

SISTEMA DE GESTIÓN DE COMEDORES COMUNITARIOS

Miguel Angel Muñoz Bohorquez

Ingeniería de Sistemas

NRC 75317: Programación Orientada a Objetos (POO)

Jasson Diaz Zamudio

Mayo 2025

TABLA DE CONTENIDOS

1. [Justificación del Proyecto](#)
 2. [Descripción del Sistema](#)
 3. [Arquitectura y Patrones de Diseño](#)
 4. [Descripción de Clases](#)
 5. [Casos De Uso](#)
 6. [Evidencia de Ejecución](#)
 7. [Conclusiones](#)
-

JUSTIFICACIÓN DEL PROYECTO

Problemática Social

Los comedores comunitarios representan una herramienta fundamental en la lucha contra la inseguridad alimentaria en comunidades vulnerables. La gestión manual de estos espacios presenta múltiples desafíos:

- **Falta de control de inventarios:** Sin un sistema adecuado, es difícil mantener un registro preciso de los alimentos disponibles
- **Criterios de selección inconsistentes:** La evaluación de beneficiarios puede verse afectada por subjetividad
- **Desperdicio de recursos:** Sin planificación adecuada, los alimentos pueden caducar o distribuirse de manera ineficiente
- **Falta de trazabilidad:** Dificultad para generar reportes y estadísticas sobre el impacto social

Solución Propuesta

El sistema desarrollado automatiza y estandariza la gestión de comedores comunitarios mediante:

1. **Gestión de inventarios automatizada:** Control preciso de alimentos y cálculo automático de platos disponibles
2. **Evaluación objetiva de vulnerabilidad:** Algoritmo estandarizado basen criterios socioeconómicos
3. **Administración centralizada:** Manejo de múltiples comedores desde una sola interfaz

4. **Trazabilidad completa:** Registro detallado de beneficiarios y distribución de alimentos

Impacto Social Esperado

- Optimización de recursos alimentarios
 - Atención equitativa y transparente a beneficiarios
 - Mejora en la planificación y distribución de alimentos
 - Generación de datos para políticas públicas
-

DESCRIPCIÓN DEL SISTEMA

Funcionalidades Principales

1. Gestión de Comedores

- Administración de 15 comedores comunitarios predefinidos en Bogotá
- Información detallada: ubicación, capacidad, horarios de atención
- Selección intuitiva mediante menú numerado

2. Control de Inventario

- Seguimiento de 5 categorías de alimentos:
 - Cereales y harinas
 - Proteínas
 - Verduras
 - Frutas
 - Bebidas nutritivas
- Cálculo automático de platos completos disponibles
- Sistema de entrega y distribución de alimentos

3. Gestión de Beneficiarios

- Registro completo de datos personales
- Sistema de evaluación de vulnerabilidad socioeconómica
- Proceso de inscripción con validación automática de aptitud
- Consulta y administración de beneficiarios por comedor

4. Algoritmo de Vulnerabilidad

Sistema de puntuación basado en criterios objetivos:

- **Alimentación:** Número de comidas diarias (1-3)
- **Ingresos:** Sueldo mensual vs personas a cargo
- **Discapacidad:** Condición de discapacidad
- **Vivienda:** Tipo de vivienda (propia/arriendo/calle)
- **Criterios adicionales:** Desplazamiento, menores a cargo

Umbral de aptitud: ≥ 15 puntos para ser beneficiario

ARQUITECTURA Y PATRONES DE DISEÑO

Patrón MVC (Modelo-Vista-Controlador)

El sistema implementa una arquitectura MVC que separa claramente las responsabilidades:

Modelo (models.py)

- Comedor: Representa la estructura física del comedor
- Alimentos: Maneja el inventario y lógica de distribución
- Beneficiario: Gestiona datos y evaluación de vulnerabilidad
- ComedorComunitario: Agregador que combina comedor, alimentos y beneficiarios

Vista (views.py)

- ComedorView: Interfaz de usuario en consola
- Presentación de menús y formularios
- Validación de entrada de datos
- Formateo de información para el usuario

Controlador (controllers.py)

- ComedorController: Lógica de negocio y coordinación
- Orquestación entre modelo y vista
- Manejo del flujo de la aplicación
- Procesamiento de operaciones complejas

Principios de Diseño Aplicados

Separación de Responsabilidades

Cada clase tiene una responsabilidad específica y bien definida:

- Modelos: Lógica de datos y reglas de negocio
- Vista: Presentación e interacción con usuario
- Controlador: Coordinación y flujo de control

Encapsulación

- Métodos privados (`_metodo`) para funcionalidad interna
- Interfaces públicas claras y consistentes
- Protección de datos sensibles

Reutilización de Código

- Métodos utilitarios reutilizables (`leer_entero`, `mostrar_mensaje`)
- Estructuras de datos consistentes
- Separación de lógica común

DESCRIPCIÓN DE CLASES

Clase Comedor

Propósito: Representar la información básica de un comedor comunitario.

