| **DUOC UC - Escuela de informática** |
| --- |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| *Proyecto: CRM* |
|  |
| **Revisión*: [01]*** |
| **03/09/2024** |

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

**Contenido**

[*DUOC UC - Escuela de informática y telecomunicaciones 1*](#_heading=h.gjdgxs)

[**Ficha del documento 4**](#_heading=h.1fob9te)

[**1. Introducción 5**](#_heading=h.3znysh7)

[1.1.](#_heading=h.2et92p0) Propósito 5

[1.2.](#_heading=h.tyjcwt) Ámbito del Sistema 5

[1.3.](#_heading=h.3dy6vkm) Definiciones, Acrónimos y Abreviaturas 5

[1.4.](#_heading=h.1t3h5sf) Referencias 5

[1.5.](#_heading=h.4d34og8) Visión General del Documento 5

[**2.**](#_heading=h.2s8eyo1) **Descripción General 6**

[2.1.](#_heading=h.17dp8vu) Perspectiva del Producto 6

[2.2.](#_heading=h.3rdcrjn) Funciones del Producto 6

[2.3.](#_heading=h.26in1rg) Características de los Usuarios 6

[2.4.](#_heading=h.lnxbz9) Restricciones 6

[2.5.](#_heading=h.35nkun2) Suposiciones y Dependencias 7

[2.6.](#_heading=h.1ksv4uv) Requisitos Futuros 7

[**3.**](#_heading=h.44sinio) **Requisitos Específicos 8**

[3.1](#_heading=h.2jxsxqh) Requisitos comunes de las interfaces 8

[*3.1.1*](#_heading=h.z337ya) *Interfaces de usuario 8*

[*3.1.2*](#_heading=h.3j2qqm3) *Interfaces de hardware 8*

[*3.1.3*](#_heading=h.1y810tw) *Interfaces de software 8*

[3.2](#_heading=h.4i7ojhp) Requisitos funcionales 9

[3.3](#_heading=h.2xcytpi) Requisitos no funcionales (Organización) y de calidad (Producto) 9

[*3.3.1*](#_heading=h.1ci93xb) *Requisitos de Rendimiento o Eficiencia 9*

[*3.3.2*](#_heading=h.3whwml4) *Requisitos de Seguridad 9*

[*3.3.3*](#_heading=h.2bn6wsx) *Requisitos de Usabilidad 9*

[*3.3.4*](#_heading=h.qsh70q) *Requisitos de Disponibilidad 9*

[*3.3.5*](#_heading=h.3as4poj) *Requisitos de Portabilidad 9*

[*3.3.6*](#_heading=h.1pxezwc) *Requisitos de Mantenibilidad 9*

[*3.3.7*](#_heading=h.49x2ik5) *Requisitos de Funcionalidad 9*

[3.4](#_heading=h.2p2csry) Requisitos No funcionales Organizacionales 10

[**4. Propuesta de Planificación 11**](#_heading=h.147n2zr)

[4.1 Descripción general acerca de la Planificación 11](#_heading=h.3o7alnk)

[*4.1.2 Definición del Equipo de Trabajo 11*](#_heading=h.23ckvvd)

[*4.1.3 Definición de Actividades principales del Proyecto 11*](#_heading=h.ihv636)

[*4.1.4 Resumen Costos del Desarrollo del Proyecto 11*](#_heading=h.32hioqz)

[*4.1.6 Carta Gantt 11*](#_heading=h.1hmsyys)

[5. Anexos 12](#_heading=h.41mghml)

[*5.1 Acta de Proyecto 12*](#_heading=h.2grqrue)

[*5.2 Matriz Especificación de Requerimientos 12*](#_heading=h.vx1227)

[*5.3 Prototipado de Software 12*](#_heading=h.3fwokq0)

[*5.4 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo 12*](#_heading=h.1v1yuxt)

[*5.5 Planilla Carta Gantt 12*](#_heading=h.4f1mdlm)

**Ficha del documento**

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
| *03/09/2024* | *Ignacio Alfaro* | *Ignacio Alfaro* | *Creación de los principales puntos del documento* |
| *09/09/2024* | *Ignacio Alfaro* | *Ignacio Alfaro* | *Término del documento* |
|  |  |  |  |

Documento validado por las partes en fecha:

**Integrantes:**

| **Nombre Integrante del Equipo** | **Rol Definido** |
| --- | --- |
| *Ignacio Alfaro* | *Jefe de Proyecto y Programador* |
| *Manuel Fuentes* | *Programador* |
| *Benjamin Ramos* | *Scrum Master* |

**1. Introducción**

El propósito de este documento ERS es definir los requisitos y especificaciones del software a desarrollar. Este documento está dirigido al equipo de desarrollo de software, así como a los clientes y usuarios finales que utilizarán el software.

