

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

(Computer Engineering Academic Area)

Programa de Licenciatura en Ingeniería en Computadores

(Licentiate Degree Program in Computer Engineering)



Programa de monitoreo en tiempo real de los sismógrafos del OVSICORI

(Monitoring program in real time of OVSICORI's seismographs)

**Informe de Trabajo de Graduación para optar por el título de Ingeniero en
Computadores con grado académico de Licenciatura**

(Report of Graduation Work in fulfillment of the requirements for the degree of Licentiate in Computer Engineering)

Alejandro José Vargas Chaves

Cartago, Noviembre, 2015

(Cartago, November, 2015)



TEC – Área Ingeniería en Computadores (CE) Acta de Aprobación de Trabajo de Graduación

Con fundamento en lo que establece el "Reglamento de Trabajos Finales de Graduación del Instituto Tecnológico de Costa Rica", el Tribunal Examinador del Trabajo Final de Graduación, nombrado con el propósito de evaluar el proyecto final de graduación.

"Programa de monitoreo en tiempo real de los sismógrafos del OVSICORI"

Habiendo analizado el resultado general del trabajo presentado por los estudiantes:

Primer Apellido	Segundo Apellido	Nombre	No. De carné
Vargas	Chaves	Alejandro	200932185

Emite el siguiente dictamen:

APROBADO CALIFICACION: <u>95</u> puntos.	<input type="radio"/> REPROBADO <input type="radio"/> SE RECOMIENDA <input type="radio"/> NO SE RECOMIENDA Brindarle una nueva oportunidad para la DEFENSA PUBLICA de su Trabajo Final NUEVA FECHA: _____
---	--

Dando fe de lo aquí expuesto firmamos

 M.Sc. Milton Villegas Lemus Profesor Lector	 Dra. Paola Vega Castillo Profesora Asesora	 Dr. Roberto Pereira Arroyo Profesor Lector
19 de Noviembre del 2015		

Resumen

Costa Rica tiene la particularidad de ser un país en el cual los movimientos sísmicos son muy frecuentes. Esto se debe a que el país está ubicado en el Cinturón de Fuego del Pacífico.

Para vigilar, analizar e interpretar todos los movimientos sísmicos y volcánicos que suceden en el país, se creó el Observatorio Vulcanológico y Sismológico de Costa Rica. De modo tal que es trabajo del OVSICORI divulgar a la sociedad el conocimiento obtenido para prevenir y mitigar los desastres que la actividad sísmica y volcánica generan.

Actualmente el OVSICORI carece de herramientas para poder mostrar toda la información sobre movimientos sísmicos a la población. Para proveer esta información dependen principalmente de su central telefónica, donde la información que se puede aportar por este medio es muy limitada. Es por ello que se ha establecido un vínculo con el proyecto iReal del Instituto Tecnológico de Costa Rica (TEC), con el cual se pretende desarrollar diversos proyectos para la visualización de datos sísmicos.

Aprovechando este vínculo entre universidades y con la intención de solucionar esta necesidad, se propone desarrollar un sistema de monitoreo de sismógrafos en tiempo real. La intención de este sistema es obtener datos desde las estaciones sismológicas de OVSICORI y mostrar esta información en su portal web a la población de una manera fácil de comprender. Esta herramienta pretende no solo obtener y registrar la información de un sismo, sino que además pretende elaborar una simulación de la evolución de la intensidad del sismo sentida en diferentes zonas del país.

Este documento detalla todo el proceso realizado para elaborar esta herramienta, el cual abarca principalmente los siguientes aspectos:

- Investigación sobre diferentes conceptos y herramientas de sismología
- Protocolos necesarios para obtener los datos desde las estaciones sismológicas del observatorio.
- Algoritmos para filtrar la señal y metodología para almacenar estos datos para ser mostrados por la aplicación.
- Herramientas web que permitan tanto la representación de sismogramas y la simulación de intensidades.
- Algoritmos de interpolación para elaboración de la simulación

Palabras claves: SIG, acelerógrafo, sismógrafo, sismograma, intensidad, Leaflet, openGL, MiniSEED, seedlink, filtro digital, IDW.

Abstract

Costa Rica has the peculiarity of being a country in which seismic events are frequent. The main reason behind this, is that the country is located in the Ring of Fire.

To watch, analyze and interpret all seismic and volcanic movements that occur in the country, the Volcanology and Seismology Observatory of Costa Rica (OVSICORI) was created. For this reason it's OVSICORI's job to spread knowledge obtained from the seismic data to prevent and mitigate the disasters generated from volcanic and seismic activity.

At the present time, OVSICORI lacks the necessary tools to show to the population all the information of seismic movements. To provide this information they depend mainly on the telephone line, in which the information provided is limited. For this reason project iReal of Tecnológico de Costa Rica and OVSICORI are working together with the intention to develop several projects to visualize seismic data.

Taking advantage of this link between universities and with the intention to solve this need, it's intended to develop a monitoring system of the seismographs in real time. The intention of this system is to obtain data from the OVSICORI seismic stations and show this information in their web page to the population in an easy way to comprehend. This tool not only pretends to obtain and register the information of an earthquake, it also intends to elaborate a simulation of the evolution of intensity sensed in different zones of the country.

This document details all the process done to elaborate this software, which covers mainly these aspects:

- Investigation of different concepts and tools of seismology
- Needed protocols to obtain data from the seismological stations
- Algorithms needed to filter the signal and methodologies to store this information so it can be shown in the application
- Web tools that allow representation of seismographs and intensity simulation
- Interpolation algorithms to elaborate the simulation

Keywords: GIS, accelerogram, seismogram, seismograph, intensity, Leaflet, openGL, MiniSEED, seedlink, digital filter, IDW.

ÍNDICE GENERAL

Resumen.....	3
Abstract.....	4
ÍNDICE GENERAL.....	5
ÍNDICE DE TABLAS.....	8
ÍNDICE DE FIGURAS.....	9
1. Introducción.....	11
1.1. Descripción general del proyecto.....	11
1.2. Antecedentes.....	12
1.2.1. Descripción de la empresa.....	12
1.2.2. Trabajos similares.....	13
1.3. Planteamiento del problema.....	15
1.3.1. Contexto del problema.....	15
1.3.2. Situación problemática.....	16
1.3.3. Definición del problema.....	16
1.4. Objetivos del proyecto.....	17
1.5. Alcances entregables y limitaciones del proyecto.....	17
2. Marco teórico.....	19
2.1. Estaciones sismológicas:.....	19
2.1.1. Instrumentos sísmicos.....	19
2.1.2. Estaciones sísmicas:.....	21
2.2. Información proporcionada por las estaciones:.....	22
2.2.1. Información obtenida del sismógrafo:.....	22
2.2.2. Información obtenida del acelerógrafo:.....	25
2.3. Protocolos de comunicación:.....	28
2.3.1. SEED.....	28
2.3.2. SeedLink.....	29
2.4. Manipulación de datos recibidos:.....	30
2.4.1. Ajuste de unidades.....	31
2.4.2. Fuentes de ruido.....	31
2.4.3. Filtros.....	32
2.5. Simulación:.....	36

