

	INFORME FINAL DEL PROYECTO DE PARRILLADAS "KANDELA"	Departamento de ciencia de Computación Carrera de Software	
		N.º Informe	ASGSW-2024
		Página:	1 de 24

A. BASE LEGAL

La auditoría se llevó a cabo en conformidad con las normas internacionales ISO/IEC/IEEE 29119-5:2016 (ISO/IEC/IEEE, 2016) e ISO 9000:2005 (ISO, 2005), las cuales establecen los principios y requisitos para la gestión y garantía de la calidad del software. Además, se consideró la norma ISO 9001 (ISO, 2015), que define los criterios para un sistema de gestión de calidad, asegurando que el servicio al cliente sea eficiente y efectivo.

B. ANTECEDENTES

- El 11 de diciembre de 2024, se llevó a cabo la reunión de apertura de la auditoría interna, cuyo propósito fue establecer las bases del proceso, definir los roles y responsabilidades tanto del equipo auditado como del auditor designado. Durante la sesión, se destacó la importancia de la normativa ISO 9000 (ISO, 2005) y ISO/IEC/IEEE 29119-5:2016 (ISO/IEC/IEEE, 2016), abordando aspectos clave para su aplicación en el contexto de la auditoría. Asimismo, se realizó una revisión detallada del Mapa de Procesos de 14 Niveles, con el objetivo de garantizar la alineación con los estándares establecidos.
- El 13 de enero de 2025, se llevó a cabo la reunión de ejecución, en la cual se presentaron las actividades del proceso de auditoría, incluyendo la revisión del backlog en Jira, la evaluación del Plan de Auditoría basado en las 53 cláusulas, la demostración del sistema en funcionamiento y la verificación de las pruebas de aceptación. Adicionalmente, se definieron las acciones correctivas y mejoras que cada equipo auditado debía implementar.
- El 21 de enero de 2025, se realizó una reunión de auditoría de seguimiento con el propósito de verificar el cumplimiento de las acciones correctivas y evaluar la resolución de las no conformidades detectadas en la auditoría previa. Durante esta sesión, se presentó la evidencia de cumplimiento correspondiente a la primera no conformidad menor, asegurando su corrección conforme a los estándares establecidos.

C. OBJETIVO

Garantizar la conformidad del proceso de pruebas de aceptación en el desarrollo del proyecto "Parrilladas Kandela" mediante una auditoría interna, con un enfoque específico en las cláusulas establecidas por la norma ISO/IEC 29119 (ISO/IEC/IEEE, 2016). Para ello, se empleará la herramienta Cucumber para la ejecución de pruebas automatizadas, lo que permitirá evaluar la calidad y eficiencia de los procesos de preparación, servicio y atención al cliente. Este análisis no solo facilitará la identificación de oportunidades de mejora, sino que también contribuirá a la mejora continua del proyecto, asegurando el cumplimiento de estándares de calidad y optimizando la experiencia gastronómica ofrecida.

	INFORME FINAL DEL PROYECTO DE PARRILLADAS "KANDELA"	Departamento de ciencia de Computación Carrera de Software	
		N.º Informe	ASGSW-2024
		Página:	2 de 24

D. ALCANCE

La auditoría abarca la evaluación de la conformidad con la norma ISO/IEC 29119 (ISO/IEC/IEEE, 2016), así como la revisión de la calidad en los procesos relacionados. Realizando un análisis de la documentación, donde se incluyen los requisitos del sistema, los casos de uso y las pruebas de aceptación aportadas por el equipo de desarrollo. Buscando verificar el acatamiento de los estándares definidos y robustecer la calidad del desarrollo en el proyecto.

E. CRONOGRAMA DE ACTIVIDADES

Fecha	Evento	Descripción breve
11 de diciembre del 2024	Reunión de apertura	Presentación de la auditoría, objetivos y alcance.
16 de diciembre del 2024	Desarrollo de estándar ISO/IEC29119-5 Fase-I	Inicio de implementación del estándar en el sistema.
18 de diciembre de 2025	Junta con el auditor	Revisión de avances y retroalimentación inicial.
08 de enero del 2025	Auditoría interna	Evaluación preliminar del cumplimiento de estándares.
12 de enero del 2025	Desarrollo de estándar ISO/IEC29119-5 Fase-II	Continuación de la implementación del estándar.
14 de enero del 2025	Junta con el auditor	Análisis de hallazgos y ajustes en el proceso.

