



Universidad Nacional de Ingeniería
UNI - RUACS
Facultad de Ciencias y Sistemas
Ingeniería de Sistemas.



Ingeniería de Software II Mantenimiento del Software.

Autores.

- Br. Norvin Antonio Gutiérrez Gutiérrez. Carnet: 2019-0015N
- Br. Axel Roberto Talavera Briones. Carnet: 2018-0376N
- Br. Ernesto Saúl Potosme Artola. Carnet: 2017-0300N

Grupo: 3T1-IS

Facilitador/ a.

- Ing. Juan José Zeledón Benavidez.

Tabla de contenido

| | |
|---|-----------|
| 1- Introducción. | 5 |
| 2- Delimitación del Problema. | 6 |
| 3- Objetivos. | 7 |
| 3.1- Objetivo General. | 7 |
| 3.2- Objetivos Específicos. | 7 |
| 4- Primer Capítulo. | 8 |
| 4.2- Estado del Arte en Software. | 8 |
| 5- Segundo Capítulo. | 10 |
| 5.2- Mantenimiento del Software. | 10 |
| 5.2.1- Tipos de mantenimiento. | 10 |
| 5.2.3- Coste de Mantenimiento | 11 |
| 5.2.4- Actividades de mantenimiento | 12 |
| 5.3- Refactorización de Software | 14 |
| 5.4- Ingeniería inversa. | 15 |
| 5.5- Reestructurar el Programa | 16 |
| 5.6- Ingeniería directa. | 16 |
| 5.7- Reutilización de Componentes. | 17 |
| 5.7.1- Proceso de reutilización. | 17 |
| 5.8- La importancia de realizar el mantenimiento del software. | 18 |
| 5.8.3- Pérdida de funcionamiento de los Softwares. | 18 |
| 5.8.4- Vulnerabilidad de los Softwares. | 18 |
| 5.9- Ventajas de la realización de un buen mantenimiento de software. | 19 |
| 5.9.1- El mantenimiento preventivo de un software empresarial | 19 |
| 5.9.2- Beneficios del mantenimiento correctivo en el software de la empresa. | 19 |
| 5.9.3- Beneficios del mantenimiento predictivo en el software de la empresa | 20 |
| 5.10- Cuestiones claves en el mantenimiento de Software. | 20 |
| 5.10.1- Asuntos Técnicos. | 21 |
| 5.10.2- Asuntos de Gestión. | 22 |
| 5.10.3- Estimación de costos del mantenimiento. | 24 |
| 5.10.4- Medidas específicas. | 24 |
| 6- Tercer Capítulo. | 26 |

| | |
|--|-----------|
| 6.1- Criterios para el diseño del plan de Mantenimiento. | 26 |
| 6.1.1- ACTIVIDADES DE MANTENIMIENTO:..... | 26 |
| 6.1.2- Dificultades presentes en el mantenimiento. | 27 |
| 6.1.3- Problemas de Mantenimiento. | 28 |
| 6.1.4- Efectos secundarios del mantenimiento. | 28 |
| 6.1.5- Distintos enfoques para la facilidad de Mantenimiento: | 29 |
| 6.2- Herramientas Informáticas para el mantenimiento | 30 |

Índice de Ilustraciones.

| | |
|--|----|
| Ilustración 1 - Costes de Mantenimiento. _____ | 11 |
| Ilustración 2 - Actividades de Mantenimiento. _____ | 14 |
| Ilustración 3 - Refactorización de Software _____ | 15 |
| Ilustración 4 - Proceso de Ingeniería Inversa. _____ | 16 |
| Ilustración 5 - Proceso de Ingeniería Directa. _____ | 16 |

1- Introducción.

El presente documento contiene el plan de mantenimiento de Software aplicado a la norma ISO/IEC 14764 para el negocio llamado Jamen's Pollo y su sistema de información como tal.

Es de mucha importancia la aplicación de este en lo que corresponde al ciclo de vida del software, ya que por medio de esto proporcionará la guía necesaria para poder realizar un mantenimiento a los softwares desarrollados y que de tal manera no acudan a terceros para su debida contratación.

Por medio de su estructura, se describen lo que viene siendo la problemática, sus objetivos a cumplir, entre otros apartados de suma importancia para la persona que vaya a guiarse por medio de este documento.

De igual manera, se contextualiza lo que es la teoría referente a los tipos de mantenimiento de Software, la norma ISO/IEC 14764 y ciertos casos de estudios de mantenimiento de software. Así mismo se abarca lo que son los aspectos necesarios a tener en cuenta al momento de llegar a desarrollar un plan de mantenimiento.

El diseño de las plantillas para el mantenimiento del Software aplicada a la norma ISO/IEC 14764 también estará presente en el desglose respectivo de este documento.

2- Delimitación del Problema.

El respectivo plan y desarrollo de un software deben de llevarse a cabo a la par de cada organización o entidad, ya que ambas generan una buena cantidad de información la cual debe analizarse y seguidamente documentarse. Ahora bien, la falta de conocimiento referente a los mantenimientos de software puede causar una perdida de valor en su importancia y de tal manera relacionarse directamente al mantenimiento del software únicamente con la corrección de algunos detalles en sus programas.

Como parte de la arquitectura del ciclo de vida de un software se da lo que es el mantenimiento del software, por cual se pretende desarrollar un plan de gestión de mantenimiento para el negocio Jamen's Pollos y aplicar la norma ISO/IEC 14764:2006, la cual proporcionará la guía correspondiente para llevar a cabo el mantenimiento de manera correcta a los softwares desarrollados.

3- Objetivos.

3.1- Objetivo General.

- Diseñar un plan de gestión de mantenimiento de software para el área de desarrollo de software del negocio Jamen´s Pollos, ubicado en la ciudad de Estelí, Nicaragua.

3.2- Objetivos Específicos.

- Identificar los tipos de mantenimiento de software disponibles en la actualidad.
- Realizar un estudio del estado del arte de los estándares para mantenimiento de software.
- Diseñar un plan de mantenimiento para el área de desarrollo del negocio Jamen´s Pollos.
- Evaluación de herramientas informáticas que automatice el mantenimiento.

4- Primer Capítulo.

4.2- Estado del Arte en Software.

Dentro de la Ingeniería de Software se presentan lo que son la gran variedad de normativas ISO para su aplicación en desarrollo de Software, las fundamentales son:

La norma ISO/IEC 14764:2006

Desarrolla con gran detalle la gestión del proceso de mantenimiento descrito en la ISO/IEC 12207, con correcciones. Establece las definiciones de los tipos de mantenimientos y guías de aplicación para la planificación, ejecución, control, revisión, evaluación y finalización del proceso de mantenimiento. Incluye la forma de mantener muchos productos con los mismos recursos.

