|  |
| --- |
| ANÁLISIS Y ESPECIFICACIÓN DE SISTEMAS SOFTWARE |
| PRÁCTICA 2 |
| Especificación de requisitos |

|  |
| --- |
| JUAN RAMÓN MORALES; JOSÉ JAVIER VEGA; MANUEL JESÚS MAZÓN; ALEJANDRO MONTOYA; JAIME HERNÁNDEZ  2024/25 |

Tabla de contenido

[Descripción de los contenidos 2](#_Toc179192972)

[1. Introducción 2](#_Toc179192973)

[1.1. Objetivo 2](#_Toc179192974)

[1.2. Alcance 2](#_Toc179192975)

[1.3. Definiciones, acrónimos y abreviaturas 2](#_Toc179192976)

[1.4. Referencias 3](#_Toc179192977)

[1.5. Descripción global de la ERS 3](#_Toc179192978)

[2. Descripción General 4](#_Toc179192979)

[2.1. Perspectiva del Software 4](#_Toc179192980)

[2.2. Funciones del Software 6](#_Toc179192981)

[2.3. Características del usuario 6](#_Toc179192982)

[2.4. Restricciones generales 6](#_Toc179192983)

[2.5. Suposiciones y Dependencias 7](#_Toc179192984)

[2.6. Distribución de requerimientos 8](#_Toc179192985)

[3. Especificación de Requerimientos 9](#_Toc179192986)

[3.1. Requerimientos de Interfaces Externas **¡Error! Marcador no definido.**](#_Toc179192987)

[3.2. Requerimientos Funcionales **¡Error! Marcador no definido.**](#_Toc179192988)

[3.3. Requerimientos de Performance 11](#_Toc179192989)

[3.4. Requerimientos lógicos de la base de datos 12](#_Toc179192990)

[3.5. Restricciones de Diseño 13](#_Toc179192991)

[3.6. Atributos del Software 14](#_Toc179192992)

[4. Información de apoyo 18](#_Toc179192993)

[4.1. Tabla de contenidos e Índice 18](#_Toc179192994)

[4.2. Apéndices 18](#_Toc179192995)

Alejandro

Jaime

Juan Ramon

Manuel

José Javier

# Descripción de los contenidos

## 1. Introducción

El sistema software es capaz de ser utilizado en dispositivos móviles, tabletas y computadores. Con la función de realizar y registrar citaciones y mantener un registro de estas.

### 1.1. Objetivo

El principal objetivo de la ERS es la capacidad de la creación de citas (realizadas por los clientes) y capaz de almacenarlas y registrarlas en la base de datos de la empresa, con todos los datos necesarios sobre estas (ubicación, número de identificación de la máquina, descripción breve del problema, solución/es empleadas…).

* Los clientes, capaces de realizar citaciones indicando los datos necesarios en estas y un historial de todas las citas realizadas por este.
* Los técnicos (tanto software como hardware) son capaces de visualizar las peticiones de los clientes con sus respectivos datos y un historial de todos estos.
* El coordinador de servicio técnico puede ver todas las peticiones de clientes pendientes y también el historial de todas las citas. Pueden aceptar o rechazar peticiones pendientes y mandarlas a los respectivos técnicos.
* Un súper usuario, capaz de visualizar y realizar lo mismo que el coordinador de servicio técnico. Y a su vez, editar, añadir y eliminar cualquier citación realizada o pendiente.
* Un administrador con absoluto control de la aplicación, añadiendo la capacidad de asignar los roles de cada usuario.

### 1.2. Alcance

Tecdoc, referente a la empresa y a la función de esta. (Tecnausa y documentar)

El software podrá permitir a los usuarios realizar citas de mantenimiento y una vez realizadas, estas se quedarán guardadas en un historial de citas ordenado por fecha, ubicación o ID, a elección del lector.

* Se podrá pedir cita indicando el lugar, la fecha y la hora. (cualquier usuario logueado)
* Las citas aparecerán en un registro.
* Solo un apartado de registro. (cliente)
* Ordenado según el lugar, fecha o ID. (técnicos y administradores)
* Software responsive, al ser usado en dispositivos móviles, tabletas y computadoras.
* Interfaz simple para los clientes y técnicos.
* Consistente con las especificaciones similares de mayor nivel (por Ej.: especificaciones de requerimiento de equipamiento), si existen.

### 1.3. Definiciones, acrónimos y abreviaturas

Esta subsección explica todas las definiciones, abreviaturas, siglas y sinónimos utilizados en la especificación de requerimientos para su correcta interpretación. Esta información debe ser provista por referencia a uno o más apéndices en la ERS o por referencia a otros documentos.

