Instituto Tecnológico de Costa Rica

Programación Orientada a Objetos

Tarea Programada 1

Prof. Luis Pablo Soto

Estudiantes:

Fabricio Elizondo

Jose Andrés Navarro

Hans Fernández

Semestre2, 2016.

Fecha de Entrega 27/09/2016

Introducción

En este documento se incluye una detallada explicación acerca del funcionamiento de la primera tarea programada correspondiente al curso de programación orientada a objetos. El programa realizado cuenta con una interfaz gráfica de usuario donde se proporcionan diversas funcionalidades, que van a permitir a los usuarios finales de la aplicación, conocer de una forma bastante interactiva y agradable sobre diversos sismos que se producen en nuestro país.

Dentro de los requerimientos funcionales del programa, se incluye una interfaz gráfica de usuario que permitirá la interacción entre la aplicación y los usuarios finales de la misma. El programa va a permitir tanto la lectura como la escritura de archivos en formato csv, donde por medio de dichos archivos, se incluirán una serie de sismos en la interfaz gráfica para que las personas puedan informarse acerca de diversas características que estos presentan, incluyendo una imagen con su respectiva localización geográfica, y una serie de gráficas donde se muestra la información de los sismos para comodidad del usuario y un mayor entendimiento de los mismos.

Por otro lado, los usuarios van a contar con la posibilidad de modificar las características de algún sismo en específico, como, por ejemplo, en algún caso que se necesite cambiar las coordenadas de localización o para cambiar aspectos de magnitud del sismo, entre otros aspectos. Además, se incorpora la posibilidad de que las personas añadan o registren un nuevo sismo, brindando los datos requeridos, y el mismo será agregado en el archivo que funcionará como un respaldo de la información guardada.

Pensando en los usuarios que deseen ser informados en el momento en que ocurre un nuevo sismo para estar alertas ante cualquier eventualidad posible, el programa posee un segmento donde se incorpora un archivo, también en formato csv, con los datos personales de diversas personas que deseen ser informadas y el sistema les notificará de acuerdo a los datos proporcionados, la información y alerta del nuevo sismo ocurrido a través de los servicios de mensajería de texto y de correo electrónico.

A continuación, se presenta un índice que incluye los distintos segmentos sobre la documentación presentada.

Índice

* Especificación de la tarea
* Metodología del trabajo
* Cronograma de trabajo
* Diagrama de clases en UML
* Justificación de las relaciones entre objetos
* Análisis de resultados
* Enlace al JavaDoc
* Aspectos relevantes y lecciones aprendidas
* Bitácora de trabajo
* Bibliografía y fuentes digitales utilizadas
* Estatus de la entrega

Especificación de la tarea

Primer Proyecto Programado

## **¿Qué se busca con esta Tarea?**

* Practicar las habilidades de modelado de aplicaciones de software.
* Practicar la experimentación y la resolución de problemas a través de la propuesta de modelado y programación utilizando el paradigma orientada a objetos.
* Aumentar el conocimiento del estudiante sobre el lenguaje de programación Java.
* Ejercitar la toma de decisiones sobre el dominio del problema y del dominio de la solución.
* Fomentar la investigación por parte del estudiante en temas relacionados con:
  + Manipulación de un archivo Excel para lectura/escritura de información
  + Ubicación de un punto particular en un mapa.
  + Envío de correo electrónico y mensajes de texto SMS desde una aplicación Java.
* Crear un ambiente de interfaz gráfica de usuario en el lenguaje Java como mecanismo de interacción con el usuario.
* Fomentar el trabajo en equipo

