

Política de Mantenimiento SGAR

Billy Alvear, Helen Bernal, Alexander Castro y Renán Pérez

LOS RÁPIDOS S.A.

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN	2
1.1. PROPÓSITO.....	2
1.2. OBJETIVOS.....	2
1.3. LIMITACIONES	3
2. SEGUIMIENTO AL PROCESO DE MANTENIMIENTO	4
2.1. NECESIDADES DE MANTENIMIENTO	4
2.2. GESTIÓN DE LA CONFIGURACIÓN.....	4
2.3. ASEGURAMIENTO DE LA CALIDAD	6
3. DESCRIPCIÓN DE LOS TIPOS DE MANTENIMIENTO DE SOFTWARE IMPLEMENTADOS.....	7
3.1. MANTENIMIENTO CORRECTIVO:	7
3.2. MANTENIMIENTO PREVENTIVO:.....	7
3.3. MANTENIMIENTO PERFECTIVO:	8
4. APLICACIÓN DE LA GUÍA DE PROCEDIMIENTOS DE MANTENIMIENTO DE SOFTWARE.	9
CONCLUSIONES Y RECOMENDACIONES	13
4.1. CONCLUSIONES.....	13
4.2. RECOMENDACIONES	13
5. APROBACIÓN	14

1. Introducción

1.1. Propósito

El propósito de esta documentación es garantizar que los requisitos y especificaciones funcionales descritas en el Proyecto (Sistema para la gestión de ambulancias los rápidos S.A.), se encuentren desarrolladas y cumplan con la funcionalidad, estándares de tiempo y de calidad esperados, minimizando la aparición de errores en la realización de pruebas unitarias, con la finalidad de que el sistema tenga una buena aceptación del usuario funcional. Para poder obtener los resultados esperados el plan de desarrollo y mantenimiento del sistema deben ejecutarse de manera simultánea, debido a que ambos planes otorgan la información necesaria para la gestión y documentación del proyecto. En el mantenimiento del software el hecho de no contar con esa información puede reducir la importancia de este, por lo cual el mantenimiento del sistema puede asociarse únicamente con la corrección de errores en el software.

1.2. Objetivos

El objetivo principal es diseñar y desarrollar un plan de gestión de mantenimiento de software para el Sistema de gestión de ambulancias los rápidos S.A. Para efectuar las pruebas vamos a verificar el comportamiento del programa sobre un determinado conjunto de mantenimientos.

Entre los principales objetivos tenemos:

- Identificar los tipos de mantenimiento de software disponibles para nuestro sistema.
- Diseñar un plan de mantenimiento para el sistema de gestión de ambulancias los rápidos S.A.

- Investigar y evaluar herramientas informáticas que ayuden a automatizar el mantenimiento del sistema.

1.3. Limitaciones

Los tipos de mantenimiento que cubre el plan de gestión de mantenimiento son: correctivo, preventivo y perfectivo.

- No hay respaldo de datos en la nube, los registros se almacenan de manera local
- No se contempla el desarrollo de los tipos de mantenimiento predictivos.

2. Seguimiento al proceso de Mantenimiento

2.1. Necesidades de Mantenimiento

El plan de mantenimiento debe realizarse en cualquier organización, en este caso en el software “Los Rápidos S.A”. Como consecuencia de esto, se debe realizar la gestión de información que debe ser documentada, así como valorar su importancia, y documentar la corrección de errores en el programa a través del plan de pruebas.

El mantenimiento de un sistema es parte fundamental del proceso de desarrollo de un software, por lo cual, es vital realizarlo. Por este motivo, se ve la necesidad de desarrollar un plan de gestión de mantenimiento para “Los Rápidos S.A.” que servirá como una guía de mantenimiento con la corrección de errores del sistema.

2.2. Gestión de la Configuración

Comprende un conjunto de actividades desarrolladas para gestionar los cambios a lo largo del ciclo de vida identificando, organizando y controlando las modificaciones del software.

Entre las actividades están:

- **Identificación de elementos**

La actividad de identificación de la configuración identifica los elementos que van a ser controlados, establece esquemas para la identificación de los elementos y sus versiones, y establece las herramientas y técnicas a usar para adquirir y gestionar los elementos controlados. Estas actividades proporcionan la base para otras actividades de gestión de configuración.