### 1.4. Referencias

* Provee la lista de todos los documentos referenciados en la ERS.
* Identifica cada documento referenciado por título, fecha y editorial.
* Especifica las fuentes donde se pueden encontrar las referencias.

### 1.5. Descripción global de la ERS

* Describe los contenidos de la ERS.
* Explica la organización de la ERS.

## 2. Descripción General

Esta sección proporciona una visión general de los factores que afectan el producto de software y sus requerimientos. No expresa requerimientos específicos, pero establece la base para esos requerimientos, que se definirán en detalle en la Sección 3. El sistema que se desarrollará está diseñado para mejorar la gestión y operación de la empresa Tecnausa, cubriendo áreas clave como la gestión de problemas, el control de inventario y la interacción con el hardware de máquinas recreativas y sistemas de señalización digital.

El software debe ser eficiente, intuitivo, y adaptable a diferentes entornos de trabajo. Además, debe cumplir con las expectativas de los diferentes usuarios que interactúan con él, como técnicos, administradores y coordinadores.

### 2.1. Perspectiva del Software

El producto de software que se desarrollará es una parte de un sistema mayor que incluye múltiples componentes. Aunque puede funcionar de manera autónoma en la mayoría de sus funcionalidades, interactúa con otros sistemas de la empresa para lograr una gestión eficiente de problemas y recursos. El software se integra con plataformas de contabilidad y con dispositivos de hardware utilizados en la reparación y mantenimiento de máquinas recreativas y sistemas de visualización digital.

#### 2.1.1. Interfaces de sistema.

Las principales interfaces de sistema incluyen:

* **Sistema de contabilidad**: Para gestionar la facturación y la contabilidad de los servicios de mantenimiento.
* **Sistema de inventario**: Controla el stock de piezas y componentes necesarios para reparaciones, alertando cuando los niveles son bajos.
* **Terminales de técnicos**: Permiten a los técnicos reportar y resolver problemas en tiempo real.

Cada una de estas interfaces permite al sistema cumplir con su funcionalidad clave de gestionar el ciclo completo de notificación y resolución de problemas.

#### 2.1.2. Interfaces de usuario.

Las interfaces de usuario están diseñadas para optimizar la interacción del sistema con los diferentes tipos de usuarios, facilitando su uso mediante configuraciones intuitivas:

* **Técnicos**: Tienen acceso a una interfaz donde pueden ver los problemas asignados, reportar el progreso y registrar soluciones.
* **Administradores**: Acceden a una interfaz más completa que incluye opciones de configuración, auditoría del sistema y generación de reportes personalizados.

Los usuarios pueden configurar:

* **Formatos de pantalla** y **diseños de página**, para personalizar su experiencia. Por ejemplo, los administradores pueden ajustar el contenido de los reportes o menús de acuerdo con sus necesidades.
* **Teclas de función**: Configurables para acciones rápidas como la generación de reportes o la asignación de técnicos.

#### 2.1.3. Interfaces de hardware.

El sistema interactúa con diferentes dispositivos de hardware, incluyendo:

* **Máquinas recreativas**: A través de conexiones estandarizadas, el software recopila datos de errores y monitorea el estado de los dispositivos.
* **Dispositivos de visualización digital**: Utilizados para mostrar en tiempo real información relacionada con los problemas y las reparaciones en curso.

#### 2.1.4. Interfaces de software.

El software debe integrarse con productos de software externos e internos, como:

* **Sistemas de contabilidad**: Para gestionar la facturación de servicios y mantener registros financieros actualizados.
* **Sistemas de inventario**: Para gestionar y actualizar en tiempo real los recursos disponibles para las reparaciones.

Propósito: Automatizar la facturación y la gestión de inventarios, facilitando las operaciones de mantenimiento.

#### 2.1.5. Interfaces de comunicación.

El software debe operar utilizando protocolos de red estándar, tales como:

* **TCP/IP** para la comunicación interna entre los terminales de los técnicos y el servidor.
* **Protocolos de comunicación en red local** para garantizar una conexión segura entre el software y los dispositivos externos.