Esta norma define el mantenimiento como el conjunto de actividades destinadas a proporcionar soporte económicamente rentable para un determinado producto software. Estas actividades se realizan tanto antes de la entrega del producto como después de la entrega. Las actividades previas a la entrega se destinan a planificar, anticipar y preparar las actividades de mantenimiento posteriores. Las actividades posteriores a la entrega incluyen modificaciones del producto software, formación y asistencia al usuario.

IEEE 1219

Este marco se basa en otros anteriores y recoge las definiciones del IEEE 610.12-1990. Define el mantenimiento como la modificación de un producto software después de su entrega al cliente o usuario para corregir defectos, mejorar el rendimiento u otras propiedades deseables, o para adaptarlo a un cambio de entorno. Establece varias fases dentro del mantenimiento:

- Modificación, identificación, clasificación y priorización de problemas.
- Análisis
- Diseño
- Implementación

- Regresión, y sistema de prueba
- Prueba de aceptación. (Asensio, s.f.)

La norma ISO/IEC 12207.

Se propone como el modelo de referencia de procesos. Aunque se puede trabajar con otros modelos de procesos, siempre se deben cumplir los requisitos mínimos establecidos por la norma ISO 12207. “un marco de referencia que contiene los procesos, las Actividades y las tareas involucradas en el desarrollo, explotación y mantenimiento de un producto software, abarcando la vida del sistema desde la definición de requisitos hasta que se deja de utilizar” (ingertec, s.f.)

5- Segundo Capítulo.

5.2- Mantenimiento del Software.

Según (Tutorialspoint, s.f.) el mantenimiento del Software es hoy en día aceptado como parte del SDLC. Se refiere a todas las modificaciones y actualizaciones que se llevan a cabo después de la entrega del producto software. Hay muchas razones por las que las modificaciones son necesarias, algunas de ellas se mencionan de manera breve a continuación:

Condiciones de mercado - Leyes, las cuales cambian con el paso del tiempo, como los impuestos o limitaciones que se han introducido de recientemente, como es el caso de: Cómo mantener la contabilidad, puede desencadenar en una necesidad de modificación.

Requisitos del cliente - A medida que pasa el tiempo, el consumidor puede pedir nuevas funciones o características en el software.

Modificaciones del servidor - Si algún tipo de hardware y/o plataforma (como es el caso de un sistema operativo) del servidor target cambia, se requerirán cambios en el software para mantener la adaptabilidad.

Cambios organizativos - Si se produjera algún cambio a nivel de negocio con el consumidor final, como por ejemplo una reducción de la fuerza organizacional, adquiriendo otra compañía, un emprendimiento organizacional en un nuevo negocio, puede que se necesite modificar algo del software original.

5.2.1- Tipos de mantenimiento

En el ciclo vital del software, el tipo de mantenimiento puede variar según la naturaleza del producto. Puede que sea simplemente una tarea rutinaria de mantenimiento porque algún usuario ha encontrado un virus, o puede tratarse propiamente de un gran evento basado en la magnitud del mantenimiento o en su naturaleza. A continuación, presentamos algunos tipos de mantenimiento fundamentados en sus características:

Mantenimiento correctivo - Este tipo incluye las modificaciones y actualizaciones que se han hecho con tal de corregir o resolver problemas descubiertos por el usuario o se han encontrado en informes de error de algún usuario.

Mantenimiento adaptable - Este tipo incluye modificaciones y actualizaciones que se han aplicado para mantener el producto software al día y en consonancia con el siempre cambiante mundo de las tecnologías y entornos de negocio.

Mantenimiento perfectivo - Esto incluye las modificaciones y actualizaciones que se han realizado con tal de mantener el software usable por un largo periodo de tiempo. Aquí se incluyen nuevas características, nuevos requisitos para perfeccionar el software y mejorar su fiabilidad y su rendimiento.

Mantenimiento preventivo - Incluye las modificaciones y actualizaciones para prevenir problemas de software en un futuro. Pretende ocuparse de problemas, que no son muy significativos por el momento pero que podrían ocasionar graves conflictos en un futuro.

5.2.3- Coste de Mantenimiento

Los informes insinúan que el coste de mantenimiento es alto. Un estudio realizado para estimar el mantenimiento de software concluyó que el coste de mantenimiento representa un 67% del coste total en el ciclo del proceso de software.

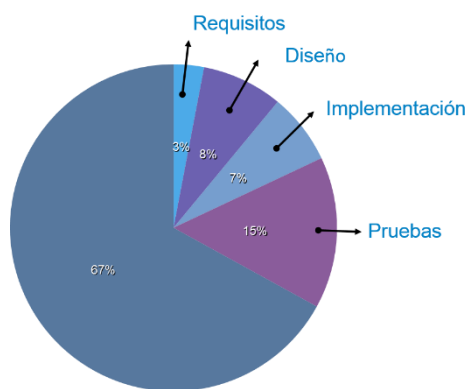


Ilustración 1 - Costes de Mantenimiento.

El promedio del coste del mantenimiento de software constituye más del 50% en todas las fases del SDLC. Hay varios factores, que inducen el aumento del coste de mantenimiento, como es el caso de:

Factores reales que afectan al coste de mantenimiento

- La edad media de un software se sitúa entre 10 y 15 años.
- Los softwares más viejos, diseñados para trabajar en máquinas lentas y con menos memoria y capacidad de almacenaje, no pueden mantenerse en el mercado con la competencia de nuevos y perfeccionados softwares en hardware modernos.
- A medida que la tecnología avanza, se vuelve más caro mantener software antiguo.
- La mayoría de ingenieros de mantenimiento son principiantes y usan métodos de error y de prueba para rectificar problemas.
- A menudo, los cambios que se hacen pueden fácilmente dañar la estructura original del software, dificultando cambios posteriores.
- Los cambios se suelen dejar indocumentados, lo que puede ocasionar más conflictos en el futuro.

Factores del software final que afectan a los costes de mantenimiento

- Estructura del programa de Software
- Lenguaje de programación
- Dependencia a entornos externos
- Disponibilidad y fiabilidad de personal

5.2.4- Actividades de mantenimiento

El IEEE proporciona un borrador para las actividades del proceso secuencial de mantenimiento. Se puede usar de forma reiterativa y puede extenderse para que los artículos personalizados puedan incluirse.

