



# DOCUMENTO FUNCIONAL DEL PROYECTO

ALUMNO/GRUPO: Iván Junior Samper Fernández y Iker Ortuño Gutiérrez

---

## 1. Introducción y contexto

*Objetivo:* explicar de qué trata el proyecto y por qué es necesario.

Describe:

- El problema o necesidad que se quiere resolver.
- Quién será el usuario o cliente final.
- Qué solución se propone y con qué propósito.

*Ejemplo:*

Se desarrollará una aplicación web para gestionar las reservas del gimnasio del centro. Permitirá a los usuarios consultar y reservar espacios desde el móvil.

Se llevará a cabo el desarrollo de una página web para gestionar los servicios de mantenimiento y reparación de equipos informáticos y redes. Lo que permitirá a los usuarios solicitar soporte, agendar visitas técnicas y consultar el estado de sus solicitudes desde cualquier dispositivo. El propósito será garantizar un servicio rápido, accesible y confiable, reduciendo tiempos de inactividad, mejorando la productividad y facilitando la interacción entre el cliente y la empresa.

---

## 2. Análisis de requisitos

### 2.1. Requisitos funcionales (RF)

*Qué debe hacer el sistema.*

Enumera las funciones principales, numeradas como RF1, RF2, etc.

<b>Código</b>	<b>Descripción del requisito funcional</b>
RF1	El sistema permitirá registrar nuevos usuarios.
RF2	Los usuarios podrán iniciar sesión y gestionar su perfil.
RF3	El sistema mostrará las reservas activas del usuario.

<b>CÓDIGO</b>	<b>DESCRIPCIÓN</b>
RF1	El sistema permitirá registrar nuevos usuarios.
RF2	Los usuarios podrán iniciar sesión y gestionar su perfil.
RF3	El sistema permitirá solicitar servicios de mantenimiento o reparación.
RF4	El usuario podrá describir la incidencia y adjuntar fotos o documentos.
RF5	Los usuarios podrán consultar el estado de sus solicitudes.
RF6	Los administradores podrán gestionar solicitudes, asignar técnicos y actualizar estados.
RF7	El sistema permitirá visualizar el historial de servicios solicitados por cada usuario.
RF8	El sistema mostrará información sobre los servicios ofrecidos y tarifas.
RF9	El usuario podrá contactar con soporte mediante formulario o chat.
RF10	El sistema permitirá a los administradores gestionar usuarios, técnicos y contenido de la web.

---

## 2.2. Requisitos no funcionales (RNF)

*Cómo debe comportarse el sistema.*

Incluye aspectos como rendimiento, seguridad, compatibilidad o facilidad de uso.

<b>Código</b>	<b>Descripción del requisito no funcional</b>
RNF1	La interfaz será accesible desde dispositivos móviles.

RNF2	Las páginas deberán cargarse en menos de tres segundos.
------	---

CÓDIGO	DESCRIPCIÓN
RNF1	La interfaz será accesible desde dispositivos móviles.
RNF2	Las páginas deberán cargarse en menos de tres segundos.
RNF3	El sistema deberá garantizar la seguridad de los datos mediante cifrado en las comunicaciones.
RNF4	El sistema debe ser compatible con los principales navegadores.
RNF5	La plataforma deberá ser intuitiva y fácil de usar para cualquier tipo de usuario.
RNF6	Todas las acciones importantes requerirán autenticación del usuario.
RNF7	El sistema permitirá escalar con facilidad para soportar un aumento de usuarios o solicitudes.
RNF8	Los datos personales serán almacenados cumpliendo la normativa vigente de protección de datos.

### 2.3. Restricciones

*Condiciones o limitaciones del proyecto.*

- Lenguajes o tecnologías obligatorias:
- Recursos disponibles (tiempo, equipo, materiales):
- Dependencias o limitaciones técnicas:

#### Lenguajes o tecnologías obligatorias:

- El desarrollo de la web deberá realizarse utilizando HTML, Python y JavaScript.
- El backend deberá implementarse con Python.
- La base de datos será SQLite.
- El sistema deberá alojarse en un servidor con soporte para Ubuntu.

