|  |
| --- |
| DUOC UC - Escuela de informática y telecomunicaciones |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| Proyecto: SMARTFLOW – SISTEMA DE GESTIÓN Y RESERVAS |
|  |
| **Revisión*: [01]*** |
| **18/08/2025** |

|  |
| --- |
| Planificación y Especificación de Requisitos según estándares; IEEE 830, ISO9000 y PMI. |

Contenido

[DUOC UC - Escuela de informática y telecomunicaciones 1](#_Toc97387164)

[Ficha del documento 4](#_Toc97387165)

[1. Introducción 5](#_Toc97387166)

[1.1. Propósito 5](#_Toc97387167)

[1.2. Ámbito del Sistema 5](#_Toc97387168)

[1.3. Definiciones, Acrónimos y Abreviaturas 5](#_Toc97387169)

[1.4. Referencias 5](#_Toc97387170)

[1.5. Visión General del Documento 5](#_Toc97387171)

[2. Descripción General 6](#_Toc97387172)

[2.1. Perspectiva del Producto 6](#_Toc97387173)

[2.2. Funciones del Producto 6](#_Toc97387174)

[2.3. Características de los Usuarios 6](#_Toc97387175)

[2.4. Restricciones 6](#_Toc97387176)

[2.5. Suposiciones y Dependencias 7](#_Toc97387177)

[2.6. Requisitos Futuros 7](#_Toc97387178)

[3. Requisitos Específicos 8](#_Toc97387179)

[3.1 Requisitos comunes de las interfaces 8](#_Toc97387180)

[3.1.1 Interfaces de usuario 8](#_Toc97387181)

[3.1.2 Interfaces de hardware 8](#_Toc97387182)

[3.1.3 Interfaces de software 8](#_Toc97387183)

[3.2 Requisitos funcionales 9](#_Toc97387184)

[3.3 Requisitos no funcionales (Organización) y de calidad (Producto) 9](#_Toc97387185)

[3.3.1 Requisitos de Rendimiento o Eficiencia 9](#_Toc97387186)

[3.3.2 Requisitos de Seguridad 9](#_Toc97387187)

[3.3.3 Requisitos de Usabilidad 9](#_Toc97387188)

[3.3.4 Requisitos de Disponibilidad 9](#_Toc97387189)

[3.3.5 Requisitos de Portabilidad 9](#_Toc97387190)

[3.3.6 Requisitos de Mantenibilidad 9](#_Toc97387191)

[3.3.7 Requisitos de Funcionalidad 9](#_Toc97387192)

[3.4 Requisitos No funcionales Organizacionales 10](#_Toc97387193)

[4. Propuesta de Planificación 11](#_Toc97387194)

[4.1 Descripción general acerca de la Planificación 11](#_Toc97387195)

[4.1.2 Definición del Equipo de Trabajo 11](#_Toc97387196)

[4.1.3 Definición de Actividades principales del Proyecto 11](#_Toc97387197)

[4.1.4 Resumen Costos del Desarrollo del Proyecto 11](#_Toc97387198)

[4.1.6 Carta Gantt 11](#_Toc97387199)

[5. Anexos 12](#_Toc97387200)

[5.1 Acta de Proyecto 12](#_Toc97387201)

[5.2 Matriz Especificación de Requerimientos 12](#_Toc97387202)

[5.3 Prototipado de Software 12](#_Toc97387203)

[5.4 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo 12](#_Toc97387204)

[5.5 Planilla Carta Gantt 12](#_Toc97387205)

# Ficha del documento

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| *18.08.2025* | *1.0* | Mallely Calfilaf Aguayo | *Creación inicial del documento. Incluye acta, requerimientos y casos de uso.* |
| *20.08.2025* | *1.1* | Juan Castillo Loyola | *Creacion de minuta de reunion* |

Documento validado por las partes en fecha:

**Integrantes:**

|  |  |
| --- | --- |
| **Nombre Integrante del Equipo** | **Rol Definido** |
| Juan Castillo Loyola | Jefe de Proyecto |
| Mallely Calfilaf Aguayo | Gerente de Proyecto |
| Banco Santander | Patrocinador Principal |
| Víctor Godoy | Patrocinador |
| Duoc UC | Representante Académico |

# 1. Introducción

En este documento se presenta la Especificación de Requerimientos de Software (ERS) del sistema SMARTFLOW, desarrollado en el marco del Portafolio de Título en Duoc UC. El propósito es definir con precisión las funcionalidades, restricciones y lineamientos que guiarán el desarrollo de la aplicación, asegurando que satisfaga las necesidades de los usuarios y cumpla con los estándares de calidad establecidos.

