



Universidad de las Fuerzas Armadas – ESPE

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

**MODELO DE PROCESOS DE DESARROLLO DE
SOFTWARE - NRC:27801**

**Tema: contratación de Servicio y desarrollo de
página web de reservas para un restaurante**

ByteCore



Líder de Proyecto:

**SAMUEL ESTEBAN
DÁVALOS MONTALVO**

Fecha: Quito, 20 de octubre de 2025



TÉRMINOS DE REFERENCIA

FECHA:	UNIDAD REQUERENTE:
23 de octubre de 2025

OBJETO DE CONTRATACIÓN

“Contratación de servicio de diseño y desarrollo de página web para una reserva de un restaurante.”

Consultora:	ByteCore	 ByteCore
Cliente:	Ing. Fernando Solís	
Líder de Proyecto:	Samuel Dávalos	

ANTECEDENTES/JUSTIFICACIÓN

ANTECEDENTES

Con fecha 16 de octubre de 2025, el Ing. Fernando Solís, en calidad de Dueño y Gerente del restaurante [Nombre del Restaurante], ha identificado la necesidad crítica de migrar la gestión de reservas de un modelo manual (telefónico/físico) a una plataforma digital. Este cambio es imperativo para mejorar la eficiencia operativa y la experiencia del cliente.

JUSTIFICACIÓN

La inversión en el desarrollo de este sistema se justifica por las siguientes razones operativas y estratégicas:

- Mitigación de Errores:** La automatización eliminará los errores comunes asociados a la gestión manual, como la doble reserva de mesas, la omisión de registros y la mala comunicación de disponibilidad.
- Optimización de Recursos:** Permitirá al personal del restaurante enfocar su tiempo en la atención al cliente en sitio, en lugar de estar dedicado a la gestión telefónica y la transcripción de datos.
- Disponibilidad 24/7:** La plataforma permitirá a los clientes realizar reservas fuera del horario laboral del restaurante, incrementando las oportunidades de negocio.
- Mejora de la Toma de Decisiones:** El módulo de administración facilitará el acceso a datos reales sobre los patrones de reserva y ocupación, lo que mejorará la gestión estratégica de las mesas y la planificación futura.

OBJETIVOS



OBJETIVO GENERAL

Implementar una plataforma web integral para el restaurante *\$Nombre del Restaurante* que automatice el proceso de reservas, permitiendo tanto la reserva del local completo como la reserva por mesas cuando corresponda. Además, la plataforma debe incorporar la funcionalidad de pre-pedido de platos (menú) y ofrecer herramientas de gestión en tiempo real, con el fin de optimizar la logística del servicio y maximizar la satisfacción del cliente.

Nº	OBJETIVO ESPECÍFICO	MÓDULO INVOLUCRADO	RELEVANCIA PARA LA BASE DE DATOS
OE.1	Gestionar la Reserva de Mesas: Permitir al cliente seleccionar la fecha, hora y número de personas, y registrar la reserva en el sistema, asegurando que no haya conflictos de horario o capacidad.	Clientes / Backend	Transaccionalidad: Debe asegurar la atomicidad del registro de la reserva para evitar duplicidades.
OE.2	Monitorear la Ocupación en Tiempo Real: Desarrollar un dashboard para el personal administrativo que muestre el estado de cada mesa (Ocupada, Reservada, Disponible, Limpieza) y permita la actualización inmediata del estado de la mesa (e.g., desocupar mesa).	Administración	Integridad de Datos: Garantizar que el estado físico de la mesa se refleje con precisión en el sistema.
OE.3	Controlar el Inventario de Platos: Implementar una funcionalidad de catálogo en el módulo de pre-pedido que permita al administrador establecer y reducir el límite de disponibilidad de platos, evitando que se puedan solicitar ítems agotados.	Clientes / Administración	Restricciones: Uso de un campo numérico para el stock de platos que se reduce al registrar un pre-pedido.
OE.4	Generar el Detalle de Consumo: Al finalizar el servicio, el	Administración	Consultas y Reportes: Debe poder generar un



	sistema debe generar un documento de consumo (Nota de Consumo) que refleje el total de los ítems pre-ordenados más alquiler.		resumen consolidado extrayendo datos de las tablas de Pedidos, Ítems y Reservas.
OE.5	Ver todas las reservas registradas: El sistema debe tener una apartado en donde se pueda observar una lista de todas las reservas que se han hecho, mostrando datos como la fecha, la mesa que se ha asignado y los datos del cliente	Administración	Recolección de datos: Se registra automáticamente la información solicitada en la base de datos, para que pueda mostrar una lista completa de esos datos cuando se necesite.

