**ADMINISTRACCION DE AMBULANCIAS**

**AUTOR:**

CAICEDO BARZOLA KENNY RICARDO

Contenido

[**RESUMEN** 2](#_Toc129662785)

[**ABSTRACT** 4](#_Toc129662786)

[**INTRODUCCIÓN** 5](#_Toc129662787)

[**HERRAMIENTAS IMPLEMENTADAS** 6](#_Toc129662788)

[**METODOLOGÍA DE DESARROLLO** 9](#_Toc129662789)

[**INCREMENTAL** 9](#_Toc129662790)

[**ESTIMACIÓN COCOMO** 10](#_Toc129662791)

[**MODELO BÁSICO.** 10](#_Toc129662792)

[**MODELO INTERMEDIO.** 12](#_Toc129662793)

[**MODELO DE DISEÑO INICIAL** 14](#_Toc129662794)

[**MODELO POST-ARQUITECTURA** 15](#_Toc129662795)

[**RESULTADOS** 17](#_Toc129662796)

[**CONCLUSIONES** 18](#_Toc129662797)

# **RESUMEN**

El objetivo de este proyecto es desarrollar e implementar un sistema de gestión para la administración de ambulancias, que permita realizar de manera eficiente las actividades administrativas y de coordinación de las mismas. Para ello, se utilizará una metodología especializada en el desarrollo de software y se empleará el lenguaje de modelamiento UML para el diseño y programación del sistema. Este sistema deberá ser capaz de gestionar la asignación de ambulancias a diferentes servicios y rutas, llevar un control de los pacientes atendidos, así como de los gastos e ingresos generados por cada servicio.

Las palabras clave son: sistema de gestión, ambulancias, metodología de software, UML, asignación, pacientes, gastos e ingresos.

# **ABSTRACT**

The goal of this project is to develop and implement a management system for ambulance administration that allows for efficient coordination and administrative activities. To accomplish this, a specialized software development methodology will be used, and the UML modeling language will be employed for system design and programming. This system should be capable of managing ambulance assignment to different services and routes, tracking patients treated, as well as expenses and revenues generated by each service.

Keywords include: management system, ambulances, software methodology, UML, assignment, patients, expenses, and revenues.

# **INTRODUCCIÓN**

La siguiente documentación ofrecerá un resumen general sobre la creación, desarrollo y progreso del proyecto planteado con la temática de "Gestión de Ambulancias", utilizando los recursos de arquitectura necesarios para describir los diferentes aspectos del sistema. El objetivo es respaldar las decisiones más significativas tomadas en cuenta para el proyecto, además de cumplir con los requisitos funcionales y no funcionales del sistema. También se detallará la creación del código y todo lo necesario para su creación desde cero, incluyendo las herramientas utilizadas y sus descripciones, así como las funciones de cada parte involucrada.

Es bien sabido que la construcción de software se enfoca en varios aspectos, por lo que es fundamental que sea estable para permitir que los requisitos del cliente se cumplan de manera correcta. Además, se debe validar la arquitectura que se utilizará para asegurarse de que sea adecuada y contribuya de manera positiva al proceso para descartar posibles errores que puedan ocurrir durante la construcción del software, lo que brinda confianza en la eficiencia presentada y posible disminución en los costos. Todas las funciones implementadas en el proyecto deben ser probadas e integradas para mantener la estabilidad mientras se avanza en el proceso.

En nuestro proyecto, nos enfocaremos y pondremos en práctica lo previamente mencionado para obtener buenos resultados. Analizaremos las funciones que intervienen en la gestión de ambulancias y los módulos involucrados, que incluyen el Módulo de Inicio, Módulo de Perfil, Módulo de Registro, Módulo de Consultas y Módulo de Reportes, así como los procesos administrativos y contables que involucran a los propietarios de las ambulancias, los presidentes de comunidad, las entidades bancarias, las secretarias y los directores.

Finalmente, reportaremos todo lo que suceda a lo largo de la construcción de nuestro software "Gestión de Ambulancias" y se respaldará en el presente documento a continuación.

# **HERRAMIENTAS IMPLEMENTADAS**

**VISUAL STUDIO 2022**

El sistema se va a desarrollar utilizando el IDE Visual Studio 2022 ya que nos permite una amplia gama de herramientas y funcionalidades para el desarrollo de aplicaciones. Esta herramienta facilita el proceso de diseño de aplicaciones tanto de escritorio como web, y proporciona características avanzadas para la depuración, prueba y despliegue de aplicaciones.