#### 2.1.6. Restricciones de memoria.

El sistema debe estar diseñado para operar eficientemente dentro de las siguientes restricciones de memoria:

* **Memoria RAM** mínima de 8 GB.
* **Almacenamiento en disco duro** de al menos 256 GB para soportar las operaciones del sistema y el almacenamiento de historiales y datos de inventario

#### 2.1.7. Operación.

El sistema soportará los siguientes modos de operación:

* **Operaciones iniciadas por el usuario**: Técnicos y administradores podrán iniciar manualmente tareas de mantenimiento, reportes y ajustes en el sistema.
* **Operaciones automáticas**: Respaldo diario automático de la base de datos de problemas y soluciones.
* **Operaciones de soporte**: El sistema facilitará la recuperación de datos en caso de fallos inesperados, asegurando que la información crítica no se pierda.

#### 2.1.8. Requerimientos de adaptación del lugar.

El software debe ser fácilmente adaptable a las especificidades de cada cliente:

* **Datos de inicialización**: Se deben configurar tablas de valores y límites de seguridad específicos para cada sala de máquinas recreativas.
* **Configuración de lugar**: El sistema debe poder ser ajustado para cumplir con las normativas locales y las características operacionales de cada cliente.

### Funciones del Software

El software debe cumplir con las siguientes funciones principales:

* **Gestión de problemas**: Registro de problemas relacionados con hardware o software, notificación al técnico adecuado y seguimiento de la resolución.
* **Organización del trabajo**: Asignación de técnicos a los problemas según su disponibilidad y el tipo de incidencia.
* **Control de inventario**: Gestión y actualización de piezas de repuesto utilizadas en las reparaciones.
* **Historial de soluciones**: Almacenar un historial completo de problemas solucionados para referencia futura y análisis de rendimiento.
* **Facturación automática**: Generar facturas automáticamente al finalizar cada reparación, sincronizándose con el sistema de contabilidad.

### 2.3. Características del usuario

Los usuarios del sistema tendrán diferentes niveles de experiencia y habilidades técnicas:

* **Administradores**: Personal con experiencia técnica avanzada, responsables de configurar y gestionar el sistema.
* **Técnicos**: Personal con experiencia en mantenimiento de hardware, responsables de ejecutar las reparaciones. No requieren altos conocimientos técnicos de software, pero deben estar familiarizados con la interfaz.
* **Clientes**: Usuarios externos que solo interactúan indirectamente con el sistema para reportar problemas y recibir notificaciones de su resolución.

### 2.4. Restricciones generales

**Regulaciones**

Dado que el sistema almacenará datos de clientes y técnicos, tendremos que asegurarnos de cumplir con la normativa de protección de datos

**Limitaciones de hardware**

El sistema tiene que ser compatible con el hardware de las máquinas recreativas y con los dispositivos que los técnicos usarán para registrar los problemas.

**Interfaces con otras aplicaciones**

Vamos a tener que integrar nuestro software con otros sistemas que ya utiliza la empresa, como el programa de contabilidad Filo y el sistema Fortune Players para el seguimiento de clientes.

**Operación en paralelo**

Muchos usuarios (técnicos, coordinadores, etc.) han de acceder al sistema simultáneamente para registrar problemas, asignar técnicos o consultar el historial. Por tanto, el sistema debe permitir la concurrencia sin afectar al rendimiento.

**Funciones de auditoría**

Es necesario que el sistema registre todas las acciones que realizan los usuarios, como quién ha resuelto qué problema o qué modificaciones se han hecho. Esto garantiza un control adecuado.

**Funciones de control**

El acceso a las diferentes partes del sistema debe estar restringido según el rol de cada usuario (técnico, administrador, cliente, etc.). No todos los usuarios deben tener acceso a toda la información o a todas las funcionalidades.

**Requerimientos de confiabilidad**

Se deberá implementar un sistema de copias de seguridad automáticas para garantizar la protección de la información almacenada en el sistema. Esto permitiría recuperar los datos en caso de fallos inesperados.

**Consideraciones de seguridad**

Dado que vamos a gestionar datos sensibles, la seguridad es muy importante. Tendremos que implementar sistemas de autenticación segura para los usuarios y encriptar los datos que se transmitan entre el cliente y el servidor.

### 2.5. Suposiciones y Dependencias

**Dependencia del sistema operativo**

Suponemos que el sistema operativo utilizado estará disponible y será compatible con nuestro software. Si hay algún cambio en esto, podríamos necesitar ajustar los requisitos del sistema.

**Conexión a internet**

Se da por hecho que los técnicos y demás usuarios tendrán acceso a internet de manera constante para poder registrar y consultar las incidencias en tiempo real. En caso de que no haya conexión, sería necesario implementar alguna solución para trabajar offline.

**Disponibilidad de dispositivos**

Suponemos que los técnicos dispondrán de dispositivos adecuados para acceder al sistema. Si este no fuera el caso, tendríamos que plantearnos el desarrollo de interfaces más ligeras o de fácil acceso desde otros dispositivos.

