DESARROLLO DE UNA APK

PARA REGISTRO UNICO DE VENDEDOR INFORMAL

PRESENTADO POR

FORERO TORRES LUIS ANTONIO

SERVICIO NACIONAL DE APRENDIZAJE

SENA

ESPECIALIZACION EN DESARROLLO DE

APLICACIONES MOVILES

DICIEMBRE DE 2019

DESARROLLO DE UNA APK

PARA REGISTRO UNICO DE VENDEDOR INFORMAL

FORERO TORRES LUIS ANTONIO

INSTRUCTOR

NESTOR ANDRES GARCIA PINZON

SERVICIO NACIONAL DE APRENDIZAJE

SENA

AREA DE DESARROLLO

DICIEMBRE DE 2019

**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 “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 soluciones tecnológicas al servicio del ciudadano es optimizar los sistemas gestión de procesos y/o comunicación con la población a la cual está dirigida.

A partir de allí, el seguimiento de los registros no existentes de vendedores formales e informales en calidad de riesgo y/o emergencia que se presentan en las zonas de una ciudad se vuelve esencial, ya que nos permite vigilar atentamente los factores que afectan la percepción de la calidad vida de los ciudadanos. La ciudad de Bogotá se encuentra en el puesto 128 de 223 en el ranking de calidad de vida (marzo 2019), esto teniendo en cuenta que no ha tenido grandes avances significativos al momento de mejorar las condiciones de calidad de vida de los comerciantes, teniendo en cuenta esto, los niveles de seguridad de los ciudadanos no han aumentado, tenido un comparativo con la ciudad de Caracas, Venezuela, estando este en el ranking 202 de esta lista.

De acuerdo a esto, este proyecto pretende desarrollar una aplicación móvil Android (desde su versión 6.0 Nougat) que permita reportar las situaciones percepción y afectación de la calidad de vida de acuerdo a los índices de registro de los comerciantes de determinada zona, con la finalidad de visualizar las zonas vulnerables en estos aspectos en la ciudad de Bogotá, Colombia.

**Alcance.**

Diseñar, desarrollar e implementar una aplicación en Android (desde su versión 7.0 Nougat) que permita registrar de manera precisa la cuantía de vendedores informales en la ciudad de Bogotá DC, Colombia con la finalidad de conocer la cantidad de comerciantes de determinada localidad reportada por las alcaldías locales a nivel distrital.

Esto con el objetivo de poner a disposición, tanto de los ciudadanos como de las entidades correspondientes, las zonas que están siendo censadas por los diferentes tipos de entes competentes y puedan brindarles una calidad de vida digna y de esta manera estar carne tizados y posterior mente evitar inconvenientes con la Alcaldía Mayor de Bogotá.

**Personal involucrado.**

|  |  |
| --- | --- |
| **Nombre** | Luis Antonio Forero Torres |
| **Rol** | Analista, diseñador y programador |
| **Categoría Profesional** | TSU-Informática |
| **Responsabilidad** | Analista de información, diseño y programación de RUVI |
| **Información de contacto** | [laforero1@misena.edu.](mailto:laforero1@misena.edu.)co |

**Convenciones, definiciones, acrónimos**

|  |  |
| --- | --- |
| ***Nombre*** | ***Descripción*** |
| **Usuario** | Persona que usará el sistema para gestionar procesos |
| **ruvi** | Aplicación de registro y conteo de comerciantes informales |
| **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** |
| **IPES** | <https://ipes.gov.co> |
| Artículo revista dinero | <https://revistadinero.com> |
| Estado del arte 1 | [www.gobiernobogota.gov.co](http://www.gobiernobogota.gov.co) |
| Estado del arte 2 |  |

## **RESUMEN**

Apoyar el desarrollo de las estrategias sectoriales, implementar y apropiar en uso de las TIC, facilitar la participación, medir, monitorear y crear conocimiento son algunos de los objetivos del Ministerio de las Tecnologías de la información y las Telecomunicaciones (Min TIC) con respecto al fortalecimiento de la gestión de la información a los ciudadanos.

El objetivo principal al momento de desarrollar soluciones tecnológicas al servicio del ciudadano es gestionar eficazmente la comunicación con los comerciantes informales para la cual está dirigida. Sin embargo, la participación de los actores principales, los comerciantes, es fundamental al momento de generar una realimentación efectiva respecto a los procesos que deben ser gestionados de acuerdo a las problemáticas que afectan su comunidad.

El conteo urbano es una de las principales necesidades a abordar respecto a las problemáticas de los comerciantes no formales de las comunidades de la ciudad de Bogotá DC; Por ende, diseñar, desarrollar e implementar soluciones tecnológicas que abarquen y permitan realizar seguimiento a las problemáticas de seguridad, justicia y convivencia, se vuelve primordial a la hora de fomentar la participación ciudadana dentro de sus entornos.

Cabe anotar que el reporte ciudadano, como herramienta de control social, es fundamental para conocer el estado y la percepción de las zonas en las cuales los habitantes conviven. Sin embargo, actualmente no se cuenta con herramientas cualitativas de participación ciudadana que nos permitan cuantificar las situaciones que afectan la calidad de los mismos

Considerando lo anterior, desarrollar e implementar sistemas que permitan realizar un censo preciso de los comerciantes informales cuantificados por las alcaldías locales de la zona y en orden de la Alcaldía Mayor de Bogotá basados en los índices de seguridad y calidad de vida, visualizar el estado de las zonas cercanas y facilitar la comunicación con las entidades correspondientes al manejo de estas

## PERSPECTIVA DEL PRODUCTO

**¿Qué herramientas practicas existen para hacer censos de vendedores informales?**

En la actualidad, la ciudad de Bogotá no cuenta con una herramienta de uso masivo que permita cuantificar los comerciantes no formales de determinada zona, con la cual se puedan visualizar las zonas con mayor número de indicadores en las cuales cohabitan los ciudadanos.

