

Validación de Coherencia de Modelos UML

Proyecto: Sistema de Pedidos de Comida

Integrantes: Juan José Balanta Agrono, Johan Andrés Carabalí Carabalí

Institución: Institución Universitaria Antonio José Camacho

Programa: Ingeniería de Software

Fecha: Noviembre de 2025

1. Objetivo de la Validación

El objetivo de esta validación es garantizar la coherencia técnica y conceptual entre los modelos UML desarrollados en el proyecto: casos de uso, clases y arquitectura. Se busca asegurar que cada caso de uso esté correctamente representado en las clases y reflejado en la arquitectura del sistema.

2. Tabla de Validación de Coherencia

Caso de uso	Diagrama de clases	Arquitectura / Componentes	¿Coherente?	Observaciones
Registrar usuario	Clase Usuario implementa método registrar()	Componente 'Gestión de usuarios'	<input checked="" type="checkbox"/> Sí	Coincide entre modelos
Iniciar sesión	Clase Usuario con método autenticar()	Componente 'Autenticación'	<input checked="" type="checkbox"/> Sí	Nombres consistentes
Ver menú	Clase Producto y Menú	Componente 'Interfaz cliente'	<input checked="" type="checkbox"/> Sí	Relación correcta con clases de presentación
Realizar pedido	Clase Pedido con relación a Usuario y Producto	Componente 'Gestión de pedidos'	<input checked="" type="checkbox"/> Sí	Flujo coherente
Pagar pedido	Clase Pago aún no	Componente 'Módulo de'	<input type="checkbox"/> No	Debe agregarse clase Pago

	implementada	pagos'		
Generar reporte	Clase Reporte no definida	Componente 'Administración / Reportes'	✗ No	Falta incluir clase y flujo correspondiente

3. Observaciones Generales

Durante la revisión se identificaron incoherencias menores entre los diagramas de clases y la arquitectura, especialmente en la ausencia de la clase 'Pago' y del componente correspondiente en el módulo de pagos. También se detectó la necesidad de incluir una clase 'Reporte' que respalde el caso de uso 'Generar reporte'. Las demás funcionalidades mantienen consistencia en nombres, relaciones y estructuras.

4. Reflexión Final

La incoherencia más significativa encontrada fue la falta de correspondencia entre el caso de uso 'Pagar pedido' y la ausencia de la clase 'Pago' en el diagrama de clases y la arquitectura. Esto se resolvió diseñando la clase 'Pago' y agregando el componente 'Módulo de pagos' en la arquitectura, asegurando así la coherencia completa entre los modelos UML del sistema.