2.5.1.	Interpolación:.....	36
3.	Desarrollo metodológico.....	39
3.1.	Selección y configuración de la plataforma para desarrollar el software	39
3.2.	Recepción y almacenamiento de datos	41
3.2.1.	Recepción de datos.....	42
3.2.2.	Almacenamiento de datos.....	43
3.3.	Manipulación de datos.....	44
3.3.1.	Ajuste de unidades.....	44
3.3.2.	Reducción del ruido de la señal	45
3.4.	Carga de datos en la herramienta web	48
3.5.	Elaboración de algoritmos de interpolación y animación de intensidades	51
3.5.1.	Detección del sismo:	51
3.5.2.	Animación del sismo:	54
4.	Análisis de resultados	58
4.1.	Elaboración de la aplicación	58
4.2.	Recepción de datos y almacenamiento	61
4.2.1.	Recepción de datos.....	61
4.2.2.	Almacenamiento	61
4.3.	Manipulación de datos:.....	66
4.3.1.	Ajuste de unidades.....	66
4.3.2.	Filtrado	66
4.4.	Carga de datos en la aplicación	70
4.5.	Elaboración de algoritmos de interpolación y animación de intensidades	74
4.5.1.	Detección del sismo:	74
4.5.2.	Animación	75
5.	Conclusiones y recomendaciones	89
5.1.	Conclusiones.....	89
5.2.	Recomendaciones.....	92
6.	Anexos y Apéndices.....	93
6.1.	Apéndices	93
6.1.1.	Apéndice A.1 Descripción de herramientas utilizadas	93
6.1.2.	Apéndice A.2 Diagramas de flujo	94
6.2.	Anexos	99
6.2.1.	Anexo B.1 Escalas para medición del tamaño de un terremoto.....	99

6.2.2.	Anexo B.2.	Nomenclatura del canal de acuerdo con el estándar SEED..	103
6.2.3.	Anexo B.3.	Filtro de alisado exponencial	104
6.2.4.	Anexo B.4.	Filtro FIR con el método de ventanas	107
7.	Referencias bibliográficas		110

ÍNDICE DE TABLAS

Tabla 2.1. Ejemplo de sensibilidad de tres sensores.....	31
Tabla 3.1 Ventana de tiempo de acuerdo con la magnitud del sismo.....	53
Tabla 6.1. Escala Mercalli original.....	100
Tabla 6.2 Relación entre magnitud, intensidad y aceleración	103
Tabla 6.3 Nomenclatura de la primera letra del canal de acuerdo con la banda y frecuencia de muestreo	103
Tabla 6.4 Nomenclatura de la segunda letra del canal de acuerdo con el tipo de instrumento.....	104
Tabla 6.5 Nomenclatura de la tercera letra del canal de acuerdo con la orientación del instrumento.....	104
Tabla 6.6 Respuesta ideal al impulso según el tipo de filtro.....	107
Tabla 6.7 Funciones de ventana	108
Tabla 6.8 Relación entre cantidad de coeficientes y ancho de transición	109

ÍNDICE DE FIGURAS

Figura 1.1	Captura de pantalla de Seiscomp3.....	13
Figura 1.2	Captura de pantalla de la animación de intensidades de USGS.....	14
Figura 2.1	El sismómetro tradicional de acuerdo con la red IRIS.....	19
Figura 2.2	Tipos de sismógrafos según su posicionamiento.....	20
Figura 2.3	Estructura de una estación sísmica.....	21
Figura 2.4	Ejemplo de onda P.....	23
Figura 2.5	Ejemplo de onda S.....	23
Figura 2.6	Ejemplo de onda Rayleigh.....	24
Figura 2.7	Ejemplo onda Love.....	24
Figura 2.8	Diferentes tipos de onda registrados en un sismograma.....	25
Figura 2.9	Handshake para modo uni-estación y multi-estación.....	30
Figura 2.10	Diagrama de bloques para un filtro IIR.....	33
Figura 2.11.	Diagrama de bloques para un filtro FIR de orden 3.....	34
Figura 2.12	Ejemplo de superficie obtenida utilizando el algoritmo IDW.....	36
Figura 3.1.	Captura de pantalla de versión inicial de la aplicación.....	41
Figura 3.2.	Diagrama de la herramienta desarrollada.....	42
Figura 3.3.	Sismograma sin utilizar filtros.....	48
Figura 3.4.	Captura de pantalla de una versión preliminar utilizando marcadores con cambio de color.....	49
Figura 3.5	Escala de velocidad utilizada para sismógrafos.....	50
Figura 3.6	Escala de intensidades utilizando acelerógrafos.....	50
Figura 3.7	Ejemplo de uso de vértices para formar triángulos para WebGL.....	55
Figura 3.8.	Escala de colores de intensidades para la animación.....	57
Figura 4.1	Herramienta web funcionalidad de monitoreo.....	58
Figura 4.2	Herramienta web funcionalidad de simulación.....	58
Figura 4.3	Herramienta web mostrando los sismógrafos.....	59
Figura 4.4	Herramienta web mostrando acelerógrafos.....	60
Figura 4.5	Paquete recibido de un sismógrafo guardado en formato JSON.....	61
Figura 4.6	Comando top, donde slclient es el módulo de recepción de datos.....	62
Figura 4.7	Comando top para una versión previa del módulo de recepción de datos utilizando las funciones de Jansson.....	64
Figura 4.8	Resultado del comando top con modulo de recepción de datos (slclient) en su versión final.....	65
Figura 4.9	Sismograma de una estación con offset.....	67
Figura 4.10	Estación ajustada luego de restar la media a la señal.....	67
Figura 4.11	Sismograma de una estación utilizando alisado exponencial.....	68
Figura 4.12	Sismograma de la estación Batán con filtro FIR.....	69
Figura 4.13	Sismograma de la estación Batán utilizando filtro IIR.....	69
Figura 4.14	Sismograma de la estación de Heredia utilizando filtro FIR.....	69
Figura 4.15	Sismograma de la estación de Heredia utilizando filtro IIR.....	70
Figura 4.16	Sismograma de una estación para 10 minutos.....	71
Figura 4.17	Sismograma de una estación de 5 minutos.....	71