ALCANCE

El Alcance define los límites precisos de lo que ByteCore entregará en el marco de esta contratación.

3.1. ALCANCE FUNCIONAL INCLUIDO (ENTREGABLES)

El sistema web a desarrollar incluirá los siguientes módulos y funcionalidades principales:

A. Módulo de Reservas (Clientes)

1. Generación de Reserva de Mesas: Permitir al usuario la selección de la fecha, hora y número de comensales, con validación inmediata de la disponibilidad.
2. Pre-Pedido de Platos: Integración con el menú digital para que el cliente pueda seleccionar y adjuntar la solicitud de platos en el momento de la reserva.

B. Módulo de Administración (Personal del Restaurante)

1. Monitor de Ocupación en Tiempo Real: Interfaz administrativa para visualizar el estado actual de todas las mesas.
2. Gestión de Mesas y Disponibilidad: Funcionalidad para que el personal pueda desocupar y reasignar mesas con un clic, reflejando el cambio de estado de manera instantánea.
3. Control de Límites de Platos (Stock): Panel para que el administrador pueda establecer el límite de disponibilidad de platos clave, evitando pre-pedidos de ítems agotados.
4. Registro de Consumo: Capacidad para que el sistema genere una Nota de Consumo detallada que consolide los ítems pre-ordenados más los consumos adicionales, mostrando el subtotal e impuestos aplicables.

C. Componentes de Infraestructura y Datos

- Diseño de Base de Datos: Entrega del modelo lógico y físico de la Base de Datos, optimizado para la integridad y eficiencia de la gestión de reservas.



- Diseño Web Adaptativo (Responsive): La interfaz será completamente funcional y estética en dispositivos móviles, tabletas y PC.

3.2. EXCLUSIONES IMPORTANTES (FUERA DE ALCANCE)

Los siguientes elementos NO están incluidos en la presente contratación:

- Facturación Electrónica: La generación de facturas electrónicas con validez tributaria o la integración con sistemas de contabilidad externos.
- Integración de Hardware: Instalación, configuración o integración con hardware (impresoras de cocina, terminales de punto de venta [POS]).
- Desarrollo de Aplicación Móvil Nativa: El proyecto solo abarca la plataforma web (accesible desde móviles).
- Módulo de Pago en Línea: El sistema no procesará pagos ni tarjetas de crédito; el pago se mantiene en el local.

PILA TECNOLÓGICA

Componente	Tecnología Seleccionada	Entorno de Trabajo
Lenguajes Principal	Python (Lógica/Backend), HTML, CSS, JavaScript (Frontend), PHP (Alternativa para-Scripts).	IDE: Visual Studio Code (VSC)
Framework/Estilos	Bootstrap (Diseño Responsive).	SO de Desarrollo: Linux (Ubuntu/Debian)
Base de Datos	MySQL (Motor Relacional).	Control de versiones: Git (en un repositorio como GitHub/GitLab).
Plataforma Final	Dominio: www.[NombreRestaurante]reservas.com	Hosting: Servidor Cloud con sistema operativo Linux.

ESTÁNDARES DE CALIDAD Y REFERENCIA

Estándar / Filosofía	Aplicación en el Proyecto
ISO/IEC 29110	Se utiliza para el Planificación y Seguimiento formal del proyecto, asegurando que cada fase se documente correctamente y se complete a tiempo.



Estándar / Filosofía	Aplicación en el Proyecto
Filosofía DevOps	Se enfoca en la Integración Continua (CI) y Entrega Continua (CD), priorizando la automatización de las pruebas de funcionalidad para detectar errores rápidamente.
ISO/IEC 25010	Guía de Calidad: Los criterios de aceptación se basarán en la Usabilidad y la Fiabilidad de las transacciones de reserva en la Base de Datos.

