

Universidad Estatal a Distancia

Vicerrectoría Académica

Escuela De Ciencias Exactas y Naturales

Carrera de Bachillerato en Ingeniería Informática

Asignatura: Ingeniería del Software

Código: 03300

PROYECTO 3

VALOR: 30% (3.0)

Estudiante:

Francisco Campos Sandi

114750560

Sede: San Vito

Grupo 01

Tutora: Iliana Segura Picado

III Cuatrimestre 2024

## Contenido

Introducción .....	4
Pregunta 1 .....	5
Prototipado módulo de servicios.....	5
Pregunta 2 .....	6
Diálogo de Validación de la Aplicación .....	6
Pregunta 3 .....	9
Tipos de flujos caso 1.....	10
Caso 1: Agregar el servicio "Tratamiento Cerámico" .....	11
Tipos de flujos caso 2.....	12
Caso 2: Modificar cola con estado "En revisión" .....	13
Tipos de flujos caso 3.....	14
Caso 3: Validación de datos para Tratamiento Cerámico .....	15
Tipos de flujos caso 4.....	16
Caso 4: Reporte con estado "En revisión" .....	17
Pregunta 4 .....	18
Pregunta 5 .....	21
Conclusiones .....	23
Referencias .....	24

## **Tabla de ilustraciones**

<b>Ilustración 1 Módulo de servicios - Lavacar .....</b>	<b>5</b>
<b>Ilustración 2 flujo caso 1 .....</b>	<b>10</b>
<b>Ilustración 3 flujos caso 2 .....</b>	<b>12</b>
<b>Ilustración 4 flujos caso 3 .....</b>	<b>14</b>
<b>Ilustración 5 flujos caso 4 .....</b>	<b>16</b>

## **Introducción**

El presente trabajo aborda aspectos esenciales de la Ingeniería de Software, específicamente relacionados con el diseño, las métricas y los procesos de pruebas en sistemas informáticos. El caso práctico desarrollado para CleanCarPlus permite explorar cómo estos elementos teóricos se aplican en un entorno real, destacando la importancia de un diseño funcional, validaciones robustas y pruebas efectivas para garantizar el éxito de un sistema.

La relevancia de este estudio radica en su contribución a la calidad del software, un elemento crítico en el mundo tecnológico actual. Un diseño sólido y bien fundamentado garantiza no solo la funcionalidad, sino también la satisfacción del usuario final. Además, las métricas de software permiten evaluar objetivamente el rendimiento y la calidad del sistema, facilitando la mejora continua.

Entre los objetivos fundamentales de este trabajo, se incluye la necesidad de conocer a profundidad los principios de diseño de software, comprendiendo cómo influyen en la creación de sistemas eficientes y adaptables. Asimismo, se busca profundizar en el conocimiento de las métricas de software, herramientas clave para evaluar y optimizar procesos y explorar el proceso de pruebas,

Este análisis se desarrolla utilizando un caso práctico centrado en un sistema para CleanCarPlus, el cual representa un entorno empresarial típico con requerimientos específicos. A través de este caso, se integran conceptos teóricos y prácticos que permiten a los estudiantes adquirir habilidades aplicables en escenarios reales. En consecuencia, el trabajo no solo fortalece el aprendizaje técnico, sino que también fomenta la capacidad de análisis y resolución de problemas en el ámbito del desarrollo de software.

## Pregunta 1

### Prototipado módulo de servicios

The image shows a software window titled "Módulo de Servicios - Lavacar". It contains two main sections. The top section, titled "REGISTRAR SERVICIO", is a form for registering a service. It includes four input fields: "Tipo de Servicio:" with a dropdown menu showing "Limpieza tapicería", "Duración del Servicio (horas):" with a text input containing "1", "Costo del Servicio (€):" with a text input containing "15000", and "Productos y Servicios Asociados:" with a dropdown menu showing "Lavado, Cera, Aspirado". Below these fields are two buttons: "Registrar" (blue) and "Cancelar" (red). The bottom section, titled "RESULTADO REGISTRO", displays the registered information in a list format. It shows the "Identificador:" as "8407598", "Tipo de Servicio:" as "Limpieza tapicería", "Duración del Servicio:" as "1 horas", "Costo del Servicio:" as "€15000.00", and "Productos Asociados:" as "Lavado, Cera, Aspirado". Below this list is a single green button labeled "Aceptar".