#### Recursos disponibles:

- Tiempo limitado para la entrega del proyecto, estimado 12 semanas.
- Equipo de trabajo reducido, compuesto por 2 desarrolladores.

- Recursos técnicos básicos: ordenadores, conexión a internet y servidor web.
- Presupuesto ajustado, por lo que no se podrán utilizar servicios de pago externos innecesarios.

**Dependencias o limitaciones técnicas:**

- La funcionalidad del sistema dependerá de la disponibilidad del servidor de hosting.
- Restricción de almacenamiento según el plan del servidor contratado.
- La plataforma dependerá del correcto funcionamiento del correo electrónico para enviar notificaciones.
- Posibles limitaciones de rendimiento si se utiliza hardware compartido o económico.

### 3. Análisis de usuarios y roles

*Objetivo:* identificar quién usará el sistema y qué podrá hacer.

Describe los distintos tipos de usuario, sus necesidades y sus permisos.

Rol	Descripción	Permisos principales
Administrador	Gestiona usuarios y recursos.	Alta, baja y modificación de datos.
Usuario	Utiliza el sistema para realizar acciones básicas.	Crear, consultar y modificar sus propios datos.
Visitante	Consulta información sin interactuar con el sistema	Sin permisos

ROL	DESCRIPCIÓN	PERMISOS
Administrador	Responsable de la gestión completa del sistema, usuarios, técnicos y solicitudes de servicio	Alta, baja y modificación de usuarios; gestión de solicitudes; asignación de técnicos; actualización de estados; administración de contenido de la web.
Técnico	Profesional encargado de revisar, reparar y dar soporte a las incidencias reportadas por los usuarios.	Consultar solicitudes asignadas; actualizar el estado del servicio; registrar diagnósticos, acciones realizadas y tiempos; comunicarse con el usuario.

Usuario registrado	Cliente que utiliza la plataforma para solicitar servicios de mantenimiento o reparación.	Crear solicitudes; adjuntar información; consultar el estado de sus servicios; modificar datos de su perfil; contactar con soporte.
--------------------	---	---

## 4. Casos de uso / Escenarios de uso

*Objetivo:* mostrar cómo interactúan los usuarios con el sistema.

Selecciona de tres a cinco casos principales y descríbelos brevemente.

Código	Nombre del caso de uso	Actor principal	Descripción	Resultado esperado
CU1	Registrar usuario	Visitante	El usuario introduce sus datos y se crea una cuenta.	Usuario registrado correctamente.
CU2	Iniciar sesión	Usuario	Introduce credenciales y accede al sistema.	Acceso concedido.
CU3	Realizar una reserva	Usuario	Selecciona día y hora para reservar un recurso.	Reserva creada.

Código	Caso	Actor	Descripción	Resultado
CU1	Registrar usuario	Visitante	El usuario introduce sus datos personales para crear una cuenta en la plataforma.	Usuario registrado correctamente.
CU2	Iniciar sesión	Usuario registrado	El usuario proporciona sus credenciales y accede al sistema.	Acceso concedido.
CU3	Crear solicitud	Usuario registrado	El usuario describe la	Solicitud registrada y en

			incidencia, adjunta información y envía la solicitud.	estado "pendiente".
CU4	Asignar solicitud	Administrador	El administrador revisa solicitudes pendientes y las asigna a un técnico disponible.	Solicitud asignada correctamente.
CU5	Actualizar estado	Técnico	El técnico accede a la solicitud asignada y actualiza su estado.	Estado de la solicitud actualizado y visible para el usuario.

## 5. Modelo de datos o estructura de la información

*Objetivo:* representar la información que gestionará el sistema.\*

Incluye las entidades principales (tablas u objetos) y las relaciones entre ellas.

Ejemplo:

- Entidades: Usuario, Reserva, Recurso.
- Relaciones: un usuario puede tener varias reservas, cada reserva corresponde a un recurso.

(Opcional: añadir diagrama ER o tabla resumen de campos.)