TRAZABILIDAD

Elemento	Propósito	Herramienta
Registro de Tareas	Registrar quién hizo qué tarea, cuándo inició y finalizó, y el resultado de su trabajo (código, <i>commits</i> , documentación).	Software de Gestión de Proyectos (Ej. Jira, Trello, Azure DevOps)
Control de Versiones	Usar Git para registrar automáticamente cada cambio en el código (<i>commit</i>), vinculando la acción al nombre del programador, la fecha/hora y el propósito del cambio.	Repositorio Git (GitHub/GitLab)
Revisión de Pares (Code Review)	Asegurar que el código de un desarrollador sea revisado y aprobado por otro, verificando la calidad y dejando una traza de la aprobación.	Funcionalidades de <i>Pull Request</i> del repositorio Git.
Métricas	El Líder de Proyecto extraerá métricas periódicas para ver el estado del esfuerzo individual y el avance general del proyecto.	Hojas de cálculo o dashboards del software de gestión.

PROCESO DE DESARROLLO DEL SOFTWARE

ESTRUCTURA DE DESGLOSE DEL TRABAJO

Nº	Tarea / Entregable	Duración (Días)	Responsable
1.1	Análisis de Requerimientos: Requisitos Funcionales y No Funcionales.	2	Analistas / Líder



Nº	Tarea / Entregable	Duración (Días)	Responsable
1.2	Modelado Conceptual (UML):	3	Analistas
1.2.1	Elaboración de Casos de Uso (Clientes, Administración, Pre-Pedidos).	3	Analistas
1.2.2	Creación del Diagrama de Clases (Definición de Entidades y Atributos).	4	Analistas / DB
1.3	Modelado de Datos: Creación del Modelo Entidad-Relación (E-R).	3	Equipo DB
1.4	Revisión Interna: Documento de Diseño Conceptual Aprobado.	2	Líder / Equipo

PROTOTIPADO

Nº	Tarea / Entregable	Duración (Días)	Responsable
2.1	Diseño Lógico de Base de Datos: Normalización de las Tablas (hasta 3FN).	4	Equipo DB
2.2	Diseño de Interfaz (Mockups): Diseño de la experiencia de usuario (UI/UX) para Clientes y Administrador.	5	Diseño/Frontend
2.3	Plan de Pruebas: Definición de escenarios de prueba unitaria y funcional.	3	QA / Lider
2.4	Configuración del Entorno DevOps: Configurar repositorio Git y entorno Linux de desarrollo (VSC).	3	DevOps / Backend



DESARROLLO CENTRAL

Nº	Tarea / Entregable	Duración (Días)	Responsable
3.1	Implementación del Schema DB: Creación del <i>script SQL</i> y tablas en MySQL.	4	Equipo DB
3.2	Desarrollo Backend (Python/PHP): Lógica de Transaccionalidad, <i>APIs</i> de control de stock.	8	Backend
3.3	Desarrollo Frontend (HTML/CSS/JS/Bootstrap): Interfaz pública de reserva y panel de administración.	7	Frontend
3.4	Integración Continua (CI): Configuración de <i>pipelines</i> para pruebas automáticas.	3	DevOps

PRUEBAS Y DESPLIEGUE

Nº	Tarea / Entregable	Duración (Días)	Responsable
4.1	Pruebas Funcionales: Verificación del cumplimiento de los OE (Ocupación en Tiempo Real, Nota de Consumo).	5	QA / Equipo
4.2	Capacitación y Manuales: Creación de Manual de Usuario para el Ing. Solis y su personal.	3	Líder / Analistas
4.3	Despliegue (CD): Migración final del código a la plataforma Linux (Dominio: www.[NombreRestaurante]reservas.com).	2	DevOps
4.4	Cierre del Proyecto: Revisión final y Entrega del Código Fuente.	2	Líder



CRONOGRAMA

Hito (Fase)	Duración Estimada	Fecha Estimada de Finalización	Entregable Clave
I. Planificación y Diseño Conceptual	3 semanas	20 de noviembre del 2025	Documento de Requerimientos Final y Diagrama Entidad-Relación (E-R) de la Base de Datos.
II. Diseño Lógico y Prototipado	3 Semanas	11 de diciembre del 2025	Esquema de Tablas Normalizadas (3FN) y Prototipos de Interfaz de Usuario (Mockups).
III. Desarrollo Central (Beta)	6 Semanas	01 de enero del 2026	Versión Beta Funcional (Módulos de Cliente y Administración listos para pruebas internas).
IV. Pruebas, Ajustes y Despliegue	5 Semanas	03 de febrero del 2026	Lanzamiento Oficial (Go-Live), Código Fuente y Manual de Usuario.
Plazo Total Estimado del Contrato	17 Semanas	17 de febrero del 2026	Entrega Final del Sistema Operativo.

