



# Trabajo Final Integrador

CARRERA: Ingeniería en Sistemas de Información

MATERIA: Paradigmas y Lenguajes de la Programación III

COMISIÓN: "U" (única) "A"

PROFESOR: Mgter. Ing. Agustín Encina

ESTUDIANTES: Kiara Arguello

FECHA: 18-11-2025

## **Documento de Documentación del Proyecto Integrador: Alondra – Sistema Web de Gestión para Múltiples Hoteles**

### **1. Introducción**

El presente documento detalla la planificación y las características del sistema Alondra – Sistema Web de Gestión para Múltiples Hoteles, un proyecto desarrollado para integrar los conocimientos adquiridos en el transcurso de la materia. Alondra es concebido como una aplicación web, diseñada para gestionar las operaciones diarias de múltiples establecimientos hoteleros desde una única plataforma. El proyecto se enfoca en proporcionar una solución eficiente para el control de la información, manteniendo la separación lógica de los datos por cada hotel.

El sistema se estructura en dos áreas principales: un sitio público que funciona como presentación del servicio y portal de acceso, y un panel de gestión privado (Back-Office) destinado a los usuarios internos (personal y dueños) para la administración de las operaciones.

Objetivo General Logrado:

Se construyó un sistema web funcional en su interfaz y arquitectura base, sentando las fundaciones para la operatividad multi-hotel, la seguridad en el acceso y la gestión de reservas mediante el diseño de la interfaz y la implementación de la lógica de autenticación.

### **2. Arquitectura y Plataforma Tecnológica**

El desarrollo de Alondra se llevó a cabo utilizando un modelo de arquitectura tradicional, donde el Frontend se desacopla lógicamente del Backend y la Base de Datos, permitiendo la distribución de la carga de trabajo y una clara modularidad del código.

#### **2.1. Componentes Tecnológicos**

<b>Componente</b>	<b>Tecnología Principal</b>	<b>Descripción de Implementación</b>
<b>Frontend</b>	HTML, CSS (SASS), JavaScript	Estructura de vistas, con diseño enfocado en la usabilidad. Se utilizó SASS para una gestión de estilos modular, coherente y fácil de mantener (variables, mixins, anidación), asegurando un diseño <i>responsive</i> para distintos dispositivos.

<b>Backend</b>	PHP	Lógica de negocio, manejo de sesiones, control de acceso y comunicación con la base de datos. Se implementaron archivos PHP dedicados a la conexión y a los procesos de autenticación y visualización inicial del panel.
<b>Base de Datos</b>	MySQL	El proyecto se diseñó para operar sobre MySQL, utilizando la extensión <b>PDO</b> de PHP para establecer la conexión y realizar las operaciones de consulta de manera segura y estandarizada, gestionando las excepciones de errores de conexión.

## 2.2. Fundamentos de Seguridad y Acceso

La seguridad básica del sistema se abordó mediante la implementación de un módulo de Autenticación (Login) y la definición de una estructura de control de sesiones.

- Gestión de Sesiones y Acceso: El archivo `dashboard.php` inicia la sesión y ejecuta una validación estricta que verifica la existencia de un identificador de usuario (`usuario_id`) en la sesión. Si el usuario no está autenticado, se redirige a la página de `login`, estableciendo la barrera inicial para el panel de gestión.
- Hash de Contraseñas: Se implementó el estándar de seguridad de almacenar y verificar las contraseñas mediante hash. El archivo `login.php` utiliza la función nativa de PHP `password_verify()` para comparar la contraseña ingresada por el usuario con el `hash` almacenado en la base de datos (`password_hash`), cumpliendo con el requisito de seguridad.

## 2.3. Estructura Multi-Hotel

La arquitectura de datos de Alondra se concibe desde el inicio para operar con la lógica multi-hotel. Esto se consigue al diseñar las consultas SQL que unen las tablas de reservas, usuarios, y habitaciones con la tabla de hoteles. En la lista de reservas inicial del `dashboard` (presente en `dashboard.php`), las uniones (`JOIN`) a la tabla `hoteles` aseguran que la información se puede filtrar y presentar basándose en el identificador único del hotel (`hotel_id`), lo que es fundamental para la separación lógica de la información entre distintos establecimientos en el mismo sistema.

