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 esta solución tecnológica es optimizar los sistemas de gestión de procesos llevados a cabo para realizar inspecciones de ascensores electromecánicos e hidráulicos, la cual es una de las principales necesidades de las empresas acreditadas para llevar a cabo el proceso de inspección basado en la NORMA TÉCNICA COLOMBIANA NTC 5926-1 la cual contiene 175 condiciones que deben cumplir estos equipos, las cuales deben contar con el visto bueno de alguna de las empresas de certificación las cuales son vigiladas por el Organismo Nacional de Acreditación de Colombia (ONAC).

A partir de esto, el seguimiento de las situaciones de riesgo y/o emergencia que se presenta en el transporte vertical en las edificaciones que utilizan estos equipos es esencial, ya que por medio de este tipo de inspecciones se permite vigilar atentamente los factores que afectan el correcto funcionamiento de estos equipos. La escena es cada vez más común: esperar un ascensor, notar que está en mantenimiento y tener que subir o bajar las escaleras. En Bogotá, apenas dos de cada diez ascensores cumplen con los requisitos mínimos de seguridad para operar. Sólo entre enero de 2016 y diciembre del 2017 se registraron 837 emergencias en estos equipos, es decir, al menos una al día, la mayoría relacionadas con fallas en el fluido eléctrico. Teniendo en cuenta esto, los niveles de inseguridad de los ciudadanos han aumentado.

De acuerdo a esto, este proyecto pretende desarrollar una aplicación móvil Android (desde su versión 7.0 Nougat) que permita agilizar el proceso de inspección y mantener controlados todos los riesgos de manera mas rápida evitando el manejo de informes técnicos en papel y sistematizando el proceso por medio de una aplicación móvil; en promedio una empresa de certificación se tarda tres días en generar un informe técnico de inspección, haciendo uso de la aplicación móvil el reporte será instantáneo ya que apenas se termine la revisión del equipo por parte del inspector del organismo de inspección este tendrá la posibilidad de generar un informe de los hallazgos encontrados con el plus de tener asociadas las fotografías tomadas en sitio al informe.

## **Alcance**

Diseñar, desarrollar e implementar una aplicación móvil Android (desde su versión 7.0 Nougat) que permita realizar inspecciones a equipos ascensores electromecánicos e hidráulicos con la finalidad de agilizar el proceso llevado a cabo por las empresas de certificación al momento de realizar la verificación de estos equipos y se pueda disminuir el riesgo de los ciudadanos al momento de hacer uso del transporte vertical ofrecido por algunas edificaciones en la ciudad.

## **Personal involucrado**

|  |  |
| --- | --- |
| **Nombre** | Juan Camilo Herrera Ardila |
| **Rol** | Analista, diseñador y programador |
| **Categoría Profesional** | Ingeniero de sistemas |
| **Responsabilidad** | Análisis de información, diseño y programación de ONApp |
| **Información de contacto** | juankmiloh@hotmail.com |

|  |  |
| --- | --- |
| **Nombre** | Miguel Antonio Mena Correa |
| **Rol** | Analista, diseñador y programador |
| **Categoría Profesional** | Ingeniero de sistemas |
| **Responsabilidad** | Análisis de información, diseño y programación de ONApp |
| **Información de contacto** |  |

## **Convenciones, definiciones, acrónimos**

|  |  |
| --- | --- |
| ***Nombre*** | ***Descripción*** |
| **Inspector** | Persona que usará el sistema para realizar inspecciones |
| **ONApp** | Aplicación para realizar inspecciones |
| **ERS** | Especificación de Requisitos Software |
| **RF** | Requerimiento Funcional |
| **RNF** | Requerimiento No Funcional |
| **FTP** | Protocolo de Transferencia de Archivos |

## **Referencias**

|  |  |
| --- | --- |
| **Título del Documento** | **Referencia** |
| Compendio NTC Transporte Vertical y Puertas Eléctricas | ICONTEC  Norma técnica colombiana NTC 5926-1 |

## **Resumen**

Agilizar los procesos de gestión de las empresas de certificación implementando y apropiando el uso de las TIC, diseñando una aplicación móvil ANDROID para la realización de inspecciones de ascensores previstas por la normativa vigente, a fin de comprobar las condiciones de seguridad de estos equipos con el uso de menos recursos en tiempo y papelería utilizados en la actualidad para la realización de este tipo de inspecciones; agilizando el proceso de inspección se busca que las empresas puedan llegar aumentar la cantidad de inspecciones realizadas con respecto a la metodología que vienen utilizando con el objetivo de disminuir riesgos a los ciudadanos que día a día hacen uso del servicio del transporte vertical.

