INDICE

[Información General. 3](#_Toc12191990)

[Propósito. 3](#_Toc12191991)

[Alcance. 4](#_Toc12191992)

[Personal involucrado. 4](#_Toc12191993)

[Convenciones, definiciones, acrónimos 4](#_Toc12191994)

[Referencias 5](#_Toc12191995)

[Resumen 5](#_Toc12191996)

[Perspectiva del producto 6](#_Toc12191997)

[Funcionalidad del producto 6](#_Toc12191998)

[Características del producto 7](#_Toc12191999)

[Restricciones 8](#_Toc12192000)

[Información funcional y no funcional. 8](#_Toc12192001)

[Proceso Funcional. 8](#_Toc12192002)

[Requerimientos Funcionales generales 9](#_Toc12192003)

[Requisitos comunes de las interfaces 12](#_Toc12192004)

[Interfaces de usuario 12](#_Toc12192005)

[Interfaces de hardware 13](#_Toc12192006)

[Interfaces de software 13](#_Toc12192007)

[Interfaces de comunicación 13](#_Toc12192008)

[Requerimientos Funcionales específicos 13](#_Toc12192009)

[Requisito funcional 1 13](#_Toc12192010)

[Requisito funcional 2 13](#_Toc12192011)

[Requisito funcional 3 13](#_Toc12192012)

[Requisito funcional 4 14](#_Toc12192013)

[Requisito funcional 5 14](#_Toc12192014)

[Requisito funcional 6 14](#_Toc12192015)

[Requisito funcional 7 14](#_Toc12192016)

[Requerimientos No Funcionales 14](#_Toc12192017)

[Requisitos de rendimiento 14](#_Toc12192018)

[Seguridad 14](#_Toc12192019)

[Fiabilidad 15](#_Toc12192020)

[Disponibilidad 15](#_Toc12192021)

[Mantenibilidad 15](#_Toc12192022)

[Portabilidad 15](#_Toc12192023)

[Información técnica. 15](#_Toc12192024)

[Básico. 15](#_Toc12192025)

[Base Datos. 15](#_Toc12192026)

[Front End 16](#_Toc12192027)

[Vista “Splashscreen” 16](#_Toc12192028)

[Vista “Login” 16](#_Toc12192029)

[Vista “registro”. 17](#_Toc12192030)

[Vista “Home” 17](#_Toc12192031)

[Backend. 17](#_Toc12192032)

[Capa de la vista 17](#_Toc12192033)

[Capa de negocio 17](#_Toc12192034)

[Capa de datos 17](#_Toc12192035)

[Capa de servicio 18](#_Toc12192036)

[Seguridad. 18](#_Toc12192037)

[Complementos. 18](#_Toc12192038)

[Pendientes de definición. 18](#_Toc12192039)

[Pendientes Técnicos. 18](#_Toc12192040)

# Información General.

## Propósito.

El motivo principal al momento de desarrollar aplicaciones para dispositivos móviles es optimizar tiempos de procesos y en un segundo plano mostrar de una forma más didáctica dichos procesos para hacerlos mas entendibles ya sea para el cliente interno o externo.

La Alcaldía Mayor de Bogotá publicó el Decreto 793 de 2018 del 20 de diciembre de 2018, por medio del cual se establecen las normas y procedimientos administrativos, técnicos y operativos para la implementación del Sistema de Emergencias Médicas-SEM en el Distrito Capital de Bogotá y se crea el Comité Distrital de Urgencias y Gestión del Riesgo en Emergencias y Desastres en Salud.

De esta manera, establece, entre otras disposiciones, que la Secretaría Distrital de Salud es la encargada de coordinar y gestionar toda la atención prehospitalaria en Bogotá, incluido el servicio de transporte de pacientes en ambulancias, público y privado, que se origine en una situación de urgencia, emergencia o desastre.