## **El contexto y las funcionalidades esperadas**

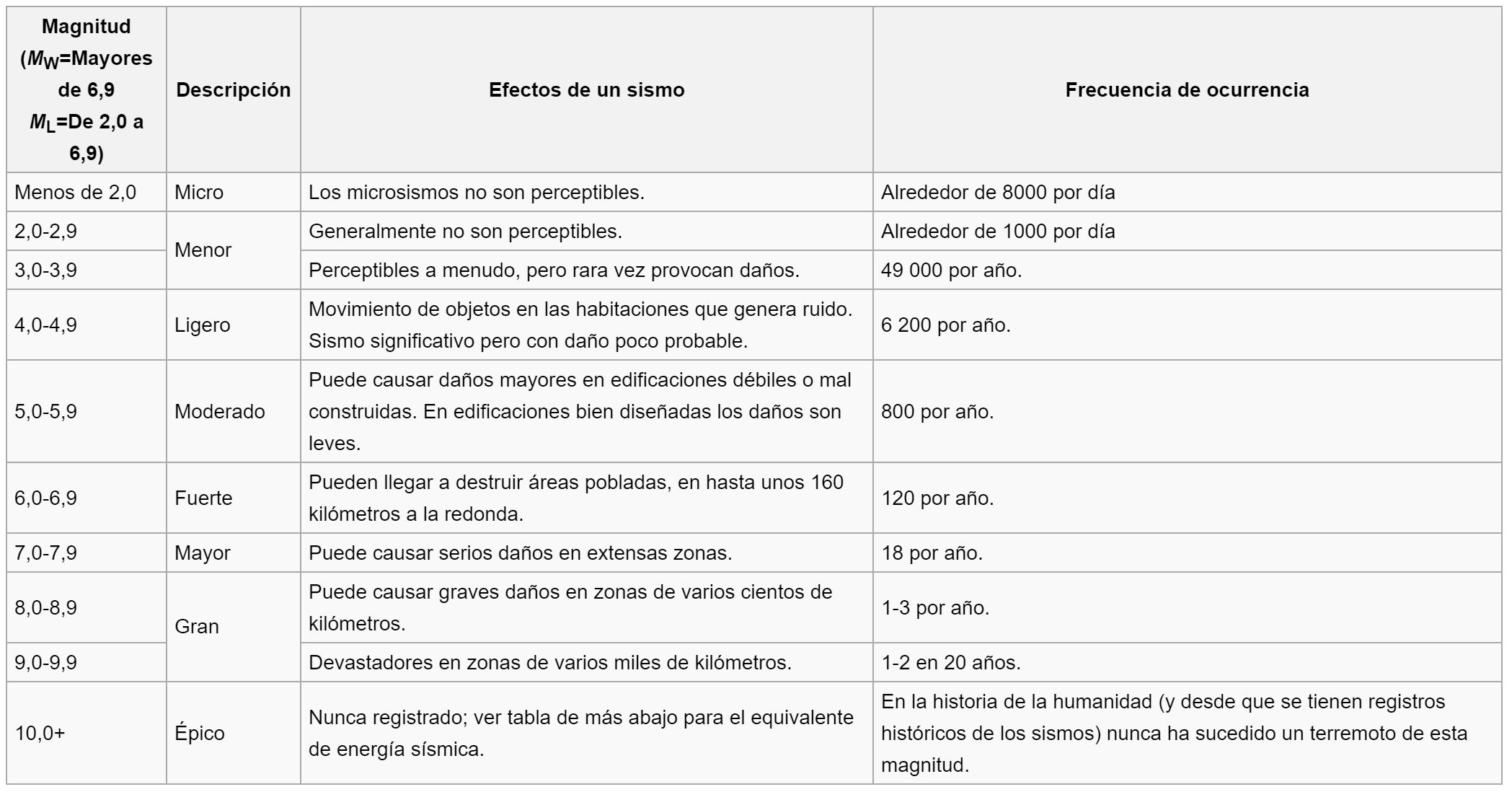
Se desea construir una aplicación utilizando el lenguaje de programación Java bajo el paradigma orientado a objetos que ofrezca información acerca de los sismos que suceden en el territorio nacional.

El principal interés es poder obtener información relevante respecto su comportamiento y poder realizar ciertos procesos de análisis de información que apoye en la toma de decisiones sobre programas de emergencia y prevención en las zonas donde hay mayor incidencia de este tipo de eventos.

