República Bolivariana de Venezuela.

Ministerio del Poder Popular para la Educación.

Instituto Universitario Mario Briceño Iragorry “IUTEMBI”.

Área: Análisis y diseño de Sistemas II.

**Sistema de Información.**

Alumna:

Karen Natacha Angel Valero.

C.I: 27.557.272.

Valera, 20 de marzo de 2023.

**INTRODUCCIÓN.**

Un sistema de información está conformado por una serie de datos vinculados entre sí, con la finalidad de gestionar y administrar los datos e información que lo componen.

Por tal motivo, a lo largo del tiempo estos sistemas han evolucionado y mejorado con la implementación de ciclos específicos de actividades, los cuales se han convertido en un componente esencial para la gestión, supervivencia y crecimiento de la organización empresarial.

El presente informe abordará el ciclo de vida de un sistema de información, el cual comprende las fases por las que debe pasar un sistema, en cuanto al análisis y el diseño, para que de esta manera se mantenga la estructura y el funcionamiento de un sistema desarrollado.

**SISTEMA DE INFORMACIÓN.**

Un sistema de información es un sistema, automatizado o manual, que engloba a personas, máquinas y/o métodos organizados para recopilar, procesar, transmitir datos que representan información. Un sistema de información engloba la infraestructura, la organización, el personal y todos los componentes necesarios para la recopilación, procesamiento, almacenamiento, transmisión, visualización, diseminación y organización de la información.

Todo sistema de información cuenta con ciertos procesos internos indispensables. Estos son:

* Recopilación de datos. Proceso en el que se reúnen todos los datos disponibles y necesarios. Esto puede hacerse de forma manual o a través de tecnologías como scanner o códigos de barra.
* Almacenamiento de datos. Proceso en el que se clasifican los datos y se los almacena, esto puede darse en soportes físicos o digitales.
* Procesamiento de datos. Proceso en el que se relacionan y transforman los datos en información disponible para su uso.
* Distribución de la información. Proceso en el que se hace uso de la información para un fin determinado, como puede ser la toma de decisiones o el planteo de objetivos.

**Sistemas de información para página web.**

Los sistemas de información han ido evolucionando a lo largo de los años empleando diferentes programas y entornos hasta encontrar, en la Web, el hábitat ideal. La Word Wide Web emplea un conjunto de recomendaciones, técnicas y tecnologías muy robustas que ofrecen un gran rendimiento y fiabilidad, así como un mecanismo de navegación muy familiar para la mayoría de usuarios. Esto favorece los procesos de enseñanza de los programas basados en la Web, ya que se parte de conceptos y herramientas ampliamente conocidos, como puede suceder con los navegadores o los métodos de búsqueda y navegación. De hecho, a partir de esas ideas se pueden definir las cuatro características fundamentales de este tipo de sistemas:

* Arquitectura. Por el simple hecho de valerse de la tecnología que mueve Internet para implementar sus servicios, estos sistemas de información cuentan con una arquitectura común, al contrario de lo que sucede con otras soluciones creadas específicamente para trabajar en sistemas aislados. Esto favorece que se puedan retroalimentar unas aplicaciones de otras, sin que necesariamente pertenezcan a la misma empresa o, incluso, sin que hayan sido diseñadas para realizar las mismas funciones. Además se facilita la migración de datos de una aplicación a otra, así como la interconexión de la información procedente de diferentes fuentes, pero que empleen la misma arquitectura.
* Evolución y desarrollo. Al estar dentro de un entorno tan dinámico y cambiante, la mayoría de estos sistemas tienen la necesidad de estar en constante evolución, presentando continuas versiones que actualizan las funciones y posibilidades de las anteriores.
* El usuario por encima del sistema. El usuario del entorno Web se ha acostumbrado a recibir la información de una manera concreta y a navegar empleando una serie de convenciones basadas en su experiencia. Es lógico pensar que un sistema de información que emplee este entorno se nutra también de esas experiencias para trasladarlas a su propuesta de aplicación Web. Primero porque el usuario se las exigirá, y segundo porque no hacerlo sería desaprovechar años de experiencia en el desarrollo de aplicaciones.
* Escalabilidad. Entendida como la habilidad que tiene un sistema para reaccionar a las necesidades cambiantes (tanto del usuario como del responsable del servicio y la información que este proporciona) y adaptarse, creciendo de manera continuada y fluida. La capacidad que tienen los sistemas de información en la Web es que pueden crecer tanto como quiera, puesto que el contexto en el que se mueven, la propia Web, se basa en ese principio.

