

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ  
Facultad de Ingeniería de Sistemas Computacionales  
Departamento de Ingeniería de Software

Lic. en Ingeniería de Sistemas y Computación y Lic. en Ingeniería de Sistemas de Información

II semestre 2021

Ingeniería de Software (8581-0371)

Profesora: Ana Gloria Cordero de Hernández M.Sc.

Alumno: Michael Solis cédula: 8-958-1219 grupo: 1IF121

**Laboratorio 3:** Elaboración de la Especificación de Requerimientos. 25 puntos.

**Objetivos:**

Utilizar una guía para la Especificación de Requisitos del Software (ERS) o Especificación de Casos de Uso, que muestre la descripción del comportamiento o funcionalidad del sistema automatizado a desarrollar.

**Descripción:**

1. ¿Qué es un requisito o requerimiento del software y por qué es necesaria? 2 pts.

Son las descripciones de los servicios, o sea es una descripción detallada de las funciones y restricciones del sistema asociadas a su funcionamiento. Contiene requisitos funcionales que son los que expresan la naturaleza del funcionamiento del sistema (cómo interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento) y Requerimientos no funcionales que imponen restricciones al diseño o funcionamiento del sistema.

Es necesaria para saber que ruta tomará la arquitectura basándose en los requerimientos vistos y conseguir una correcta implementación del software.

**Bibliografía consultada**

colaboradores de Wikipedia. (2020, April 1). *Requisito (sistemas)*. Wikipedia, la enciclopedia libre.

[https://es.wikipedia.org/wiki/Requisito\\_\(sistemas\)](https://es.wikipedia.org/wiki/Requisito_(sistemas))

2. ¿Cómo se presenta o redactan las especificaciones del software? Coloque cinco ejemplos. 5 pts.

No hay dos documentos SRS idénticos porque todos los proyectos de software son diferentes. Sin embargo, todavía es posible destilar los componentes principales de un SRS y crear un esquema aproximado de cómo debería ser:

1. Introducción
  1. Propósito
  2. Audiencia
  3. Uso previsto
  4. Ámbito de aplicación
  5. Acrónimos y definiciones
2. Descripción general
  1. Necesidades del usuario
  2. Dependencias y Asunciones
3. Requisitos y características del sistema
  1. Requisitos funcionales

2. Requisitos de interfaz externa
3. Características del sistema
4. Requisitos no funcionales

La primera sección describe el producto que se está desarrollando, su propósito, audiencia objetivo, uso previsto y alcance.

La segunda sección proporciona más información sobre las necesidades de los usuarios y los factores que podrían impedir que se cumplan los requisitos establecidos en la SRS.

La última sección principal está dedicada a los requisitos específicos, tanto funcionales como no funcionales.

Un buen SRS debe cumplir con varias características clave. Debería ser:

- **Correcto:** Se debe reflejar la funcionalidad y las especificaciones del producto.
- **Sin ambigüedades:** Es mejor ser demasiado específico que ambiguo
- **Completo:** Nunca es una buena idea omitir cualquier característica solicitada por el cliente.
- **Consistente:** Todos los acrónimos y definiciones deben ser utilizados de manera consistente en todo el SRS.
- **Clasificación por importancia:** El tiempo es a menudo un recurso escaso durante el proceso de desarrollo, por lo que es una buena idea clasificar los requisitos según su importancia.
- **Verificable:** Debe haber un método de verificación para cada requisito.
- **Modificable:** Los cambios en los requisitos deben hacerse de manera sistemática, y debe tenerse en cuenta su impacto en otros requisitos.
- **Rastreable:** Todos los requisitos deben ser trazables desde su origen.