De un sismo normalmente se tiene registro de información relevante como:

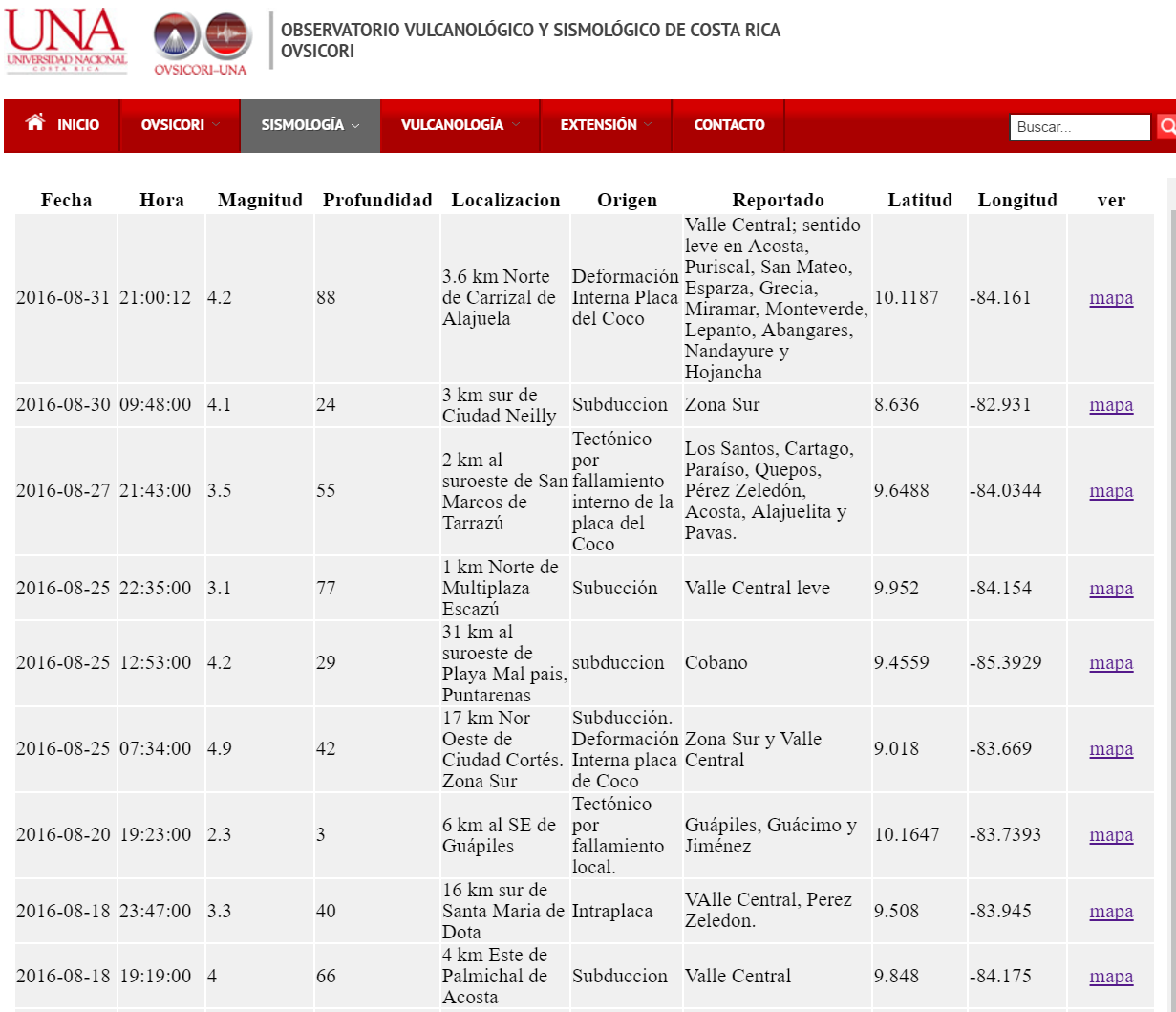
* Fecha en que ocurre (día, mes, año)
* Instante exacto en que ocurre (hora, minuto, segundo)
* Profundidad en kilómetros
* Origen de la falla:
  + Subducción
  + Choque de placas
  + Tectónico por falla local
  + Intra placa
  + Deformación Interna
* Magnitud: si la magnitud oscila entre 2.0 y 6.9 se indica según *la escala sismológica de medición local Richter (****ML****)*, si supera los 6.9 la escala de medición utilizada es la *escala sismológica de magnitud de momento* ***(MW).***
* Localización, la cual se indica en latitud y longitud, acompañada de una descripción detallada de la zona del epicentro. Es importante anotar que la latitud y longitud se expresa en grados y es REQUERIDO que la descripción detallada incluya la provincia donde se originó el epicentro. Debe considerarse además un sismo pudo haber sido originado en zona terrestre o marítima, en cuyo caso no tiene una provincia registrada por lo que se debe poder diferenciar entre estas dos condiciones.

