

**INFORME DE INGENIERÍA
TALLER DE APLICACIÓN GMAP**

**MARIA CAMILA LENIS RESTREPO
JAVIER ANDRÉS TORRES REYES**

PROYECTO INTEGRADOR I

2019-1

INFORME DE INGENIERÍA

Identificación del problema

Definición del problema:

Debido a la curiosidad despertada por los eventos sísmicos que se han presentado en las últimas semanas, se desea poder conocer la información de los últimos terremotos ocurridos en todo el mundo.

Requerimientos funcionales:

1. Se debe mostrar información del terremoto tal como: lugar, longitud, latitud, magnitud, fecha y hora.
2. Se debe visualizar de manera gráfica un mapamundi con marcadores en los lugares donde han ocurrido terremotos.
3. Se debe actualizar la información sobre los terremotos cada que se inicia la aplicación.

Requerimientos no funcionales:

- El aplicativo debe estar implementado en Visual Studio. Net

Recopilación de la información

Terremoto y sus características:

Los terremotos son movimientos bruscos de tierra que se originan por la liberación de energía producida cuando chocan o se rozan las placas tectónicas de la Tierra que se encuentran en desplazamiento. (UDC).

Clasificación:

La escala más utilizada para clasificar terremotos es la escala de Richter a partir de lo que se ha registrado en el sismógrafo. Según esta, es considerado terremoto a partir de una magnitud mayor o igual a 4 (Geoencilopedia). ***Por lo tanto, para efectos de la aplicación serán considerados terremotos los sismos de grado mayor a 4.***

Características relevantes:

Tomando como base los datos abiertos de Colombia, a la hora de guardar el registro de un sismo incluyen datos como: fecha, hora, tipo de sismo, latitud, longitud, profundidad, magnitud, agencia, municipio, departamento entre otras. (SGC)

Por otro lado, según el Data Set “Significant Earthquakes, 1965-2016” (USGS, 2017) se registran datos tales como: fecha, hora, latitud, longitud, tipo, profundidad, entre otros.

Tomando en cuenta ambos registros de datos, se concluye que para el aplicativo se debe mostrar información de: ***lugar, longitud, latitud, magnitud, fecha y hora.***

Sobre datos abiertos:

La organización Open Knowledge International (2014) define los datos abiertos en su versión 2.1 de la Definición de Conocimiento Abierto de la siguiente manera: “El conocimiento es abierto si cualquiera es libre de acceder a él, usarlo, modificarlo y compartirlo, estando sujeto a lo sumo a medidas que preserven su autoría y su apertura.” En esta definición de conocimiento, se incluyen datos tanto científicos, históricos, geográficos o de cualquier otro tipo.

Numerosos servicios incluyen datos abiertos sobre terremotos, siendo ejemplo de esto el SGC (Servicio Geológico Colombiano), la web de datos abiertos de Colombia, la web de datos abiertos de México y el USGS (Servicio Geológico de los Estados Unidos). Este último destaca en que presenta una herramienta para desarrolladores (una API) que permite obtener información en tiempo real sobre sismos, filtrando por fecha, magnitud, ubicación, entre otros. Esta información se presenta en diversos formatos como JSON y XML (USGS, 2015).

API y JSON

Una API (Application Programming Interface) es un conjunto de especificaciones que permiten la comunicación entre programas diferentes. Se puede considerar similar a la manera en que la interfaz de usuario facilita la interacción de los usuarios con el software. Uno de sus principales beneficios es que permite usar funciones ya existentes en otra aplicación, evitando reinventar la rueda (Merino, 2014). Gran parte de las API cuya funcionalidad reside en presentar información solicitada (por ejemplo, Servicio Geológico de los Estados Unidos) lo hacen en formato JSON.

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato para intercambiar datos. Se caracteriza por ser simple de leer y escribir para las personas, y fácil de interpretar y generar para las máquinas. Presenta dos tipos de estructuras básicas: pares llave-valor y arreglos de datos. A su vez, se pueden agrupar varias de estas estructuras básicas en otras más complejas llamadas objetos (JSON.org, 2006).

