
Especificación de requisitos de software

Proyecto: Dialsafa
Revisión [99.99]



Septiembre de 2025

Instrucciones para el uso de este formato

Este formato es una plantilla tipo para documentos de requisitos del software.

Está basado y es conforme con el estándar IEEE Std 830-1998.

Las secciones que no se consideren aplicables al sistema descrito podrán de forma justificada indicarse como no aplicables (NA).

Notas:

Los textos en color azul son indicaciones que deben eliminarse y, en su caso, sustituirse por los contenidos descritos en cada apartado.

Los textos entre corchetes del tipo “[Inserte aquí el texto]” permiten la inclusión directa de texto con el color y estilo adecuado a la sección, al pulsar sobre ellos con el puntero del ratón.

Los títulos y subtítulos de cada apartado están definidos como estilos de MS Word, de forma que su numeración consecutiva se genera automáticamente según se trate de estilos “Titulo1, Titulo2 y Titulo3”.

La sangría de los textos dentro de cada apartado se genera automáticamente al pulsar Intro al final de la línea de título. (Estilos Normal indentado1, Normal indentado 2 y Normal indentado 3).

El índice del documento es una tabla de contenido que MS Word actualiza tomando como criterio los títulos del documento.

Una vez terminada su redacción debe indicarse a Word que actualice todo su contenido para reflejar el contenido definitivo.

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
[Fecha]	[Rev]	[Descripcion]	[Firma o sello]

Documento validado por las partes en fecha: [Fecha]

Por el cliente	Por la empresa suministradora
Fdo. D./ Dña [Nombre]	Fdo. D./Dña [Nombre]

Contenido

FICHA DEL DOCUMENTO	3
CONTENIDO	4
1 INTRODUCCIÓN	6
1.1 Propósito	6
1.2 Alcance	6
1.3 Personal involucrado	6
1.4 Definiciones, acrónimos y abreviaturas	8
1.5 Referencias	8
1.6 Resumen	9
2 DESCRIPCIÓN GENERAL	9
2.1 Perspectiva del producto	9
2.2 Funcionalidad del producto	10
2.3 Características de los usuarios	10
2.4 Restricciones	11
2.5 Suposiciones y dependencias	12
2.6 Evolución previsible del sistema	12
3 REQUISITOS ESPECÍFICOS	13
3.1 Requisitos comunes de los interfaces	14
3.1.1 Interfaces de usuario	16
3.1.2 Interfaces de hardware	16
3.1.3 Interfaces de software	17
3.1.4 Interfaces de comunicación	18
3.2 Requisitos funcionales	19
3.2.1 Requisito funcional 1	Error! Bookmark not defined.
3.2.2 Requisito funcional 2	Error! Bookmark not defined.
3.2.3 Requisito funcional 3	Error! Bookmark not defined.
3.2.4 Requisito funcional n	Error! Bookmark not defined.
3.3 Requisitos no funcionales	22
3.3.1 Requisitos de rendimiento	22
3.3.2 Seguridad	22
3.3.3 Fiabilidad	23
3.3.4 Disponibilidad	24



3.3.5	Mantenibilidad	25
3.3.6	Portabilidad	26
3.4	Otros requisitos	27
4	APÉNDICES	28

1 Introducción

Esta Especificación de Requisitos de Software (SRS) presenta una visión integral del sistema de gestión de reservas para un hotel, el cual será desarrollado utilizando el lenguaje de programación Python y la metodología ágil Scrum. El documento tiene como finalidad establecer de manera precisa y estructurada los requisitos funcionales y no funcionales que guiarán el diseño, desarrollo, validación y mantenimiento del sistema.

Este SRS está dirigido a todos los actores involucrados en el proyecto, incluyendo el equipo de desarrollo, el Product Owner, el Scrum Master y los stakeholders clave, como el personal administrativo del hotel. Su propósito es garantizar una comprensión compartida del producto a construir, facilitar la planificación de los sprints y servir como referencia técnica durante todo el ciclo de vida del software.

1.1 Propósito

El propósito de esta Especificación de Requisitos de Software (SRS) es establecer de manera clara, precisa y estructurada los requisitos que debe cumplir el sistema de gestión de reservas para un hotel. Este documento sirve como base técnica y funcional para guiar el desarrollo del sistema, asegurando que todas las partes involucradas compartan una comprensión común del producto a construir.

1.2 Alcance

El producto a desarrollar se denomina **Sistema de Gestión de Reservas para un Hotel**, una aplicación orientada a objetos construida en Python, cuyo propósito es automatizar y optimizar el proceso de reservas hoteleras. Este sistema permitirá registrar clientes, gestionar habitaciones, controlar disponibilidad, procesar pagos y generar reportes administrativos.

1.3 Personal involucrado

Nombre	Fabian Ortega
Rol	Líder del proyecto
Categoría profesional	Ingeniero de software
Responsabilidades	Coordinar y supervisar todas las fases del proyecto (análisis, diseño, desarrollo, pruebas y entrega). Validar entregables técnicos y funcionales en cada sprint. Facilitar la comunicación entre el equipo técnico y los stakeholders. Aprobar el cronograma, los recursos y los cambios de alcance. Participar en reuniones de planificación, revisión y retrospectiva. Garantizar el cumplimiento de los estándares de calidad y documentación. Tomar decisiones estratégicas sobre arquitectura y evolución del sistema. Gestionar riesgos y asegurar la continuidad del proyecto.

Información de contacto	nestorf92@gmail.com
Aprobación	Aprobado

Nombre	Alan Sepúlveda
Rol	Desarrollador Backend
Categoría profesional	Desarrollador Python
Responsabilidades	<p>Implementar la lógica del sistema utilizando programación orientada a objetos en Python.</p> <p>Diseñar y estructurar clases, métodos y módulos reutilizables.</p> <p>Aplicar principios de encapsulamiento, modularidad y buenas prácticas de codificación.</p> <p>Integrar funcionalidades como gestión de clientes, habitaciones, reservas y reportes.</p> <p>Realizar pruebas unitarias y validar el comportamiento de cada componente.</p> <p>Colaborar en la definición de la arquitectura técnica del sistema.</p> <p>Documentar el código y generar manuales técnicos para mantenimiento futuro.</p> <p>Participar activamente en los sprints, entregando incrementos funcionales del producto.</p>
Información de contacto	alanjoelsepulveda04@gmail.com
Aprobación	Aprobado

Nombre	Samuel Cano
Rol	Analista de requisitos
Categoría profesional	Analista funcional
Responsabilidades	<p>Levantar, analizar y documentar requisitos funcionales y no funcionales.</p> <p>Redactar historias de usuario claras y priorizadas para el Product Backlog.</p> <p>Validar la coherencia entre los requisitos del negocio y las soluciones técnicas.</p> <p>Coordinar entrevistas con usuarios clave y recopilar necesidades operativas.</p> <p>Apoyar en la definición de criterios de aceptación por funcionalidad.</p> <p>Mantener la trazabilidad de requisitos durante todo el ciclo de vida del proyecto.</p> <p>Participar en sesiones de refinamiento y planificación de sprint.</p> <p>Detectar ambigüedades o inconsistencias en los requerimientos y proponer ajustes.</p>
Información de contacto	sadacaho723@gmail.com
Aprobación	Aprobado