Para su información, se adjunta un cuadro de características de los sismos de acuerdo al **valor de la magnitud** que se registre:



A continuación se aporta ejemplares de la información que se obtiene de las páginas oficiales de la Red Sismológica Nacional y el OVSICORI referentes a los últimos sismos sucedidos:





La aplicación debe desplegar de forma predeterminada una tabla mostrando el registro de sismos.

**Requerimiento 1: Mantenimiento a la información de los sismos registrados**

Al iniciar la aplicación, debe cargarse un archivo de Excel que contiene la información de los sismos que se han registrado con anterioridad. Dicha información debe mostrarse en un componente visual dentro de una ventana de la aplicación con el objetivo de poder conocer el detalle del sismo registrado. Así mismo se proveerá de un mecanismo para que una vez seleccionado un sismo particular, pueda visualizarse por medio de un mapa la ubicación geográfica exacta de su epicentro.

En esta misma ventana, debe ofrecerse la posibilidad de registrar un nuevo sismo para lo cual se requiere incluir todos los datos relevantes y al guardarlo deberá añadirse al archivo en Excel base en el que se almacenan todos los sismos registrados. Todos los datos del sismo son requeridos.

Además se puede modificar la información del sismo con el objeto de actualizar la posición geográfica u otros detalles de localización del epicentro.

Al registrarse el nuevo sismo se debe refrescar la zona donde se muestra la información de todos los sismos para reflejar el nuevo evento registrado.

Todos los cambios que se realicen mediante esta funcionalidad deben quedar almacenados en el archivo csv.

**Requerimiento 2: Análisis de información de los sismos registrados**

Una vez cargada la información de registro de los sismos, se desea se pueda ofrecer la siguiente información estadística en el formato que se solicita a continuación:

1. Cantidad de sismos por provincia Formato Histograma
2. Cantidad de sismos por tipo de origen Formato Gráfico de pastel
3. Sismos ocurridos en un rango de fechas Formato Tabular
4. Cantidad de sismos por mes en un año Formato Gráfico de barras
5. Clasificación de sismos por magnitud Formato Tabular

*(Para este último resumen debe considerar el cuadro*

*de clasificación por magnitud suministrado arriba)*

Debe considerar esta funcionalidad como parte del menú de reportes de la aplicación.

**Requerimiento 3: Servicio de notificación en provincias de interés**

Cargue un archivo de Excel (.csv) donde almacene los datos de varias personas que podrían estar interesadas en ser notificadas en el momento que un sismo suceda y sea registrado en la aplicación. De estas personas interesa el nombre, el correo electrónico, el celular y las provincias en las que pueda tener interés de ser notificado.

Al momento de registrar un nuevo sismo, la aplicación debe revisar la lista de interesados en ser notificados y enviarles un correo electrónico, un mensaje de texto o ambos según haya sido suministrados.

En el archivo no puede haber una persona sin las dos referencias de contacto, es decir, pueden estar las dos, o una de las dos pero no es posible que no tenga ambas, y debe tener al menos una provincia registrada.

El diseño de este archivo queda a decisión del equipo de trabajo.

Pruebe con los datos de los miembros del grupo y el profesor.