	INFORME FINAL DEL PROYECTO DE PARRILLADAS "KANDELA"	Departamento de ciencia de Computación Carrera de Software	
		N.º Informe	ASGSW-2024
		Página:	3 de 24

17 de enero del 2025	Informe de cierre de auditoría	Presentación de resultados y conclusiones preliminares.
21 de enero del 2025	Junta con el audito	Evaluación de correcciones y seguimiento de acciones.
22 de enero del 2025	Desarrollo de estándar ISO/IEC29119-5 Fase-III	Ajustes y mejoras en la implementación del estándar.
22 de enero del 2025	Desarrollo de estándar ISO/IEC29119-5 Fase-IV	Finalización de ajustes según auditoría previa.
12 de febrero del 2025	Desarrollo de estándar ISO/IEC29119-5 Fase-V	Verificación de cumplimiento total del estándar.
19 de febrero del 2025	Entrega final	Presentación del informe final con resultados y conclusiones.

F. DESARROLLO

1. Para llevar a cabo la auditoría interna con un enfoque estructurado y alineado a las mejores prácticas, el líder del equipo auditor inició el proceso explicando a los equipos auditados los aspectos clave a abordar. Entre estos se incluyeron el objetivo, alcance y metodología de la auditoría, así como su impacto en futuras evaluaciones y mejoras del sistema.
2. En la fase inicial de la auditoría, se solicitó acceso al backlog del repositorio del proyecto, así como a los requisitos funcionales relacionados con la gestión de pedidos, asignación de mesas y coordinación entre meseros y cocina. Posteriormente, el auditor realizó un análisis exhaustivo para verificar la correcta implementación de las cláusulas establecidas por el equipo auditado, garantizando su cumplimiento con los estándares definidos.

	INFORME FINAL DEL PROYECTO DE PARRILLADAS "KANDELA"	Departamento de ciencia de Computación Carrera de Software	
		N.º Informe	ASGSW-2024
		Página:	4 de 24

3. A fin de evaluar la conformidad del sistema, se analizaron las cláusulas establecidas en la norma ISO/IEC/IEEE 29119-5 y su alineación con las pruebas presentadas. Este proceso permitió identificar tanto conformidades como no conformidades dentro del marco de auditoría, asegurando la correcta aplicación de los principios normativos en el contexto del sistema de gestión de pedidos del restaurante (ISO, 2005).
4. Como resultado de la revisión, se proporcionó una retroalimentación detallada sobre las áreas que requerían ajustes, acompañada de recomendaciones específicas para mejorar la implementación de los requisitos evaluados. En respuesta a estos hallazgos, el equipo auditado se comprometió a optimizar los informes de pruebas, refinar los casos de prueba y corregir las cláusulas que no cumplían completamente con los estándares establecidos.
5. En la siguiente fase de auditoría, se llevó a cabo una revisión secundaria del backlog del proyecto, de los requisitos funcionales y de las mejoras implementadas por el equipo auditado en respuesta a las observaciones previas. Se verificó que todas las pruebas presentadas cumplieran con las cláusulas establecidas en la norma ISO/IEC/IEEE 29119-5, garantizando su alineación con los estándares de calidad y la correcta operación del sistema (ISO, 2015).
6. Además, para fortalecer la trazabilidad del proceso de auditoría, se utilizó la herramienta Cucumber para generar un informe consolidado de todas las pruebas ejecutadas. Este enfoque permitió evaluar la efectividad de las mejoras implementadas, facilitar el seguimiento detallado del cumplimiento normativo y robustecer el control sobre la calidad del software.
7. En la fase final de la auditoría, se realizaron pruebas de aceptación con Cucumber para validar que el sistema cumpliera con los requisitos establecidos en auditorías anteriores. Para este análisis, se tomaron como referencia las cláusulas 5.1, 5.2.3, 5.3.2, 6.1, 7.4.7, 7.4.8 y 7.4.10 de la norma ISO/IEC/IEEE 29119-5 (ISO, 2005).
8. Con el objetivo de optimizar la validación del sistema, se identificaron palabras clave que facilitaron la construcción de casos de prueba, tales como: Seleccionar Mesa, Registrar Pedido, Modificar Pedido, Cancelar Pedido, Generar Orden de Cocina y Consultar Estado del Pedido. A través de este enfoque basado en palabras clave, se logró evaluar la interacción con el sistema (entrada) y la verificación de los resultados esperados (salida), asegurando una cobertura integral en la validación del software.
9. El uso combinado de Cucumber y su integración con Jira permitió ejecutar pruebas de aceptación de manera detallada y bien estructurada. Al fundamentarse en las cláusulas de la norma ISO/IEC/IEEE 29119-5 y utilizar palabras clave específicas, se estableció un marco de pruebas sólido que garantizó el cumplimiento de los

	INFORME FINAL DEL PROYECTO DE PARRILLADAS "KANDELA"	Departamento de ciencia de Computación Carrera de Software	
		N.º Informe	ASGSW-2024
		Página:	5 de 24

requisitos del usuario y el correcto funcionamiento del sistema de gestión de pedidos de "Parrilladas Kandela" (ISO/IEC/IEEE, 2016).