Nombre	Diana Noa
Rol	Gestora de Requisitos
Categoría profesional	Ingeniera de Sistemas / Especialista en aseguramiento de calidad
Responsabilidades	<p>Coordinar el levantamiento, análisis y documentación de requisitos funcionales y no funcionales</p> <p>Facilitar la comunicación entre stakeholders y el equipo de</p>

	desarrollo Mantener la trazabilidad de requisitos durante todo el ciclo de vida del proyecto Validar que los entregables técnicos cumplan con las necesidades del negocio Apoyar en la priorización del Product Backlog y refinamiento de historias de usuario
Información de contacto	diansirnoa@gmail.com
Aprobación	Aprobado

1.4 Definiciones, acrónimos y abreviaturas

SGR	Sistema de Gestión de Reservas. Aplicación desarrollada para administrar las reservas de habitaciones en el hotel.
SRS	Software Requirements Specification. Documento que define los requisitos del sistema.
POO	Programación Orientada a Objetos. Paradigma de desarrollo utilizado para estructurar el sistema.
IDE	Entorno de Desarrollo Integrado. Herramienta utilizada para escribir, ejecutar y depurar el código fuente.
Scrum	Metodología ágil basada en sprints, utilizada para gestionar el desarrollo del proyecto.
Cliente	Persona que realiza una reserva en el hotel.
Reserva	Registro que vincula a un cliente con una habitación en fechas específicas.
Habitación	Unidad de alojamiento disponible en el hotel, con atributos como tipo, precio y estado.
Sprint	Iteración de desarrollo dentro del marco Scrum, con duración definida y entregables concretos.
Stakeholder	Parte interesada en el desarrollo del sistema, como gerentes, recepcionistas o usuarios finales.
Tester	Persona encargada de verificar el correcto funcionamiento del sistema mediante pruebas.
Backlog	Lista priorizada de funcionalidades y tareas pendientes por desarrollar en el proyecto.
Apéndice A	Contiene el glosario extendido de términos técnicos y funcionales utilizados en el sistema.
Manual de Usuario	Documento complementario que describe el uso operativo del sistema por parte del personal del hotel.

1.5 Referencias

Referencia	Título	Ruta	Fecha	Autor
Ref. 01	IEEE Std 830-1998	https://ieeexplore.ieee.org/document/720574	1998	IEEE Standards Association
Ref. 02	Guía Scrum – Versión oficial	https://scrumguides.org/scrum-guide.html	2020-11-01	Scrum.org
Ref. 03	Documentación oficial de Python	https://docs.python.org/3/	Actualizada	Python Software Foundation

1.6 Resumen

Este documento de Especificación de Requisitos de Software (SRS) describe de manera estructurada los requerimientos del sistema de gestión de reservas para el Hotel Boutique El Refugio. Su propósito es servir como guía técnica y funcional para el equipo de desarrollo, el Product Owner y los stakeholders, asegurando una comprensión común del producto a construir.

El contenido del documento está organizado en secciones que permiten abordar el sistema desde distintos niveles de detalle:

- La **Sección 1** presenta la introducción general, incluyendo el objetivo del documento, el alcance del sistema, el personal involucrado, las definiciones clave, las referencias utilizadas y esta síntesis.
- La **Sección 2** describe el sistema en términos generales, incluyendo sus funcionalidades principales, usuarios previstos, restricciones y dependencias.
- La **Sección 3** detalla los **requisitos funcionales**, organizados por módulos y casos de uso, especificando entradas, procesos y salidas esperadas.
- La **Sección 4** aborda los **requisitos no funcionales**, como rendimiento, seguridad, mantenibilidad y escalabilidad.
- La **Sección 5** incluye las **interfaces externas**, como bases de datos, servicios de pago o sistemas contables.
- La **Sección 6** presenta las **restricciones de diseño**, estándares técnicos y consideraciones arquitectónicas.
- Finalmente, los **apéndices** complementan el documento con glosarios, diagramas, criterios de aceptación y otros elementos de soporte.

Esta organización permite una lectura progresiva, desde la visión general hasta los detalles técnicos, facilitando la trazabilidad de requisitos y el trabajo colaborativo entre todos los actores del proyecto.

2 Descripción general

2.1 Perspectiva del producto

El **Sistema de Gestión de Reservas para el Hotel Boutique El Refugio (SGR)** es un producto **independiente**, diseñado para operar de forma autónoma en el entorno administrativo del hotel. Su propósito es automatizar los procesos de reserva de habitaciones, registro de clientes, control de disponibilidad y generación de reportes, sin depender de otros sistemas externos en su versión inicial.

Aunque el sistema está concebido como una solución modular y escalable, en esta primera fase no se integra con plataformas contables, sistemas de facturación electrónica ni portales web. Sin embargo, su arquitectura orientada a objetos y su diseño modular permiten futuras extensiones, como:

- Integración con sistemas contables o ERP del hotel
- Conexión con pasarelas de pago en línea
- Sincronización con motores de reserva web o aplicaciones móviles

El sistema se compone de los siguientes módulos internos:

- **Gestión de clientes**
- **Gestión de habitaciones**
- **Gestión de reservas**
- **Procesamiento de pagos**
- **Generación de reportes**

Cada módulo interactúa de forma controlada a través de interfaces internas bien definidas, siguiendo el patrón de arquitectura **MVC (Modelo-Vista-Controlador)**, aunque sin interfaz gráfica en esta versión.

2.2 Funcionalidad del producto

El **Sistema de Gestión de Reservas para el Hotel Boutique El Refugio (SGR)** está diseñado para automatizar los procesos operativos relacionados con la administración de reservas, clientes y habitaciones. A continuación, se presenta un resumen de las funcionalidades principales que el sistema debe realizar, organizadas de forma clara para facilitar su comprensión por parte de cualquier interlocutor técnico o no técnico:

Gestión de Clientes

- Registro de nuevos clientes con sus datos personales.
- Consulta y modificación de información de clientes existentes.
- Asociación de clientes a reservas específicas.

Gestión de Habitaciones

- Registro y clasificación de habitaciones por tipo, precio y estado.
- Actualización del estado de ocupación (disponible, ocupada, en mantenimiento).
- Visualización de disponibilidad por fechas.

Gestión de Reservas

- Registro de nuevas reservas con selección de cliente, habitación y fechas.
- Modificación o cancelación de reservas existentes.
- Validación de disponibilidad antes de confirmar la reserva.

Procesamiento de Pagos

- Asociación de métodos de pago a cada reserva (efectivo, tarjeta, transferencia).
- Registro del estado de pago (pendiente, parcial, completo).
- Generación de comprobantes internos de pago.

Generación de Reportes

- Reportes de ocupación por rango de fechas.
- Reportes de ingresos por tipo de habitación o método de pago.
- Exportación de reportes para análisis administrativo.