## Ejemplo 1

El “Sistema de Información Geográfica (GIS)” debe ser capaz de:

1. Visualizar un mapa: Interpretar un archivo en formato ShapeFile para mostrar en pantalla un mapa.
2. Desplazarse a lo largo de la imagen del mapa: Una vez que el mapa se dibuja en la pantalla deberá ser posible manipular la imagen de tal forma que el usuario se pueda mover con el mouse de un lado a otro.
3. Hacer zoom en alguna región del mapa: El sistema debe permitir al usuario aumentar la visualización de una imagen poderse desplazar dentro de ella y posteriormente restaurarla.
4. Agregar y quitar capas al mapa: Se requiere que el usuario pueda visualizar un mapa con varias capas simultáneamente. Podrá agregar y quitar una por una.
5. Ventana de ayuda: Proporcionar ayuda al usuario por medio de ventanas de ayuda que expliquen el funcionamiento del sistema.
6. Obtener información del objeto.- Al hacer click en un objeto, el sistema deberá desplegar la información contenida en el archivo .dbf, tal como nombre del establecimiento y su dirección.

## Ejemplo 2

El “Visualizador Molecular” debe ser capaz de:

1. Visualizar una molécula en 3D: Interpretar un archivo XYZ para mostrar en pantalla un modelo 3D de una molécula.

2. Rotar y trasladar la imagen de la molécula: Una vez que la molécula se dibuja en la pantalla deberá ser posible manipular la imagen de tal forma que esta se pueda rotar y mover de un lado a otro.
3. Hacer zoom en la molécula presentada en 3D Se requiere desarrollar una herramienta que permita hacer más grande la imagen en 3D y restaurarla.
4. Guardar la imagen 2D de la molécula a partir de la vista seleccionada por el usuario: se requiere de una herramienta que convierta a 2D la vista de la imagen 3D que seleccionó el usuario.
5. Visualizar una molécula en 2D: mostrar una imagen plana, que fue previamente renderizada, esto puede hacerse cuando se elige el modo (ii).
6. Ver un archivo XYZ: ver el contenido de un archivo que esta en este formato, esto puede hacerse cuando se elige el modo (iii).
7. Dibujar molécula: Hacer el dibujo de una molécula en 3D a partir de un dibujo anterior o bien comenzando desde el principio y guardarlo en un archivo.

### **Ejemplo 3**

“Cuaji-Ajedrez” podrá realizar lo siguiente:

1. Cargar y Guardar partidas.
2. Validar si el movimiento que lleva a cabo el usuario es posible y si es así, realizarlo.
3. Permitir que el usuario consulte las jugadas posibles.
4. Cancelar la partida actual.

### **Ejemplo 4**

En este documento detallamos la Especificación de Requerimientos del Sistema de Gestión de Recorrido MultiOro de la ciudad de Machala, en el cual se ha trabajado con el apoyo del Sr. Manuel Peralta, Gerente y Socio de la empresa.

Los procesos o funciones que conforman el sistema de información son los siguientes:

1. Gestión de control de horario: se encargará de la distribución en los horarios de recorrido que realizará cada bus durante su turno.
2. Gestión de registros por cada recorrido: Se encargará de registrar todos los recorridos realizados por el bus durante el día.
3. Gestión de contratación de socios: Se encargará de la contratación de nuevas unidades para la empresa.
4. Gestión de reportes: Se encargará de generar reportes diarios de los recorridos hechos por cada unidad durante el día, los cuales serán entregado a los socios (dueños de los buses), como constancia de los recorridos efectuados.

### **Ejemplo 5**

El sistema de administración de la frutería se llevará a cabo mediante una base de datos en la que se recogerá, aparte de la información sobre los productos y sus precios, información sobre los usuarios que utilizarán la aplicación, ya sean de tipo Empleado o de tipo Administrador. La interacción con los usuarios de la aplicación se realizará a través de una interfaz gráfica que podrá ser utilizada con una

pantalla táctil.

Las funciones principales de nuestra aplicación son las que a continuación comentamos:

1. Gestión de una base de datos (con posibilidad de actualización y borrado) en la que se almacena el precio de venta, el precio de compra, foto, el proveedor, el origen, fecha de compra, caducidad y número de existencias de cada producto; así como un campo que recoja información sobre los usuarios de la aplicación y su estatus dentro de la empresa (Empleado, Administrador o Dueño).
2. Calcular la suma de productos de una venta, dando el precio en diferentes sistemas monetarios (modificando a su vez la BD).