G. CONCLUSIONES

1. A través de la auditoría interna del sistema de gestión de pedidos de Parrilladas Kandela, se pudo analizar en profundidad su conformidad con las normas ISO/IEC/IEEE 29119-5 e ISO 9000 (ISO/IEC/IEEE, 2016; ISO, 2015), lo que permitió garantizar la calidad y eficiencia en sus procesos. Gracias a la implementación de pruebas automatizadas con Cucumber y su integración con Jira, fue posible validar de manera estructurada los requisitos del sistema. Adicionalmente, se identificaron y corrigieron no conformidades menores en ciclos de mejora continua, lo que reforzó significativamente el control de calidad.
2. La aplicación de pruebas basadas en palabras clave resultó ser una estrategia eficaz para optimizar la validación de los procesos críticos. Esto no solo aseguró una cobertura más amplia, sino que también minimizó de manera notable la posibilidad de errores en la gestión de pedidos. Además, este enfoque favoreció una mejora significativa en la experiencia del usuario final y en la eficiencia operativa del restaurante. Como consecuencia, se estableció un marco sólido que servirá con referencia para futuras auditorías y la implementación de mejores prácticas en la ejecución de pruebas.
3. La auditoría puso en evidencia la importancia de mantener un monitoreo constante y contar con una documentación precisa de los resultados logrados para salvaguardar la estabilidad del software a lo largo del tiempo. La combinación de metodologías ágiles con herramientas de automatización consolidó una estrategia eficaz para asegurar la calidad del sistema. Como resultado, se establecieron bases sólidas para la implementación de futuras mejoras e innovaciones, favoreciendo la evolución digital del restaurante y garantizando un desarrollo sostenible que pueda adaptarse a las necesidades cambiantes del mercado.

Cláusula	Estado	Observación
5.1	NO OK	Se identificó que esta cláusula requiere ajustes adicionales en futuras pruebas para su correcta implementación.
5.2.3	NO OK	Se identificó que esta cláusula requiere ajustes adicionales en futuras pruebas para su correcta implementación.
5.3.2	NO OK	Se identificó que esta cláusula requiere ajustes adicionales en futuras pruebas para su correcta implementación.
6.1	OK	Se ejecutó correctamente esta cláusula, garantizando que las pruebas basadas en palabras clave se integren adecuadamente con el sistema y cumplan con los requisitos establecidos.

7.4.7	OK	Se ejecutó correctamente esta cláusula, asegurando que el motor de ejecución facilite la automatización de las pruebas y optimice la velocidad de ejecución.
7.4.8	OK	Se ejecutó correctamente esta cláusula, garantizando que la biblioteca de palabras clave permite la organización y reutilización eficiente de pruebas automatizadas.
7.4.10	OK	Se ejecutó correctamente esta cláusula, asegurando que el repositorio de scripts facilite el control de versiones y el acceso eficiente a los scripts de prueba.


H. RECOMENDACIONES

1. Para mejorar la eficiencia del sistema, se recomienda ampliar la cobertura de las pruebas automatizadas, incluyendo escenarios más complejos y excepcionales en la gestión de pedidos.
2. Realizar capacitaciones en herramientas de automatización como Cucumber y Jira, para la optimización de tiempos de ejecución de pruebas y mejorar la detección temprana de errores.
3. Integrar nuevas tecnologías de IA para el análisis de datos, permitiendo una mejora efectiva de los pedidos y optimizando la experiencia del cliente.

I. APROBACIÓN

Rubro	Nombre Apellido	Unidad/Cargo	Firma
Elaborado por	Barriga Mateo	Líder de Equipo	
	Bazurto Christopher	Desarrollador Frontend	
	Macas Karol	Desarrollador Frontend	
	Otuna Shirley	Desarrollador Backtend	

Sangolquí, 13 de febrero de 2024

	INFORME FINAL DEL PROYECTO DE PARRILLADAS "KANDELA"	Departamento de ciencia de Computación Carrera de Software	
		N.º Informe	ASGSW-2024
		Página:	7 de 24

J. REFERENCIAS

ISO/IEC/IEEE. (2016). ISO/IEC/IEEE 29119-5:2016 Software and systems engineering — Software testing — Part 5: Keyword-Driven Testing. International Organization for Standardization. Recuperado de <https://www.iso.org/standard/62821.html>

ISO. (2005). ISO 9000:2005 Quality management systems — Fundamentals and vocabulary. International Organization for Standardization. Recuperado de <https://www.iso.org/standard/42180.html>

ISO. (2015). ISO 9001:2015 Quality management systems — Requirements. International Organization for Standardization. Recuperado de <https://www.iso.org/standard/62085.html>