El hecho de que el trabajo se pueda realizar en un ambiente que es independiente del sistema operativo, tanto del lado del servidor como del cliente, favorece aún más si caben las ventajas de este entorno de trabajo.

Cabe destacar que las aplicaciones web son muy usadas hoy en día, debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo.

Usar aplicaciones web ahorra dinero. Empleará mejor su tiempo por no tener que ocuparse de aprender a manejar nuevos programas, ni mantenerlos o hacer copias de seguridad de sus datos y podrá trabajar desde cualquier sitio. Será más eficiente; podrá ganar más y gastará menos.

Ventajas de las aplicaciones web

* Ahorran costes de hardware y software.
* Fáciles de usar.
* Facilitan el trabajo colaborativo y a distancia.
* Escalables y de rápida actualización.
* Provocan menos errores y problemas.
* Los datos son más seguros.

**Caso de uso.**

Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Normalmente, en los casos de usos se evita el empleo de jergas técnicas, prefiriendo en su lugar un lenguaje más cercano al usuario final. En ocasiones, se utiliza a usuarios sin experiencia junto a los analistas para el desarrollo de casos de uso.

En otras palabras, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

Características.

Los casos de uso evitan típicamente la jerga técnica, prefiriendo la lengua del usuario final o del experto del campo del saber al que se va a aplicar. Los casos del uso son a menudo elaborados en colaboración por los analistas de requerimientos y los clientes.

Cada caso de uso se centra en describir cómo alcanzar una única meta o tarea de negocio. Desde una perspectiva tradicional de la ingeniería de software, un caso de uso describe una característica del sistema. Para la mayoría de proyectos de software, esto significa que quizás a veces es necesario especificar diez o centenares de casos de uso para definir completamente el nuevo sistema. El grado de la formalidad de un proyecto particular del software y de la etapa del proyecto influenciará el nivel del detalle requerido en cada caso de uso.

Los casos de uso pretenden ser herramientas simples para describir el comportamiento del software o de los sistemas. Un caso de uso contiene una descripción textual de todas las maneras que los actores previstos podrían trabajar con el software o el sistema. Los casos de uso no describen ninguna funcionalidad interna (oculta al exterior) del sistema, ni explican cómo se implementará. Simplemente muestran los pasos que el actor sigue para realizar una tarea.

Un caso de uso debe:

* Describir una tarea del negocio que sirva a una meta de negocio
* Tener un nivel apropiado del detalle
* Ser bastante sencillo como que un desarrollador lo elabore en un único lanzamiento

Situaciones que pueden darse:

* Un actor se comunica con un caso de uso (si se trata de un actor primario la comunicación la iniciará el actor, en cambio sí es secundario, el sistema será el que inicie la comunicación).
* Un caso de uso extiende otro caso de uso.
* Un caso de uso utiliza otro caso de uso.

Ventajas.

La técnica de caso de uso tiene éxito en sistemas interactivos, ya que expresa la intención que tiene el actor (su usuario) al hacer uso del sistema.

Como técnica de extracción de requerimiento permite que el analista se centre en las necesidades del usuario, qué espera éste lograr al utilizar el sistema, evitando que la gente especializada en informática dirija la funcionalidad del nuevo sistema basándose solamente en criterios tecnológicos.

A su vez, durante la extracción, el analista se concentra en las tareas centrales del usuario describiendo por lo tanto los casos de uso que mayor valor aportan al negocio. Esto facilita luego la priorización del requerimiento.

Limitaciones.

