	Aurea Innovaciones Proyecto sistema para “SoFac Bar la 27”	
		Documento de arquitectura del sistema	Fecha: 11 octubre 2025
			Versión: 1.0

Documento de arquitectura del software

Cliente:

SoFac Bar la 27

Desarrollado por:

Equipo de Aurea Innovaciones:

Alexander Rincón Rincón

Daniel Manrique Ruiz

Camilo Ayala Arévalo

Versión 1.0

Aurea Innovaciones,

Bogotá DC, 11 de octubre de 2025





		Aurea Innovaciones Proyecto sistema para “SoFac Bar la 27”		
		Documento de arquitectura del sistema	Fecha:	11 octubre 2025
			Versión:	1.0

Tabla de contenido

Introducción	3
Visión general del sistema	3
Objetivos principales:	3
Arquitectura general.....	4
Capa de Presentación (Frontend)	4
Capa Lógica (Backend).....	4
Capa de Datos (Base de Datos).....	4
Patrón de diseño (MVC)	4
Diagrama de arquitectura	5
Descripción diagrama de arquitectura	5
Diagrama de despliegue	6
Descripción diagrama de despliegue	6
Diagrama de componentes	7
Descripción diagrama de componentes	7
Modelo entidad relación	8
Buenas prácticas de desarrollo.....	9

		Aurea Innovaciones Proyecto sistema para “SoFac Bar la 27”	
		Documento de arquitectura del sistema	Fecha: 11 octubre 2025
			Versión: 1.0

Introducción

Este documento presenta la arquitectura del sistema desarrollado para SoFac Bar la 27 un bar que requiere un sistema completo que funcione para los colaboradores. Esto como parte de la iniciativa de modernización tecnológica liderada por Áurea Innovaciones. Su propósito es ofrecer una visión integral de los componentes, la estructura y las decisiones técnicas que definen la solución, con un enfoque claro, formal y orientado a presentación.

El cliente requiere una plataforma centralizada para gestionar usuarios, productos, inventarios, mesas, pedidos y pagos en múltiples sedes. El sistema debe ser seguro, escalable y con alta usabilidad. Aurea Innovaciones proveerá diseño, desarrollo, implementación e infraestructura, además de soporte y acompañamiento.

Requisitos clave del sistema:



- Multi-sede con inventario centralizado.
- Gestión de pedidos por mesa y estado de mesas en tiempo real.
- Procesamiento de pagos (un método por transacción).
- Roles: Administrador, Cajero, Mesero.
- Seguridad: login robusto, sesiones de 3 minutos de expiración, una sesión por usuario.
- Exportes en CSV/XLSX y reportes rápidos.

Visión general del sistema

El sistema para SoFac Bar la 27 es una plataforma web integral que centraliza la gestión operativa del bar, incluyendo control de inventario, ventas, pedidos y reportes administrativos. Su diseño modular permite una administración eficiente y escalable, con una interfaz moderna con modo claro y oscuro. Y soporte para autenticación por roles (administrador, mesero, y cajero).

Objetivos principales:

- Digitalizar los procesos de operación y gestión interna del establecimiento.
- Mejorar la trazabilidad de los pedidos y el control de inventarios.
- Proveer información sobre ventas y desempeño del bar.
- Ofrecer una experiencia de usuario fluida y visualmente coherente con la identidad del bar.

		Aurea Innovaciones Proyecto sistema para “SoFac Bar la 27”	
		Documento de arquitectura del sistema	Fecha: 11 octubre 2025
			Versión: 1.0

Arquitectura general

Capa de Presentación (Frontend)

Tecnologías: HTML, CSS, JavaScript.

Características: Interfaz adaptable (responsive), modo oscuro, autenticación y paneles de control diferenciados por rol (dashboard).

Capa Lógica (Backend)

Lenguajes: php, JavaScript.

Servicios: Manejo de sesiones, lógica de negocio, control de inventarios, pedidos, reservas y reportes.