K. ANEXOS

Requisito 1:

```
1 Feature: Gestión de pedidos en el restaurante parrilladas Kande
2
3 Como mesero
4 Quiero crear y gestionar pedidos seleccionando platos y bebidas del menú digital
5 Para reducir errores y enviar las órdenes a la cocina en tiempo real
6
7 Scenario: Crear un pedido exitosamente
8 Given el mesero está en la pantalla del menú digital
9 When selecciona "Plato A" y "Bebida B"
10 And confirma el pedido
11 Then el sistema debe mostrar el pedido con "Plato A" y "Bebida B" en la lista de pedidos activos
12 And enviar el pedido a la pantalla de la cocina
13
14 Scenario: Error al crear un pedido sin seleccionar platos
15 Given el mesero está en la pantalla del menú digital
16 When intenta confirmar el pedido sin seleccionar platos ni bebidas
17 Then el sistema debe mostrar un mensaje de error "Debe seleccionar al menos un plato o bebida"
```

Resultado:

Como resultado la pruebas sin error paso y las prueba con error presentó dificultades.

▼ Feature: Gestión de pedidos en el restaurante parrilladas Kande 1ms 1 1

Como mesero
Quiero crear y gestionar pedidos seleccionando platos y bebidas del menú digital
Para reducir errores y enviar las órdenes a la cocina en tiempo real

▼ Scenario: 1ms 5
Crear un pedido exitosamente

✓ Given el mesero está en la pantalla del menú digital 1ms

✓ When selecciona "Plato A" y "Bebida B" < 1ms

✓ And confirma el pedido < 1ms

✓ Then el sistema debe mostrar el pedido con "Plato A" y "Bebida B" en la lista de pedidos activos < 1ms

✓ And enviar el pedido a la pantalla de la cocina < 1ms

▼ Scenario: < 1ms 1 2
Error al crear un pedido sin seleccionar platos

✓ Given el mesero está en la pantalla del menú digital < 1ms

⚡ When intenta confirmar el pedido sin seleccionar platos ni bebidas < 1ms

Then(/^intenta confirmar el pedido sin seleccionar platos ni be
// Write code here that turns the phrase above into concrete
callback(null, 'pending');
});

⚡ Then el sistema debe mostrar un mensaje de error "Debe seleccionar al
menos un plato o bebida" < 1ms

Then(/^el sistema debe mostrar un mensaje de error "[^"]*"\$/
// Write code here that turns the phrase above into concrete
callback(null, 'pending');
});

Requisito 2:

Feature: Envío del pedido a la cocina

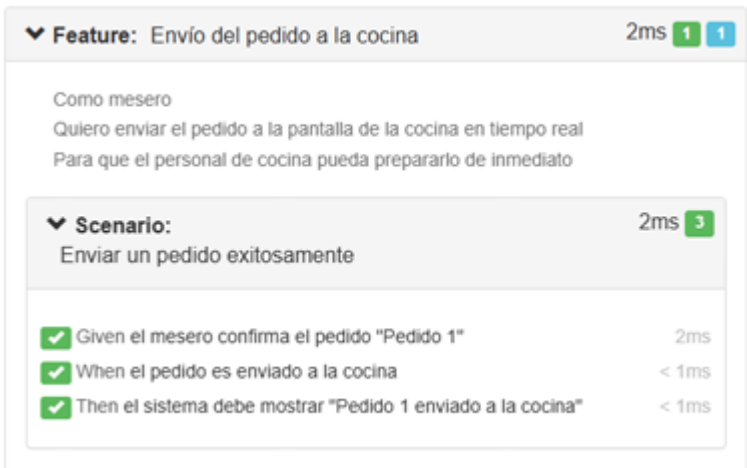
Como mesero
Quiero enviar el pedido a la pantalla de la cocina en tiempo real
Para que el personal de cocina pueda prepararlo de inmediato

Scenario: Enviar un pedido exitosamente
Given el mesero confirma el pedido "Pedido 1"
When el pedido es enviado a la cocina
Then el sistema debe mostrar "Pedido 1 enviado a la cocina"

Scenario: Error al intentar enviar un pedido vacío
Given el mesero confirma un pedido vacío
When intenta enviar el pedido a la cocina
Then el sistema debe mostrar un mensaje de error "El pedido está vacío"

Resultado de prueba exitosa:

Dentro de las pruebas realizadas pasaron todas.



Resultado de prueba fallida:

Dentro de las pruebas con error se presentó dificultad.