## 1.1. Propósito

## El propósito de este documento es describir de manera clara y detallada los requerimientos del sistema SMARTFLOW, el cual será una aplicación web orientada a la gestión de procesos administrativos, reservas y generación de reportes.Este documento está dirigido a:

## Equipo de desarrollo: servirá como guía para la implementación técnica del sistema.

## Stakeholders (clientes y usuarios finales): permitirá validar que los requerimientos cubren las necesidades planteadas.

## Docentes evaluadores de Duoc UC: asegurarán que el proyecto cumple con los lineamientos metodológicos y académicos.

## 1.2. Ámbito del Sistema

## Lo que hará el sistema:

## Permitirá a los usuarios autenticarse y gestionar sus perfiles.

## Contará con al menos 8 mantenedores que otorguen autonomía al administrador.

## Incluirá un módulo de reservas que permitirá gestionar horarios, recursos o salas.

## Generará reportes con filtros exportables en PDF y XLS.

## Incorporará notificaciones por correo electrónico o dispositivos móviles.

## Será responsivo, garantizando una adecuada visualización en distintos dispositivos.

## Lo que no hará el sistema:

## No gestionará pagos en línea.

## No tendrá aplicación móvil nativa (solo versión web responsiva).

## No se integrará con sistemas externos de terceros.

## Beneficios, objetivos y metas esperadas:

## Optimización de los procesos administrativos.

## Facilitar la autogestión de reservas por parte de los usuarios.

## Generar reportes confiables y rápidos para la toma de decisiones.

## Asegurar escalabilidad y mantenibilidad del sistema a futuro.

## 1.3. Definiciones, Acrónimos y Abreviaturas

## ERS: Especificación de Requerimientos de Software.

## SMARTFLOW: Nombre del sistema desarrollado en el marco del Portafolio de Título.

## CRUD: Operaciones básicas de creación, lectura, actualización y eliminación de datos.

## Mockup: Prototipo gráfico de una interfaz de usuario.

## PMI: Project Management Institute, marco de referencia en gestión de proyectos.

## IEEE 830: Estándar de IEEE para especificaciones de requerimientos de software.

## ISO 9000: Normativa internacional de gestión de calidad.

## Stakeholders: Interesados en el proyecto (clientes, usuarios, administradores, docentes evaluadores).

## 1.4. Referencias

El desarrollo de este documento se basa en los siguientes lineamientos y fuentes:

* IEEE Std 830-1998: Recommended Practice for Software Requirements Specifications.
* ISO 9000: Normas de gestión de calidad aplicables al desarrollo de software.
* PMI – Project Management Body of Knowledge (PMBOK).
* Documentación del proyecto SMARTFLOW:
  + Acta de Constitución.
  + Documento de Casos de Uso Extendido.
  + Mockups del Sistema.
  + Matriz EDT.
  + Carta Gantt.

## 1.5. Visión General del Documento

Se organiza de la siguiente forma:

* **Sección 1 – Introducción**: establece el propósito, ámbito, definiciones, referencias y visión general del documento.
* **Sección 2 – Descripción General**: presenta la perspectiva del sistema, sus funciones principales, tipos de usuarios, restricciones y dependencias.
* **Sección 3 – Requisitos Específicos**: detalla los requerimientos funcionales y no funcionales del sistema, incluyendo interfaces y casos de uso.
* **Sección 4 – Propuesta de Planificación**: expone las fases principales del proyecto, equipo de trabajo, costos estimados y carta Gantt.
* **Sección 5 – Anexos**: incluye documentos complementarios como acta de constitución, prototipos gráficos, matriz EDT y planillas de control.

# 2. Descripción General

## 2.1. Perspectiva del Producto

El sistema SMARTFLOW se desarrollará como una aplicación web basada en arquitectura de capas. Será independiente de otros sistemas externos, aunque podrá escalarse en el futuro para integraciones. El sistema tendrá un módulo principal (gestión de reservas y procesos) y módulos secundarios de administración y reportes..

## 2.2. Funciones del Producto

SMARTFLOW ofrecerá las siguientes funciones principales:

* Autenticación y control de accesos por roles.
* Gestión de usuarios (administrador, estudiante/cliente, funcionario).
* Mantenedores (mínimo 8 CRUD) para gestión de datos.
* Módulo de reservas para asignación de horarios y recursos.
* Generación de reportes con filtros avanzados, exportables a PDF y XLS.
* Envío de notificaciones por correo electrónico o mensaje a dispositivo móvil.
* Panel administrativo para seguimiento y control.