Seguridad y Control de Acceso

- Acceso restringido por roles (administrador, recepcionista, auditor).
- Validación de credenciales para ingreso al sistema.
- Registro de actividad por usuario para trazabilidad.

2.3 Características de los usuarios

Tipo de usuario	Cliente del Hotel
Formación	Varía según perfil (turista, ejecutivo, visitante ocasional)
Habilidades	Uso básico de tecnología, navegación web o móvil
Actividades	Consultar disponibilidad, realizar reservas, cancelar o modificar reservas

Tipo de usuario	Recepcionista
Formación	Técnico o tecnólogo en administración hotelera o afines
Habilidades	Manejo básico de sistemas informáticos, atención al cliente, gestión de reservas
Actividades	Registrar clientes, crear/modificar reservas, consultar disponibilidad

Tipo de usuario	Administrador Hotelero
Formación	Profesional en administración, turismo o gerencia hotelera
Habilidades	Toma de decisiones, análisis de reportes, supervisión operativa
Actividades	Consultar reportes, validar pagos, supervisar estado de habitaciones y reservas

Tipo de usuario	Contador / Financiero
Formación	Profesional en contaduría o finanzas
Habilidades	Manejo de cifras, conciliación de pagos, validación de ingresos
Actividades	Revisar reportes de pagos, validar registros contables, auditar transacciones

Tipo de usuario	Equipo Técnico (Desarrolladores/Testers)
Formación	Ingenieros de sistemas, desarrolladores backend, testers funcionales
Habilidades	Programación en Python, pruebas unitarias, documentación técnica
Actividades	Implementar funcionalidades, realizar pruebas, mantener el sistema

2.4 Restricciones

Durante el diseño y desarrollo del sistema se deben tener en cuenta las siguientes limitaciones técnicas, metodológicas y operativas que condicionan la implementación del producto:

Restricciones Tecnológicas

- **Lenguaje de programación:** El sistema será desarrollado exclusivamente en **Python 3.x**, utilizando programación orientada a objetos (POO) como paradigma principal.
- **Arquitectura:** Se empleará una arquitectura **modular tipo MVC**, aunque en esta versión no se implementará la capa de interfaz gráfica.
- **Base de datos:** Se utilizará **SQLite** como sistema de gestión de base de datos local, por su ligereza y facilidad de integración con Python.
- **Sistema operativo:** El sistema está diseñado para ejecutarse en entornos **Windows 10 o superior**, aunque se prevé compatibilidad futura con Linux.

Restricciones de Desarrollo

- **Metodología:** Se aplicará el marco de trabajo **Scrum**, con sprints de duración fija, backlog priorizado y entregables incrementales.
- **Normas de documentación:** Toda la documentación técnica y funcional seguirá el estándar **IEEE 830**, incluyendo trazabilidad de requisitos, roles definidos y criterios de aprobación.
- **Control de versiones:** El código fuente será gestionado mediante **Git**, con repositorio privado y ramas por sprint.

Restricciones de Hardware

- El sistema debe poder ejecutarse en equipos con las siguientes especificaciones mínimas:
 - Procesador: Intel Core i3 o superior
 - Memoria RAM: 4 GB
 - Almacenamiento: 500 MB disponibles
 - Resolución mínima: 1024x768 (solo para futuras versiones con interfaz gráfica)

Restricciones de Seguridad y Acceso

- El sistema debe implementar **control de acceso por roles**, con validación de credenciales y registro de actividad por usuario.
- No se contempla en esta versión el acceso remoto ni la exposición a redes públicas.

Restricciones de Entorno

- El sistema será instalado y operado **localmente en el hotel**, sin dependencia de servicios en la nube.
- No se requiere conexión a internet para su funcionamiento básico.

2.5 Suposiciones y dependencias

El desarrollo del **Sistema de Gestión de Reservas (SGR)** se basa en una serie de suposiciones y dependencias técnicas, operativas y organizacionales que, en caso de modificarse, podrían afectar directamente los requisitos definidos en este documento. A continuación, se detallan los principales factores considerados:

Suposiciones

- Se asume que el sistema será instalado y operado en equipos con **sistema operativo Windows 10 o superior**. Si el hotel decide migrar a otro sistema operativo (Linux, macOS), se requerirá adaptación del entorno de ejecución y pruebas.
- Se presupone que el personal del hotel cuenta con **formación básica en el uso de sistemas informáticos**, lo cual permite operar el sistema sin necesidad de interfaz gráfica.
- Se asume que el hotel no requiere **integración inmediata con sistemas contables, pasarelas de pago o plataformas web**, por lo que el sistema funcionará de forma autónoma en su primera versión.
- Se considera que el equipo técnico tendrá acceso a **entornos de desarrollo compatibles con Python 3.x**, incluyendo bibliotecas estándar y herramientas como SQLite y Git.
- Se asume que los usuarios internos (recepcionistas, administradores) seguirán los procedimientos definidos en el Manual de Usuario, evitando el uso incorrecto del sistema.

Dependencias

- El sistema depende de la disponibilidad del **lenguaje Python 3.x** y sus librerías estándar para el desarrollo y ejecución de funcionalidades.
- Depende del correcto funcionamiento de **SQLite** como motor de base de datos local. Cualquier cambio en el motor requeriría ajustes en la lógica de persistencia.
- Depende del uso de la metodología **Scrum**, por lo que cualquier cambio en el enfoque de desarrollo (por ejemplo, migrar a cascada o Kanban) implicaría reestructurar el ciclo de vida del proyecto.
- Depende de la disponibilidad de **hardware mínimo** en el hotel (ver sección 2.4), ya que un entorno con recursos insuficientes podría limitar el rendimiento del sistema.
- Depende del cumplimiento de las **normas de documentación IEEE 830**, ya que cualquier cambio en los estándares requeriría reformular la estructura del SRS.

2.6 Evolución previsible del sistema

El **Sistema de Gestión de Reservas (SGR)** ha sido diseñado con una arquitectura modular y orientada a objetos que permite su evolución progresiva sin comprometer la estabilidad de la versión inicial. A continuación se identifican posibles mejoras y extensiones que podrán ser analizadas e implementadas en futuras fases del proyecto:

Integración con plataformas externas

- **Portal web de reservas:** Desarrollo de una interfaz web para que los clientes puedan consultar disponibilidad, realizar reservas y pagos en línea.

- **Aplicación móvil:** Versión para dispositivos móviles que permita a los usuarios gestionar sus reservas desde cualquier lugar.
- **Pasarelas de pago:** Integración con servicios como PayU, Stripe o MercadoPago para procesar pagos electrónicos de forma segura.

Expansión funcional

- **Gestión de promociones y descuentos:** Módulo para aplicar tarifas especiales por temporada, fidelización o eventos.
- **Control de inventario hotelero:** Registro y seguimiento de productos y suministros utilizados en el servicio de habitaciones.
- **Gestión de eventos y paquetes turísticos:** Administración de reservas para actividades adicionales ofrecidas por el hotel.

