**REQUISITOS FUNCIONALES**

**Requerimientos:**

* El sistema debe permitir al usuario realizar la búsqueda de diferentes destinos, lugares u hoteles para encontrar opciones que se ajusten a sus preferencias y necesidades específicas.
* El sistema debe permitir al usuario ingresar la fecha de entrada y salida para para determinar la disponibilidad y reservar el alojamiento en los destinos seleccionados.
* El sistema debe permitir al usuario ingresar la cantidad de adultos y niños que se hospedarán, para ajustar la disponibilidad y opciones de alojamiento según el número de personas.
* El sistema debe permitir al usuario especificar el número de habitaciones necesarias, para asegurar que el alojamiento pueda acomodar a todos los huéspedes.
* El sistema debe permitir al usuario realizar un filtro por hoteles, para encontrar opciones que se ajusten a sus preferencias y necesidades específicas.
* El sistema debe permitir al usuario la visualización de las actividades turísticas o los servicios adicionales con los que cuenten los diferentes destinos, lugares u hoteles, para que el usuario tenga la mayor información posible
* El sistema debe permitir al usuario realizar búsquedas por ciudad, país etc. Para encontrar opciones que se ajusten a sus preferencias y necesidades específicas.
* El sistema debe permitir al usuario filtrar diferentes destinos, lugares u hoteles por precio para ayudarle a encontrar opciones que se ajusten a su presupuesto.
* El sistema debe permitir al usuario filtrar diferentes destinos, lugares u hoteles por valoración para encontrar opciones que se ajusten a sus preferencias.
* El sistema debe permitir al usuario filtrar diferentes destinos, lugares u hoteles por tipos de alojamiento para que pueda elegir el tipo de estancia que mejor se adapte a sus expectativas.
* El sistema de permitir al usuario filtrar diferentes destinos, lugares u hoteles por género, para encontrar opciones que se ajusten a sus preferencias y necesidades específicas.
* El sistema debe permitir al usuario filtrar diferentes destinos, lugares u hoteles por orientación sexual, para garantizar que el alojamiento elegido sea inclusivo y respetuoso con sus identidades y preferencias.
* El sistema debe permitir al usuario, al momento de escoger un destino, lugar u hotel, la visualización de los detalles de estos, para que pueda tomar una decisión informada basada en la información completa sobre cada opción.
* El sistema debe permitir al usuario ver reseñas y calificaciones de otros huéspedes, para evaluar la calidad del destino, lugar o hotel.
* El sistema debe permitir al usuario comparar precios de diferentes proveedores para el mismo destino, lugar o hotel, para asegurar la mejor oferta disponible.
* El sistema debe permitir al usuario visualizar un listado de actividades turísticas disponibles en el destilo elegido.
* El sistema debe permitir al usuario compartir sus destinos, lugares o hoteles favoritos en redes sociales, para facilitar la recomendación y la planificación colaborativa con amigos y familiares.
* El sistema debe permitir al usuario realizar reservas directamente a través de la plataforma, para simplificar el proceso de reserva y confirmar el alojamiento al instante.
* El sistema debe permitir al usuario visualizar las reservas realizadas, para tener toda la información a su alcance.
* El sistema debe permitir al usuario editar las reservas realizadas, para ajustar detalles como fechas, número de huéspedes o tipo de habitación, en caso de cambios en sus planes.
* El sistema debe permitir al usuario eliminar las reservas realizadas, para cancelar la reserva en caso de cambios en sus planes o si ya no es necesario el alojamiento.
* El sistema debe permitir al usuario el registro en la plataforma, para acceder a todas sus características.
* El sistema debe permitir al usuario iniciar sesión para que este pueda ingresar a la plataforma y hacer uso de ella.
* El sistema debe proporcionar un mapa interactivo que muestre la ubicación de destinos, lugares u hoteles, para facilitar la planificación de viaje y la evaluación de puntos de interés.
* El sistema de permitir al usuario tener una sección de favoritos para guardar hoteles, destinos y actividades favoritas.

**REQUISITOS NO FUNCIONALES**

Aquí detallaremos las cualidades, restricciones, características que no determinan una funcionalidad del sistema a desarrollar pero que lo mantendrán en funcionamiento.