Figura 4.18	Sismograma de una estación de 1 minuto	72
Figura 4.19	Captura de pantalla de funcionalidad de alertas por color	73
Figura 4.20	Sismograma para una estación durante el sismo de Nicoya	73
Figura 4.21	Captura de pantalla de la aplicación mostrando los últimos sismos registrados.....	74
Figura 4.22	Captura de pantalla de versión previa de la representación gráfica.....	76
Figura 4.23	Carga de capa de puntos en QGIS	76
Figura 4.24	Capa vectorial de Costa Rica.....	77
Figura 4.25	Capa de puntos resultante de la operación de cortar	78
Figura 4.26	Captura de pantalla de la visualización con únicamente área terrestre de color	78
Figura 4.27	Captura de pantalla de la aplicación con pixelado debido a la distancia de los vértices.....	79
Figura 4.28	Captura de pantalla de la aplicación con pixelado reducido	79
Figura 4.29	Resultado del mapa de intensidades utilizando interpolación IDW para el sismo de Nicoya	81
Figura 4.30	Resultado de mapa de intensidades de LIS para el sismo de Nicoya.....	81
Figura 4.31	Mapa de acelerógrafos de LIS con colores de acuerdo con la intensidad utilizando escala MM	83
Figura 4.32	Anomalía presentada por influencia de estaciones lejanas con respecto a un punto.....	84
Figura 4.33	Anomalía obtenida al utilizar un radio como criterio de ajuste.....	84
Figura 4.34	Captura de pantalla de la aplicación con el ajuste de la ecuación 4.5.1	85
Figura 4.35	Captura de pantalla de la animación en un previo a un cambio abrupto	86
Figura 4.36	Captura de pantalla de la animación con luego de un salto.....	87
Figura 4.37	Captura de pantalla de la aplicación con ajuste previa al cambio abrupto	87
Figura 4.38	Captura de pantalla de la aplicación con ajuste en donde ocurría la discontinuidad	88
Figura 6.1	Diagrama de flujo para algoritmo de recepción de datos.....	94
Figura 6.2	Diagrama de flujo para el algoritmo de almacenamiento de datos	95
Figura 6.3	Algoritmo de monitoreo de eventos.....	96
Figura 6.4	Diagrama de flujo para algoritmo de obtención de datos del sismo.....	97
Figura 6.5	Diagrama de flujo para interpolación IDW	98

1. Introducción

1.1. Descripción general del proyecto

El proyecto realizado corresponde a una solución de una necesidad del Observatorio Vulcanológico y Sismológico de Costa Rica, donde se desarrolló una herramienta web, la cual realiza un monitoreo en tiempo real de las estaciones sismológicas de OVSICORI y presenta la información obtenida de las estaciones de una forma entendible para la población.

Actualmente OVSICORI no cuenta con una herramienta para mostrar los datos obtenidos de un sismo, de forma que al ocurrir el sismo el observatorio se ve limitado en sus facultades para divulgar a la población la información sobre el evento. La única forma en la que la población puede saber sobre el sismo es mediante la consulta por medio de la línea telefónica o por medio de la información que brindan los noticieros, donde la información que se aporta usualmente es la hora y el epicentro.

Este proyecto pretende hacer uso de los datos obtenidos por las estaciones sismológicas del OVSICORI y mostrarlos a la población en una forma fácil de comprender mediante el uso de simulaciones. Estas simulaciones van a mostrar con mayor detalle un evento sísmico y permitirán a la población comprender mejor la evolución del sismo a lo largo de su duración. Este tipo de simulaciones le permitirán a la población entender qué zonas se vieron más afectadas, dónde y cuándo ocurrió el evento además de su evolución.

Finalmente, para desarrollar esta herramienta se tuvo que utilizar diversos conocimientos de programación en JavaScript y el lenguaje de programación C. Por otro lado se exploraron herramientas que permitieran la visualización de los datos como el Leaflet para la manipulación de mapas, WebGL (openGL web) para la animación y canvas.js para la elaboración de los sismogramas. También se programaron algoritmos complejos de interpolación para la animación y filtros digitales para eliminar el ruido de diversas fuentes que alteran la señal recibida desde el sismógrafo. Además se tuvo que realizar un estudio sobre el funcionamiento de los sismógrafos para entender los datos obtenidos, se debió comprender la nomenclatura utilizada para los diferentes canales que utilizan las estaciones y se tuvo que comprender los protocolos necesarios para obtener datos de las estaciones.

1.2. Antecedentes

1.2.1. Descripción de la empresa

El proyecto desarrollado, se elaboró dentro del proyecto iReal-TEC en conjunto con OVSICORI-UNA. A continuación se presenta una descripción de cada uno de los grupos involucrados en la elaboración de la herramienta.

OVSICORI-UNA

De acuerdo con el sitio web del Observatorio Vulcanológico y Sismológico de Costa Rica (s.f): “OVSICORI es un Instituto de Investigación Universitaria dedicado a la investigación de los volcanes, los sismos y otros procesos tectónicos, con el propósito de encontrar aplicaciones útiles a la sociedad que ayuden a mitigar los efectos negativos de estos fenómenos al desarrollo económico y social”.

Al ser este un observatorio, sus esfuerzos están orientados a documentar la actividad sísmica y volcánica de modo que se retroalimenten las actividades de investigación propias de un instituto de investigación universitaria.

La creación de OVSICORI se gesta dentro de la Escuela de Ciencias Geográficas de la Universidad Nacional. Donde de acuerdo con la información de OVSICORI, se señala lo siguiente: “En 1986, el Consejo Universitario de la Universidad Nacional, en virtud del desarrollo académico alcanzado por el programa de vulcanología y sismología de la Escuela de Ciencias Geográficas, lo transforma en un Instituto de Investigación Universitaria, creándose así el OVSICORI-UNA”.

Por otro lado la instalación de la red sismográfica de OVSICORI para el seguimiento de la actividad volcánica y tectónica se inició 1983, y se señala que se produjo luego “de un proceso de conversaciones con otras instituciones nacionales y extranjeras”. Donde con el aporte de la Universidad de California en Santa Cruz y con la ayuda económica de la Asociación Internacional de Desarrollo, se logró construir la red sismográfica. Actualmente la red sismográfica cuenta con un sistema de registro analógico y otro digital. Este último permite el análisis en línea de las señales sísmicas, lo que facilita el análisis de las señales y su estudio utilizando métodos computarizados.

Actualmente, OVSICORI desarrolla un programa permanente de observación, seguimiento y análisis de la actividad sísmica y volcánica, con el objetivo de reconocer oportunamente peligros que puedan impactar al territorio nacional.

Proyecto iReal Tecnológico de Costa Rica

De acuerdo con Jorge Monge Fallas (s.f): “El proyecto iReal nace en el 2012 dentro del programa eScience. El proyecto pretende desarrollar la tecnología para dotar al TEC de una instalación de realidad virtual, donde se desarrollen los elementos de interfaz, software y hardware necesarios para proyectar en tiempo real ambientes tridimensionales. Dentro de estos ambientes se podrá experimentar fenómenos espaciales de forma que el usuario esté inmerso en el ambiente ya sea física o virtualmente”.

Los ambientes descritos están actualmente en investigación en todo el mundo y el centro de este proyecto consistirá en la definición y elaboración de interfaces para estos ambientes inmersivos. Para lograr este objetivo se han implementado interfaces 3D y 2D utilizando la configuración de clusters, con la cual se permite trabajar con gran cantidad de datos. Así mismo se ha incorporado la tecnología auto-estereoscópica para modelar objetos en 3D sin tener que utilizar lentes para observar este efecto y se han implementado tecnologías que permitan al usuario interactuar con el objeto de forma más inmersiva utilizando tecnologías como el ipad.

1.2.2. Trabajos similares

Dentro del campo de la sismología se pueden encontrar trabajos similares a la herramienta que se desarrolló. Por ejemplo el mismo OVSICORI cuenta con una herramienta Seiscomp3. La cual muestra la información de estaciones en todo el mundo. Si se hace clic sobre una estación, se muestra un sismograma que indica la actividad sísmica registrada por la estación. Esta funcionalidad está incorporada a la aplicación que se desarrolló, con la ventaja de que la herramienta desarrollada es de acceso al público. La aplicación que tienen en OVSICORI es solamente de escritorio.