En el Decreto se fijan los lineamientos con los que tendrá que operar el Sistema de Emergencias Médicas – SEM, en Bogotá, bajo la coordinación del Centro Regulador de Urgencias y Emergencias CRUE; se determina que toda solicitud o llamada para atención médica de urgencias en la ciudad deberá realizarse mediante la línea de emergencias 123, quien direccionará dichos requerimientos al CRUE y este a su vez gestionará los servicios que considere necesarios.

Para este caso en específico, la aplicación se desarrolló móvil Android (desde su versión 7.0 Nougat), contiene un control del inventario de productos y/o insumos solicitados por el Ministerio de Salud como mínimo para poder ejercer y a su vez regular las fechas de vencimiento de estos, estableciendo una semaforización en donde se pueda agrupar por esta variable y así dar prioridad a los que lo requieren.

## Alcance.

Diseñar, desarrollar e implementar una aplicación en Android (desde su versión 7.0 Nougat) que permita controlar los inventarios ya sea en su cantidad como en sus fechas de vencimiento para cumplir los lineamientos tanto de la Secretaría Distrital de Salud como de la Alcaldía Mayor de Bogotá.

Esto con el objetivo de poner a disposición para los funcionarios de las ambulancias como de su administrador, una aplicación mucho más dinámica en donde optimice este proceso y pueda ser regulado a diario.

## Personal involucrado.

|  |  |
| --- | --- |
| **Nombre** | Carlos Andrés Vergara Parra |
| **Rol** | Analista, diseñador y programador |
| **Categoría Profesional** | Especialista tecnológico en desarrollo de aplicaciones para dispositivos móviles |
| **Responsabilidad** | Análisis de información, diseño y programación de Móvil 54 |
| **Información de contacto** | [cavergara252@misena.edu.co](mailto:cavergara252@misena.edu.co) |

## Convenciones, definiciones, acrónimos

|  |  |
| --- | --- |
| ***Nombre*** | ***Descripción*** |
| **Usuario** | Persona que usará el sistema para gestionar procesos |
| **MOVIL 54** | Aplicación de control de inventarios para ambulancias |
| **ERS** | Especificación de Requisitos Software |
| **RF** | Requerimiento Funcional |
| **RNF** | Requerimiento No Funcional |
| **FTP** | Protocolo de Transferencia de Archivos |
| **Moodle** | Aula Virtual |

## Referencias

|  |  |
| --- | --- |
| **Título del Documento** | **Referencia** |
| Regulación Ambulancias en Bogotá | <http://www.saludcapital.gov.co/Paginas2/Noticia_Portal_Detalle.aspx?IP=731> |
| Dotación de ambulancias; Transporte Asistencial Básico y Medicalizado | <https://proyectostipo.dnp.gov.co/images/pdf/ambulancias/PTambulancias.pdf> |
| RESOLUCIÓN NÚMERO 00002003 DE 2014 | <https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/DIJ/Resolucion-2003-de-2014.pdf> |
| RESOLUCION NUMERO 9279 DE 1993 | <https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/DIJ/RESOLUCION-9279-DE-1993.pdf> |
| Estado del arte 1 | https://www.xnr-sisbi.com/stock-portal/ |
| Estado del arte 2 | <https://www.microsoft.com/es-co/p/inventory-control/9wzdncrdp95p?activetab=pivot:overviewtab> |
| Diagrama de procesos |  |

## Resumen

El control de inventarios de una ambulancia actualmente se lleva de forma manual en donde se debe reportar por lo menos una vez al mes a la Secretaría Distrital de Salud y ellos mismos realizan visitas sorpresas a las empresas de ambulancias para verificar si el inventario esta completo y así garantizar que se están cumpliendo los lineamientos.

El inventario disminuye cada vez que brinde servicios a los ciudadanos que son trasladados a los centros hospitalarios y aumenta por medio de los cargues que realiza el farmaceuta con la respectiva autorización del administrador.