Un ejemplo básico de JSON es el siguiente, recuperado de Wikipedia (Desconocido):

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        {
          "value": "New", "onclick": "CreateNewDoc()"
        }, {
          "value": "Open", "onclick": "OpenDoc()"
        }
      ]
    }
  }
}
```

```

        "value": "Close", "onclick": "CloseDoc()"
    }
}
}
}
}

```

Para poder aprovechar las API en C#, primero es necesario comunicarse con ellas. Esto se realiza mediante peticiones HTTP. Microsoft (2018) presenta un ejemplo de cómo realizar peticiones a páginas web con C#, que luego se puede adaptar a las necesidades del problema:

```

using System;
using System.Net;
using System.IO;

namespace MakeAGETRequest_charp
{
    /// <summary>
    /// Descripción de resumen para Class1.
    /// </summary>
    class Class1
    {
        static void Main(string[] args)
        {
            string sURL;
            sURL = "http://www.microsoft.com";

            WebRequest wrGETURL;
            wrGETURL = WebRequest.Create(sURL);

            WebProxy myProxy = new WebProxy("myproxy",80);
            myProxy.BypassProxyOnLocal = true;

            wrGETURL.Proxy = WebProxy.GetDefaultProxy();

            Stream objStream;
            objStream = wrGETURL.GetResponse().GetResponseStream();

            StreamReader objReader = new StreamReader(objStream);

            string sLine = "";
            int i = 0;

            while (sLine!=null)
            {

```

```

        i++;
        sLine = objReader.ReadLine();
        if (sLine!=null)
            Console.WriteLine("{0}:{1}",i,sLine);
    }
    Console.ReadLine();
}
}
}

```

Una vez se ha obtenido la información en formato JSON, es necesario deserializarla para poder usar sus campos en clases creadas por el desarrollador en C#. En StackOverflow, Pavlov (2013) propone usar la librería Json.NET, dando el siguiente ejemplo, que luego se puede adaptar al problema:

```
dynamic stuff = JsonConvert.DeserializeObject("{ \"Name\": \"Jon Smith\", \"Address\": { \"City\": \"New York\", \"State\": \"NY\" }, \"Age\": 42 }");
```

```
string name = stuff.Name;
string address = stuff.Address.City;
```

En el ejemplo anterior, la variable name tendrá el valor “Jon Smith”, y la variable address será un objeto al que se le pueden pedir los campos City y State.

Sobre herramientas para visualización de mapas:

Una de las herramientas más usadas en C# para visualizar mapas es GMap.NET, que permite no solo visualizar mapas, sino también añadir información sobre estos como marcadores y polígonos. Los marcadores serán útiles para visualizar los lugares exactos donde ocurrieron los terremotos. A continuación, se presenta un ejemplo de cómo usar estos marcadores en C#, dado por la web Independent Software (2016), y que luego se puede adaptar al problema.

```

gmap.MapProvider = BingHybridMapProvider.Instance;

GMaps.Instance.Mode = AccessMode.ServerOnly;

gmap.SetPositionByKeywords ("Paris, France");

gmap.ShowCenter = false;

GMapOverlay markers = new GMapOverlay("markers");

GMapMarker marker = new GMarkerGoogle(

    new PointLatLng(48.8617774, 2.349272),

```

```
GMarkerGoogleType.blue_pushpin);

markers.Markers.Add(marker);