Capa de Datos (Base de Datos)

Sistema: MySQL.

Diseño: Modelo relacional optimizado con claves foráneas y relaciones N:N controladas mediante tablas intermedias.

Patrón de diseño (MVC)

El sistema adopta el patrón Modelo–Vista–Controlador (MVC) dentro del backend, con el fin de garantizar una separación clara de responsabilidades y facilitar la mantenibilidad del código.

Modelo: Gestiona la lógica de datos, las entidades y la comunicación con la base de datos.

Vista: Corresponde a las respuestas generadas por la API de php-MySQL y entregadas al cliente (frontend).

Controlador: Coordina la interacción entre vistas y modelos, procesando las peticiones y aplicando las reglas de negocio.



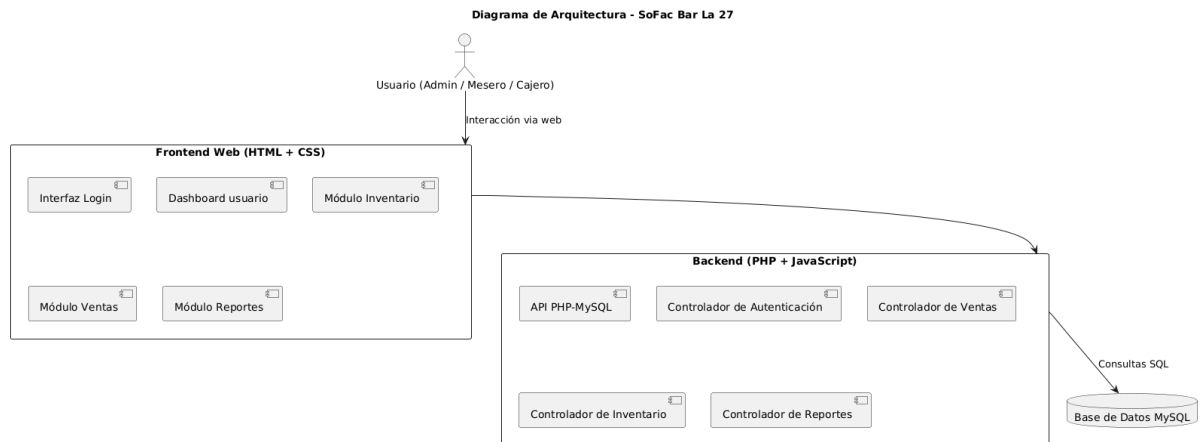
		Aurea Innovaciones Proyecto sistema para “SoFac Bar la 27”	
		Documento de arquitectura del sistema	Fecha: 11 octubre 2025
			Versión: 1.0

Diagrama de arquitectura



Descripción diagrama de arquitectura

Este diagrama muestra la arquitectura lógica del sistema:

- Los usuarios interactúan con la interfaz web (frontend) desarrollada en CSS/JS, que se comunica con el backend mediante consultas PHP-SQL.
- El backend (PHP/JS) gestiona la lógica del negocio a través de varios controladores.
- Toda la información se almacena y consulta en una base de datos MySQL, alojada en localhost (XAMPP).
- Esta estructura facilita la separación de responsabilidades, escalabilidad y mantenimiento.