Considerando lo anterior, desarrollar e implementar sistemas que permitan realizar reportes a los administradores de las ambulancias basados en los lineamientos de la Secretaría Distrital de Salud y la Alcaldía mayor de Bogotá, visualizar el inventario, realizar cargue de este y registro de pacientes con su respectivo uso de productos y/o insumos.

## Perspectiva del producto

**¿Qué aplicaciones pueden controlar el inventario de las ambulancias en específico teniendo en cuenta los requisitos del Ministerio de Salud?**

En la actualidad, no encontré alguna aplicación que específicamente tratara el tema de las ambulancias para la ciudad de Bogotá teniendo en cuenta que es una parte de los lineamientos que debe tener para brindar servicios a la comunidad y más con el tema de las fechas de vencimiento y elaboración para un mayor control ya que esto debe mantenerse controlado y dentro de los estándares de calidad por la importancia que este tiene.

Adicionalmente, al momento de intentar reportar estas situaciones, se realiza de forma manual y mensualmente cuando se puede llevar un control diario. Por otro lado, se realiza de forma manual y ocupa bastante tiempo a comparación del tiempo que se empleara con la aplicación.

## Funcionalidad del producto

Aplicación móvil para Smartphone con sistema operativo Android que permite realizar reportes de pacientes una vez recogidos por ambulancias en la ciudad de Bogotá, registrando los productos y/o insumos utilizados durante el evento y este se estará modificando directamente en el inventario general que maneja la ambulancia.

Por otro lado, se llevará un control en la fecha de vencimientos de los productos por medio de una semaforización para llevar un alto estándar de calidad durante los eventos presentados.

## Características del producto

La aplicación MÓVIL 54 está diseñada para dispositivos móviles con sistema operativo Android a partir de su versión 7.0 (Nougat).

El diseño se maneja actualmente con colores azul y blanco los cuales se encuentran en proceso de mejoramiento para manejar una gama de colores, pero estos deben ir alineamos al campo de la salud.

Para la semaforización en cuanto a las fechas de vencimiento de los productos y/o insumos se manejarán los siguientes:

* **Rojo:** Para los que presenten una fecha de vencimiento entre 1 a 30 días calendario posterior a la fecha actual.
* **Amarillo:** Para los que presenten una fecha de vencimiento entre 31 a 120 días calendario posterior a la fecha actual.
* **Verde:** Para los que presenten una fecha de vencimiento mayor o igual a 121 días calendario posterior a la fecha actual.

Todo ello desarrollado bajo el Framework desarrollo de aplicaciones móviles IONIC en su versión 4, a través del editor de código Visual Studio Code, construcción y conexión de la base de datos con Microsoft SQL Server y las diferentes librerías requeridas de acuerdo a las funcionalidades principales de la app.

## Restricciones

Para el funcionamiento correcto de la aplicación, esta debe ser instalada en un dispositivo móvil con sistema operativo Android desde su versión 7.0.1 (Nougat).

La orientación de su visualización es totalmente vertical, su funcionamiento requiere de conexión de datos a rede móvil o wifi, Utilización de Sistemas de Posicionamiento Global GPS. Registro de usuario y login dentro de la plataforma.

* Interfaz para ser usada con internet.
* Usa geolocalización GPS
* Lenguajes y tecnologías en uso: HTML, CSS, Sass, .NET, Javascript, Typescript, Ionic Framework v4, SQL Transac.
* Los servidores deben ser capaces de atender consultas concurrentemente.
* El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