Atributos:

- `nombre_comedor`: Identificador del comedor
- `direccion`: Ubicación física
- `capacidad`: Número máximo de beneficiarios
- `horario`: Horario de atención

Métodos:

- `__init__()`: Constructor con parámetros opcionales
- `__str__()`: Representación string para visualización

Clase Alimentos

Propósito: Gestionar el inventario de alimentos y cálculo de porciones.

Atributos:

- `cereales_harinas`: Cantidad en gramos

- proteínas: Cantidad en gramos
- verduras: Cantidad en gramos
- frutas: Cantidad en gramos
- bebidas_nutritivas: Cantidad en gramos
- platos_comida: Platos completos calculados automáticamente

Métodos:

- entrega_alimentos(): Agregar inventario recibido
- repartir_alimentos(): Distribuir comida a beneficiarios
- _actualizar_platos(): Calcular platos disponibles (método privado)

Lógica de Cálculo:

Platos disponibles = min(
 cereales÷110, proteínas÷80, verduras÷70,
 frutas÷110, bebidas÷225
)

Clase Beneficiario

Propósito: Modelar beneficiarios y evaluar su vulnerabilidad socioeconómica.

Atributos:

- nombre, num_cedula, edad, sexo, estrato: Datos personales
- discapacidad: Estado de discapacidad
- puntaje_vulnerabilidad: Puntuación calculada

Métodos:

- calcular_puntaje_vulnerabilidad(): Algoritmo de evaluación
- es_apto(): Determina elegibilidad (≥ 15 puntos)

Algoritmo de Vulnerabilidad:

Puntaje = 0

+ Comidas diarias (1=5pts, 2=3pts, 3=0pts)

+ Ingresos per cápita (<200k=5pts, 200-300k=3pts, >300k=0pts)

+ Discapacidad (Sí=5pts, No=0pts)

+ Vivienda (calle=5pts, arriendo=3pts, propia=0pts)

+ Desplazamiento (Sí=5pts, No=0pts)

+ Menores a cargo <5 años (Sí=5pts, No=0pts)

Clase ComedorComunitario

Propósito: Agregador que combina comedor, inventario y beneficiarios.

Atributos:

- comedor: Instancia de Comedor
- alimentos: Instancia de Alimentos
- beneficiarios: Lista de beneficiarios inscritos

Métodos:

- agregar_beneficiario(): Inscribir nuevo beneficiario
- retirar_beneficiario(): Dar de baja beneficiario
- obtener_beneficiario(): Consultar beneficiario específico

Clase ComedorView

Propósito: Interfaz de usuario y presentación de información.

Responsabilidades:

- Mostrar menús y opciones
- Solicitar datos al usuario
- Validar entrada de información
- Formatear salida de datos
- Manejo de errores de entrada

Métodos Principales:

- mostrar_menu_*(): Presentación de menús
- solicitar_datos_*(): Formularios de entrada
- leer_entero(): Validación de números enteros
- mostrar_*(): Formateo de información

Clase ComedorController

Propósito: Controlador principal que orquesta todo el sistema.

Responsabilidades:

- Inicialización del sistema con 15 comedores
- Manejo del flujo principal de la aplicación
- Coordinación entre vista y modelo
- Procesamiento de lógica de negocio

Métodos de Gestión:

- `_gestionar_comedor()`: Bucle principal por comedor
- `_entregar_alimentos()`: Proceso de recepción de alimentos
- `_repartir_comida()`: Distribución a beneficiarios
- `_agregar_beneficiario()`: Process completo de inscripción
- `_retirar_beneficiario()`: Proceso de baja

CASOS DE USO

Caso de Uso Principal: Inscribir Beneficiario

Actor: Administrador del Comedor **Precondiciones:** Sistema iniciado, comedor seleccionado **Flujo Principal:**

1. Administrador selecciona "Añadir beneficiario"
2. Sistema solicita datos personales
3. Administrador ingresa: nombre, cédula, edad, sexo, estrato
4. Sistema solicita datos de vulnerabilidad
5. Administrador ingresa: comidas diarias, sueldo, personas a cargo, discapacidad, vivienda
6. Sistema calcula puntaje inicial
7. **Si puntaje 10-14:** Sistema solicita datos adicionales
8. **Si puntaje ≥ 15 :** Sistema inscribe beneficiario
9. **Si puntaje < 15 :** Sistema rechaza inscripción

Flujos Alternativos:

- **A1:** Datos incompletos → Sistema solicita completar información
- **A2:** Beneficiario ya registrado → Sistema muestra error

- **A3:** Puntaje límite (10-14) → Evaluación adicional requerida

Caso de Uso: Repartir Comida

Actor: Administrador del Comedor **Precondiciones:** Comedor con beneficiarios inscritos

Flujo Principal:

1. Administrador selecciona "Repartir comida"
2. Sistema cuenta beneficiarios inscritos
3. Sistema verifica inventario disponible
4. **Si hay suficientes platos:** Sistema distribuye comida
5. Sistema actualiza inventario
6. Sistema confirma distribución exitosa

Flujos Alternativos:

- **A1:** Sin beneficiarios → Mensaje "No hay beneficiarios"
- **A2:** Inventario insuficiente → Mensaje con cantidad faltante

EVIDENCIA DE EJECUCIÓN

Pantalla Principal

EN CUAL COMEDOR DESEA TRABAJAR

1. Los luceros - Cra 17F No. 69A-32 Sur (Lucero Sur)
2. Potosí - Calle 81 Sur No. 42-09
3. Caracolí - Calle 76A Sur No. 74B-05
- ...
15. Vista Hermosa - Calle 80B Sur No. 44-10
16. Salir del programa

Menú de Comedor

OPCIONES COMEDOR COMUNITARIO (Los luceros)

1. Inventario de Comida

2. Entrega de Comida
3. Repartir comida
4. Beneficiarios inscritos
5. Añadir beneficiario
6. Retirar Beneficiario
7. Salir, escoger otro comedor

Ejemplo de Inventario

COMIDA ACTUAL:

552g en cereales y harinas

558g en proteínas

765g en verduras

240g en frutas

865g en bebidas nutritivas

Este comedor cuenta con 2 platos de comida disponibles

Proceso de Inscripción

Ingrese los siguientes datos sobre el nuevo beneficiario

Nombre: María González

Número de Cédula: 12345678

Edad: 45

Sexo (M/F): F

Estrato: 1

Llene los siguientes datos para determinar si el beneficiario es apto o no

¿Cuántas veces al día come?: 2

Ingrese su sueldo mensual: 180000

¿Cuántas personas dependen de usted económicamente?: 3

¿Tiene alguna discapacidad? (si/no): no

Tipo de vivienda (propia/arriendo/calle): arriendo

Es necesario hacer una verificación adicional...

¿El beneficiario ha sufrido de desplazamiento? (si/no): si

Beneficiario María González inscrito con éxito

Cálculo de Puntaje (Ejemplo María González)

- Comidas diarias (2): 3 puntos
- Ingresos per cápita ($180000/3 = 60000$): 5 puntos
- Sin discapacidad: 0 puntos
- Vivienda en arriendo: 3 puntos
- Desplazamiento: 5 puntos
- **Total: 16 puntos** → APTO

CONCLUSIONES

Logros Técnicos

1. Arquitectura Sólida

- Implementación exitosa del patrón MVC
- Separación clara de responsabilidades
- Código mantenible y escalable

2. Funcionalidad Completa

- Sistema funcional para 15 comedores reales de Bogotá
- Algoritmo robusto de evaluación de vulnerabilidad

- Gestión integral de inventarios y beneficiarios

3. Usabilidad

- Interfaz intuitiva de consola
- Validación exhaustiva de datos de entrada
- Mensajes claros y orientación al usuario

IMPACTO SOCIAL POTENCIAL

1. Transparencia

- Criterios objetivos y reproducibles
- Eliminación de sesgos en la selección
- Trazabilidad completa de decisiones

2. Eficiencia Operacional

- Automatización de cálculos complejos
- Optimización del uso de recursos
- Reducción de errores humanos

3. Escalabilidad

- Arquitectura preparada para crecimiento
- Fácil adición de nuevos comedores
- Extensión de funcionalidades

Reflexión Final

El sistema desarrollado representa una solución tecnológica viable para la gestión de comedores comunitarios, con potencial de impacto social significativo. La implementación de criterios objetivos para la evaluación de vulnerabilidad contribuye a la equidad y transparencia en la atención a poblaciones vulnerables.

La arquitectura MVC empleada facilita el mantenimiento y la evolución del sistema, permitiendo adaptaciones futuras según las necesidades específicas de cada comunidad. El enfoque modular y la separación de responsabilidades garantizan que el sistema pueda crecer y adaptarse a nuevos requerimientos.

Este proyecto demuestra cómo la tecnología puede ser una herramienta poderosa para abordar problemas sociales complejos, proporcionando soluciones sistemáticas, transparentes y escalables que benefician directamente a las comunidades más necesitadas.