**REGISTRAR SERVICIO**

Tipo de Servicio: Limpieza tapicería

Duración del Servicio (horas): 1

Costo del Servicio (€): 15000

Productos y Servicios Asociados: Lavado, Cera, Aspirado

Registrar Cancelar

**RESULTADO REGISTRO**

Identificador: 8407598

Tipo de Servicio: Limpieza tapicería

Duración del Servicio: 1 horas

Costo del Servicio: €15000.00

Productos Asociados: Lavado, Cera, Aspirado

Aceptar

*Ilustración 1 Módulo de servicios - Lavacar*

## Pregunta 2

### Diálogo de Validación de la Aplicación

**Analista:**

Aquí puedes ver cómo será la funcionalidad para registrar un servicio. ¿Qué opinas de la interfaz?

**Usuario:**

Déjame ver... ¿Qué representa este campo "Tipo de Servicio"?

**Analista:**

El campo "Tipo de Servicio" permite elegir entre las opciones de servicios que ofrecemos, como "Limpieza tapicería", "Lavado general", "Mantenimiento Cuero", y "Cambio de aceite". Este campo es obligatorio para registrar el servicio.

**Usuario:**

Entiendo, esas opciones están bien. ¿Y el campo "Duración del Servicio"?

**Analista:**

El campo "Duración del Servicio" debe ingresar el tiempo estimado para el servicio en horas. Es un campo numérico que se usará para calcular la duración del trabajo. ¿Está claro?

**Usuario:**

Sí, entiendo. ¿Y qué pasa con el "Costo del Servicio"?

**Analista:**

El campo "Costo del Servicio" es obligatorio y debe contener el valor del servicio en

colones (₡). Este es un campo importante para procesar el pago. Si se deja vacío, el sistema mostrará un mensaje de error pidiendo que se complete.

**Usuario:**

Eso tiene sentido. ¿Y qué es este campo "Productos y Servicios Asociados"?

**Analista:**

En este campo, puedes seleccionar los productos adicionales relacionados con el servicio, como "Lavado Externo" o "Lavado y Cera". Estos son servicios complementarios que puedes agregar. ¿Te gustaría incluir más opciones aquí?

**Usuario:**

Las opciones actuales están bien. Pero, ¿qué sucede cuando presiono "Registrar"?

**Analista:**

Cuando presionas "Registrar", el sistema verificará si todos los campos obligatorios están completos. Si todo está correcto, mostrará un resumen con los detalles del servicio registrado. Si falta algún campo, se mostrará un mensaje pidiendo que lo completes.

**Usuario:**

Perfecto, y si por alguna razón quiero cancelar, ¿cómo se hace?

**Analista:**

Si presionas "Cancelar", se borrarán todos los datos ingresados y el sistema volverá al estado inicial, listo para registrar un nuevo servicio.

**Usuario:**

Eso está claro. Me parece que todo está bien diseñado hasta ahora. ¿Qué pasa con los resultados después de registrar un servicio?

**Analista:**

Después de hacer el registro, los detalles del servicio aparecerán en la sección "Resultado Registro". Allí se mostrará el tipo de servicio, la duración, el costo y los productos adicionales que hayas seleccionado.

**Usuario:**

Eso está perfecto. ¿Sería posible ver también la fecha en la que se registra el servicio?

**Analista:**

Sí, podemos agregar la fecha de registro si lo deseas. La mostraríamos junto con los demás datos en el resultado.

**Usuario:**

Me parece una buena idea. Agreguémoslo.

**Analista:**

Listo, lo incorporamos. Ahora, ¿todo está claro con la interfaz y los campos que hemos discutido?

**Usuario:**

Sí, todo está claro. Podemos continuar con la implementación.



### **Pregunta 3**

#### **Requerimientos de emergencia**

##### **1. Tratamiento Cerámico:**

Este nuevo servicio debe ser incorporado al módulo de servicios. Para implementarlo, se seguiría el flujo básico del sistema: un administrador ingresa al módulo de servicios, agrega el nuevo tipo de servicio, define su duración, costo y productos asociados. En el flujo alternativo, si algún dato necesario no está disponible, el sistema debe advertir al usuario para su corrección antes de guardar el registro. La validación asegura que los datos ingresados sean correctos y no duplicados.