### Entidades principales

**Usuario:** Contiene la información de todos los usuarios registrados en la plataforma.

**Técnico:** Datos del personal técnico encargado de atender las solicitudes.

**Solicitud Servicio:** Registra cada incidencia o petición creada por un usuario.

**Asignación:** Relaciona una solicitud con un técnico específico.

**Historial Servicio:** Registra la evolución de cada solicitud: cambios de estado, comentarios y acciones realizadas.

**Notificación:** Mensajes enviados a los usuarios para informar de cambios en sus solicitudes.

### Relaciones entre entidades:

- Un usuario puede crear varias solicitudes de servicio.
  - Cada solicitud puede tener una asignación a un técnico, pero un técnico puede estar asignado a muchas solicitudes.
  - Una solicitud puede tener varios registros en el historial de servicio.
  - Un usuario puede recibir varias notificaciones.
  - Un técnico puede aparecer en múltiples registros del historial y asignaciones.
- 

## 6. Diseño de la interfaz

*Objetivo:* visualizar la estructura y navegación del sistema antes de desarrollarlo.\*

Incluye bocetos o capturas de las pantallas principales y una breve descripción de su función.

Para cada pantalla, indica:

- Nombre de la pantalla.
- Funcionalidad principal.
- Casos de uso relacionados.

### Pantalla 1: Página de inicio

Funcionalidad principal:

- Presenta información general de la empresa, servicios disponibles y acceso al inicio de sesión o registro.
- Permite al visitante navegar sin autenticarse.

Casos de uso relacionados:

- CU1: Registrar usuario
- CU2: Iniciar sesión

### Pantalla 2: Registro de usuario

Funcionalidad principal:

- Permite a un visitante introducir sus datos personales para crear una cuenta.

Casos de uso relacionados:

- CU1: Registrar usuario

### **Pantalla 3: Inicio de sesión**

Funcionalidad principal:

- El usuario introduce sus credenciales para entrar al sistema.

Casos de uso relacionados:

- CU2: Iniciar sesión

### **Pantalla 4: Panel de usuario**

Funcionalidad principal:

- Muestra el resumen de solicitudes creadas, su estado y accesos rápidos para crear nuevas incidencias.
- Permite editar el perfil del usuario.

Casos de uso relacionados:

- CU3: Crear solicitud de servicio

### **Pantalla 5: Crear solicitud**

Funcionalidad principal:

- El usuario describe la incidencia, adjunta imágenes o documentos y envía la solicitud.

Casos de uso relacionados:

- CU3: Crear solicitud de servicio

### **Pantalla 6: Gestión de solicitudes de Administrador**

Funcionalidad principal:

- El administrador visualiza todas las solicitudes, filtra por estado y asigna técnicos disponibles.

Casos de uso relacionados:

- CU4: Asignar solicitud a técnico

### **Pantalla 7: Panel del técnico**

Funcionalidad principal:

- El técnico consulta las solicitudes asignadas y actualiza su estado.

Casos de uso relacionados:

- CU5: Actualizar estado de servicio

## Pantalla 8: Detalle de solicitud

Funcionalidad principal:

- Muestra toda la información de una solicitud: descripción, usuario, técnico asignado, historial y estado.
- Permite agregar comentarios o actualizaciones.

Casos de uso relacionados:

- CU3: Crear solicitud de servicio
  - CU5: Actualizar estado de servicio
- 

## 7. Planificación técnica

*Objetivo:* planificar el desarrollo del proyecto.\*

Indica las tecnologías y herramientas que se utilizarán, y cómo se organizará el trabajo.

- Lenguajes y frameworks:
- Base de datos:
- Herramientas de diseño o edición:
- Reparto de tareas (si es en grupo):
- Cronograma (puede incluir un diagrama de Gantt):

Lenguajes y frameworks:

- El desarrollo de la web deberá realizarse utilizando HTML, Python y JavaScript.
- El backend deberá implementarse con Python.

Base datos:

- La base de datos será SQLite.