gmap.Overlays.Add(markers);
```

Un acercamiento al aplicativo:

Existen varias aplicaciones web que son parecidas a lo que se quiere lograr con este aplicativo. Una de ellas es la del Servicio Geológico Colombiano que luce de la siguiente manera:



Imagen tomada de la interfaz de la pagina del Servicio Geológico Colombiano. Link: <https://www2.sgc.gov.co/sismos/sismos/ultimos-sismos.html>

En la esquina inferior izquierda aparece una lista con la información principal, tal como magnitud, una descripción breve y profundidad. Al momento en el que se hace clic en algún registro, la parte superior cambia y muestra más información acerca del terremoto seleccionado. En el centro de la ventana se encuentra el mapa de Colombia con marcadores en los lugares donde hubo actividad sísmica reciente; si se hace clic en alguno de los marcadores se señala en la tabla el sismo al cual corresponde, y en la parte superior se muestra la información completa. En la parte derecha se encuentran funcionalidades de filtro y consulta más avanzadas que se salen del alcance del aplicativo que se está desarrollando. Cabe resaltar que el mapa permite acercarse o moverse sobre él como el usuario lo desee.

Búsqueda de soluciones creativas

Las funcionalidades del aplicativo pueden ser divididas en dos: las relacionadas con la visualización de mapas y las relacionadas con la obtención y el tipo de información sobre los terremotos. Por lo tanto, se han propuesto las siguientes soluciones:

Para la visualización de mapas:

Se decidió realizar una *lluvia de ideas* para encontrar la solución a la visualización de los mapas, de manera que sea de fácil comprensión visual y permita hacer un marcador en el lugar donde ha ocurrido un terremoto. Las ideas son las siguientes:

1. Pintar un mapamundi y dibujar los puntos donde han ocurrido los terremotos.
2. Implementar la funcionalidad de GMaps, para usar los mapas reales y su funcionalidad de agregar marcadores.
3. Implementar un grafo para organizar los terremotos, donde la conexión entre ellos sea la distancia en la que se encuentran unos de otros y pintarlo en la pantalla.
4. Guardar por cada terremoto un mapa del lugar donde ocurrió y mostrarlo cuando se seleccione dicho terremoto.
5. Permitir que el usuario seleccione un terremoto y que se abra la página de Google Maps en las coordenadas del terremoto.
6. Organizar los terremotos de acuerdo a sus coordenadas y poner al lado un mapamundi para que el usuario encuentre el punto de ubicación.
7. Dibujar un mapamundi y permitir que el usuario al hacer clic añada el marcador, el cual también será dibujado.