- **Control de versiones**

El esquema sugerido para la identificación de las distintas versiones del producto es un esquema numérico conformado de la siguiente manera:

<major_version> . <minor_version> . <patch_number>

Major Version: indica un cambio de funcionalidad sustancial a otras versiones del sistema

Minor Version: indica que el sistema es funcionalmente idéntico, pero es distinto desde el punto de vista no-funcional a otras versiones

Patch Version: indica el número de un parche aplicado al sistema, que significa la corrección de un defecto encontrado

- **Control de cambios**

Como primer paso para gestionar los cambios sobre los elementos controlados es determinar los cambios que se van a realizar. El proceso de petición de cambios proporciona procedimientos formales para enviar y registrar peticiones de cambio, evaluar el coste e impacto potencial del cambio propuesto, y aceptar, modificar, o rechazar el cambio propuesto.

- **Auditar la configuración**

Una auditoría de software es una actividad llevada a cabo para evaluar, de forma independiente y objetiva, la conformidad de los productos y procesos software con respecto a las regulaciones, estándares, guías, planes y procedimientos aplicables.

- **Generación de informes**

El informe del estado de la configuración reporta la información necesaria para gestionar de forma efectiva la configuración de software. En esta actividad se deberá diseñar y operar un sistema para la captura y reporte de la información necesaria a medida que avanza el proceso de desarrollo.

La finalidad de la Gestión y configuración del Software es el conocer la estructura de procesos y herramientas para aplicar dentro de la construcción del software que nos ayudan a controlar los cambios.

Es importante considerar ciertas modificaciones que pueden ocurrirle al software dentro de todo el proceso de ingeniería para asegurar su control y calidad.

2.3. Aseguramiento de la Calidad

El aseguramiento de la calidad del software debe tenerse en cuenta al realizar cambios en un sistema existente. Durante el proceso de mantenimiento, debe realizar los siguientes tipos y niveles de control de calidad. Por ejemplo, inspecciones, revisiones, auditorías, verificaciones, etc. El nivel y formato de la revisión se especifica en el plan de mantenimiento.

Una parte importante a tener en cuenta es asegurarse de que la documentación original del sistema continúe describiendo el producto actual. Lo bueno de crear este plan de mantenimiento es que puede obtener la documentación que necesita “Los Rápidos S.A.” porque es necesario documentar el proceso de mantenimiento detallado para que todos los trabajadores de mantenimiento sigan el mismo proceso.

3. Descripción de los tipos de Mantenimiento de Software implementados.

El tipo de mantenimiento está basado en función de la actividad y que se pretende conseguir con aquella acción. Según la norma ISO/IEC 14764 existen varias categorías para clasificar las actividades de mantenimiento por lo que acorde al progreso del proyecto se vio necesario hacer uso de los siguientes: Mantenimiento correctivo y preventivo.

3.1. Mantenimiento Correctivo:

El propósito de este tipo de mantenimiento es corregir los errores que van surgiendo a lo largo del desarrollo, los cuales producen errores de procesamiento y fallas de rendimiento, también porque el software no es capaz de cumplir con los requerimientos iniciales, ya que suelen haber muchos cambios durante el desarrollo, ya sea por el tipo de tecnología a usar o por solicitudes del cliente.

En nuestro proyecto aplicamos las siguientes actividades de mantenimiento.

- Validaciones para el registro y modificación de datos almacenados en el sistema, entre los que se contemplan, los usuarios, clientes y ambulancias.
- Módulos de validación para evitar la redundancia e inconsistencia de datos.
- Pruebas unitarias para la detección de errores en ciertos módulos.
- Actualizaciones periódicas haciendo uso del sistema de control de versiones, Github.

3.2. Mantenimiento Preventivo:

El propósito del mantenimiento preventivo es evitar que ocurran problemas al modificar los productos de software. Se trata de analizar e identificar los riesgos futuros y problemas desconocidos, y tomar medidas para que no se produzcan estos problemas.

A continuación, presentamos el listado de las actividades realizadas hasta el momento.

