|  |
| --- |
| DUOC UC - Escuela de informática y telecomunicaciones |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| *Proyecto: [Insertar Nombre de Proyecto]* |
|  |
| **Revisión*: [01]*** |
| **[Seleccionar fecha]** |

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

Contenido

[Ficha del documento 3](#_Toc473208938)

[1. Introducción 4](#_Toc473208939)

[1.1. Propósito 4](#_Toc473208940)

[1.2. Ámbito del Sistema 4](#_Toc473208941)

[1.3. Definiciones, Acrónimos y Abreviaturas 4](#_Toc473208942)

[1.4. Referencias 4](#_Toc473208943)

[1.5. Visión General del Documento 4](#_Toc473208944)

[2. Descripción General 5](#_Toc473208945)

[2.1. Perspectiva del Producto 5](#_Toc473208946)

[2.2. Funciones del Producto 5](#_Toc473208947)

[2.3. Características de los Usuarios 5](#_Toc473208948)

[2.4. Restricciones 5](#_Toc473208949)

[2.5. Suposiciones y Dependencias 6](#_Toc473208950)

[2.6. Requisitos Futuros 6](#_Toc473208951)

[3. Requisitos Específicos 7](#_Toc473208952)

[3.1 Requisitos comunes de las interfaces 8](#_Toc473208953)

[3.1.1 Interfaces de usuario 8](#_Toc473208954)

[3.1.2 Interfaces de hardware 8](#_Toc473208955)

[3.1.3 Interfaces de software 8](#_Toc473208956)

[3.1.4 Interfaces de comunicación 8](#_Toc473208957)

[3.2 Requisitos funcionales 8](#_Toc473208958)

[3.3 Requisitos no funcionales 9](#_Toc473208959)

[3.3.1 Requisitos de rendimiento 9](#_Toc473208960)

[3.3.2 Seguridad 9](#_Toc473208961)

[3.3.3 Fiabilidad 10](#_Toc473208962)

[3.3.4 Disponibilidad 10](#_Toc473208963)

[3.3.5 Mantenibilidad 10](#_Toc473208964)

[3.3.6 Portabilidad 10](#_Toc473208965)

[3.4 Otros Requisitos 10](#_Toc473208966)

[4. Propuesta de Planificación 11](#_Toc473208967)

[4.1 Descripción general acerca de la Planificación 11](#_Toc473208968)

[4.1.2 Definición del Equipo de Trabajo 11](#_Toc473208969)

[4.1.3 Definición de Actividades principales del Proyecto 11](#_Toc473208970)

[4.1.4 Diagrama EDT 11](#_Toc473208971)

[4.1.5 Carta Gantt 11](#_Toc473208972)

[4.1.6 Resumen Costos del Desarrollo del Proyecto 11](#_Toc473208973)

[4.2 Plan de Control de Cambio 12](#_Toc473208974)

[5. Anexos 12](#_Toc473208975)

[5.1 Acta de Proyecto 12](#_Toc473208976)

[5.2 Matriz Especificación de Requerimientos 12](#_Toc473208977)

[5.3 Diagrama de Casos de Uso General 12](#_Toc473208978)

[5.4 Planilla Casos de Uso 12](#_Toc473208979)

[5.5 Prototipado de Software 13](#_Toc473208980)

[5.6 Resultado Análisis de Calidad Diagramas Modelamiento 13](#_Toc473208981)

[5.7 Resultado Análisis de Calidad Prototipado No funcional del Sistema 13](#_Toc473208982)

[5.8 Planilla entregables del Proyecto 13](#_Toc473208983)

[5.9 Matriz de Control de Cambios 13](#_Toc473208984)

[5.10 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo 13](#_Toc473208985)

5.- Reflexión Individual respecto al trabajo………………………………..…………………………………………………14

# Ficha del documento

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

Documento validado por las partes en fecha:

**Integrantes:**

|  |  |
| --- | --- |
| **Nombre Integrante del Equipo** | **Rol Definido** |
| *Nicolas silva* | *Diseñador* |
| *Leinab muga* | *Diseñador* |
| *Benjamín Alvarado* | *diseñador* |
|  |  |
|  |  |

