HOTEL SIL

System-Wide Requirements Specification

# Introduction

# System-Wide Functional Requirements.

* Todo usuario del sistema debe estar registrado y autenticado.
* El comité administrativo pueda obtener la información actualizada de lo que requiera

|  |
| --- |
| Registrar, buscar y eliminar clientes y habitaciones. |
| Crear Cliente Persona Jurídica |
| Crear Cliente Persona Natural |
| Realizar reserva. |
| Realizar Chek-In |
| Gestionar Clientes |
| Consultar estado de habitaciones |

* El cliente en la aplicación móvil, podrá consultar la disponibilidad de los servicios y puede realizar una reserva de estos.

|  |
| --- |
| Realizar reservas de habitación |
| Realizar consultas de cada servicio |
| Realizar consultas de los horarios de atención |
| Realizar consulta de los precios de cada servicio. |

# System Qualities

## Usability

* La interfaz de usuario debe ser orientada a ventanas, y al manejo del programa a través del teclado y del ratón.
* La interfaz del sistema es intuitiva, de fácil uso y entrenamiento.
* Muestra mensajes de error.
* No podrá cerrarse una aplicación hasta no concluir con las demás.
* Se aplicaran algunos puntos de los siguientes estándares, para el desarrollo del sistema:
* Los estándares utilizados ISO 14915 para la Ergonomía de software para interfaz multimedia
* ISO 11064: Diseño ergonómico para centros de control.
* IEC TR 61997: Guías de interfaz de usuario en equipos multimedia de uso general.

## Reliability

* Existirá una persona indicada de brindar soporte, en el caso de que sea requerido.
* El sistema tendrá un tiempo de evaluación del sistema de seis meses, en el cual el usuario reportara de algún error, falló en el mismo.
* Se realizaran un BackUp cada cierto periodo para asegurar la información.

## Performance

* El tiempo de respuesta a consultas, actualizaciones, altas, modificaciones y bajas ha de

ser inferior a 10 segundos.

* El inicio del sistema será de 5 minutos como máximo
* El tiempo que el sistema tarda en apagarse es de 3 minutos como amximo

## Supportability

El sistema está implementado en un estilo arquitectónico de cuatro capas, el cual nos permite:

* Abstracción ya que los cambios se realizan a alto nivel y se puede incrementar o reducir el nivel de abstracción que se usa en cada capa del modelo.
* Aislamiento ya que se pueden realizar actualizaciones en el interior de las capas sin que esto afecte al resto del sistema.
* Rendimiento ya que distribuyendo las capas en distintos niveles físicos se puede mejorar la escalabilidad, la tolerancia a fallos y el rendimiento.
* Testeabilidad ya que cada capa tiene una interfaz bien definida sobre la que realizar las pruebas y la habilidad de cambiar entre diferentes implementaciones de una capa.
* Independencia ya que elimina la necesidad de considerar el hardware y el despliegue así como las dependencias con interfaces externas.