- Uso de TryCatch para capturar los posibles errores que puedan ocurrir durante la ejecución del programa entre ellos se contempla el acceso a la base de datos para las consultas o registro de datos.
- Modularizar las clases para agregar futuras funcionalidades al sistema.
- Realizar copias de seguridad para los datos.

3.3. Mantenimiento Perfectivo:

El propósito de mantenimiento perfectivo es hacer una variedad de mejoras, es decir, la experiencia del usuario, la eficiencia de procesamiento, y la mantenibilidad.

Las actividades

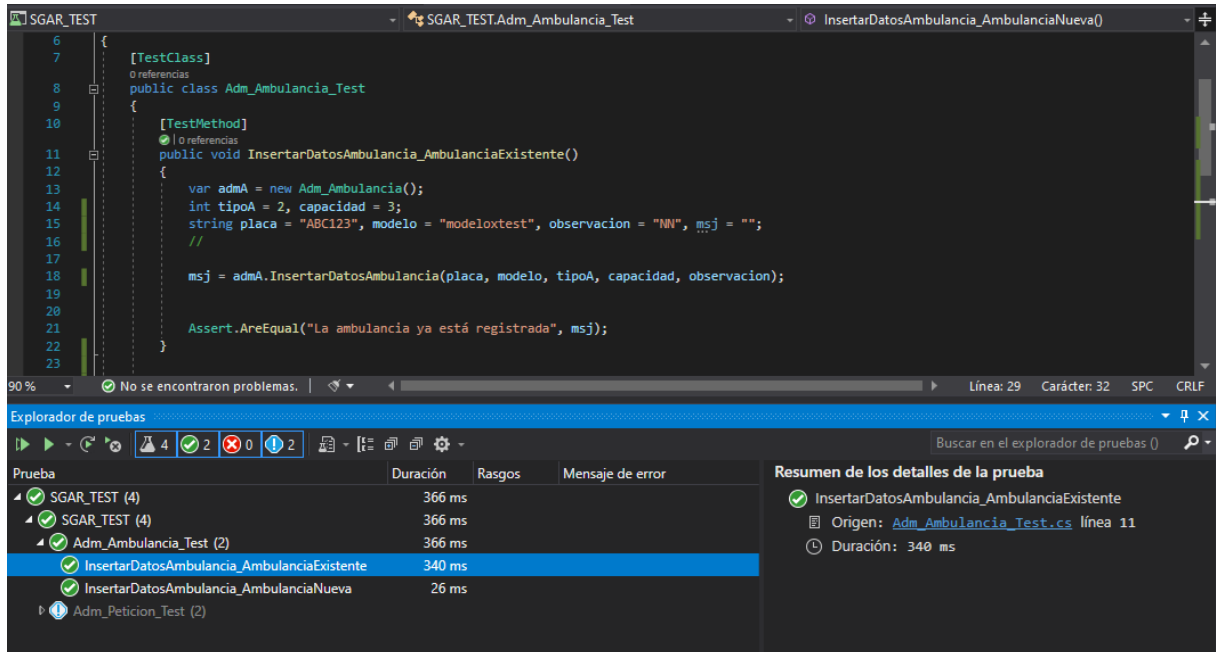
- Refactorización del código para hacerlo más legible para los desarrolladores y futuro personal de mantenimiento.
- Uso de comentarios para la descripción de métodos.
- Rediseño de los diagramas de clase, casos de uso y paquetes.
- Actualización de la documentación.
- Optimización del flujo de las interfaces gráficas del usuario para la ejecución de sus tareas.
- Uso de notificaciones emergentes para mostrar el éxito o fracaso de alguna de los procesos de los que se encarga el software.
- Corrección el modelo arquitectónico de diseño (MVC).

4. Aplicación de la guía de procedimientos de Mantenimiento de Software.

Esta guía muestra un historial específico de ciertos mantenimientos correctivos, con el fin de brindar un mejor servicio a los clientes de Los Rápidos S.A, y que puedan administrar de forma eficaz y eficiente los datos de sus ambulancias y el de sus clientes.

A continuación, se mostrarán los mantenimientos correctivos realizados por medio de las pruebas unitarias, que nos permitió encontrar ciertas fallas en el sistema aplicando sus respectivas soluciones.