Los casos de uso pueden ser útiles para establecer requisitos de comportamiento, pero no establecen completamente los requisitos funcionales ni permiten determinar los requisitos no funcionales. Los casos de uso deben complementarse con información adicional como reglas de negocio, requisitos no funcionales, diccionario de datos que complementen los requerimientos del sistema. Sin embargo la ingeniería del funcionamiento especifica que cada caso crítico del uso debe tener un requisito no funcional centrado en el funcionamiento asociado.

Buenas prácticas y recomendaciones de uso.

Los casos de uso, como el resto de los requisitos, deben tener una redacción cuidada para evitar problemas de interpretación. En general, algunas recomendaciones a tener en cuenta son:

El caso de uso debe describir qué debe hacer el sistema a desarrollar en su interacción con los actores y no cómo debe hacerlo. Es decir, debe describir sólo comportamiento observable externamente, sin entrar en la funcionalidad interna del sistema.

El nombre del caso de uso debe ilustrar el objetivo que pretende alcanzar el actor al realizarlo.

El caso de uso debe describir interacciones con los actores sin hacer referencias explícitas a elementos concretos de la interfaz de usuario del sistema a desarrollar.

La invocación de unos casos de uso desde otros casos de uso (lo que se conoce como inclusión, o extensión si es condicional, en UML), sólo debe usarse como un mecanismo para evitar repetir una determinada secuencia de pasos que se repite en varios casos de uso. Nunca debe usarse para expresar posibles menús de la interfaz de usuario.

Se debe ser cuidadoso al usar estructuras condicionales en la descripción del caso de uso, ya que los clientes y usuarios no suelen estar familiarizados con este tipo de estructuras, especialmente si son complejas.

Se debe intentar que todos los casos de uso de una misma ERS estén descritos al mismo nivel de detalle.

En los diagramas de casos de uso, debe evitarse que se crucen las líneas que unen los actores a los casos de uso.

**Explicar un ciclo de vida para implementar una página web y una aplicación Mobile.**

Se habla de ciclo de vida porque una web tiene como todo objeto o producto un final de uso, en el caso de una web al ser un producto muy abstracto y variable necesitamos pensar lo que nos marca su ciclo de vida y una vez que no cumple sus objetivos, termina la campaña, se renueva, finaliza su vida útil.

El ciclo de vida de una página web es cuando se inician una serie de procesamientos requeridos por el usuario al servidor, el usuario al «pedir» la web al navegador comienza una serie de peticiones de código a medida que lo va leyendo y según los eventos que realice el usuario hará más o menos peticiones al servidor remoto devolviendo la web solicitada. Normalmente tecleamos el dominio de la web en el navegador y el servidor busca la información que hemos solicitado, esta información se divide en partes muy pequeñas de código dependiendo del sistema operativo desde el que ejecutemos y traduce, coloca los elementos y realiza más peticiones de elementos anidados en el código, este proceso se realiza en segundos y nos devuelve una página web que hemos solicitado.

Existen distintas fases de desarrollo implicadas en el ciclo de vida del desarrollo del sistema. No obstante, previo a estas fases, existe un proceso denominado levantamiento de requisitos, que pertenece a la ingeniería de software. Es la encargada de aplicar técnicas usadas en otros ámbitos de la ingeniería para lograr desarrollar productos de software que cumplan con los requisitos del cliente, estén dentro del presupuesto y tengan la duración estimada sin importar su complejidad.

Existen dos tipos de requerimientos para el ciclo de vida de un sistema.

* Funcionales: como su nombre indica, corresponden a los requisitos necesarios para hacer que el sistema funcione, es decir, para el ciclo de vida de un sistema. Por ejemplo, que el sistema permita registrar usuarios, que permita loguearse, que permita acceder a un listado de clientes…
* No funcionales: son los que no están directamente relacionados con la funcionalidad del sistema, sino que van a ser importantes para que el sistema se desenvuelva bien en su entorno. Por ejemplo, el sistema deberá utilizar navegador Mozilla Firefox versión n o más.