**1.1. Propósito**

El propósito de este documento es definir los requisitos y especificaciones del software a desarrollar, así como proporcionar una guía para el equipo de desarrollo de software.

Este documento está dirigido al equipo de desarrollo de software, así como a los clientes y usuarios finales (stakeholders) que utilizarán el software.

**1.2. Ámbito del Sistema**

En esta subsección:

Nombre del proyecto: **Farmacia Simple SPA**

1. Portal Web:
   1. Interfaz de usuario atractiva y de fácil navegación.
   2. Seguridad y protección de datos de los clientes.
2. Proceso de Implementación y Gobernanza:
   1. Definición de roles y responsabilidades.
   2. Planificación y cronograma detallados.
   3. Control de costos y recursos.
   4. Proceso de pruebas y validación.

**1.3. Definiciones, Acrónimos y Abreviaturas**

* Ejemplo:
* Framework: Un marco de trabajo estructurado para desarrollar aplicaciones o sistemas.

**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**

Nuestro objetivo es implementar un sistema de Gestión de Relaciones con Clientes (CRM) que centralice y optimice la interacción con los clientes, permitiendo una gestión integral y personalizada que incremente la satisfacción del cliente, mejore la eficiencia operativa y fomente la fidelización. Este CRM será una herramienta clave para soportar nuestras estrategias de crecimiento, brindando a nuestro equipo de ventas, marketing y servicio al cliente una plataforma intuitiva y unificada que les permita tomar decisiones informadas y proactivas.

#### Lenguajes de programación y servicios a utilizar por proyecto:

* Página web: Django/Python
* Base de datos: Por Definir

**2. Descripción General**

1. Portal web (Python con Visual Studio):

Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Con ello se trabajará con el Framework Django, ya que nos facilitara el trabajo de agrupar las diferentes funciones en una gran colección de módulos reutilizables, llamada marco de aplicación web.

**2.1. Perspectiva del Producto**

1. Plataforma Web (Python con Visual Studio):

El producto CRM estará diseñado para ser una plataforma integral que centralice toda la información relacionada con los clientes, desde el historial de interacciones hasta datos de ventas y soporte. Se espera que este sistema no solo registre y organice estos datos, sino que también proporcione herramientas avanzadas para el análisis y la toma de decisiones. Esto incluye capacidades de automatización para la gestión de campañas de marketing, seguimiento de oportunidades de ventas, y atención al cliente, permitiendo a los equipos trabajar de manera más eficiente y proactiva.

**2.2. Funciones del Producto**

El producto comprende de los siguientes componentes clave:

Plataforma Web: Por definir

**2.3. Características de los Usuarios**

Usuarios Internos: Empleados de la Empresa que gestionan las operaciones y la administración de la plataforma.

**2.4. Restricciones**

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

• Políticas de la empresa: El producto debe cumplir con las políticas de la empresa en términos de seguridad, accesibilidad, usabilidad y estándares de desarrollo de software.

• Limitaciones del hardware: El producto debe ser diseñado de tal manera que pueda ejecutarse en hardware de bajo rendimiento, como computadores anteriores a 2016 pero posteriores al 2006.

• Interfaces con otras aplicaciones: El producto debe ser compatible con otras aplicaciones utilizadas por la empresa, como sistemas de gestión de contenido y sistemas de pago.

• Operaciones paralelas: El producto debe ser capaz de realizar operaciones paralelas y gestionar múltiples solicitudes de forma eficiente para garantizar un rendimiento óptimo.

• Funciones de auditoría: El producto debe incluir funciones de auditoría para permitir el seguimiento de las acciones realizadas por los usuarios y garantizar la integridad de los datos.

• Funciones de control: El producto debe incluir funciones de control para garantizar que los usuarios solo puedan acceder a las funciones y datos a los que tienen permiso.

• Lenguaje(s) de programación (Web): Se utilizará el framework de Django que utiliza como lenguaje de programación Python, además de utilización de lenguaje de etiquetas HTML5.

