

# Título: Mr. Cuerno - Sistema Integral de Gestión para Restaurante

## Contexto Actual

- Presupuesto total: \$50,000 MXN
- Equipo: 5 desarrolladores asignados
- Infraestructura inicial:
  - Sin equipos de cómputo existentes
  - Sin conexión a internet disponible
  - Requiere manejo de pedidos locales y a domicilio

## Alcance del Proyecto

Aplicación de escritorio en Python con:

1. Módulo para clientes:
  - Interfaz gráfica para pedidos en local
  - Sistema para recibir pedidos a domicilio vía SMS/WhatsApp
  - Visualización completa del menú
  - Generación automática de tickets
2. Módulo administrativo:
  - Gestión de inventario en tiempo real
  - Registro detallado de ventas
  - Control de usuarios y permisos
  - Generación de reportes financieros

## Desglose de Presupuesto

Concepto	Costo (MXN)	Detalles
Equipos de cómputo	20,000	2 computadoras con Windows 10 + pantalla táctil

Periféricos	6,000	Impresora térmica, lector QR, dispositivos de red
Servicios de comunicación	3,000	API para SMS/WhatsApp (1 año)
Desarrollo	18,000	Salarios para equipo de desarrollo (2 meses)
Contingencia	3,000	Soporte técnico e imprevistos

## Roles del Equipo y Responsabilidades

1. Product Owner / Project Manager (David)
  - Definir prioridades y roadmap del producto
  - Gestionar comunicación con stakeholders
  - Aprobar entregables finales
2. Business Analyst (Julio Inda)
  - Documentar requisitos del negocio
  - Diseñar flujos de pedidos local/domicilio
  - Validar casos de uso con el restaurante
3. Frontend Developer (Fabián)
  - Implementar interfaz gráfica con Tkinter/PyQt
  - Diseñar pantallas para ambos modos (cliente/admin)
  - Asegurar experiencia de usuario intuitiva
4. Backend Developer (Alexander)
  - Desarrollar lógica central de la aplicación
  - Implementar sistema de procesamiento de SMS/WhatsApp
  - Gestionar conexión con base de datos
5. Database Administrator (Gabriel)
  - Diseñar estructura óptima de base de datos
  - Implementar sistema de backups automáticos
  - Optimizar consultas para mejor rendimiento

## Arquitectura Propuesta

Componentes principales:

- Interfaz gráfica unificada (Tkinter/PyQt)
- Base de datos local (SQLite) con cifrado básico
- Módulo de comunicación para pedidos externos
- Sistema de impresión térmica integrado
- Generador de reportes en Excel/PDF

Flujo de trabajo completo:

1. Clientes en local realizan pedidos mediante interfaz táctil
2. Clientes a domicilio envían pedidos vía SMS/WhatsApp
3. Sistema procesa y registra todas las órdenes en la base de datos
4. Cocina recibe tickets impresos automáticamente
5. Administración genera reportes consolidados

## **Consideraciones Técnicas**

Ventajas:

- Operación completamente offline
- Solución todo-en-uno para ambos tipos de pedidos
- Bajo costo operativo post-implementación
- Fácil capacitación para personal

Limitaciones:

- Capacidad limitada por hardware disponible
- Requiere supervisión manual de comunicaciones
- Escalabilidad limitada sin actualización de equipos

## **Plan de Implementación**

1. Fase 1 - Preparación (Semana 1-2):
  - Adquisición e instalación de equipos
  - Configuración inicial del entorno
  - Definición detallada de requerimientos
2. Fase 2 - Desarrollo (Semana 3-6):
  - Construcción de módulo administrativo
  - Implementación de interfaz cliente
  - Desarrollo de sistema de comunicación externa
3. Fase 3 - Despliegue (Semana 7-8):

- Pruebas integrales del sistema
- Capacitación completa al personal
- Implementación final y ajustes

## **Recomendaciones Operativas**

1. Establecer protocolo estricto para:
  - Respaldos diarios de información
  - Monitoreo de insumos para impresión
  - Actualización periódica del menú
2. Implementar sistema de:
  - Numeración única para todos los pedidos
  - Validación automática de formatos SMS
  - Notificaciones por sonido para nuevos pedidos
3. Considerar a futuro:
  - Migración a hardware más potente
  - Sistema de sincronización con cocina
  - Integración con servicios de reparto

## **Tecnologías Utilizadas**

- Lenguaje principal: Python 3.x
- Interfaz gráfica: Tkinter + CustomTkinter
- Base de datos: SQLite con SQLAlchemy
- Comunicaciones: Twilio API (SMS) / pywhatkit (WhatsApp)
- Reportes: Pandas + OpenPyXL + FPDF