



**Semana 1**

**Ingeniería de Software (PRY3211)**

Formato de respuesta

| **Nombre estudiante:** | |
| --- | --- |
| **Asignatura:** | **Carrera:** |
| **Profesor:** | **Fecha:** |

**Descripción de la actividad**

En esta primera semana realizarás una actividad formativa grupal por encargo llamada "Definiendo la visión inicial del proyecto para el desarrollo de una aplicación de software", donde tendrás que generar la definición de la visión inicial del Proyecto, del producto, sus objetivos y alcances en base a estándares de ingeniería de software, reconociendo metodologías de desarrollo tradicionales y criterios de Calidad. Además, reconocerás la utilidad de definir requerimientos bajo un documento técnico ERS (Especificación de Requisitos del Sistema) para definir las principales funcionalidades del sistema y sus propiedades, esto también integra una Planilla de Especificación de Requisitos.

**Instrucciones específicas**

A continuación, se te presenta el documento técnico ERS (Especificación de Requisitos del Sistema) deberás trabajarlo y al final recuerda subirlo a un ZIP con la Planilla de Especificación de Requisitos (Excel) y sube al AVA en el espacio correspondiente.

Icono

Descripción generada automáticamente

**Dato**

Para comenzar con tu trabajo colaborativo, crea un repositorio en alguna cuenta Google Drive con tu equipo en una carpeta con nombre “Proyecto-Nombre del proyecto”, donde realizarán la carga del documento ERS en la Carpeta y la Planilla de especificación de Requisitos en formato Planilla de cálculo Excel.

**Coloca el enlace del drive aquí para que el docente revise las versiones y participación de los/as integrantes del proyecto:** [**https://drive.google.com/drive/folders/1x9QuNU-js9fAEmU7UN5cn5OXtX3nnVZ1?usp=sharing**](https://drive.google.com/drive/folders/1x9QuNU-js9fAEmU7UN5cn5OXtX3nnVZ1?usp=sharing)



**Links de Interés**

¿Sabías que puedes utilizar las herramientas disponibles en Google Drive accediendo con tu cuenta de estudiante Duoc?

Solo tienes que ingresar a: <https://drive.google.com/>

| [ESCRIBIR EL NOMBRE DE LA COMPAÑÍA] |
| --- |
| Especificación de requisitos de software |
| *Proyecto: [Nombre del proyecto]* |
|  |
| **Revisión*: [99.99]*** |
| **[Seleccionar fecha]** |

| Especificación de Requisitos según estándar de IEEE 830. |
| --- |

**Contenido**

Contenido

[Descripción de la actividad 8](#_heading=h.30j0zll)

[Instrucciones específicas 8](#_heading=h.1fob9te)

[Contenido 11](#_heading=h.tyjcwt)

[Ficha del documento 12](#_heading=h.3dy6vkm)

[1. Introducción 13](#_heading=h.1t3h5sf)

[1.1. Propósito 13](#_heading=h.4d34og8)

[1.2. Ámbito del Sistema 13](#_heading=h.2s8eyo1)

[1.3. Definiciones, Acrónimos y Abreviaturas 13](#_heading=h.17dp8vu)

[1.4. Referencias 13](#_heading=h.3rdcrjn)

[1.5. Visión General del Documento 14](#_heading=h.26in1rg)

[2.](#_heading=h.lnxbz9) Descripción General 15

[2.1. Perspectiva del Producto 15](#_heading=h.35nkun2)

[2.2. Funciones del Producto 15](#_heading=h.1ksv4uv)

[2.3. Características de los Usuarios 15](#_heading=h.44sinio)

[2.4. Restricciones 16](#_heading=h.2jxsxqh)

[2.5. Suposiciones y Dependencias 16](#_heading=h.z337ya)

[2.6. Requisitos Futuros 17](#_heading=h.3j2qqm3)