# 1. Introducción

Bueno en este proyecto tenemos una tienda de autos para promocionar a los nuevos autos que puedan tener un propósito en nuestra tienda

## 1.1. Propósito

Va a ser para los amantes de los autos deportivos y los autos simple o para otro motivo para los que quieren ahorrar bencina para los amantes del drift para todo hombre o mujer que quiera sentir la satisfacción de un motor

## 1.2. Ámbito del Sistema

Se llamara Melipilla automotriz NUESTRO PROYECTO HARA QUE SE PUEDA VER TODO TIPOS DE AUTOS PARA QUE LOS PUEDAN COTIZAR O VENDER Y NO SE PODRA HACER PRESTAMOS O VER ADELANTOS Y TENER EL PRODUCTO NUESTRA META ES CLARA TENER MAS Y MAS CLIENTES SATISFECHOS PARA PODER TENER BENEFICIOS .

## 1.3. Definiciones, Acrónimos y Abreviaturas

Vehículo de combustión interna (VCI): Automóvil que utiliza un motor de combustión interna para generar energía. Vehículo eléctrico (VE): Automóvil que utiliza un motor eléctrico o sistemas de propulsión eléctrica para funcionar. SUV: Vehículo Utilitario Deportivo (por sus siglas en inglés, Sport Utility Vehicle). SEDÁN: Tipo de automóvil con carrocería cerrada que generalmente tiene cuatro puertas y un maletero

## 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

En esta subsección se describe brevemente los contenidos y la organización del resto de la ERS.

# 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

Esta subsección debe relacionar el futuro sistema (producto software) con otros productos. Si el producto es totalmente independiente de otros productos, también debe especificarse aquí. Si la ERS define un producto que es parte de un sistema mayor, esta subsección relacionará los requisitos del sistema mayor con la funcionalidad del producto descrito en la ERS, y se identificarán las interfaces entre el producto mayor y el producto aquí descrito. Se recomienda utilizar diagramas de bloques.

## 2.2. Funciones del Producto

En esta subsección de la ERS se mostrará un resumen, a grandes rasgos, de las funciones del futuro sistema. Las funciones deberán mostrarse de forma organizada, y pueden utilizarse gráficos, siempre y cuando dichos gráficos reflejen las relaciones entre funciones y no el diseño del sistema. (Se recomienda algún tipo de Diagrama de los componentes del sistema)

## 2.3. Características de los Usuarios

Esta subsección describirá las características generales de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia técnica. Además debes definir los Tipos de Usuarios con sus perfiles.

## 2.4. Restricciones

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

• Políticas de la empresa.

• Limitaciones del hardware.

• Interfaces con otras aplicaciones.

• Operaciones paralelas.

• Funciones de auditoría.

• Funciones de control.

• Lenguaje(s) de programación.

• Protocolos de comunicación.

• Requisitos de habilidad.

• Criticidad de la aplicación.

• Consideraciones acerca de la seguridad.

## 2.5. Suposiciones y Dependencias

Esta subsección de la ERS describirá aquellos factores que, si cambian, pueden afectar a los requisitos. Por ejemplo, los requisitos pueden presuponer una cierta organización de ciertas unidades de la empresa, o pueden presuponer que el sistema correrá sobre cierto sistema operativo. Si cambian dichos detalles en la organización de la empresa, o si cambian ciertos detalles técnicos, como el sistema operativo, puede ser necesario revisar y cambiar los requisitos.

## 2.6. Requisitos Futuros

Esta subsección esbozará futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.

# 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.

• Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características:

* **Corrección**: La ERS es correcta si y sólo si todo requisito que figura aquí (y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.
* **No ambiguos**: Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en el glosario.
* **Completos**: Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos.
* **Consistentes**: Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorio no es implementable.
* **Clasificados**: Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.
* **Verificables**: La ERS es verificable si y sólo si todos sus requisitos son verificables. Un requisito es verificable (testeable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable. Expresiones como a veces, bien, adecuado, etc. Introducen ambigüedad en los requisitos. Requisitos como “en caso de accidente la nube tóxica no se extenderá más allá de 25Km" no es verificable por el alto costo que conlleva.
* **Modificables**: La ERS es modificable si y sólo si se encuentra estructurada de forma que los cambios a los requisitos pueden realizarse de forma fácil, completa y consistente. La utilización de herramientas automáticas de gestión de requisitos facilitan enormemente esta tarea.
* **Trazables**: La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del diseño y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica que componentes del sistema son los que realizan el requisito R.