Estas actividades van cogidas de la mano con cada una de las siguientes fases:

- Identificación & Seguimiento - Incluye las actividades que pertenecen a la identificación de requisitos de modificación o mantenimiento. Es generado

por el usuario o el mismo sistema puede anunciar a través de mensajes de error o registros. Aquí, el tipo de mantenimiento también se clasifica.

- **Análisis** - La modificación se analiza por su impacto en el sistema, incluyendo implicaciones de seguridad. Si un probable impacto es severo, se busca una solución alternativa. Un conjunto de modificaciones requeridas se materializa entonces en requisitos del sistema. El coste del mantenimiento/modificación se analiza y se concluye con una estimación.
- **Diseño** - Nuevos módulos, que necesitan ser modificados o reemplazados, se diseñan en contra de los requisitos que ya se han fijado en la fase previa. Las pruebas de casos se han creado para la validación y la verificación.
- **Implementación** - Los nuevos módulos son codificados con la ayuda del diseño estructurado creado en la fase de diseño. Cada programador debe hacer pruebas unitarias en paralelo.
- **Evaluación del sistema** - Las pruebas de integración se hace entre nuevos módulos creados. Las pruebas de integración también se llevan a cabo entre módulos nuevos y el sistema. Finalmente, el sistema se evalúa en su conjunto, siguiendo procedimientos evaluativos reaccionarios.
- **Pruebas de aceptación** - Después de evaluar el sistema de manera interna, se evalúa la aceptación con la ayuda de los consumidores. Si en esta etapa, los consumidores se quejan de algún asunto, son redirigidos o se les notifica que se dirijan a la siguiente repetición.
- **Entrega** - Después del test de aceptación, el sistema se implementa en la totalidad de la organización con pequeños paquetes de actualizaciones o con la instalación nueva del sistema. La evaluación final se da con el consumidor final después de entregar el software. Se provee formación si se requiere, además de una copia en papel del manual del usuario.
- **Gestión de mantenimiento** - La gestión de la configuración es una parte esencial del mantenimiento del sistema. Es auxiliado con herramientas de control de versiones, semi - versiones o Gestión de parches.



Ilustración 2 - Actividades de Mantenimiento.

5.3- Refactorización de Software

Cuando necesitamos actualizar el software para mantenerlo en el mercado actual, sin afectar a su funcionalidad, estamos ante un caso de refactorización de software. Es un proceso en el que el diseño del software se cambia y los programas se escriben de nuevo.

El software heredado no puede adaptarse a las nuevas y más recientes tecnologías disponibles en el mercado. Como que el hardware se vuelve obsoleto, la actualización de software se convierte en un dolor de cabeza. Aunque el software envejezca con el tiempo, sus funcionalidades no hacen los mismo.

Por ejemplo, Unix fue desarrollado en lenguaje ensamblador. Cuando el lenguaje C empezó a existir, Unix fue refactorizado en C, porque trabajando en el lenguaje previo era difícil.

A parte de este caso, a veces los programadores notan que en algunas partes del software se necesita más mantenimiento que en otras, y también necesitan refactorización.

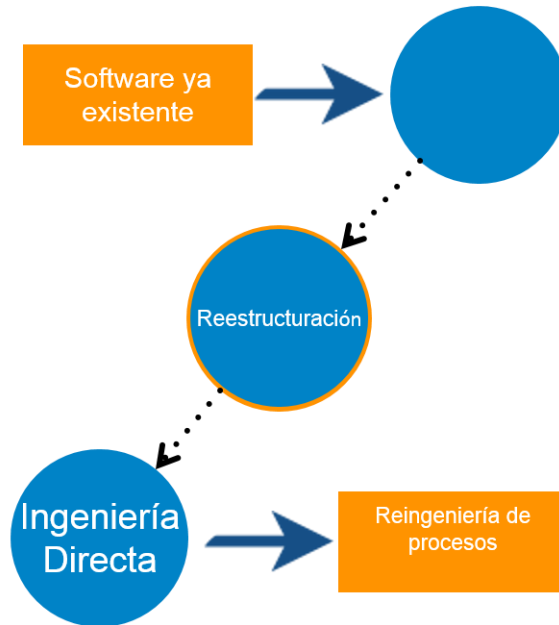


Ilustración 3 - Refactorización de Software

Proceso de refactorización

- Decidir Lo que va a ser refactorizado, si va a ser una parte del software o su totalidad.
- Desarrollar La ingeniería inversa, con tal de obtener especificaciones de software ya existentes.
- Reestructurar el programa Si se requiere. Por ejemplo, cambiar programas orientados a la función en programas orientados al objeto.
- Reestructurar datos tal y como se requiera.
- Aplicar conceptos de Ingeniería directa o deductiva con tal de que el software pase por un proceso de refactorización.

5.4- Ingeniería inversa

Es un proceso para lograr especificaciones del sistema a través del análisis y el entendimiento del sistema existente. Este proceso puede verse como un modelo inverso de SDLC, esto es, intentamos lograr mayor nivel de abstracción analizando niveles de abstracción más bajos.

En un sistema existente se implementa un diseño previamente, sobre el que no sabemos nada. Los diseñadores utilizan entonces ingeniería inversa observando el código e intentan lograr el diseño. Con el diseño en mano, intentan encontrar las especificaciones. Por consiguiente, van al revés des del código hasta los requisitos del sistema.

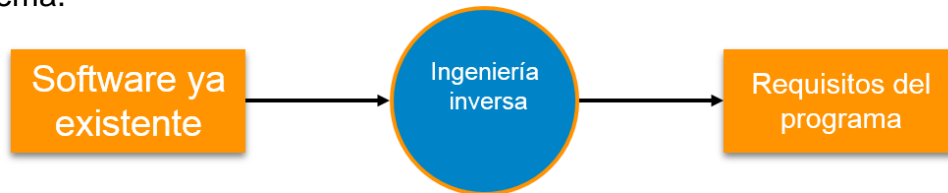


Ilustración 4 - Proceso de Ingeniería Inversa.

5.5- Reestructurar el Programa

Es un proceso de reestructuración y construcción de software ya existente. Se vuelve a organizar el código de origen, en el mismo lenguaje de programación o con otro distinto. La reestructuración puede tener su origen en la reestructuración del código, de los datos o de ambos.

La reestructuración no tiene un impacto en la funcionalidad del software, pero aumenta la fiabilidad y la capacidad de su mantenimiento. Los elementos del Programa, que ocasionan errores frecuentemente pueden cambiarse o actualizarse con la reestructuración.

La dependencia del software en plataformas hardware obsoletas se puede eliminar a través de la reestructuración.

5.6- Ingeniería directa

La ingeniería directa es un proceso para obtener el software deseado a partir de los requisitos que se han obtenido mediante la ingeniería invertida. Asume que había alguna ingeniería de software ya hecha en el pasado.