[3.](#_heading=h.1y810tw) Requisitos Específicos 18

[3.1](#_heading=h.4i7ojhp) Requisitos comunes de los interfaces 20

[3.1.1](#_heading=h.2xcytpi)  Interfaces de usuario 20

[3.1.2](#_heading=h.1ci93xb) Interfaces de hardware 20

[3.1.3](#_heading=h.3whwml4) Interfaces de software 20

[3.1.4](#_heading=h.2bn6wsx)  Interfaces de comunicación 20

[3.2](#_heading=h.qsh70q) Requisitos funcionales 21

[3.3](#_heading=h.3as4poj) Requisitos no funcionales 22

[3.3.1](#_heading=h.1pxezwc)  Requisitos de rendimiento 22

[3.3.2](#_heading=h.49x2ik5) Seguridad 22

[3.3.3](#_heading=h.2p2csry) Fiabilidad 22

[3.3.4](#_heading=h.147n2zr)  Disponibilidad 22

[3.3.5](#_heading=h.3o7alnk) Mantenibilidad 23

[3.3.6](#_heading=h.23ckvvd) Portabilidad 23

[3.4](#_heading=h.ihv636) Planilla de especificación de Requisitos 23

[**Dato** 24](#_heading=h.3znysh7)

[**Links de Interés** 24](#_heading=h.2et92p0)

**Ficha del documento**

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Documento validado por las partes en fecha:

| Por el cliente |  | Por la empresa suministradora |
| --- | --- | --- |
| [Firma] |  | [Firma] |
| Sr./Sra. |  | Sr./Sra. |

**1. Introducción**

En esta sección se proporcionará una introducción a todo el documento de Especificación de Requisitos Software (ERS). Consta de varias subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento.

1.1. Propósito

El propósito de este documento es el de sentar una base sólida que sirva como medio de comunicación entre clientes, usuarios y desarrolladores, esto con el fin de desarrollar exitosamente y en conjunto el sistema.

1.2. Ámbito del Sistema

El sistema, denominado "Gestión de Reserva Hotelera", permitirá a los clientes realizar reservas de habitaciones y a los empleados del hotel gestionar estas reservas. No incluirá funcionalidades relacionadas con la gestión de recursos humanos o actividades ajenas a la reserva y registro de hospedaje. Se espera que el sistema mejore la eficiencia y la experiencia de reserva para los clientes, así como la gestión interna para el personal del hotel.

1.3. Definiciones, Acrónimos y Abreviaturas

**HTTP**

El **HTTP** es la abreviatura de “protocolo de transferencia de hipertexto” en inglés, un protocolo subyacente utilizado por la [World Wide Web](https://www.arimetrics.com/glosario-digital/w3c). El HTTP define cómo se formatean y transmiten los mensajes y qué acciones deben tomar los servidores web y navegadores en respuesta a varios comandos.

**NAVEGADOR O BROWSER**

Un **navegado**r, browser o navegador web es un **software o aplicación informática** que mediante una interfaz gráfica permite a los usuarios desplazarse o **navegar** a través de la web a través de hipervínculos y direcciones URL. El navegador comunica con los servidores mediante lenguaje [HTTP](https://www.arimetrics.com/glosario-digital/http), recibe archivos en código HTML, lo traduce y lo muestra en la pantalla al usuario.

**API**

Las API son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos. Por ejemplo, el sistema de software del instituto de meteorología contiene datos meteorológicos diarios. La aplicación meteorológica de su teléfono “habla” con este sistema a través de las API y le muestra las actualizaciones meteorológicas diarias en su teléfono.

1.4. Referencias

En esta subsección se mostrará una lista completa de todos los documentos referenciados en la ERS.

1.5. Visión General del Documento

Este documento consta de 3 secciones. Esta sección es la introducción y proporciona una visión general de la ERS. En la Sección 2 se da una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles. En la sección 3 se definen detalladamente los requisitos que debe satisfacer el sistema.

**2. Descripción General**

En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto. Esto permitirá definir con detalle los requisitos en la sección 3, haciendo que sean más fáciles de entender.

Normalmente, esta sección consta de las siguientes subsecciones: Perspectiva del producto, funciones del producto, características de los usuarios, restricciones, factores que se asumen y futuros requisitos.

2.1. Perspectiva del Producto

El sistema "Gestión de Reserva Hotelera" es independiente de otros sistemas y se implementará como un producto autónomo. No obstante, puede interactuar con sistemas de terceros, como plataformas de pago o sistemas de correo electrónico para confirmaciones de reserva.