**Compatibilidad con sistemas externos**

La integración con otras aplicaciones como Filo o Fortune Players depende de que esos sistemas continúen funcionando como lo hacen actualmente. Si ellos cambian o dejan de estar disponibles, tendríamos que adaptar nuestro sistema para mantener la compatibilidad.

### 2.6. Distribución de requerimientos

**Gestión de inventario**

El control del inventario de piezas necesarias para las reparaciones es importante, pero podría considerarse una funcionalidad secundaria y, por lo tanto, se podría retrasar para versiones posteriores si no es imprescindible en la primera fase.

**Portal del cliente avanzado**

Aunque el portal básico, donde los clientes pueden reportar problemas, será prioritario, algunas funcionalidades más avanzadas, como el acceso a un historial detallado de las incidencias, podrían dejarse para futuras versiones.

**Integración a dispositivos móviles**

Debido a los diferentes SO de los dispositivos móviles y la dificultad de alcanzar la compatibilidad entre ellos, este requerimiento puede ser más extenso y complejo por lo que sería integrado en futuras versiones del sistema.

## 3. Especificación de Requerimientos

El sistema de gestión de problemas y soluciones de Tecnausa debe proporcionar una interfaz de usuario intuitiva y eficiente que se adapte a las necesidades de los diferentes tipos de usuarios identificados (administrador, técnico, coordinador de servicio técnico, súper usuario y cliente). La interfaz debe ser responsive, funcionando correctamente tanto en dispositivos de escritorio como en dispositivos móviles, para facilitar el acceso de los técnicos durante las visitas a los clientes

La interfaz de usuario debe incluir las siguientes características:

* Un panel de control personalizado para cada tipo de usuario, mostrando información relevante y acciones frecuentes.
* Un sistema de navegación claro y coherente en toda la aplicación.
* Formularios intuitivos para la entrada de datos, con validación en tiempo real.
* Visualización clara de los problemas reportados, su estado actual y el historial de soluciones.
* Un sistema de notificaciones para alertar a los usuarios sobre nuevos problemas, actualizaciones o tareas pendientes.
* Herramientas de búsqueda y filtrado para facilitar la localización de información específica.
* Gráficos y reportes visuales para representar datos importantes, como el rendimiento del servicio técnico o las estadísticas de problemas resueltos.

Interfaces de hardware:

El sistema debe ser compatible con una amplia gama de dispositivos hardware utilizados por Tecnausa y sus clientes. Esto incluye:

* Ordenadores de escritorio y portátiles utilizados por el personal de Tecnausa.
* Dispositivos móviles (tablets y smartphones) para los técnicos en campo.
* Máquinas recreativas y sistemas de cartelería digital de Tecnausa.
* Impresoras para la generación de informes y facturas.

El sistema debe ser capaz de comunicarse con estos dispositivos para recopilar datos, realizar diagnósticos y aplicar actualizaciones cuando sea necesario.

Interfaz de software:

* El sistema de gestión de problemas y soluciones debe integrarse con otros sistemas de software utilizados por Tecnausa, incluyendo:
* El programa de contabilidad Filo para la gestión financiera y facturación.
* El sistema de fidelización Fortune Players para el seguimiento de la actividad de los clientes.
* Herramientas de desarrollo como Visual Studio Code, Visual Basic y Sublime Text.
* Sistemas operativos Windows y Linux, utilizados en las máquinas recreativas.
* Software de control de versiones y gestión de proyectos utilizado por el equipo de desarrollo.
* Winscp para la conexión segura a servidores y gestión de archivos.

La integración con estos sistemas debe ser fluida, permitiendo el intercambio de datos en tiempo real y la sincronización de información relevante.

Interfaz de comunicaciones:

El sistema debe utilizar protocolos de comunicación seguros y eficientes para garantizar la integridad y confidencialidad de los datos. Esto incluye:

* Protocoles de red requeridos (HTTPS) para todas las comunicaciones de internet.
* VPN para el acceso remoto seguro al sistema por parte de los empleados.
* Protocolos de comunicaciones específicos para la interacción con las máquinas recreativas y sistemas de cartelería digital.
* APIs RESTful para la integración con sistemas externos y la posible expansión futura a aplicaciones móviles.

