultado esperado	El repuesto es agregado correctamente al inventario con el código "R001", nombre "Filtro de aceite", stock actual 50 y precio unitario de \$15.00.
Resultado obtenido	Repuesto agregado con código "R001".
Estado	Exitoso

Observaciones	El nuevo repuesto aparece correctamente en el listado de inventarios con la cantidad y precio especificados.
----------------------	--

Código	CP-024
Caso de prueba	Consulta de repuesto en el inventario
Responsable	Grupo D
Descripción de la prueba	<p>Esta prueba tiene como objetivo verificar que el sistema permita consultar correctamente la información de un repuesto existente en el inventario.</p> <p>Se hace uso del método buscarRepuestoPorCodigo del StockRepository al cual se le pasa el siguiente parámetro:</p> <ul style="list-style-type: none"> código: "R001"
Requisito previo	El repuesto con código "R001" debe estar registrado y activo en el inventario. El usuario debe estar autenticado con permisos para visualizar el módulo de stock.
Resultado esperado	El método buscarRepuestoPorCodigo devuelve un objeto con los datos del repuesto: código, nombre, cantidad disponible y precio unitario.
Resultado obtenido	{ código: "R001", nombre: "Filtro de aceite", stock actual: 50, precioUnitario: 15.00 }
Estado	Exitoso
Observaciones	La información del repuesto se muestra correctamente en pantalla y coincide con los valores almacenados en la base de datos.

Código	CP-025
Caso de prueba	Actualización de cantidad de repuesto en el inventario
Responsable	Grupo D
Descripción de la prueba	Esta prueba tiene como objetivo verificar que el sistema permita actualizar la cantidad de un repuesto en el inventario. Se utiliza el método actualizarCantidad del StockRepository con los siguientes parámetros:

	<ul style="list-style-type: none"> codigo: "R001" nuevaRepuesto: "Cambio de llantas"
Requisito previo	El repuesto con codigo "R001" debe existir en el inventario. El usuario debe estar autenticado con rol administrativo.
Resultado esperado	El nuevo repuesto con código "R001" se actualiza correctamente.
Resultado obtenido	Repuesto de llantas actualizada.
Estado	Exitoso
Observaciones	El nuevo repuesto "Cambio de llantas" se refleja correctamente en el inventario.

Código	CP-026
Caso de prueba	Eliminación lógica de un repuesto del inventario
Responsable	Grupo D
Descripción de la prueba	Esta prueba tiene como objetivo verificar que el sistema permita eliminar lógicamente un repuesto del inventario. Se utiliza el método eliminarRepuesto del StockRepository con el siguiente parámetro: <ul style="list-style-type: none"> codigo: "R001"
Requisito previo	El repuesto con codigo "R001" debe existir en el inventario. El usuario debe estar autenticado con rol administrativo.
Resultado esperado	El método eliminarRepuesto devuelve "True", y el estado del repuesto se actualiza a "Eliminado".
Resultado obtenido	True
Estado	Exitoso
Observaciones	El repuesto "Filtro de aceite" ya no aparece en el inventario activo, pero sigue disponible en la base de datos para auditoría.

6. Criterios de entrada y de salida

Es muy importante definir correctamente estos criterios y comunicarlos a toda la organización para que no queden dudas sobre ellos.

6.1. Criterio de entrada del plan de pruebas

- El paquete completo de código fuente está terminado y ha sido revisado de manera informal por los miembros del grupo D.
- Los casos de prueba están debidamente documentados y han sido revisados.
- Todas las herramientas necesarias para realizar las pruebas están correctamente instaladas y configuradas.
- Los datos de prueba necesarios han sido preparados y validados para garantizar que cubren los escenarios más relevantes posibles.

6.2. Criterio de aceptación del plan de pruebas

- Para considerar que el plan de pruebas ha sido concluido con éxito, deben cumplirse los siguientes requisitos:
- Se han realizado todas las pruebas planificadas y los resultados han sido debidamente documentados.
- Se ha alcanzado un nivel aceptable de defectos pendientes.
- Se ha verificado que el sistema cumple con las especificaciones previamente establecidas.