Para la obtención de la información sobre los terremotos:

Se decidió realizar una *lista de atributos* que deben cumplir los datos abiertos:

- Deben ser actualizados
- Deben contener información acerca de: lugar, longitud, latitud, magnitud, fecha y hora.
- Deben ser de alcance mundial, es decir debe contener la información de los últimos terremotos alrededor del mundo, no solo de un país en específico.
- Deben poderse leer por la aplicación.

Siguiendo estas especificaciones, las opciones que podrían servir para la obtención de la información son:

1. Registrar la información obtenida de los periódicos y guardarla en formato .txt
2. Obtener los datos de Earthquake USGS (actualizable)
3. Descargar un base de datos con los registros de los últimos terremotos en formato .xls
4. El usuario registra los terremotos ocurridos desde la aplicación, y se guardan de manera persistente en archivos .txt

Transición de ideas a los diseños preliminares

Para la visualización de mapas:

Después de haber realizado la lluvia de ideas y haber repasado los requerimientos quedan descartadas las ideas 3, 6 y 7 por los motivos expuestos a continuación:

- Lo que se busca con el requerimiento de visualización de los mapas es que el usuario comprenda mejor donde ocurrieron; poner al usuario a adivinar en que punto de mapa va una coordenada para añadir el marcador no hace parte de las funcionalidades de la aplicación debido a que se espera que sea una información verídica, es decir, tanto la información mostrada como texto como las imágenes deben exponer fielmente el lugar donde ocurrió. Por lo tanto, la idea 3 daría una información incompleta porque solo tomaría en cuenta la distancia entre un terremoto y otro, pero no, como tal, su ubicación en el mapa; en este punto si todos los terremotos ocurrieron en el Medio Oriente la imagen parecería mostrar un conjunto de aristas que corresponden a todo el planeta, cuando solo se está mostrando una parte de este.
- En la idea 6 y 7 se espera que el usuario busque el lugar donde ocurrieron los terremotos para que después se ubique en el mapa, o bien añada un marcador. Aunque estas ideas tengan una perspectiva más real al mostrar el mapa completo, existen factores de riesgo por errores humanos; no se puede suponer que el usuario conoce las coordenadas, y aunque conociera de ellas los dígitos de cada una son casi seis, lo que podría ocasionar diferencias de interpretación entre un usuario a otro. Se considera que debe ser la aplicación la encargada de mostrar la ubicación, no que el usuario la encuentre, para así ser fiel a la información real.