Visual Studio 2022 ofrece una gran cantidad de beneficios, entre ellos:

* Interfaz de usuario intuitiva y fácil de usar.
* Herramientas de depuración y prueba avanzadas.
* Capacidad para desarrollar en múltiples lenguajes de programación como C++, C#, Visual Basic, F#, Python y otros.
* Capacidad para trabajar en aplicaciones web, de escritorio, móviles y de la nube.
* Posibilidad de integrar con múltiples herramientas de desarrollo y terceros.
* Funciones avanzadas de control de código fuente.
* Posibilidad de personalizar y extender la funcionalidad a través de extensiones y complementos.

Visual Studio 2022 también incluye una variedad de características adicionales, como IntelliSense, edición de código enriquecida, integración con Azure, soporte de Git y mucho más. Estas características hacen que el desarrollo de aplicaciones sea más fácil y rápido.

**.NET FRAMEWORK**

Para desarrollar aplicaciones en Visual Studio 2022, se utiliza el .NET Framework. Este es un conjunto de bibliotecas de clases y servicios que se ejecutan en la plataforma Microsoft Windows. Proporciona una arquitectura común para el desarrollo de aplicaciones y permite a los desarrolladores crear aplicaciones más rápidamente, con menos errores y más eficientes.

El .NET Framework también incluye el Common Language Runtime (CLR), que proporciona un entorno de ejecución para el código administrado. El código administrado se compila en un lenguaje intermedio común que se ejecuta en el CLR.

**SQL SERVER**

Una base de datos es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos. Cada base de datos ha sido diseñada para satisfacer los requisitos de información de una empresa u otro tipo de organización. Una base de datos se puede percibir como un gran almacén de datos que se define y se crea una sola vez, y que se utiliza al mismo tiempo por distintos usuarios. En una base de datos todos los datos que se integran con una mínima cantidad de duplicidad. (Marqués 2020)

SQL Server es un sistema de gestión de base de datos relacional desarrollado por Microsoft. Utiliza el lenguaje de programación Structured Query Language (SQL) para realizar consultas y manipulaciones de los datos almacenados. SQL Server es utilizado para almacenar, consultar, y gestionar grandes cantidades de datos para aplicaciones empresariales y de uso personal. (Microsoft, s.f.)

La base de datos se utilizará en el presente proyecto para poder almacenar la información de cada una de las entidades ya antes detalladas. Esto se realizará a través de un servidor local utilizando SQL Server Management Studio, que permitirá la interacción con la base de datos a través de consultas y procedimientos almacenados.

**SQL SERVER MANAGEMENT STUDIO**

SQL Server Management Studio (SSMS) es una herramienta de administración de base de datos y entorno integrado de desarrollo para SQL Server. Permite la creación y administración de bases de datos, tablas, vistas, procedimientos almacenados y otros objetos de base de datos, así como la ejecución de consultas y la depuración de procedimientos almacenados. SSMS también incluye herramientas para administrar la seguridad, programar tareas y realizar copias de seguridad y restauración de bases de datos. (Microsoft, s.f.)

**¿Para qué sirve SQL Server Management Studio?**

SSMS es una herramienta esencial para administrar y gestionar bases de datos de SQL Server, ya que permite realizar tareas de administración, programación y depuración de forma visual y a través de consultas SQL. También es útil para la creación y mantenimiento de objetos de base de datos, como tablas, vistas y procedimientos almacenados, así como para la gestión de la seguridad y las copias de seguridad de la base de datos. (TechTarget, s.f.)

**Funciones de SQL Server Management Studio:**

Modelado de datos SQL Server Management Studio permite a los desarrolladores, arquitectos de datos y demás clientes diseñar, modelar, gestionar y generar bases de datos de manera visual o gráfica, incluyendo todos los elementos necesarios para realizar modelos con un alto nivel de complejidad. Esta plataforma permite, además, obtener información o diseño a partir de un producto, con el objetivo de establecer cuáles son sus componentes, cómo interactúan entre sí y cómo fueron creados. Este proceso se conoce como ingeniería inversa o retro-ingeniería, y SQL Server Management Studio permite llevarlo a cabo entre una base de datos que esté activa y un esquema.