El sistema tendrá funciones tales como altas-bajas, compras, ventas e inventarios.

3. Altas-bajas: estará relacionado con el registro de nuevos productos, así como con los datos individuales de cada producto (nombre, precio, etc.). Esta función también se podrá utilizar para dar de alta-baja a nuevos usuarios (empleados) de la aplicación, proporcionándoles un login y una contraseña para acceder a la aplicación.

Compras: tendrá relación con la cantidad de productos en existencias, es decir solo se encargará de interactuar con el aumento en la cantidad de productos.

Ventas: es lo contrario de compras, es decir esta función solo reducirá las existencias de productos.

Inventarios: se relacionará con todos los datos, para hacer informes acerca del control de productos en la frutería.(existencias, faltantes, pérdidas).

#### Bibliografía consultada

(S/f). Uam.mx. Recuperado el 20 de octubre de 2021, de

[http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas\\_Analisis\\_Requerimiento.pdf](http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas_Analisis_Requerimiento.pdf)

2.2. *Funciones del Producto - fusm calidad del software*. (s/f). Google.com. Recuperado el 20 de octubre

de 2021, de <https://sites.google.com/site/fusmcalidaddelsoftware/proyecto/g-informe-de-especificacion-de-requerimientos/2-descripcion-general/2-2-funciones-del-producto>

(S/f). Ugr.es. Recuperado el 20 de octubre de 2021, de

<https://lsi2.ugr.es/~mvega/docis/aluwork/colectivo/Ejercicio%20en%20clase%20version%202>

*Especificación de requisitos de software (SRS): consejos y plantillas - Visure Solutions*. (2019, octubre 8).

Visuresolutions.com. <https://visuresolutions.com/es/software-requirement-specification-srs-tips-template/>

3. Investigue los formatos o plantillas para la ERS de RUP, el estándar IEEE Std 830 y una tercera ERS. Para cada uno, coloque definición, autores, plantilla y referencia en la web. 5 pts.

### Plantillas para la ERS de RUP

Definición: La Especificación de requisitos de software captura los requisitos de software completos para el sistema o una parte del sistema. es útil para recopilar los requisitos de software de su proyecto en un documento formal al estilo IEEE830.

Autor: Grady Booch, Ivar Jacobson y James Rumbaugh.

## Especificación de Caso de Uso: <Nombre de Caso de Uso>

### 1 Nombre del Caso de Uso

#### 1.1 Breve Descripción

*[La descripción debe expresar brevemente el rol y propósito del caso de uso. Un solo párrafo debería ser suficiente para esta descripción.]*

### 2 Flujo de Eventos

#### 2.1 Flujo Básico

*[Este caso de uso inicia cuando el actor hace algo. Un actor siempre inicia los casos de uso. El caso de uso debe describir lo que hace el actor y los que el sistema hace en respuesta. Debería ser expresado en la forma de un diálogo entre el actor y el sistema.]*

*El caso de uso describe lo que sucede dentro del sistema, pero no cómo o porqué. Si se intercambia información, sea específico en lo que es enviado y recibido. Por ejemplo, no es muy revelador el decir que el Actor introduce la información del cliente; es mejor decir que el Actor introduce el nombre y dirección del cliente. Un Glosario de términos es a menudo útil para mantener manejable la complejidad del caso de uso; talvez deseará definir cosas como información del cliente ahí, para evitar que el caso de uso se ahogue en detalles.*

*Las alternativas simples se podrán presentar dentro del texto del caso de uso. Si solo tomará unas pocas oraciones para describir lo que pasa cuando hay una alternativa, hágalo directamente dentro de la sección del flujo de eventos básico. Si los flujos alternativos son más complejos, use una sección aparte para describirlo. Por ejemplo, “Un Flujo Alternativo”, describe cómo describir alternativas más complejas.*

