Diagramas Modelo de Dominio – Sistema de Gestión de Pedidos para Restaurantes

Formación: Análisis y Desarrollo de Software

Nombre aprendiz:

Fecha: Septiembre 2025

Table of Contents

[Introducción 3](#_Toc209612194)

[1. Tabla de Clases 4](#_Toc209612195)

[2. Diagrama de Clases 5](#_Toc209612196)

[3. Lectura de Relaciones 5](#_Toc209612197)

[4. Esquema Narrativo con Ejemplos 6](#_Toc209612198)

5. Diagrama de Paquetes …………………………………………………………………………………………………6

6. Relaciones entre paquetes …………………………………………………………………………………………...7

[Conclusiones 8](#_Toc209612199)

[Referencias 9](#_Toc209612200)

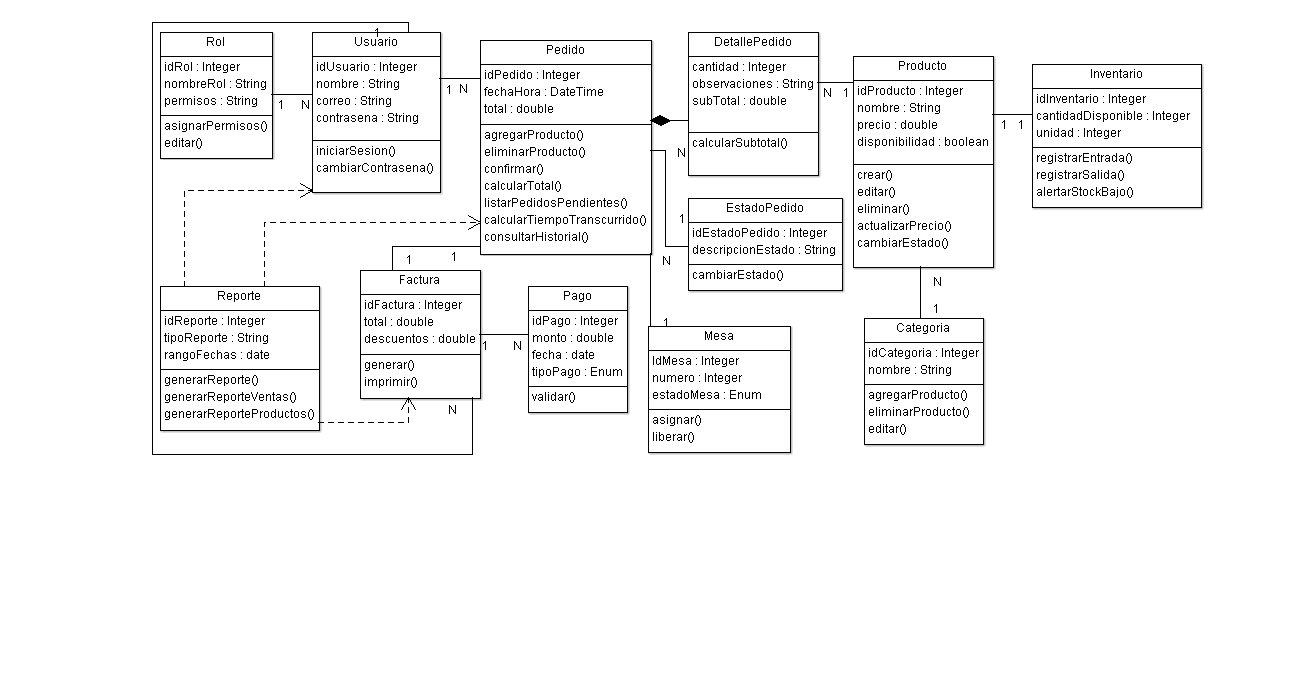
# Introducción

El diagrama de clases y el diagrama de paquetes son artefactos fundamentales dentro del análisis y diseño de sistemas de software, ya que permite identificar las clases principales del dominio, sus atributos, métodos y las relaciones entre ellas. En el caso del Sistema de Gestión de Pedidos para Restaurantes, estos diagramas organizan y modelan los elementos esenciales como pedidos, facturas, productos, usuarios, roles, inventario y reportes. El objetivo de este informe es presentar el modelo de dominio del sistema, mostrar las relaciones entre sus componentes, y evidenciar cómo dichas relaciones permiten cumplir los requerimientos funcionales definidos.

