

SISTEMA DE CAFETERÍA UNIVERSITARIA

Estudiante: Mateo Bolivar Arroyave
Fecha: 09/09/2025
Universidad: ITM

TABLA DE CONTENIDO

SISTEMA DE CAFETERÍA UNIVERSITARIA	1
TABLA DE CONTENIDO	1
1. CASO DE NEGOCIO	2
1.1 Problema Identificado	2
1.2 Usuarios del Sistema	2
1.3 Valor Esperado	3
2. HISTORIAS DE USUARIO	3
HU-01: Gestión de Productos	3
HU-02: Realizar Pedido	3
HU-03: Procesar Pago	4
HU-04: Gestionar Cola de Pedidos	4
HU-05: Control de Inventario	4
HU-06: Reportes de Ventas	5
HU-07: Cancelar Pedido	5
HU-08: Consultar Estado de Pedido	5
3. REQUERIMIENTOS DEL SISTEMA	6
3.1 Requerimientos de Negocio (RN)	6
3.2 Requerimientos Funcionales (RF)	6
3.3 Requerimientos No Funcionales (RNF)	7
4. MODELO CONCEPTUAL	8
4.1 Entidades Principales	8
4.2 Relaciones	9
4.3 Estados del Sistema	9
5. DISEÑO POO	9
5.1 Aplicación de los 4 Pilares	9
Encapsulamiento	9
Herencia	9
Polimorfismo	10
Abstracción	10
5.2 Diagrama de Clases Simplificado	10
6. PLAN DE PRUEBAS	11
6.1 Casos de Prueba Funcionales	11
Caso 1: Crear Pedido Exitoso	11

Caso 2: Manejo de Stock Insuficiente	11
Caso 3: Flujo de Estados de Pedido	12
Caso 4: Validación de Métodos de Pago	12
Caso 5: Reportes de Administrador	12
6.2 Casos de Prueba No Funcionales	13
Caso 6: Rendimiento	13
Caso 7: Usabilidad	13
6.3 Datos de Prueba	13

1. CASO DE NEGOCIO

1.1 Problema Identificado

La cafetería de la universidad maneja pedidos de forma manual, generando múltiples problemáticas que afectan tanto a estudiantes como al personal operativo:

Problemas Críticos:

- **Largas colas** especialmente en horas pico (10:00-12:00h y 14:00-16:00h)
- **Errores frecuentes** en pedidos por comunicación verbal deficiente
- **Tiempo perdido** de estudiantes que deben esperar en filas extensas
- **Control de inventario manual** que genera productos agotados sin previo aviso
- **Métodos de pago limitados** únicamente a efectivo, creando problemas de cambio
- **Falta de trazabilidad** en pedidos y entregas
- **Ausencia de reportes** para toma de decisiones administrativas

1.2 Usuarios del Sistema

Estudiante (Usuario Principal)

- Perfil: Estudiantes universitarios de pregrado y posgrado
- Necesidades: Pedidos rápidos, múltiples formas de pago, seguimiento de pedidos
- Frecuencia de uso: Diaria (desayuno, almuerzo, refrigerios)

Operador de Cafetería

- Perfil: Personal encargado de preparación y entrega de pedidos
- Necesidades: Gestión eficiente de cola, control de inventario, estados de pedidos
- Frecuencia de uso: Durante horario laboral (6:00-18:00h)

Administrador

- Perfil: Coordinador o gerente de la cafetería

- Necesidades: Reportes de ventas, gestión de productos, análisis de rentabilidad
- Frecuencia de uso: Diaria para supervisión, semanal para reportes

1.3 Valor Esperado

Beneficios Cuantificables:

- **Reducción del 60%** en tiempo de espera promedio (de 15 min a 6 min)
- **Disminución del 80%** en errores de pedidos
- **Incremento del 25%** en ventas por mayor rotación de clientes
- **Ahorro de 40%** en tiempo de gestión administrativa

Beneficios Cualitativos:

- **Mejor experiencia del usuario** con pedidos digitales intuitivos
- **Control en tiempo real** del inventario y estados de pedidos
- **Diversificación de métodos de pago** incluyendo bonos estudiantiles
- **Información estratégica** para toma de decisiones comerciales
- **Profesionalización del servicio** con tecnología moderna

Retorno de Inversión: El sistema se pagará en 3 meses mediante el incremento en ventas y la reducción de costos operativos, especialmente en gestión manual de inventario y errores de pedidos.

