UNIVERSIDAD ICESI

Método de la Ingeniería

Paso 1. Identificación del problema

Se identifican de manera adecuada las necesidades que se presentan en el problema propuesto así como sus síntomas y condiciones por los que debe ser resuelto.

Identificación de sus síntomas y necesidades

- Desconocimiento de la información sobre los grupos de investigación.
- Se necesita tener estadísticas acerca de los artículos frecuentados por los grupos de investigación.
- Se requiere saber en qué región se encuentran los diferentes grupos de investigación en el país.

Descripción del problema:

Una importante institución como Colciencias y la Asociación Colombiana para el Avance de la Ciencia busca mejorar la capacidad competitiva entre los grupos de investigación del país para poder así desarrollar mejores soluciones a los diferentes problemas sociales fundamentales.

El sistema debe estar en la capacidad de:

Requerimientos funcionales:

Nombre	**R1 - Gestionar información de grupos de investigación**
Resumen	El requerimiento implica la utilización de información de los diferentes grupos de investigación reconocidos del país, para poder brindarle de manera visual.
Entradas	
(Archivo.txt) Con la información de los grupos de investigación requerida, la cual lo provee Gobierno Digital Colombia.	
Resultados	

Nombre	**R2 - Registro de grupo de investigación**
Resumen	Este requerimiento implica la utilización de información de un grupo de investigación, para así agregarlo al registro de los grupos existentes del país.
Entradas	
(String nombre) Indica el nombre del grupo de investigación (int código) Indica el código del grupo de investigación (String clasificación) Indica la clasificación que corresponde al grupo de investigación. (int numArticulos) Indica el numero de artículos del grupo de investigación (String ciudad) Indica la ciudad a la cual pertenece el grupo de investigación (String área de investigación) Indica el área de investigación en la cual se enfoca el grupo de investigación. (String región) Indica la región a la cual pertenece el grupo de investigación.	
Resultados	
un bool “true” indicando que el grupo de investigación se registró correctamente, un “false” indicando que el grupo de investigación no se pudo registrar.	

Nombre	**R3 - Actualización de datos de grupo de investigación**
Resumen	Este requerimiento implica la actualización de la información guardada de un grupo de investigación existente.
Entradas	
(String nombre) Indica el nombre del grupo de investigación (String ciudad) Indica la ciudad a la cual pertenece el grupo de investigación (String clasificación) Indica la clasificación que corresponde al grupo de investigación.	

(int numArticulos) Indica el numero de artículos del grupo de investigación (String área de investigación) Indica el área de investigación en la cual se enfoca el grupo de investigación. (String región) Indica la región a la cual pertenece el grupo de investigación.
Resultados
Un bool "true" indicando si se actualizo la información del grupo de investigación, "false" en el caso contrario o en caso de que no se encuentre el grupo solicitado.

Nombre	**R4 - Generar reportes**
Resumen	Este requerimiento implica la generación de reportes los cuales impliquen consolidar la información de los grupos por regiones,ciudades, áreas de investigación y clasificación.
Entradas	
(Archivo.txt) Con la información de los grupos de investigación requerida, la cual lo provee Gobierno Digital Colombia.	
Resultados	

Nombre	**R5 - Consultar cantidad de artículos frecuentados**
Resumen	Este requerimiento implica la opción de poder realizar consultas en la aplicación donde se muestre los grupos de artículos más frecuentados por los diferentes grupos de investigación
Entradas	

Resultados
un reporte con las siguientes características: (String cant) indicando la cantidad de articulos frecuentados.

Nombre	**RN1 - Visualizar información gráficamente.**
Resumen	La aplicación debe mostrar las reportes generados mediante una interfaz gráfica que contenga un mapa, donde se puedan visualizar la posición de los diferentes grupos de investigación , las regiones, ciudades, áreas de investigación y clasificación.

Nombre	**RN2 - Interfaz gráfica**
Resumen	La aplicación debe ser amigable con el usuario, proporcionando una interfaz gráfica sencilla en la cual se pueda mover facil y rapidamente.