Editor de tablas Este entorno gráfico cuenta también con la función de editor de tablas, que permite la modificación de todos los aspectos de la base de datos. Esto ofrece facilidades de uso para el proceso de configuración de las tablas, índices, columnas, opciones, permisos y particiones, entre otros elementos.

Migración de bases de datos Esta herramienta de SQL Server ofrece una solución completa y de fácil uso enfocada en migrar de MySQL, Oracle, PostgreSQL y otras tablas, datos y objetos de un sistema de gestión de bases de datos relacionales (RDBMS) a SQL Server. Esto permite que los usuarios tengan la posibilidad de convertir de manera rápida y sencilla las aplicaciones para que puedan ser ejecutaras en SQL Server tanto en Windows como en plataformas bajo el sistema operativo Linux o Mac OS.

Panel de rendimiento visual Otra de las funciones de SQL Server Management Studio es que le ofrece a sus usuarios una serie de elementos destinados a aumentar el rendimiento de las aplicaciones. De manera que, con el uso de esta herramienta gráfica, los clientes pueden observar de forma rápida todo lo relacionado con los indicadores clave de productividad, a través del llamado panel de rendimiento visual. Esta función proporciona un conjunto de informes que ofrecen una fácil identificación y opciones para que el usuario vea dónde optimizar su consulta.

Desarrollo de SQL SQL Server Management Studio se encarga también de la creación y gestión de conexiones a servidores de bases de datos, además del proceso de configuración de parámetros de conexión y la ejecución de consultas SQL en las conexiones de la base de datos, a través del uso del editor incorporado. Este editor también ofrece un resaltado de sintaxis en color, la reutilización de fragmentos de SQL, el historial de ejecución y la función de autocompletado.

Gestión del servidor La plataforma SQL Server Management Studio proporciona una consola visual que permite la gestión de manera fácil de los entornos de SQL Server. Esto permite que el usuario obtenga una mejor visibilidad de sus bases de datos y que, además, pueda hacer uso de herramientas visuales que contribuyan en la configuración de servidores, administración de usuarios, realización de backups o copias de seguridad y recuperación, al tiempo que inspecciona los datos de auditoria y ve el estado actual de la base de datos, monitoreando su rendimiento.

# **METODOLOGÍA DE DESARROLLO**

**INCREMENTAL**

La idea básica es desarrollar el sistema siguiendo etapas incrementales caracterizadas por generación de sucesivas versiones que van abarcando requerimientos hasta completar el sistema. (González). La metodología cada versión se integra los avances del proyecto así se genera una versión de aceptación y sentido del cliente.

Estas etapas, consisten en requerimientos, diseño, codificación, pruebas y entrega. Permite entregar al cliente un producto más rápido en comparación del modelo en cascada.

Ventajas:

* La solución se va mejorando en forma progresiva a través de las múltiples iteraciones, incrementa el entendimiento del problema y de la solución por medio de los refinamientos sucesivos.
* Los clientes no esperan hasta el fin del desarrollo para utilizar el sistema. Pueden empezar a usarlo desde el primer incremento.
* Los clientes pueden aclarar los requisitos que no tengan claros, conforme ven las entregas del sistema.
* Se disminuye el riesgo de fracaso de todo el proyecto, ya que se puede distribuir en cada incremento.
* Las partes más importantes del sistema son entregadas primero, por lo cual se realizan más pruebas en estos módulos y se disminuye el riesgo de fallos.

Desventaja:

* Requiere de mucha planeación, tanto administrativa como técnica
* Requiere de metas claras para conocer el estado del proyecto.
* Es un proceso de desarrollo de software, creado en respuesta a las debilidades del modelo tradicional de cascada. (Maida, 2015, pp 45)