2. HISTORIAS DE USUARIO

HU-01: Gestión de Productos

Como Administrador

Quiero gestionar el catálogo de productos (crear, editar, desactivar)

Para mantener actualizada la oferta de la cafetería y controlar la disponibilidad

Criterios de Aceptación:

- **Given** que soy administrador autenticado
- **When** creo un producto con nombre, precio base y stock válidos
- **Then** el producto queda disponible para pedidos con precio final (IVA incluido)
- **And** si el stock es 0, no aparece en el menú para estudiantes
- **And** puedo desactivar productos sin eliminarlos del historial

HU-02: Realizar Pedido

Como Estudiante

Quiero crear un pedido seleccionando múltiples productos disponibles

Para ahorrar tiempo en la cola física de la cafetería

Criterios de Aceptación:

- **Given** que hay productos con stock disponible
- **When** selecciono productos y cantidades válidas
- **Then** se crea el pedido con subtotal calculado automáticamente
- **And** se reserva temporalmente el stock por 10 minutos
- **And** puedo agregar o quitar productos antes del pago
- **And** veo el total actualizado en tiempo real

HU-03: Procesar Pago

Como Estudiante

Quiero pagar mi pedido con diferentes métodos (efectivo, tarjeta, bono estudiantil)

Para completar la transacción de forma conveniente según mi preferencia

Criterios de Aceptación:

- **Given** un pedido con total calculado y items válidos
- **When** selecciono método de pago y proporciono datos válidos
- **Then** se procesa el pago y se confirma el pedido automáticamente
- **And** se descuenta definitivamente del inventario
- **And** recibo un comprobante digital del pago
- **And** si uso bono estudiantil, se aplica 10% de descuento

HU-04: Gestionar Cola de Pedidos

Como Operador de Cafetería

Quiero ver pedidos pendientes en orden cronológico de llegada

Para prepararlos y entregarlos de manera eficiente y organizada

Criterios de Aceptación:

- **Given** que hay pedidos confirmados y pagados
- **When** marco un pedido como "En Preparación"
- **Then** cambia su estado y el estudiante puede ver el progreso
- **And** cuando esté listo, lo marco como "Listo para Recoger"
- **And** puedo ver tiempo estimado de preparación por pedido
- **And** se notifica automáticamente al estudiante

HU-05: Control de Inventario

Como Operador

Quiero actualizar stock de productos en tiempo real

Para evitar pedidos de productos agotados y mantener disponibilidad

Criterios de Aceptación:

- **Given** un producto con stock actual conocido
- **When** actualizo la cantidad disponible manualmente
- **Then** se refleja inmediatamente en el sistema
- **And** si llega a 0, se oculta del menú automáticamente
- **And** recibo alertas cuando el stock está bajo (≤ 5 unidades)
- **And** puedo reponer stock de forma masiva

HU-06: Reportes de Ventas

Como Administrador

Quiero generar reportes de ventas diarias y productos más vendidos

Para tomar decisiones informadas sobre inventario, precios y estrategia comercial

Criterios de Aceptación:

- **Given** que existen pedidos completados en el período seleccionado
- **When** genero reporte de ventas diarias o por rango de fechas
- **Then** veo total de ingresos, cantidad de pedidos y productos vendidos
- **And** identifico productos más y menos vendidos con ranking
- **And** analizo distribución por métodos de pago
- **And** puedo exportar datos para análisis externo

HU-07: Cancelar Pedido

Como Estudiante

Quiero cancelar mi pedido antes del pago

Para corregir errores o cambios de decisión sin penalizaciones

Criterios de Aceptación:

- **Given** un pedido en estado "Pendiente de Pago"
- **When** solicito cancelación del pedido
- **Then** se libera inmediatamente el stock reservado
- **And** el pedido se marca como "Cancelado"
- **And** no se cobra ningún monto ni penalización
- **And** puedo crear un nuevo pedido inmediatamente

HU-08: Consultar Estado de Pedido

Como Estudiante

Quiero consultar el estado actual de mi pedido en cualquier momento

Para saber cuándo estará listo para recoger y planificar mi tiempo

Criterios de Aceptación:

- **Given** un pedido con ID válido que me pertenece
 - **When** consulto el estado del pedido
 - **Then** veo el estado actual detallado (Pendiente/En Preparación/Listo/Completado)
 - **And** recibo estimación de tiempo si está en preparación
 - **And** veo historial completo del pedido
 - **And** puedo acceder con solo el ID del pedido
-

3. REQUERIMIENTOS DEL SISTEMA

3.1 Requerimientos de Negocio (RN)

RN-01: Estructura de Precios

Los precios de productos deben incluir IVA del 19% en el precio final mostrado al cliente.

RN-02: Gestión de Reservas

El stock reservado se libera automáticamente tras 10 minutos sin confirmación de pago.