## 3. Módulos Implementados: Front-Office (Sitio Público)

El sitio público fue desarrollado con el objetivo de funcionar como una carta de presentación del servicio Alondra y proporcionar los puntos de acceso al sistema privado.

### 3.1. Landing y Navegación

El archivo principal (`index.html`) organiza la información en secciones claras y accesibles:

- **Landing:** Presentación inicial con el nombre y la propuesta de valor del sistema (Gestor de Hoteles).
- **About:** Descripción breve sobre el rol de la empresa Alondra como proveedora del servicio.
- **Beneficios y Planes:** Listados informativos que detallan las ventajas de utilizar el sistema y las estructuras de planes disponibles (datos a modo de demostración).
- **Navegación:** La barra de navegación superior permite la transición directa entre todas las secciones públicas y cuenta con un acceso visible al módulo de Login ([login.php](#)).

### **3.2. Formulario de Contacto**

El módulo de contacto ([contactanos.html](#)) incluye un formulario detallado para que potenciales clientes hoteleros puedan solicitar información. El formulario incluye campos para la recolección de datos clave, como Nombre, Apellido, Nombre del Hotel, Dirección, Correo Electrónico, y un campo de Consulta extendida. El diseño de este formulario es robusto en su estructura HTML5 y está preparado para integrarse con una validación dinámica en el Frontend.

### **3.3. Interfaz de Acceso**

La interfaz de login se presenta con un diseño limpio y funcional. Es la puerta de entrada que exige la autenticación mediante correo electrónico y contraseña, y está directamente vinculada con la lógica de autenticación de PHP para el control de acceso.

## **4. Módulos Implementados: Back-Office (Panel de Gestión)**

El Back-Office, accesible a través del *dashboard*, es el núcleo del sistema de gestión, diseñado para la operativa diaria y la visualización rápida de la información.

### **4.1. Estructura del Panel (Dashboard)**

El panel de control ([dashboard.html](#) y [dashboard.php](#)) está organizado para ofrecer una visión de alto nivel de las operaciones:

- **Barra Superior:** Contiene la marca del sistema y un área de acciones, que incluye un **Buscador** unificado para reservas, huéspedes o habitaciones, y un botón de acción rápida "+ Nueva reserva".
- **Indicadores:** Sección dedicada a la visualización de métricas principales (KPIs), como ocupación actual o ingresos, esenciales para la toma de decisiones.
- **Filtros de Datos:** Incluye controles para filtrar la información de la tabla principal por **Fecha** y por **Hotel**, reforzando la capacidad multi-hotel de la aplicación.
- **Resumen por Hotel:** Una sección dedicada a mostrar datos de resumen agregados por cada establecimiento, alineándose con el objetivo de gestión de múltiples sucursales.

### **4.2. Módulo de Reservas: Visualización y Listado**

La funcionalidad central implementada es la **visualización de las reservas**.

- **Consulta de Datos:** Contiene la consulta SQL modelo para obtener las reservas más recientes. Esta consulta es un ejemplo funcional de cómo se extraería la información relevante, incluyendo el código de la reserva, el nombre del huésped, el hotel asociado, el número de habitación, el régimen de alojamiento, las fechas de *check-in* y *check-out*, el estado de la reserva y el monto total.
- **Tabla de Resultados:** La información obtenida se presenta en una tabla dinámica que lista las reservas. Los campos como el estado y el total se formatean para una lectura clara, facilitando la identificación rápida de la información operativa.