La ingeniería directa es igual que en el proceso de ingeniería de software, pero con una diferencia – Siempre se lleva a cabo después de la ingeniería inversa.

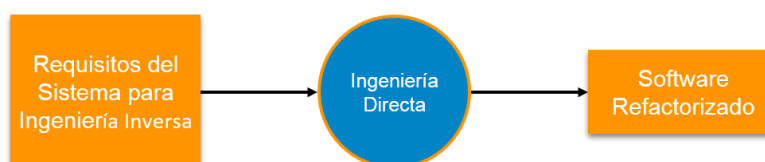


Ilustración 5 - Proceso de Ingeniería Directa.

5.7- Reutilización de Componentes.

Un componente es parte del código del programa software, el cual ejecuta tareas independientes en el sistema. Puede ser un módulo pequeño o un subsistema en sí mismo.

La reutilización se puede hacer en varios niveles

- Nivel de aplicación - Cuando una aplicación entera se usa como sub sistema de software nuevo.
- Nivel de Componente - Cuando el subsistema de una aplicación es usado.
- Nivel de módulos - Donde los módulos funcionales son reutilizados.

Los componentes del Software aportan interfaces, que pueden usarse para establecer comunicación entre los distintos componentes.

5.7.1- Proceso de reutilización

Se pueden adoptar dos tipos de método: o manteniendo los requisitos de la misma manera y ajustando los componentes, o manteniendo los componentes igual, pero modificando los requisitos.

- Requisitos del sistema - Los requisitos funcionales y no funcionales del producto software se especifican, y este debe cumplirlos, con la ayuda del sistema existente, una entrada (input) de un usuario, o con ambos.
- Diseño - Esto es también un paso del proceso estándar de SDLC, donde los requisitos se definen en términos de jerga de software. La arquitectura básica del sistema en su totalidad y sus subsistemas son creados.
- Especificar componentes - Mediante el estudio del diseño software, los diseñadores dividen la totalidad del sistema en pequeños componentes o subsistemas. Un diseño de software completado se vuelve una gran colección de componentes trabajando juntos.
- Buscar componentes adecuados - El repositorio de componentes software es consultado por los diseñadores para buscar el componente que combina, en base a la funcionalidad y a los requisitos del software.
- Incorporar Componentes - Todos los componentes emparejados y combinados se empaquetan juntos para convertirlos en software completo.

5.8- La importancia de realizar el mantenimiento del software.

La importancia de realizar el mantenimiento del software en una empresa radica en que los equipos mejoran en cuanto a operatividad, además que se refuerzan los parámetros de seguridad de los programas informáticos. Estos programas son los que soportan las actividades financieras y comerciales de la empresa, por ello contribuyen a mejorar los programas que proveen las soluciones logísticas de cara al mercado, en este enlace encontrarás más información sobre ello. Un buen mantenimiento de software no solo modifica y corrige, sino que permite que se conozca el estado y las condiciones de funcionamiento.

5.8.3- Pérdida de funcionamiento de los Softwares.

Que los equipos y el software pierdan la capacidad de operar de manera correcta, puede acarrear un incremento del coste y pérdidas importantes para la empresa. En cuanto al software, esta situación impulsa, otras consecuencias, puntuales, tanto en el procesamiento, rendimiento, programación y documentación.

El funcionamiento incorrecto del software significa que, los programas instalados presentarán salidas incorrectas y el tiempo de respuesta se incrementará. Adicionalmente, aparecerán múltiples inconsistencias. Es importante destacar que, cada software responde a un diseño de inicio y con el pasar del tiempo, el modelo queda desfasado.

5.8.4- Vulnerabilidad de los Softwares.

Un buen mantenimiento del software de la empresa contribuye a evitar los accesos no autorizados y los códigos maliciosos, que representan una amenaza para todo el sistema informático. Las actualizaciones y las mejoras de los programas contribuyen a detectar irregularidades exteriores que pueden causar gran daño a los sistemas operativos de la empresa.

El mantenimiento más importante, que puede recibir el software de la empresa, es el preventivo. Este constituye un proceso en el que se realizan actualizaciones de antivirus, actualización de aplicaciones y controladores, cambio de contraseñas, limpieza del registro del sistema y la desfragmenta los discos duros, entre otras actividades, que aumentan la seguridad del software y optimizan su funcionamiento.

Los hackers están constantemente al acecho en busca de las vulnerabilidades del software de las empresas. Aunque las empresas que desarrollan software mantengan el enfoque en garantizar un producto con niveles de protección a través de parches y actualizaciones, el riesgo es un hecho latente, es por ello que tanto la autenticidad de la licencia de software y el mantenimiento son estratégicos.

5.9- Ventajas de la realización de un buen mantenimiento de software.

Existen varios tipos de mantenimiento de software, perfectamente aplicables a los niveles empresariales y que constituyen una garantía de seguridad y funcionalidad dentro de los estándares óptimos. Veamos las ventajas de un buen mantenimiento de software.

5.9.1- El mantenimiento preventivo de un software empresarial

Entendiendo que el software en una empresa puede representar el cerebro de operaciones estratégicas, tanto de la gestión comercial y de trabajo. El mantenimiento preventivo confiere una serie de beneficios:

- Constituye el mecanismo periódico que mejora la protección de los datos y del sistema en general. Evita la incursión de malware y contrarresta las posibilidades que despliegue sus formas de infección. Por lo tanto, las tareas preventivas de mantenimiento ayudan a incrementar los niveles de seguridad que necesita el software de la empresa.
- Reduce los fallos en programas y sus componentes.
- Disminución del tiempo de respuesta a cualquier acción dentro de los programas instalados.
- Realiza tareas de Backups, de manera que se cuente con las copias de seguridad que exige la organización, como una manera de resguardar su información.

5.9.2- Beneficios del mantenimiento correctivo en el software de la empresa.

El mantenimiento correctivo, corrige fallos detectados en el software, a través de una serie de acciones. El fundamento de este tipo de mantenimientos, ronda en el objetivo de reparar, más que de sustituir. Trabaja en base a los resultados de los mantenimientos preventivos y predictivos. Entre las ventajas que se pueden obtener a través de este tipo de mantenimiento, tenemos:

- Daños causados por virus, spyware o malwares, el mantenimiento correctivo se encarga de ello, y restablece el sistema operativo de manera limpia. Para ello destaca las medidas de seguridad con las que se intenta reforzar la seguridad del software.
- Corrige la incompatibilidad de drivers.
- Instala y configura los parámetros de los servicios de Internet, además de la programación de routers Wifi.