Luego de haber descartado las ideas no factibles se explicará más a fondo en qué consiste cada una, para así proceder a la evaluación y selección de la mejor solución.

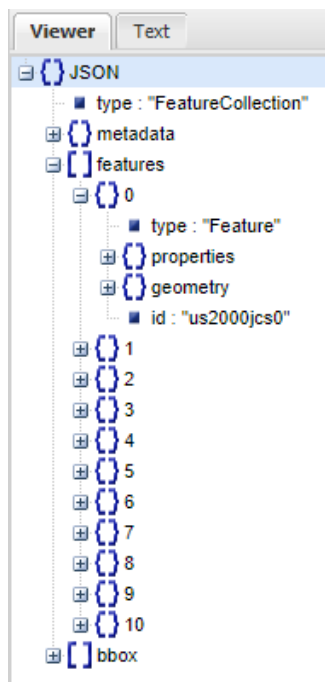
- **Idea 1:** En un panel si dibujará un mapamundi -puede ser una imagen simplemente a escala- y, de acuerdo a esa escala, por cada terremoto se tomará su longitud y latitud para ubicar un punto en la imagen y dibujar un círculo dicha coordenada.
- **Idea 2:** Visual Studio puede implementar una librería llamada GMaps que permite, por medio de un proveedor de mapas, mostrar el mapa completo, desplazarse sobre él, ubicarse en las coordenadas deseadas y agregar marcadores en una coordenada específica. Esta funcionalidad permite que el proveedor de mapas sea Google Maps, por lo que tendría varios modos de visualización ya sea de calles o la vista el satélite.
- **Idea 4:** De acuerdo a los datos guardados del terremoto, guardar también un mapa el país donde ocurrió y así, cuando el usuario de clic en el terremoto, se muestre la imagen del país donde ocurrió.
- **Idea 5:** El aplicativo solo mostrará la información como un texto, pero si el usuario desea ver el mapa, podrá presionar un botón que de acuerdo a las coordenadas el terremoto abrirá <https://www.google.com/maps/@3.4779323,-76.5199193,15z> donde los números después del @ serán reemplazados por las coordenadas, lo que lo ubicará en el punto donde ocurrió el terremoto.

Para la obtención de la información sobre los terremotos:

Así como se buscaba aislar al usuario de modificar el punto en el que se encuentran los terremotos en la funcionalidad anterior, así se decide aislar al usuario de ingresar la información de los terremotos. Primero, no hace parte de los requerimientos del programa, de hecho, lo que busca el programa es todo lo contrario, que el usuario no tenga que buscar en otras fuentes dónde han ocurrido, sino que directamente en el aplicativo se de cuenta de dónde han ocurrido. Por otra parte, la información mostrada debe ser actualizada, y si se guarda de manera persistente los terremotos, no se sabrá cuando se convierte en una información vieja y deba ser removida. Por lo tanto, la idea 4 queda descartada.

Luego de haber descartado las ideas no factibles se explicará más a fondo en qué consiste cada una, para así proceder a la evaluación y selección de la mejor solución.

- **Idea 1:** Los administradores de la aplicación buscarán la información actualizada en los noticieros o periódicos y añadirán los registros a un archivo .txt de cual tienen completo poder de administración y pueden borrar los registros antiguos.