##### **2. Estado "En revisión":**

Este estado se implementaría en el módulo de estado de servicios. El flujo básico implica que, al actualizar el estado del vehículo, el usuario selecciona "En revisión", lo que permite que el vehículo mantenga su lugar en la cola sin avanzar al siguiente paso. En un flujo alternativo, si no hay un vehículo en cola, el sistema debe notificarlo. La validación confirmará que este nuevo estado sea compatible con el método FIFO y pueda integrarse al reporte de estado.

## Tipos de flujos caso 1

### Caso 1: Agregar el servicio "Tratamiento Cerámico"



Ilustración 2 flujo caso 1

### Caso 1: Agregar el servicio "Tratamiento Cerámico"

Atributo	Descripción
Nombre del caso de prueba	Adición del servicio Tratamiento Cerámico.
Objetivo	Validar que el sistema permita agregar correctamente un nuevo servicio.
Requisitos	Acceso al módulo de servicios como administrador, además la base de datos conectada.
Pasos	<b>Pasos:</b> <ol style="list-style-type: none"><li>1. Iniciar sesión como administrador.</li><li>2. Acceder al módulo de servicios.</li><li>3. Seleccionar "Agregar servicio".</li><li>4. Introducir los datos: nombre (Tratamiento Cerámico), duración, costo y productos asociados.</li><li>5. Guardar los cambios.</li></ol>
Datos de Entrada	<ul style="list-style-type: none"><li>• Nombre: Tratamiento Cerámico.</li><li>• Duración: 120 minutos.</li><li>• Costo: ₡ 25 000.</li><li>• Productos asociados: Sellador cerámico, esponjas.</li></ul>
Resultados Esperados	El sistema muestra un mensaje de confirmación: "Servicio agregado correctamente".  El nuevo servicio aparece en el listado del módulo.
Ambiente de Pruebas	Base de datos en entorno local con acceso controlado.
Estado	Aprobado