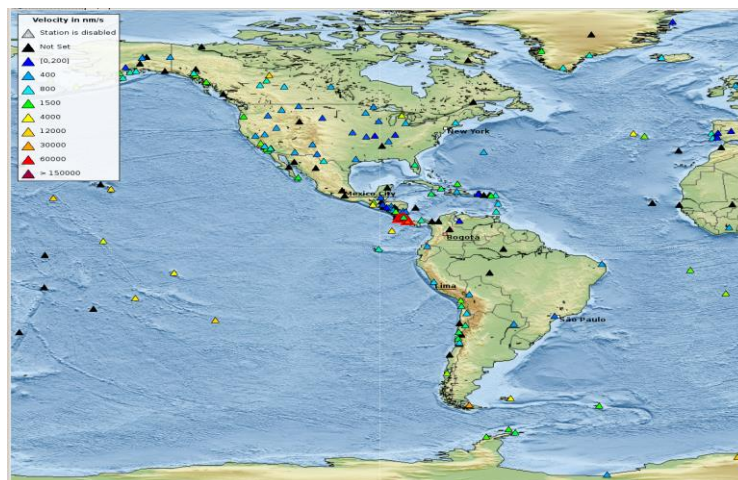


Figura 1.1 Captura de pantalla de Seiscomp3

Por otro lado la Red Integrada de Instituciones de Investigación de Sismología (IRIS en sus siglas en inglés) cuenta con la información de estaciones de diversas partes del mundo, entre estas estaciones se encuentran algunas de las que tiene OVSICORI. En este sitio se pueden realizar diversas consultas y se puede ver la actividad de una estación, sin embargo para el usuario es complicado realizar estas consultas ya que por el diseño del sitio web, la búsqueda de la información requiere de mucha navegación en el sitio web de IRIS. Por otro lado esta funcionalidad solamente muestra el nombre de la estación y el registro, sin embargo no se observa un mapa para ubicar la estación. Ninguno de los 2 casos mencionados anteriormente incorpora simulaciones utilizando los datos.

Por otro lado con respecto a la simulación de sismos, se pueden encontrar algunos servicios en línea que generan simulaciones de terremotos. Por ejemplo el sitio web de Estudios Geológicos de Estados Unidos (USGS) contiene una plataforma de simulación de intensidades de un sismo, sin embargo solo presenta simulaciones para ciertos sismos y no genera el monitoreo de las estaciones sismológicas.

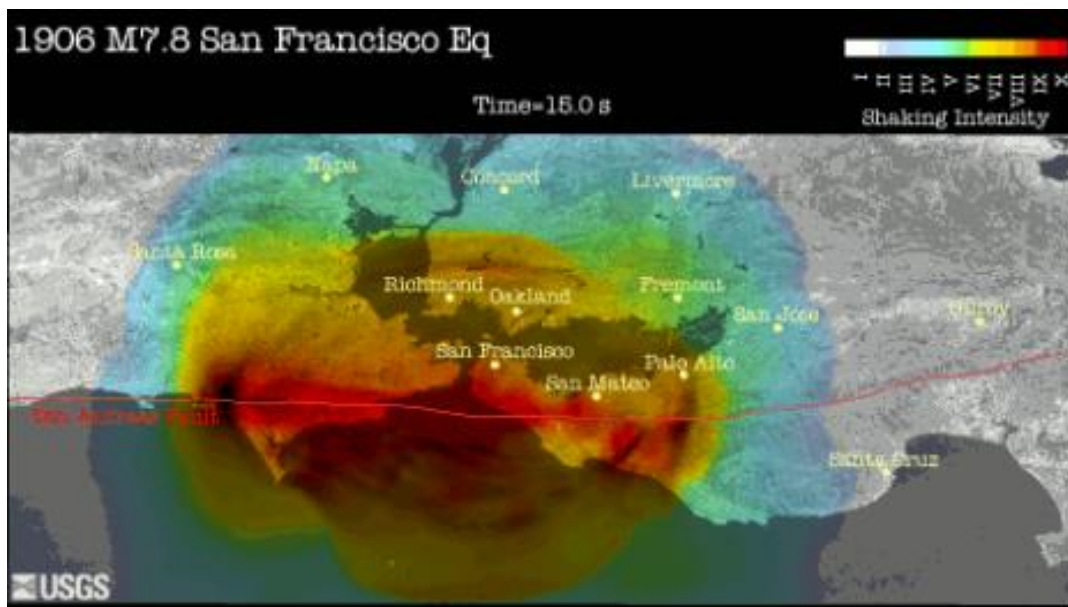


Figura 1.2 Captura de pantalla de la animación de intensidades de USGS

En Costa Rica, el Laboratorio de Ingeniería Sísmica, cuenta con una plataforma para generar un mapa de intensidades basado en una ecuación empírica para obtener los valores de intensidad. Sin embargo esta funcionalidad es solo una imagen y no representa ninguna animación, además esta plataforma no hace monitoreo de los sismógrafos ni presenta los sismogramas de las estaciones.

De modo tal que existen diversos servicios en el mundo que realizan funciones similares a las planteadas, sin embargo de alguna u otra medida no incorporan todas las funcionalidades desarrolladas en este proyecto. Hay que resaltar además que esta herramienta seguirá en crecimiento y pretende incorporar más funciones como lo es la simulación de ondas sísmicas.

1.3.Planteamiento del problema

1.3.1.Contexto del problema

Desde su creación el proyecto iReal ha tenido el objetivo de dotar al TEC con un laboratorio de visualización. Para ello el proyecto se encargó de comprar el equipo de hardware que permitiese visualizar contenido de forma distribuida utilizando un cluster. En el caso del proyecto iReal el cluster se utiliza para realizar simulaciones de diferentes tipos de elementos. Por ejemplo se puede visualizar el modelo de un corazón humano o el modelo de un puente. Estas visualizaciones se hacen en forma distribuida es decir utilizando varios monitores se percibe una sola imagen. El laboratorio actualmente cuenta con 2 clusters uno de visualización en dos dimensiones pero con alta resolución y otro de visualización autoestereoscópica.

El proyecto iReal ha atravesado varias fases y actualmente se encuentra en lo que se llamaría iReal 3.0. Donde en sus 2 primeras fases se trabajó principalmente en buscar metodologías de configuración de los clusters, búsqueda de herramientas que permitieran fácilmente la distribución de objetos, configuración de herramientas que permitan un ambiente inmersivo y finalmente metodologías y herramientas que permitiesen visualizar contenido en tres dimensiones utilizando la tecnología autoestereoscópica.

Actualmente en esta tercera fase se creó un vínculo con el OVSICORI con una serie de objetivos en particular. Primeramente para el caso de iReal es de su interés utilizar el equipo que tienen en el laboratorio de visualización, con el objetivo de poder demostrar la utilidad del uso de herramientas de visualización para comprender mejor una situación o un problema. Por el lado del OVSICORI se tiene el problema que durante los últimos 30 años han estado obteniendo información de sismos, sin embargo actualmente no cuentan con herramientas para poder visualizar esta información de alguna manera. De esta manera se pretende que con el uso de la simulación y equipos como el cluster, se pueda representar la información de sismos de una forma entendible y que permita analizar diferentes situaciones.