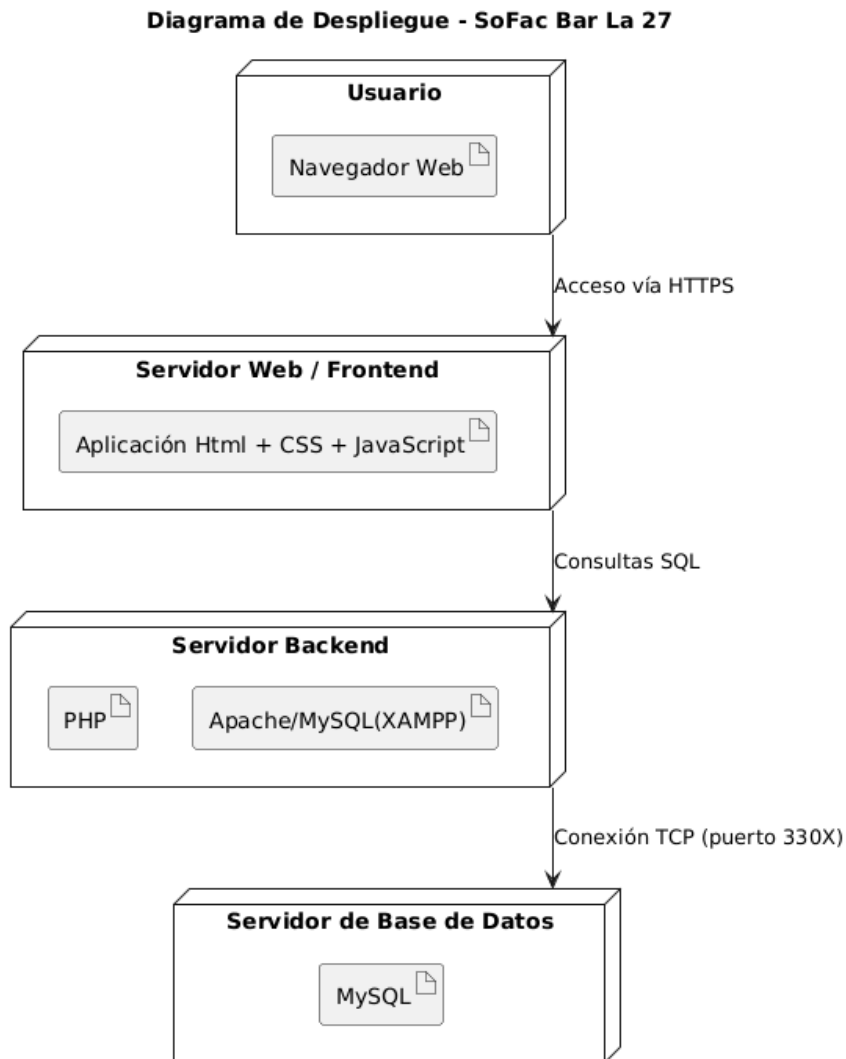
		Aurea Innovaciones Proyecto sistema para “SoFac Bar la 27”	
		Documento de arquitectura del sistema	Fecha: 11 octubre 2025
			Versión: 1.0

Diagrama de despliegue



Descripción diagrama de despliegue

Este diagrama representa cómo se despliega físicamente el sistema:

- Los usuarios acceden a la aplicación desde un navegador web.
- El frontend funciona con HTML, CSS y JS.
- El backend php corre en un servidor alojado en Apache, gestionado por XAMPP para asegurar la disponibilidad en el equipo.
- La base de datos MySQL se encuentra en el servidor MySQL de XAMPP, normalmente asignado en el puerto 3306 o 3307.