Paso 2. Recopilación de Información

Con el propósito de tener claros todas las definiciones involucradas se hace una búsqueda de los términos relacionados con el problema planteado. Esta búsqueda se realiza en fuentes reconocidas y confiables para reconocer que conceptos hacen parte del problema y no.

Fuentes:
https://es.wikipedia.org

http://www.alegsa.com.ar/Dic/texto_plano.php

https://docs.microsoft.com/es-es/dotnet/framework/winforms/windows-forms-overview

GitHub:

Es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de computadora. El software que opera GitHub fue escrito en Ruby on Rails.

Archivos de texto plano:

Son aquellos formados exclusivamente por texto sin ningún formato, es decir, que no requieren ser interpretados para leerse. También son llamados archivos de texto llano, simple o sin formato.

Windows Forms:

Las características principales de la programación de Windows Forms y cómo puede usar Windows Forms para compilar smart clients que satisfagan las necesidades actuales de las empresas y usuarios.

C#

C# (pronunciado si sharp en inglés) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común

Microsoft .NET

.NET es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones.

Visual Studio

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET,

F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades online bajo Windows Azure en forma del editor Monaco.

Colciencias

Es el departamento administrativo de ciencia, tecnología e innovación (CTeI) que depende de la presidencia de la república y lidera el sistema nacional de ciencia, tecnología e innovación. La entidad enfoca sus esfuerzos en 4 grandes áreas de trabajo:

ACAC(asociación colombiana para el avance de la ciencia)

Contribuir al fomento de la ciencia, la tecnología y la innovación creando conciencia pública de su importancia y desarrollando estrategias para el beneficio de la sociedad.

Datos Abiertos (datos.gov.co)

Con el proyecto de Datos Abiertos, el Gobierno Colombiano promueve la transparencia, el acceso a la información pública, la competitividad, el desarrollo económico, y la generación de impacto social a través de la apertura, la reutilización de los datos públicos, y el uso y apropiación de las TIC.

Gmaps

GMaps.js es una librería con .dll que se basa en Google Maps y que permite publicar mapas en la web de forma extremadamente sencilla. Utiliza jquery lo que permite reducir el código al máximo y hacerlo fácilmente entendible.

Nos permite crear mapas con marcadores, rutas, geolocalización, perfiles longitudinales… y muchas otras funciones. La documentación es un poco deficiente, pero a cambio tiene un buen catálogo de ejemplos.

Paso 3. Búsqueda de Soluciones Creativas.

Para este paso, se decidió proponer soluciones propias utilizando como herramienta de trabajo grupal de lluvia de ideas. Las soluciones que se plantearon fueron:

Alternativa 1:
Se propone usar una herramienta de representación de ideas, preferiblemente visuales,como lo puede ser un croquis. Esto con el objetivo de representar un mapa de colombia con las regiones dadas por la base de datos. Se prefiere que sea dibujado por un programa especializado, además de eso la herramienta estaría en capacidad de la lectura de datos y generar reportes necesarios para la solución de la problemática principal.

Alternativa 2:
Se sugiere utilizar el programa de visualización vial Waze, como bien se sabe este software es una herramienta en la vida cotidiana de muchos, al utilizarla además de solucionar problemas de viabilidad hacia los puntos de investigación también ayudará con la visualización de los grupos. El principal objetivo será complementar el programa para la representación de la información y ubicación de los grupos de investigación.

Alternativa 3:
Se plantea realizar un congreso semestral con cada uno de los diferentes grupos de investigación, para compartir hallazgos, reportes o quejas, donde cada equipo de trabajo debe realizar un reporte antes de atender para tener las últimas noticias acerca de la investigación y acerca del grupo de investigación. El Congreso es donde se incentiva a cada grupo para que siga participando y se premian sus logros mientras mejoran algún problema social.

Alternativa 4:
Entre las amplias opciones nos encontramos con una herramienta de ubicación desarrollada por Apple, llamada Apple Maps, esta herramienta desarrollada para el sistema operativo IOS, permite visualizar los diferentes puntos en el mapa, dando así una solución al problema propuesto de visualización. Se estaría en capacidad de generar datos para mejorar la capacidad competitiva del país.