## Tipos de flujos caso 2

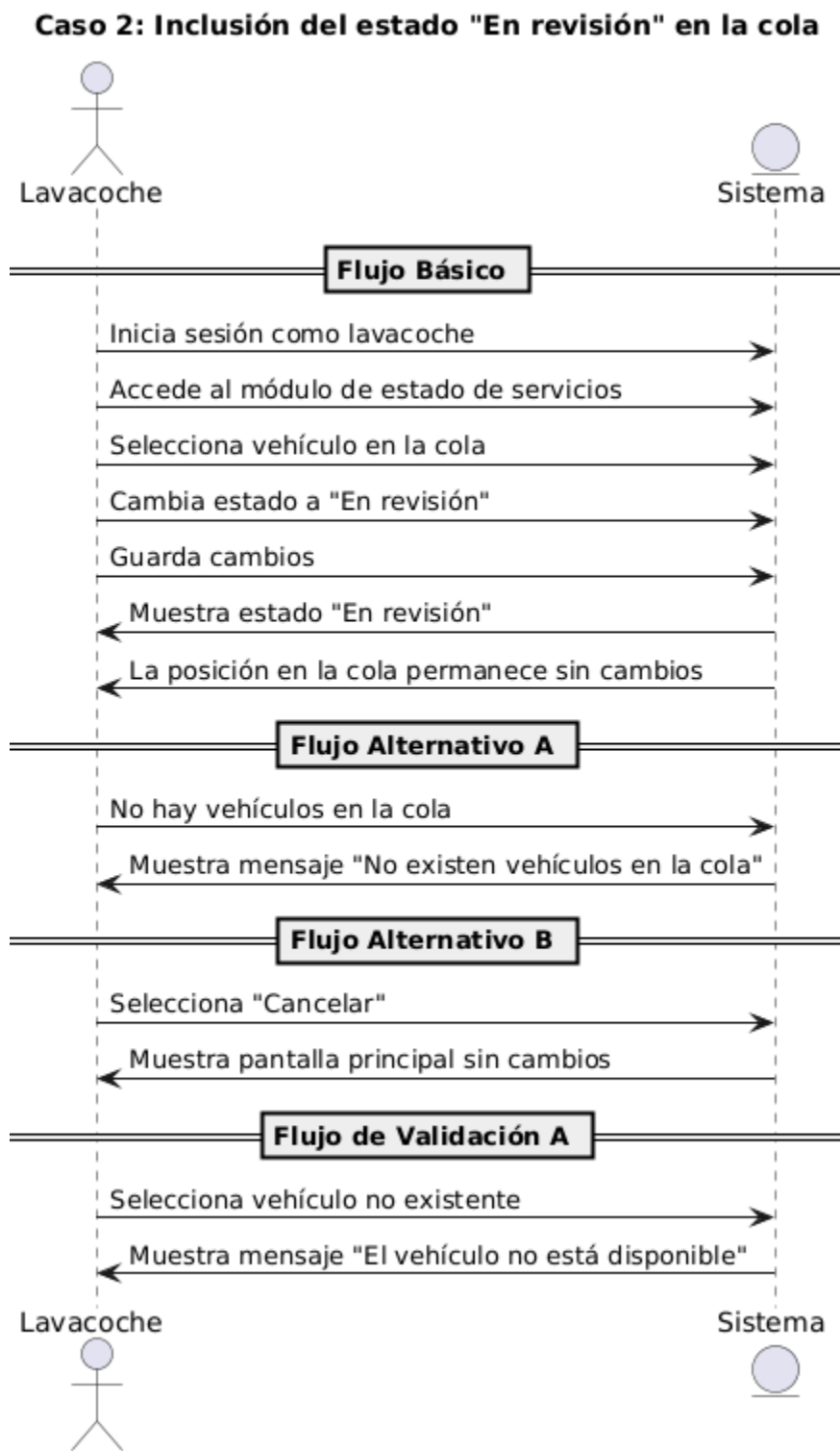


Ilustración 3 flujos caso 2

### Caso 2: Modificar cola con estado "En revisión"

Atributo	Descripción
Nombre del caso de prueba	Inclusión del estado "En revisión" en la cola.
Objetivo	Validar que el nuevo estado pueda ser asignado y que el sistema respete el método FIFO.
Requisitos	Módulo de estado de servicios funcional y la base de datos activa.
Pasos	<b>Pasos:</b> <ol style="list-style-type: none"><li>1. Iniciar sesión como lavacoche.</li><li>2. Acceder al módulo de estado de servicios.</li><li>3. Seleccionar un vehículo en la cola.</li><li>4. Cambiar el estado a "En revisión".</li><li>5. Guardar los cambios y observar el comportamiento de la cola.</li></ol>
Datos de Entrada	<ul style="list-style-type: none"><li>• Placa: BFC808.</li><li>• Estado nuevo: En revisión.</li></ul>
Resultados Esperados	El vehículo seleccionado muestra el estado "En revisión".  La posición en la cola permanece sin cambios.
Ambiente de Pruebas	Entorno controlado con datos simulados.
Estado	En revisión.