*Un dibujo a veces vale por mil palabras (aunque no hay sustituto para una prosa clara y limpia). Si mejora la claridad, siéntase libre de introducir ilustraciones gráficas de interfaces de usuario, flujos de procesos u otras figuras dentro del caso de uso. Si un cuadro de flujo es útil para presentar un proceso de decisión complejo, ¡no dude en usarlo! De la misma forma, para comportamiento dependiente de estados, un diagrama de transición de estados a menudo clarifica el comportamiento de un sistema mejor que páginas y páginas de texto. Use el medio correcto de presentación para su problema, pero sea cauteloso de utilizar terminología, notación o figuras que su audiencia no comprenderá. Recuerde que su propósito es el de clarificar, no el de oscurecer.]*

## 2.2 Flujos Alternos

### 2.2.1 < Primer Flujo Alterno >

*[Las alternativas más complejas deberían ser descritas en una sección aparte, la cual es referenciada en la sección del flujo básico de eventos. Piense en las secciones de flujos alternativos como comportamiento alternativo – cada flujo alternativo representa un comportamiento alternativo (muchas veces, dado por excepciones que ocurren en el flujo principal). Deben ser tan extensas como sea necesario para describir los eventos asociados con un comportamiento alternativo. Cuando termina un flujo alternativo, se retorna a los eventos del flujo principal a menos que se especifique lo contrario.]*

#### 2.2.1.1 < Primer Sub - Flujo Alterno >

*[Los flujos alternativos podrían ser divididos en subsecciones, si eso ayuda a clarificar.]*

### 2.2.2 < Segundo Flujo Alterno >

*[Puede haber, y posiblemente habrá, un número de flujos alternativos en un caso de uso. Mantenga cada alternativa aparte para mejorar la claridad. El uso de flujos alternativos mejora la legibilidad del caso de uso, así como para prevenir que los casos de uso sean descompuestos en jerarquía de casos de uso. Tenga presente que los casos de uso son sólo descripciones textuales y que su principal propósito es el de documentar el comportamiento de un sistema de una manera clara, concisa y entendible.]*

## 3 Requerimientos Especiales

*[Un requerimiento especial es típicamente un requerimiento no-funcional que es específico a un caso de uso pero que no es fácil o naturalmente especificable en el texto del flujo de eventos del caso de uso. Ejemplos de requerimientos especiales incluyen requerimientos legales, estándares de la aplicación, y atributos de calidad del sistema a ser construido, incluyendo requerimientos de usabilidad, confiabilidad, desempeño y de soporte (URPS). Adicionalmente, deben ser capturados en esta sección otros requerimientos como sistemas operativos y entornos, requerimientos de compatibilidad, y restricciones de diseño.]*

### 3.1 < Primer requerimiento especial >

## 4 Precondiciones

*[Una precondición (de un caso de uso) es un estado del sistema que debe estar presente antes de que el caso de uso sea ejecutado.]*

### 4.1 < Precondición Uno >

## 5 Poscondiciones

*[Una poscondición (de un caso de uso) es una lista de posibles estados en los que el sistema puede estar inmediatamente después de que el caso de uso ha finalizado..]*

### 5.1 < poscondición Uno >

## 6 Puntos de Extensión

*[Puntos de extensión del caso de uso.]*

### 6.1 < Nombre del punto de extensión >

*[Definición de la localización del punto de extensión en el flujo de eventos.]*

## Bibliografía consultada

Guest. (s/f). *RUP SRS template.doc*. PDFCOFFEE.COM. Recuperado el 20 de octubre de 2021, de  
<https://pdfcoffee.com/rup-srs-templatedoc-pdf-free.html>

*Artifact: Software requirements specification*. (s/f). Uhcl.edu. Recuperado el 20 de octubre de 2021, de  
[https://sceweb.uhcl.edu/helm/RationalUnifiedProcess/process/artifact/ar\\_srs.htm](https://sceweb.uhcl.edu/helm/RationalUnifiedProcess/process/artifact/ar_srs.htm)

Wikipedia contributors. (s/f). *Especificación de requisitos de software*. Wikipedia, The Free Encyclopedia. Recuperado el 20 de octubre de 2021, de  
[https://es.wikipedia.org/w/index.php?title=Especificaci%C3%B3n\\_de\\_requisitos\\_de\\_software&oldid=138867652](https://es.wikipedia.org/w/index.php?title=Especificaci%C3%B3n_de_requisitos_de_software&oldid=138867652)

**Definición:** En general las definiciones de los términos usados en estas especificaciones están conforme a las definiciones proporcionadas en IEEE Std 610.12-1990.