RN-03: Descuentos Estudiantiles

Los bonos estudiantiles otorgan descuento del 10% sobre el total del pedido.

RN-04: Políticas de Cancelación

Los pedidos solo pueden cancelarse antes del procesamiento del pago.

RN-05: Control de Inventario

El inventario no puede ser negativo; stock mínimo permitido es 0.

RN-06: Criterios de Reportes

Los reportes de ventas solo incluyen pedidos con estado "Completado".

3.2 Requerimientos Funcionales (RF)

RF-01: Gestión de Productos

El sistema debe permitir CRUD completo de productos (nombre, precio, stock, descripción, categoría).

RF-02: Creación de Pedidos

Los estudiantes pueden crear pedidos seleccionando múltiples productos con cantidades específicas.

RF-03: Cálculo Automático

El sistema calcula automáticamente subtotal, IVA y total del pedido en tiempo real.

RF-04: Métodos de Pago

Soporte para múltiples métodos: Efectivo (con cálculo de cambio), Tarjeta de Crédito/Débito, Bono Estudiantil.

RF-05: Estados de Pedido

Gestión completa de estados: Pendiente → Pago Procesado → En Preparación → Listo → Completado.

RF-06: Actualización de Inventario

Control de stock en tiempo real con reservas temporales y confirmación de ventas.

RF-07: Cancelación de Pedidos

Funcionalidad para cancelar pedidos antes del pago con liberación automática de stock.

RF-08: Reportes de Ventas

Generación de reportes diarios, por período, productos más vendidos e ingresos totales.

RF-09: Ranking de Productos

Identificación y listado de productos más vendidos con estadísticas de cantidad y ingresos.

RF-10: Interfaces por Usuario

Menús específicos y contextualizados para Estudiante, Operador y Administrador.

RF-11: Persistencia de Datos

Almacenamiento en memoria durante la sesión con opción de exportación (bonus: JSON).

3.3 Requerimientos No Funcionales (RNF)

RNF-01: Rendimiento

El sistema debe responder en menos de 2 segundos para operaciones con hasta 200 productos y pedidos.

RNF-02: Manejo de Errores

Mensajes de error específicos y claros (stock insuficiente, pago fallido, datos inválidos).

RNF-03: Logging Básico

Registro de transacciones críticas: pagos procesados, cambios de inventario, estados de pedidos.

RNF-04: Mantenibilidad

Código limpio con nombres descriptivos en español, arquitectura en capas, principios SOLID.

RNF-05: Testabilidad

Arquitectura con interfaces separadas, dependencias inyectadas, clases con responsabilidades únicas.

RNF-06: UX de Consola

Interfaz intuitiva con menús numerados, colores diferenciados, emojis contextuales, navegación clara.

RNF-07: Portabilidad

Compatible con Windows, Linux y macOS usando .NET 8 multiplataforma.

RNF-08: Validación de Entrada

Validación robusta de todos los inputs del usuario (precios > 0, stock >= 0, datos requeridos).

4. MODELO CONCEPTUAL

4.1 Entidades Principales

Producto (Abstracto)

- Atributos: ID, Nombre, PrecioBase, Stock, StockReservado, Descripción, Activo
- Métodos: ReservarStock(), LiberarReserva(), ConfirmarVenta(), ActualizarPrecio()

Bebida (Hereda de Producto)

- Atributos específicos: TipoBebida, EsCaliente
- Comportamiento: $\text{PrecioFinal} = \text{PrecioBase} * 1.19$

Comida (Hereda de Producto)

- Atributos específicos: TipoComida, RequierePreparacion, TiempoPreparacion
- Comportamiento: $\text{PrecioFinal} = \text{PrecioBase} * 1.19$

Pedido

- Atributos: ID, EstudianteID, FechaCreacion, FechaPago, Estado, MetodoPago
- Relaciones: Contiene múltiples ItemPedido
- Métodos: AgregarItem(), ConfirmarPago(), CambiarEstado()

ItemPedido

- Atributos: Producto, Cantidad, PrecioUnitario, Subtotal
- Relación: Pertenece a un Pedido

MetodoPago (Abstracto)

- Subclases: PagoEfectivo, PagoTarjeta, PagoBono
- Métodos: Procesar(), ObtenerRecibo()

4.2 Relaciones

Producto (1) \longleftrightarrow (N) ItemPedido

ItemPedido (N) \longleftrightarrow (1) Pedido

Pedido (1) \longleftrightarrow (1) MetodoPago

Estudiante (1) \longleftrightarrow (N) Pedido

4.3 Estados del Sistema

Estados de Pedido: Pendiente \rightarrow PagoProcesado \rightarrow EnPreparacion \rightarrow ListoParaRecoger \rightarrow Completado \downarrow Cancelado