▼ **Scenario:**

Error al intentar enviar un pedido vacío

< 1ms

12

✓ Given el mesero confirma un pedido vacío

< 1ms

⚡ When intenta enviar el pedido a la cocina

< 1ms

Then(/^intenta enviar el pedido a la cocina\$/, function(callback) {
 // Write code here that turns the phrase above into concrete
 callback(null, 'pending');
});

⚡ Then el sistema debe mostrar un mensaje de error "El pedido está vacío"

< 1ms


Then(/^el sistema debe mostrar un mensaje de error "([^\"]*)"\$/,
 // Write code here that turns the phrase above into concrete
 callback(null, 'pending');
});

Requisito 3:

```
1  Feature: Modificar o cancelar pedidos
2
3  Como mesero
4  Quiero modificar o cancelar pedidos antes de ser preparados
5  Para corregir errores o cambios solicitados por los clientes
6
7  Scenario: Modificar un pedido exitosamente
8    Given el pedido "Pedido 1" está en estado "Pendiente"
9    When el mesero modifica el pedido añadiendo "Plato C"
10   Then el sistema debe mostrar el pedido actualizado con "Plato C"
11
12  Scenario: Error al intentar modificar un pedido ya preparado
13    Given el pedido "Pedido 2" está en estado "Listo"
14    When el mesero intenta modificar el pedido añadiendo "Plato C"
15    Then el sistema debe mostrar un mensaje de error "No se puede modificar un pedido ya preparado"
```

Resultado de prueba exitosa:

Dentro de las pruebas realizadas pasaron todas.

	INFORME FINAL DEL PROYECTO DE PARRILLADAS "KANDELA"	Departamento de ciencia de Computación Carrera de Software	
		N.º Informe	ASGSW-2024
		Página:	11 de 24

▼ Feature: Modificar o cancelar pedidos1ms11

Como mesero

Quiero modificar o cancelar pedidos antes de ser preparados

Para corregir errores o cambios solicitados por los clientes

▼ Scenario:1ms3

Modificar un pedido exitosamente

✓ Given el pedido "Pedido 1" está en estado "Pendiente"1ms

✓ When el mesero modifica el pedido añadiendo "Plato C"< 1ms

✓ Then el sistema debe mostrar el pedido actualizado con "Plato C"< 1ms

Resultado de prueba fallida:

Dentro de las pruebas con error se presentó dificultad.

▼ Scenario:< 1ms111

Error al intentar modificar un pedido ya preparado

✓ Given el pedido "Pedido 2" está en estado "Listo"< 1ms

⚡ When el mesero intenta modificar el pedido añadiendo "Plato C"< 1ms

Then(/^el mesero intenta modificar el pedido añadiendo "([^\"]*)"// Write code here that turns the phrase above into concrete callback(null, 'pending');});

Then el sistema debe mostrar un mensaje de error "No se puede modificar un pedido ya preparado"< 1ms

Requisito 4:

```
1 Feature: Mostrar el estado de los platos
2
3 Como mesero
4 Quiero ver el estado de cada plato
5 Para saber cuál está listo y cuál sigue en preparación
6
7 Scenario: Ver el estado de los platos exitosamente
8   Given el plato "Plato A" está en estado "En preparación"
9   When el mesero consulta el estado de "Plato A"
10  Then el sistema debe mostrar "Plato A: En preparación"
11
12 Scenario: Error al consultar un plato inexistente
13   Given no está registrado el plato "Plato X"
14   When el mesero consulta el estado de "Plato X"
15  Then el sistema debe mostrar un mensaje de error "El plato no existe"
```

Resultado de prueba exitosa:

Dentro de las pruebas realizadas pasaron todas.

▼ Feature: Mostrar el estado de los platos1ms2

Como mesero
Quiero ver el estado de cada plato
Para saber cuál está listo y cuál sigue en preparación

▼ Scenario:< 1ms3

Ver el estado de los platos exitosamente


✓ Given el plato "Plato A" está en estado "En preparación">< 1ms

✓ When el mesero consulta el estado de "Plato A">< 1ms

✓ Then el sistema debe mostrar "Plato A: En preparación">< 1ms

Resultado de prueba fallida:

Dentro de las pruebas con error el error que se puso dentro de la prueba si paso.

