**Método de la Ingeniería**

**Identificación del problema**

La empresa de mensajería colombiana 4-72 busca la innovar en la época post-cuarentena, por esta razón a decidido crear un aplicativo para que sus usuarios puedan conocer la ubicación de sus oficinas y horarios, además, también pueden visualizar la ubicación en un servicio de mapas (Por ejemplo, Google Maps u Open Street Maps) para así conocer cual es la mas cercana a su ubicación.

La información de las ubicaciones y horarios de las oficinas del País le será entregada en un documento de Excel, usted deberá manejar esa información con la estructura que considere más adecuada, tenga en cuenta además que deberá también mostrarle al usuario un mapa interactivo donde el usuario pueda escoger la oficina de acuerdo con criterios, bien sea por departamento, ciudad u horario. Tenga en cuenta tanto la información, el mapa y la búsqueda deberán tener una interfaz amigable con el usuario

## Recopilación de la información necesaria

Una Estructura en términos de ingeniería de software es un conjunto de técnicas y métodos para almacenar y procesar datos según (Granada & Sánchez Morales, 2001) Una estructura es un tipo de dato compuesto que permite almacenar un conjunto de datos de diferente tipo. Los datos que contiene una estructura pueden ser de tipo simple (caracteres, números enteros o de coma flotante etc.) o a su vez de tipo compuesto (vectores, estructuras, listas, etc.).

Además de lo anterior se utilizará también una API para mostrar el mapa, sin importar la solución utilizar una API es la manera mas eficiente de acuerdo a las técnicas de reutilización que se han visto desde el principio de la carrera.

Una API (Application Programming Interface) en español (interfaz de programación de aplicaciones) es una serie de técnicas, definiciones y protocolos para construir e integran software, permite comunicar diferentes productos y servicios sin saber cómo están implementadas, esto con la finalidad de simplificar la parte del desarrollo con herramientas previamente desarrolladas y probadas en funcionamiento y eficacia (RedHat, 2017).

Dentro de estos APIs, se debe utilizar uno llamado *GMaps,* un API creada por Google para la implementación de sus servicios de Geolocalización en aplicaciones, esta disponible para su implementación en diferentes lenguajes (Por ejemplo Java,Kotlin,C#, Python,JavaScript entre muchos otros)

Los datos para los mapas deben ser de bases de datos abiertas, dichas bases son iniciativas de gobiernos y organizaciones para la transparencia de sus datos, de acuerdo con (Ministerio de Tecnologia, Informacion Y Comunicaciones de Colombia, 2020)

GOV.CO es el Portal Único del Estado Colombiano que ofrece en un solo lugar toda la oferta de trámites, servicios, ejercicios de participación e información del país. Somos una estrategia nacional de integración digital que de manera creciente brindará una experiencia efectiva en el relacionamiento Estado- ciudadano. Es un Estado ágil, eficiente y seguro.

## Búsqueda de soluciones creativas

**Ideas para la creación de la aplicación:**

1. Usar códigos de aplicaciones que tengan algún aplicativo de mapas prediseñados en algún lenguaje de programación que nos sea de interés como C# o Java.
2. Usar algún Framework de bases de datos para poder recibir de forma eficiente grandes cantidades de datos para su procesamiento y visualización.
3. Construir nuestra propia API para mapas usando herramientas del lenguaje de programación con el que el aplicativo será creado.
4. Usar el API y herramientas de Google Maps que se encuentren disponibles como librerías o packages para los lenguajes de programación con los que vaya a trabajar el aplicativo.

## Ideas para la carga y visualización de los datos.

1. Utilizar estructuras de Datos previamente establecidas (Listas, Arboles, Listas Enlazadas).
2. Utilizar algún API de C# para la carga de tablas extrayendo los datos desde la base de datos proporcionada.
3. Utilizar algún API de C# para la carga de mapas.
4. Usar librerías externas que faciliten la carga de bases de Datos por el explorador de archivos como aplicativos LinQ o External.
5. Contratarlo en algún programa de FreeLancer como Fiver y no hacer nada.
6. Preguntarle al monitor.

## Transición de la formulación de ideas a los diseños preliminares

Alternativa 1:

* Esta alternativa es viable debido a que sigue uno de los principios de la Ingeniería de Software, que es la reutilización de código ya existente. Por tanto, esta es una alternativa mucho mas viable que construir desde 0, un sistema de procesamiento de bases de datos y visualización de estos

Alternativa 2:

* Esta alternativa es mucho mas viable que construir el API desde 0 además el API esta provista por Google, ya que esta depurada y es mucho mas eficaz que alguna API construida por nosotros mismos, por estas razones es mejor utilizar el API de Google.

Alternativa 3:

- Esta alternativa es muy importante ya que no vamos a construir ninguna estructura de datos, ya que las que conocemos actualmente soportan perfectamente la base de datos. Es por ello, que esta alternativa es viable.