### Tipos de flujos caso 3

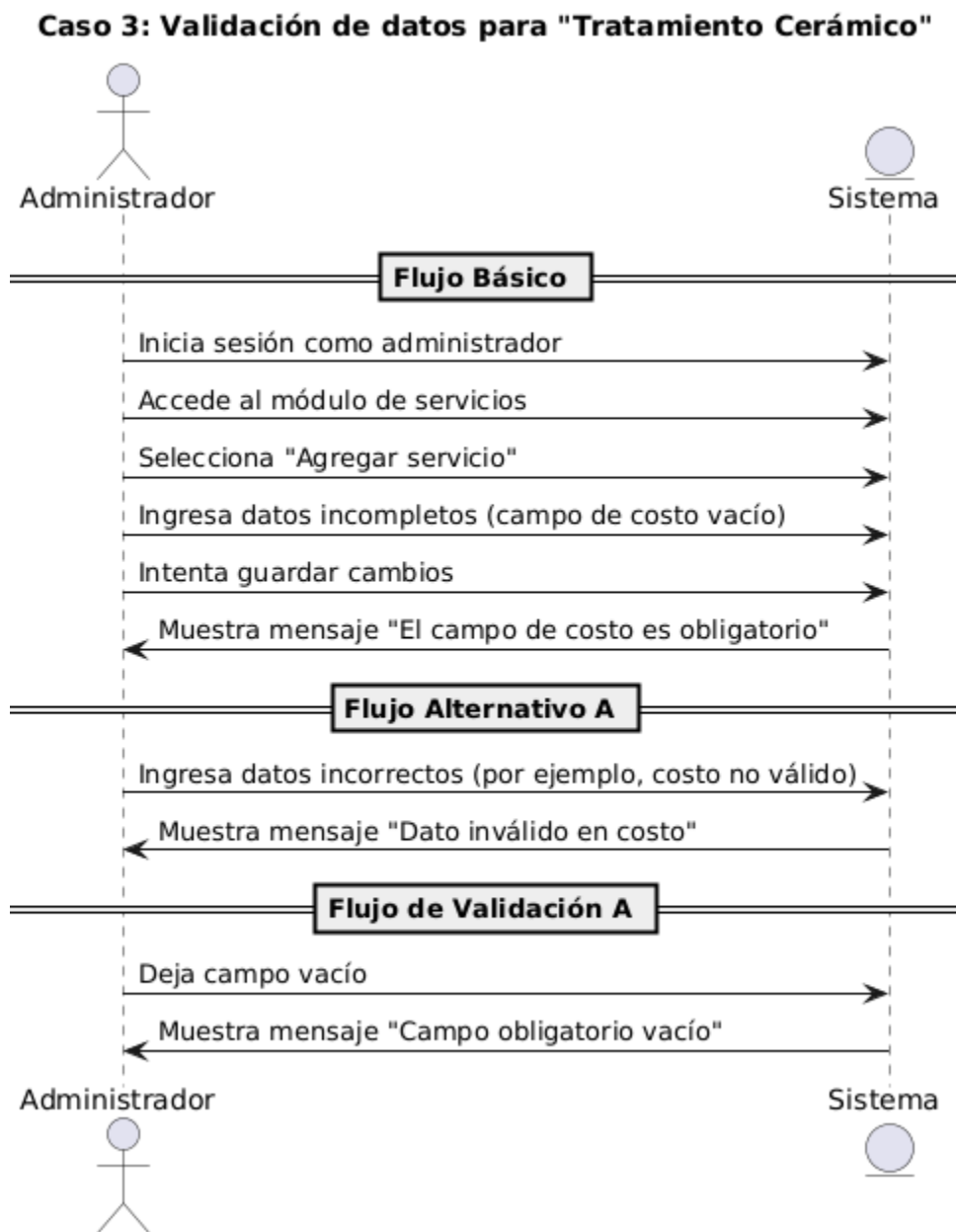
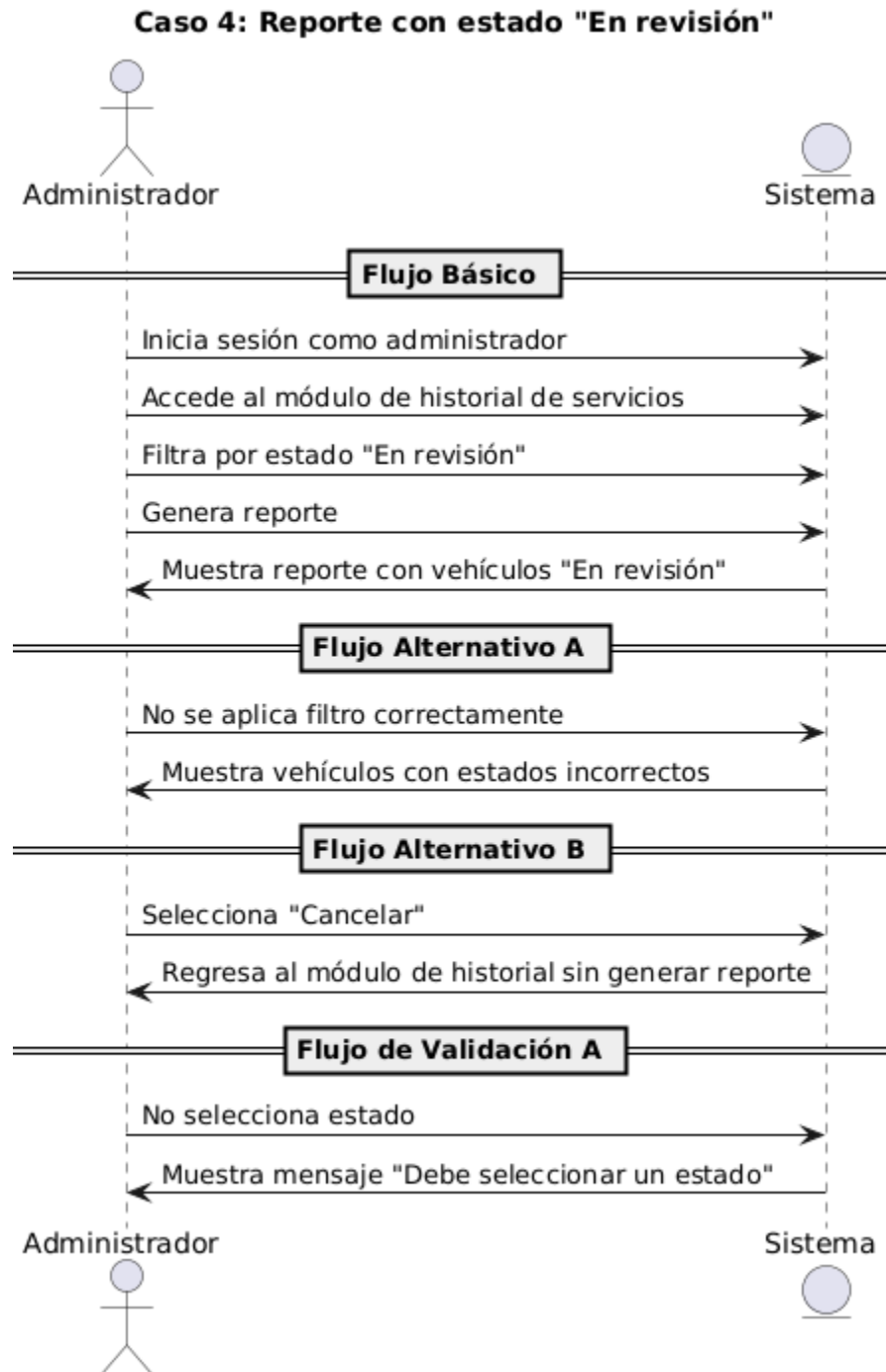