5.9.3- Beneficios del mantenimiento predictivo en el software de la empresa

Este tipo de mantenimiento es estratégico, porque se hace para detectar posibles errores en el futuro, además que se maneja con el tiempo de vida útil de los componentes y del sistema lógico.

Responde a un tipo de programación o plan que se realiza en base a información de adquisición e instalación de elementos, programas o drivers, asimismo del monitoreo de los diversos fallos en el software que se puedan presentar en un determinado tiempo.

La ventaja del mantenimiento predictivo consiste en lograr proyectar el tiempo de vida de componentes esenciales, lo que garantiza que se corrija a tiempo y evite que se produzca un caos en el software de la empresa. (GRUPOSIM, 2020)

5.10- Cuestiones claves en el mantenimiento de Software.

Para asegurar que el mantenimiento de software sea eficaz, es importante comprender que el mantenimiento del software ofrece desafíos técnicos y de gestión únicos para los ingenieros de software. A continuación, se presenta algunas de las cuestiones técnicas y de gestión relacionados con el mantenimiento del software.

Se han agrupado en los siguientes apartados temáticos:

- Asuntos técnicos.
- Asuntos de gestión.
- Estimación de costos del mantenimiento.
- Medidas específicas

5.10.1- Asuntos Técnicos.

Comprensión limitada

La comprensión limitada es la rapidez con la que un ingeniero de software puede entender dónde hacer un cambio o una corrección en el software que este individuo no desarrolló.

Alrededor del 40% al 60% del esfuerzo de mantenimiento está dedicada a comprender el software a ser modificado. Por lo tanto, los ingenieros de software pueden tener inicialmente una comprensión limitada del software, y mucho tiene que hacer para remediar esto.

Pruebas

El repetir una prueba es significativo en tiempo y dinero. Las pruebas de regresión, la repetición de pruebas selectivas de un componente de software o para verificar que las modificaciones no han causado efectos no deseados, es importante para el mantenimiento.

Cuando software realiza funciones críticas, puede que sea imposible llevarla fuera de línea para la prueba

Análisis de impacto

El análisis de impacto detalla cómo realizar técnicamente un análisis completo del impacto de un cambio en el software existente. Los encargados del mantenimiento deben poseer un conocimiento profundo de la estructura y el contenido del software, todo esto para poder realizar una estimación de los recursos necesarios para llevar a cabo el cambio y conocer el riesgo de hacer el cambio determinado. La solicitud de cambio se llama una petición de modificación (MR) y, a menudo llamado un informe de problemas (PR), primero debe ser analizado y traducido en términos de software. Los objetivos de análisis de impacto son:

- Determinación del alcance de un cambio con el fin de planificar y ejecutar el trabajo.

- Desarrollo de estimaciones precisas de los recursos necesarios para realizar el trabajo.
- El análisis de los costos / beneficios del cambio solicitado.
- Comunicación con otros de la complejidad de un cambio dado.

Mantenibilidad

La mantenibilidad en IEEE 1219 es definida como: la facilidad con la que el software se puede mantener, mejorar, adaptar, o ser corregido para satisfacer los requisitos especificados. La norma ISO 9126-01 define la mantenibilidad como una de las características de calidad.

La condición de mantenibilidad reduce los costos de mantenimiento, por lo tanto, el proceso de mantenimiento del software mejorará. Muchas veces durante el desarrollo se ignora la mantenibilidad considerando que no es importante por parte de los desarrolladores siendo incluso muchas veces un requerimiento del encargado de mantenimiento. Esto se desencadena en la falta de documentación del sistema, que es la principal causa de las dificultades en la comprensión del programa y análisis de impacto.

5.10.2- Asuntos de Gestión.

Alineación con los objetivos de la organización

Los objetivos de la organización detallan el retorno de la inversión de las actividades de mantenimiento de software. No obstante, el mantenimiento del software a tiene el objetivo de alargar la vida del software. Las actividades de mantenimiento pueden ser impulsado por la necesidad de satisfacer la demanda del usuario para las actualizaciones y mejoras del software. En ambos casos, el retorno de la inversión es mucho menos clara, de modo que la vista en alto nivel de gestión es a menudo de una de las principales actividades que consumen recursos significativos sin ningún beneficio cuantificable clara para la organización.

Contratación de personal

El mantenimiento a menudo no es visto como un trabajo glamoroso, en muchas organizaciones el personal de mantenimiento de software frecuentemente es mal visto. De allí que es importante atraer y mantener al personal de mantenimiento de software.

Proceso

El proceso de software es el conjunto de actividades, métodos, prácticas y transformaciones que la gente usa para desarrollar y mantener el software y los productos asociados. A nivel de proceso, las actividades de mantenimiento de software comparten mucho en común con el desarrollo de software, es así, que la gestión de configuración de software es una actividad crucial en ambos. El mantenimiento también requiere varias actividades que no se encuentran en el desarrollo de software como son: actividades únicas, de apoyo, de planificación del mantenimiento, Gestión de configuración de software y calidad del software.

Aspectos organizativos del mantenimiento

Los aspectos organizativos describen quien será responsable del mantenimiento del software, el equipo que desarrolla el software no debería estar asignado para mantener el software. Sin embargo, la decisión debe ser tomada en una base de caso. Lo que es importante es la delegación o asignación de la responsabilidad de mantenimiento a un solo grupo o persona independientemente de la estructura de la organización.

Externalización

Actualmente la externalización del mantenimiento es una gran industria, las grandes empresas están externalizando carteras enteras de sistemas de software, incluyendo el mantenimiento del software. Uno de los principales retos para los subcontratistas es determinar el alcance de los servicios de mantenimiento requeridos y los detalles contractuales. Las empresas de subcontratación suelen pasar varios meses evaluando el software antes de que entren en una relación contractual.

5.10.3- Estimación de costos del mantenimiento.

A efectos de planificación, estimación de costos es un aspecto importante del mantenimiento del software.

La estimación de costos

Las estimaciones de los costos de mantenimiento se ven afectados por muchos factores técnicos y no técnicos, la norma ISO/IEC 14764 establece que "los dos enfoques más populares a los recursos que estiman para el mantenimiento del software son el uso de modelos paramétricos y el uso de la experiencia". Muy a menudo, se utiliza una combinación de éstos.

Los modelos paramétricos:

Algunos trabajos se han realizado en la aplicación de modelos de costos paramétrico para el mantenimiento del software. Muy importante en este tipo de casos es poseer los datos de proyectos anteriores con el fin de utilizar los modelos.