En el ciclo de vida de un sistema de información de una página web, pueden distinguirse siete fases. Se trata de las siguientes:

* Fase de planificación. En esta fase se prepara el diseño y posterior implementación del sistema. Es necesario definir el alcance del proyecto, justificarlo y escoger una metodología para su desarrollo. También es preciso asociar las diferentes actividades a plazos de tiempo y designar roles y responsabilidades.
* Fase de análisis. Una vez que el equipo de proyecto se decide por una metodología de desarrollo determinada, da comienzo la segunda etapa en el ciclo de vida de un sistema de información. Es la que tiene que ver con el análisis y donde se busca concretar una serie de requisitos, que son los que regirán el nuevo sistema o los cambios a introducir en el antiguo, si con el proyecto se busca su actualización.
* Fase de diseño. En este estadio el equipo de proyecto tendrá que determinar cómo el nuevo sistema de información cumplirá con los requisitos aplicables. Es por ello que, a estas alturas del ciclo de vida de un sistema de información conviene identificar soluciones potenciales, evaluarlas y elegir la más conveniente. Ésta será o la más efectiva, o la más eficiente en costes o la menos compleja. Una vez completadas esas tareas, habrá que continuar haciendo la selección tecnológica de software y hardware, desarrollando las especificaciones para las distintas aplicaciones y obteniendo aprobación de la gerencia para poder proceder a la implementación del nuevo sistema.
* Fase de desarrollo. El desarrollo software marca un antes y un después en la vida del sistema y significa, además, el inicio de la producción. El cambio es una constante durante esta etapa, en la que suele ser recomendable poner el foco en la formación y capacitación de los usuarios y el equipo técnico.
* Fase de integración y periodo de pruebas. El objetivo de esta etapa es corroborar que el diseño propuesto cumple con los requisitos de negocio establecidos. Puede ser necesario repetir las pruebas tantas veces como haga falta para evitar errores y, de hecho, conviene que el usuario final dé su conformidad con el resultado. Por último, este estadio concluye con la verificación y validación, que ayudan a asegurar la compleción del programa con éxito.
* Fase de implementación. En esta etapa del ciclo de vida de un sistema de información hay que proceder a la instalación del hardware y software elegidos, crear las aplicaciones correspondientes, someterlas a pruebas, crear la documentación pertinente y capacitar a los usuarios. La conversión de datos es importante en este estadio, en el que ya se empieza a trabajar en el nuevo sistema.
* Fase de mantenimiento. Esta etapa del ciclo de vida de un sistema de información está relacionada con las operaciones del día a día. Por lo general, consiste en introducir los ajustes necesarios para mejorar el rendimiento y corregir los problemas que puedan surgir. Una vez concluye la etapa de implementación se suele abrir un periodo de operación supervisada, durante el que las actividades de mantenimiento cobran una especial importancia, al orientarse a hacer backups, dar soporte a los usuarios, resolver fallos, optimizar el sistema para cuestiones relacionadas con la seguridad o la velocidad y revisar el software para garantizar la alineación con las metas del negocio.

Ciclo de vida de desarrollo de software móvil.

El ciclo de vida de desarrollo móvil es, en gran medida, parecido al SDLC para aplicaciones web o de escritorio. Al igual que con estos, normalmente hay 5 partes importantes del proceso:

* Inicio: todas las aplicaciones se inician con una idea. Normalmente, la idea se perfecciona en una base sólida para una aplicación.
* Diseño: la fase de diseño consiste en definir la experiencia del usuario (UX) de la aplicación (como cuál es el diseño general, cómo funciona, etc.), así como convertir esa experiencia del usuario en un diseño de interfaz de usuario (UI) adecuado, normalmente con la ayuda de un diseñador gráfico.
* Desarrollo: normalmente, es la fase con un uso más intensivo de recursos, esta es la creación real de la aplicación.
* Estabilización: cuando el desarrollo ha avanzado lo suficiente, normalmente el control de calidad empieza a probar la aplicación y se corrigen los errores. A veces, una aplicación pasará a una fase beta limitada en la que una audiencia de usuarios más amplia tiene la oportunidad de usarla, enviar comentarios y notificar cambios.
* Implementación

A menudo, muchas de estas partes se superponen, por ejemplo, es común que el desarrollo siga mientras se finaliza la interfaz de usuario e incluso puede afectar al diseño de la interfaz de usuario. Además, una aplicación puede estar en una fase de estabilización al mismo tiempo que se agregan nuevas características a una nueva versión.