Mejora en seguridad y trazabilidad

- **Auditoría avanzada:** Registro detallado de operaciones por usuario, con reportes de actividad y alertas por comportamiento anómalo.
- **Autenticación multifactor (MFA):** Refuerzo del acceso al sistema mediante validación por correo o dispositivo móvil.

Inteligencia de negocio

- **Análisis predictivo de ocupación:** Uso de algoritmos para anticipar demanda según temporada, eventos locales o comportamiento histórico.
- **Dashboard gerencial:** Panel visual con indicadores clave de desempeño (KPI), ingresos, ocupación y satisfacción del cliente.

Escalabilidad técnica

- **Migración a base de datos relacional robusta (PostgreSQL, MySQL)** para soportar mayor volumen de datos.
- **Despliegue en la nube (Azure, AWS)** para facilitar acceso remoto, copias de seguridad automáticas y alta disponibilidad.
- **Contenedores y microservicios:** Reestructuración técnica para facilitar mantenimiento, despliegue y escalabilidad horizontal.

3 Requisitos específicos

Número de requisito	RF-01
Nombre de requisito	Registro de clientes
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Manual de Usuario / Recepcionista
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF-02
Nombre de requisito	Consulta de disponibilidad
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Reunión con stakeholders
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF-03
Nombre de requisito	Creación de reservas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Plan de Proyecto / Recepcionista
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF-04
---------------------	-------

Nombre de requisito	Modificación de reservas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Recepcionista / Administrador
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF-05
Nombre de requisito	Cancelación de reservas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Recepcionista
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF-06
Nombre de requisito	Registro de pagos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Administrador / Auditor
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF-07
Nombre de requisito	Generación de reportes
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Gerente / Administrador
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF-08
Nombre de requisito	Control de acceso por roles
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Documento de seguridad
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF-09
Nombre de requisito	Actualización de estado de habitación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Personal de mantenimiento
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF-10
Nombre de requisito	Búsqueda de clientes
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Recepcionista
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

3.1 Requisitos comunes de los interfaces

Esta sección describe las entradas y salidas comunes del sistema de software, considerando que en esta versión no se implementará una interfaz gráfica, sino una interfaz de consola o escritorio basada en comandos estructurados. Las entradas y salidas están organizadas por módulo funcional para facilitar su trazabilidad y validación.

Entradas del sistema

Módulo	Entrada	Formato / Tipo de dato	Validaciones
Gestión de clientes	Nombre completo, documento, teléfono, correo	Texto / Numérico / Email	No nulo, formato válido, documento único
Gestión de habitaciones	Tipo de habitación, precio, estado	Texto / Numérico / Enumerado	Precio positivo, estado válido
Gestión de reservas	Cliente, habitación, fecha de entrada/salida	Referencia / Fecha / Enumerado	Disponibilidad, fechas válidas, cliente existente
Registro de pagos	Monto, método de pago, estado de pago	Numérico / Enumerado	Monto positivo, método permitido
Búsqueda de clientes	Nombre o documento	Texto / Numérico	Coincidencia parcial o exacta
Control de acceso	Usuario, contraseña	Texto / Texto	Credenciales válidas, rol asignado

Salidas del sistema

Módulo	Salida	Formato / Tipo de dato	Destino / Uso
Gestión de clientes	Confirmación de registro / listado de clientes	Texto / Tabla	Consola / Reporte interno
Gestión de habitaciones	Estado actualizado / listado de habitaciones	Texto / Tabla	Consola / Reporte de disponibilidad
Gestión de reservas	Confirmación de reserva / listado de reservas	Texto / Tabla	Consola / Reporte de ocupación
Registro de pagos	Comprobante interno / estado de cuenta	Texto / Tabla	Consola / Auditoría interna
Generación de reportes	Reporte de ocupación / ingresos	Tabla / Exportable (CSV, PDF futuro)	Consola / Gerencia
Control de acceso	Mensaje de acceso permitido o denegado	Texto	Consola / Registro de actividad

3.1.1 Interfaces de usuario

En esta primera versión del sistema, el producto será operado mediante una **interfaz de consola**, sin componentes gráficos, orientada a usuarios internos del hotel como recepcionistas, administradores y auditores. La interfaz está diseñada para ser clara, estructurada y funcional, priorizando la eficiencia operativa y la facilidad de uso.

Estilo y presentación

- **Tipo de interfaz:** Consola de texto
- **Formato visual:**
 - Menús jerárquicos con numeración para navegación
 - Prompts claros para ingreso de datos
 - Mensajes de confirmación, error y validación en texto plano
- **Colores:**
 - Fondo negro o blanco según configuración del sistema operativo
 - Texto en blanco o gris para instrucciones
 - Texto en verde para confirmaciones exitosas
 - Texto en rojo para errores o advertencias
- **Lenguaje:** español neutro, con mensajes breves y directos

Estructura funcional

Cada módulo del sistema presenta una interfaz específica con opciones numeradas.

Los formularios de entrada se presentan secuencialmente, solicitando los datos necesarios uno por uno, con validaciones inmediatas.

Acceso por rol

Al iniciar sesión, el sistema identifica el rol del usuario (recepcionista, administrador, auditor) y muestra únicamente las opciones habilitadas para ese perfil. Esto garantiza seguridad y simplicidad en la navegación.

Salidas y reportes

Los reportes generados por el sistema se presentan en formato tabular dentro de la consola, con opción de exportación futura a CSV o PDF

3.1.2 Interfaces de hardware

El sistema de gestión de reservas está diseñado para operar en un entorno local, utilizando hardware estándar disponible en el hotel. A continuación se describen las interfaces lógicas entre el software y los componentes de hardware, incluyendo las características de configuración necesarias para su correcto funcionamiento.

Interfaz con estación de trabajo

- **Tipo de conexión:** Local, sin dependencia de red externa
- **Sistema operativo requerido:** Windows 10 o superior
- **Configuración mínima del equipo:**
 - Procesador: Intel Core i3 o superior
 - Memoria RAM: 4 GB
 - Almacenamiento disponible: 500 MB
 - Resolución de pantalla: 1024x768 (para futuras versiones con GUI)
- **Interacción:** El sistema se ejecuta desde una terminal de consola, permitiendo entrada por teclado y salida por pantalla.

Interfaz con periféricos de entrada

- **Teclado:**
Entrada principal para comandos, datos de clientes, fechas y navegación por menús.
Requiere configuración regional en español para facilitar el ingreso de caracteres especiales.
- **Mouse (opcional):**
No requerido en esta versión, ya que la interfaz es completamente textual.
Podrá ser habilitado en futuras versiones con interfaz gráfica.