Alternativa 5:
Se decide realizar una plataforma amplia que utilice como herramienta Google Maps, dado que esta plataforma provee una solución más cómoda para cualquier dispositivo, por lo que se utilizará esta herramienta de ubicación con el objetivo de visualizar los diferentes puntos

donde se encuentran ubicados los grupos de investigación en las diferentes ciudades y regiones. Además que se puede acoplar con los demás requerimientos en un programa conjunto que cumpla con los datos recibidos y el trato que estos deben tener.

Paso 4. Transición de las Ideas a los Diseños Preliminares.

Lo primero que haremos en este paso es descartar las soluciones que se considere que no son factibles. En este caso descartamos la *Alternativa 3* debido a que es muy poco factible que todos los grupos de investigación puedan realizar los viajes para al congreso que se realice cada cierto tiempo, también porque no podríamos tener la información a tiempo real de los grupos, y por último se sabe que realizar un evento de esa magnitud y guardar estos reportes podría resultar muy costoso. También descartamos la *Alternativa 1* debido a la alta complejidad que conlleva construir un croquis que represente el mapa de Colombia, junto con sus limitaciones por regiones Por esas razones se decide eliminar estas alternativas de las posibles soluciones.

A partir de una revisión cuidadosa de las otros alternativas, se saca saca lo siguiente:

Alternativa 2:

- Waze puede ser una gran herramienta de ubicación vial , pero a la hora de consultar posibles ubicaciones, la aplicación es infructuosa con respecto a lo que se necesita, dado que la manera de representación que tiene el software no permite mostrar de mejor manera los requerimientos y añade información innecesaria como el tráfico u otras con respecto a las vías que conectan a dos grupos. Esto torna la funcionalidad de la aplicación un poco vana para el objetivo principal de la solución del problema.

Alternativa 4:

- Apple Maps, es una buena herramienta de ubicación, pero al ser desarrollada para el sistema operativo IOS, limita la funcionalidad de esta aplicación a sólo los sistemas operativos que cuenten con la posibilidad de tener este programa.

Alternativa 5:

- Google Maps, brinda una gran diversidad de servicios, es una excelente herramienta la cual es posible ejecutarla tanto en dispositivos Android, Apple, entre otros. Además tiene una interfaz bastante agradable, permitiendo visualizar los mapas con diferentes perspectivas (predeterminado, vista satélite y relieve).

Paso 5. Evaluación y selección de la mejor solución.

Criterios

Deben definirse los criterios que permitirán evaluar las alternativas de solución y con base en este resultado elegir la solución que mejor satisface las necesidades del problema planteado. Los criterios que escogimos en este caso son los que enumeramos a continuación. Al lado de cada uno se ha establecido un valor numérico con el objetivo de establecer un peso que indique cuáles de los valores posibles de cada criterio tienen más peso, es decir cuanto mas puntaje tenga sera el adecuado para realizar la solución.

- *Criterio A*. Eficacia. Se requiere una plataforma la cual cuente con una herramienta de mapas lo suficientemente compleja para abarcar los requerimientos necesarios.
  Presenta una herramienta con un nivel de eficacia:
  - [4] Muy alto (Brinda más de 3 servicios)
  - [3] Alto (Brinda entre 2 y 3 servicios)
  - [2] Medio (Brinda entre 1 y 2 servicios)
  - [1] Bajo (Solo permite visualizar el mapa)

- *Criterio B.* Facilidad en implementación algorítmica. La alternativa cuenta con un nivel de dificultad que permita al desarrollador generar la solución de la mejor manera con la implementación de menor cantidad de recursos informáticos para así mejorar el rendimiento del programa.
  {Entendiéndose recurso informático como plataformas, plug-in, repositorios, bases de datos etc.}
  - [1] Difícil. Se requiere que el desarrollador utilice una gran cantidad de recursos (mayor a 10) para el desarrollo de la solución.
  - [2] Neutral. Se requiere que el desarrollador utilice una cantidad suficiente (entre 3 a 10) de recursos para desarrollar la solución.
  - [3] Fácil. No se requiere que el desarrollador utilice una cantidad mayor de recursos (menos de 3) para desarrollar la solución.
  -
- *Criterio C.* Costo Monetario. El software debe realizarse con el menor costo monetario que se pueda obtener.
  - [1] Requiere un gran costo monetario.
  - [2] Requiere un costo monetario moderado.
  - [3] Requiere poco costo monetario.