6.3. Criterio de suspensión y reanudación

Condiciones de suspensión:

- Fallos en el acceso de usuarios que impidan el ingreso al sistema.
- Inestabilidad en el entorno de pruebas, como fallos en Visual Studio o NUnit.
- Errores graves en la estructura de la base de datos.
- Problemas de conexión con la base de datos.

Condiciones de reanudación:

- Restablecimiento y estabilización del entorno de pruebas.
- Resolución y verificación de errores críticos en las funcionalidades principales.
- Revisión y ajuste de la estructura de la base de datos.
- Comprobación y solución de fallos en la conexión con la base de datos.

7. Entregables

Dentro del proceso de pruebas se contemplan los siguientes entregables.

7.1. Protocolo de pruebas

El documento de plan de pruebas que contiene los casos de pruebas que se realizarán y los resultados.

7.2. Informe de pruebas

El informe de pruebas contendrá la evaluación de resultados con los siguientes puntos:

- Resumen de pruebas, el cual incluye la identificación del caso de prueba y el resultado de la ejecución.
- Acciones sugeridas.
- Análisis de Resultados.

7.3. Reporte de observaciones

Durante las pruebas del Sistema de Gestión de Taller Vehicular, cada incidencia detectada se documenta en un reporte que incluye el módulo afectado (Clientes, Vehículos, Órdenes o Repuestos), el nivel de severidad, la prioridad de solución y los pasos para reproducir el error. El líder de Proyecto evalúa cada reporte y, si se requiere modificar algún requerimiento, se genera una solicitud de cambio y se actualiza el documento de especificaciones. Este proceso asegura que cualquier modificación se mantenga alineada con los objetivos originales del sistema.

8. Ambiente de pruebas

8.1. Requerimientos base de hardware

Recurso	Cantidad	Nombre y tipo
Computadora o laptop	1	Procesador Intel i5 o superior, 8 GB de RAM, 256 GB SSD

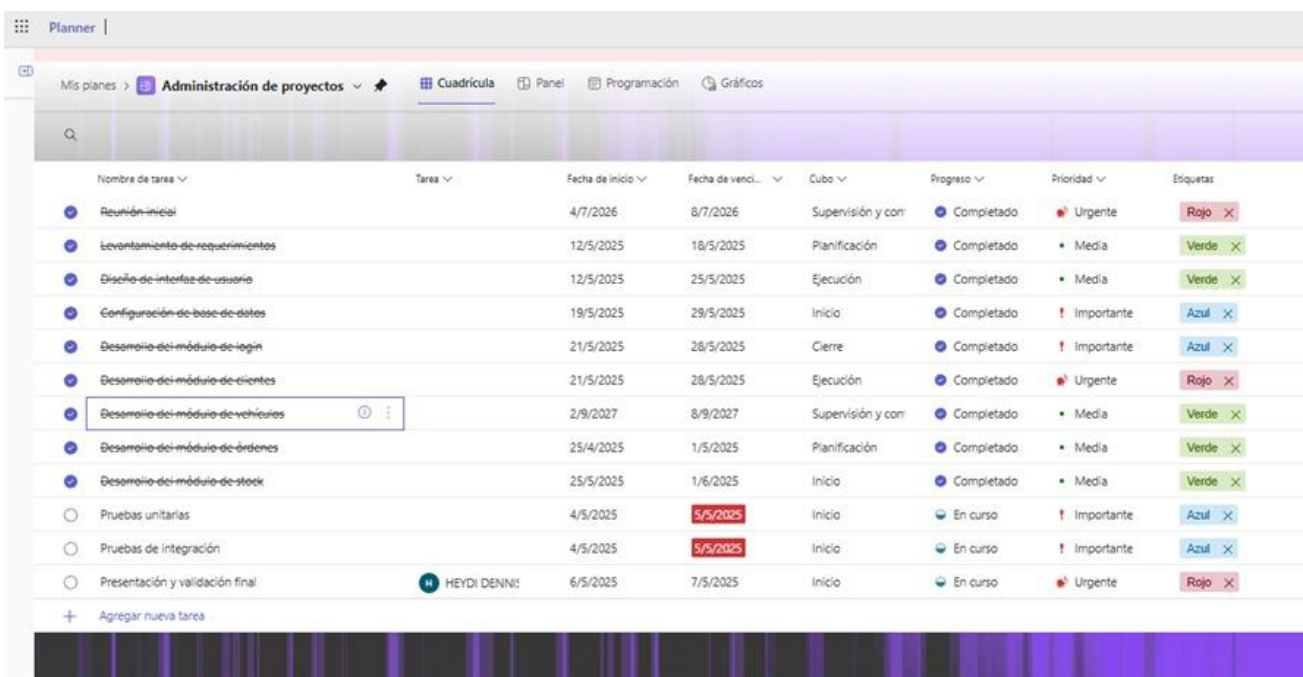
8.2. Requerimientos base de software en el ambiente de pruebas