Fecha de recepción	18/03/2022	Id. Solicitud	MR-01
Prioridad	Media	Creador	Renán Pérez
Nombre del proyecto	Gestión de ambulancias: Los rápidos S.A.		
Tipo de mantenimiento	Mantenimiento Correctivo		
Mantenedor	Helen Bernal		
Descripción de la solicitud	Realizar las pruebas unitarias del ingreso de las ambulancias, ya que con estas pruebas se puede verificar si el código funciona correctamente. Tener en cuenta haber ingresado correctamente la credencial al sistema.		
RESULTADO DE LA SOLICITUD			
Estado de la solicitud	Realizada		
Autorización	Revisado por	Aprobado por Grupo 2 Responsables del Mantenimiento del Software	



```

6  {
7  [TestClass]
8  public class Adm_Ambulancia_Test
9  {
10     [TestMethod]
11     public void InsertarDatosAmbulancia_AmbulanciaExistente()
12     {
13         var admA = new Adm_Ambulancia();
14         int tipoA = 2, capacidad = 3;
15         string placa = "ABC123", modelo = "modeloxtest", observacion = "NN", msj = "";
16         //
17
18         msj = admA.InsertarDatosAmbulancia(placa, modelo, tipoA, capacidad, observacion);
19
20         Assert.AreEqual("La ambulancia ya está registrada", msj);
21     }
22 }
23
    
```

Explorador de pruebas

Prueba	Duración	Rasgos	Mensaje de error
SGAR_TEST (4)	366 ms		
SGAR_TEST (4)	366 ms		
Adm_Ambulancia_Test (2)	366 ms		
InsertarDatosAmbulancia_AmbulanciaExistente	340 ms		
InsertarDatosAmbulancia_AmbulanciaNueva	26 ms		
Adm_Peticion_Test (2)			

Resumen de los detalles de la prueba

- InsertarDatosAmbulancia_AmbulanciaExistente
 - Origen: *Adm_Ambulancia_Test.cs* línea 11
 - Duración: 340 ms

Fecha de recepción	18/03/2022	Id. Solicitud	MR-02
Prioridad	Media	Creador	Alexander Castro
Nombre del proyecto	Gestión de ambulancias: Los rápidos S.A.		
Tipo de mantenimiento	Mantenimiento Preventivo		
Mantenedor	Helen Bernal		
Descripción de la solicitud	Realizar las pruebas unitarias del ingreso al sistema mediante el login, ya que con estas pruebas se puede verificar si es posible ingresar un usuario no existente y cómo reacciona el sistema.		
RESULTADO DE LA SOLICITUD			
Estado de la solicitud	Realizada		
Autorización	Revisado por	Aprobado por Grupo 2 Responsables del Mantenimiento del Software	

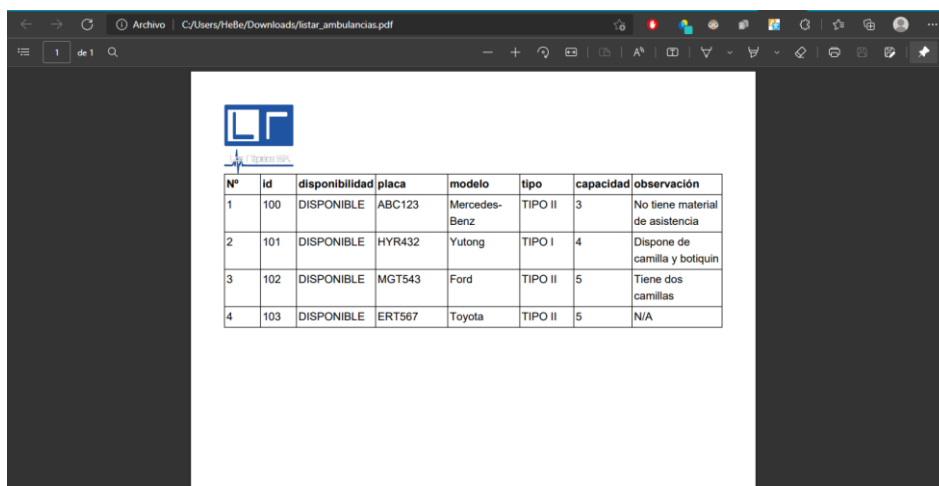
The screenshot shows the Visual Studio IDE with a C# test file. The test method `Inicio_de_sesion_Exitoso()` is highlighted. The test passes, as indicated by the green checkmark in the Test Explorer. The Test Explorer shows a list of tests, with `Inicio_de_sesion_Exitoso` selected, showing a duration of 837 ms.