**Autores:** Ivar Jacobson, Tony Gorschek, John Mylopoulos, David Parnas, Michael Felderer, Gustavo Héctor Rossi.

## Plantilla

### 1 Introducción

[Inserte aquí el texto]

*La introducción de la Especificación de requisitos de software (SRS) debe proporcionar una vista general de la SRS. Debe incluir el objetivo, el alcance, las definiciones y acrónimos, las referencias, y la vista general del SRS.*

#### 1.1 Propósito

[Inserte aquí el texto]

- *Propósito del documento*
- *Audiencia a la que va dirigido*

#### 1.2 Alcance

[Inserte aquí el texto]

- *Identificación del producto(s) a desarrollar mediante un nombre*
- *Consistencia con definiciones similares de documentos de mayor nivel (ej. Descripción del sistema) que puedan existir*

#### 1.3 Personal involucrado

Nombre	[Inserte aquí el texto]
Rol	[Inserte aquí el texto]
Categoría profesional	[Inserte aquí el texto]
Responsabilidades	[Inserte aquí el texto]
Información de contacto	[Inserte aquí el texto]
Aprobación	[Inserte aquí el texto]

*Relación de personas involucradas en el desarrollo del sistema, con información de contacto. Esta información es útil para que el gestor del proyecto pueda localizar a todos los participantes y recabar la información necesaria para la obtención de requisitos, validaciones de seguimiento, etc.*

#### 1.4 Definiciones, acrónimos y abreviaturas

[Inserte aquí el texto]

*Definición de todos los términos, abreviaturas y acrónimos necesarios para interpretar apropiadamente este documento. En ella se pueden indicar referencias a uno o más apéndices, o a otros documentos.*

#### 1.5 Referencias

Referencia	Título	Ruta	Fecha	Autor
------------	--------	------	-------	-------



[Ref.]	[Título]	[Ruta]	[Fecha]	[Autor]

*Relación completa de todos los documentos relacionados en la especificación de requisitos de software, identificando de cada documento el título, referencia (si procede), fecha y organización que lo proporciona.*

## 1.6 Resumen

[Inserte aquí el texto]

- *Descripción del contenido del resto del documento*
- *Explicación de la organización del documento*

## 2 Descripción general

### 2.1 Perspectiva del producto

[Inserte aquí el texto]

*Indicar si es un producto independiente o parte de un sistema mayor. En el caso de tratarse de un producto que forma parte de un sistema mayor, un diagrama que sitúe el producto dentro del sistema e identifique sus conexiones facilita la comprensión.*

### 2.2 Funcionalidad del producto

[Inserte aquí el texto]

*Resumen de las funcionalidades principales que el producto debe realizar, sin entrar en información de detalle.*

*En ocasiones la información de esta sección puede tomarse de un documento de especificación del sistema de mayor nivel (ej. Requisitos del sistema).*

*Las funcionalidades deben estar organizadas de manera que el cliente o cualquier interlocutor pueda entenderlo perfectamente. Para ello se pueden utilizar métodos textuales o gráficos.*

### 2.3 Características de los usuarios

Tipo de usuario	[Inserte aquí el texto]
Formación	[Inserte aquí el texto]
Habilidades	[Inserte aquí el texto]
Actividades	[Inserte aquí el texto]

*Descripción de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia técnica.*

### 2.4 Restricciones

[Inserte aquí el texto]

*Descripción de aquellas limitaciones a tener en cuenta a la hora de diseñar y desarrollar el sistema, tales como el empleo de determinadas metodologías de desarrollo, lenguajes de programación, normas particulares, restricciones de hardware, de sistema operativo etc.*

### 2.5 Suposiciones y dependencias

[Inserte aquí el texto]