- *Criterio D.* Compatibilidad. La plataforma debe tener la posibilidad de poder ejecutarse en diferentes dispositivos o sistemas operativos.
    - [3] Compatible en más de 3 sistemas
    - [2] Compatible entre 1 y 2 sistemas
    - [1] Compatible solo con un sistema

- *Criterio E.* Consumo de MB. La plataforma al usar una herramienta de mapas, es claro que necesita consumir una cantidad de MB por el tiempo en el que se encuentre en uso. La plataforma consume:
    - [1] 7 mb o mayor
    - [2] 5-7 mb
    - [3] 2- 5 mb
    - [4] 2 mb o menos

Evaluación

Según la evaluación con los criterios anteriores en las alternativas se obtiene la siguiente tabla:

	Criterio A Eficacia	**Criterio B Facilidad de Implementación Algorítmica**	**Criterio C Costo Monetario**	**Criterio D Compatibilidad**	**Criterio E Consumo MB**	**Total**
Alternativa 2	Alto (Brinda entre 2 y 3 servicios) [3]	Difícil. [1]	Requiere un costo monetario moderado [2]	Compatible entre 1 y 2 sistemas [2]	2-5 mb [3]	**11**
Alternativa 4	Medio (Brinda entre 1 y 2 servicios) [2]	Difícil [1]	Requiere un gran costo monetario [1]	Compatible solo con un sistema [1]	2 mb o menos [4]	**9**
Alternativa 5	Muy alto (Brinda más de 3 servicios) [4]	Neutral [2]	Requiere poco costo monetario [3]	Compatible entre más de 3 sistemas [3]	5-7 mb [2]	**14**

Selección

En base a los criterios referenciados anteriormente, y la calificación que le dimos a cada una de las alternativas en base a estos criterios, tomamos la decisión de utilizar la alternativa número cinco (Alternativa 5), esto debido a la gran eficacia que tiene, y que cuenta con las mejores características para la resolución de este problema a pesar de consumir una gran cantidad de Megabytes.

Paso 6. Preparación de informes y especificaciones.

Especificación del Problema (en términos de entrada/salida)

Problema: Se requiere una plataforma que permita interactuar con los datos de los diferentes grupos de investigación en Colombia.

Entrada: Información abstraída de datos libres tomando como proveedor Gobierno Digital Colombia

Salida: Reporte de manera gráfica (presente en el mapa) de la información de los diferentes grupos de investigación.