• Protocolos de comunicación: El producto debe cumplir con los protocolos de comunicación estándar de la empresa y garantizar la seguridad y privacidad de la información transmitida.

• Requisitos de habilidad: El producto debe ser diseñado de tal manera que la mayoría de usuarios lo puedan ocupar

• Criticidad de la aplicación: El producto debe ser diseñado para manejar adecuadamente situaciones de alta criticidad, como fallos de hardware o pérdida de datos.

• Consideraciones acerca de la seguridad: El producto debe ser diseñado para garantizar la seguridad de la información almacenada y procesada, incluyendo medidas de autenticación, autorización y encriptación de datos.

**2.5. Suposiciones y Dependencias**

Dependencias de terceros: Si un requisito depende de un tercero, como un proveedor de software o un servicio en línea, un cambio en esa dependencia podría afectar el requisito. Por ejemplo, si un requisito presupone el uso de una API de un proveedor de terceros, pero ese proveedor cambia su API, el requisito podría necesitar ser revisado.

Suposiciones sobre la base de datos: Si un requisito presupone una estructura específica de base de datos o un esquema de nombres, un cambio en esa estructura podría afectar el requisito. Por ejemplo, si un requisito presupone un esquema de nombres específico para las tablas de la base de datos, pero luego se cambia el esquema de nombres, el requisito podría necesitar ser revisado.

Suposición de que la renovación tecnológica impulsará un crecimiento del 25% anual en ingresos durante los próximos 5 años.

**2.6. Requisitos Futuros**

* Implementación de portal de autoayuda para ayudar al usuario a solucionar problemas de diferentes ámbitos relacionados con la empresa y su servicio.
* Posibilidad de agregar nuevas características y servicios a medida que evoluciona el mercado.

**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 describe 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 cuerpo de la página debe ser blanco
* El NavBar es de color celeste claro
* El pie de página debe ser celeste y con imagen
* Los formularios deben tener fondo blanco
* La letra de los formularios debe ser negra

**3.1.2 Interfaces de hardware**

* Dispositivos de entrada (Web): Los usuarios podrán acceder al software mediante un teclado y un mouse para computadoras. Además, los usuarios podrán acceder, dependiendo de su rol, Las funciones y operaciones que pueden realizar dentro de este.
* Dispositivos de salida (Web): Los usuarios podrán ver la información y servicios, en su computadora.
* Dispositivos de salida (Web): Los técnicos y administradores podrán ver, modificar y eliminar acuerdos, correos u usuarios, entre otros.
* Conectividad: La aplicación web requerirá una conexión a internet estable y confiable para funcionar correctamente. Los usuarios podrán acceder a la plataforma a través de sus navegadores web en computadoras y celulares Android.
* Configuración: La configuración del hardware deberá estar optimizada para garantizar una experiencia fluida del usuario. Por ejemplo, se recomienda un tamaño de pantalla mínimo para una visualización clara y fácil de la información.
* Mecanismos de seguridad: Es importante contar con mecanismos de seguridad para proteger los datos personales de los usuarios y evitar el acceso no autorizado al software. La plataforma deberá contar con una autenticación segura y medidas de encriptación para la transmisión de datos sensibles.

**3.2 Requisitos funcionales**

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

**1. Gestión de Soporte al Cliente**

- Tickets de Soporte: Permitir que los clientes reporten problemas o soliciten ayuda mediante la creación de tickets.

- Base de Conocimientos: Crear una base de conocimientos con artículos, guías y preguntas frecuentes para que los clientes puedan resolver problemas comunes por sí mismos.

- Seguimiento de Resoluciones: Monitorear el tiempo de resolución de los tickets y la satisfacción del cliente con el soporte recibido.

**2. Informes y Análisis**

- Dashboards Personalizados: Ofrecer dashboards personalizables donde los usuarios puedan ver métricas clave como ventas, rendimiento de campañas, resolución de tickets, etc.

- Informes Detallados: Generar informes detallados sobre el rendimiento del equipo de ventas, la efectividad de las campañas, la retención de clientes, etc.

- Análisis Predictivo: Implementar algoritmos básicos de análisis predictivo para identificar tendencias y prever comportamientos futuros.

**3. Gestión de Documentos**

- Almacenamiento de Documentos: Proveer una sección para almacenar documentos relacionados con clientes, como contratos, propuestas, facturas, etc.

- Control de versiones: Asegurar que los documentos tengan control de versiones para evitar la pérdida de información importante.