Interfaz con impresora

- **Tipo:** Impresora local conectada por USB o red interna
- **Uso previsto:**
Impresión de reportes internos (ocupación, ingresos, estado de habitaciones)
Generación de comprobantes de pago o confirmaciones de reserva
- **Formato de salida:** Texto plano o tabla estructurada, compatible con impresión directa desde consola o exportación futura a PDF

Interfaz con almacenamiento

- **Base de datos:** SQLite
Archivo local .db almacenado en el disco duro del equipo
No requiere servidor externo ni configuración avanzada
Compatible con copias de seguridad manuales o automatizadas
- **Exportación de datos:**
En futuras versiones, se habilitará la exportación de reportes a formatos como CSV o PDF
Requiere permisos de escritura en el sistema de archivos local

Interfaz de autenticación

- **Dispositivo:** Estación de trabajo con acceso restringido por usuario
- **Validación:** Ingreso de credenciales por teclado
- **Configuración:** Archivo de usuarios y roles almacenado localmente, con cifrado básico

3.1.3 Interfaces de software

El sistema de gestión de reservas está diseñado para operar en un entorno local, utilizando hardware estándar disponible en el hotel. A continuación se describen las interfaces lógicas entre el software y los componentes de hardware, incluyendo las características de configuración necesarias para su correcto funcionamiento.

Interfaz con estación de trabajo

- **Tipo de conexión:** Local, sin dependencia de red externa
- **Sistema operativo requerido:** Windows 10 o superior
- **Configuración mínima del equipo:**
Procesador: Intel Core i3 o superior
Memoria RAM: 4 GB
Almacenamiento disponible: 500 MB
Resolución de pantalla: 1024x768 (para futuras versiones con GUI)
- **Interacción:** El sistema se ejecuta desde una terminal de consola, permitiendo entrada por teclado y salida por pantalla.

Interfaz con periféricos de entrada

- **Teclado:**
Entrada principal para comandos, datos de clientes, fechas y navegación por menús.
Requiere configuración regional en español para facilitar el ingreso de caracteres especiales.
- **Mouse (opcional):**
No requerido en esta versión, ya que la interfaz es completamente textual.

Podrá ser habilitado en futuras versiones con interfaz gráfica.

Interfaz con impresora

- **Tipo:** Impresora local conectada por USB o red interna
- **Uso previsto:**
 - Impresión de reportes internos (ocupación, ingresos, estado de habitaciones)
 - Generación de comprobantes de pago o confirmaciones de reserva
- **Formato de salida:** Texto plano o tabla estructurada, compatible con impresión directa desde consola o exportación futura a PDF

Interfaz con almacenamiento

- **Base de datos:** SQLite
 - Archivo local .db almacenado en el disco duro del equipo
 - No requiere servidor externo ni configuración avanzada
 - Compatible con copias de seguridad manuales o automatizadas
- **Exportación de datos:**
 - En futuras versiones, se habilitará la exportación de reportes a formatos como CSV o PDF
 - Requiere permisos de escritura en el sistema de archivos local

Interfaz de autenticación

- **Dispositivo:** Estación de trabajo con acceso restringido por usuario
- **Validación:** Ingreso de credenciales por teclado
- **Configuración:** Archivo de usuarios y roles almacenado localmente, con cifrado básico

3.1.4 Interfaces de comunicación

En la versión actual del sistema, el **SGR (Sistema de Gestión de Reservas)** opera de forma **local y autónoma**, sin necesidad de comunicación con sistemas externos. Sin embargo, se ha previsto una arquitectura extensible que permitirá futuras integraciones con otros servicios y plataformas. A continuación, se detallan los requisitos actuales y las proyecciones de comunicación del sistema:

Estado actual

- **Tipo de comunicación:** Local, sin conexión a red externa ni servidores remotos.
- **Protocolo interno:** Comunicación entre módulos mediante llamadas directas en Python, utilizando estructuras de datos compartidas y persistencia en SQLite.
- **Sincronización:** No se requiere sincronización con otros sistemas en tiempo real.
- **Seguridad:** No se contempla en esta fase el uso de cifrado en tránsito, ya que no hay transmisión de datos fuera del equipo.

Proyección de interfaces externas

En futuras versiones, se prevé que el sistema se comunique con los siguientes componentes externos:

Sistema externo	Propósito de comunicación	Protocolo previsto
Portal web de reservas	Permitir a clientes consultar disponibilidad y reservar	HTTP/HTTPS – API REST
Pasarela de pagos	Procesar pagos electrónicos de reservas	HTTPS – API REST / Webhooks
Sistema contable del hotel	Exportar ingresos y transacciones	CSV/XML vía FTP o API REST
Aplicación móvil	Consulta y gestión de reservas desde dispositivos	HTTPS – API REST
Servicio de correo electrónico	Envío de confirmaciones y notificaciones	SMTP / API de terceros (SendGrid, Mailgun)

Requisitos técnicos para futuras comunicaciones

- El sistema deberá implementar **control de errores y validación de respuesta** en cada intercambio de datos.
- Se requerirá **autenticación por token o clave API** para servicios externos.
- Las comunicaciones deberán cumplir con estándares de seguridad como **TLS 1.2 o superior**.
- Se deberá definir un **modelo de datos común (JSON/XML)** para garantizar interoperabilidad.

3.2 Requisitos funcionales

Esta sección define las acciones fundamentales que debe realizar el sistema al recibir información, procesarla y generar resultados. Cada requisito funcional incluye validaciones de entrada, secuencia de operaciones, manejo de errores, parámetros, salidas esperadas y especificaciones de almacenamiento en base de datos.

3.2.1 Requisito funcional 1 – Registro de clientes

- Entrada:** Nombre completo, número de documento, teléfono, correo electrónico
- Validaciones:**
 - Campos obligatorios no nulos
 - Documento único (no duplicado en base de datos)
 - Formato de correo válido
- Secuencia de operaciones:**
 - Solicitar datos al usuario
 - Validar formato y unicidad
 - Registrar cliente en base de datos
 - Confirmar registro exitoso
- Respuesta a errores:**
 - Si el documento ya existe, mostrar mensaje de duplicado
 - Si algún campo está vacío o mal formado, solicitar corrección
- Parámetros:** Datos personales del cliente
- Salida:** Mensaje de confirmación y visualización del cliente registrado
- Relación entrada/salida:** Entrada → Validación → Registro → Confirmación
- Base de datos:** Tabla clientes con campos: id, nombre, documento, telefono, correo (todos requeridos)

3.2.2 Requisito funcional 2 – Consulta de disponibilidad

- **Entrada:** Fecha de entrada y fecha de salida
- **Validaciones:**
 - Fechas válidas y en formato dd/mm/aaaa
 - Fecha de salida posterior a la de entrada
- **Secuencia de operaciones:**
 1. Solicitar rango de fechas
 2. Verificar habitaciones no ocupadas en ese rango
 3. Mostrar listado de habitaciones disponibles
- **Respuesta a errores:**
 - Si las fechas son inválidas, mostrar mensaje de error
 - Si no hay disponibilidad, mostrar mensaje informativo
- **Parámetros:** Fechas de búsqueda
- **Salida:** Listado de habitaciones disponibles con tipo y precio
- **Relación entrada/salida:** Fechas → Consulta → Listado
- **Base de datos:** Consulta sobre tabla reservas y habitaciones