Diagrama de Clases

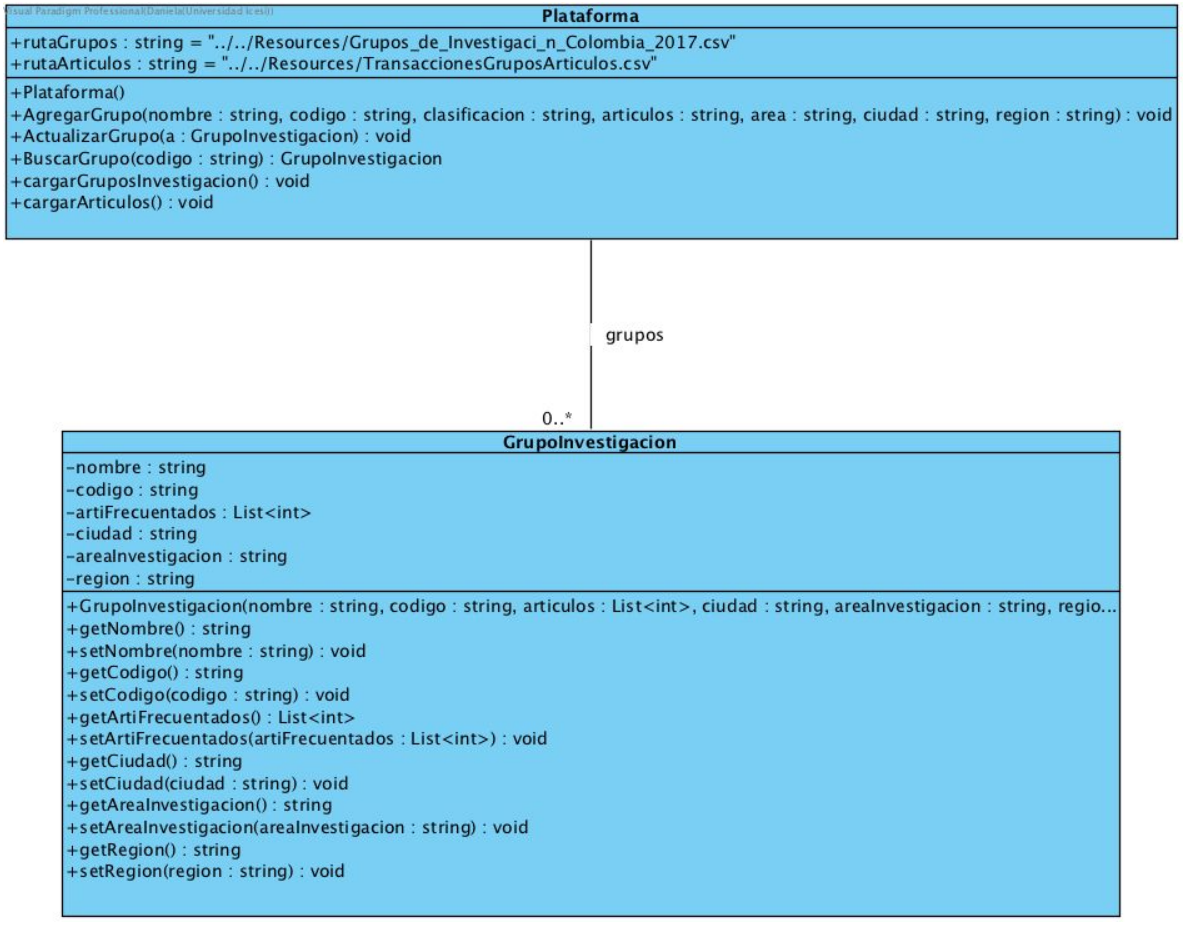
