

TG

# HOTEL LUAR



MANUAL  
TECNICO

# ÍNDICE

- 01 TIPO DE APLICACION
- 02 PUERTOS DE USO
- 03 CODIGO HTML
- 04 CODIGO CSS
- 05 CODIGO JS



# TIPO DE APLICACION



---

**EL PROPÓSITO DE ESTE MANUAL TÉCNICO ES PROPORCIONAR UNA GUÍA DETALLADA PARA FUTUROS PROGRAMADORES INTERESADOS EN COMPRENDER, DESARROLLAR Y MANTENER LA APLICACIÓN DE GESTIÓN HOTELERA BASADA EN NODE.JS. ESTE DOCUMENTO ESTÁ DISEÑADO PARA FACILITAR LA INCORPORACIÓN DE NUEVOS DESARROLLADORES AL PROYECTO, OFRECIENDO INFORMACIÓN CLARA Y CONCISA SOBRE LA ARQUITECTURA, LAS FUNCIONALIDADES Y LOS PROCEDIMIENTOS NECESARIOS PARA TRABAJAR CON LA APLICACIÓN.**

La aplicación de gestión hotelera es una solución integral diseñada para optimizar y facilitar la administración de hoteles. Desarrollada sobre la plataforma Node.js, esta aplicación permite a los administradores y al personal del hotel gestionar eficientemente las operaciones diarias, mejorando la experiencia tanto para el cliente como para el personal.

- Permite a los usuarios realizar reservas en tiempo real, consultar disponibilidad de habitaciones y gestionar cambios o cancelaciones.
- Registro y seguimiento de diferentes tipos de habitaciones (sencillas, dobles, suites).
- Control de disponibilidad y estado de las habitaciones.
- Conexión a una base de datos que gestiona los usuarios de nuestros clientes
- Análisis de datos por medio del apartado de comentarios que los clientes pueden hacer para ayudar en la toma de decisiones estratégicas
- Panel de control intuitivo para el personal del hotel, que permite gestionar todos los aspectos operativos desde un solo lugar.
- Funciones de autenticación y autorización para garantizar la seguridad de los datos.
- Sistema para procesar pagos en línea, incluyendo opciones para tarjetas de crédito y otros métodos de pago.

# PUERTOS DE USO

## EXPLICACION DE LOS PUERTOS QUE UTILIZAMOS

El puerto 27017 es el puerto predeterminado utilizado por MongoDB para las conexiones de base de datos. Este puerto permite que las aplicaciones se conecten al servidor MongoDB para realizar operaciones de lectura y escritura en las bases de datos. Al igual que con otros servicios, es crucial asegurarse de que el puerto 27017 esté abierto en el firewall del servidor donde se ejecuta MongoDB, especialmente en entornos de producción, para permitir que las aplicaciones accedan a la base de datos de forma segura. También es recomendable implementar medidas de seguridad, como autenticación y cifrado, para proteger los datos y las conexiones, dado que este puerto puede ser un objetivo para ataques si no está adecuadamente asegurado. Además, en entornos de producción, es común vincular MongoDB a una dirección IP específica y no a la dirección 0.0.0.0 para limitar el acceso solo a las aplicaciones necesarias.

Información del Puerto 3000

El puerto 3000 es un puerto comúnmente utilizado para aplicaciones de desarrollo en Node.js y Express. Al iniciar tu servidor con `app.listen(3000)`, le indicas a tu aplicación que escuche las solicitudes HTTP en este puerto. Esto permite a los desarrolladores acceder a la aplicación a través de `http://localhost:3000`, facilitando pruebas rápidas y el desarrollo iterativo. Elegir el puerto 3000 es ventajoso porque evita conflictos con puertos más comunes, como el 80 (HTTP) y el 443 (HTTPS), que son utilizados por servidores web en producción. Además, el uso de puertos alternativos como el 3000 permite que múltiples aplicaciones se ejecuten simultáneamente en el mismo servidor, lo que es ideal para entornos de desarrollo y pruebas. Al desplegar en producción, es recomendable configurar un servidor proxy inverso (como Nginx o Apache) para redirigir el tráfico desde los puertos estándar hacia tu aplicación en el puerto 3000.





