

**PROYECTO FORMATIVO \ SOFTWARE BUFFET**

**CENTRO DE FORMACION  
CHIA – CUNDINAMARCA**

**PROGRAMA  
ANALISIS Y DESARROLLO DE SOFTWARE**

**FICHA  
2834909**

**INSTRUCTOR  
LUIS FELIPE LAITON CORTES**

**APRENDIZ  
HAROLD EDISON CRUZ REYES**

**BOGOTA D.C.; 29 DE JUNIO DE 2025**

## **INTRODUCCION**

**En la actualidad, los negocios del sector gastronómico enfrentan el reto de adaptarse a nuevas tecnologías que permitan una mejor administración, optimización de recursos y una experiencia más satisfactoria para el cliente. En particular, los buffets, al ofrecer un modelo de autoservicio con alto volumen de clientes y una rotación rápida de productos, requieren un sistema digital que facilite el control de operaciones, gestión de inventarios, atención al cliente, reservas, facturación y seguimiento de servicios.**

**Este proyecto propone el diseño y desarrollo de un software integral para la gestión de un buffet, que permita automatizar y mejorar los procesos administrativos, operativos y de atención al cliente. La solución está orientada tanto a buffets presenciales como aquellos que ofrecen catering a domicilio o servicios para eventos empresariales.**

# REQUISITOS Y RECOMENDACIONES PARA EL SOFTWARE DE BUFFET

## 1. Instalación en computador (software de escritorio)

### Requisitos y Recomendaciones de Hardware:

- Procesador: Mínimo Intel Core i3 (4ta generación), Recomendado Intel Core i5/i7 o AMD Ryzen 5+
- Memoria RAM: Mínimo 4 GB, Recomendado 8 GB o más
- Disco Duro: Mínimo 250 GB HDD o 128 GB SSD, Recomendado 256 GB SSD o superior
- Pantalla: Mínimo 13" (1366x768), Recomendado 15.6" Full HD (1920x1080)
- Puertos USB: Mínimo 2 puertos USB 2.0/3.0, Recomendado 3+ incluyendo HDMI
- Red: Ethernet y WiFi integrados
- Periféricos: Teclado, mouse, impresora térmica (opcional), escáner de código de barras, UPS

### Requisitos y Recomendaciones de Software:

- Sistema Operativo: Mínimo Windows 10 (64 bits), Recomendado Windows 11 Pro (64 bits)
- Frameworks: Mínimo .NET 4.7+, Java JRE 8+, Recomendado .NET 6, Java 11
- Base de Datos Local: SQLite o MySQL Community, Recomendado MySQL Server o PostgreSQL
- Navegador Web: Chrome / Firefox actualizados
- Antivirus: Microsoft Defender (mínimo), Recomendado ESET, Bitdefender
- Otros: PDF Reader, Office Suite (Microsoft 365 o LibreOffice)

## 2. Instalación en celular android

### Requisitos y Recomendaciones de Hardware:

- Procesador: Mínimo Quad-Core 1.4 GHz, Recomendado Octa-Core 2.0 GHz
- RAM: Mínimo 2 GB, Recomendado 4 GB o más
- Almacenamiento: Mínimo 16 GB libres, Recomendado 64 GB + microSD
- Pantalla: Mínimo 5" 720p, Recomendado 6"+ Full HD
- Batería: Mínimo 3000 mAh, Recomendado 4000 mAh con carga rápida
- Conectividad: 3G/4G + WiFi, Recomendado 4G/5G dual band
- Extras: Cámara, GPS, Bluetooth; Recomendado NFC, lector de huella

### Requisitos y Recomendaciones de Software:

- Sistema Operativo: Mínimo Android 8.0, Recomendado Android 11+
- Espacio libre: Mínimo 100 MB, Recomendado 500 MB

- Permisos: Internet, almacenamiento; opcional ubicación y cámara
- Frameworks: Android SDK, Kotlin/Java; Recomendado Jetpack Compose, Room, Retrofit
- Navegador: Chrome
- Aplicaciones de apoyo: Google Play Services, Google Drive