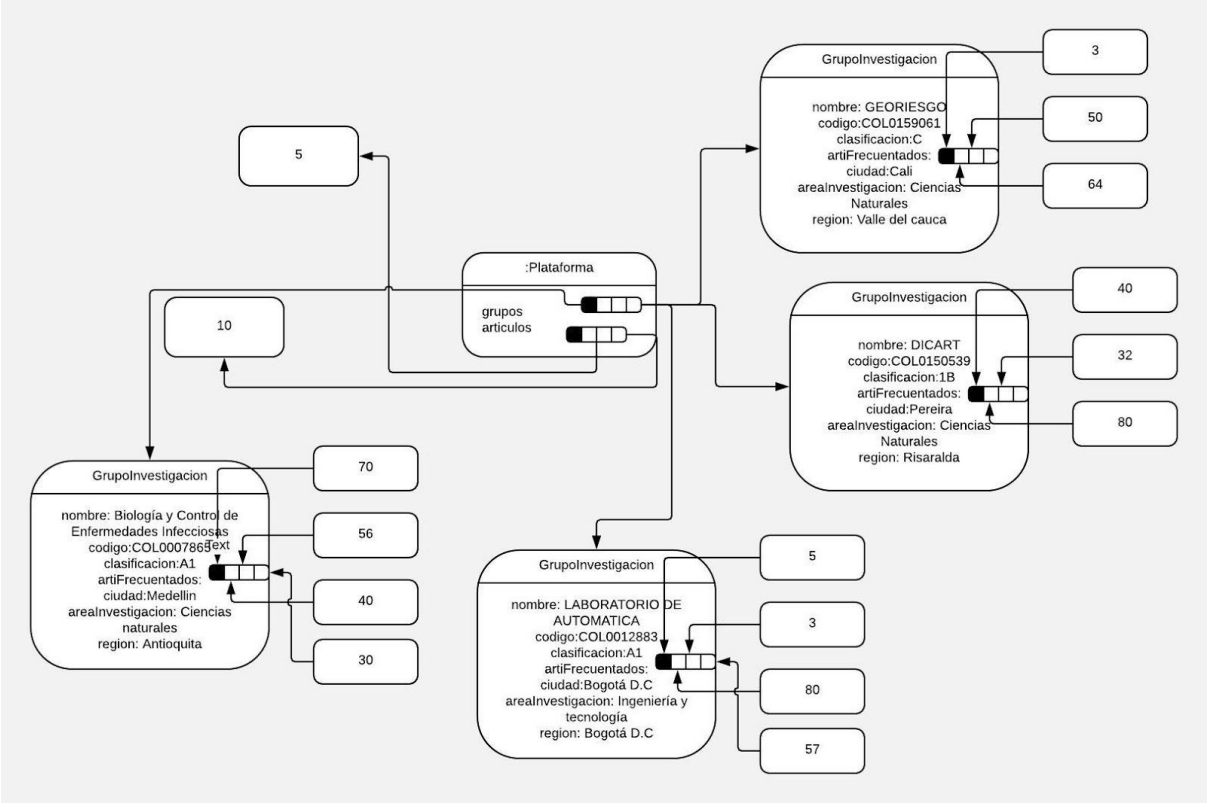