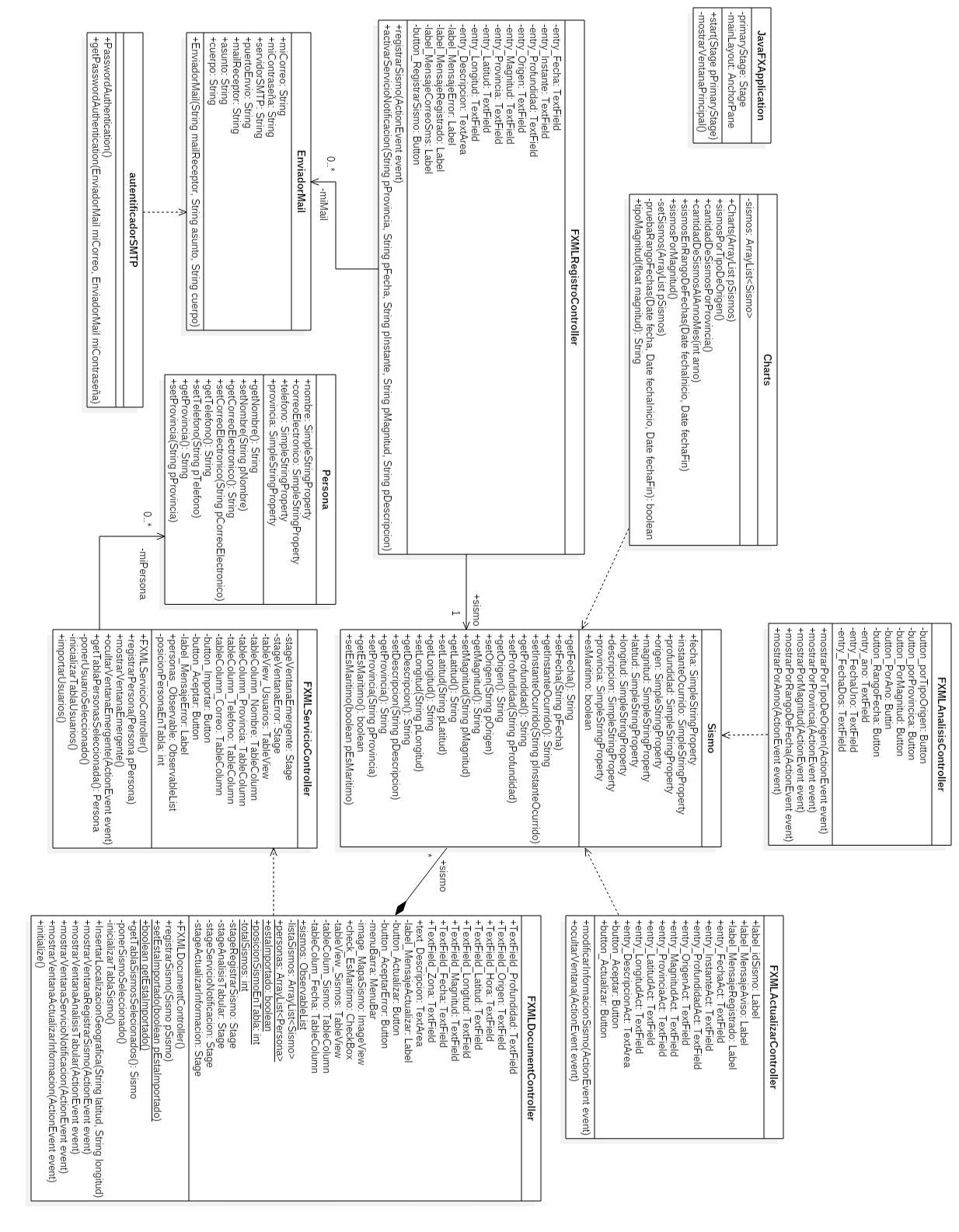
Para realizar la carga de este archivo debe incluir una opción en el menú.