- **Idea 2:** Earthquake USGS es un API de los Estados Unidos que permite acceder libremente a la información sobre los últimos terremotos ocurridos alrededor del mundo. Mantiene en constante actualización, por lo tanto, no se descargará un único archivo, sino que cada que se abre la aplicación y esta se encuentre con conexión a internet se descargará el último reporte. El archivo que se descarga es json lo que permite organizar la información lógicamente de la siguiente manera:



Donde cada *feature* es un terremoto con sus propiedades a las cuales se podrá acceder desde la aplicación, utilizando así solo las que sean pertinentes, es decir: lugar, longitud, latitud, magnitud, fecha y hora.

- **Idea 3:** Se descargará un archivo .xls de datos abiertos de Colombia, para así tener los datos organizados en columnas con su respectivo indicador, donde cada registro es una fila.

Evaluación o selección de la mejor solución

Para la visualización de mapas:

Criterio A: Completa comprensión del espacio. Es decir, que la visualización permita desplazarse por el mapa, ver las distintas ciudades, enfocarse en un punto en particular y hacer zoom sobre un punto en el mapa.

- [3] Permite hacer todas las funcionalidades descritas anteriormente.
- [2] Permite visualizar las distintas ciudades y enfocarse en un punto.
- [1] Permite una visualización estática del mapamundi.

Criterio B: Lugar de visualización. La visualización del mapa se realiza dentro de la aplicación desarrollada.

- [3] Permite la visualización del mapa dentro de la aplicación
- [1] Usa otras aplicaciones externas para visualizar el mapa

Criterio C: Añadir marcadores. Permite añadir un marcador en un punto en específico dadas las coordenadas.

- [3] Lo permite
- [1] No lo permite, solo muestra el punto en específico.

Criterio D: Exactitud de los marcadores. El punto donde se añadieron los marcadores es exacto al de las coordenadas.

- [3] Se utilizan coordenadas exactas y el marcador esta ubicado en ese punto.
- [2] El marcador añadido está en un punto aproximado de las coordenadas.
- [1] No aplica, no permite añadir marcadores.

	Criterio A	Criterio B	Criterio C	Criterio D	Total
Idea 1	1	3	3	2	9
Idea 2	3	3	3	3	12
Idea 4	2	3	1	1	7
Idea 5	3	1	1	1	6

De acuerdo con la evaluación realizada, la mejor solución es usar la librería de GMaps, puesto que ella permite una completa comprensión del lugar en el que ocurrieron los terremotos, permite añadir marcadores y desplazarse por el mapa haciendo zoom o alejándose para así ver los distintos puntos y poder hacer clic sobre ellos para ver más información; todo desde la aplicación desarrollada, es decir, sin requerir salir de ella para visualizar los mapas, siendo esto cómodo para el usuario.