# 1. Tabla de Clases

|  |  |  |
| --- | --- | --- |
| Clase | Atributos | Métodos |
| Usuario | idUsuario, nombre, correo, contraseña | iniciarSesion(), cambiarContraseña() |
| Rol | idRol, nombreRol, permisos | asignarPermisos(), editar() |
| Mesa | idMesa, numero, estadoMesa | asignar(), liberar() |
| Pedido | idPedido, fechaHora, total | agregarProducto(), eliminarProducto(), confirmar(), calcularTotal(), listarPedidosPendientes(), calcularTiempoTranscurrido(), consultarHistorial() |
| EstadoPedido | idEstadoPedido, descripcionEstado | cambiarEstado() |
| DetallePedido | cantidad, observaciones, subTotal | calcularSubtotal() |
| Producto | idProducto, nombre, precio, disponibilidad | crear(), editar(), eliminar(), actualizarPrecio(), cambiarEstado() |
| Categoria | idCategoria, nombre | agregarProducto(), eliminarProducto(), editar() |
| Inventario | idInventario, cantidadDisponible, unidad | registrarEntrada(), registrarSalida(), alertarStockBajo() |
| Factura | idFactura, total, descuentos | generar(), imprimir() |
| Pago | idPago, monto, fecha, tipoPago | validar() |
| Reporte | idReporte, tipoReporte, rangoFechas | generarReporte(), generarReporteVentas(), generarReporteProductos() |

# 2. Diagrama de Clases



# 3. Lectura de Relaciones

1. Un Rol puede estar asignado a muchos Usuarios (1..\*), pero cada Usuario tiene exactamente un Rol.

2. Un Usuario puede registrar muchos Pedidos (1..\*), pero cada Pedido está asociado a un único Usuario (Mesero).

3. Una Mesa puede tener muchos Pedidos, pero cada Pedido pertenece a una sola Mesa.

4. Un Pedido está compuesto por uno o más Detalles de Pedido, y cada DetallePedido pertenece exactamente a un Pedido (composición).