## 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

Describir los requisitos del interfaz de usuario para el producto. Esto puede estar en la forma de descripciones del texto o pantallas del interfaz. Por ejemplo, posiblemente el cliente ha especificado el estilo y los colores del producto. Describa exacto cómo el producto aparecerá a su usuario previsto.

### 3.1.2 Interfaces de hardware

Especificar las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema. Se incluirán características de configuración.

### 3.1.3 Interfaces de software

Indicar si hay que integrar el producto con otros productos de software.

* Para cada producto de software debe especificarse lo siguiente:
* Descripción del producto software utilizado
* Propósito del interfaz
* Definición del interfaz: contiendo y formato

### 3.1.4 Interfaces de comunicación

Describir los requisitos del interfaz de comunicación si hay comunicaciones con otros sistemas y cuáles son los protocolos de comunicación.

## 3.2 Requisitos funcionales

Definición de acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados.

En ellas se incluye:

* Comprobación de validez de las entradas
* Secuencia exacta de operaciones
* Respuesta a situaciones anormales (desbordamientos, comunicaciones, recuperación de errores)
* Parámetros
* Generación de salidas
* Relaciones entre entradas y salidas (secuencias de entradas y salidas, formulas para la conversión de información)
* Especificación de los requisitos lógicos para la información que será almacenada en base de datos (tipo de información, requerido)

Los requisitos funcionales principales pueden ser divididos en sub-secciones.

3.2.1 Requisito funcional 1

3.2.2 Requisito funcional 2

3.2.3 Requisito funcional 3

3.2.4 Requisito funcional n

Nota: Los Requerimientos específicos se detallarán en los anexos de Planillas de Requerimientos.

## 3.3 Requisitos no funcionales

### 3.3.1 Requisitos de rendimiento

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 conectados, número de transacciones por segundo que deberá soportar el sistema, etc.

Todos estos requisitos deben ser mesurables. Por ejemplo, indicando “el 95% de las transacciones deben realizarse en menos de 1 segundo”, en lugar de “los operadores no deben esperar a que se complete la transacción”.

### 3.3.2 Seguridad

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

* Empleo de técnicas criptográficas.
* Registro de ficheros con “logs” de actividad.
* Asignación de determinadas funcionalidades a determinados módulos.
* Restricciones de comunicación entre determinados módulos.
* Comprobaciones de integridad de información crítica.

### 3.3.3 Fiabilidad

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 permisible.

### 3.3.4 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.

### 3.3.5 Mantenibilidad

Identificación del tipo de mantenimiento necesario del sistema.

Especificación de quien debe realizar las tareas de mantenimiento, por ejemplo usuarios, o un desarrollador.

Especificación de cuándo debe realizarse las tareas de mantenimiento. Por ejemplo, generación de estadísticas de acceso semanales y mensuales.

### 3.3.6 Portabilidad

Especificación de atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos. Pueden incluirse:

* Porcentaje de componentes dependientes del servidor.
* Porcentaje de código dependiente del servidor.
* Uso de un determinado lenguaje por su portabilidad.
* Uso de un determinado compilador o plataforma de desarrollo.
* Uso de un determinado sistema operativo.

## 3.4 Otros Requisitos

Cualquier otro requisito que no encaje en otra sección.