Requerimientos funcionales:

1. Registro y gestión de problemas:

El sistema debe permitir el registro detallado de problemas reportados por los clientes, así como la gestión completa del ciclo de vida de estos problemas hasta su resolución;

Las entradas serían las siguientes:

* + Datos del cliente (nombre, id, ubicación)
  + Tipo de problema (hardware, software, otro)
  + Descripción detallada del problema
  + Nivel de urgencia
  + Fecha y hora del reporte
  + Archivos adjuntos (opcional para mayor detalle del problema y hacer un estudio previo de este)

Procesos:

* + Validación de los datos de entrada
  + Asignación automática de un numero ubica de tique
  + Categorización del problema basada en problemas clava
  + Asignación inicial a un técnico o equipo basad en la categoría y urgencia
  + Notificación al cliente de la recepción del problema
  + Registro de problema en la base de datos

Salidas:

* + Confirmación de registro del problema
  + Numero de tique asignado
  + Estimación inicial del tiempo de resolución
  + Notificación al técnico o equipo asignado

#### 3.2.2. Requerimiento Funcional #2 ...

Podemos usar como base el ejemplo visto en clase.

|  |  |
| --- | --- |
| Número del requisito | **RF 2.1** |
| Nombre del requisito | **Visualizar ventana de login** |
| Justificación | Cualquier usuario no logueado no podrá acceder a ninguna herramienta |
| Precondiciones | Haber iniciado el SW |
| Datos de entrada |  |
| Descripción | Ventana de login, donde tendrá que indicar un correo/usuario y una contraseña |
| Pruebas de aceptación | * Si |
| Postcondiciones |  |

### Requerimientos de Performance:

### El sistema de gestión de problemas y soluciones de Tecnausa debe ser capaz de manejar un volumen significativo de datos y usuarios concurrentes, garantizando un rendimiento óptimo en todo momento. A continuación, se detallan los requerimientos de performance tanto estáticos como dinámicos:

### Requerimientos Estáticos:

### Numero de terminales:

El sistema debe soportar al menos 200 terminales concurres mes, incluyendo ordenadores de escritorio, portátiles y dispositivos móviles utilizados por el personal de Tecnausa y sus clientes.

### Cantidad de usuarios concurrentes:

El sistema debe ser capaz de manejar al menos 200 usuarios concurrente sin degradación del rendimiento.

### Cantidad y tipo de información a manejar: (por hacer)

### Requerimientos Dinámicos: (mejorar y terminar)

### Procesamientos de transacciones

### Tiempo de respuesta

### Estabilidad

### Disponibilidad

### Resiliencia

### Requerimientos lógicos de la base de datos

Aquí se deben especificar los requerimientos lógicos para cualquier información que se ingrese en la base de datos.

* Tipos de información usados por varias funciones
* Frecuencia de uso
* Capacidades de acceso
* Entidades de datos y sus relaciones
* Restricciones de integridad
* Requerimientos de retención de datos

Los tipos de información utilizados por diversas funciones dentro de la empresa incluyen datos sobre máquinas recreativas, inventario de componentes, registros de ventas, información de clientes y proveedores datos de mantenimiento y reparación.

En cuanto a la frecuencia de uso, los datos de ventas y la información de inventario son consultados y actualizados diariamente. Por otro lado, la información de clientes y proveedores se consulta semanalmente.

Respecto a las capacidades de acceso, se debe garantizar que los usuarios autorizados puedan acceder a la información necesaria. Esto implica diferentes niveles de acceso según el rol que ejerce el trabajador en la empresa. Por ejemplo, los operarios podrían tener acceso a la información de inventario y registro de máquinas, mientras que el departamento de ventas podría acceder a datos de clientes y ventas.

Las principales entidades que tiene la base de datos es la siguiente:

"Máquinas", "Componentes", "Ventas", "Clientes", "Proveedores" y "Mantenimiento" conectadas de la siguiente manera.

* **Relación entre "Máquinas" y "Componentes"** es de uno a muchos lo que significa que cada máquina puede estar compuesta por múltiples componentes, pero cada componente pertenece a una sola máquina.
* **Relación entre "Ventas" y "Clientes"** la relación es de muchos a uno ya que cada venta es realizada por un cliente, pero un cliente puede realizar múltiples compras.
* **Relación entre “Ventas" y "Máquinas"** es de muchos a uno donde cada venta puede incluir una o varias máquinas, aunque cada máquina solo puede ser vendida una vez en una transacción
* **Relación entre "Mantenimiento" y "Máquinas"** es de uno a muchos ya que cada máquina puede tener múltiples registros de mantenimiento, pero cada registro se asocia a una sola máquina.
* **Relación entre “Mantenimiento " y “Proveedores "** tienen una relación de muchos a uno ya que cada servicio de mantenimiento puede estar asociado a un proveedor, mientras que un proveedor puede realizar múltiples mantenimientos en diferentes máquinas
* **Relación entre "Componentes" y "Proveedores"** es de muchos a uno ya que cada componente puede ser suministrado por un único proveedor, pero un proveedor puede ofrecer múltiples componentes.