Experiencia:

La experiencia, en forma de juicio de expertos, analogías, y una estructura de desglose del trabajo, varios enfoques que deberían utilizarse para aumentar los datos de modelos paramétricos. Es evidente que el mejor enfoque para la estimación de mantenimiento es combinar los datos y la experiencia empírica. Estos datos deben ser proporcionados como resultado de un programa de medición.

5.10.4- Medidas específicas.

El encargado de mantenimiento debe determinar qué medidas son adecuadas para la organización, existe una lista de serie de medidas para cada uno de los mantenimientos:

Analizabilidad: Medidas de esfuerzo o recursos gastados en tratar de diagnosticar deficiencias o causas del fracaso del mantenedor, o en la identificación de las piezas que se desea modificar.

Mutabilidad: Medidas del esfuerzo del mantenedor asociados con la implementación de una modificación especificada.

Estabilidad: Medidas del comportamiento inesperado de software, incluyendo que se encontró durante la prueba.

Comprobabilidad: Medidas del mantenedor del esfuerzo y de los usuarios en el intento de probar el software modificado

Algunas de las medidas de la capacidad de mantenimiento de software se pueden obtener utilizando herramientas comerciales disponibles. (DESARROLLO DE UN PLAN DE GESTIÓN DE MANTENIMIENTO, s.f.)

6- Tercer Capítulo.

6.1- Criterios para el diseño del plan de Mantenimiento.

6.1.1- ACTIVIDADES DE MANTENIMIENTO:

Las actividades de mantenimiento se agrupan en tres categorías funcionales:

Comprensión del software y de los cambios a realizar (Comprender):

es necesario el conocimiento a fondo de la funcionalidad, objetivos, estructura interna y requisitos del software. Alrededor del 50% de tiempo de mantenimiento se dedica a esta actividad, a consecuencia de lo cual, las herramientas CASE incorporan utilidades que automatizan este tipo de tareas aumentando de manera notable la productividad.

Modificación del software (Corregir):

crear y modificar las estructuras de datos, la lógica de procesos, las interfaces y la documentación. Los programadores deben evitar los efectos laterales provocados por sus cambios. Esta actividad representa $\frac{1}{4}$ del tiempo total de mantenimiento.

Realización de pruebas (Comprobar):

realizar pruebas selectivas que nos aseguren la corrección del software.

| Categoría. | Actividad | % Tiempo |
|--|-----------------------------|----------|
| Comprensión del software y de los cambios a realizar | Estudiar las Peticiones. | 18% |
| | Estudiar la documentación. | 6% |
| | Estudiar el código. | 23% |
| Modificación del software | Modificar el código | 19% |
| | Actualizar la documentación | 6% |
| Realización de pruebas | Diseñar y realizar pruebas | 28% |

Tabla 1 - Actividades de Mantenimiento.

6.1.2- Dificultades presentes en el mantenimiento.

CÓDIGO HEREDADO:

La mayor parte del software en la actualidad está formada por código antiguo “heredado” (legacy code); esto es, código desarrollado hace tiempo, con técnicas y herramientas en desuso y, para más INRI, por personas que actualmente no se encargan de su mantenimiento. Además, puede que incluso este código haya pasado varias actividades de mantenimiento; y, por otra parte, el volver a reescribirlo no compensa por la carga financiera que supuso y la necesidad de su amortización.

Leyes del mantenimiento del Software.

Continuidad del cambio.

Un programa utilizado en un entorno del mundo real debe cambiar si no quiere dejar de ser usado. Esto se debe a que surgen nuevas funcionalidades, nuevo hardware puede permitir mejoras en el software, se corrigen defectos, se instala en otro sistema operativo/máquina o el software necesita ser más eficiente.

Incremento de la Complejidad.

Cuando se realizan cambios en un programa la estructura se hace más compleja si no se utilizan técnicas de ingeniería del software.

Evolución del programa.

Es un proceso autorregulado. Se mantienen las tendencias e invariantes de las propiedades del programa.

Conservación de la Estabilidad Organizacional.

La carga que supone el desarrollo de un programa es aproximadamente constante e independiente de los recursos dedicados a lo largo del tiempo de vida del mismo.

Conservación de la Familiaridad.

El incremento en el número de cambios introducidos con cada versión (reléase) es aproximadamente constante.

6.1.3- Problemas de Mantenimiento.

- Es habitual realizar el mantenimiento de forma ad hoc en un estilo libre del programador. Esto es debido a que no existen o son poco conocidos los métodos, técnicas y herramientas que proporcionan soluciones globales al problema del mantenimiento.
- Después de cada cambio los programas tienden a ser menos estructurados. Como consecuencia se produce una documentación desfasada, código que no cumple los estándares, incremento en el tiempo de comprensión de los programas o incremento de los efectos secundarios de los cambios.
- Los sistemas que son mantenidos son cada vez más difíciles de cambiar.
- Los usuarios participan poco en el desarrollo del software, con el riesgo de que no satisfaga sus necesidades y aumenten los esfuerzos en el mantenimiento.
- Problemas de gestión. Existe una visión de que el trabajo de mantenimiento es de una escala inferior al trabajo de desarrollo de software. Se realiza mantenimiento precipitado, no documentado adecuadamente y poco integrado en el código existente.

6.1.4- Efectos secundarios del mantenimiento.

En el mantenimiento del software existe el riesgo del llamado efecto bola de nieve; que consiste en que los cambios introducidos por una petición de mantenimiento conllevan efectos secundarios que implican futuras peticiones de mantenimiento.

Efectos secundarios sobre el código:

1. Cambios en el diseño que suponen muchos cambios en el código.
2. Eliminación o modificación de un subprograma.
3. Eliminación o modificación de una etiqueta.
4. Eliminación o modificación de un identificador.
5. Cambios para mejorar el rendimiento.
6. Modificación de la apertura/cierre de ficheros.
7. Modificación de operaciones lógicas.

Efectos secundarios sobre los datos:

1. Redefinición de constantes locales o globales.
2. Modificación de los formatos de registros o archivos.
3. Cambio en el tamaño de una matriz u otras estructuras similares.
4. Modificación de la definición de variables globales.
5. Reinicialización de indicadores de control o punteros.
6. Cambios en los argumentos de los subprogramas. Es importante una correcta documentación de los datos.

Efectos secundarios sobre la documentación:

1. Modificar el formato de las entradas interactivas.
2. Nuevos mensajes de error no documentados.
3. Tablas o índices no actualizados.
4. Texto no actualizado correctamente.