Estados de Producto: Activo/Inactivo + Stock (Disponible/Bajo/Agotado)

5. DISEÑO POO

5.1 Aplicación de los 4 Pilares

Encapsulamiento

Protección del estado interno de las entidades críticas mediante propiedades privadas y métodos controlados:

- **Clase Producto:** El stock interno (`_stock`, `_stockReservado`) está protegido y solo se modifica a través de métodos validados como `ReservarStock()` y `ConfirmarVenta()`.
- **Clase Pedido:** La lista de items (`_items`) es privada y se expone como `ICollection`, garantizando que solo se modifique mediante métodos controlados como `AgregarItem()`.
- **Validaciones internas:** Cada método público incluye validaciones que protegen la integridad del estado (stock no negativo, cantidades positivas, estados válidos).

Herencia

Jerarquías de clases que permiten reutilización de código y especialización:

- **Producto \rightarrow Bebida/Comida:** Clase abstracta `Producto` define comportamiento común (gestión de stock, precio base) mientras que `Bebida` y `Comida` implementan cálculo específico de `PrecioFinal`.
- **MetodoPago \rightarrow PagoEfectivo/PagoTarjeta/PagoBono:** Cada subclase implementa su lógica específica de procesamiento manteniendo la interfaz común.

- **Uso de abstract y sealed:** Producto es abstracto para forzar implementación específica, mientras que Bebida y Comida son sealed para evitar herencia innecesaria.

Polimorfismo

Comportamientos dinámicos a través de interfaces y métodos virtuales:

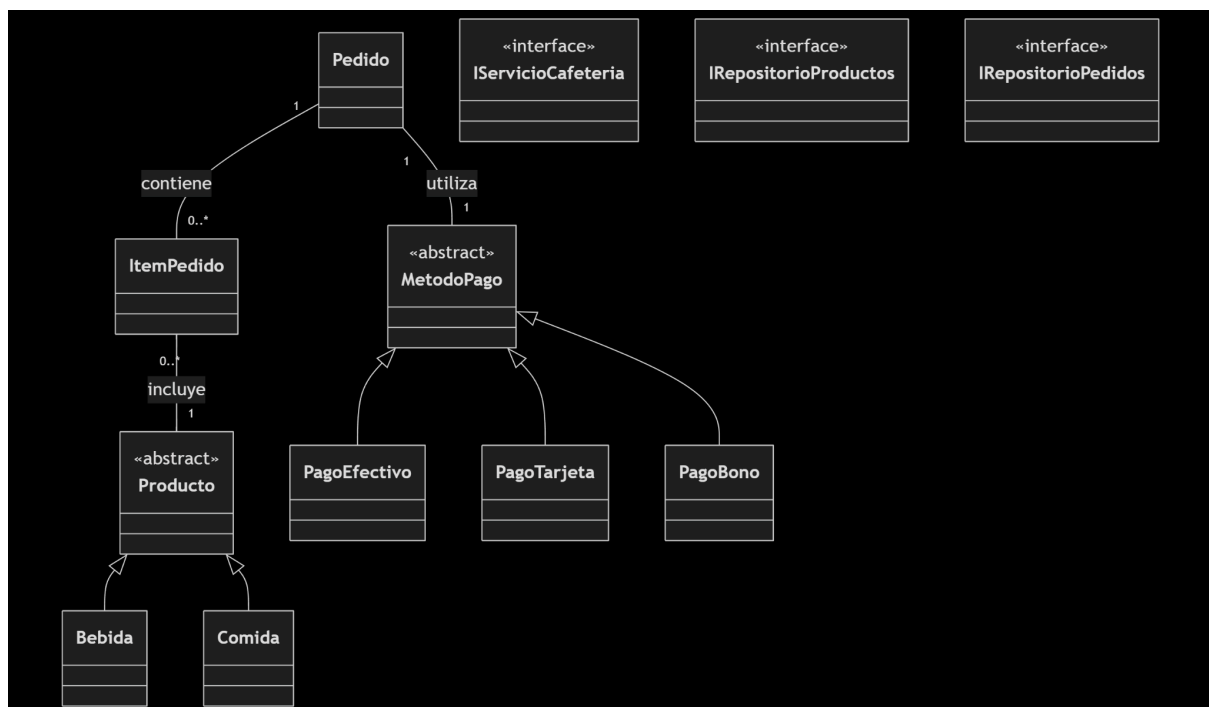
- **Interfaces de servicio:** IServicioCafeteria, IRepositorioProductos permiten intercambiar implementaciones sin afectar el código cliente.
- **Métodos de pago dinámicos:** El mismo método ProcesarPago() se comporta diferente según el tipo concreto (PagoEfectivo valida monto, PagoBono aplica descuento).
- **Estrategias intercambiables:** Los repositorios pueden cambiar de memoria a JSON a base de datos sin modificar la lógica de negocio.