Considerando lo anterior, se busca desarrollar e implementar sistemas que permitan mejorar los índices de seguridad y calidad de vida del ciudadano y al mismo tiempo permitan optimizar los recursos de las empresas.

## **Perspectiva del producto**

**¿Qué herramientas tecnológicas existen para llevar a cabo el proceso de inspección a ascensores?**

En la actualidad, no se cuenta con una herramienta tecnológica que permita realizar una valoración del equipo ascensor en todos sus componentes: cabina, cuarto de máquinas, foso y pozo, por ello desarrollar e implementar una aplicación móvil que permita realizar este proceso de verificación permitirá obtener con mayor rapidez los hallazgos con algún tipo de inconformidad con el objetivo que puedan ser corregidos a la mayor brevedad posible.

## **Funcionalidad del producto**

Aplicación móvil para Smartphone con sistema operativo Android que permite realizar el registro de inspecciones de ascensores basándose en la NORMA TÉCNICA COLOMBIANA NTC 5926-1 permitiendo a su vez tomar registros fotográficos de cada uno de los ítems de inspección y además generar informes técnicos en sitio y un reporte final de inspección de manera automática. Adicionalmente, la aplicación funcionara de manera offline asegurando que no haya perdida de la información capturada en las instalaciones del ascensor donde no se cuenta con cobertura a internet.

## **Características del producto**

La aplicación ONApp está diseñada para dispositivos móviles con sistema operativo Android a partir de su versión 7.0 (Nougat). El diseño de su interfaz fue desarrollado con la intención de ser fácil e intuitivo, basado en material design, como color principal el azul, usado para acciones importantes en la aplicación. Esta corriente en el diseño de interfaces permite la visualización de los elementos de manera óptima, facilitando la lectura, usabilidad, funcionalidad y optimización de los recursos del dispositivo. La tipografía seleccionada, los colores de contraste usados en combinación con sus aplicaciones regular y médium permiten diferenciar jerárquicamente la información principal de la secundaria, respecto a la distribución y visualización de los textos y tarjetas utilizadas para organizar la información. Los formularios están acompañados de texto e iconografía alusiva al tipo de dato que debe ingresar el usuario, permitiéndole guiarlo sobre la información que debe llenar en cada uno. Adicionalmente se generaron piezas distintivas donde el usuario puede llevar un control de la inspección que está realizando.

El Icon App y el Splashscreen contienen el logo correspondiente a la aplicación, en colores contrastantes al azul con fondos oscuros para darle suavidad y simpleza. La aplicación tiene tres tipos de navegación, menú superior lateral izquierdo, que es accesible en los primeros niveles de la aplicación y permite ver la información del usuario que esta utilizando la aplicación así como acceder a las configuraciones de usuario, la aplicación consta también de un menú principal que permite navegar por las opciones de realizar inspección o cargar las inspecciones realizadas, también consta de un menú selector de equipos a inspeccionar el cual luego de ser seleccionado permite una navegación tipo tabs, donde se accede directamente a las secciones importantes de la lista de verificación definida para ascensores: cabina, cuarto de máquinas, pozo y foso.

Las listas de verificación principales requieren el uso del scroll, por ende, el uso de la pantalla completa es fundamental, requiriendo limpieza visual al momento de su uso e implementación. Todo ello desarrollado bajo el Framework de desarrollo de aplicaciones móviles IONIC en su versión 4 a través del editor de código Visual Studio Code, construcción y conexión del modelo de datos con FIREBASE y las diferentes librerías requeridas de acuerdo a las funcionalidades principales de la app.