6.1.5- Distintos enfoques para la facilidad de Mantenimiento:

Una primera aproximación externa sería medir el proceso de mantenimiento; si es efectivo, entonces el producto es mantenible. En esta línea se pueden utilizar medidas de la cualificación del equipo de mantenimiento, de las herramientas disponibles y de la madurez del proceso.

Otra aproximación a la cual se recurre a menudo consiste en identificar atributos internos del producto y determinar cuáles son predictivos. Se basa en la métrica “tiempo medio de cambio” (TMC); que es el tiempo que transcurre desde que se recibe la solicitud de cambio hasta que el sistema cambiado es de nuevo operativo. Esta medida se obtiene de las siguientes variables: tiempo que se tarda en analizar la solicitud de cambio, tiempo de especificación y diseño del cambio, tiempo activo para implementar el cambio y tiempo en probarlo y distribuirlo.

Cuantos más atributos cualitativos se gestionen durante el desarrollo del software, mejor será su facilidad de mantenimiento (mantenibilidad). Desde un enfoque de calidad, estos atributos son: fiabilidad, modularidad, facilidad de comprensión, facilidad de prueba y facilidad de expansión. (Informatica, s.f.)

6.2- Herramientas Informáticas para el mantenimiento

Existen comercialmente numerosos productos en el mercado disponibles para el mantenimiento del software. Un tipo de producto son las herramientas de seguimiento de errores, que desempeñan un papel importante en el mantenimiento. Bugzilla por Mozilla Foundation es un ejemplo de una herramienta de este tipo. Otros productos de control de errores son Test director de Mercury Interactive, Silk Radar de Segue Software, SQA Manager de Rational Software, y QA director de Compuware. Proteus III Expert CMMS de Eagle Technology, Inc. es un paquete de software que permite a los usuarios programar el mantenimiento preventivo, generar órdenes de trabajo automático, historial de mantenimiento de equipos, seguimiento de los activos, inventario, personal, crear órdenes de compra, y generar informes.

Microsoft Visual Source Safe es una herramienta de control del sistema de origen que es utilizada para la administración de configuración. Productos que son específicos de los lenguajes de programación son CCFinder y JAAT que está diseñado específicamente para los programas Java. CCFinder identifica clones de código en el programa JAVA. JAAT ejecuta análisis de programas Java. Para los programas en C++, existe OCL que es una herramienta de depuración de programas C++ mediante consultas formuladas en el lenguaje de restricción de objetos, OCL.

La tarea de mantenimiento de software se ha convertido en un proceso tan vital y complejo que se requiere soporte automatizado para hacerlo con eficacia. El uso de herramientas simplifica las tareas, la eficiencia y aumento de productividad. Existen numerosas herramientas disponibles en el mercado para el mantenimiento. MANTOOL es una herramienta para apoyar el proceso de mantenimiento del software y que permite automatizar su gestión de acuerdo a MANTEMA, una metodología rigurosa para el mantenimiento. Este software permite a los usuarios gestionar las solicitudes de modificación siguiendo las diferentes etapas de la metodología. Los datos guardados en MANTOOL se pueden utilizar para extraer diferentes tipos de informes y para hacer estimaciones de futuras intervenciones de mantenimiento.

7- Cuarto Capítulo.

7.1- Presentación del software a implementar Mantenimiento.

Jamen's Pollos Estelí, Nicaragua.



Plan de Mantenimiento de Software.

Sistema de Información Jamen's Pollos.

Versión 1.1

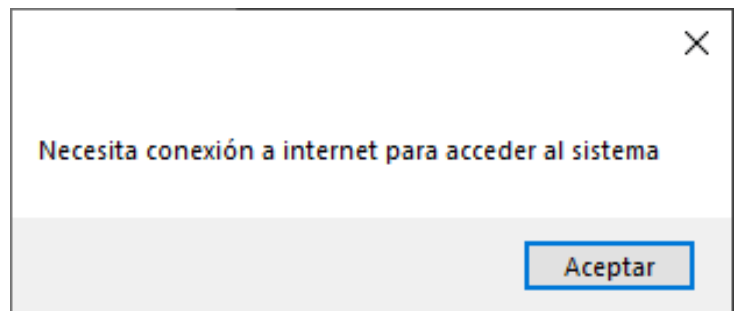
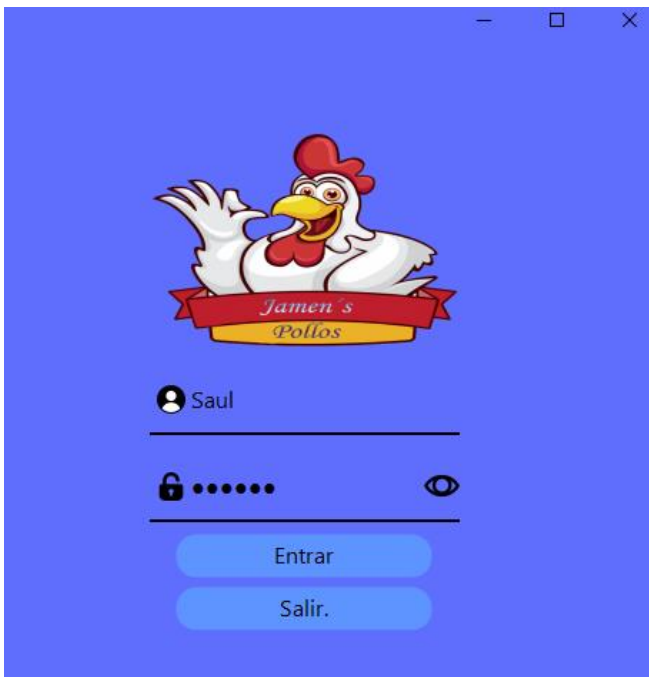
Elaborado por

Smart Solutions.