MITIGACIÓN Y ERRORES

6.1 Problemas Actuales a Mitigar - Justificación

El proyecto se justifica por la necesidad de migrar de una gestión manual de la reserva de mesas a una digital, siendo los principales errores asociados al modelo manual los siguientes:

6.1.1 Doble Reserva de Mesas: Error en la asignación de mesas, resultando en conflictos de disponibilidad.

6.1.2 Omisión de Registros: Pérdida o falta de registro de las reservas.

6.1.3 Mala Comunicación de Disponibilidad: Información incorrecta o desactualizada sobre la disponibilidad de mesas para una reserva.

6.1.4 Uso Ineficiente de Recursos: El personal dedica tiempo a la gestión telefónica y transcripción de datos en lugar de la atención al cliente en sitio.

6.1.5 Limitación de Horario: La imposibilidad de realizar o anular reservas fuera del horario laboral del restaurante.



6.2 Estrategias de Mitigación de Errores:

El proyecto implementa ciertas medidas de mitigación en el nivel operativo como a nivel de desarrollo para cubrir los errores descritos anteriormente.

6.2.1 Mitigación Operativa

6.2.1.1 Mitigación de Errores Comunes (Doble Reserva/Omisión):

- **Automatización:** El sistema digital eliminará los errores comunes asociados a la gestión manual.
- **Validación de Disponibilidad:** La funcionalidad de reserva permitirá la validación inmediata de la disponibilidad de mesas para una fecha, hora y número de comensales específicos.
- **Transaccionalidad:** Se debe asegurar la atomicidad del registro de la reserva para evitar duplicidades (OE.1).

6.2.1.2 Mitigación de Errores de Comunicación/Disponibilidad:

- **Monitor en Tiempo Real:** El personal administrativo tendrá un dashboard que muestra el estado de cada mesa (Ocupada, Reservada, Disponible) y permita la actualización inmediata.
- **Disponibilidad 24/7:** La plataforma permitirá a los clientes realizar y anular reservas fuera del horario laboral, reduciendo la dependencia de la comunicación manual

6.2.1.3 Mitigación de Errores de Pedido/Stock:

- **Control de Inventario:** Implementación de una funcionalidad que permita al administrador establecer y reducir el límite de disponibilidad de platos, evitando que se puedan solicitar ítems agotados (OE.3)

6.2.2 Mitigación en el Proceso de Desarrollo

6.2.2.1 Mitigación de Errores de Calidad:

- **Filosofía DevOps:** Automatización de las pruebas de funcionalidad para detectar errores rápidamente a través de la Integración Continua (CI) y Entrega Continua (CD).
- **ISO/IEC 25010:** Los criterios de aceptación, para la calidad del producto Software

6.2.2.2 Mitigación de Errores de Integridad de Código:

- **Uso de Revisión de Pares:** Asegura que el código de un desarrollador sea revisado y aprobado por otro code review
- **Control de Versiones (Git):** Registra automáticamente cada cambio en el código, vinculando la acción al programador y su propósito, lo que permite revertir errores.

6.2.2.3 Mitigación de Fallos de Datos:

- **Normalización de Base de Datos:** El Diseño Lógico de la Base de Datos incluye la Normalización de las Tablas (hasta 3FN) para optimizar la integridad de los datos



Nota: La presente planificación se estructura en base al estándar IEEE 1058 – Software Project Management Plan (SPMP). Para más información, consulte <https://standards.ieee.org/1058>.

8. Anexos

El presente documento de planificación se fundamenta en el estándar IEEE 1058 – Software Project Management Plan (SPMP), emitido por el Institute of Electrical and Electronics Engineers (IEEE). Este estándar define la estructura, contenido y procedimientos recomendados para la elaboración de un plan de gestión de proyectos de software, estableciendo lineamientos sobre:

La organización del proyecto, incluyendo roles y responsabilidades.

Los procesos de gestión relacionados con el seguimiento, control y comunicación.

Los procesos técnicos, que abarcan el desarrollo, aseguramiento de calidad y control de configuración.

Los mecanismos de documentación y trazabilidad, para garantizar la correcta planificación y control del proyecto.

El uso de este estándar busca asegurar que la planificación del proyecto cumpla con prácticas reconocidas internacionalmente en la ingeniería de software, promoviendo la claridad, consistencia y control formal durante todo su ciclo de vida.

Referencia: IEEE 1058 – Software Project Management Plan (SPMP). Disponible en
<https://standards.ieee.org/1058>.