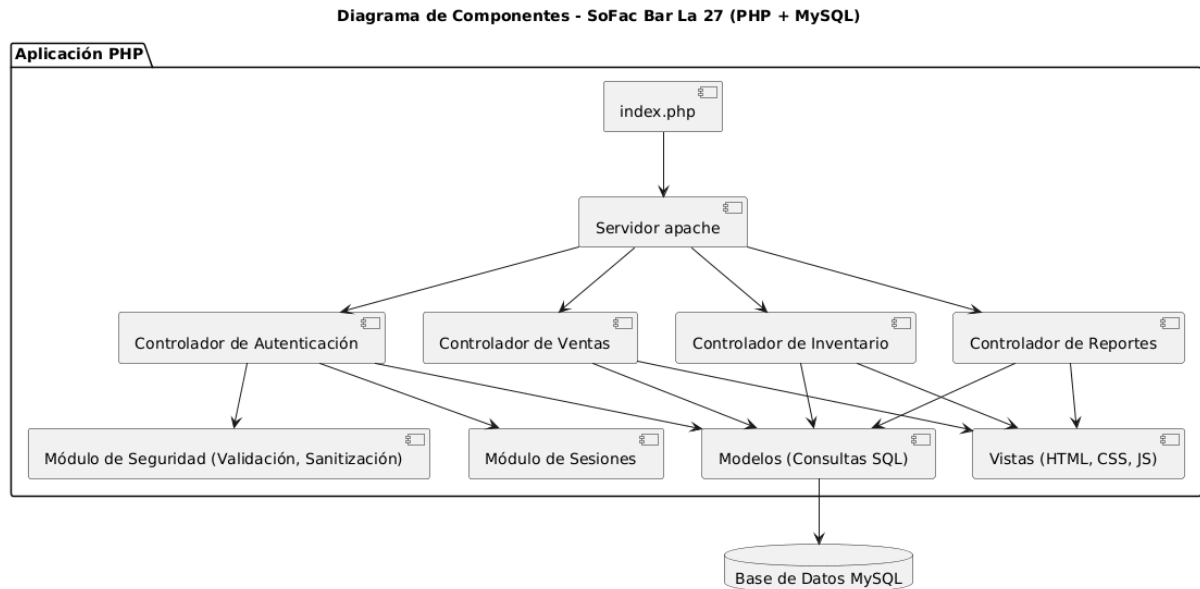
		Aurea Innovaciones Proyecto sistema para “SoFac Bar la 27”	
		Documento de arquitectura del sistema	Fecha: 11 octubre 2025
			Versión: 1.0