	INFORME FINAL DEL PROYECTO DE PARRILLADAS "KANDELA"	Departamento de ciencia de Computación Carrera de Software	
		N.º Informe	ASGSW-2024
		Página:	13 de 24

▼ **Scenario:**

Error al consultar un plato inexistente

< 1ms **3**

✓ Given no está registrado el plato "Plato X"

< 1ms

✓ When el mesero consulta el estado de "Plato X"

< 1ms

✓ Then el sistema debe mostrar un mensaje de error "El plato no existe"

< 1ms

Requisito 5:

```
1 Feature: Notificación de pedidos listos
2
3 Como mesero
4 Quiero recibir notificaciones cuando un pedido esté listo
5 Para poder servirlo a los clientes rápidamente
6
7 Scenario: Notificar un pedido listo exitosamente
8 Given el pedido "Pedido 1" está en estado "En preparación"
9 When el estado de "Pedido 1" cambia a "Listo"
10 Then el sistema debe notificar al mesero con el mensaje "El pedido Pedido 1 está listo"
11
12 Scenario: Error al intentar notificar un pedido no existente
13 Given no existe el pedido "Pedido 99"
14 When el mesero intenta cambiar el estado de "Pedido 99" a "Listo"
15 Then el sistema debe mostrar un mensaje de error "El pedido no existe"
```

Resultado de prueba exitosa:

Dentro de las pruebas realizadas pasaron todas.

▼ Feature: Notificación de pedidos listos2ms11

Como mesero
Quiero recibir notificaciones cuando un pedido esté listo
Para poder servirlo a los clientes rápidamente

▼ Scenario:2ms3

Notificar un pedido listo exitosamente

✓

Given el pedido "Pedido 1" está en estado "En preparación"

1ms

✓

When el estado de "Pedido 1" cambia a "Listo"

< 1ms

✓

Then el sistema debe notificar al mesero con el mensaje "El pedido Pedido 1 está listo"

< 1ms

Resultado de prueba fallida:
Dentro de las pruebas con error se presentó dificultad.

▼ Scenario:< 1ms111

Error al intentar notificar un pedido no existente

✓

Given no existe el pedido "Pedido 99"

< 1ms

⚡

When el mesero intenta cambiar el estado de "Pedido 99" a "Listo"

< 1ms

Then(/^el mesero intenta cambiar el estado de "([^\"]*)" a "([^\"]" // Write code here that turns the phrase above into concrete callback(null, 'pending'); });

✖

Then el sistema debe mostrar un mensaje de error "El pedido no existe"

< 1ms

Requisito 6:

```
1  Feature: Cambiar el estado de los platos
2
3  Como mesero
4  Quiero cambiar el estado de los platos individualmente
5  Para reflejar su progreso en la preparación
6
7  Scenario: Cambiar el estado de un plato exitosamente
8    Given el plato "Plato A" está en estado "En preparación"
9    When el mesero cambia el estado de "Plato A" a "Listo"
10   Then el sistema debe mostrar "Plato A: Listo"
11
12 Scenario: Error al cambiar el estado de un plato inexistente
13   Given el plato "Plato X" no está registrado
14   When el mesero intenta cambiar el estado de "Plato X" a "Listo"
15   Then el sistema debe mostrar un mensaje de error "El plato no existe"
```

Resultado de prueba exitosa:

Dentro de las pruebas realizadas pasaron todas.

▼ Feature: Cambiar el estado de los platos2ms11

Como mesero
Quiero cambiar el estado de los platos individualmente
Para reflejar su progreso en la preparación

▼ Scenario:1ms3

Cambiar el estado de un plato exitosamente

✓ Given el plato "Plato A" está en estado "En preparación"1ms

✓ When el mesero cambia el estado de "Plato A" a "Listo"< 1ms

✓ Then el sistema debe mostrar "Plato A: Listo"< 1ms

Resultado de prueba fallida:

Dentro de las pruebas con error se presentó dificultad.

▼ **Scenario:**

< 1ms 1 1 1

Error al cambiar el estado de un plato inexistente

✓

Given el plato "Plato X" no está registrado

< 1ms

⚡

When el mesero intenta cambiar el estado de "Plato X" a "Listo"

< 1ms

```
Then(/^el mesero intenta cambiar el estado de "([^\"]*)" a "([^\"]*"
// Write code here that turns the phrase above into concrete
callback(null, 'pending');
});
```

⏏

Then el sistema debe mostrar un mensaje de error "El plato no existe"

< 1ms

Requisito 7:

```
1 Feature: Mostrar pedidos por cobrar
2
3 Como mesero
4 Quiero ver una lista de pedidos por cobrar
5 Para identificar rápidamente los pedidos pendientes de pago
6
7 Scenario: Ver pedidos por cobrar exitosamente
8 Given existen los pedidos "Pedido 1" y "Pedido 2" con estado "Por cobrar"
9 When el mesero solicita la lista de pedidos por cobrar
10 Then el sistema debe mostrar "Pedido 1, Pedido 2"
11
12 Scenario: Error al solicitar una lista vacía de pedidos por cobrar
13 Given no hay pedidos con estado "Por cobrar"
14 When el mesero solicita la lista de pedidos por cobrar
15 Then el sistema debe mostrar un mensaje de error "No hay pedidos por cobrar"
```

Resultado de prueba exitosa:

Dentro de las pruebas realizadas pasaron todas.