## 2.3. Características de los Usuarios

* **Administrador del sistema:** Gestiona usuarios, mantenedores y disponibilidad de recursos.
* **Usuario general (estudiante/cliente):** Realiza reservas, gestiona su perfil y consulta reportes personales.
* **Funcionario/Coordinador:** Supervisa la correcta utilización de recursos, valida o rechaza solicitudes.
* **Docente evaluador (rol académico):** Accede a reportes para verificar la implementación del sistema.

Los usuarios tendrán distintos niveles de experiencia técnica, por lo que la interfaz será sencilla, responsiva y con guías visuales intuitivas.

## 2.4. Restricciones

El sistema debe ser desarrollado en un lenguaje orientado a objetos (ejemplo: .NET Core o Java).

La base de datos será SQL Server.

La interfaz será desarrollada con frameworks web modernos razar

La seguridad debe garantizarse mediante autenticación de usuarios y enmascaramiento de contraseñas.

El sistema debe estar disponible para navegadores web modernos (Chrome, Edge, Firefox).

## 2.5. Suposiciones y Dependencias

Los usuarios disponen de conexión a internet estable para acceder al sistema.

Los administradores mantendrán actualizados los datos en los mantenedores.

Se asume que el uso del sistema se realizará dentro del contexto académico (Duoc UC) pero puede adaptarse a entornos organizacionales similares.

En futuras versiones se podría integrar con otros sistemas institucionales, como Google Calendar o sistemas internos de Duoc UC.

## 2.6. Requisitos Futuros

Incorporación de aplicación móvil nativa para Android/iOS.

Integración con pasarelas de pago para reservas de servicios con costo

Implementación de un módulo de analítica avanzada para predecir la demanda de reservas.

Integración con herramientas de Business Intelligence (Power BI o similares).

# 3. Requisitos Específicos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:

• El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.

• Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.

• Todo requisito deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.

## 3.1 Requisitos comunes de las interfaces

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

### 3.1.1 Interfaces de usuario

### El sistema debe contar con una interfaz web responsiva accesible desde computadores,

### tablets y smartphones.

### El diseño debe ser intuitivo, minimalista y responsivo, adaptándose a diferentes tamaños de

### pantalla.

### Los formularios deben incluir validaciones de datos (campos obligatorios, formatos de correo, contraseñas seguras).

### Se debe proveer un menú de navegación claro para acceder a los módulos principales: Inicio, Reservas, Mantenedores, Reportes, Administración

### .3.1.2 Interfaces de hardware

### El sistema funcionará en computadores estándar con procesador Intel i5 o superior, mínimo

### 8 GB RAM y conexión a internet.

### Los dispositivos móviles deberán tener un navegador actualizado y conexión estable a

### internet.

### 3.1.3 Interfaces de software

El sistema debe integrarse con una base de datos SQL Server

Debe utilizar frameworks web modernos (Razor y .NET Core en backend).

El sistema debe poder exportar reportes a PDF y XLS.

Se debe implementar autenticación mediante librerías de seguridad estándar

## 3.2 Requisitos funcionales

Los requisitos funcionales deben estar enumerados (diagrama caso uso) y pueden ser divididos en sub-secciones.

**3.2.1 Autenticación, sesiones y roles**

* **RF01**: Iniciar/cerrar sesión con credenciales válidas.
* **RF02**: Cierre de sesión por inactividad configurable.
* **RF03**: Gestión de **roles y permisos** (crear/editar/asignar a usuarios).

**3.2.2 Gestión de usuarios (CRUD)**

* **RF04**: Crear, editar, desactivar/reactivar y consultar usuarios (con búsqueda y filtros).
* **RF05**: Validaciones en formularios de usuario (campos obligatorios, formato de correo).

**3.2.3 Servicios y parámetros del negocio (CRUD)**

* **RF06**: Crear/editar/eliminar/listar **servicios** (nombre, duración, cupos, estado).
* **RF07**: Consistencia de duración/cupos conforme a reglas del módulo de reservas.

**3.2.4 Calendario y reservas**

* **RF08**: Consultar **disponibilidad** por servicio/fecha (calendario con libres/ocupados).
* **RF09**: **Reservar** servicio (selección servicio, fecha y horario; verificación en tiempo real).
* **RF10**: **Evitar doble reserva** (bloqueo de horarios/recursos en conflicto).
* **RF11**: **Modificar/Cancelar** reserva con actualización inmediata de disponibilidad.

**3.2.5 Notificaciones**

* **RF12**: Envío automático de **confirmaciones, recordatorios y cambios/cancelaciones** por correo/móvil, disparado por eventos.
* **RF13**: Configuración administrativa de mensajes/plantillas.