Ilustración 4 flujos caso 3

### Caso 3: Validación de datos para Tratamiento Cerámico

Atributo	Descripción
Nombre del caso de prueba	Validación de datos al registrar Tratamiento Cerámico.
Objetivo	Comprobar que los datos requeridos sean validados antes de guardar.
Requisitos	Módulo de servicios con validación activa; conexión a base de datos.
Pasos	<b>Pasos:</b> <ol style="list-style-type: none"><li>1. Iniciar sesión como administrador.</li><li>2. Acceder al módulo de servicios.</li><li>3. Seleccionar "Agregar servicio".</li><li>4. Dejar en blanco el campo de costo y guardar.</li></ol>
Datos de Entrada	<b>Datos de entrada:</b> <ul style="list-style-type: none"><li>• Nombre: Tratamiento Cerámico.</li><li>• Duración: 120 minutos.</li><li>• Costo:</li></ul>
Resultados Esperados	El sistema muestra un mensaje de error: "El campo de costo es obligatorio".
Ambiente de Pruebas	Base de datos con validación activa.
Estado	Aprobado.

## Tipos de flujos caso 4



*Ilustración 5 flujos caso 4*



#### Caso 4: Reporte con estado "En revisión"

Atributo	Descripción
Nombre del caso de prueba	Generar reporte con estado "En revisión".
Objetivo	Validar que el nuevo estado pueda ser incluido en los reportes generados.
Requisitos	Módulo de historial funcional; reporte configurado para incluir nuevos estados.
Pasos	<b>Pasos:</b> <ol style="list-style-type: none"><li>1. Iniciar sesión como administrador.</li><li>2. Acceder al módulo de historial de servicios.</li><li>3. Filtrar por estado: En revisión.</li><li>4. Generar reporte.</li></ol>
Datos de Entrada	<b>Datos de entrada:</b> <ul style="list-style-type: none"><li>• Filtro: Estado = "En revisión".</li></ul>
Resultados Esperados	El reporte muestra todos los vehículos con estado "En revisión".
Ambiente de Pruebas	Sistema conectado a base de datos local.
Estado	Aprobado

## **Pregunta 4**

### **Criterios de un buen diseño desde la perspectiva de la Ingeniería de Software**

#### **1. Economía**

El diseño debe evitar características innecesarias y complejidades que no aporten al propósito del sistema. Esto implica que los atributos del diseño deben ser proporcionales al problema que se busca resolver. Según se señala, "la flexibilidad arbitraria, características innecesarias y complejidad en el diseño afectan la economía del sistema al generar sobrecarga" (Pantaleo & Rinaudo, 2015, p. 222). Por tanto, la reutilización debe priorizarse sin comprometer la usabilidad.