1.3.2.Situación problemática

Actualmente en el OVSICORI se presenta la siguiente situación siempre que sucede un sismo.

Tras ocurrir el evento muchas personas tratan de obtener información sobre el sismo y recurren al observatorio para obtener esta información. Sin embargo para obtener esta información se debe hacer uso de la central telefónica. Por este medio no se puede cubrir la demanda que ocurre en ese momento, ya que en los primeros 30 minutos se satura. Así mismo la información que se le entrega a población se limita principalmente a hora y epicentro del sismo, de modo que la información es limitada y no da una idea general sobre el efecto que tuvo el sismo en diversas zonas del país.

Por otro lado una vez que llegan los reporteros a las oficinas del observatorio para realizar el reportaje, la institución no tiene una forma entendible de mostrarle a la población el efecto del sismo y como su intensidad se sintió en diferentes zonas del país, de manera que es común ver en las noticias como se hace un enfoque de un sismograma el cual muestra una información que muy pocas personas comprenden de que se trata.

De esta forma OVSICORI al no tener una herramienta para representar la información de los sismos, tiene una limitación no solo en capacidad de distribuir la información a la población, sino que además la información que puede aportar es muy limitada y no permite a la población comprender el evento.

1.3.3.Definición del problema

Actualmente el OVSICORI carece de herramientas para mostrar los datos de un sismo a la población, para tener acceso a esta información solamente se puede obtener mediante una consulta telefónica. Al utilizar este método, la línea telefónica no es capaz de soportar la cantidad de llamadas que reciben después de un evento importante. Así mismo utilizando este tipo de consulta la información aportada es muy limitada ya que no permite comprender diferentes características del sismo tales como zonas afectadas y duración del sismo.

Para tratar con esta situación es necesario desarrollar una herramienta de software, en la cual se pueda realizar un monitoreo de las estaciones sismológicas ubicadas en diferentes sectores del país. Esta herramienta será un servicio web que se presentaría en el sitio web del observatorio, lo que permite a la población consultar fácilmente esta información. Esta herramienta va a ser capaz de registrar eventos sismológicos, de forma tal que cuando las personas acceden a la página del OVSICORI, estas podrán consultar la información del sismo sentido y comprender mejor el evento.

1.4. Objetivos del proyecto

Objetivo General:

Desarrollar un programa de monitoreo de los sismógrafos del OVSICORI en tiempo real que permita la representación de datos de sismos de forma comprensible

Objetivos específicos:

- Reconocer el funcionamiento de un sismógrafo como dispositivo generador de datos
- Identificar protocolos y herramientas que permitan el transporte e interpretación de los datos de un sismógrafo
- Interpretar datos recibidos desde los sismógrafos y definir cómo representarlos en información de fácil comprensión
- Analizar diferentes algoritmos de procesamiento de datos que permitan la visualización de datos en un mapa
- Desarrollar un programa capaz de procesar datos provenientes de los sismógrafos independiente de la plataforma

1.5. Alcances entregables y limitaciones del proyecto

Alcances:

La herramienta desarrollada cuenta con las siguientes características y funcionalidades:

- Recepción de datos en tiempo real desde las estaciones sismológicas del OVSICORI
- Filtrado de datos utilizando técnicas en DSP para eliminar el ruido de la señal proveniente de diversas fuentes
- La herramienta mostrará los últimos 10 minutos de datos registrados en una estación en forma de un sismograma
- Escala de color de acuerdo con los datos recién obtenidos que indicarán las variaciones en velocidad registradas por cada una de las estaciones
- Simulación de intensidad del sismo de acuerdo con los datos de aceleración obtenidos

Limitaciones:

- El proyecto únicamente se limitará a registrar datos específicamente de los canales verticales para acelerógrafos y sismógrafos de las estaciones sismológicas. Esto implica que se excluyen los sismógrafos horizontales.
- Se pretende incluir más funcionalidades a la aplicación como la simulación de onda sísmica utilizando los datos de velocidad de los sismógrafos, sin embargo en este proyecto únicamente se realizará la simulación de intensidad utilizando los datos de aceleración
- Las simulaciones realizadas dependen de los datos registrados por las estaciones al momento del sismo, sin embargo para el caso de intensidades únicamente se cuentan con 21 estaciones las cuales no siempre están en funcionamiento al mismo tiempo. De modo tal que la precisión de la simulación se puede ver afectada por estaciones que no estén activas en el momento del sismo.
- La recepción de datos depende de la respuesta enviada desde los servidores de la red de OVSICORI, en algunos casos los datos obtenidos pueden tener un retraso de algunos segundos.
- A lo largo del proyecto por una limitación de seguridad que imposibilita conectarse al servidor de OVSICORI desde una red externa se utilizó el servidor de IRIS, el cual cuenta únicamente con la información de sismógrafos de OVSICORI, de modo que el retraso de los datos obtenidos es significativo utilizando este servidor. Las verificaciones de las funcionalidades utilizando datos directo del servidor de OVSICORI se realizaron en las visitas mensuales a OVSICORI.

Entregables:

A continuación se describen los entregables de este proyecto.

- Informes de avances: Se escribirá un informe mensual en el cual se detalla el avance realizado en el proyecto, metodologías implementadas, bibliotecas investigadas y justificación de las decisiones tomadas.
- Software de monitoreo de sismógrafos: Se desarrollará y entregará un software que reciba datos desde sismógrafos. Este software debe ser capaz de manipular estos datos y representarlos en tiempo real en un mapa de Costa Rica.
- Código y documentación de software: Se debe entregar el código creado para la aplicación y un documento que describa las funcionalidades del software, bibliotecas utilizadas, capacidades y limitaciones de la herramienta.
- Documentación sobre proceso de configuración del software: Se realizará un documento que especifique todo el proceso de instalación requerido para poder poner a funcionar el programa. Este documento detallará otras opciones sobre como agregar nuevas estaciones sismológicas a la aplicación.

2. Marco teórico

2.1. Estaciones sismológicas:

En el área de la sismología se utilizan diversas herramientas para poder estudiar los terremotos y la propagación de las ondas sísmicas que se generan en el interior y la superficie de la tierra. Estos instrumentos varían en funciones como lo son los medidores de inclinación y tensión, sismómetro y acelerógrafos. Para este proyecto se requiere trabajar sobre los sismógrafos y acelerógrafos.

2.1.1. Instrumentos sísmicos

El sismógrafo:

De acuerdo con el Laboratorio de Ingeniería Sísmica (LIS) (2011), el sismógrafo o sismómetro es “el instrumento que registra el movimiento del suelo”. El sismógrafo permite obtener parámetros de origen de un terremoto tales como: hora, localización epicentral, profundidad y magnitud. El sismómetro tradicional consiste de los siguientes elementos: resorte, base, tambor, marcador y resorte (ver figura 2.1).