## **ESTIMACIÓN COCOMO**

**MODELO BÁSICO.**

**PROYECTO ORGÁNICO.**

**Salario:** 1200

**Líneas de código:** 6370 Mil (6,37 KLOC)

**ESFUERZO EN PERSONA-MES:**

**TIEMPO DE DESARROLLO:**

**PERSONAS NECESARIAS EN EL PROYECTO:**

**COSTO TOTAL DEL PROYECTO:**

**PRODUCTIVIDAD:**

**MODELO INTERMEDIO.**

**PROYECTO ORGÁNICO.**

Fórmula = **Effort = a \* (SIZE)b \* M**

**Proyecto: Administración de Fincas**

**Atributos para el cálculo del FAE**

|  |  |
| --- | --- |
| **Atributos** | **Valor** |
| **De software** | |
| Fiabilidad | 1.15 (alto) |
| Tamaño de base de datos | 1 (nominal) |
| Complejidad | 0,85 (baja) |
| **De hardware** | |
| Restricciones de tiempo de ejecución | 1,11 (alto) |
| Restricciones de memoria virtual | 1 (nominal) |
| Volatilidad de la máquina virtual | 0,87 (bajo) |
| Tiempo de respuesta | 1,15 (alto) |
| **De personal** | |
| Capacidad de análisis | 1 (nominal) |
| Experiencia en la aplicación | 1,13 (bajo) |
| Calidad de los programadores | 1 (nominal) |
| Experiencia en la máquina virtual | 1,1 (bajo) |
| Experiencia en el lenguaje | 1 (nominal) |
| **Del proyecto** | |
| Técnicas actualizadas de programación | 1,1 (bajo) |
| Utilización de herramientas de software | 0,91 (alto) |
| Restricciones de tiempo de desarrollo | 1,04 (alto) |

.

**SIZE con puntos de función.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Factores funcionales de peso** | **Factores de peso** | | | | |
| **Parámetros de media (1)** | | | **Contador**  **(2)** | **Total mult.**  **(1)\*(2)** |
| **Simple** | **Media** | **Compleja** |
| **N. entrada usuario** | 7 | 10 | 15 | 8 | 56 |
| **N. salida usuario** | 5 | 7 | 10 | 6 | 30 |
| **N. consultas usuario.** | 3 | 4 | 6 | 7 | 21 |
| **N. archivos Lógicos internos** | 4 | 5 | 7 | 14 | 56 |
| **N. interfaces externas** | 3 | 4 | 6 | 0 | 0 |
| **TOTAL** | | | | | 163 |

**PF = 163 \* [0.65 +( 0.01 \* 15.41)] = 131,0683**

**SIZE = 131,0683 \* 46 /1000 = 6,029**

**Conversión a LOC**

|  |  |
| --- | --- |
| **LENGUAJE** | **Correlación código fuente por PF (aprox)** |
| C# | 46 |

**LOC = PF \* correlación**

**KLOC = (131,0683 \* 46) / 1000**

**KLOC = 6 (miles de líneas de código)**

**Esfuerzo (E) = 3,2 \* (6)1.05 \* 1.404 = 29.48 meses/hombre**

**Duración(D) = 2.5 \* (29.48)0.38 = 9.04 meses**

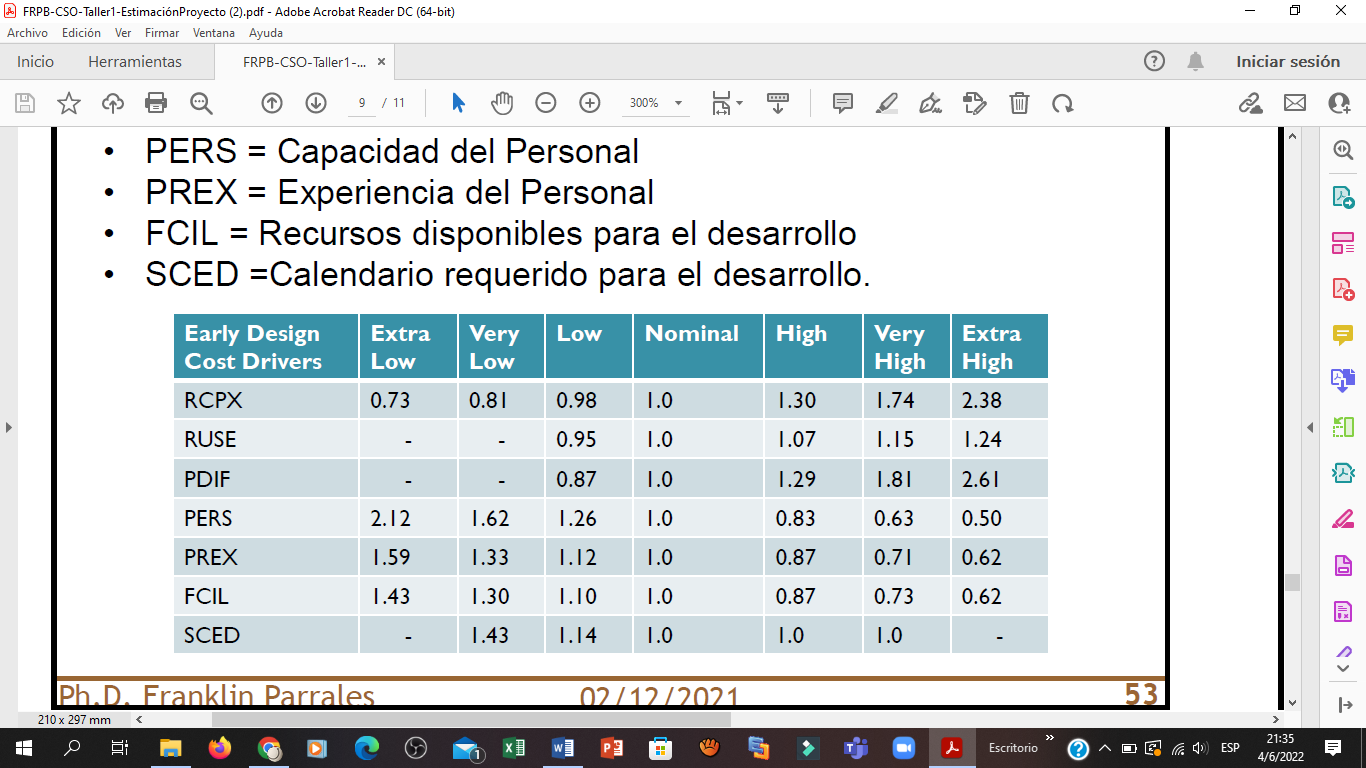
**Personal(P) 29.49/9.04 = 3.26 personas**