2.2. Funciones del Producto

El sistema permitirá a los clientes consultar disponibilidad de habitaciones, realizar reservas, gestionar sus cuentas, y recibir confirmaciones de reserva. Para el personal del hotel, proporcionará herramientas para gestionar reservas, actualizar información de habitaciones, y generar reportes de reservas.

2.3. Características de los Usuarios

Los usuarios incluyen tanto a clientes del hotel como a empleados. Los clientes pueden ser de diferentes niveles de experiencia y familiaridad con la tecnología, mientras que los empleados deben tener conocimientos básicos de informática y formación específica en el uso del sistema.

2.4. Restricciones

Esta subsección describe aquellas limitaciones que se imponen sobre los desarrolladores del producto:

* El sistema debe cumplir con las políticas y regulaciones de la empresa.
* Debe ser compatible con los sistemas hardware existentes en el hotel.
* Debe integrarse con otras aplicaciones, como sistemas de pago y servicios de correo electrónico.
* Se espera que el sistema cumpla con altos estándares de seguridad y disponibilidad.

2.5. Suposiciones y Dependencias

* Se asume que los usuarios recibirán la formación necesaria para utilizar eficazmente el sistema.
* El sistema depende de la disponibilidad y estabilidad de la conexión a internet para su funcionamiento adecuado.
* Los requisitos del sistema pueden estar sujetos a cambios según las necesidades del hotel y las actualizaciones tecnológicas.

2.6. Requisitos Futuros

Se espera que en futuras iteraciones del desarrollo se implementen mejoras adicionales, como la integración con sistemas de gestión de eventos.

**3. Requisitos Específicos**

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son **esenciales**, es decir, no sería aceptable un sistema que no satisfaga alguno de los requisitos aquí presentados. Estos requisitos se han especificado teniendo en cuenta, entre otros, el criterio de “testabilidad”: dado un requisito, debería ser fácilmente demostrable si es satisfecho o no por el sistema.

3.1 Requisitos comunes de los interfaces

Descripción detallada de todas las entradas y salidas del sistema de software.

**3.1.1 Interfaces de usuario**

El sistema debe proporcionar una interfaz de usuario intuitiva y fácil de usar, con opciones claras para realizar consultas y reservas de habitaciones. Además debe ser multilingüe, con opciones en inglés o español.

**3.1.2 Interfaces de hardware**

El sistema debe ser compatible con computadoras y smartphones (android y ios).

**3.1.3 Interfaces de software**

El sistema debe integrarse con sistemas de pago externos y servicios de correo electrónico para confirmaciones de reserva.

**3.1.4 Interfaces de comunicación**

El sistema debe comunicarse de forma segura y eficiente con sistemas externos a través de protocolos de comunicación estándar en este caso **HTTP**.

3.2 Requisitos funcionales

**RF-1 Consultar disponibilidad de habitaciones por fechas**

* Comprobar la validez de la fecha ingresada.
* Secuencia exacta: Mostrar una lista de habitaciones disponibles para las fechas seleccionadas.
* Respuesta a situaciones anormales: Informar si no hay habitaciones disponibles para las fechas ingresadas.
* Parámetros: Fechas de entrada y salida.
* Generación de salidas: Lista de habitaciones disponibles.

**RF-2 Ver disponibilidad de habitaciones y Equipamiento**

* Comprobar la validez de la habitación seleccionada.
* Secuencia exacta: Mostrar información detallada de la habitación seleccionada, incluyendo su equipamiento.
* Respuesta a situaciones anormales: Informar si la habitación seleccionada no está disponible.
* Parámetros: Habitación seleccionada.
* Generación de salidas: Información detallada de la habitación.

**RF-3 Registrar reserva de habitación los días que se requiera**

* Comprobar la validez de la fecha de reserva y la disponibilidad de la habitación.
* Secuencia exacta: Procesar la reserva de la habitación para los días seleccionados.
* Respuesta a situaciones anormales: Informar si la habitación no está disponible para los días seleccionados.
* Parámetros: Habitación seleccionada, fechas de reserva.
* Generación de salidas: Confirmación de reserva.

**RF-4 Calcular el pago de la reserva**

* Comprobar la validez de la información de reserva.
* Secuencia exacta: Calcular el monto a pagar por la reserva.
* Respuesta a situaciones anormales: Manejar errores en los cálculos de precios.
* Parámetros: Información de reserva.
* Generación de salidas: Monto a pagar por la reserva.