#### **4.3. Base para la Gestión de Reservas (CRUD)**

Aunque se presenta la funcionalidad de lectura (Listado), la interfaz para la creación y edición de reservas está diseñada y estructurada en un Modal de Nueva Reserva en el dashboard. Esta interfaz incluye los campos necesarios para:

- Selección del **Hotel** (clave para la separación de datos).
- Identificación de la **Habitación**.
- Definición del **Régimen** de alojamiento (BB, HB, AI).
- Fechas de **Check-in** y **Check-out**.
- **Monto Total** de la reserva.

La presencia de los campos de fecha de entrada y salida establece el requisito de concurrencia simplificada, donde la lógica posterior en PHP se encargará de realizar las consultas para verificar la disponibilidad en el rango de fechas solicitado, evitando la sobreventa antes de la confirmación de la reserva.

#### **5. Cumplimiento de Requisitos Funcionales y No Funcionales**

A continuación, se presenta un resumen de los requisitos establecidos en la planificación y cómo fueron abordados en la implementación:

##### **Requisitos Funcionales**

Requisito Funcional	Estado de Implementación	Observación
<b>Landing pública</b>	Implementada	Vistas <code>index.html</code> y <code>contactanos.html</code> con estructura de navegación completa (Landing, About, Beneficios, Planes).

<b>Registro/login</b>	Implementada	Interfaz de login ( <a href="#">login.html</a> y <a href="#">login.php</a> ) con lógica de validación de credenciales y control de sesión inicial en PHP.
<b>Gestión de hoteles</b>	Sentada la base	El diseño de base de datos y la consulta de <i>dashboard</i> ( <a href="#">dashboard.php</a> ) incluyen la referencia a <a href="#">hotel_id</a> , lo que establece el marco para la separación de datos.
<b>Reservas con control de disponibilidad</b>	Sentada la base (Interfaz)	El modal "Nueva Reserva" en <a href="#">dashboard.html</a> define los campos de entrada para las fechas ( <i>check-in</i> y <i>check-out</i> ), que son la base para la posterior validación de concurrencia.
<b>CRUD de Habitaciones</b>	Sentada la base (Diseño de Datos)	La inclusión de <a href="#">habitaciones</a> en la consulta del <i>dashboard</i> indica que la tabla ha sido integrada en la estructura de datos, permitiendo asociar reservas a un número de habitación específico.

#### *Requisitos No Funcionales*

Requisito No Funcional	Estado de Implementación	Observación
<b>Multiusuario (Separación por <a href="#">hotel_id</a>)</b>	Sentada la base	El <i>dashboard</i> en PHP implementa un <i>JOIN</i> con la tabla de <a href="#">hoteles</a> para listar las reservas, confirmando que la estructura de datos mantiene el identificador de hotel en las transacciones clave.
<b>Seguridad: Hash de contraseñas</b>	Implementada	Uso de <a href="#">password_verify</a> en el módulo <a href="#">login.php</a> para la verificación segura de credenciales.

<b>Arquitectura</b>	Implementada	La estructura de archivos separa claramente Frontend (HTML/CSS/Javascript) y Backend (PHP/Control de Sesiones), utilizando PDO para la abstracción de la base de datos (MySQL).
---------------------	--------------	---

## 6. Conclusión

El proyecto **Alondra** ha establecido una base sólida y funcional que cumple con los fundamentos arquitectónicos y de seguridad requeridos. Se ha diseñado un Front-Office informativo y un Back-Office con una estructura de panel de control profesional, lista para la gestión de datos. La implementación del módulo de **Autenticación** y el uso de **PDO** en la capa de persistencia demuestran la aplicación de buenas prácticas en el manejo de sesiones y la comunicación con la Base de Datos.

La elección de una arquitectura que contempla la separación de datos por `hotel_id` desde las primeras consultas sienta las bases necesarias para escalar el sistema a un entorno multi-hotel completo. La interfaz del *dashboard*, con sus filtros y el modal de creación de reservas, provee un *layout* eficiente para la futura integración de las funcionalidades completas de **CRUD** y **control de concurrencia** de manera lógica. El resultado es una aplicación con un diseño *responsive* y un código estructurado, que proporciona la plataforma necesaria para la gestión hotelera avanzada.