**Productividad(´P) = 6,029/29.39 = 205.137 LDC/persona mes**

### **MODELO DE DISEÑO INICIAL**

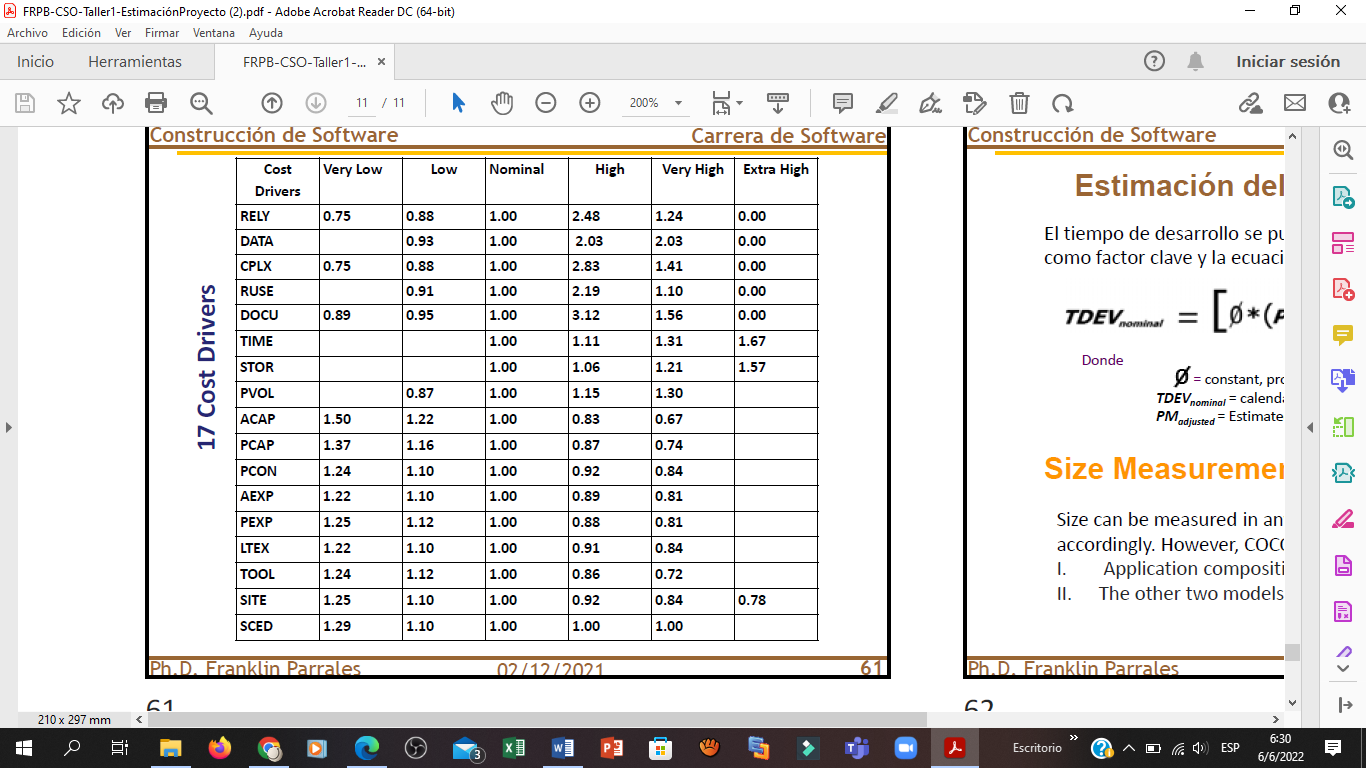
**Estimación de esfuerzo**

**Los 7 costos son:**



**MODELO POST-ARQUITECTURA**

**Personas - Mes**



**Los 17 costos son:**

)

**PERSONAS Y SUS ROLES**

|  |  |  |
| --- | --- | --- |
| **NOMBRES** | **CONTACTO** | **ROL** |
| Caicedo Barzola Kenny | Kenny.caicedob@ug.edu.ec | Desarrollador |

# **RESULTADOS**

El principal resultado que se genera con el proyecto es el diseño de un modelo de gestión para un sistema de ambulancias. En el cual se identifican claramente los procesos asociados y los módulos funcionales necesarios en el software para lograr rapidez y eficiencia en las operaciones, y a su vez facilitar la toma de decisiones en la estrategia de administración. Además, del desarrollo de un producto innovador, la propuesta de un modelo para la administración de ambulancias instanciado, controlado y ejecutado en una herramienta tecnológica. El sistema cuenta con tres módulos, que son el módulo de inicio, módulo de registro y el módulo de Perfil, los cuales en conjunto permiten administrar correctamente la información que se ingresa al sistema. En el sistema se pueden registrar conductores, médicos y ambulancias. Luego de ingresar al sistema, estos actores contarán con un perfil el cual pueden alimentar con toda la información necesaria sobre las ambulancias. El conductor podrá tener acceso a registrar el proceso de traslados, llegadas y salidas que estén asociados a las ambulancias, no obstante, el sistema creado no es autónomo ya que depende de entidades como el hospital para realizar y verificar sus traslados y pacientes. Por otra parte, el sistema también permitirá a los médicos definir su disponibilidad. Esta información se verá reflejada en el perfil de cada