Para la obtención de la información sobre los terremotos:

Criterio A: Los datos son actualizados. Se asegura que los datos mostrados en la información son actualizados y permiten mostrar información sobre los últimos terremotos ocurridos alrededor del mundo.

- [3] Los datos corresponden a los últimos terremotos ocurridos alrededor del mundo
- [2] Los datos corresponden cercanamente a los últimos terremotos ocurridos alrededor del mundo
- [1] Los datos fácilmente pueden volverse obsoletos.

Criterio B: Modo de actualización. Los datos deben ser actualizados de manera automática, es decir sin intervención humana.

- [3] Los datos son actualizados automáticamente cada que se abre la aplicación
- [2] Los datos son actualizados en medida que el administrador los actualice.
- [1] Los datos no son actualizados, ya que se usa un archivo estático descargado durante el desarrollo de la aplicación.

Criterio C: Información completa. Se asegura que la información del terremoto tal como: lugar, longitud, latitud, magnitud, fecha y hora va a estar contenida.

- [3] Se asegura completamente
- [2] Podría estarlo
- [1] No podría estarlo

Criterio D: Alcance mundial. La información descargada corresponde con los últimos terremotos registrados alrededor del mundo

- [3] La información es de alcance mundial
- [2] La información es de alcance regional
- [1] La información es de alcance local

	Criterio A	Criterio B	Criterio C	Criterio D	Total
Idea 1	2	2	2	3	9
Idea 2	3	3	3	3	12
Idea 3	1	1	2	1	5

Por lo tanto, la idea más factible es utilizar el API de los Estados Unidos, para así descargar el último reporte cada que se inicie la aplicación y tener la información completa, actualizada y de fácil acceso dentro del desarrollo. Para esto se debe trabajar con archivos. json que fueron descritos en la recopilación de la información. Cabe resaltar que esto será posible en medida que el usuario este conectado a internet, de no estarlo usará el último reporte descargado.

Preparación de informes

Diagrama de clases

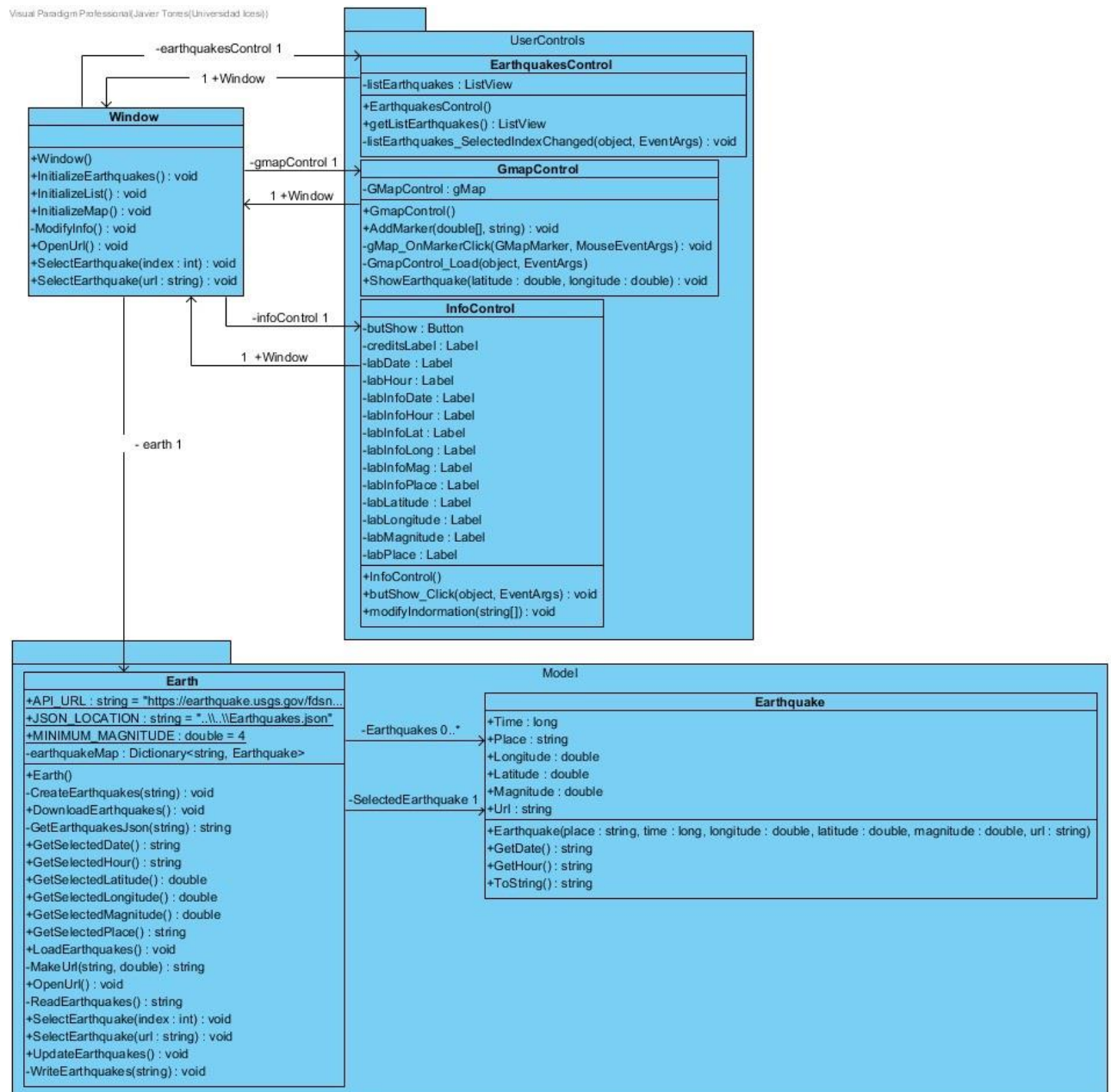
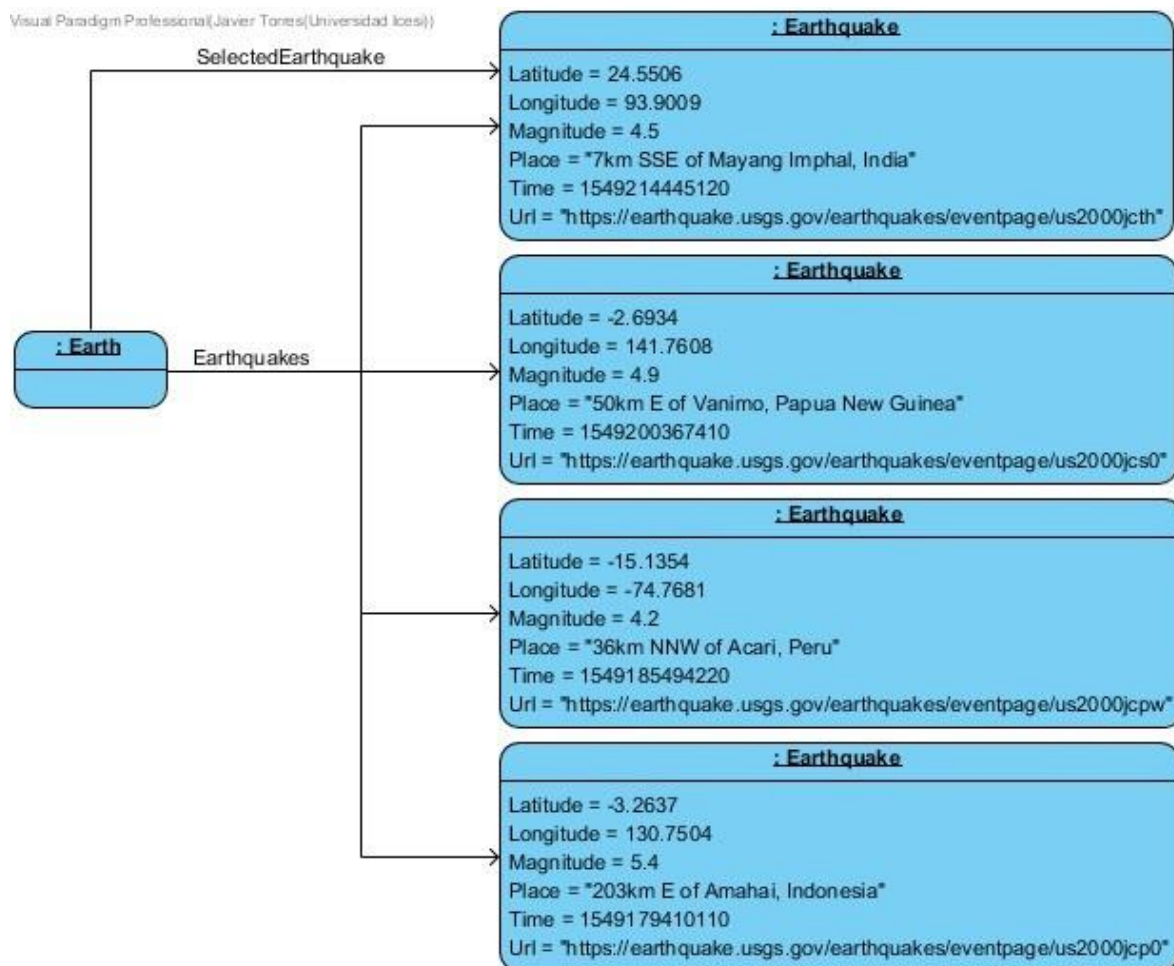


Diagrama de objetos



Síntesis reflexiva