# CODIGO HTML

## PUNTOS IMPORTANTES DEL CODIGO HTML DE NUESTRA APP

---



- Se especifica el conjunto de caracteres mediante `<meta charset="UTF-8">`, lo que asegura que los caracteres se muestren correctamente. También se incluye `<meta name="viewport" content="width=device-width, initial-scale=1.0">`, que permite que el diseño sea responsivo en dispositivos móviles.
- Se utilizan fuentes de Google Fonts para mejorar la tipografía, así como archivos CSS para aplicar estilos. Esto permite personalizar la apariencia del sitio web y mejorar la experiencia del usuario.
- los menús de navegación son claros y accesibles, con enlaces a diferentes secciones del sitio. Esto es crucial para la experiencia del usuario, ya que permite una fácil navegación entre las páginas.
- Se utilizan imágenes representativas de las habitaciones en el primer código, lo que ayuda a los usuarios a visualizar el contenido. En el código, se incluye un `<iframe>` que integra un mapa de Google, proporcionando una representación visual de la ubicación del hotel.
- incluye un párrafo que describe la ubicación del hotel, lo que es útil para los visitantes que buscan información sobre cómo llegar.
- Ambos códigos siguen la estructura básica de un documento HTML, que incluye las secciones `<head>` y `<body>`. Esta estructura es esencial para asegurar que los navegadores interpreten correctamente el contenido.

# CODIGO CSS



- 
- Ambos archivos CSS utilizan una estructura clara y organizada, aplicando estilos globales y específicos a diferentes elementos. Se utiliza un enfoque de diseño responsivo, lo que es esencial para la usabilidad en dispositivos móviles y de escritorio.
  - Se establece un estilo base utilizando el selector universal (\*), que elimina márgenes y rellenos predeterminados y aplica box-sizing: border-box, asegurando que el ancho y alto de los elementos incluyan el relleno y el borde.
- 
- El encabezado está diseñado con display: flex, lo que permite una disposición flexible de los elementos. Se utilizan propiedades de alineación para centrar el contenido y separar el logotipo del menú de navegación.
  - Los menús de navegación utilizan listas desordenadas (ul) y se ocultan o muestran según el tamaño de la pantalla mediante media queries. Los enlaces cambian de color al pasar el ratón, mejorando la interactividad.
- 
- Se implementan media queries para ajustar el diseño en diferentes tamaños de pantalla (tabletas y móviles). Esto incluye cambios en la disposición de los menús y el tamaño de los elementos, asegurando que la página sea accesible en cualquier dispositivo.
  - Se incluyen estilos para un modal y un carrusel, que son elementos interactivos. Los modales están ocultos por defecto y se muestran según la interacción del usuario, mientras que el carrusel utiliza transiciones suaves para mejorar la experiencia visual.

# CODIGO JS

## BREVE DESCRIPCION DE LAS FUNCIONES QUE POSEE MI CODIGO JS



### JS PRINCIPAL

- El código utiliza Express.js para crear un servidor web, lo que permite manejar rutas y servir archivos estáticos de manera eficiente. Se establece una conexión con una base de datos MongoDB para almacenar información de usuarios.
- Se importan los módulos necesarios: express para el servidor, path para manejar rutas de archivos, y mongoose para interactuar con MongoDB.



### RUTA DE REGISTRO

- Ruta de Registro (/api/register): Maneja las solicitudes POST para registrar nuevos usuarios. Recibe datos en formato JSON, crea una nueva instancia del modelo User, y guarda el usuario en la base de datos. Responde con un mensaje de éxito o error según corresponda.
- Ruta Principal (/): Envía el archivo Pantalla1.html como respuesta cuando se accede a la raíz del servidor.



### ERRORES Y SERVIDOR

- Se inicia el servidor y se imprime un mensaje en la consola indicando que está corriendo en el puerto especificado.
- Se implementa manejo de errores en las rutas de registro e inicio de sesión. En caso de error, se devuelve un estado 500 con un mensaje descriptivo.

---

## CONTÁCTANOS

Email: [soporte@luarluzdeluna.com](mailto:soporte@luarluzdeluna.com) ☐  
Teléfono: 7333-0643