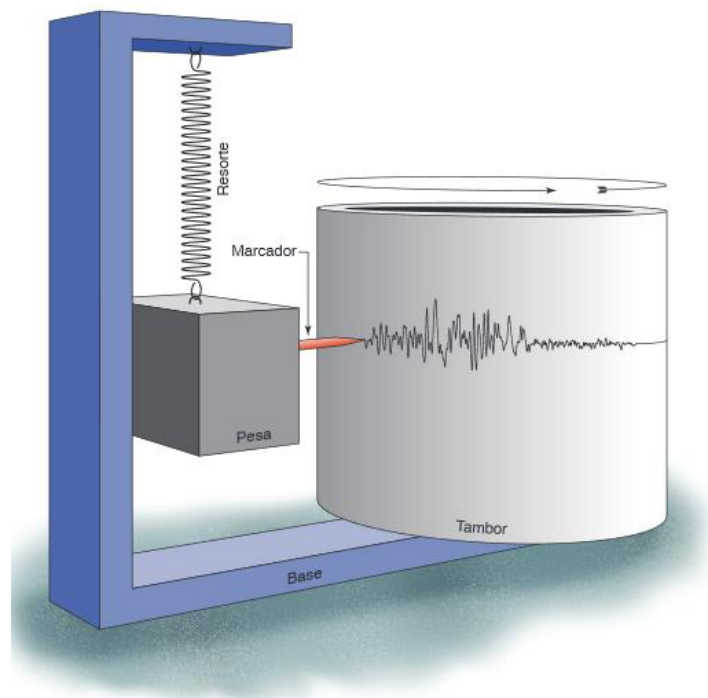


Figura 2.1 El sismómetro tradicional de acuerdo con la red IRIS¹

¹ IRIS (s.f) ¿Cómo trabaja un sismógrafo?

Este instrumento se basa en el principio de la inercia de una masa que se encuentra suspendida a un soporte rígido. De modo que el sistema es inercial ya que cumple con la primera ley de Newton. "Un cuerpo en reposo, se mantiene en reposo".

Durante un sismo, la masa se mantendrá inmóvil mientras que el soporte rígido o base se moverá junto con el suelo. Por otro lado el tambor va a estar avanzando a la velocidad de un segundero, de modo que el marcador va a estar registrando el movimiento sísmico en función del tiempo, la traza obtenida se llama sismograma.

Puesto que las ondas sísmicas presentan diferentes movimientos, estas hacen que el suelo vibre tanto de forma horizontal como vertical. De modo que es necesario que en una estación sísmica se requiera de tres sismógrafos para registrar el movimiento completo. De modo tal que los sismógrafos (y acelerógrafos) pueden ser verticales u horizontales como se observa en la figura 2.2. La utilidad del sismógrafo vertical es que permite registrar el movimiento en altura que produce el movimiento sísmico. Los sismómetros horizontales sirven para registrar el movimiento sísmico para ondas cuyo movimiento es horizontal, normalmente los sismómetros horizontales están ubicados en direcciones: nort-sur y este-oeste.

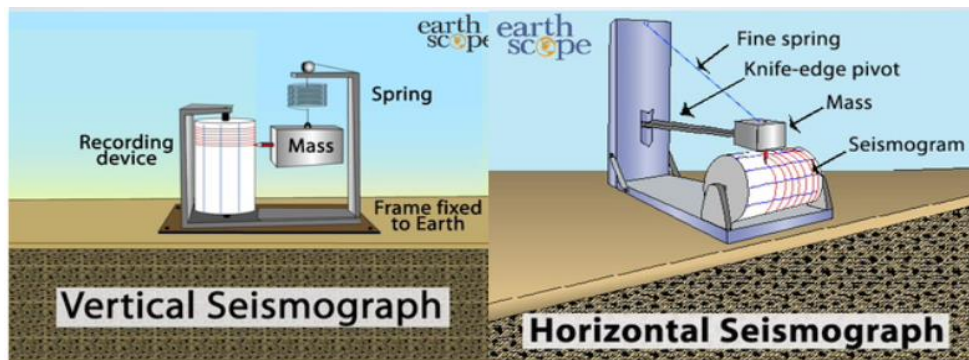


Figura 2.2 Tipos de sismógrafos según su posicionamiento²

Acelerógrafo:

De acuerdo con el Laboratorio de Ingeniería Sísmica (2011) "el acelerógrafo registra la aceleración del suelo durante un terremoto. Estos aparatos son utilizados en análisis de movimiento fuerte (grandes sismos) ya que han sido diseñados para resistir tales sacudidas", a diferencia del sismógrafo, el cual al ser más sensible, al presentarse un sismo fuerte tiende a saturarse de información. Al igual que el sismógrafo puede haber acelerógrafo horizontal o vertical.

² IRIS (2008) How does seismographs work

2.1.2. Estaciones sísmicas:

Una estación sísmica está compuesta por varios elementos, cada uno con una funcionalidad específica. Utilizando la figura 2.3 se pueden distinguir los instrumentos necesarios para establecer una estación sísmica.

- Paneles solares: Elementos necesarios para la alimentación de los dispositivos eléctricos de la estación
- Receptor GPS: Define la localización de estación
- Bóveda subterránea: Es donde estará ubicado el sensor, la bóveda subterránea pretende ubicar al sensor bajo tierra para reducir fuentes de ruido que afecten los registros del sensor.
- Sistema de recolección de datos: Se encarga de obtener la información registrada desde el sensor. Este sistema contiene al digitalizador, el cual se encarga de realizar la conversión de la señal analógica a digital. Este sistema es también el encargado de controlar las comunicaciones
- Satélite o antena: Se encarga de enviar la información a la central de datos ya sea vía internet o por radio frecuencia
- Desagüe francés: Estructura diseñada para dar una salida al agua en caso de lluvia.

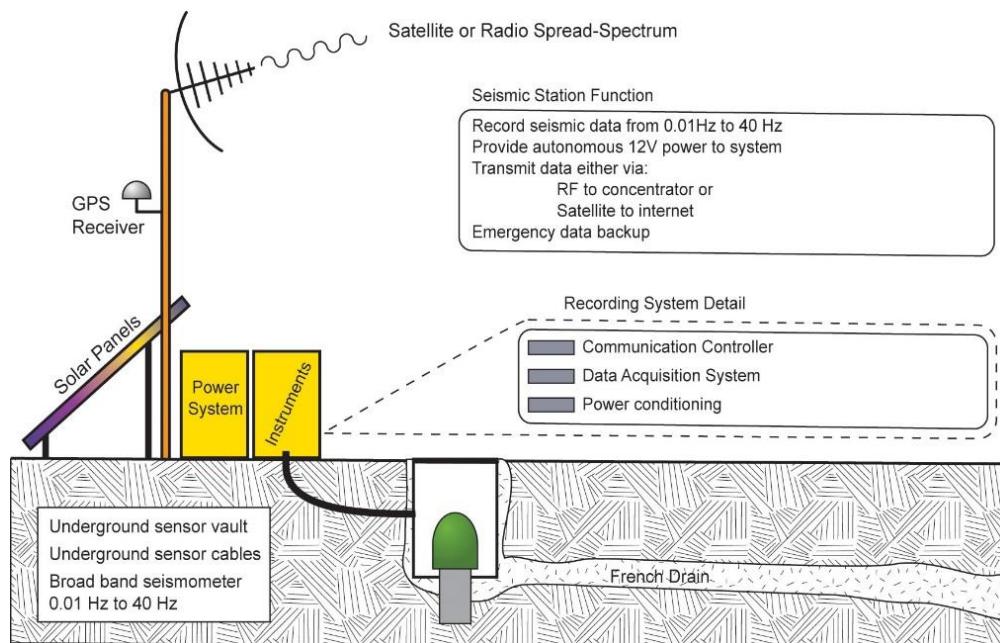


Figura 2.3 Estructura de una estación sísmica³

³ Earth Scope (s.f) Transportable Seismic Station Design Diagram

2.2. Información proporcionada por las estaciones:

Como se observó de la sección anterior, los distintos instrumentos mencionados sirven para medir diferentes datos. A continuación se describen el tipo de datos que se obtiene de los instrumentos y se definen diversos conceptos necesarios de comprender para la elaboración de la herramienta.