**ATRIBUTOS DE CALIDAD**

**Confiabilidad**

* El sistema debe ser capaz de recuperarse automáticamente de fallos no críticos sin afectar las operaciones normales como, por ejemplo: errores de validación de datos, problemas de conexión temporales, caída de servicios secundarios, problemas de carga leve.
* El usuario no puede experimentar más de 3 fallas mensuales en la aplicación.
* El promedio de duración de fallas no críticas no podrá ser mayor a 15 minutos.

**Disponibilidad**

* El sistema debe estar disponible continuamente, es decir con un nivel de disponibilidad para los usuarios 7 días por 24 horas.
* El sistema debe apuntar a un tiempo de actividad del 99.9999% durante las horas de operación normales.
* El sistema estará en la capacidad de manejar un aumento del 20% en la carga de trabajo sin degradación significativa del rendimiento.
* Se programarán ventanas bimestrales específicas de 2 horas, para llevar a cabo actividades de mantenimiento planificado en el sistema.

**Seguridad:**

* La longitud de las claves de la aplicación debe ser de mínimo 8 caracteres y debe incluir símbolos, al menos una mayúscula y al menos un número.
* Toda la comunicación entre el cliente y el servidor estará cifrada utilizando protocolos seguros como TLS/SSL.

**Infraestructura**

* Los servidores tendrán un rendimiento que garantice tiempos de respuesta rápidos para las solicitudes del usuario.
* Se garantizará un ancho de banda adecuado para satisfacer las necesidades de tráfico actual y futuro.
* Se realizarán pruebas de rendimiento para evaluar la capacidad de la red y se realizarán mejoras según sea necesario.
* Se implementarán medidas para minimizar la latencia y mantenerla dentro de los límites aceptables, como el uso de servidores en ubicaciones geográficas estratégicas.
* Se establecerán políticas de acceso en los cortafuegos para permitir el tráfico necesario y bloquear cualquier actividad no autorizada.

**Mantenibilidad**

* El tiempo promedio para reparar un error debe no ser mayor a 8 horas.
* El código fuente seguirá estándares de codificación claros y estar bien documentado para facilitar su mantenimiento.
* El sistema contara con documentación que le permitirá al personal encargado del mantenimiento informarse sobre cómo está constituido el sistema, para que pueda solucionar cualquier problema o fallo eventual que se pueda presentar.
* Se usarán metodologías agiles que proporcionaran prácticas y principios que facilitaran la evolución continua del sistema y la gestión efectiva del mantenimiento a lo largo del tiempo.

**Portabilidad**

* El sistema será independiente de navegadores, permitiendo su ejecución en diferentes Exploradores web como Chrome, Firefox o Edge.
* La interfaz de usuario será adaptable a diversas resoluciones de pantalla, incluyendo dispositivos móviles y tabletas.
* Se realizarán pruebas de portabilidad regulares para garantizar que el sistema funcione de manera correcta en diferentes entornos.

**RESTRICCIONES**

**Exactitud**

* El sistema deberá actualizar la disponibilidad de habitaciones con una precisión del 99.5% después de cada reserva o cancelación, asegurando la información sobre la disponibilidad.
* El sistema deberá registrar y mostrar los detalles de las reservas con una exactitud del 99% en la información relacionada con fechas, tipo de habitación, número de huéspedes y monto pagado.
* Las tarifas mostradas en el sistema deben ser precisas en un 99.5%, incluyendo detalles como impuestos aplicados, y costos adicionales, para garantizar que el usuario reciba la información correcta sobre el costo total de su estadía.

**Diseño**

* **Responsive Web Design:**

El sistema será compatible con dispositivos de diferentes tamaños y resoluciones. Utilizará CSS flexible y aplicará técnicas de diseño responsivo, con lo cual se garantizará una experiencia de usuario consistente en todas las plataformas.

* **Buena Nomenclatura y Estructura del Código:**

Se utilizarán nombres de variables, funciones y archivos descriptivos. Además, se organizará el código en estructuras lógicas para mejorar la legibilidad y facilidad de mantenimiento del código.

* **Control de Versiones:**

Se mantendrá un control de versiones para un desarrollo colaborativo y organizado, con el fin de brindar un seguimiento efectivo de cambios en el código a lo largo del tiempo.   
  
**INTERFAZ**

**Externas**

* + El sistema utilizara la biblioteca Chart.js para visualizar datos de inventario mediante gráficos interactivos en tiempo real.