médico. La aplicación en sí servirá de intermediario y respaldo de todos estos trámites para facilitar la gestión de todas las ambulancias. Gracias a la elección del lenguaje de desarrollo (Java) se logró una independencia de plataforma que permite al sistema ser ejecutado desde cualquier computadora que cuente con un sistema operativo (Windows, Linux, MacOs).

# **CONCLUSIONES**

* La documentación presentada demostró el diseño de un modelo de gestión para un sistema de gestión de ambulancias.
* Se han identificado los procesos asociados y los módulos funcionales necesarios en el software para lograr la productividad y eficiencia de las operaciones, y así facilitar la toma de decisiones en la estrategia de gestión de emergencias médicas.
* Se eligió un lenguaje de programación adecuado para el desarrollo del sistema de gestión de ambulancias, lo que permite una fácil integración y compatibilidad con otras herramientas y sistemas.
* Los modelos utilizados en el diseño del sistema de gestión de ambulancias han tenido en cuenta la complejidad de los factores que afectan la eficiencia y la productividad de los servicios de emergencia médica, lo que garantiza la calidad y la eficacia del sistema.
* Se han realizado pruebas de funcionalidad y se ha comprobado la efectividad del tiempo de respuesta a las solicitudes que se realizan en la aplicación, lo que garantiza la rapidez y eficacia en la atención de emergencias médicas.
* Se han cumplido con los requisitos planteados y se ha obtenido un resultado satisfactorio en los cálculos y en la gestión del sistema, lo que garantiza la eficacia y la calidad en la gestión de emergencias médicas.
* Finalmente, se ha generado un manual de usuario para los futuros usuarios administradores del sistema, lo que garantiza la fácil utilización y gestión del sistema de gestión de ambulancias.