**4. Integración con Otros Sistemas**

- Integración con Correos Electrónicos: Conectar el CRM con proveedores de correo como Gmail o Outlook para centralizar la comunicación.

- API Abierta: Proveer una API que permita la integración con otros sistemas de la empresa (ERP, sistemas de facturación, etc.).

- Importación/Exportación de Datos: Permitir la importación y exportación de datos en formatos comunes como CSV, Excel, etc.

**5. Seguridad y Roles de Usuario**

- Gestión de Permisos: Implementar un sistema de roles y permisos para controlar qué usuarios tienen acceso a diferentes secciones y funciones del CRM.

- Cifrado de Datos: Asegurar que los datos sensibles estén cifrados tanto en tránsito como en reposo.

- Auditoría de Actividades: Registrar todas las actividades realizadas por los usuarios dentro del sistema para fines de auditoría.

**6. Interfaz de Usuario Amigable**

- Diseño Responsivo: Asegurarse de que la interfaz sea accesible y usable en diferentes dispositivos, incluyendo móviles y tablets.

- Navegación Intuitiva: Facilitar la navegación a través de menús claros y una estructura de interfaz lógica.

- Accesibilidad: Considerar principios de accesibilidad para garantizar que el CRM pueda ser utilizado por personas con diferentes habilidades.

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

**3.3.1 Requisitos de Rendimiento o Eficiencia**

Especificación de los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente

1. El sistema debe proporcionar respuestas rápidas a las consultas (no más de 5 segundos).
2. El tiempo de respuesta para la generación de lista de usuarios y la actualización de nuevos correos en el sistema debe ser mínimo (no más de 3 segundos).

**3.3.2 Requisitos de Seguridad**

Especificación de elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales.

1. El sistema debe garantizar la confidencialidad de la información del cliente, como datos personales y transacciones financieras.
2. El sistema no debe permitir que los usuarios accedan a áreas restringidas (BD).
3. Se debe implementar autenticación y control de acceso para garantizar que solo los usuarios autorizados puedan acceder a las funcionalidades del sistema.

**3.3.3 Requisitos de Usabilidad**

Especificación de los factores de fiabilidad necesaria del sistema. Esto se expresa generalmente como el tiempo entre los incidentes permisibles, o el total de incidentes permisibles.

1. El sistema debe ser capaz de recuperarse en caso de fallo en menos de 30 minutos
2. En caso de ataque el sistema será bajado y levantado en un plazo no máximo de 3 días

**3.3.4 Requisitos de Disponibilidad**

Especificación de los factores de disponibilidad final exigidos al sistema. Normalmente expresados en % de tiempo en los que el software tiene que mostrar disponibilidad.

1. El sistema debe estar encendido como mínimo el 70% de su tiempo de uso, en caso de ataque el sistema puede permanecer apagado por un tiempo de no más de 1 días
2. En caso de un pleno funcionamiento el sistema debe estar encendido el 100% del tiempo si no hay algo que lo amenace

**3.3.5 Requisitos de Portabilidad**

1. Mantenimiento de hardware: Compatibilidad del software web con el hardware en el que se ejecuta. Es importante asegurarse de que el software sea compatible con diferentes sistemas operativos y versiones de navegadores web. En caso de que se actualice el hardware, se debe comprobar que el software web continúe funcionando correctamente.
2. Mantenimiento de software: Implica mantener el software web actualizado. Esto puede incluir parches de seguridad, actualizaciones de software, corrección de errores y mejoras de funcionalidad. También es importante asegurarse de que el software sea compatible con las últimas versiones de los lenguajes de programación y marcos utilizados para su desarrollo.
3. Mantenimiento de datos: Implica asegurar que los datos almacenados en el software web sean precisos y estén actualizados. Se debe realizar una copia de seguridad regular de los datos y establecer medidas de seguridad para proteger los datos de posibles amenazas.

**3.3.6 Requisitos de Mantenibilidad**

1. Mantenimiento preventivo: Es fundamental realizar un mantenimiento regularmente para prevenir problemas antes de que ocurran. Esto puede incluir la limpieza de hardware, la actualización de software y la optimización de la configuración de la base de datos.
2. Mantenimiento correctivo: Es importante tener un plan de acción en caso de que ocurra un problema y saber cómo corregirlo de manera rápida y eficiente.
3. Mantenimiento predictivo: El uso de técnicas de análisis de datos y estadísticas puede ayudar a predecir cuándo es probable que ocurra un problema y tomar medidas preventivas para evitarlo.
4. Mantenimiento evolutivo: Este tipo de mantenimiento tiene como objetivo adaptar el software a los cambios y nuevas necesidades del negocio. En este caso, se deben agregar nuevas funcionalidades y mejoras al software para que siga siendo útil y eficiente para los usuarios.