Las restricciones de integridad son importantes para asegurar una buena gestión de los datos. Se deben establecer reglas que garanticen que los datos ingresados cumplan con ciertos criterios coherentes. Por ejemplo, no se debe permitir la entrada de registros de ventas con cantidades negativas o la asignación de un componente a una máquina que no existe en el sistema de datos.

Los requerimientos de retención de datos son de carácter obligatorio para cumplir con las normativas legales que existen. Por ejemplos, los datos de ventas y clientes se deben de mantener durante un mínimo de cinco años. Mientras que la información relacionada con el mantenimiento de las máquinas debe de estar guardada hasta la vida útil que dura la máquina para facilitar el seguimiento y el rendimiento de la misma.

### 3.5. Restricciones de Diseño

#### 3.5.1. Cumplimiento de Estándares

* Requerimientos de Estándares tales como: formato de informes, convenciones de nombres, procedimientos contables, pistas de auditoría.

#### 3.5.2. Limitaciones de Hardware

* Describe requerimientos para que el software pueda operar en determinados tipos de hardware.

**Formato de Informes**:

El informe debe comenzar con el logo de la empresa en el encabezado. El título debe ser claro, alineado al centro, con un tamaño de letra de 18 pt y en negrita. Debajo del título se incluirá la fecha de generación del informe, con un tamaño de letra de 14 pt y en cursiva. Para el resto del informe, se utilizará un tamaño de letra de 15 pt y un interlineado de 1.5. Los márgenes del documento deben estar justificados. Las páginas deben estar numeradas en la parte inferior central. Todos los gráficos y tablas deben incluir títulos descriptivos. Finalmente, el informe debe generarse en formato PDF para que pueda ser enviado electrónicamente.

**Convenciones de Nombres**:

Las convenciones de nombres establecidas permiten una mejor organización y claridad en archivos, documentos, proyectos y bases de datos, facilitando su búsqueda y gestión. A continuación, se describen las convenciones detalladamente:

* **Archivos de documentos**: Para nombrar archivos de documentos, se debe seguir el formato TipoDocumento\_Fecha\_Autor\_Versión.extensión. Esto facilita la identificación rápida del tipo de archivo, quién lo creó y la versión actual del mismo.
* **Nombres de carpetas**: Las carpetas deben tener un formato que refleje el contenido y el proyecto al que pertenecen. El formato a seguir es Proyecto\_TipoDeContenido\_Año. Esto facilita que cada carpeta esté claramente identificada por su contenido.
* **Nombres de base de datos**: Para las bases de datos, las tablas y columnas deben seguir el formato Tabla\_TipoDato\_Atributo.
* **Nombres de proyectos**: Los proyectos deben seguir el formato NombreDelProyecto\_Año, permitiendo una rápida identificación del proyecto y el año en que se llevó a cabo.
* **Control de versiones**: Cada archivo debe incluir un número de versión y la fecha de modificación.
* **Comentarios en código fuente**: Al introducir comentarios en el código, estos deben ser claros describiendo de manera breve la funcionalidad o cambios realizados en esa sección del código.

**Procedimientos Contables**:

Las transacciones financieras deben clasificarse según corresponda a ingresos, gastos, activos o pasivos. Esto permite una mejor organización y análisis de la situación financiera de la empresa. Cada transacción debe registrarse con la fecha exacta en que ocurrió, y es esencial anotar la cantidad exacta, especificando si se trata de un ingreso o un gasto. Es necesario incluir una descripción detallada de la transacción para facilitar su revisión en posibles auditorías en el futuro. El método de pago utilizado debe ser documentado, ya sea efectivo, tarjeta, transferencia o Bizum. Cada transacción debe tener un ID único que funcione como clave primaria en la base de datos contable, asegurando así su correcta identificación y evitando duplicidades. Para garantizar la exactitud y transparencia de los registros, se deben realizar auditorías periódicas. Por último, todos los registros de transacciones deben contar con documentación ya sea en formato físico o digital para asegurar que cada movimiento esté debidamente documentado y pueda ser verificado cuando sea necesario.