#### **2. Visibilidad**

La claridad en el diseño es esencial para facilitar su comprensión y mantenimiento. En este sentido, el uso de patrones de diseño permite que el diseño sea más evidente, ya que "estos patrones hacen más visible el diseño al estandarizar y organizar sus elementos" (Pantaleo & Rinaudo, 2015, p. 222). Esto asegura que los desarrolladores puedan trabajar de forma más efectiva con el sistema.

#### **3. Espaciamento**

El espaciamento en el diseño implica una adecuada separación de elementos, como paquetes, basada en la cohesión. Este criterio contribuye a la modularidad del sistema, ya que "la separación en paquetes con base en un criterio de cohesión garantiza un diseño organizado y funcional" (Pantaleo & Rinaudo, 2015, p. 223). Esto facilita el mantenimiento y la evolución del sistema.

#### **4. Simetría**

Un diseño simétrico asegura un balance adecuado entre sus componentes, evitando desbalances funcionales. Como ejemplo, los patrones creacionales deben enfocarse únicamente en la creación, ya que "la falta de simetría se expresa cuando estos patrones asumen responsabilidades adicionales" (Pantaleo & Rinaudo, 2015, p. 223). Un diseño bien balanceado mejora su comprensión y calidad.

#### **5. Emergencia**

Un buen diseño debe generar propiedades emergentes que sean coherentes con las necesidades del problema. Estas propiedades no deben ser forzadas, ya que "es deseable que emerjan naturalmente a partir de la alternativa seleccionada acorde al problema a resolver" (Pantaleo & Rinaudo, 2015, p. 223). Esto refuerza la adaptabilidad y funcionalidad del diseño.

#### **6. Inversión en la cadena de dependencia**

Un diseño robusto establece que los módulos de alto nivel no dependan directamente de los de bajo nivel. Esto se logra mediante abstracciones, ya que "ambos niveles deben depender de abstracciones, lo que fomenta la independencia y reutilización del código" (Pantaleo & Rinaudo, 2015, p. 223). Así, se mejora la modularidad y se reducen las dependencias innecesarias.

#### **7. Código clausurado ante cambios**

Este criterio busca que las clases puedan extenderse sin modificar su estructura existente. Un diseño sólido asegura que "el código debe estar abierto a la extensión, pero cerrado a modificaciones" (Pantaleo & Rinaudo, 2015, p. 223). Esto evita impactos negativos en otras partes del sistema al introducir cambios.

## **8. Principio de sustitución**

Los subtipos deben ser intercambiables por sus tipos base sin alterar la funcionalidad del sistema. Este principio asegura coherencia, ya que "los subtipos deben respetar las expectativas definidas por sus tipos base" (Pantaleo & Rinaudo, 2015, p. 223). Esto mejora la cohesión y la compatibilidad en el diseño.

## **9. Segregación de interfaces**

Un diseño efectivo evita interfaces que incluyan métodos sin cohesión, ya que esto puede generar desventajas en su uso. Se establece que "las interfaces con métodos no cohesionados representan dificultades al ser utilizadas por clientes con intenciones diferentes" (Pantaleo & Rinaudo, 2015, p. 223). Una buena segregación asegura interfaces más claras y específicas.

## **10. Dependencias no cíclicas**

El diseño debe evitar ciclos de dependencia que dificulten el desarrollo y las pruebas. Se indica que "las dependencias cíclicas generan complicaciones innecesarias tanto en el desarrollo como en las pruebas del sistema" (Pantaleo & Rinaudo, 2015, p. 223). Al eliminar estas dependencias, se mejora la escalabilidad y estabilidad del sistema.