Herramientas de diseño o edición:

- Photoshop + Chatgpt para imágenes.
- Visual Studio Code para el código.

Reparto de tareas (si es en grupo):

- Reparto de tareas será equilibrado por ambos integrantes del grupo.
- 

## 8. Análisis de riesgos

*Objetivo:* identificar posibles problemas y cómo se afrontarán.\*

## 8.1. Identificación de riesgos

Ejemplos:

- Falta de tiempo o mala planificación.
- Problemas técnicos o incompatibilidades.
- Pérdida de datos.
- Abandono de miembros del grupo.
- Cambios en los requisitos.

1- Falta de tiempo o mala planificación: El proyecto podría retrasarse si no se cumplen los plazos establecidos o se subestiman las tareas.

2- Problemas técnicos o incompatibilidades: Fallas en la integración entre frontend y backend, errores en la base de datos o incompatibilidad entre navegadores y dispositivos.

3- Pérdida de datos: Posible pérdida de información de usuarios, solicitudes o historial si no se realizan copias de seguridad periódicas.

4- Abandono de miembros del grupo: Si algún integrante clave deja de participar, podría afectar el progreso de desarrollo o la cobertura de tareas.

5- Cambios en los requisitos: Modificaciones de funcionalidades por parte del cliente o usuario final que puedan generar retrabajo.

6- Fallas en el servidor o conectividad: Problemas de hosting, interrupción del servicio o caídas del servidor que afecten la disponibilidad de la plataforma.

7- Falta de experiencia en nuevas tecnologías: Curva de aprendizaje para manejar frameworks o herramientas utilizadas en el proyecto.

8- Problemas de seguridad: Vulnerabilidades en la web que puedan comprometer datos de usuarios o notificaciones.

## 8.2. Valoración y respuesta

Clasifica cada riesgo según su probabilidad e impacto, e indica cómo se mitigará.

Riesgo	Probabilidad	Impacto	Plan de prevención o contingencia
Falta de tiempo	Alta	Alta	Dividir tareas y fijar entregas intermedias.

Problemas técnicos	Media	Media	Probar tecnologías antes de programar.
Pérdida de datos	Baja	Alta	Hacer copias de seguridad semanales.

<b>Riesgo</b>	<b>Probabilidad</b>	<b>Impacto</b>	<b>Plan de prevención</b>
Falta de tiempo	Alta-Media	Alto	Dividir tareas, fijar entregas intermedias y revisar avances semanalmente.
Problemas técnicos	Media	Medio	Probar tecnologías antes de programar, realizar pruebas parciales y documentar problemas.
Pérdida de datos	Baja	Alto	Realizar copias de seguridad semanales y almacenar datos en servidores confiables.
Abandono de miembros del equipo	Baja	Alto	Asignar tareas de forma flexible y documentar procesos para que otros puedan continuar.
Cambios en los requisitos	Media	Alto	Mantener comunicación constante con el cliente/usuario y registrar cambios para evaluarlos antes de implementar
Falla del servidor o conectividad	Media	Alta	Contratar hosting confiable, monitorear el servidor y tener plan de contingencia para restaurar servicios.
Falta de experiencia	Media	Medio	Capacitación previa, tutoriales, y pruebas piloto antes de

			implementar funcionalidades críticas.
Problemas de seguridad	Baja	Alto	Implementar cifrado de datos, autenticación segura y pruebas de vulnerabilidad periódicas.

## 9. Validación y criterios de éxito

*Objetivo:* definir cómo sabremos que el proyecto funciona correctamente.\*

### Criterios de aceptación (qué debe cumplirse para darlo por válido).

- La plataforma permite registrar usuarios y autenticar sus cuentas correctamente.
- Los usuarios pueden crear solicitudes de servicio, adjuntar información y consultar su estado.
- Los administradores pueden gestionar solicitudes, asignar técnicos y actualizar estados.
- Los técnicos pueden ver sus solicitudes asignadas y actualizar su progreso.
- La web es accesible desde dispositivos móviles y navegadores principales.
- Las notificaciones llegan correctamente a los usuarios cuando cambia el estado de sus solicitudes.
- Todos los datos se almacenan de manera segura y con respaldo.