**Limitaciones de Hardware**

El software debe ser compatible con sistemas operativos Windows 8 o versiones superiores y en Linux de la versión 18.04 en adelante. Para un rendimiento óptimo, se requiere un mínimo de 16 GB de RAM recomendándose el uso de memoria DDR4 o superior para garantizar una mayor velocidad y eficiencia. Además, el sistema debe contar con al menos 100 GB de espacio en el disco duro, aunque se recomienda el uso de unidades SSD en lugar de HDD, ya que ofrecen un rendimiento mejor. Para una mejor visión se recomienda el uso de monitores de al menos 24 pulgadas. En cuanto a periféricos, se sugiere emplear impresoras multifunción que permitan realizar tareas de impresión, escaneo, copiado y envío de fax. En términos de conectividad, el hardware debe contar con capacidad de red Ethernet y Wi-Fi para asegurar conexiones rápidas y estables. Por último, es esencial que cada equipo esté protegido con software de seguridad actualizado, incluyendo antivirus y firewall, para garantizar la protección frente a amenazas y vulnerabilidades.

### 3.6. Atributos del Software

#### 3.6.1. Confiabilidad

Especifica los factores necesarios para determinar el nivel de confiabilidad del software en el momento de entrega

#### 3.6.2. Disponibilidad

Especifica los factores necesarios para garantizar el nivel de disponibilidad del software tales como puntos de control, recuperación ante caídas, arranque, etc.

#### 3.6.3. Seguridad

Especifica los elementos de protección contra accesos accidentales o maliciosos para usar, modificar, destruir o revelar. Requerimientos específicos en esta área pueden incluir:

* Uso de técnicas criptográficas
* Mantenimiento de un historial de transacciones
* Asignación de funciones a diferentes módulos
* Restricción de comunicaciones entre algunas áreas del programa
* Verificación de la integridad de los datos para variables críticas

#### 3.6.4. Facilidad de Mantenimiento

Especifica qué tipos de métricas y qué valores de estos aseguran la facilidad de mantenimiento. Puede ser cierta modularidad, interfaces, complejidad, etc.

#### 3.6.5. Portabilidad

Especifica los atributos del software relacionados con la facilidad para cambiarlo a otro servidor o sistema operativo. Puede incluir:

* Porcentaje de componentes que dependen del servidor
* Porcentaje de código que depende del servidor
* Uso de lenguaje portable probado
* Uso de un compilador particular
* Uso de un sistema operativo determinado

**Confiabilidad**

Para asegurar la confiabilidad del software en el momento de entrega, se deben considerar los siguientes aspectos. En primer lugar, cuánto tiempo puede funcionar el software sin fallar a mayor tiempo de funcionamiento continuado mayor será su confiabilidad. En caso de fallos, se debe considerar cuánto tarda el software en recuperarse y volver a estar operativo. Cuanto menor sea los tiempos de reparación mayor es la confianza. El software debe ser capaz de responder adecuadamente cuando haya una gran demanda garantizando eficiencia en situaciones de alta carga. Otro punto sería como reacciona el software cuando una parte del programa falla, este debería tener mecanismos para que el resto del sistema siga funcionando sin largas interrupciones. Si en el futuro se necesita actualizar o corregir alguna parte del software, estas modificaciones deben poder realizarse de manera que afecten lo menos posible al sistema evitando que quede detenido por largos periodos.

**Disponibilidad**

El sistema ha sido diseñado para garantizar una disponibilidad total, operando al 100% del tiempo, 24 horas al día, los 7 días de la semana. Esto asegura que los usuarios de Tecnausa puedan acceder a todas las funciones del sistema de manera continua, independientemente de la hora o el día. Para aumentar la seguridad y confiabilidad del sistema, se han implementado puntos de control regulares que permiten restaurar el sistema a un estado anterior en caso de fallos. Esto garantiza que, en caso de problemas, el sistema pueda ser rápidamente recuperado sin pérdidas significativas de datos.

Además, para asegurar un rendimiento óptimo, incluso durante momentos de alta demanda, se realizan pruebas de carga de forma regular. Estas pruebas verifican que el sistema es capaz de manejar una gran cantidad de usuarios simultáneamente, sin afectar significativamente su rendimiento. Esto asegura que el sistema sea robusto y confiable permitiendo un acceso continuo y estable a los usuarios bajo cualquier condición.

**Seguridad**

Para garantizar la **seguridad** del sistema se han implementado unas series de medidas de control de acceso. Cuando un usuario intente conectarse al sistema, deberá ingresar su identificación (DNI) y su clave de acceso. El sistema verificará que los datos correspondan a un usuario autorizado, y si alguno de los campos es incorrecto o no coincide con un usuario válido, se mostrará un mensaje de error. Si el usuario falla cinco veces consecutivas en ingresar correctamente su información, el sistema se cerrará automáticamente por razones de seguridad.