*Descripción de aquellos factores que, si cambian, pueden afectar a los requisitos. Por ejemplo una asunción puede ser que determinado sistema operativo está disponible para el hardware requerido. De hecho, si el sistema operativo no estuviera disponible, la SRS debería modificarse.*

## 2.6 Evolución previsible del sistema

[Inserte aquí el texto]

*Identificación de futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.*

## 3 Requisitos específicos

*Esta es la sección más extensa y más importante del documento.*

*Debe contener una lista detallada y completa de los requisitos que debe cumplir el sistema a desarrollar. El nivel de detalle de los requisitos debe ser el suficiente para que el equipo de desarrollo pueda diseñar un sistema que satisfaga los requisitos y los encargados de las pruebas puedan determinar si éstos se satisfacen.*

*Los requisitos se dispondrán en forma de listas numeradas para su identificación, seguimiento, trazabilidad y validación (ej. RF 10, RF 10.1, RF 10.2,...).*

*Para cada requisito debe completarse la siguiente tabla:*

Número de requisito	[Inserte aquí el texto]		
Nombre de requisito	[Inserte aquí el texto]		
Tipo	<input type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	[Inserte aquí el texto]		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

*y realizar a continuación la descripción del requisito*

*La distribución de los párrafos que forman este punto puede diferir del propuesto en esta plantilla, si las características del sistema aconsejan otra distribución para ofrecer mayor claridad en la exposición.*

### 3.1 Requisitos comunes de los interfaces

[Inserte aquí el texto]

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

#### 3.1.1 Interfaces de usuario

[Inserte aquí el texto]

*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

[Inserte aquí el texto]

*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

[Inserte aquí el texto]

*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**

[Inserte aquí el texto]

*Describir los requisitos del interfaces de comunicación si hay comunicaciones con otros sistemas y cuales son las protocolos de comunicación.*

### **3.2 Requisitos funcionales**

[Inserte aquí el texto]

*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)*

*Las requisitos funcionales 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**

### **3.3 Requisitos no funcionales**

#### **3.3.1 Requisitos de rendimiento**

[Inserte aquí el texto]

*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 medibles. 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**

[Inserte aquí el texto]

*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**

*[Inserte aquí el texto]*

*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**

*[Inserte aquí el texto]*

*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**

*[Inserte aquí el texto]*

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

### **3.3.6 Portabilidad**

*[Inserte aquí el texto]*

*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**

*[Inserte aquí el texto]*

*Cualquier otro requisito que no encaje en ninguna de las secciones anteriores.*

*Por ejemplo:*

*Requisitos culturales y políticos*

*Requisitos Legales*

## **4 Apéndices**

*[Inserte aquí el texto]*

*Pueden contener todo tipo de información relevante para la SRS pero que, propiamente, no forme parte de la SRS.*

## Bibliografía consultada

(S/f). Unican.es. Recuperado el 21 de octubre de 2021, de

[https://www.ctr.unican.es/asignaturas/is1/plantilla\\_formato\\_ieee830.doc](https://www.ctr.unican.es/asignaturas/is1/plantilla_formato_ieee830.doc)

de Los Requisitos del, I.-S.-830-1998 :. E. (s/f). *En general las definiciones de los términos usados en estas especificaciones están conforme a las definiciones proporcionadas en IEEE Std 610.12-1990*. Unican.es. Recuperado el 20 de octubre de 2021, de

[https://www.ctr.unican.es/asignaturas/is1/IEEE830\\_esp.pdf](https://www.ctr.unican.es/asignaturas/is1/IEEE830_esp.pdf)

### TERCERA ERS – Plantilla VOLERE

Definición: La Plantilla de Especificación de Requisitos Volere está creada para ser utilizada como una base para sus especificaciones de requisitos. La plantilla provee secciones por cada tipo de los requisitos apropiados para los actuales sistemas de software. Volere es el resultado de muchos años de práctica, consultas, e investigación en ingeniería de requisitos.

Autor: The Atlantic Systems Guild Limited.