	INFORME FINAL DEL PROYECTO DE PARRILLADAS "KANDELA"	Departamento de ciencia de Computación Carrera de Software	
		N.º Informe	ASGSW-2024
		Página:	17 de 24

▼ **Feature:** Mostrar pedidos por cobrar2ms2

Como mesero
Quiero ver una lista de pedidos por cobrar
Para identificar rápidamente los pedidos pendientes de pago

▼ **Scenario:**1ms3

Ver pedidos por cobrar exitosamente

✓ Given existen los pedidos "Pedido 1" y "Pedido 2" con estado "Por cobrar"1ms

✓ When el mesero solicita la lista de pedidos por cobrar< 1ms

✓ Then el sistema debe mostrar "Pedido 1, Pedido 2"< 1ms

Resultado de prueba fallida:

Dentro de las pruebas con error el error que se puso dentro de la prueba si paso.

▼ **Scenario:**< 1ms3

Error al solicitar una lista vacía de pedidos por cobrar

✓ Given no hay pedidos con estado "Por cobrar"< 1ms

✓ When el mesero solicita la lista de pedidos por cobrar< 1ms

✓ Then el sistema debe mostrar un mensaje de error "No hay pedidos por cobrar"< 1ms

Requisito 8:

```
1 Feature: Registro del estado de pago de los pedidos
2
3 Como mesero
4 Quiero registrar el estado de pago de cada pedido
5 Para mantener un control claro de los pedidos cobrados y pendientes
6
7 Scenario: Registrar un pedido como pagado exitosamente
8   Given el pedido "Pedido 1" está en estado "Pendiente de pago"
9   When el mesero marca el pedido como "Pagado"
10  Then el sistema debe mostrar "Pedido 1: Pagado"
11
12 Scenario: Error al intentar registrar un pedido como pagado que ya está pagado
13   Given el pedido "Pedido 2" está en estado "Pagado"
14   When el mesero intenta marcar el pedido como "Pagado" nuevamente
15   Then el sistema debe mostrar un mensaje de error "El pedido ya está registrado como pagado"
```

Resultado de prueba exitosa:

Dentro de las pruebas realizadas pasaron todas.

▼ Feature: Registro del estado de pago de los pedidos2ms11

Como mesero
Quiero registrar el estado de pago de cada pedido
Para mantener un control claro de los pedidos cobrados y pendientes

▼ Scenario:Registrar un pedido como pagado exitosamente2ms3


✓ Given el pedido "Pedido 1" está en estado "Pendiente de pago"2ms

✓ When el mesero marca el pedido como "Pagado">< 1ms

✓ Then el sistema debe mostrar "Pedido 1: Pagado">< 1ms

Resultado de prueba fallida:

Dentro de las pruebas con error se presentó dificultad.

	INFORME FINAL DEL PROYECTO DE PARRILLADAS "KANDELA"	Departamento de ciencia de Computación Carrera de Software	
		N.º Informe	ASGSW-2024
		Página:	19 de 24

▼ **Scenario:**

< 1ms 1 1 1

Error al intentar registrar un pedido como pagado que ...

✓ Given el pedido "Pedido 2" está en estado "Pagado"

< 1ms

⚡ When el mesero intenta marcar el pedido como "Pagado" nuevamente

< 1ms

Then(/^el mesero intenta marcar el pedido como "([^"]*)" nuevam

// Write code here that turns the phrase above into concrete

callback(null, 'pending');

});

⚠ Then el sistema debe mostrar un mensaje de error "El pedido ya está

registrado como pagado"


< 1ms

Requisito 9:

```
1 Feature: Agregar nuevos platos al menú
2
3 Como administrador
4 Quiero agregar nuevos platos al menú
5 Para mantener el menú actualizado y variado
6
7 Scenario: Agregar un nuevo plato exitosamente
8 Given el menú actual no contiene "Plato Nuevo"
9 When el administrador agrega "Plato Nuevo" al menú
10 Then el sistema debe mostrar "Plato Nuevo añadido al menú"
11
12 Scenario: Error al intentar agregar un plato ya existente
13 Given el menú actual contiene "Plato Existente"
14 When el administrador intenta agregar "Plato Existente" al menú
15 Then el sistema debe mostrar un mensaje de error "El plato ya existe en el menú"
```

Resultado de prueba exitosa:

Dentro de las pruebas realizadas pasaron todas.