### Pruebas previstas (funcionales, de usuario, de rendimiento).

1. Pruebas funcionales:
  - a. Registro e inicio de sesión de usuarios.
  - b. Creación y gestión de solicitudes de servicio.
  - c. Asignación y actualización de solicitudes por parte de técnicos y administradores.
2. Pruebas de usuario:
  - a. Validación de la interfaz por usuarios reales o simulados.
  - b. Navegación intuitiva y facilidad de uso de todas las pantallas.
3. Pruebas de rendimiento:
  - a. Tiempo de carga de las páginas (<3 segundos).
  - b. Capacidad del sistema para manejar múltiples usuarios y solicitudes simultáneas.
4. Pruebas de seguridad:
  - a. Autenticación y permisos correctamente aplicados.
  - b. Protección de datos sensibles y comunicación cifrada.

### Indicadores de calidad o resultados esperados.

- Tiempos de respuesta: Solicitudes procesadas y notificadas en menos de 24 horas.

- Ausencia de errores críticos: Ningún fallo que impida la funcionalidad principal durante las pruebas finales.
- 

## 10. Conclusión

*Objetivo:* cerrar el análisis y preparar la siguiente fase.\*

Resume las decisiones principales tomadas durante el análisis:

- Qué funciones tendrá el sistema.
- Qué tecnologías se utilizarán.
- Qué valor aporta el proyecto.

Funciones principales del sistema:

- Registro y gestión de usuarios, incluyendo perfiles y autenticación.
- Creación, seguimiento y actualización de solicitudes de servicio.
- Asignación de solicitudes a técnicos y registro de historial de acciones.
- Gestión de notificaciones para mantener informados a los usuarios sobre el estado de sus servicios.
- Acceso seguro, intuitivo y compatible desde dispositivos móviles y navegadores principales.

Tecnologías seleccionadas:

- HTML, JavaScript y Python para montaje de código.
- Backend en Python.
- Base de datos SQLite
- Visual Studio Code, Photoshop y ChatGPT.

Valor que aporta el proyecto:

- Mejora la eficiencia y productividad de empresas y usuarios al garantizar un soporte técnico rápido y confiable.
- Facilita la gestión de solicitudes de mantenimiento y reparación mediante una plataforma centralizada y accesible desde cualquier dispositivo.
- Reduce riesgos de fallos críticos y pérdida de información mediante historial, notificaciones y almacenamiento seguro.
- Proporciona un sistema escalable y flexible, capaz de adaptarse a futuras necesidades de los usuarios y del negocio.

Indica los próximos pasos:

- Preparar el entorno de desarrollo.
  - Instalar y configurar los editores de código Visual Studio Code.
  - Configurar JavaScript, gestor de paquetes y dependencias necesarias.

- Preparar herramientas de control de versiones y pruebas.
  
- Crear la estructura inicial del repositorio y la base de datos.
  - Configurar repositorio con carpetas separadas para backend y documentación/imágenes necesarias.
  - Crear la base de datos con las tablas principales: Usuario, Técnico, Solicitud Servicio, Asignación, Historial Servicio y Notificación.
  - Definir relaciones, llaves primarias y foráneas según el modelo de datos.
- Empezar la implementación de los casos de uso prioritarios.
  - Registro e inicio de sesión de usuarios.
  - Creación de solicitudes de servicio y consulta de estado.
  - Gestión básica de solicitudes por administradores y técnicos.
  - Integración inicial del frontend con el backend y la base de datos.

Finaliza con una breve reflexión o frase de cierre, por ejemplo:

“El análisis funcional proporciona una visión clara y realista del proyecto. A partir de aquí, comienza la fase de desarrollo.”

El análisis funcional ha permitido definir de manera clara y estructurada los objetivos, requisitos y diseño del sistema. Con esta base, el proyecto está listo para avanzar hacia la fase inicial de desarrollo, asegurando un servicio confiable y eficiente para los usuarios.