Alternativa 4:

* Esta alternativa esta estrechamente relacionada con la alternativa 1, puesto que, una vez procesada la base de datos, es necesario mostrarle datos al usuario, para esto se utilizarán librerías propias de C#

## Evaluación y selección de la mejor solución

**Criterio 1 Carga y Visualización de Datos**

La aplicación debe ser capaz de mostrar una gran cantidad de datos que se envían de forma externa.

[1]. La Aplicación no muestra de manera correcta la carga de datos y su visualización se da de manera incompleta y desordenada.

[3]. La Aplicación carga de manera correcta los datos y los muestra en una tabla. Sin embargo, la información se encuentra en desorden y no se muestra de forma clara para el usuario

[2]. La Aplicación carga de manera correcta los datos y los muestra en una tabla agrupando la información de manera que sea entendible para el usuario.

**Criterio 2 Visualización de los datos de Latitud y Longitud**

La aplicación trabaja con la información de varias ubicaciones en un mapa desplegando los datos de longitud y latitud de manera adecuada.

[1]. No es posible visualizar los datos en el mapa.

[2]. Las ubicaciones se visualizan de manera dispersa o incorrecta (incluye lugares no previstos).

[3]. Las ubicaciones se muestran correctamente.

**Criterio 3 Visualización de un Grafico de un grupo de datos.**

La aplicación muestra graficas estadísticas sobre un grupo de datos las cuales deben dar información sobre los datos contenidos en el Dataset.

[1]. No se visualiza ninguna grafica con información sobre los datos.

[2]. Las gráficas se muestran de manera desorganizada y sus variables no son coherentes con un modelo estadístico.

[3]. Las gráficas muestran resultados apropiados sobre la información del Dataset y su visualización se muestra de forma ordenada.

|  |  |  |  |
| --- | --- | --- | --- |
| **Alternativa** | **Criterio 1** **Carga y Visualización de Datos** | **Criterio 2 Visualización de los datos de Latitud y Longitud** | **Criterio 3 Visualización de un Gráfico de un grupo de datos.** |
| Alternativa 1 | 3 | 1 | 3 |
| Alternativa 2 | 1 | 3 | 1 |
| Alternativa 3 | 1 | 1 | 3 |
| Alternativa 4 | 3 | 1 | 3 |

**Visualización de los datos:**

* Se decide usar la librería de Windows Forms ya que esta tiene mas documentación y por tanto es mas fácil de hacer, sin contar que es mucho mas sencillo de utilizar las librerías y APIs externas

**Visualización de los datos**

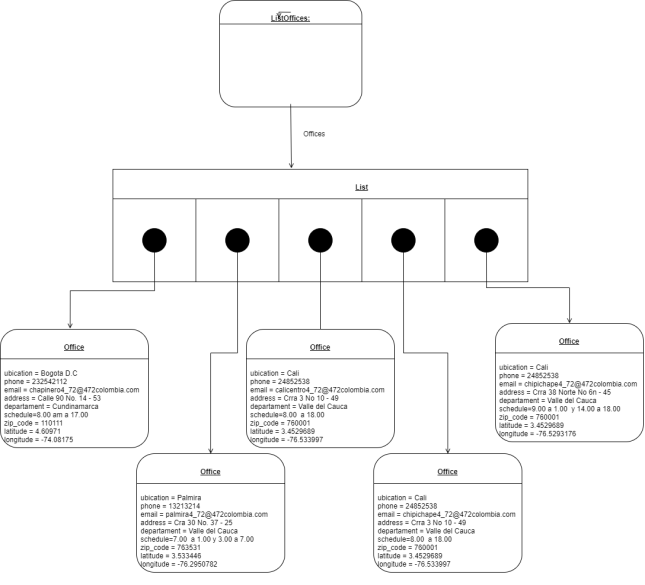
Se decidió utilizar un .csv o archivo delimitado por comas puesto que es accesible por la mayoría de bases de datos comerciales por tanto si se desea se puede optimizar, además de ello se decidió también mostrarla gráficamente con elementos del Windows Forms

## Modelo de software

Se decide utilizar la librería de Google Maps (GMaps) para los mapas y el lenguanje de C# para consultas y operaciones en los datos.

## Diagramas

## 

****

## Video

## Bibliografía

Granada, U. d., & Sánchez Morales, J. I. (20 de 03 de 2001). *Departamento de Ciencias de la Computacion e Inteligencia Artifcial de la Universidad de Granda*. Obtenido de http://decsai.ugr.es/~jfv/ed1/c/cdrom/cap7/cap71.htm

Ministerio de Tecnologia, Informacion Y Comunicaciones de Colombia. (25 de 04 de 2020). *Datos Abiertos Colombia*. Obtenido de https://www.gov.co/home/

RedHat. (31 de 10 de 2017). *RedHat*. Obtenido de https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces

+