2.2.1. Información obtenida del sismógrafo:

Utilizando los sismógrafos se puede registrar los diferentes tipos de ondas sísmicas. Es necesario resaltar que existen diferentes tipos de ondas en el movimiento sísmico, donde dependiendo del tipo de onda la velocidad de la misma es distinta y dependiendo de condiciones como distancia del epicentro, la diferencia entre el tiempo de llegada de las diferentes ondas es mayor o menor en diferentes estaciones sismográficas.

De acuerdo con los datos de la Red Sísmica de Puerto Rico (s.f) y del Laboratorio de Ingeniería Sísmica de Costa Rica (2011) se definen 4 tipos de onda distintos divididas en 2 grupos según su propagación en la tierra:

- Ondas de cuerpo:
 - Ondas P o Longitudinales (primarias)
 - Ondas S (secundarias)

- Ondas superficiales:
 - Ondas Love
 - Ondas Rayleigh

Ondas de cuerpo:

Este tipo de ondas son llamadas de esta manera porque estas tienden a viajar a través del interior del planeta. Se definen 2 tipos de ondas:

- Ondas P o Primarias:

Son las ondas que viajan a mayor velocidad, de modo que su nombre se debe a que son las primeras en ser registradas por las estaciones. Estas tienen un movimiento de compresión y dilatación. Las ondas P pueden moverse a través de sólidos, líquidos o gases.



Figura 2.4 Ejemplo de onda P⁴

- Ondas S:

Presentan un movimiento ondulatorio, de modo que el suelo es desplazado perpendicularmente a la dirección de propagación. A diferencia de la onda p, este tipo de onda no se propaga por medios líquidos, únicamente se propagan en medios sólidos. Esta onda usualmente es sentida por la población ya que la onda S tiene mayor amplitud que la onda P.

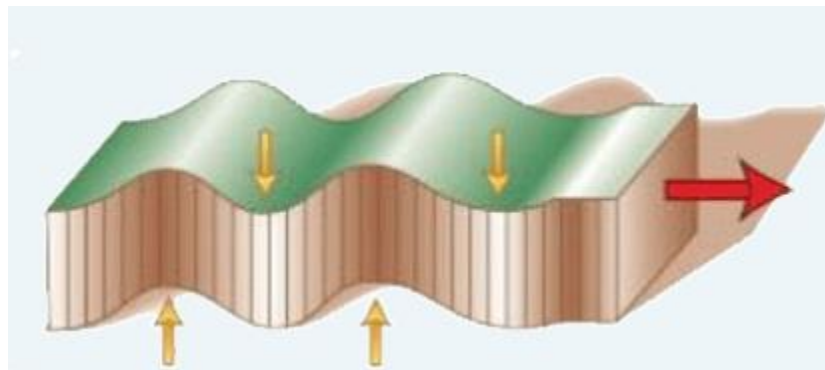


Figura 2.5 Ejemplo de onda S⁵

Ondas superficiales:

Estas ondas solamente se desplazan en la superficie del terreno. Una característica de estas ondas es que presentan amplitud máxima en la superficie y disminuyen exponencialmente con la profundidad. Además el resultado destructivo de un terremoto es producto de la propagación de estas ondas, ya que estas poseen amplitudes mayores a las ondas de cuerpo. Este tipo de ondas tiene una velocidad de propagación menor a las ondas de cuerpo y se pueden diferenciar 2 tipos.

⁴ Red Sísmica de Puerto Rico (s.f) Ondas Sísmicas

⁵ Red Sísmica de Puerto Rico (s.f) Ondas Sísmicas

- Ondas Rayleigh:

Estas ondas viajan como ondulaciones, similares a las que se producen en el agua

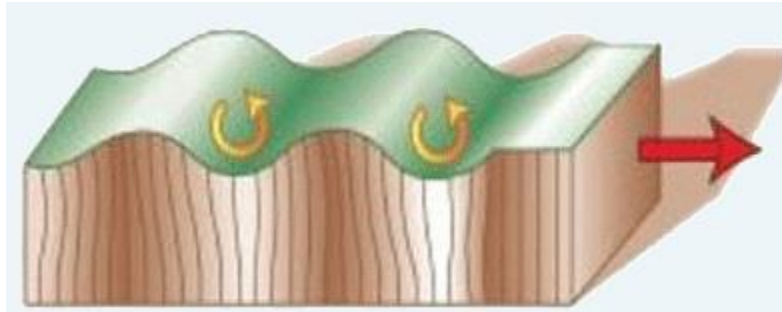


Figura 2.6 Ejemplo de onda Rayleigh⁶

- Ondas Love:

El movimiento de estas ondas es perpendicular a la propagación y paralela a la superficie de la Tierra. Este tipo de onda es una de las últimas en ser registradas en los sismógrafos. Las ondas Love producen torsión del terreno.

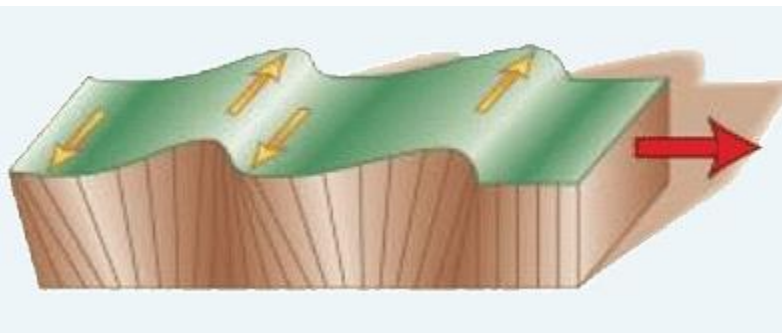


Figura 2.7 Ejemplo onda Love⁷

Mientras que las ondas de cuerpo siempre aparecen en los sismogramas, las ondas superficiales solo aparecen en aquellos registros de terremotos cuyo hipocentro no haya sido muy profundo. Además como se mencionó anteriormente las ondas sísmicas viajan a diferentes velocidades, por lo que su registro en las estaciones es en diferentes tiempos. Las ondas P son más veloces que las ondas S las cuales viajan a la mitad de la velocidad que las ondas P. Mientras más distante se encuentra el terremoto de la estación sísmica, la diferencia entre el tiempo de llegada de las ondas P y S también aumenta. Así mismo para el caso de las ondas superficiales la diferencia entre el tiempo de llegada de estas ondas con respecto a las ondas de cuerpo es aún mayor.