**RF-5 Implementar método de pago y generar ticket de reserva pagada**

* Comprobar la validez de la información de pago.
* Secuencia exacta: Procesar el método de pago y generar un ticket de reserva pagada.
* Respuesta a situaciones anormales: Manejar errores en el proceso de pago.
* Parámetros: Información de pago.
* Generación de salidas: Ticket de reserva pagada.

**RF-6 Contar con un Catálogo virtual de las habitaciones con su información pertinente**

* Comprobar la validez de la información solicitada.
* Secuencia exacta: Mostrar un catálogo virtual de las habitaciones con detalles.
* Respuesta a situaciones anormales: Informar si no se puede acceder al catálogo.
* Parámetros: N/A
* Generación de salidas: Catálogo virtual de habitaciones.

**RF-7 Registro y administración de datos de clientes**

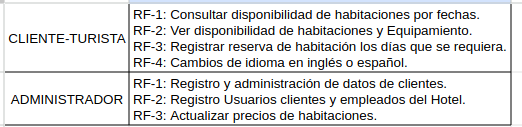
* Comprobar la validez de la información de registro.
* Secuencia exacta: Registrar y administrar los datos de los clientes.
* Respuesta a situaciones anormales: Manejar errores en el proceso de registro.
* Parámetros: Información de cliente.
* Generación de salidas: Confirmación de registro.

**RF-8 Registro Usuarios clientes y empleados del Hotel**

* Comprobar la validez de la información de registro.
* Secuencia exacta: Registrar y administrar los datos de los usuarios del sistema.
* Respuesta a situaciones anormales: Manejar errores en el proceso de registro.
* Parámetros: Información de usuario.
* Generación de salidas: Confirmación de registro.

**RF-9 Cada Usuario debe iniciar sesión antes de interactuar en el sistema**

* Comprobar la validez de las credenciales de inicio de sesión.
* Secuencia exacta: Permitir el acceso al sistema después de una verificación exitosa.
* Respuesta a situaciones anormales: Denegar el acceso en caso de credenciales incorrectas.
* Parámetros: Credenciales de inicio de sesión.
* Generación de salidas: Acceso al sistema.



3.3 Requisitos no funcionales

**3.3.1 Requisitos de rendimiento**

El sistema de Gestión de Reserva Hotelera debe ser capaz de manejar una carga de trabajo significativa para garantizar una experiencia fluida para los usuarios. A continuación se detallan los requisitos de rendimiento específicos:

* El sistema debe ser capaz de soportar un mínimo de 100 terminales simultáneamente conectados.
* Se espera un pico de hasta 20 usuarios concurrentes realizando consultas y reservas durante las horas pico.
* El tiempo de respuesta promedio para consultas de disponibilidad de habitaciones no debe exceder los 2 segundos.
* El 95% de las transacciones de reserva de habitaciones deben completarse en menos de 3 segundos.
* El sistema debe ser capaz de procesar un mínimo de 20 transacciones por segundo durante períodos de alta demanda.
* La disponibilidad del sistema debe ser del 99.9%, excluyendo períodos de mantenimiento programado.

**3.3.2 Seguridad**

1. El sistema deberá usar HTTPS para el transporte seguro de la data.
2. Se aplicará el principio de mínimo privilegio, donde cada usuario y módulo del sistema tendrá acceso solo a las funcionalidades y datos necesarios para realizar sus tareas específicas. Esto minimizará el riesgo de accesos no autorizados y limitará el impacto de posibles ataques.
3. Se implementarán controles de acceso y mecanismos de autenticación para garantizar que solo los usuarios autorizados puedan acceder a ciertas partes del sistema. Se establecerán políticas de seguridad para regular la comunicación entre los diferentes módulos del sistema y prevenir la propagación de amenazas de seguridad.

**3.3.3 Fiabilidad**

El sistema debe ser altamente confiable y estar disponible las 24 horas del día, los 7 días de la semana, para garantizar que los usuarios puedan realizar reservas en cualquier momento.

**3.3.4 Disponibilidad**

El sistema debe tener una disponibilidad del 99.99% para garantizar que esté siempre accesible para los usuarios.

**3.3.5 Mantenibilidad**

Mantenimiento correctivo de software: Se espera que ante cualquier error en el sistema, luego de descubrirse el error, el equipo de desarrollo de una solución lo antes posible.