El sistema contará con diferentes tipos de usuarios, a los cuales se les asignarán permisos específicos según su rol:

* **Responsable de Almacén**: Tendrá acceso para registrar entradas en el almacén de máquinas recreativas, componentes y accesorios.
* **Operarios**: Podrán reservar componentes y registrar nuevas máquinas recreativas.
* **Departamento de Compras**: Podrá dar de alta los pedidos de los proveedores.
* **Departamento de Ventas**: Podrá dar de alta los pedidos de clientes.

Además, el sistema incluirá funciones de auditoría que registrarán automáticamente la fecha, hora y el usuario que realice cada transacción. Esta funcionalidad asegura un seguimiento detallado de todas las actividades dentro del sistema, mejorando la seguridad y facilitando el control sobre cualquier acción realizada. Esto garantiza que todas las transacciones sean registradas fortaleciendo su seguridad.

**Facilidad de Mantenimiento**

A continuación, se describen las métricas y principios que garantizan un mantenimiento eficiente:

* **Modularidad**: El sistema de Tecnausa está diseñado de manera modular, lo que significa que las diferentes funciones como la gestión de máquinas recreativas, el control de inventario y la administración de pedidos, están divididas en módulos independientes. Esta estructura permite que cualquier actualización se realice en el módulo específico afectado, sin comprometer el funcionamiento de los demás.
* **Interfaz clara**: En el sistema de Tecnausa, las interfaces entre los módulos están bien definidas y documentadas. Esto permite que los desarrolladores realicen cambios sin afectar la comunicación entre los diferentes componentes. Por ejemplo, si se quiere añadir un módulo para gestionar la atención al cliente, este se puede desarrollar de forma independiente y luego conectarse fácilmente con el sistema de ventas.
* **Complejidad**: En el sistema de Tecnausa, se ha trabajado para mantener una baja complejidad en el código y la arquitectura del software. Esto significa que el sistema sea más fácil de entender y modificar. Al utilizar estructuras de código simples, los desarrolladores pueden identificar y solucionar problemas.

**Portabilidad**

Al utilizar herramientas y lenguajes de software libre garantiza una mayor portabilidad de las aplicaciones en diversos entornos. Al optar por Python como lenguaje prioritario, el sistema hace que sea portable y compatible con diversos sistemas operativos como Windows, macOS y Linux. Esto permite que el software pueda ser ejecutado en múltiples plataformas sin necesidad de modificaciones importantes. En cuanto al software utilizado para las bases de datos es MySQL ya que ofrece la flexibilidad para trabajar en diferentes entornos. Este tipo de base de datos es ventajosa por su compatibilidad con diversas arquitecturas.

El uso de compiladores estándar asegura que el código fuente pueda ser compilado y ejecutado sin importar el sistema operativo evitando la necesidad de hacer reconfiguraciones avanzadas. Esto significa que los desarrolladores pueden compilar el software en cualquier plataforma que soporte estos compiladores. Además, el uso de herramientas de software libre conlleva un ahorro económico significativo, ya que se elimina la necesidad de comprar licencias costosas. Al tener acceso al código fuente, la empresa puede personalizar su software para que se adapte a sus requisitos reduciendo así los costos.

Se puede mencionar que menos del 20% de los componentes del sistema dependen de la configuración del servidor lo que significa que la mayoría de las funcionalidades son independientes También cabe destacar que menos del 15% del código total está vinculado a configuraciones específicas del servidor lo que facilita la portabilidad .El sistema se ha desarrollado utilizando el compilador GCC ya que es compatible en múltiples plataformas, asegurando así que el código pueda ser compilado y ejecutado sin problemas en diferentes entornos.

## Información de apoyo

Esta información hace a la ERS más fácil de usar. Incluye:

* Tabla de contenidos
* Índice
* Apéndices

### 4.1. Tabla de contenidos e Índice

### 4.2. Apéndices

Los apéndices no siempre son considerados parte de la especificación de requerimientos actuales y no siempre son necesarios. Pueden incluir:

* Ejemplos de formatos de E/S, estudios de análisis de costos, o resultados de encuestas a usuarios.
* Información de referencia o apoyo que puede ayudar a los lectores de la ERS.
* Una descripción de los a ser resueltos por el software.
* Instrucciones especiales de empaquetamiento para el código y el medio en cuanto a seguridad, carga inicial, exportación u otro requerimiento.

Cuando se incluyen apéndices, la ERS debe explicitar si deben ser considerados como parte de los requerimientos.