Tras la realización del método de la ingeniería se encontró la mejor solución para todas las necesidades planteadas. El interés por los sismos en los últimos días llevó a orientar la aplicación a personas comunes, sin estudios en sismología o algún tópico cercano, sin embargo, se quiso que la comprensión fuera completa y la información actualizada. Durante el proceso de recopilación de información se encontraron otras aplicaciones que tenían el mismo propósito, lo que sirvió como mockup para el diseño de la interfaz. De la misma forma, se encontraron nuevas alternativas para cumplir con el propósito de llevar la información actualizada; para esto se hizo uso del Api de los Estados Unidos y se estudio el manejo de archivos JSON; esto último, hizo que se manejaran nuevas tecnologías que no se habían implementado en cursos anteriores tales como las peticiones https y los archivos JSON. Finalmente, se considera que hubo un buen manejo del proceso de selección de la mejor solución porque las alternativas elegidas para cada problema cumplieron por completo con los requerimientos y de forma efectiva con los recursos disponibles.

Bibliografía

Geoenciclopedia (Desconocido). Terremoto. Recuperado de <https://www.geoenciclopedia.com/terremoto/>

Independent Software (2016). GMAP.NET BEGINNERS TUTORIAL ADDING CLICKABLE MARKERS TO YOUR MAP (UPDATED FOR VS2015 AND GMAP.NET 1.7). Recuperado de: <http://www.independent-software.com/gmap-net-beginners-tutorial-adding-clickable-markers-to-your-map-updates-for-vs2015-and-gmap-1-7.html>

JSON.org (2006). Introducción a JSON. Recuperado de: <https://www.json.org/json-es.html>

Merino, M. (2014). ¿Qué es una API y para qué sirve? Recuperado de: <https://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/>

Microsoft (2018). CÓMO: Crear una solicitud GET mediante C# .NET. Recuperado de: <https://support.microsoft.com/es-co/help/307023/how-to-make-a-get-request-by-using-visual-c>

Open Knowledge International (2014). Definición de Conocimiento Abierto Versión 2.1. Recuperado de: <https://opendefinition.org/od/2.1/es/>

Pavlov D. (2013). How can I parse JSON with C#? Recuperado de: <https://stackoverflow.com/questions/6620165/how-can-i-parse-json-with-c>

UDC (Desconocido) ¿Qué es un terremoto? Recuperado de: https://www.udc.es/dep/dtcon/estructuras/ETSAC/Investigacion/Terremotos/QUE_ES.htm

USGS (2015). API Documentation - Earthquake Catalog. Recuperado de <https://earthquake.usgs.gov/fdsnws/event/1/>

USGS (2017) Significant Earthquakes, 1965-2016. Recuperado de <https://www.kaggle.com/usgs/earthquake-database>

Servicio Geológico Colombiano (2017). Reporte de Sismicidad del Servicio Geológico Colombiano-Año 2015. Recuperado de <https://www.datos.gov.co/Minas-y-Energia/Reporte-de-Sismicidad-del-Servicio-Geol-gico-Colom/c6z5-qfp4>

SGS (Desconocido) Últimos sismos en Colombia. Recuperado de <https://www2.sgc.gov.co/sismos/sismos/ultimos-sismos.html>

Wikipedia (Desconocido). JSON. Recuperado de: <https://es.wikipedia.org/wiki/JSON>