The screenshot shows the Visual Studio IDE with a C# test file. The test method `Inicio_de_sesion_Fallido()` is highlighted. The test fails, as indicated by the red 'X' in the Test Explorer. The Test Explorer shows a list of tests, with `Inicio_de_sesion_Fallido` selected, showing a duration of 772 ms. The error message is visible in the Test Explorer details pane.

Fecha de recepción	18/03/2022	Id. Solicitud	MR-03
Prioridad	Media	Creador	Billy Alvear
Nombre del proyecto	Gestión de ambulancias: Los rápidos S.A.		
Tipo de mantenimiento	Mantenimiento Perfectivo		
Mantenedor	Helen Bernal		
Descripción de la solicitud	Realizar las pruebas de aceptación para demostrar el comportamiento frente a errores del usuario en la interfaz al momento de imprimir un reporte.		
RESULTADO DE LA SOLICITUD			
Estado de la solicitud	Realizada		

Autorización	Revisado por	Aprobado por Grupo 2 Responsables del Mantenimiento del Software
---------------------	---------------------	--

Si existen datos de ambulancias para imprimir:



Nº	Id	disponibilidad	placa	modelo	tipo	capacidad	observación
1	100	DISPONIBLE	ABC123	Mercedes-Benz	TIPO II	3	No tiene material de asistencia
2	101	DISPONIBLE	HYR432	Yutong	TIPO I	4	Dispone de camilla y botiquín
3	102	DISPONIBLE	MGT543	Ford	TIPO II	5	Tiene dos camillas
4	103	DISPONIBLE	ERT567	Toyota	TIPO II	5	N/A

Si no existen datos de ambulancias para imprimir:



The screenshot shows the 'Consulta de Ambulancias' application. On the left is a sidebar with user information (Alexander Castro, Gerente) and navigation options (Secretaría, Cliente, Ambulancia, Registrar, Consultar, Modificar, Conductor). The main area has search filters (Dato: 0, Placa, Modelo, Disponibles, Tipo) and buttons for 'Consultar', 'Mostrar Todos', 'Eliminar', 'Modificar', and 'Imprimir'. A modal dialog box is displayed in the center with the message 'No hay datos para imprimir' and an 'Aceptar' button. The bottom status bar shows the time '20:53:04' and the date 'martes, 22 de marzo de 2022'.

Conclusiones y Recomendaciones

4.1. Conclusiones

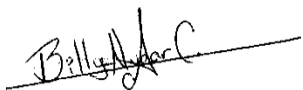

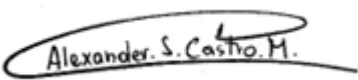

- En conclusión, el mantenimiento de software es necesario para preservar el valor del software a lo largo del tiempo y prevenir problemas en un futuro, es decir que, se realizarán diferentes acciones o cambios que garanticen que el sistema funcione de manera correcta sin importar si en la actualidad funcione bien.
- En virtud de lo elaborado, ahora sabemos que el mantenimiento de software es una actividad amplia y compleja pero importante debido a que esta incluye la corrección de errores, mejoras, eliminación de funciones obsoletas y optimización al sistema.

4.2. Recomendaciones

- Se recomienda llevar un control de los cambios realizados sobre el sistema con la finalidad de tener un registro y que en las futuras acciones que se implementen no causen inconvenientes.
- Se recomienda establecer un método/medio que permita analizar los errores obtenidos y tener conocimientos previos sobre los problemas más usuales que ocurren al dar mantenimiento a un sistema software.
- Se recomienda dialogar con todos los usuarios del sistema (empleados de la empresa) que notifiquen errores o posibles funciones obsoletas al momento de ser detectados para brindar mantenimiento en el menor tiempo posible y que el sistema continúe funcionando de manera correcta.

5. Aprobación

22/03/2022

Elaborado por:	
 <hr/>	 <hr/>
Billy Alvear	Helen Bernal
 <hr/>	 <hr/>
Alexander Castro	Renán Pérez