	INFORME FINAL DEL PROYECTO DE PARRILLADAS "KANDELA"	Departamento de ciencia de Computación Carrera de Software	
		N.º Informe	ASGSW-2024
		Página:	20 de 24

▼ **Feature:** Agregar nuevos platos al menú 5ms 1 1

Como administrador
Quiero agregar nuevos platos al menú
Para mantener el menú actualizado y variado

▼ **Scenario:** 5ms 3

Agregar un nuevo plato exitosamente

✓

Given el menú actual no contiene "Plato Nuevo" 4ms

✓

When el administrador agrega "Plato Nuevo" al menú < 1ms

✓

Then el sistema debe mostrar "Plato Nuevo añadido al menú" < 1ms

Resultado de prueba fallida:

Dentro de las pruebas con error se presentó dificultad.

▼ **Scenario:** < 1ms 1 1 1

Error al intentar agregar un plato ya existente

✓

Given el menú actual contiene "Plato Existente" < 1ms

⚡

When el administrador intenta agregar "Plato Existente" al menú < 1ms

Then(/^el administrador intenta agregar "([^\"]*)" al menú\$/, function() {
// Write code here that turns the phrase above into concrete actions
callback(null, 'pending');
});

✖

Then el sistema debe mostrar un mensaje de error "El plato ya existe en el menú" < 1ms

Requisito 10:


```
1 Feature: Desactivar platos del menú
2
3 Como administrador
4 Quiero desactivar platos existentes
5 Para ocultarlos temporalmente sin eliminarlos del menú
6
7 Scenario: Desactivar un plato exitosamente
8 Given el menú contiene "Plato A"
9 When el administrador desactiva "Plato A"
10 Then el sistema debe mostrar "Plato A desactivado"
11
12 Scenario: Error al intentar desactivar un plato ya desactivado
13 Given el plato "Plato B" ya está desactivado
14 When el administrador intenta desactivar "Plato B" nuevamente
15 Then el sistema debe mostrar un mensaje de error "El plato ya está desactivado"
```

Resultado de prueba exitosa:

Dentro de las pruebas realizadas pasaron todas.

▼ Feature: Desactivar platos del menú2ms11

Como administrador
Quiero desactivar platos existentes
Para ocultarlos temporalmente sin eliminarlos del menú

▼ Scenario:Desactivar un plato exitosamente2ms3

✓ Given el menú contiene "Plato A"1ms

✓ When el administrador desactiva "Plato A"< 1ms

✓ Then el sistema debe mostrar "Plato A desactivado"< 1ms

Resultado de prueba fallida:

Dentro de las pruebas con error se presentó dificultad.

▼ **Scenario:**

< 1ms 1 2

Error al intentar desactivar un plato ya desactivado

✓ Given el plato "Plato B" ya está desactivado

< 1ms

⚡ When el administrador intenta desactivar "Plato B" nuevamente

< 1ms

Then(/^el administrador intenta desactivar "([^\"]*)" nuevamente

// Write code here that turns the phrase above into concrete

callback(null, 'pending');

});

⚡ Then el sistema debe mostrar un mensaje de error "El plato ya está desactivado"

< 1ms

Then(/^el sistema debe mostrar un mensaje de error "([^\"]*)"\$/

// Write code here that turns the phrase above into concrete

callback(null, 'pending');

});

Requisito 11:

```
1  Feature: Historial de cambios en el menú
2
3  Como administrador
4  Quiero guardar un historial de cambios en el menú
5  Para mantener un registro de todas las modificaciones
6
7  Scenario: Registrar un cambio en el menú exitosamente
8  Given el historial de cambios está vacío
9  When el administrador agrega "Plato A" al menú
10 Then el sistema debe registrar "Plato A añadido al menú" en el historial
11
12 Scenario: Error al intentar acceder al historial vacío
13 Given el historial de cambios está vacío
14 When el administrador consulta el historial
15 Then el sistema debe mostrar "El historial está vacío"
```

Resultado de prueba exitosa:

Dentro de las pruebas realizadas pasaron todas.

▼ **Feature:** Historial de cambios en el menú 3ms

Como administrador
Quiero guardar un historial de cambios en el menú
Para mantener un registro de todas las modificaciones

▼ **Scenario:** 2ms 3
Registrar un cambio en el menú exitosamente

- ✓ Given el historial de cambios está vacío 2ms
- ✓ When el administrador agrega "Plato A" al menú < 1ms
- ✓ Then el sistema debe registrar "Plato A añadido al menú" en el historial < 1ms

Resultado de prueba fallida:

Dentro de las pruebas con error se presentó dificultad.

▼ **Scenario:** < 1ms 3
Error al intentar acceder al historial vacío

- ✓ Given el historial de cambios está vacío < 1ms
- ✓ When el administrador consulta el historial < 1ms
- ✓ Then el sistema debe mostrar "El historial está vacío" < 1ms

	INFORME DE HECHOS RESPECTO A LA AUDITORÍA INTERNA	Departamento de Ciencias de la Computación	
		N.º Informe	ASGSW-2024
		Página:	24 de 24