Asimismo, estas fases se pueden usar en varias metodologías de SDLC como Agile, Spiral, Waterfall, etc.

Fragmentación del sistema operativo y del dispositivo

Es importante tener en cuenta distintos dispositivos en todo el ciclo de vida de desarrollo de software:

* Planeación y conceptualización: tenga en cuenta que el hardware y las características varían de un dispositivo a otro, puede que una aplicación que se basa en determinadas características no funcione correctamente en algunos dispositivos. Por ejemplo, no todos los dispositivos tienen cámaras, por lo que, si está creando una aplicación de mensajería de vídeo, puede que algunos dispositivos reproduzcan vídeos, pero no los puedan grabar.
* Diseño: al diseñar la experiencia del usuario (UX) de una aplicación, preste atención a las diferentes relaciones y tamaños de pantalla de los dispositivos. Además, al diseñar la interfaz de usuario (UI) de una aplicación, se deben tener en cuenta diferentes resoluciones de pantalla.
* Desarrollo: al usar una característica del código, siempre se debe probar primero la presencia de esa característica. Por ejemplo, antes de usar una característica de dispositivo, como una cámara, confirme siempre primero que el sistema operativo tenga esa característica. Después, al inicializar el dispositivo o característica, asegúrese de solicitar la compatibilidad actual del sistema operativo sobre el dispositivo y después use esas opciones de configuración.
* Pruebas: es muy importante probar la aplicación al principio y con frecuencia en dispositivos reales. Puede haber incluso dispositivos con las mismas especificaciones de hardware en que varíe mucho su comportamiento.

Consideraciones de Windows.

* Multitarea.

La multitarea en UWP también tiene dos partes: el ciclo de vida de páginas y aplicaciones, y los procesos en segundo plano. Cada pantalla de una aplicación es una instancia de una clase de página, que tiene eventos asociados que se ponen en estado activo o inactivo (con reglas especiales para controlar el estado inactivo o "con marcador de exclusión").

La segunda parte consiste en proporcionar agentes en segundo plano para tareas de procesamiento, incluso cuando la aplicación no se está ejecutando en primer plano.

* Capacidades del dispositivo.

Aunque el hardware de UWP es bastante homogéneo, hay componentes que son opcionales y, por tanto, requieren consideraciones especiales durante la codificación. Entre las capacidades de hardware opcionales se incluyen la cámara, la brújula y el giroscopio. También hay una clase especial de memoria insuficiente (256 MB) que requiere una consideración especial, o los desarrolladores pueden no participar en la compatibilidad con memoria insuficiente.

* Consideraciones de seguridad.

Para obtener más información sobre las consideraciones de seguridad importantes en UWP, consulte la documentación relativa a la seguridad.

**Definiciones y diagramas.**

1. Diagrama del Sistema Informático y Comunicación por Redes: La seguridad es establecida en principio sobre la base de la identificación de cada uno de los usuarios y sus correspondientes permisos de lectura, escritura y ejecución. Dado que varios usuarios pueden trabajar en un mismo proyecto, todos ellos están capacitados para leer la información de los otros más no modificarla. Desde el punto de vista práctico, el uso de un servidor central de archivos permitió controlar la información y el progreso de los trabajos, además que facilitó el acceso de todos los usuarios a los mismos programas de digitalización y edición. Otro de los elementos de seguridad presentes en el sistema, es el sistema automatizado de back-up, programa desarrollado en el lenguaje de script csh, que tiene como base distintos comandos del sistema operativo UNIX (Linux Red Hat). El back-up se realiza en forma centralizada sobre el servidor de archivos, y en forma automática en horarios nocturnos. La seguridad también se entendió como seguridad de funcionamiento o estabilidad de equipos. Con esta idea se trata de contar con la máxima disponibilidad de todos los componentes informáticos y garantizar la producción cartográfica en todo tiempo. Es así que se adoptaron sistemas operativos robustos (Windows NT 4, Sun Solaris y Linux), se definió una política informática que entendió a las PCs como plataformas de trabajo cartográfico, es decir que los equipos solamente están preparados para digitalizar y editar la información geográfica.