# Información funcional y no funcional.

## Proceso Funcional.

|  |
| --- |
|  |

## Requerimientos Funcionales generales

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF01 |
| **Nombre del Requerimiento:** | Autentificación de Rol. |
| **Características:** | Los usuarios deberán identificarse para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad. |
| **Requerimiento NO funcional:** |  |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF02 |
| **Nombre del Requerimiento:** | Registrar Paciente. |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al Usuario registrarse. Se deben suministrar datos como: Tipo de documento, Número de documento, Nombres, Apellidos, Fecha de nacimiento, Género, Edad, Número de contacto, Correo electrónico, Contraseña. |
| **Requerimiento NO funcional:** |  |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF02 |
| **Nombre del Requerimiento:** | Cargar Inventario. |
| **Características:** | Los usuarios pueden cargar el inventario. |
| **Descripción del requerimiento:** | El sistema permitirá al Usuario registrarse. Se deben suministrar datos como: Tipo de documento, Número de documento, Nombres, Apellidos, Fecha de nacimiento, Género, Edad, Número de contacto, Correo electrónico, Contraseña. |
| **Requerimiento NO funcional:** |  |
| **Prioridad del requerimiento:**  Alta | |

## 

## Requisitos comunes de las interfaces

### Interfaces de usuario

La interfaz con el usuario consistirá en un conjunto de vistas con botones, listas y campos de textos. Ésta deberá ser construida específicamente para el sistema propuesto y, será visualizada a través de Smartphone con sistema operativo Android a partir de su versión 7.0.1 (Nougat).

### Interfaces de hardware

Será necesario disponer de equipos de cómputos en perfecto estado con las siguientes características:

* Adaptadores de red.
* Procesador de 1.66GHz o superior.
* Memoria mínima de 256Mb.
* Mouse.
* Teclado

### Interfaces de software

* Sistema Operativo: Windows XP o superior.
* Explorador: Mozilla o Chrome.

### Interfaces de comunicación

Los servidores, clientes y aplicaciones se comunicarán entre sí, mediante protocolos estándares en internet, siempre que sea posible.

## Requerimientos Funcionales específicos

### Requisito funcional 1

**Autentificación de Usuarios:** los usuariosdeberán identificarse para acceder a la aplicación.

* El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre.

### Requisito funcional 2

**Registrar Usuarios:** El sistemapermitirá al usuario (estudiante, docente y Administrador) registrarse. El usuario debe suministrar datos como: Tipo de documento, Número de documento, Nombres, Apellidos, Fecha de nacimiento, Género, Edad, Número de contacto, Correo electrónico, Contraseña.

## Requerimientos No Funcionales

### Requisitos de rendimiento

* Garantizar que el diseño de las consultas u otro proceso no afecte el desempeño de la base de datos, ni considerablemente el tráfico de la red.

### Seguridad

* Garantizar la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
* Garantizar la seguridad del sistema con respecto a la información y datos que se manejan tales sean documentos, archivos y contraseñas.
* Facilidades y controles para permitir el acceso a la información al personal autorizado a través de Internet, con la intención de consultar y subir información pertinente para cada una de ellas.

### Fiabilidad

* El sistema debe tener una interfaz de uso intuitiva y sencilla
* La interfaz de usuario debe ajustarse a las características de la web de la institución, dentro de la cual estará incorporado el sistema de gestión de procesos y el inventario

### Disponibilidad

* La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días por 24 horas, garantizando un esquema adecuado que permita la posible falla en cualquiera de sus componentes, contar con una contingencia, generación de alarmas.

### Mantenibilidad

* El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento con el menor esfuerzo posible
* La interfaz debe estar complementada con un buen sistema de ayuda (la administración puede recaer en personal con poca experiencia en el uso de aplicaciones informáticas).

### Portabilidad

* El sistema será implantado bajo la plataforma Android

# Información técnica.

## Básico.

1. Las hojas tienen columnas escondidas. Las hojas con columnas escondidas tienen un objetivo de proceso y de formulación que ayuda al control del modelo. Es importante que cuando se requiera adicionar líneas o columnas se haga un análisis de impacto sobre el modelo y sobre las rutinas o macros que utilizan información o referencias de celdas y rangos fijos.