**3.2.6 Reportes y exportación**

* **RF14**: Generar **reportes de reservas** con filtros (fecha, servicio, usuario, estado).
* **RF15**: Exportar reportes a **PDF** y **XLS** manteniendo integridad de datos.
* **RF16**: Mostrar reportes en pantalla con paginación y totales.
* **RF17**: Cumplir con el mínimo del curso: **dos reportes con filtros** exportables a PDF/XLS.

**3.2.7 Cálculo de ingreso de matrícula (si aplica a tu alcance)**

* **RF18**: Ingresar arancel, descuentos/becas y cuotas; **calcular ingreso total** y emitir comprobante.

**3.2.8 Auditoría**

* **RF19**: Registrar eventos (login, CRUD, reservas, exportaciones) con usuario, fecha/hora y detalle; consulta y exportación de log.

Nota de coherencia con la guía del curso: la solución debe incluir **≥ 8 mantenedores**, un **módulo de negocio completo** (reservas) y **autenticación/roles**, además de **validaciones de formularios** y **notificaciones**; todo **responsivo**. Esto queda exigido en tu plantilla de presentación inicial.

## 3.3 Requisitos no funcionales (Organización) y de calidad (Producto)

### 3.3.1 Requisitos de Rendimiento o Eficiencia

### RNF01: Tiempo de respuesta “consultar disponibilidad” y “reservar” ≤ 3 s en condiciones nominales.

### RNF02: La exportación de reportes debe completarse sin errores y en tiempo razonable (escala con filtros)

### 3.3.2 Requisitos de Seguridad

### RNF03: Enmascarar contraseñas y almacenarlas con hash + salt; uso de HTTPS en tránsito.

### RNF04: Control de acceso por roles/permisos en todas las funcionalidades sensibles.

### RNF05: Trazabilidad mediante auditoría (registro de eventos críticos)

### 3.3.3 Requisitos de Usabilidad

### RNF06: Diseño responsivo y navegación clara (PC, tablet, móvil).

### RNF07: Formularios con validaciones y mensajes de error comprensibles.

### 3.3.4 Requisitos de Disponibilidad

### RNF08: Disponibilidad operacional ≥ 95% en horario hábil; auditoría con alta disponibilidad sugiere meta superior.

### 3.3.5 Requisitos de Portabilidad

### RNF09: Soporte para navegadores modernos (Chrome, Edge, Firefox) sin instalación de plugins.

### 3.3.6 Requisitos de Mantenibilidad

### RNF10: Arquitectura en capas y módulos desacoplados (frontend/back‑end/BD) para facilitar cambios y pruebas.

### RNF11: Convenciones de código y documentación técnica mínima por módulo (README y diagramas).

### 3.3.7 Requisitos de Funcionalidad

RNF12: Funcionalidad completa según CU y plan mínimo exigido (8 mantenedores, módulo negocio, reportes).

## 3.4 Requisitos No funcionales Organizacionales

RNF13: Uso de colores/branding institucional en interfaz cuando aplique (logo y paleta).

# 4. Propuesta de Planificación

## 4.1 Descripción general acerca de la Planificación

El desarrollo del sistema **SMARTFLOW** se planificará en un período comprendido entre el **18 de agosto y diciembre de 2025**, con una duración aproximada de **16 semanas**. Durante este tiempo se abordarán las distintas etapas del proyecto: levantamiento de requerimientos, diseño, desarrollo, pruebas e implementación.

El trabajo será ejecutado bajo la metodología **ágil Scrum**, organizando el cronograma en **5 sprints de 3 semanas** cada uno, más una semana adicional destinada a integración final y entrega. Esta forma de trabajo permitirá un **avance iterativo y validado constantemente con el cliente (Duoc UC)**, reduciendo riesgos y garantizando que los entregables cumplan con los requerimientos establecidos.

En la ejecución estarán involucrados los siguientes perfiles:

* **Jefe de Proyecto**: encargado de la coordinación general, planificación y cumplimiento de plazos.
* **Analista de Requerimientos**: responsable del levantamiento, análisis y documentación de necesidades del cliente.
* **Diseñador UI/UX**: encargado de los mockups, la experiencia de usuario y la interfaz gráfica.
* **Desarrolladores Frontend y Backend**: responsables de la construcción de los módulos del sistema.
* **Tester/QA**: encargado de la validación, pruebas unitarias, integrales y de aceptación.