Diagrama del Sistema Informático y Comunicación por Redes.

1. Diagrama de casos de uso:

El diagrama de casos de uso muestra de forma gráfica las características del software (sus funcionalidades) incluyendo:

* Los casos de uso.
* Los roles que los usuarios desempeñan en estos casos de uso. A estos roles se les denomina “Actores”.
* La interrelación entre los elementos.

Un caso de uso bien elaborado debe:

* Demostrar las funcionalidades que satisfacen las necesidades del usuario.
* Apoyar el proceso de verificación y validación: El diagrama de casos de uso debe poderse utilizar para revisar el software y validar que todas las funcionalidades requeridas por los usuarios están incluidas en este.
* Como principio, debe ser de fácil lectura e interpretación para todos. No sólo para los desarrolladores de software sino también para clientes y usuarios finales.

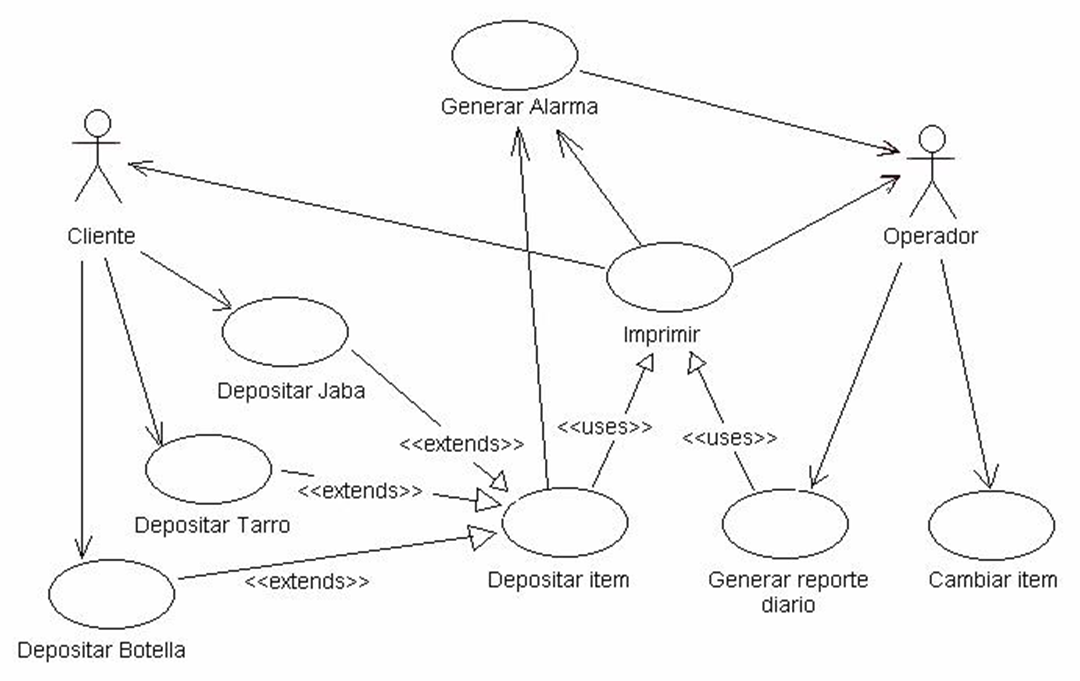


Diagrama de Casos de Uso.