## **Pregunta 5**

### **Selección de tres tipos de prueba para el módulo del caso de estudio "Sistema para CleanCarPlus"**

Los 3 tipos de pruebas que se van utilizar para el desarrollo son:

- **Test de Usabilidad**
- **Test de Estrés**
- **Test de Seguridad**

#### **1. Test de Usabilidad**

El módulo seleccionado requiere probar la facilidad de uso de las interfaces de usuario, ya que el éxito del sistema CleanCarPlus depende de su interacción intuitiva con los administradores y empleados de la empresa. Este tipo de prueba es esencial porque permite identificar problemas en la navegabilidad, diseño y presentación de las funciones del sistema, asegurando que los usuarios puedan realizar tareas como el registro de clientes, control de servicios o generación de reportes sin complicaciones. Además, "trabajar con prototipos flexibles es crucial para ajustar rápidamente cualquier aspecto que no sea funcional o intuitivo en la interfaz" (Pantaleo & Rinaudo, 2015). En un sistema que prioriza la eficiencia operativa, es fundamental garantizar que las interfaces sean accesibles y claras, reduciendo los tiempos de capacitación y minimizando errores humanos.

#### **2. Test de Estrés**

Dado que el sistema CleanCarPlus manejará altos volúmenes de transacciones diarias, como el registro de servicios, actualizaciones de inventario y generación de facturas, un test de estrés es vital para comprobar su estabilidad bajo condiciones extremas de carga. Este tipo de prueba evalúa el rendimiento del módulo en escenarios que simulan picos

inesperados en la demanda, asegurando que el sistema no colapse. Según las características descritas, estas pruebas pueden realizarse con prototipos y luego verificarse en la versión final, lo que "permite identificar problemas tempranos de arquitectura y establecer mejoras antes de su implementación en ambientes productivos". En un negocio como CleanCarPlus, donde la rapidez y continuidad son cruciales, fallas en estos momentos críticos afectarían tanto la experiencia del cliente como la operación interna.

### **3. Test de Seguridad**

El módulo de CleanCarPlus debe garantizar que la información de los clientes, los servicios y los empleados esté protegida frente a accesos no autorizados. Un test de seguridad es esencial para validar que el sistema cumple con las normas de seguridad definidas y que se han implementado correctamente medidas como el control de acceso por roles, la encriptación de datos sensibles y la protección contra vulnerabilidades comunes. Este tipo de prueba es especialmente relevante porque "los componentes deben seguir estrictamente la política de seguridad definida desde las primeras etapas de desarrollo". La confidencialidad de la información y la protección contra brechas de seguridad son requisitos prioritarios en un sistema que gestiona datos privados, por lo que estas pruebas minimizan riesgos de filtraciones o manipulaciones indebidas.

## **Conclusiones**

En conclusión, la validación de datos en el sistema es un aspecto fundamental para garantizar la integridad de la información registrada. Este proceso previene errores operativos al exigir que campos como "Tipo de Servicio" y "Costo del Servicio" sean completados antes de continuar, mejorando significativamente la experiencia del usuario y la precisión de las operaciones.

Asimismo, se puede afirmar que el diseño centrado en la usabilidad es clave para asegurar que los usuarios interactúen de forma eficiente con la plataforma. Elementos como mensajes de error claros y valores predeterminados facilitan el entendimiento del sistema y reducen la posibilidad de cometer errores, especialmente en tareas críticas como el registro de servicios.

Además, cabe destacar que la flexibilidad del diseño, reflejada en la capacidad de incorporar solicitudes del cliente como la inclusión de la fecha de registro, mejora la utilidad y aceptación del sistema.

Por otra parte, incluir servicios complementarios en el diseño, como los "Productos y Servicios Asociados", contribuye a personalizar la oferta según las preferencias del cliente. Este enfoque no solo incrementa la percepción de valor del servicio, sino que también fomenta una mayor fidelización al permitir una experiencia más integral y personalizada.

Finalmente, se concluye que un sistema eficiente, estructurado y con validaciones adecuadas optimiza la gestión operativa de la empresa. Al automatizar tareas como la verificación de datos y proporcionar resúmenes claros, el sistema no solo ahorra tiempo, sino que también mejora la productividad y garantiza un manejo ágil y profesional en las operaciones diarias de CleanCarPlus.

## Referencias

Pantaleo, G., & Rinaudo, L. (2015). *Ingeniería de software*. Alfaomega Grupo Editor  
Argentina.