5. Un DetallePedido está asociado a un único Producto, mientras que un Producto puede aparecer en muchos DetallesPedido.

6. Cada Producto pertenece a una Categoría, y una Categoría puede agrupar múltiples Productos.

7. Cada Producto puede estar relacionado con un Inventario para controlar la disponibilidad.

8. Un Pedido genera una única Factura, y cada Factura corresponde a un Pedido.

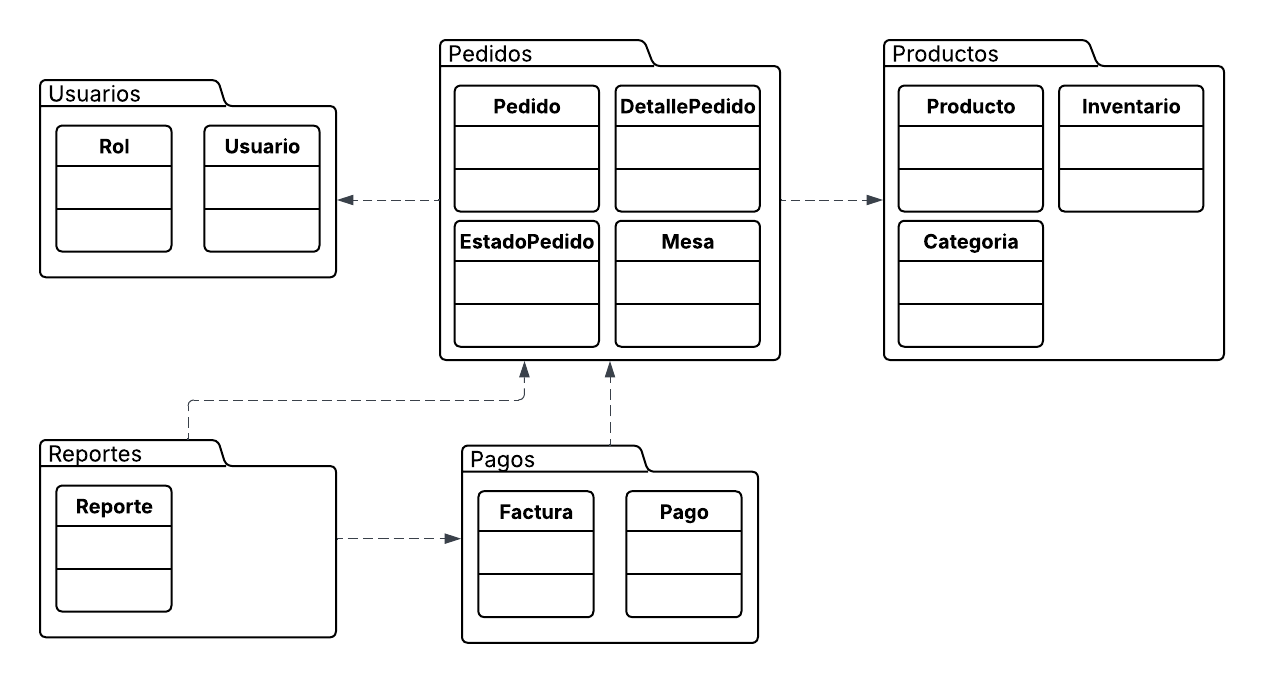
9. Una Factura puede dividirse en uno o varios Pagos, y cada Pago pertenece a una única Factura.

10. Un Reporte depende de los datos de Pedido y Factura, y es solicitado por un Usuario administrador.

# 4. Esquema Narrativo con Ejemplos

En un día de operación del restaurante, un mesero (Usuario con rol Mesero) asigna la Mesa 4 a un grupo de clientes. El mesero registra un nuevo Pedido para esa mesa, seleccionando los productos desde el menú digital. Cada producto seleccionado se convierte en un DetallePedido, donde se puede incluir observaciones como 'sin sal' o 'extra queso'. El Pedido queda registrado con fecha y hora, y se envía automáticamente a la cocina. El cocinero visualiza los pedidos pendientes en su pantalla, observa el tiempo transcurrido y cambia el EstadoPedido a 'en preparación' o 'listo'. Una vez que los clientes terminan, el mesero genera la Factura correspondiente al Pedido. La Factura puede dividirse en varios Pagos: por ejemplo, un cliente paga en efectivo y otro con tarjeta. Finalmente, el administrador revisa el Reporte de Ventas del día y los productos más vendidos, usando los datos de Factura y Pedido.

**5. Diagrama de Paquetes**

****

**6. Relaciones entre paquetes**

* Pedidos depende de usuarios (porque un pedido lo hace un usuario).
* Pedidos depende de productos (porque un pedido contiene productos).
* Pagos depende de pedidos (porque las facturas se generan a partir de pedidos).
* Reportes depende de pagos y pedidos (porque los reportes se generan con esa información).

# Conclusiones

Los diagramas de clases y de paquetes permiten estructurar los elementos clave del sistema, mostrando las clases, sus atributos y métodos, y las relaciones entre ellas. Gracias a este modelo, es posible comprender cómo los objetos del sistema colaboran para cumplir con los requerimientos funcionales definidos. Además, la correcta identificación de asociaciones, multiplicidades y dependencias asegura trazabilidad con los casos de uso y soporte para el diseño detallado posterior.

# Referencias

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). El lenguaje unificado de modelado (UML). Addison Wesley.  
Sommerville, I. (2011). Ingeniería de software (9a ed.). Pearson Educación.  
Pressman, R. S. (2010). Ingeniería del software: Un enfoque práctico (7a ed.). McGraw-Hill.  
Object Management Group (OMG). (2017). Unified Modeling Language (UML) Specification. https://www.omg.org/spec/UML/