Metodología de Trabajo

Para realizar esta tarea, se utilizó el lenguaje de programación Java, de acuerdo a lo solicitado en las indicaciones de la tarea. Para abordar el problema del modelo de la solución se decidió como grupo, reunirnos en los momentos en que se podía, y debido a las circunstancias, la mayor parte del trabajo se coordinó por medio de mensajes.

Se repartieron tareas (las cuáles se especifican en la bitácora y el cronograma de trabajo) de investigación como en el caso de las librerías que se tenían que utilizar para el programa, entre ellas: JavaFx para la interfaz gráfica de usuario, Mail para el envío de correos, API de Google Maps para la imagen que proporciona la localización geográfica del sismo, y el caso de la librería JFreeChart para el manejo y creación de las gráficas y la librería necesaria para el manejo de archivos csv. Además, se repartieron tareas de codificación donde cada uno de los integrantes del grupo realizaba una parte del programa y posteriormente se coordinaba por mensajes para agrupar las partes en un solo programa.

Diagrama de Clases en UML



Justificación de las relaciones entre objetos

- La composición de la clase FXMLDocumentController (Document) con la clase Sismo, es por el motivo de que la clase Document establece como atributo un ArrayList de los sismos que se registran en el archivo cvs, y que a la vez, es inicializada en el constructor de la clase Document, junto con los demás métodos que la necesitan. Por tales motivos, se estableció la relación de composición porque presenta contención física, por forma parte de la clase sismo.

- La dependencia de la clase FXMLDocumentController (Document) con la clase FXMLServicioController (Servicio) es por el motivo de que la clase Document solo necesita conocer la lista de personas registradas en la clase Servicio, con el fin de establecer una señal si la lista contiene elementos (objetos persona) y si ya ha sido leído el archivo csv de usuarios.

- La dependencia de la clase FXMLActualizarController (Actualizar) con la clase Sismo, se debe a que la clase Actualizar solo necesita conocer la lista de sismos que se han registrado en la aplicación con el fin de modificar los valores en la tabla (interfaz). No se estableció una relación todo/parte ya que, no fue posible implementar la función que modificara el archivo csv.

- La dependencia de la clase FXMLAnalisisController (Análisis) con la clase Sismo, se debe a que la clase Actualizar no tiene relación estructural con la clase Sismo, por lo tanto, la clase actualizar solo necesita conocer, en este caso, la lista de sismos. No se estableció una relación de todo/parte.

- La dependencia de la clase Charts con a clase Sismo, se debe a que la clase Charts no presenta relación estructural con Sismo, además de no presentar relación todo/parte porque no presenta atributos que estén relacionados en lo que a sismos se refiere. Por tanto, se consideró como una dependencia, ya que Charts solo necesita conocer los atributos de la clase Sismo.

- La asociación directa de la clase FXMLServicioController (Servicio) con la clase Persona se debe a que la clase Servicio tiene como referencia una lista de las personas registradas, con el fin tener una referencia de la clase Persona, por tanto, se implementa un método en la clase Servicio del cual hace uso de esta referencia.

- La asociación directa de la clase FXMLRegistroController (Registro) con la clase Sismo y con la clase EnviadorMail, se debe a que la clase Registro tiene como un atributo una referencia de la clase Sismo y de la clase EnviadorMail. Todo con el fin, de que la clase Registro obtenga la lista de las personas que han sido registradas, para luego activar el servicio de email (EnviadorMail) en el método correspondiente, caso de encontrar un mail disponible.

- Por último, la relación de dependencia de la clase autentificadorSMTP con la clase EnviadorMail, es por el motivo de que autentificadorSMTP necesita conocer solamente los atributos, correo y contraseña de la clase EnviadorMail. Por lo tanto, se implementó en el método de la clase los parámetros de tipo EnviadorMail.

Análisis de Resultados

El proyecto cumple sus funcionalidades principales. La única funcionalidad incluida entre sus principales que no fue implementada es la de Enviar SMS, aparte de algunas pulgas que no pudieron ser corregidas que son mínimas. Con esas excepciones mencionadas, el software funciona correctamente.