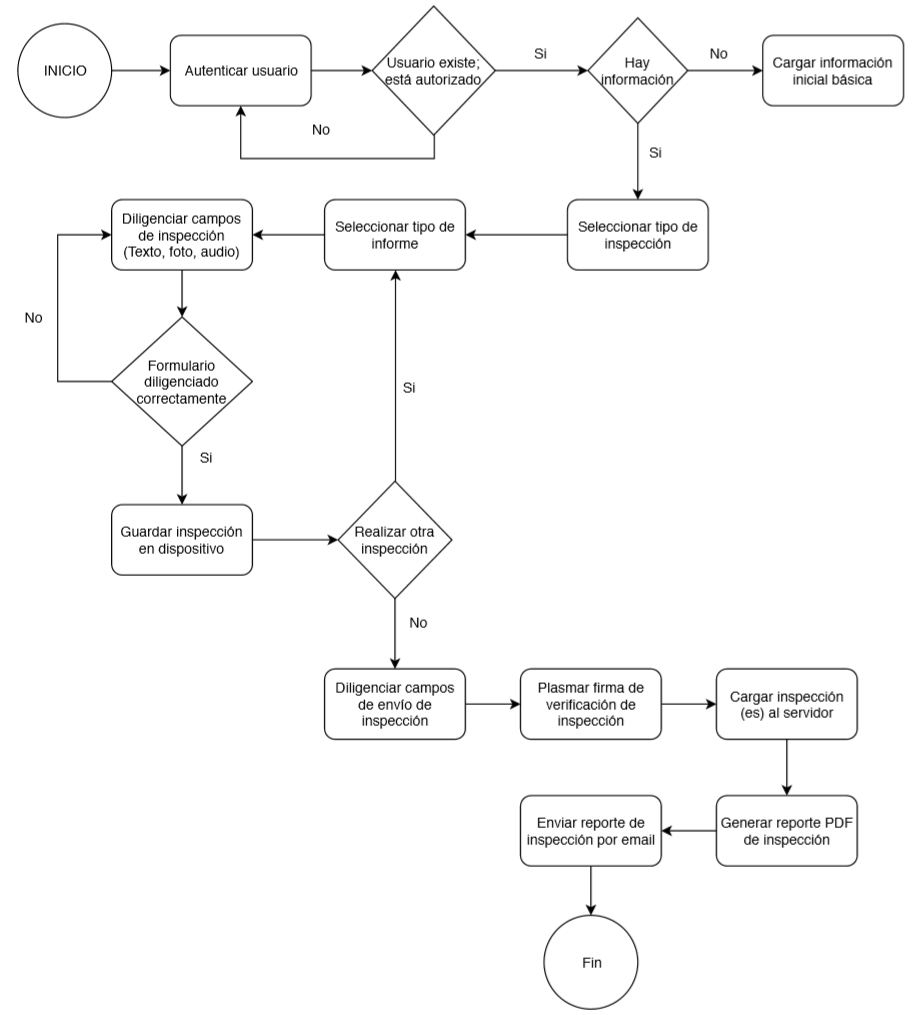
## **Restricciones**

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 no necesariamente requiere de conexión de datos a redes móviles o wifi. Registro de usuario y login dentro de la plataforma.

* Interfaz para ser usada online/offline teniendo en cuenta el módulo a utilizar.
* Lenguajes y tecnologías en uso: HTML, CSS, Sass, FIREBASE, Javascript, Typescript, Ionic Framework v4.
* 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 Usuario |
| **Características:** | Los usuarios deberán identificarse para acceder a cualquier parte del sistema. Esta funcionalidad funciona online/offline. |
| **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:** | Realizar inspección |
| **Características:** | El usuario podrá acceder a las diferentes listas de verificación para realizar inspecciones. |
| **Descripción del requerimiento:** | Este módulo podrá ser consultado por un usuario con rol de inspector. |
| **Requerimiento NO funcional:** |  |
| **Prioridad del requerimiento:** | Alta |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF03 |
| **Nombre del Requerimiento:** | Cargar inspección |
| **Características:** | El usuario podrá acceder al modulo de carga de inspecciones para visualizar las inspecciones realizadas y su posterior envío para revisión del administrador. |
| **Descripción del requerimiento:** | Este módulo podrá ser consultado por un usuario con rol de inspector. |
| **Requerimiento NO funcional:** |  |
| **Prioridad del requerimiento:** | Alta |