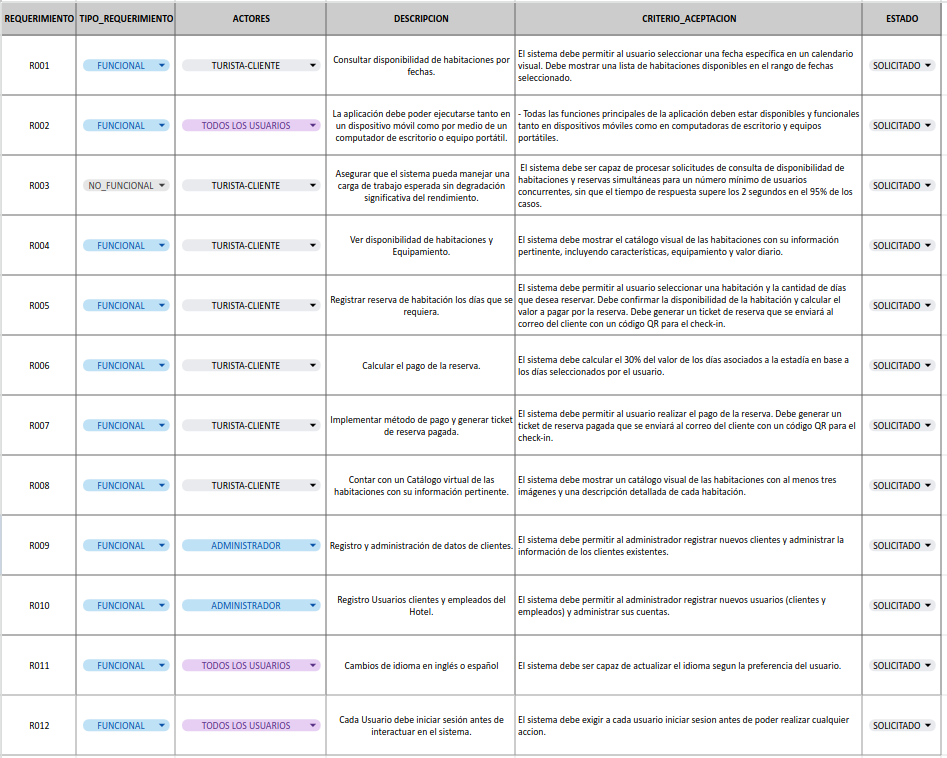
Mantenimiento adaptable de software: Se espera que todo el equipo realice un mantenimiento del sistema, ante cualquier cambio en el sistema.

**3.3.6 Portabilidad**

El sistema de Gestión de Reserva Hotelera debe ser diseñado teniendo en cuenta la portabilidad para facilitar su traslado a otras plataformas o entornos. A continuación se detallan los atributos que deben presentar el software para garantizar su portabilidad:

1. **Porcentaje de componentes dependientes del servidor:**
   * Se espera que la mayoría de los componentes del sistema sean independientes del servidor para facilitar su portabilidad. Idealmente, sólo los componentes de acceso a la base de datos estarán directamente vinculados al servidor.
2. **Porcentaje de código dependiente del servidor:**
   * Se espera que una parte mínima del código del sistema dependa directamente del servidor. La interfaz de usuario debe ser independiente del servidor para garantizar una mayor portabilidad.
3. **Uso de un determinado lenguaje por su portabilidad:**
   * El sistema debe estar diseñado utilizando un lenguaje de programación que sea conocido por su portabilidad, lo que permite que el software pueda ejecutarse en diferentes plataformas sin necesidad de modificaciones significativas.
4. **Uso de un determinado compilador o plataforma de desarrollo:**
   * Se debe utilizar un compilador o una plataforma de desarrollo que permita generar código que sea compatible con múltiples sistemas operativos y arquitecturas de hardware.
5. **Uso de un determinado sistema operativo:**
   * El sistema debe ser compatible con una amplia gama de sistemas operativos para garantizar su disponibilidad en diferentes entornos. Idealmente, el software debería funcionar sin problemas en sistemas operativos populares como Windows, macOS y Linux.

3.4 Planilla de especificación de Requisitos







**Duoc UC**