Elemento de software	Versión	Tipo
Windows	11	Sistema Operativo
.NET	6.0	Framework
Visual Studio Code	Última	IDE
PostgreSQL	15	RDBMS (gestor de base de datos)
pgAdmin	6.0+	IDE para PostgreSQL

8.3. Herramientas de apoyo para la ejecución de pruebas

Elemento de software	Versión	Tipo
NUnit	4.1	Framework de pruebas de unidad para Microsoft
Microsoft Project	x.0	Software de administración de proyectos y programas de proyectos desarrollados.

9. Cronograma



10. Riesgos, dependencias, suposiciones y restricciones

10.1. Riesgos

Riesgo	Estrategia de mitigación	Contingencia
Falta de tiempo para ejecutar todas las pruebas	Incluir un margen del 10% en el cronograma original para absorber posibles demoras.	Redistribuir tareas o aumentar temporalmente el equipo de pruebas.
Errores complejos que retrasen la integración	Priorizar la solución de errores que impidan la comunicación entre módulos clave.	Asignar un desarrollador adicional para resolver errores de integración críticos.

Fallas en la configuración del entorno de pruebas	Validar previamente las herramientas necesarias (xUnit, NUnit, mocks, CI/CD) antes de iniciar las pruebas.	Redefinir un entorno alternativo o local para pruebas urgentes.
---	--	---

10.2. Dependencias

Dependencia con otras áreas	Impacto de dependencia
Interdependencia entre proyectos	Bajo
Dependencia de herramientas de desarrollo	Alto

10.3. Suposiciones

Suposición a ser aprobada	Impacto de suposición incorrecta
El equipo de pruebas contará con acceso al entorno de desarrollo y a los ambientes de prueba.	<ul style="list-style-type: none"> - Imposibilidad de ejecutar pruebas. - Retrasos en las entregas. - Dependencia de otros equipos.
Los requerimientos funcionales estarán claros y completos.	<ul style="list-style-type: none"> - Resultados de pruebas inconsistentes. - Retrasos en los tiempos de entrega. - Necesidad de redefinir requerimientos. - Afectación en módulos relacionados.
El código fuente está modularizado y cumple con los principios de diseño para facilitar pruebas unitarias.	<ul style="list-style-type: none"> - Dificultad para aislar componentes. - Aumento del tiempo de prueba. - Resultados no confiables.

10.4. Restricciones

Restricciones	Impacto de la restricción
Fecha límite de entrega del plan de pruebas	<ul style="list-style-type: none"> - El cronograma ajustado requiere completar todas las pruebas antes de la fecha definida. - Riesgo de no cumplir con los entregables si surgen imprevistos.
Uso exclusivo de herramientas gratuitas o con licencia vigente	<ul style="list-style-type: none"> - Posibles limitaciones en las funcionalidades de pruebas (mocking, generación de reportes, cobertura). - Riesgo de interrupción si se vence alguna licencia gratuita de prueba.
Disponibilidad parcial del equipo de pruebas	<ul style="list-style-type: none"> - Posible retraso en la ejecución de pruebas unitarias e integración. - Sobrecarga de trabajo para los integrantes disponibles.

	- Disminución en la revisión cruzada y validación de casos.
Cambios en los módulos del sistema durante la fase de pruebas	<ul style="list-style-type: none">- Necesidad de modificar o reescribir pruebas unitarias e integración.- Mayor esfuerzo de mantenimiento de los scripts de prueba.- Posible pérdida de trazabilidad.
Presupuesto restringido para automatización o herramientas avanzadas	<ul style="list-style-type: none">- Limitación para implementar pruebas automatizadas o ejecutar pruebas en múltiples entornos.- Dependencia de herramientas básicas o manuales para la validación.