3.2.3 Requisito funcional 3 – Creación de reservas

- **Entrada:** Cliente, habitación, fecha de entrada, fecha de salida
- **Validaciones:**
 - Cliente debe existir en base de datos
 - Habitación debe estar disponible en el rango solicitado
 - Fechas válidas
- **Secuencia de operaciones:**
 1. Solicitar datos de reserva
 2. Validar disponibilidad y cliente
 3. Registrar reserva en base de datos
 4. Confirmar reserva
- **Respuesta a errores:**
 - Si el cliente no existe, solicitar registro previo
 - Si la habitación está ocupada, mostrar mensaje de no disponibilidad
- **Parámetros:** Datos de reserva
- **Salida:** Mensaje de confirmación y código de reserva
- **Relación entrada/salida:** Datos → Validación → Registro → Confirmación
- **Base de datos:** Tabla reservas con campos: id, cliente_id, habitacion_id, fecha_entrada, fecha_salida

3.2.4 Requisito funcional 4 - Modificación de reservas

- **Entrada:** Código de reserva, nuevos datos
- **Validaciones:** Reserva existente, disponibilidad nueva habitación
- **Secuencia:** Solicitar código → Validar → Actualizar → Confirmar
- **Errores:** Reserva no encontrada, conflicto de fechas
- **Parámetros:** Código de reserva, nuevos datos
- **Salida:** Confirmación de modificación
- **Relación:** Entrada → Validación → Actualización → Salida
- **Base de datos:** Tabla reservas

3.2.5 Requisito funcional 5 - Cancelación de reservas

- **Entrada:** Código de reserva
- **Validaciones:** Reserva existente
- **Secuencia:** Solicitar código → Validar → Eliminar → Confirmar
- **Errores:** Reserva no encontrada
- **Parámetros:** Código de reserva
- **Salida:** Confirmación de cancelación
- **Relación:** Entrada → Validación → Eliminación → Salida
- **Base de datos:** Tabla reservas

3.2.6 Requisito funcional 6 – Registro de pagos

- **Entrada:** Código de reserva, monto, método de pago, estado de pago
- **Validaciones:**
 - Reserva debe existir
 - Monto positivo
 - Método permitido (efectivo, tarjeta, transferencia)
- **Secuencia de operaciones:**
 1. Solicitar datos de pago
 2. Validar reserva y monto
 3. Registrar pago en base de datos
 4. Actualizar estado de reserva si aplica
- **Respuesta a errores:**
 - Si la reserva no existe, mostrar mensaje de error
 - Si el monto es inválido, solicitar corrección
- **Parámetros:** Datos financieros
- **Salida:** Comprobante interno de pago
- **Relación entrada/salida:** Datos → Validación → Registro → Comprobante
- **Base de datos:** Tabla pagos con campos: id, reserva_id, monto, metodo, estado

3.2.7 Requisito funcional 7 - Generación de reportes

- **Entrada:** Rango de fechas, tipo de reporte
- **Validaciones:** Fechas válidas, tipo permitido
- **Secuencia:** Solicitar parámetros → Consultar datos → Generar reporte
- **Errores:** Fechas inválidas, sin datos disponibles
- **Parámetros:** Fechas, tipo de reporte
- **Salida:** Tabla de resultados
- **Relación:** Entrada → Consulta → Salida
- **Base de datos:** Tablas reservas, pagos, habitaciones

3.2.8 Requisito funcional 8 - Control de acceso por roles

- **Entrada:** Usuario, contraseña
- **Validaciones:** Credenciales válidas, rol asignado
- **Secuencia:** Solicitar login → Validar → Mostrar menú por rol
- **Errores:** Usuario no registrado, contraseña incorrecta
- **Parámetros:** Credenciales
- **Salida:** Acceso permitido o denegado
- **Relación:** Entrada → Validación → Salida
- **Base de datos:** Tabla usuarios (id, usuario, contraseña, rol)

3.2.9 Requisito funcional 9 - Actualización de estado de habitación

- **Entrada:** Código de habitación, nuevo estado
- **Validaciones:** Habitación existente, estado válido
- **Secuencia:** Solicitar datos → Validar → Actualizar → Confirmar
- **Errores:** Habitación no encontrada, estado inválido
- **Parámetros:** Código y estado
- **Salida:** Confirmación de actualización
- **Relación:** Entrada → Validación → Actualización → Salida
- **Base de datos:** Tabla habitaciones (id, tipo, precio, estado)

3.2.10 Requisito funcional 10 - Búsqueda de clientes

- **Entrada:** Nombre o documento
- **Validaciones:** Formato válido
- **Secuencia:** Solicitar criterio → Buscar → Mostrar resultados
- **Errores:** Cliente no encontrado

- **Parámetros:** Texto de búsqueda
- **Salida:** Listado de coincidencias
- **Relación:** Entrada → Búsqueda → Salida
- **Base de datos:** Tabla clientes

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

El sistema ha sido diseñado para operar en un entorno local con carga controlada, considerando el tamaño y flujo operativo del Hotel Boutique El Refugio. A continuación, se especifican los requisitos de rendimiento esperados, todos ellos medibles y alineados con las condiciones reales del entorno:

Usuarios y terminales

- El sistema debe soportar hasta **5 terminales locales** operando simultáneamente (recepción, administración, auditoría, mantenimiento, gerencia).
- Se espera un máximo de **3 usuarios concurrentes** en horas pico (check-in/check-out).

Transacciones

- El sistema debe ser capaz de procesar hasta **20 transacciones por minuto**, incluyendo reservas, pagos, actualizaciones y consultas.
- El **95% de las transacciones** deben completarse en **menos de 1 segundo**, bajo condiciones normales de operación.
- Las operaciones de consulta (disponibilidad, búsqueda de clientes) deben responder en **menos de 2 segundos** en el 100% de los casos.

Reportes

- La generación de reportes debe completarse en **menos de 5 segundos** para rangos de fechas de hasta 30 días.
- Para reportes más extensos (hasta 1 año), el tiempo de generación no debe superar los **10 segundos**.

Acceso a base de datos

- Las operaciones de lectura y escritura en la base de datos SQLite deben ejecutarse en **menos de 500 milisegundos** por transacción.
- El sistema debe mantener una tasa de éxito de **99.9%** en operaciones de persistencia bajo condiciones normales.

Inicio de sesión

- La validación de credenciales y carga del menú por rol debe completarse en **menos de 2 segundos**.

Recuperación ante errores

- En caso de error de entrada, el sistema debe mostrar el mensaje correspondiente en **menos de 1 segundo**.
- En caso de fallo en la base de datos, el sistema debe registrar el incidente y permitir reintento en **menos de 3 segundos**.

3.3.2 Seguridad

El sistema ha sido diseñado para operar en un entorno local, pero incorpora medidas de seguridad que protegen la integridad, confidencialidad y disponibilidad de la información, así como el control de acceso a funcionalidades

críticas. A continuación, se detallan los requisitos de seguridad implementados y previstos:

Control de acceso

- El sistema debe validar credenciales de usuario mediante **usuario y contraseña**, almacenados de forma cifrada.
- Cada usuario debe tener asignado un **rol específico** (repcionista, administrador, auditor), que determina las funcionalidades disponibles.
- El sistema debe impedir el acceso a funciones no autorizadas según el rol activo.