Para asegurar el buen término del proyecto se aplicarán las siguientes **buenas prácticas**:

* Uso de **control de versiones en GitHub**, con ramas por sprint y revisiones colaborativas.
* **Reuniones semanales de equipo** y **reuniones quincenales con el cliente** para validación de avances.
* Documentación constante y clara de cada fase del proyecto (ERS, mockups, minutas de reunión, reportes de sprint).
* Aplicación de pruebas continuas para garantizar la calidad del sistema.
* Comunicación efectiva y retroalimentación continua entre los miembros del equipo y con el cliente.

Las **condiciones necesarias para el éxito del proyecto** incluyen:

* Cumplimiento riguroso de los plazos definidos en el cronograma.
* Participación activa del cliente en la validación de requerimientos y entregables.
* Disponibilidad de los recursos técnicos necesarios (servidores, herramientas de desarrollo y prueba).
* Compromiso de cada integrante del equipo en sus roles y responsabilidades asignadas.

### 4.1.2 Definición del Equipo de Trabajo

|  |  |  |
| --- | --- | --- |
| Rol | Nombre / Responsable | Funciones principales |
| Jefe de Proyecto | Juan Castillo Loyola | - Coordinar todas las actividades del proyecto. - Supervisar cumplimiento de plazos. - Gestionar comunicación con el cliente (Duoc UC). |
| Analista de Requerimientos | [Asignado] | - Levantar y documentar requerimientos. - Mantener actualizado el ERS. - Validar requisitos con el cliente. |
| Diseñador UI/UX | [Asignado] | - Elaborar mockups y prototipos. - Diseñar la interfaz gráfica. - Optimizar la experiencia del usuario. |
| Desarrollador Frontend | [Asignado] | - Implementar el diseño en el sistema. - Desarrollar vistas y componentes interactivos. - Integrar frontend con backend. |
| Desarrollador Backend | [Asignado] | - Implementar la lógica del negocio. - Gestionar la base de datos. - Crear servicios y API necesarios. |
| Tester / QA | [Asignado] | - Diseñar y ejecutar planes de prueba. - Validar el correcto funcionamiento del sistema. - Documentar incidencias y dar retroalimentación. |
| Cliente / Product Owner | Duoc UC (representante) | - Validar los requerimientos. - Revisar avances en reuniones. - Aprobar entregables finales. |

### 4.1.3 Definición de Actividades principales del Proyecto

De acuerdo con la planificación establecida en la Carta Gantt y la Matriz EDT, las etapas principales del proyecto son:

1. **Inicio del Proyecto**
   * Elaboración del Acta de Constitución.
   * Definición de objetivos generales y específicos.
   * Asignación de roles y responsabilidades.
2. **Levantamiento de Requerimientos**
   * Identificación de usuarios y casos de uso.
   * Elaboración del documento ERS (Especificación de Requisitos de Software).
   * Validación de requerimientos con el cliente (DUOC UC).
3. **Análisis y Diseño**
   * Diagramas de casos de uso extendidos.
   * Mockups y prototipos de la interfaz.
   * Diseño de arquitectura de base de datos y lógica del sistema.
4. **Desarrollo**
   * Implementación del frontend (Razor).
   * Implementación del backend ( .NET Core).
   * Conexión con base de datos (SQL Server).
5. **Pruebas**
   * Pruebas unitarias y de integración.
   * Validación de funcionalidades principales (reservas, gestión de usuarios).
   * Ajustes según retroalimentación.
6. **Implementación**
   * Migración al entorno productivo.
   * Capacitación de usuarios finales.
   * Documentación final del sistema.
7. **Cierre del Proyecto**
   * Evaluación de cumplimiento de objetivos.
   * Informe final y entrega a DUOC UC.
   * Retrospectiva del equipo.

### 4.1.4 Resumen Costos del Desarrollo del Proyecto

* Costos por FASE
* Costos por Actor o Rol

### 4.1.6 Carta Gantt

Agregar las fechas más importantes definidas en la carta Gantt dejando claro los hitos o las fases principales y la fecha de término del proyecto

## 5. Anexos

### 5.1 Acta de Proyecto

Insertar Acta de Constitución del Proyecto

### 5.2 Matriz Especificación de Requerimientos

Insertar la Matriz en formato planilla sobre la especificación de Requerimientos

### 5.3 Prototipado de Software

Insertar documento con Mockups de las interfaces de usuario del Sistema

### 5.4 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo

[Insertar matriz EDT en formato Planilla que nos permite realizar el cálculo de estimación de esfuerzo en base a jornadas laborales.]

### 5.5 Planilla Carta Gantt

Insertar Excel de la carta Gantt