1. Diagrama de un sistema de información.

Un diagrama de flujo de datos (DFD) traza el flujo de la información para cualquier proceso o sistema. Emplea símbolos definidos, como rectángulos, círculos y flechas, además de etiquetas de texto breves, para mostrar las entradas y salidas de datos, los puntos de almacenamiento y las rutas entre cada destino.

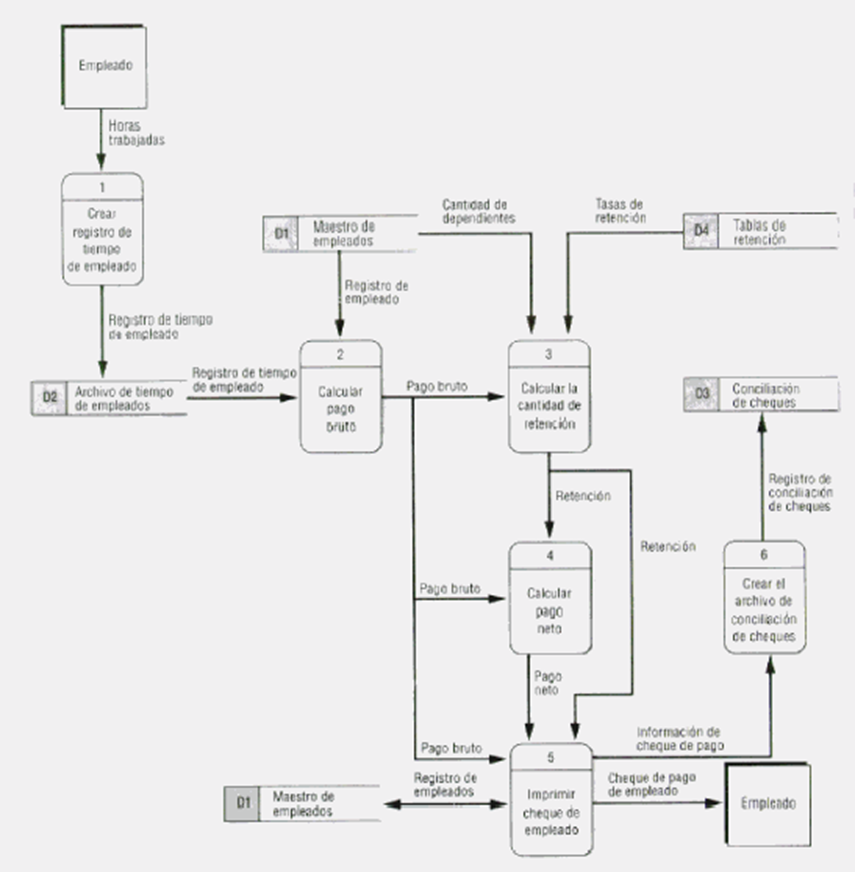


Diagrama de flujo de datos.