Asignación de funcionalidades por módulo

- Las funcionalidades estarán restringidas por perfil:
Recepcionista: gestión de clientes, reservas, pagos básicos
Administrador: reportes, pagos avanzados, gestión de habitaciones
Auditor: consulta de registros, verificación de trazabilidad
- El menú de opciones se adapta dinámicamente al rol del usuario autenticado.

Registro de actividad (logs)

- El sistema debe generar un **log interno** que registre:
Fecha y hora de acceso
Usuario autenticado
Operaciones realizadas (alta, modificación, cancelación, consulta)
- Los logs deben almacenarse en un archivo local protegido contra escritura externa.

Comprobación de integridad

- Antes de ejecutar operaciones críticas (modificación de reservas, pagos), el sistema debe verificar:
Existencia de los registros involucrados
Coherencia entre fechas, estados y montos
- En caso de inconsistencias, se debe bloquear la operación y registrar el intento en el log.

Técnicas criptográficas

- Las contraseñas de usuario deben almacenarse utilizando **hash criptográfico (SHA-256 o superior)**.
- En futuras versiones, se contempla el uso de **tokens de sesión cifrados** para mejorar la autenticación.

Restricciones de comunicación

- En esta versión, el sistema no se comunica con módulos externos, pero se prevé:
Validación de origen y destino en futuras integraciones
Uso de protocolos seguros (HTTPS, TLS) para transmisión de datos
Autenticación por clave API en servicios externos

Protección contra sabotajes y errores

- El sistema debe validar todas las entradas antes de procesarlas, evitando inyecciones o desbordamientos.
- En caso de error crítico, el sistema debe:
Mostrar un mensaje claro al usuario
Registrar el incidente en el log
Permitir reintento seguro sin pérdida de datos

3.3.3 Fiabilidad

El sistema ha sido diseñado para operar de forma continua y estable en un entorno local, con una carga controlada y sin dependencia de servicios externos.

A continuación, se especifican los factores de fiabilidad necesarios para garantizar su disponibilidad y correcto funcionamiento:

Tiempo medio entre fallos (MTBF)

- El sistema debe mantener un **tiempo medio entre fallos (MTBF)** de al menos **500 horas de operación continua**, bajo condiciones normales de uso.

Tasa de incidentes permisibles

- Se permite un máximo de **1 incidente crítico por mes**, entendiendo como incidente crítico cualquier fallo que impida el registro de reservas o pagos.
- Se permite un máximo de **3 incidentes menores por semana**, como errores de validación o mensajes incorrectos, siempre que no afecten la integridad de los datos.

Recuperación ante fallos (MTTR)

- El sistema debe permitir la **recuperación ante fallos menores en menos de 5 minutos**, mediante reinicio o corrección de entrada.
- En caso de fallo crítico, el sistema debe permitir restaurar la operación en **menos de 30 minutos**, utilizando respaldo local de la base de datos.

Integridad de datos

- El sistema debe garantizar una **tasa de integridad de datos del 99.99%**, evitando pérdidas o corrupciones en operaciones de lectura/escritura.
- Todas las operaciones deben ser **atómicas**, asegurando que los datos se registren completamente o no se registren en absoluto.

Registro de errores

- Todos los errores deben ser registrados en un **log interno**, incluyendo:
 - Fecha y hora del incidente
 - Usuario activo
 - Módulo afectado
 - Descripción del error
- El log debe ser accesible solo por usuarios con rol de auditor o administrador.

Pruebas de fiabilidad

- El sistema debe superar pruebas de estrés con al menos **100 operaciones consecutivas sin fallo**.
- Debe ser capaz de operar durante **8 horas continuas** sin interrupciones ni degradación de rendimiento.

3.3.4 Disponibilidad

El sistema ha sido diseñado para operar de forma continua en un entorno local, garantizando que los usuarios internos del hotel puedan acceder a sus funcionalidades en todo momento durante la jornada laboral. A continuación se especifican los factores de disponibilidad exigidos al sistema:

Nivel de disponibilidad

- El sistema debe mantener una **disponibilidad mínima del 99.5% mensual**, lo que equivale a un máximo de **3 horas y 40 minutos de inactividad permitida por mes**.
- Durante el horario operativo del hotel (6:00 a.m. a 10:00 p.m.), el sistema debe estar disponible en al menos **99.9% del tiempo**, lo que representa un máximo de **9 minutos de inactividad mensual** en ese intervalo.

Recuperación ante interrupciones

- En caso de interrupción del servicio por error interno, el sistema debe permitir la **recuperación completa en menos de 30 minutos**, utilizando respaldo local de la base de datos.
- Las interrupciones planificadas (mantenimiento, actualizaciones) deben ser notificadas con al menos **24 horas de anticipación** y ejecutarse fuera del horario operativo.

Pruebas de disponibilidad

- El sistema debe superar pruebas de disponibilidad simulada durante **8 horas continuas**, sin presentar fallos de acceso, lentitud o pérdida de datos.
- Se debe garantizar que los módulos críticos (reservas, pagos, reportes) estén accesibles en todo momento, incluso si otros módulos presentan fallos menores.

Mecanismos de respaldo

- El sistema debe permitir la creación de **copias de seguridad manuales** al finalizar cada jornada, con posibilidad de restauración inmediata.
- En futuras versiones, se contempla la implementación de **respaldo automático diario** y replicación en la nube para mejorar la disponibilidad general.

3.3.5 Mantenibilidad

El sistema ha sido diseñado con una arquitectura modular y orientada a objetos que facilita su mantenimiento correctivo, adaptativo y evolutivo. A continuación se identifican los tipos de mantenimiento requeridos, los responsables de su ejecución y la periodicidad recomendada para cada tarea.

Tipos de mantenimiento

- **Mantenimiento correctivo:**
Corrección de errores detectados durante la operación (fallos de validación, inconsistencias de datos, errores de ejecución).
Responsable: **Desarrollador del sistema**
Frecuencia: **Según ocurrencia**, con tiempo de respuesta máximo de 48 horas.
- **Mantenimiento adaptativo:**
Ajustes derivados de cambios en procesos internos del hotel (nuevas políticas de reserva, modificación de tarifas, incorporación de nuevos tipos de habitación).
Responsable: **Desarrollador en coordinación con el administrador del hotel**
Frecuencia: **Mensual o según necesidad operativa**
- **Mantenimiento evolutivo:**
Incorporación de nuevas funcionalidades (interfaz gráfica, exportación de reportes, integración con pasarelas de pago).
Responsable: **Equipo de desarrollo**
Frecuencia: **Planificada por versión**, según roadmap del proyecto
- **Mantenimiento preventivo:**
Revisión periódica del estado de la base de datos, limpieza de registros obsoletos, verificación de logs y respaldo de información.
Responsable: **Administrador del sistema**
Frecuencia: **Semanal (verificación de logs) y diaria (respaldo de base de datos)**

Tareas programadas

Tarea	Responsable	Frecuencia
Verificación de integridad de base de datos	Administrador del sistema	Semanal
Generación de estadísticas de ocupación	Administrador / Gerente	Semanal y mensual
Limpieza de logs y registros antiguos	Desarrollador / Auditor	Mensual
Actualización de tarifas y habitaciones	Administrador del hotel	Según necesidad
Revisión de código y dependencias	Equipo de desarrollo	Trimestral

Procedimientos de mantenimiento

Todas las tareas deben ser registradas en un log de mantenimiento, indicando fecha, responsable, tipo de intervención y resultado.