## Base Datos.

1. Esta base de datos es el repositorio donde se almacenan los valores comunes a los usuarios
2. El concepto general de tener una base de datos con la información centralizada es tener el control de adición y modificación de datos utilizados
3. Modelo MER:

|  |
| --- |
|  |

1. Path: [**\\10.500.500.204\**](file:///\\10.500.500.204\AxedeBC\AxeBC.accdb)
2. Los perfiles administradores pueden modificar cualquier campo o tabla que reside en esta base datos.

Los perfiles usuarios pueden leer cualquier campo o tabla que reside en esta base datos.

1. Script

## Front End

El front end se realizo en asp .net core versión 2.0}

### Vista “Splashscreen”

Nos permite crear una vista de carga de la aplicación con la cual visualizaremos el logo/icono y los colores base correspondientes.

### Vista “Acceso\_Rol”

Esta vista permite el ingreso al sistema de usuarios que están registrados es la base de datos, sino están registrados, encontrarán un link de acceso que les permitirá el registro a la aplicación. El formulario de login validará si la información ingresada por el usuario está correctamente diligenciada, al realizar esto el botón de iniciar sesión se activará y permitirá validar los datos en la base de datos.

### Vista “registro\_paciente”.

El usuario seleccionará e ingresará los datos correspondientes a: tipo de documento, número de documento, nombres, apellidos, fecha de nacimiento, género, número de contacto, correo electrónico y contraseña.

El formulario validará que los datos ingresados como correo electrónico y contraseña coincidan entre sí, al realizar esto, el usuario debe aceptar los términos y condiciones del servicio. Y dar clic en registrarme.

Esta acción, enviará inmediatamente los datos a la base de datos y el usuario quedará registrado en la plataforma. Inmediatamente sus datos serán guardados y procederá al home principal.

Imágenes de las vistas.

## Backend.

El backend se realizo en asp .net core 2.0 con una arquitectura de 2 capas, models y controlers

### Capa de negocio

Se manejó la lógica de negocio de la app por medio de las clases A, B, y la clase C solo funciona como modelo de nuestro proyecto

### Capa de datos

Se creó un modelo que se conecta a nuestra base de datos que esta en el servidor 10.100.100.255:3306 por medio del usuario aaa y el password bbb.

### Capa de servicio

Se crearon las clases A, b, c, d y e que permite el uso de la api de nuestro servidor 10.100.100.255:3306 por medio del usuario aaa y el password bbb.

La clase A nos realiza los métodos get, post, put y delete de la tabla mama

La clase B nos realiza los métodos get, post y delete de la tabla papa

La clase B nos realiza los métodos get, de la tabla tio

## Seguridad.

1. La conexión con las api maneja un nivel de autenticación basado en auth 2.0 .
2. Autenticación básica con correo electrónico y contraseña proporcionado por el usuario

## Complementos.

Para compilar el programa se debe instalar el framework IONIC en su versión 4 y las siguientes dependiendencias para que pueda ser ejecutado correctamente a través del comando *npm i* en el editor de texto a través de la terminarSin embargo se debe asegurar de instalar node.js en primera instancia para ejecutar todas las instalaciones correspondientes (incluida IONIC)

El servidor de base de datos debe tener 1 Terabite de disco duro,16GB de RAM, Procesador AMD A8

# Pendientes de definición.

* Envío y recepción de Push notificación
* Mensajería
* Alertas
* Definición de Machine Learning

# Pendientes Técnicos.

* Encriptación de información
* Conexión al servidor en la nube.
* Trazado de información de directorio telefónico de acuerdo a geolocalización
* Edición de perfil de usuario

# Links artículos de referencia.

* <https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/DIJ/Resolucion-2003-de-2014.pdf>
* <https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/DIJ/RESOLUCION-9279-DE-1993.pdf>
* <https://proyectostipo.dnp.gov.co/images/pdf/ambulancias/PTambulancias.pdf>
* <http://www.saludcapital.gov.co/Paginas2/Noticia_Portal_Detalle.aspx?IP=731>