**CONCLUSIONES.**

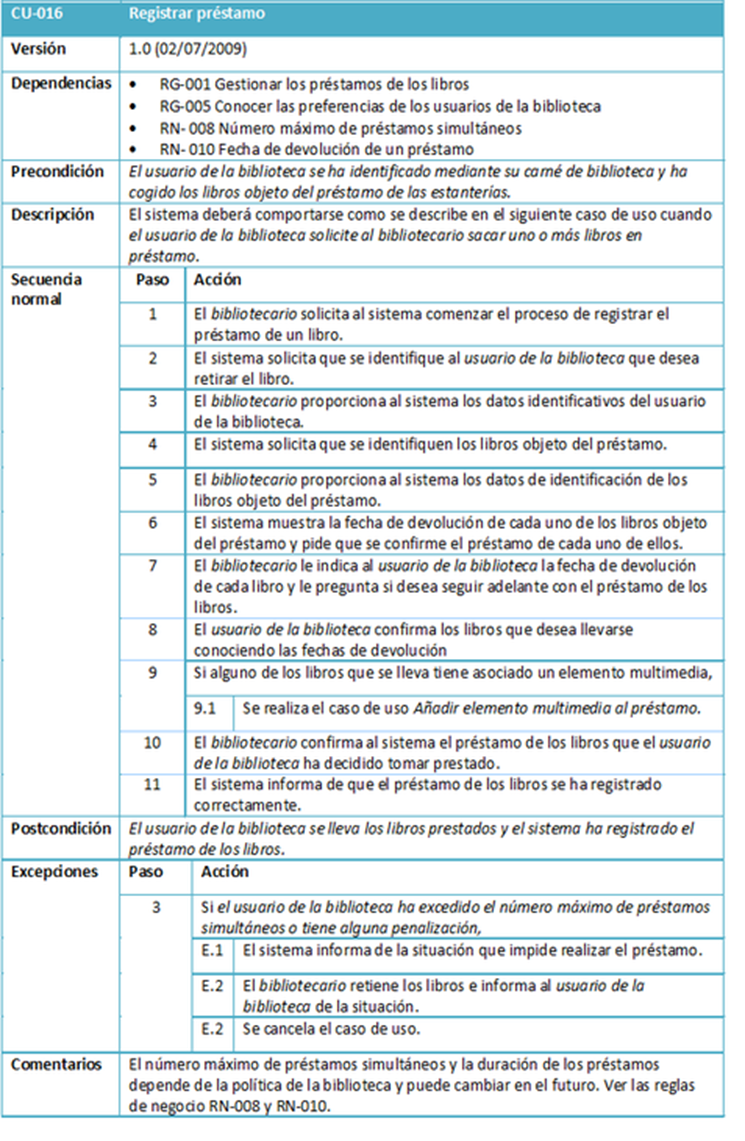
Al finalizar el informe se puede concluir que el Sistema de Información es muy esencial para la gerencia y administración de las empresas, ya que comprende varios componentes o pasos, llevados a cabo durante la etapa del análisis, el cual ayuda a traducir las necesidades del cliente, en un modelo de Sistema que utiliza uno o más de sus componentes.

El ciclo de vida de estos sistemas, son muy importantes, debido a que permiten a una organización o empresa, tener mejor conocimiento sobre el sistema y además permite analizarlos, con la finalidad de mejorarlos.

Por tal motivo para que este ciclo funcione correctamente es necesario contar con la participación de un software, hardware, datos, red y personas.

Finalmente, debemos considerar que los sistemas de información, poseen un ciclo de vida que deben cumplir. Este tipo de ciclo es esencial, puesto que los datos y respaldos de información dependen de su óptimo funcionamiento.

**ANEXOS.**

****

Casos de Uso de Sistemas de Información Web (F.1).

****

Ciclo de vida clásico de un sistema de información, modelo en cascada (F.2).

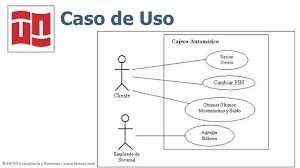


Diagrama de Casos de Uso (F.3).

**REFERENCIAS BIBLIOGRÁFICAS.**

Jose A. Senso. 28 septiembre, 2016. Los sistemas de información basados en la web. Disponible en: https://blogs.ugr.es/tecweb/los-sistemas-informacion-basados-la-web/

Junta de Andalucía. Casos de Uso. Disponible en: https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/416#:~:text=Un%20caso%20de%20uso%20contiene,ni%20explican%20c%C3%B3mo%20se%20implementar%C3%A1.

Power data. 2022. Ciclo de vida de un Sistema de Información Web. Disponible en: https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/ciclo-de-vida-de-un-sistema-de-informacion-fases-y-componentes

Redacción KeepCoding. 25 de enero de 2023. ¿Qué es el ciclo de vida del desarrollo del sistema?. Disponible en: https://keepcoding.io/blog/ciclo-de-vida-del-desarrollo-del-sistema/#:~:text=El%20ciclo%20de%20vida%20del%20desarrollo%20del%20sistema%20(SDLC%2C%20Systems,en%20un%20sistema%20de%20informaci%C3%B3n.