Diagrama de Objetos



Paso 7.Implementación del diseño

Implementación en el lenguaje de programación C# bajo la plataforma Visual Studio 2017

Lista de tareas a implementar:

- Implementación del modelo con sus diferentes clases, objetos, métodos, relaciones y atributos que ayuden a cumplir los requerimientos establecidos.
- Implementación de la interfaz gráfica con los mapas tomando como proveedor a Google maps.
- Implementación de los diferentes requerimientos de vista sobre el mapa
- Implementación de la funcionalidad de poder contabilizar la frecuencia en que los grupos obtienen acceden a diferentes artículos.

Construcción en el lenguaje C#:

```
public class Plataforma
   {
      public const String rutaGrupos =
"../../Resources/Grupos_de_Investigaci_n_Colombia_2017.csv";
      public const String rutaArticulos = "../../Resources/TransaccionesGruposArticulos.csv";
      private List<GrupoInvestigacion> grupos;

      public List<GrupoInvestigacion> Grupos { get => grupos; set => grupos = value; }

      public Plataforma()
      {
         Grupos = new List<GrupoInvestigacion>();
         cargarGruposInvestigacion();
         cargarArticulos();
      }

      public void AgregarGrupo(String nombre, String codigo, String clasificacion, String
articulos, String ciudad, String area, String region)
      {
        String [] listado = { nombre, codigo, clasificacion, ciudad, area, region };

         var lista = articulos.Split(' ').Select(i => Int32.Parse(i));
         List<int> art = new List<int>();
         List<int> listaNueva = art.Concat(lista).ToList<int>();
         GrupoInvestigacion nuevo = new GrupoInvestigacion(nombre, codigo, clasificacion,
listaNueva , ciudad, area, region);
            Grupos.Add(nuevo);



      }
      public void ActualizarGrupo(GrupoInvestigacion a)
      {
            for (int i = 0; i < grupos.Count; i++)
            {
               if (a.Codigo.Equals(grupos[i])) grupos[i] = a;
            }
      }
      public GrupoInvestigacion BuscarGrupo(String codigo)
      {
         GrupoInvestigacion nuevo = null;
         nuevo = Grupos.Find(i => i.Codigo.Equals(codigo));
         return nuevo;
      }
      public void cargarGruposInvestigacion()
```

```
{
    String line;
    try
    {
        StreamReader sr = new StreamReader(rutaGrupos);
        line = sr.ReadLine();
        while ((line = sr.ReadLine()) != null)
        {
            String[] todo = line.Split(',');
            String nombre="";
            String codigo = todo[2];
            String clasi = "";
            String ciudad = "";
            String area = "";
            String region = "";
            int contador = 3;
            if (todo[contador].StartsWith("\""))
            {
                while (!todo[contador].EndsWith("\""))
                {
                    nombre += todo[contador];
                    contador++;
                }
            }
            else
            {
                nombre = todo[contador];
                contador++;
            }
            if (todo[contador + 1].StartsWith("\""))
            {
                ciudad = "Bogotá, D. C.";
                region = "Bogotá, D. C.";
                contador += 2;
            }
            else
            {
                ciudad = todo[contador + 1];
                region = todo[contador + 2];
            }
            contador += 6;
            if(contador>= todo.Length)
            {
                clasi = "No especificada";
                area = "No especificada";
            }
```

```
                else
                {
                    clasi = todo[contador];
                    if(contador + 1>= todo.Length) area = "No especificada";
                    else area = todo[contador+1];
                }
                if (ciudad.Contains("/")) ciudad = "Colombia";
                AgregarGrupo(nombre, codigo, clasi, "-1 -1", ciudad, area, region);
            }
            sr.Close();
        }
        catch(Exception E)
        {
            throw new Exception(E.Message);
        }
    }


    public void cargarArticulos()
    {
        String line;
        try
        {
            StreamReader sr = new StreamReader(rutaArticulos);
            while ((line = sr.ReadLine()) != null)
            {
                String[] todo = line.Split(':');
                List<int> arti = new List<int>();
                String[] articulos = todo[2].Split(' ');
                articulos.ToList().ForEach(i => arti.Add(i.Equals("")? -1: Convert.ToInt32(i)));
                BuscarGrupo(todo[1]).ArtFrecuentados = arti;
            }
            sr.Close();
        }
        catch(Exception E)
        {
            throw new Exception(E.Message);
        }

    }

}


public class GrupoInvestigacion
{
```

```
    //-------------------------------------------------------------------------------------------------------------
    //Atributos
    //-------------------------------------------------------------------------------------------------------------
    // Nada
      private String nombre;
      private String codigo;
      private String clasificacion;
      private List<int> artiFrecuentados;
      private String ciudad;
      private String areaInvestigacion;
      private String region;

     //-------------------------------------------------------------------------------------------------------------
     //Constructor
     //-------------------------------------------------------------------------------------------------------------
      public GrupoInvestigacion(String nombre, String codigo ,String clasificacion, List<int>
articulos, String ciudad, String areaInvestigacion, String region){
          this.nombre = nombre;
          this.codigo = codigo;
          this.clasificacion = clasificacion;
          artiFrecuentados = articulos;
          this.ciudad = ciudad;
          this.areaInvestigacion = areaInvestigacion;
          this.region = region;
      }
    //-------------------------------------------------------------------------------------------------------------
    //Metodos
    //-------------------------------------------------------------------------------------------------------------
      public String Nombre
      {
          get
          {
              return nombre;
          }
          set
          {
              nombre = value;
          }
      }
      public String Codigo
      {
          get
          {
              return codigo;
          }
          set
```

```
    {
        codigo = value;
    }
}
public String Clasificacion
{
    get
    {
        return clasificacion;
    }
    set
    {
        clasificacion = value;
    }
}
public List<int> ArtFrecuentados
{
    get
    {
        return artiFrecuentados;
    }
    set
    {
        artiFrecuentados = value;
    }
}
public String Ciudad
{
    get
    {
        return ciudad;
    }
    set
    {
        ciudad = value;
    }
}
public String AreaInvestigacion
{
    get
    {
        return areaInvestigacion;
    }
    set
    {
        areaInvestigacion = value;
```

```
        }
    }
    public String Region
    {
        get
        {
            return region;
        }
        set
        {
            region = value;
        }
    }
}
```