**Usuario**

* + Los elementos de navegación estarán organizados de manera lógica y clara, permitiendo a los usuarios moverse de manera fácil y eficiente por la aplicación.
  + Se utilizarán iconos y etiquetas fácilmente comprensibles para representar acciones y funciones, de manera que los usuarios puedan identificar rápidamente las opciones disponibles.
  + Se garantizará que la interfaz sea igualmente intuitiva y fácil de usar en pantallas más pequeñas, utilizando principios de diseño responsivo.
  + Se minimizará la necesidad de instrucciones externas o manuales detallados, permitiendo que los usuarios comprendan fácilmente cómo utilizar la aplicación sin una curva de aprendizaje demasiado alta.

**CONTROL DE ERRORES**

* Se proporcionará al usuario según el caso una página de error personalizada que sea clara y amigable, evitando mensajes técnicos que puedan ser confusos.
* Se proporcionarán mensajes de error descriptivos y comprensibles para el usuario.

Se implementarán estrategias de copias de seguridad y recuperación de datos para garantizar que los datos críticos estén protegidos y puedan ser restaurados en caso de pérdida.

**DISEÑO Y ARQUITECTURA**

**HERRAMIENTAS O LENGUAJES**

Para la realización del sistema se usarán las siguientes herramientas:

**Node.js**

Entorno de ejecución para JavaScript del lado del servidor, basado en el motor V8 de Google Chrome. Permite la creación de aplicaciones web y servidores de manera eficiente y escalable.

Este será usado debido a características como:

**Asincronía y No Bloqueo:** Node.js utiliza un modelo de entrada/salida no bloqueante, lo que permite manejar múltiples conexiones simultáneamente sin esperar a que se complete cada operación, resultando en un rendimiento más rápido y eficiente.

**Facilidad de Integración con JavaScript:** Al usar JavaScript tanto en el cliente como en el servidor, Node.js simplifica el desarrollo y mantenimiento del código, facilitando la integración entre el front-end y el back-end.

**Soporte para NPM (Node Package Manager):** Node.js cuenta con un vasto ecosistema de módulos y bibliotecas a través de NPM, lo que acelera el desarrollo al permitir la reutilización de paquetes y herramientas existentes.

**Escalabilidad y Rendimiento:** Node.js es ideal para aplicaciones que requieren alta escalabilidad y rendimiento en tiempo real, gracias a su arquitectura basada en eventos y su capacidad para manejar una gran cantidad de conexiones simultáneas.

**Amplia Comunidad y Documentación:** Node.js tiene una activa comunidad de desarrolladores y una extensa documentación, lo que facilita el aprendizaje y la resolución de problemas.

**Versatilidad:** Puede utilizarse para el desarrollo de aplicaciones web, APIs RESTful, y otros servicios del lado del servidor, ofreciendo flexibilidad en la creación de diferentes tipos de aplicaciones.

**Express**

Framework minimalista y flexible para Node.js que proporciona un conjunto robusto de características para el desarrollo web y la creación de APIs. Se utilizará para el diseño y arquitectura del sistema debido a sus ventajas como:

**Simplicidad y Eficiencia:** Express simplifica la configuración del servidor y el manejo de rutas, permitiendo construir aplicaciones web de manera rápida y eficiente.

**Middleware:** Permite el uso de middleware para manejar solicitudes, procesar datos y gestionar sesiones, proporcionando una estructura modular y extensible.

**Enrutamiento:** Ofrece un sistema de enrutamiento robusto y fácil de usar, lo que facilita la creación y gestión de rutas dentro de la aplicación.

**Integración con Módulos y Librerías:** Se integra fácilmente con otros módulos y bibliotecas de NPM, ampliando las capacidades del sistema sin complicaciones adicionales.

**Documentación y Comunidad:** Posee una buena documentación y una comunidad activa que contribuye al desarrollo de soluciones y mejores prácticas para la implementación.

**MYSQL Licencia pública general:**

Sistema de gestión de bases de datos relacional (RDBMS) de código abierto. Es ampliamente utilizado para almacenar y gestionar datos en aplicaciones web. MySQL utiliza el lenguaje de consulta estructurado (SQL) para realizar operaciones en la base de datos, como la inserción, actualización, recuperación y eliminación de datos.