## **Requerimientos No Funcionales**

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RNF01 |
| **Nombre del Requerimiento:** | Interfaz de la aplicación |
| **Características:** | La aplicación presentara una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios. |
| **Descripción del requerimiento:** | La aplicación debe tener una interfaz de uso intuitiva y sencilla. |
| **Prioridad del requerimiento:** | Alta |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RNF02 |
| **Nombre del Requerimiento:** | Desempeño |
| **Características:** | La aplicación garantizara a los usuarios un desempeño en cuanto a los datos almacenado en el sistema ofreciéndole una confiabilidad a esta misma. |
| **Descripción del requerimiento:** | Garantizar el desempeño de la aplicación 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. |
| **Prioridad del requerimiento:** | Alta |

## **Requisitos comunes de las interfaces**

### **Interfaces de usuario**

La interfaz de usuario consistirá en un conjunto de vistas con botones, listas con checklist, imágenes 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 cómputo en perfecto estado con las siguientes características:

* Adaptador de red
* Procesador de 1.66GHz o superior
* Memoria RAM mínima de 4gb
* Mouse
* Teclado

### **Interfaces de software**

* Sistema Operativo: Windows 7 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, este módulo servirá con o sin conexión a internet.

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

### **Requisito funcional 2**

**Realizar inspección:** El sistemapermitirá al usuario (inspector) acceder a las listas de verificación con el objetivo de levantar la información relacionada con el equipo que se inspeccione.

### **Requisito funcional 3**

**Cargar inspección:** El sistemapermitirá al usuario (inspector) acceder a la lista de inspecciones realizadas, para su envío a FIREBASE y posterior consulta por el administrador del sistema.

## **Requerimientos No Funcionales**

### **Requisitos de rendimiento**

* Garantizar que el diseño de las colecciones 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, fotografías 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.

### **Fiabilidad**

* El sistema debe tener una interfaz de uso intuitiva y sencilla.

### **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. La vista de la observación final de inspección será visible siempre y cuando todas las listas de verificación (datos básicos, cabina, máquina, foso y pozo) estén completamente diligenciadas.

## **Base Datos**

1. El sistema maneja una base de datos no relacional (colecciones) auspiciada por FIREBASE.
2. El concepto general de tener una base de datos no relacional es tener el control de la adición y la modificación de la información de una manera ágil y sencilla.
3. Los perfiles administradores pueden modificar cualquier colección creada en la base de datos.
4. El perfil de inspector puede acceder las colecciones que contienen las listas de verificación e inspecciones realizadas en la base datos.

## **Frontend**

El frontend se realizó en IONIC versión 4 y angular 8.

### **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 “Login”**

Esta vista permite el ingreso al sistema de los usuarios que están registrados en la base de datos, sino están registrados, el administrador del sistema deberá generar el respectivo registro. 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”**

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.dinero.com/tendencias/articulo/como-le-fue-a-bogota-en-ranking-de-calidad-de-vida-de-mercer/268324>

\* <https://pulsosocial.com/2013/12/16/5-tecnologias-para-mejorar-la-seguridad-ciudadana-en-america-latina/>

\*<https://www.mintic.gov.co/gestionti/615/w3-propertyvalue-6780.html>

\*<https://www.colciencias.gov.co/convocatorias/2014/convocatoria-para-el-desarrollo-soluciones-tecnologicas-gobierno-en-linea-2014>

\*<https://www.mintic.gov.co/gestionti/615/w3-propertyvalue-6782.html>

\*<http://www.pensamientopenal.com.ar/system/files/2015/09/doctrina41974.pdf>

\*<https://blogs.iadb.org/seguridad-ciudadana/es/tecnologia-contra-el-crimen-entusiasmo-con-criterio/>