# 4. Propuesta de Planificación

## 4.1 Descripción general acerca de la Planificación

[Insertar una descripción de cómo se abordará el trabajo en cuanto a los días totales estimados y las personas involucradas en su ejecución, las buenas prácticas y condiciones necesarias a considerar para implementar para su buen término]

### 4.1.2 Definición del Equipo de Trabajo

[Describir el equipo de trabajo definido para el Proyecto e insertar Tabla de definición de Roles y funciones]

### 4.1.3 Definición de Actividades principales del Proyecto

[Descripción de las Principales fases y actividades que considera nuestra Programación de la Planificación argumentando bajo que estándares y buenas prácticas se basan (Gestión de la planificación PMI e Ingeniería de Software – es sólo enunciarlas]

### 4.1.4 Diagrama EDT

[Insertar la Estructura EDT en formato diagrama consolidada que resolviste con tu equipo]

### 4.1.5 Carta Gantt

[Insertar y Describir la Carta Gantt resultante de la programación estimada a modo de PLANIFICACIÓN donde se debe explicar la lógica aplicada para reducir el total de días lineales resultantes en la EDT y como las llevaste a la economía de calendario de la Carta Gantt que programaste con actividades paralelas y porqué.]

### 4.1.6 Resumen Costos del Desarrollo del Proyecto

[OBS.

Crear una tabla resumen extraída del EDT de cálculo de esfuerzo que desglose los principales costos asociados al proyecto: en base a la Hora hombre y roles profesionales definidos

* Costo total base esfuerzo hora hombre
* Costos por FASE
* Costos por Actor o Rol

]

## 4.2 Plan de Control de Cambio

[Se recomienda primero describir los tipos de cambio que se podrán resolver y sus alcances]

[Insertar Tabla de Control de Cambios]

[ Obs.

Insertar Descripción de los aspectos del desarrollo en los que se permitirá aplicar cambios como parte del Desarrollo del Software definiendo sus alcances y limitaciones asociadas.

El control de cambios es una actividad paralela al desarrollo del proyecto que responde a eventos que surgen del mismo, sea por requerimientos propios del usuario o por mejoras o correcciones detectadas por el mismo equipo del proyecto.

 Se describe de manera independiente de las demás fases de la metodología pues puede ser aplicada indistintamente a proyectos en marcha o proyectos ya implementados, y porque es necesario resaltar su importancia y no relegarla como una actividad posterior al desarrollo, sino reconocerla como una actividad que debe estar definida, presente y es crítica desde el inicio del proyecto.  Deberá describir que tipo aspectos Funcionalidades y no funcionales se podrán modificar con cambio, en que instancia de proyecto se podrán aplicar y que motivos los validarían para ser aplicables y en qué caso no será posible aplicar cambios.

Luego esto se debe complementar con la observación de que en el anexo encontrarán la Planilla de Control de Cambio con los Tipos de Cambio que podrán aplicarse en la cual posteriormente se debe completar la planilla al ejecutarse la instancia. ]

## 5. Anexos

### 5.1 Acta de Proyecto

Insertar Acta de Constitución del Proyecto

### 5.2 Matriz Especificación de Requerimientos

Matriz en formato planilla sobre la especificación de Requerimientos con su identificador y columnas de datos correspondiente. RF1. O RNF.1

### 5.3 Diagrama de Casos de Uso General

Insertar Diagrama de Caso de Uso General.

### 5.4 Planilla Casos de Uso

Insertar Planilla detallada de Caso de Uso para cada Actor o acción clave del proceso que lleva el sistema.

### 5.5 Prototipado de Software

Insertar Mockups y Wareframe de las interfaz de usuario del Sistema

### 5.6 Resultado Análisis de Calidad Diagramas Modelamiento

Insertar Resultado del Análisis de Calidad basado en los estándares y la Planilla de Análisis de Calidad de modelado de Software.

### 5.7 Resultado Análisis de Calidad Prototipado No funcional del Sistema

Insertar Resultado del Análisis de Calidad basado en los estándares y la Planilla de Análisis de Calidad de Prototipo de Interfaz de Usuario.

### 5.8 Planilla entregables del Proyecto

Insertar la Planilla que define los Módulos y Artefactos asociados al Caso de Uso a los que se pueden aplicar cambios en un punto de su desarrollo.

### 5.9 Matriz de Control de Cambios

Insertar la Planilla que define los Módulos y Artefactos asociados al Caso de Uso a los que se pueden aplicar cambios en un punto de su desarrollo.

### 5.10 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.]