#### **Planilla VOLERE para Documentación de Requisitos**

<i>Identificador del Requisito:</i>	<i>Tipo de Requisito:</i>	<i>Caso de Uso/Evento:</i>
<i>Descripción:</i>		
<i>Justificación del requisito:</i>		
<i>Fuente (que interesado lo propone):</i>	<i>Unidad en la que se origina:</i>	
<i>Criterios de validación:</i>		
<i>Grado de satisfacción del interesado:</i>	<i>Grado de insatisfacción del interesado:</i>	
<i>Dependencias (qué requisitos depende de este):</i>	<i>Conflictos (qué requisitos son incompatibles o inconsistentes con este):</i>	
<i>Documentos de soporte:</i>	<i>Histórico de cambios:</i>	
<i>Proyecto:</i>	<i>Analista:</i>	

#### Bibliografía consultada

11—Febrero, E. (s/f). *Plantilla de Especificación de Requisitos*. Volere.org. Recuperado el 20 de octubre de 2021, de [https://www.volere.org/wp-content/uploads/2018/12/template\\_es.pdf](https://www.volere.org/wp-content/uploads/2018/12/template_es.pdf)

4. Para el proyecto familia, seleccione un componente y proponga el diagrama de casos de uso correspondiente. Utilizando una de la guía de RUP, redacte la ERS de dos de los Casos de Uso, propuestos. 13 pts.

La propuesta de automatización para el componente de presupuesto incluye los siguientes casos de usos o requisitos funcionales, que son mostrados en el siguiente diagrama.

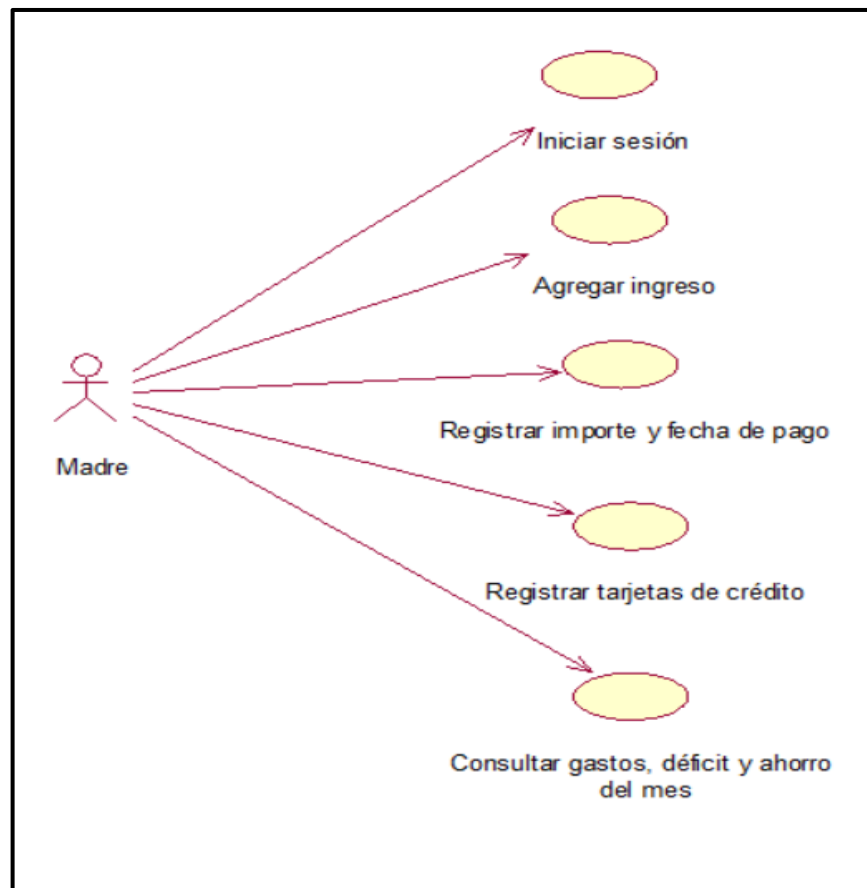


Ilustración 1. Modelo de Casos de Uso Propuesto

### Especificación de Caso de Uso: Iniciar sesión.

#### 1. Iniciar sesión.

##### 1.1 Breve Descripción

Permite el acceso de la madre como usuario al sistema del software por medio de datos proporcionados por la empresa del Software, para así poder acceder a las funcionalidades de este.