### 3. Página web

#### Requisitos de Servidor (Hosting):

- Dominio: Propio con SSL
- Hosting: Mínimo 1 GB espacio / 10 GB transferencia, Recomendado VPS o dedicado
- Base de Datos: MySQL 5.7 mínimo, Recomendado MySQL 8 / PostgreSQL
- Lenguaje Backend: PHP 7.4 mínimo, Recomendado PHP 8+, Node.js, Django
- CMS: WordPress (opcional), Laravel, Django
- SSL: Let's Encrypt (mínimo), certificado pago recomendado
- Panel de control: cPanel (mínimo), cPanel + SSH (óptimo)

#### Requisitos de Desarrollo Web:

- Diseño: HTML5, CSS3; Recomendado Bootstrap o Tailwind
- Frontend: JavaScript Vanilla; Recomendado React, Vue.js
- Backend: CRUD básico; Recomendado API RESTful
- Seguridad: Validaciones, autenticación, cifrado, protección CSRF
- Compatibilidad: Navegadores modernos (Chrome, Firefox, Edge, Safari)
- Actualizaciones: Manuales; Recomendado CI/CD con GitHub Actions
- QA: Manual mínimo, recomendado pruebas unitarias y funcionales automatizadas

### 4. Consideraciones generales

- Escalabilidad modular desde el inicio.
- Seguridad integral en todos los módulos.
- Accesibilidad para usuarios sin conocimientos técnicos.
- Soporte y mantenimiento: backups, logs, contacto técnico.

# DESARROLLO DEL SOFTWARE PARA BUFFET

## 1. Metodología de desarrollo de software

✓ Metodología Seleccionada: SCRUM (Ágil)

🔧 Justificación:

SCRUM es una metodología ágil centrada en entregar valor de forma continua, colaborativa e incremental.

En el contexto de un buffet, donde pueden surgir cambios en requerimientos (gestión de menús, reservas, pedidos online, etc.), esta metodología permite adaptarse rápidamente sin comprometer la calidad.

🔧 Características Clave:

- Iterativo e incremental
- Participación del cliente
- Equipos autoorganizados
- Transparencia y mejora continua

🔧 Fases Principales:

- Product Backlog: Lista priorizada de funcionalidades requeridas.
- Sprint Planning: Selección de tareas para el sprint actual.
- Sprint (2-4 semanas): Desarrollo completo de funcionalidades.
- Daily Scrum: Reuniones diarias de sincronización.
- Sprint Review: Presentación al cliente de los avances.
- Sprint Retrospective: Evaluación interna del proceso.

## 2. Frameworks recomendados

🔧 Aplicación Web

Frontend:

- React.js (JavaScript): Interfaces dinámicas y reutilizables.
- Vue.js (JavaScript): Ligero, ideal para MVP.
- Tailwind CSS (CSS): Estilos personalizables y modernos.

Backend:

- Laravel (PHP): Robusto, mantenible, con gran documentación.
- Django (Python): Seguro y rápido.
- Node.js con Express (JavaScript): Eficiente en tiempo real.

Base de Datos:

- MySQL / MariaDB: Confiable, tradicional.
- PostgreSQL: Robusto, excelente para integridad.
- SQLite: Ideal para entornos móviles o desarrollo local.

🚦 Aplicación Móvil

- Flutter (Dart): Multiplataforma (Android/iOS).
- React Native (JavaScript): Reutilización de código web.
- Kotlin (nativo Android): Alto rendimiento.

### 3. Buenas prácticas

🚦 Gestión:

- Documentación constante.
- Comunicación fluida.
- Trazabilidad de requisitos.
- Uso de herramientas colaborativas: Trello, Jira, GitHub.

🚦 Programación:

- Principios SOLID.
- Nomenclatura clara.
- Separación de responsabilidades.
- Control de versiones con Git.

🚦 Pruebas:

- Pruebas unitarias y funcionales.
- Pruebas automatizadas.
- Entorno de pruebas separado.
- Pruebas de carga.

🚦 Seguridad:

- Autenticación robusta (JWT, OAuth).
- Control de acceso por roles.
- Validación de entrada.
- Encriptación de contraseñas (bcrypt).

🚦 Despliegue y Mantenimiento:

- CI/CD (Integración y Despliegue Continuos).
- Backups regulares.
- Monitoreo y logging.
- Arquitectura escalable.