Abstracción

Interfaces simples que ocultan la complejidad interna:

- **IServicioCafeteria:** Expone operaciones de alto nivel (CrearPedido, ProcesarPago) ocultando la coordinación entre repositorios e inventario.
- **Separación de responsabilidades:** Cada interfaz tiene una responsabilidad específica y bien definida.
- **Contratos claros:** Los métodos de las interfaces definen exactamente qué hacen sin exponer cómo lo hacen.

5.2 Diagrama de Clases



6. PLAN DE PRUEBAS

6.1 Casos de Prueba Funcionales

Caso 1: Crear Pedido Exitoso

Objetivo: Verificar el flujo completo de creación de pedido

Datos de Entrada:

- Estudiante: "EST001"
- Productos: Café Americano (2 unidades), Empanada (1 unidad)
- Método de pago: Bono estudiantil "BONO123"

Pasos:

1. Crear pedido para estudiante EST001
2. Agregar Café Americano x2 (precio: \$3,500 c/u)
3. Agregar Empanada x1 (precio: \$3,200)
4. Procesar pago con bono estudiantil
5. Verificar descuento del 10%

Resultado Esperado:

- Pedido creado con ID único
- Subtotal: \$10,200
- Descuento: \$1,020
- Total final: \$9,180
- Stock reservado correctamente

Caso 2: Manejo de Stock Insuficiente

Objetivo: Validar control de inventario

Datos de Entrada:

- Producto: Empanada con stock = 2
- Cantidad solicitada: 5

Pasos:

1. Intentar agregar 5 empanadas al pedido
2. Verificar mensaje de error

Resultado Esperado:

- Error: "Stock insuficiente para Empanada. Disponible: 2, Requerido: 5"
- No se modifica el stock
- Pedido no se actualiza

Caso 3: Flujo de Estados de Pedido

Objetivo: Verificar transiciones de estado

Datos de Entrada:

- Pedido pagado con ID válido

Pasos:

1. Operador inicia preparación → Estado: EnPreparacion
2. Operador marca como listo → Estado: ListoParaRecoger
3. Operador completa entrega → Estado: Completado

Resultado Esperado:

- Cada transición actualiza estado correctamente
- No se permiten transiciones inválidas
- Estudiante puede consultar estado en tiempo real

Caso 4: Validación de Métodos de Pago

Objetivo: Probar diferentes métodos de pago

Sub-caso 4.1: Pago Efectivo

- Monto del pedido: \$10,000
- Monto recibido: \$15,000
- Resultado: Cambio = \$5,000

Sub-caso 4.2: Pago Tarjeta Válida

- Número: "1234567890123456" (16 dígitos)
- Titular: "Juan Pérez"
- Resultado: Pago aprobado

Sub-caso 4.3: Pago Tarjeta Inválida

- Número: "123456" (6 dígitos)
- Resultado: Error "Número de tarjeta inválido"

Caso 5: Reportes de Administrador

Objetivo: Verificar generación de reportes

Datos de Entrada:

- 3 pedidos completados en el día

- Ventas: Café (5 unidades), Empanada (3 unidades)

Resultado Esperado:

- Reporte muestra ingresos totales correctos
- Top productos más vendidos en orden correcto
- Distribución por métodos de pago exacta

6.2 Casos de Prueba No Funcionales

Caso 6: Rendimiento

Objetivo: Verificar tiempos de respuesta

Datos: 100 productos, 50 pedidos **Métrica:** Operaciones < 2 segundos

Caso 7: Usabilidad

Objetivo: Evaluar experiencia de usuario

Criterios:

- Menús intuitivos y numerados
- Mensajes de error claros
- Navegación sin confusión

6.3 Datos de Prueba

Productos de Prueba:

- BEB001: Café Americano - \$3,500 (Stock: 50)
- COM001: Sandwich Pollo - \$8,500 (Stock: 20)
- SNK001: Empanada - \$3,200 (Stock: 10)

Estudiantes de Prueba:

- EST001, EST002, EST003

Métodos de Pago de Prueba:

- Efectivo: Varios montos
- Tarjeta: 1234567890123456
- Bono: BONO123, BONO456