Fecha:

20 de abril del año 2022.

| Solicitud de Modificaciones | | | |
|---|---|---|-------------------------------------|
| Sección I | | | |
| Solicitante: Norvin Gutiérrez. | Fecha de recepción: 04 / 04 / 2022 | Sistema: Jamen's Pollos | Creador: Norvin, Axel, Saúl. |
| Descripción del problema: El sistema no mandaba notificación que mostrará que sin uso de internet no funcionaba. | | Prioridad: 1- Podría hacer perder el trabajo correspondiente del día en caso de no conocer sobre esto. | |
| Sección II | | | |
| Numero de MR: 1 | | Mantenedor: Norvin Gutiérrez. | |
| Prioridad: 1 | | Tipo de mantenimiento: Mantenimiento Correctivo. | |
| Sección III | | | |
| Id Opción. 1 | Estado del MR: Finalizado. | | Fecha: 11 / 04 / 2020 |
| Resultado de Análisis: El sistema no funcionaba por falta de conexión a internet por ser cliente – servidor (el Login no daba acceso al sistema en general) y de igual manera no mostraba alguna notificación que lo indicara. | | | |
| Aprobado por: Norvin Gutiérrez. | | | |



Solicitud de Modificaciones

Sección I

| | | | |
|---|---|---|-------------------------------------|
| Solicitante: Norvin Gutiérrez. | Fecha de recepción: 04 / 04 / 2022 | Sistema: Jamen's Pollos | Creador: Norvin, Axel, Saúl. |
| Descripción del problema: En la categoría de usuarios, justamente en la opción de "Agregar Usuarios" permitía que en sus datos se escribieran caracteres fuera de lugar. | | Prioridad: 1- Mal coordinación al momento de recolectar los datos del usuario. | |

Sección II

| | |
|------------------------|---|
| Numero de MR: 2 | Mantenedor: Norvin Gutiérrez. |
| Prioridad: 1 | Tipo de mantenimiento: Mantenimiento Correctivo. |

Sección III

| | | |
|---------------------|-----------------------------------|------------------------------|
| Id Opción. 1 | Estado del MR: Finalizado. | Fecha: 11 / 04 / 2020 |
|---------------------|-----------------------------------|------------------------------|

Resultado de Análisis: El sistema permitía que dentro de los espacios correspondientes para ingresar datos personales del usuario aceptara (@, ..., /, etc), lo cual era a causa de una mala validación en la base de datos.

Aprobado por: Norvin Gutiérrez.



Ana

Lopez

Trabajador

Jamens Pollos Plaza inter

AnaLB

•••••

Guardar

Se ha insertado el registro de manera exitosa

Aceptar

Solicitud de Modificaciones

Sección I

| | | | |
|--|---|---|-------------------------------------|
| Solicitante: Norvin Gutiérrez. | Fecha de recepción: 04 / 04 / 2022 | Sistema: Jamen's Pollos | Creador: Norvin, Axel, Saúl. |
| Descripción del problema: En la categoría de Clientes, el problema presente data en las validaciones de las cajas de textos y previamente permite exceder los datos correspondientes. | | Prioridad: 2- Se pueden dar falsificaciones de datos, mala coordinación y coherencia de ellos. | |

Sección II

| | |
|------------------------|---|
| Numero de MR: 3 | Mantenedor: Norvin Gutiérrez. |
| Prioridad: 1 | Tipo de mantenimiento: Mantenimiento Correctivo. |

Sección III

| | | |
|---------------------|-----------------------------------|------------------------------|
| Id Opción. 2 | Estado del MR: Finalizado. | Fecha: 11 / 04 / 2020 |
|---------------------|-----------------------------------|------------------------------|

Resultado de Análisis: El sistema permitía que dentro de los espacios correspondientes para ingresar datos personales del proveedor aceptara (@, ..., /, etc), lo cual era a causa de una mala validación en la base de datos y no obstante se podía exceder los datos a ingresar.

Aprobado por: Norvin Gutiérrez.

Agregar Cliente

Nombre

Apellido

Telefono

Dirección

Cedula

Fecha Nacimiento

Cientes

| Id | Nombre | Apellido | Dirección | Telefono | Cedula | Fecha de |
|----|--------|-----------|--------------------|----------|------------------|-----------|
| 32 | Norvin | Gutierrez | De la bomba el ... | 58678964 | 161-160601-10... | 16/6/2001 |

Cliente Agregado

Aceptar

| Clientes | | | | | | |
|----------|---------|-----------|---------------------|----------|-------------------|-----------|
| Id | Nombre | Apellido | Dirección | Telefono | Cedula | Fecha de |
| 32 | Norvin | Gutierrez | De la bomba el ... | 58678964 | 161-160601-100... | 16/6/2001 |
| 33 | Ernesto | Potosme | De la Gsolinera ... | 82226571 | 441-180599-100... | 18/5/1999 |

Registro del Historial del MR

| ID MR | Fecha de recepción | Fecha de análisis | Estado del MR | Documentos Afectados |
|-------|--------------------|-------------------|---------------|----------------------|
| 1 | 04 / 04 / 2022 | 11 / 04 / 2022 | Finalizado. | ----- |
| 2 | 04 / 04 / 2022 | 11 / 04 / 2022 | Finalizado. | ----- |
| 3 | 04 / 04 / 2022 | 11 / 04 / 2022 | Finalizado. | ----- |

Registro de Pruebas del MR

| | | |
|---|-----------------|--|
| Id prueba: P1 | Id MR: 1 | Nombre del sistema: Jamen´s Pollos. |
| Responsable de Prueba. Norvin Gutiérrez. | | |
| Objetivo: Verificar si los cambios aplicados al código correspondiente al Login fueron de utilidad para que de esta manera haya podido solucionar el problema presente. | | |
| Resultado Obtenido: Luego de haber realizado el respectivo testeo a esta parte del sistema y de igual manera sus pruebas se llegaron al objetivo planteado para que de este modo se diera solución al problema presente; el cual trataba de que el sistema no accedía a su interior si no tenía una conexión a internet. | | |

Registro de Pruebas del MR

| | | |
|--|-----------------|--|
| Id prueba: P2 | Id MR: 2 | Nombre del sistema: Jamen´s Pollos. |
| Responsable de Prueba. Norvin Gutiérrez. | | |
| Objetivo: Determinar que, si los cambios y validaciones realizadas en la Base de datos hayan solucionado el problema presente, con respecto a agregar a un usuario nuevo. | | |
| Resultado Obtenido: Luego de haber realizado el respectivo testeo a esta parte del sistema (Usuarios) se logró concretar lo esperado, que consistía en que al momento de ingresar un usuario nuevo estos no dejarán pasar caracteres no adecuados para sus datos (@. -*/) | | |

Registro de Pruebas del MR

| | | |
|--|-----------------|--|
| Id prueba: P3 | Id MR: 3 | Nombre del sistema: Jamen´s Pollos. |
| Responsable de Prueba. Norvin Gutiérrez. | | |
| Objetivo: Verificar si los cambios aplicados al código correspondiente, junto a sus respectivas validaciones dentro de las respectivas consultas de la basa de datos hayan dado solución al problema que se presentaba al momento de digitar los datos del cliente. | | |
| Resultado Obtenido: Luego de haber realizado el respectivo testeo a esta parte del sistema y de igual manera sus pruebas se llegaron a obtener el resultado que se esperaba. Ya el sistema no permite caracteres inadecuados para los datos del cliente, de igual manera la parte de cédula ya valida correctamente y no se da lo que era en exceso de datos. | | |