Diagrama de componentes



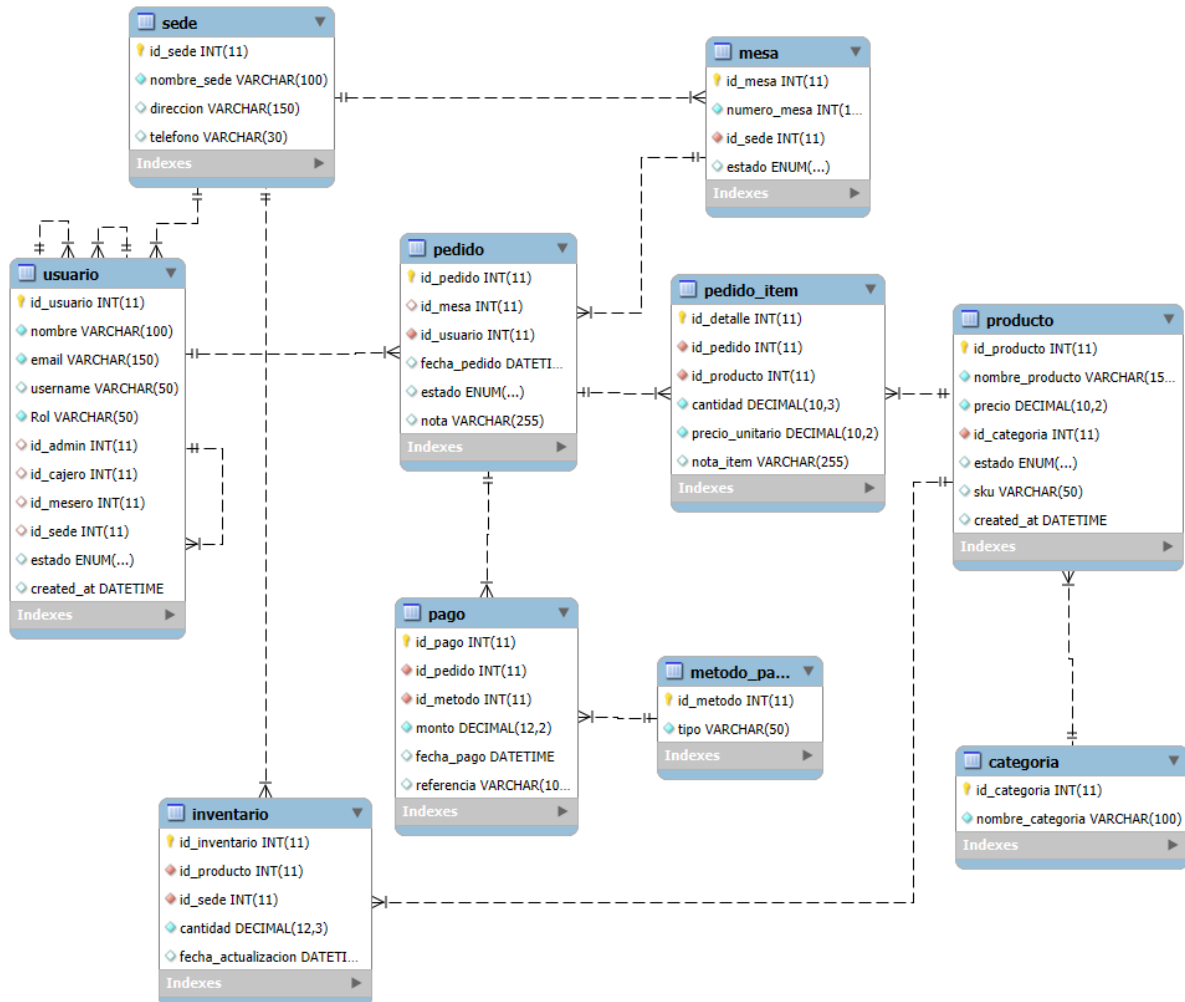
Descripción diagrama de componentes



El diagrama de componentes detalla cómo se organiza el sistema internamente:

- Cada módulo del frontend (login, ventas, inventario, reportes) se comunica con su correspondiente controlador en el backend.
- El backend se estructura en controladores, lo que favorece una arquitectura limpia y modular.
- La capa de seguridad gestiona la autenticación.
- La capa de datos se maneja a través de un servicio dedicado de conexión a MySQL, centralizando el acceso a la base de datos.

		Aurea Innovaciones Proyecto sistema para “SoFac Bar la 27”		
		Documento de arquitectura del sistema	Fecha:	11 octubre 2025
			Versión:	1.0

Modelo entidad relación



		Aurea Innovaciones Proyecto sistema para “SoFac Bar la 27”	
		Documento de arquitectura del sistema	Fecha: 11 octubre 2025
			Versión: 1.0

Buenas prácticas de desarrollo

Selección de Tecnologías:

Elegir tecnologías adecuadas a los requisitos del sistema, asegurando escalabilidad, rendimiento y fácil mantenimiento. Implementar CSS y JS para el frontend. Y JS y PHP para el backend, según la compatibilidad con los módulos de inventario, pedidos y pagos.

Control de Versiones de Código:

Implementar Git como sistema de control de versiones, permitiendo un seguimiento detallado de los cambios y facilitando la colaboración entre desarrolladores. Definir ramas específicas para nuevas funcionalidades (feature), corrección de errores (fix) y versiones estables (main o release).

Desarrollo de Componentes Reutilizables:

Diseñar componentes y módulos reutilizables que puedan implementarse en distintas partes del sistema. Por ejemplo, un módulo de gestión de inventarios adaptable en frontend y backend sin duplicar código. Esto mejora la eficiencia, la mantenibilidad y reduce el tiempo de desarrollo.

Calidad del Código:

Mantener altos estándares de calidad mediante revisiones de código (code reviews) y análisis automatizados con herramientas como Seguir principios de Clean Code y SOLID para garantizar legibilidad, mantenibilidad y extensibilidad del código.

Pruebas Unitarias y de Integración:

- Implementar pruebas unitarias para validar el correcto funcionamiento de los módulos individuales (por ejemplo, gestión de pedidos o cierre de cuentas)
- Ejecutar pruebas de integración que verifiquen la correcta interacción entre módulos, como inventario, facturación y pagos.