**3.3.7 Requisitos de Funcionalidad**

1. Mantenimiento Correctivo: Corrección de errores o fallos que surjan en el software. Se realiza con el objetivo de restaurar la funcionalidad original del sistema.
2. Mantenimiento Preventivo: Prevenir errores y fallos en el software antes de que estos ocurran. Este tipo de mantenimiento incluye actividades como la revisión y optimización de código, pruebas de carga y estrés, y la actualización de componentes y librerías.
3. Mantenimiento Adaptativo: Adaptación del software a los cambios en el entorno en el que opera. Esto puede incluir cambios en los requisitos de los usuarios, en las tecnologías utilizadas, o en la arquitectura del software.
4. Mantenimiento Perfectivo: Mejorar la funcionalidad del software sin cambiar su comportamiento original. Por ejemplo, puede incluir mejoras en la interfaz de usuario, la optimización de procesos o la adición de nuevas funcionalidades.

**3.4 Requisitos No funcionales Organizacionales**

1. El sistema debe utilizar una base de datos MySQLy hacer uso de procedimientos almacenados.

**4. Propuesta de Planificación**

**4.1 Descripción general acerca de la Planificación**

1. Días estimados de la creación del proyecto: 4 meses
2. Personas Involucradas:
   1. Ignacio Alfaro
   2. Benjamin Ramos
   3. Manuel Fuentes
3. Para el correcto desarrollo del proyecto todos los actores involucrados deben poder realizar su planificación en el tiempo dado o menos, en caso de ser menos se le dará más trabajo para acotar los tiempos del proyecto. Además de que los actores involucrados podrán ir rotando en sus roles si así es requerido para que todos se involucren en las diversas áreas del proyecto y sepan responder dudas en caso de ser necesario.

**4.1.2 Definición del Equipo de Trabajo**

| DICCIONARIO EDT | |
| --- | --- |
| ROL ACTOR | NOMBRE ACTOR |
| Jefe de Proyecto | Ignacio Alfaro |
| Analista Programador | Manuel Fuentes |
| Scrum Master | Benjamín Ramos |

**4.1.3 Definición de Actividades principales del Proyecto**

Enumeres las etapas principales del proyecto de acuerdo a carta Gantt y EDT

1. Fase de Documentación
2. Fase de Desarrollo
3. Fase de Desarrollo 2
4. Fase de Finalización y Presentación

**4.1.4 Resumen Costos del Desarrollo del Proyecto**

* Costos por FASE

| PAGO POR FASE | | | |
| --- | --- | --- | --- |
| FASE | PAGO TOTAL POR HORA | HORA | TOTAL |
| Fase de Documentación |  |  |  |
| Fase de Desarrollo |  |  |  |
| Fase de Desarrollo 2 |  |  |  |
| Fase de Finalización y Presentación |  |  |  |
|  |  | Total Proyecto |  |

* Costos por Actor o Rol

| ROL ACTOR | PAGO POR HORA | HORA | TOTAL |
| --- | --- | --- | --- |
| Jefe de Proyecto | $9231 | 976 | 9.009.456 |
| Analista Programador | $5538 | 976 | 5.405.088 |
| Scrum Master | $12154 | 976 | 11.862.304 |
| Total |  | Total a pagar | 26.276.848 |

**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

Fechas Importantes:

Ejemplo:

1. 13 de Septiembre: Primera entrega del proyecto, donde se entregan los principales documentos para su creación
2. 11 de Octubre: Segunda entrega del proyecto, donde se entrega el software completo o en un 80% de preparación
3. 13 de Diciembre: Última entrega del proyecto, donde se presentará el proyecto completo al 100% al cliente.

Fases principales del proyecto:

Ejemplo:

1. 08 de Septiembre: Término de la fase de Fase Documentación
2. 10 de Octubre: Término de la fase de Fase de Desarrollo
3. 21 de Noviembre: Término de la fase de Fase de Desarrollo 2
4. 12 de Diciembre: Término de la fase de Fase de Finalización y Presentación