Adicionalmente, al momento de intentar censar a los propietarios de los establecimientos, los ciudadanos desconocen las entidades a las cuales deben dirigirse o en su defecto no tiene al alcance el contacto directo al cual creen pertinente acudir. Por lo general acuden a la Alcaldía mayor de Bogotá, quienes direccionan correctamente la atención de estas circunstancias.

Sin embargo, las comunidades poseen redes de apoyo a las cuales acudir si sus miembros se encuentran en situaciones de riesgo o emergencia evidente, a pesar ello, el apoyo puede ser poco o nulo, las personas desean informarse de las situaciones de riesgo que afectan su estilo de vida con respecto a la seguridad de las zonas en las cuales laboran, todo con la intención de salvaguardar su integridad, su patrimonio laboral y el de los miembros de su familia.

Por ello, desarrollar e implementar una aplicación móvil que permita reportar las situaciones de riesgo, cuantificar, conocer sus condiciones actuales permitirá aportar a la comunidad la información correspondiente al estado de seguridad de sus zonas cercanas, contar con los miembros de la comunidad si es necesario y comunicarse con las entidades correspondientes de acuerdo al tipo de situación en la cual se encuentren.

## **FUNCIONALIDAD DEL PRODUCTO**

Aplicación móvil para Smartphone con sistema operativo Android que permite realizar reportes georreferenciados con los cuales se trazará en el mapa las zonas con más puntos bajo los siguientes índices: cantidad, nacionalidad, y condición de vida actual. Con esta información, los usuarios podrán consultar a su alrededor, de acuerdo a la ubicación actual obtenida con el Sistema de Posicionamiento Global (GPS) de su dispositivo móvil, los puntos en el mapa donde se han reportado incidentes o situaciones de riesgo.

Adicionalmente, la aplicación contará con un directorio telefónico con el cual, el usuario podrá dirigirse a los establecimientos y aliados comerciales. Sin embargo, soló arrojará los números de los entes de competentes de acuerdo a su ubicación actual. Los usuarios solo podrán registrarse si se encuentran dentro de la aplicación.

## **CARACTERÍSTICAS DEL PRODUCTO**

La aplicación RUVI 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 el Mockup de Adobe XD con colores de alto contraste, como color principal el azul y su complementario el blanco, 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 y funcionalidad. La tipografía seleccionada, los colores de contraste a blanco y gris 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.

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 gráficas distintivas para los marcadores de posición de acuerdo al tipo de indicador de seguridad que se visualiza en la pantalla.

El Icon App contiene el logo correspondiente a la aplicación, en colores contrastantes al azul, como el rojo, y el gris, con fondos blancos en opacidades para darle suavidad y simpleza.

La aplicación tiene dos tipos de navegación, menú superior lateral izquierdo, que es accesible en los primeros niveles de la aplicación y me permite navegar por todas las opciones correspondientes a la misma, y en el home principal navegación tipo en blanco, donde se accede directamente a las tres secciones importantes de la app: Usuarios, registros.

Estas funciones principales, requieren el uso del mapa/ geolocalización, 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 desarrollo de aplicaciones móviles IONIC en su versión 4.O, 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, .NET, Javascript, Typescript, Ionic Framework v4, MYSQL.
* 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.

**REQUERIMIENTOS FUNCIONALES GENERALES**

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF01 |
| **Nombre del Requerimiento:** | Loguear |
| **Características:** | Los usuarios deberán identificarse para acceder a cualquier para ingresar al sistema. |
| **Descripción del requerimiento:** | El sistema podrá ser consultado por el administrador y/o usuario dependiendo al 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:** | Autenticación y registro de usuario |
| **Características:** | Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del aplicativo. |
| **Descripción del requerimiento:** | El sistema permitirá al Usuario registrarse. Se deben suministrar datos como: Nombres, Apellidos, Edad, Género, Número de contacto, Correo electrónico, Usuario, Contraseña. |
| **Requerimiento NO funcional:** |  |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF03 |
| **Nombre del Requerimiento:** | Menú, consulta y registro de documento del usuario Informal. |
| **Características** | Los usuarios deberán registrar en el sistema cada usuario Informal para completar el formulario del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al Usuario registrar datos personales. Se deben suministrar datos como: Tipo de documento, Número de documento, |
| **Requerimiento NO func­­­­­­ional:** |  |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF04 |
| **Nombre del Requerimiento:** | Registro de Datos |
| **Características** | Los usuarios deberán registrar en el sistema cada usuario Informal para completar el formulario del sistema. |
| **Descripción del requerimiento:** | El sistema permitirá al Usuario registrar datos. Se deben suministrar datos como: Nombres, Apellidos, Edad, Genero, Dirección, Número de contacto, Correo electrónico, Discapacidades, Desplazados, EPS, Tiempo de labor Informal, Dirección de labor Informal y Producto que comercializa. |
| **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 usuario. |
| **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 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 realizó en Framework IONIC versión 4.0 con una arquitectura compuesta por 17 capas.

### Vista “Home”

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 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”.

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 Framework IONIC versión 4.0 con una arquitectura de 6 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 está 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 dependencias para que pueda ser ejecutado correctamente a través del comando *npm i* en el editor de texto a través de la terminar Sin 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