#### 2. Flujo de Eventos

##### 2.1 Flujo Básico

2.1.1. El usuario accede a la aplicación e ingresa al sistema del software.

2.1.2. Seleccionando la opción de “Iniciar sesión”.

2.1.3. Esta opción le dirige a llenar los siguientes campos de un formulario mediante el teclado móvil: usuario\*, contraseña\*.

Nota: Los campos obligatorios están marcados con asteriscos.

2.1.4. Una vez que el usuario presiona “Enviar”, estos datos son enviados al servidor para su validación de acuerdo con la base de datos de este software.

2.1.5. Si el proceso de validación es satisfactorio procede a ingresar al usuario al sistema.

## **2.2 Flujos Alternos**

2.2.1 En el punto 2.1.3

Al no llenar los campos obligatorios, se le manda un mensaje para nuevamente tener la oportunidad de llenar los mismos.

2.2.2 En el punto 2.1.5

En el proceso de validación el usuario envía datos incorrectos y son rechazados por la base de datos, se le da 2 oportunidades más de ingresar los datos correctos, de lo contrario el sistema le impedirá cualquier intento de inicio de sesión temporalmente.

## **3. Requerimientos Especiales**

3.1 Tener acceso a Internet y un dispositivo móvil para acceder a la aplicación.

## **4. Precondiciones**

4.1 El acceso a la aplicación.

## **5. Poscondiciones**

5.1 El sistema muestra todas las opciones del software a elegir como pueden ser: ingresos, gastos, tarjetas, compras.

## **Especificación de Caso de Uso: Agregar ingreso.**

### **1. Agregar ingreso.**

#### **1.1 Breve Descripción**

Dentro de la interfaz de usuario se cuenta con la opción de Ingresos, la cual permite al usuario Madre agregar ingresos.

#### **2. Flujo de Eventos**

##### **2.1 Flujo Básico**

2.1.1. Aparecerá un calendario y el usuario debe seleccionar el mes que desea trabajar.

2.1.2. El usuario selecciona uno de los 12 meses del año.

2.1.3. Para agregar un ingreso el usuario debe seleccionar el ícono “+”.

2.1.4. Esta opción le dirige a llenar los siguientes campos mediante el teclado móvil: categoría\*,



concepto\*, importe\*.

Nota 1: Los campos obligatorios están marcados con asteriscos.

Nota 2: Los campos obligatorios son de selección única de múltiples opciones que se muestran a continuación:

categoría\* muestra las siguientes opciones a elegir: Incentivos, prestaciones y sueldo.

concepto\* muestra las siguientes opciones a elegir: Honorarios/Comisiones, Salario neto.

importe\* muestra el teclado del móvil para ingresar los valores numéricos.

2.1.5. Una vez que el usuario presiona “Guardar”, estos datos son enviados al servidor para su posterior guardado en la tabla de la base de datos correspondiente.

## **2.2 Flujos Alternos**

2.2.1 En el punto 2.1.4

Al no llenar los campos obligatorios, se le manda un mensaje para nuevamente tener la oportunidad de llenar los mismos.

## **3. Requerimientos Especiales**

3.1 Haber iniciado sesión.

## **4. Precondiciones**

4.1 El acceso a la aplicación con su usuario.

## **5. Poscondiciones**

5.1 El sistema muestra en pantalla una lista de ingresos que se van registrando.

5.2 En la parte inferior de la lista, se le brinda la opción al usuario de “copiar ingresos al mes siguiente”.

**Fecha de entrega:** lunes 18 de octubre de 2021.

**El trabajo, es de realización individual.**

En e-campus:

Un archivo de texto en pdf, con el siguiente formato:

Lab#3-ProyectoFamilia-ERS\_ApellidoSiglasNombre-Grupo.pdf

Ejemplo: Lab#3-ProyectoFamilia-ERS\_CorderoAG-1XX999.pdf