Las actualizaciones del sistema deben realizarse fuera del horario operativo (preferiblemente entre 10:00 p.m. y 6:00 a.m.) para evitar interrupciones.

Se recomienda mantener una copia de seguridad diaria antes de ejecutar cualquier tarea de mantenimiento que afecte la base de datos.

3.3.6 Portabilidad

El sistema ha sido diseñado para facilitar su traslado y ejecución en distintos entornos operativos, priorizando la independencia de plataforma y la adaptabilidad futura. A continuación, se especifican los atributos que garantizan su portabilidad:

Dependencia de servidor

- El sistema actual opera de forma **local y autónoma**, sin necesidad de servidor externo.
- **Porcentaje de componentes dependientes del servidor:** 0%
- **Porcentaje de código dependiente del servidor:** 0%

Lenguaje de programación

- El sistema está desarrollado en **Python 3.x**, un lenguaje altamente portable, compatible con múltiples sistemas operativos (Windows, Linux, macOS).
- La elección de Python permite migrar el sistema a otros entornos sin necesidad de reescribir el núcleo funcional.

Plataforma de desarrollo

- Se utiliza un entorno de desarrollo estándar basado en **IDEs como VS Code o PyCharm**, que son multiplataforma.
- No se emplean compiladores específicos, ya que Python es interpretado, lo que facilita su ejecución en cualquier máquina con el intérprete instalado.

Sistema operativo

- El sistema ha sido probado y validado en **Windows 10 y superior**, pero puede ejecutarse en:
Linux (Ubuntu, Debian, Fedora) con instalación de Python y SQLite
macOS con configuración mínima del entorno Python

- La única adaptación requerida es la configuración de rutas de archivos y permisos de ejecución.

Estructura modular

- El sistema está organizado en módulos independientes (clientes, reservas, pagos, reportes), lo que permite:
 - Migrar funcionalidades específicas a otros entornos sin afectar el resto del sistema
 - Reutilizar componentes en futuras versiones web o móviles

Migración futura

- En versiones futuras, se contempla la migración a una arquitectura cliente-servidor o basada en microservicios, con:
 - Backend en Python (Flask/Django)
 - Frontend web o móvil
 - Base de datos relacional (PostgreSQL/MySQL)
 - Despliegue en contenedores Docker para portabilidad total

3.4 Otros requisitos

Esta sección contempla requisitos adicionales que no se encuadran directamente en las categorías anteriores, pero que son relevantes para el correcto funcionamiento, aceptación y cumplimiento normativo del sistema.

Requisitos legales

- El sistema debe cumplir con la **Ley 1581 de 2012** de protección de datos personales en Colombia, garantizando:
 - Consentimiento informado para el registro de datos de clientes
 - Acceso restringido a información sensible
 - Posibilidad de rectificación o eliminación de datos a solicitud del titular
- Los registros financieros deben conservarse por un período mínimo de **5 años**, conforme a las normas contables vigentes.
- El sistema debe permitir la **trazabilidad de operaciones** para auditorías internas y externas.

Requisitos culturales y lingüísticos

- La interfaz del sistema debe estar completamente en **español neutro**, evitando regionalismos o tecnicismos innecesarios.
- Los mensajes del sistema deben ser **claros, respetuosos y culturalmente apropiados**, especialmente en interacciones con clientes.
- En futuras versiones, se contempla la posibilidad de **multilinguaje**, priorizando inglés como segundo idioma.

Requisitos de localización

- El sistema debe permitir el uso de **formato de fecha local (dd/mm/aaaa)** y moneda en **pesos colombianos (COP)**.
- Los reportes deben incluir encabezados y etiquetas adaptadas al contexto operativo del hotel.

Requisitos organizacionales

- El sistema debe adaptarse a la **estructura operativa del Hotel Boutique El Refugio**, respetando los roles definidos en el documento de análisis.
- Las funcionalidades deben alinearse con los procesos internos del hotel, sin imponer cambios drásticos en la forma de trabajo del personal.

Requisitos de aceptación

- El sistema será considerado aceptado cuando:
 - Cumpla con todos los requisitos funcionales y no funcionales definidos en este documento
 - Sea validado por el equipo de recepción, administración y gerencia del hotel
 - Pase las pruebas de funcionalidad, rendimiento y seguridad sin errores críticos

4 Apéndices

Esta sección incluye información adicional relevante para el desarrollo, validación y comprensión del sistema, que no forma parte directa del cuerpo principal del documento SRS pero que aporta valor técnico, organizacional o documental.

Glosario de términos

Término	Definición
Cliente	Persona que realiza una reserva en el hotel
Reserva	Registro que asocia un cliente con una habitación en un rango de fechas
Recepcionista	Usuario del sistema encargado de gestionar clientes y reservas
Administrador	Usuario con acceso a reportes, pagos y configuración del sistema
Auditor	Usuario con acceso a logs y trazabilidad de operaciones
SRS	Software Requirements Specification (Especificación de Requisitos de Software)
SQLite	Motor de base de datos ligero y embebido utilizado en el sistema
MTBF	Mean Time Between Failures (Tiempo medio entre fallos)
MTTR	Mean Time To Recovery (Tiempo medio de recuperación)

Referencias normativas

- IEEE 830 – Recomendaciones para la especificación de requisitos de software
- Ley 1581 de 2012 – Protección de datos personales (Colombia)
- Normas contables locales – Conservación de registros financieros
- Guía Scrum – Marco de trabajo ágil utilizado en el proyecto

Estructura de base de datos (resumen)

- clientes(id, nombre, documento, telefono, correo)
- habitaciones(id, tipo, precio, estado)
- reservas(id, cliente_id, habitacion_id, fecha_entrada, fecha_salida)
- pagos(id, reserva_id, monto, metodo, estado)
- usuarios(id, usuario, contraseña, rol)
- logs(id, usuario_id, fecha, operación, módulo, resultado)

Casos de prueba sugeridos

- Registro de cliente con datos válidos e inválidos
- Consulta de disponibilidad en fechas ocupadas y libres
- Creación de reserva con habitación no disponible
- Registro de pago con método no permitido
- Acceso por rol con restricciones activas
- Generación de reporte sin datos en el rango solicitado

Consideraciones para futuras versiones

- Interfaz gráfica web y móvil
- Integración con pasarelas de pago
- Exportación de reportes en PDF/CSV
- Autenticación multifactor
- Migración a base de datos relacional (PostgreSQL)
- Despliegue en contenedores (Docker)