Enlace al JavaDoc

Aspectos relevantes y lecciones aprendidas

Entre las lecciones aprendidas durante este trabajo, cabe rescatar como punto fundamental, en el caso de las habilidades blandas, el trabajo en equipo, ya que tanto en el transcurso de nuestra carrera como en la vida laboral es de suma importancia aprender a trabajar en equipo. Y es que, en muchas ocasiones en la vida, se nos presentan retos o situaciones donde por más preparados que se encuentre uno como persona, siempre es necesaria la ayuda de otras personas, además que esto permite reducir el tiempo de trabajo individual, facilita la comunicación y permite mejorar la eficiencia del proceso en todos los aspectos.

Por otro lado, entre los aspectos más técnicos de la tarea, se mejoró mucho en el manejo del lenguaje de programación de Java por medio de la práctica constante y el uso de diversas librerías, lo cual es beneficioso tanto para este curso como para el resto de la carrera.

A manera de concluir, otra de las lecciones aprendidas es la mejora en el proceso de investigación, ya que todos los miembros del grupo, como parte de las tareas asignadas, en uno u otro momento tuvieron que investigar diversas fuentes tanto digitales como consultas a personas externas que manejaban un dominio del tema.

Bitácora de Trabajo

-Domingo 18 de Setiembre: se hizo una reunión para dividir las diferentes tareas para llevar el sistema a cabo. A Jose se le asignó el desarrollo de la interfaz gráfica, la funcionalidad de la interfaz gráfica, la clase sismos y la funcionalidad de guardar archivos, a Fabricio se le asignó la funcionalidad de localización y envío de correo, a Hans se le asignó el desarrollo de las gráficas y la funcionalidad de las gráficas. Además, se crea el primer prototipo del UML del sistema de software.

-Lunes 19 de Setiembre: Jose comienza la investigación de cómo utilizar interfaces gráficas y el desarrollo de la interfaz gráfica. Se termina el desarrollo gráfico de la interfaz.

-Martes 20 de Setiembre: Jose comienza y culmina el desarrollo de la clase Sismo.

-Jueves 22 de Setiembre: Fabricio comienza la investigación y desarrollo de la funcionalidad de localización.

-Sábado 24 de Setiembre: Hans comienza la investigación y desarrollo de las gráficas. Fabricio termina el desarrollo de la funcionalidad de localización y comienza la investigación y desarrollo de la funcionalidad de envío de correo.

-Domingo 25 de Setiembre: Fabricio termina el desarrollo de la funcionalidad enviar correo.

-Lunes 26 de Setiembre: Hans termina el desarrollo de las gráficas y su funcionalidad. Se implementan todas las funcionalidades de la interfaz gráfica. Se concluye el desarrollo del sistema de software. Jose realiza la versión final del UML del sistema. Hans y Fabricio comienzan a redactar la documentación del sistema de software.

-Martes 27 de Setiembre: Hans y Fabricio terminan de redactar la documentación del sistema de software. Se termina por completo la realización del proyecto.

Nota: los Javadocs se fueron realizando conforme se hacía la implementación de las funcionalidades.

Bibliografía y fuentes digitales utilizadas

API de Google Maps

<https://developers.google.com/maps/documentation/static-maps/intro>

Manipular URL y solicitudes HTTP con Java

<http://aljavier.github.io/manipulando-urls-y-haciendo-solicitudes-http-con-java.html>

Ejemplo del ImageView de JavaFx

<https://examples.javacodegeeks.com/core-java/javafx-imageview-example/>

Cargar Imágenes desde un URL

<http://www.java2s.com/Code/Java/JavaFX/LoadImagefromURL.htm>

API para el envío de correos desde Java

<http://www.oracle.com/technetwork/java/index-138643.html>

Guía para el envío de correos

<https://www.adictosaltrabajo.com/tutoriales/javamail/>

Ejemplo de uso de gráficas

<http://jonathanmelgoza.com/blog/como-hacer-graficos-con-java/>

Estatus de la entrega

Por la calidad del trabajo realizado y las funcionalidades llevadas a cabo, la calificación personal del proyecto es de: Bueno.