⁶ Red Sísmica de Puerto Rico (s.f) Ondas Sísmicas

⁷ Red Sísmica de Puerto Rico (s.f) Ondas Sísmicas

Utilizando el ejemplo de la figura 2.8, tomado de la página de LIS se puede observar con mayor claridad a través de un sismograma como un sismógrafo registra los diferentes tipos de onda. Se puede observar como dependiendo de la velocidad de propagación de la onda la primera en registrarse será la onda P, luego llegará la onda S y finalmente se observa con color azul la onda L. Es necesario aclarar que la onda Rayleigh no se muestra en este sismograma, ya que este sismograma es la representación de los datos obtenidos por un sismógrafo horizontal, para obtener los datos de esta onda se requiere un sismógrafo vertical. Además se puede observar que el sismograma presenta la coda, que es la parte del sismograma que decae conforme pasa el tiempo hasta llegar al nivel original del pre-evento cuando los otros tipos de ondas han pasado. La coda puede llegar a durar un tiempo significativo dependiendo del tamaño del sismo.

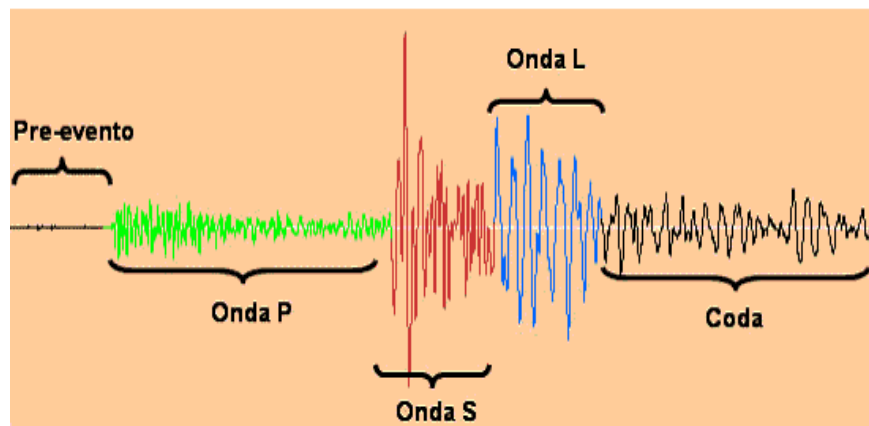


Figura 2.8 Diferentes tipos de onda registrados en un sismograma⁸

2.2.2. Información obtenida del acelerógrafo:

A partir de los datos obtenidos de un acelerógrafo, se puede obtener una relación del tamaño de un sismo. Sin embargo no solamente la aceleración es la única medida para determinar el tamaño de un sismo. Con respecto a la forma de medir el tamaño de un terremoto, la Red Sísmica de Puerto Rico (s.f) indica lo siguiente:

“Existen básicamente tres términos para describir el tamaño de un terremoto: intensidad, magnitud y aceleración”.

Es necesario definir cada uno de estos términos y hacer la relación entre estos tres para comprender que significa cada uno de estos parámetros y cómo estos servirán para realizar representaciones por medio de la simulación sobre la evolución y el impacto de un sismo.

⁸ Laboratorio de Ingeniería Sísmica (2011). Instrumentación: El sismógrafo

Intensidad:

De acuerdo con la red sísmica de Puerto Rico (s.f): “La intensidad es el aparente grado de sacudida que se siente en diferentes lugares. La intensidad es determinada en un sitio en particular, anotando los efectos que produce la sacudida en objetos, edificios, personas y en el terreno mismo”.

La escala de intensidad utilizada generalmente, se llama Escala de Intensidad Modificada de Mercalli. Esta escala va desde el I (no sentido) al XII (destrucción total), la escala original constaba de diez grados (ver tabla 6.1 en anexos) y la que se usa actualmente, que la escala de Mercalli modificada, consta de doce grados (ver tabla 6.2 en anexos).

La intensidad es una medida de los efectos de un sismo sobre un área determinada. De modo que la intensidad es una medida subjetiva, y cuanto más lejos del epicentro se tome la medida, tanto menor será esta.

Magnitud:

De acuerdo con LIS (2011): “La magnitud de un sismo es una medida de su tamaño. Es un dato objetivo, es decir, es el mismo en cualquier parte del mundo donde se calcule”.

Así mismo, la Red Sísmica de Puerto Rico (s.f) indica que para obtener la magnitud de un sismo, se debe tomar el logaritmo en base 10 del mayor movimiento del suelo registrado durante la llegada de un tipo de onda sísmica.

Se debe resaltar que al ser la escala logarítmica, la magnitud aumenta en una unidad con el aumento de diez unidades en la amplitud. Sin embargo, en cuanto a energía liberada por un terremoto, un incremento de unidad en la magnitud incrementa la cantidad de energía liberada en un factor de aproximadamente 30. Aunque existen diferentes escalas de magnitud, basadas en diferentes ondas, la mayoría de las mismas son reportadas en la escala Richter.

Aceleración:

Finalmente, de acuerdo con la Red Sísmica de Puerto Rico (s.f), “el tamaño de un terremoto puede expresarse en aceleración debido a la gravedad, que es la aceleración con la que cae una pelota abandonada en el vacío⁹”. La aceleración de un sismo se puede obtener mediante el uso de acelerógrafos, sin embargo tomando en cuenta las definiciones de cada uno de los 3 términos de esta sección se puede observar que este dato no es el más apto para los objetivos del proyecto, ya que no permite comprender en qué grado afecto un sismo diferentes zonas del país.

⁹ 1.0 g, donde g es igual a 980 cm/s²

Tomando en cuenta las 3 definiciones se puede entender que la medida que permitirá a la población comprender mejor los efectos de un sismo en distintas zonas es la intensidad. Sin embargo tanto el acelerógrafo como sismógrafo no presentan esta información directamente.

En la tabla 6.2 en anexos se puede observar una tabla comparativa de los valores de intensidad, magnitud y aceleración. En ella se puede notar que hay una relación entre las 3 medidas, donde obteniendo los valores de aceleración, se puede determinar los valores de las otras medidas, para el caso de interés sería la intensidad.

En el país se han realizado estudios para obtener la relación que hay entre la aceleración y la intensidad. Específicamente el estudio del Laboratorio de Ingeniería Sísmica (LIS) con el título: “Relación entre la aceleración pico del suelo y la intensidad Mercalli modificada en Costa Rica” define una relación entre aceleración e intensidad. Según los resultados de la investigación se obtienen 2 ecuaciones, con las cuales obteniendo el valor de la aceleración pico se puede determinar el valor de intensidad en diferentes zonas del país. Estas ecuaciones son las que LIS utiliza para hacer sus mapas de intensidad en su página web. Las ecuaciones generadas en este estudio son las siguientes:

$$MMI = 2,30 * \log(PGA_{max}) + 0,92$$

Ecuación 2.2.1 Ecuación para obtener MMI entre grados III-V (aceleraciones entre 4.9 cm/s² y 36.0 cm/s²)

$$MMI = 3,82 * \log(PGA_{max}) - 1,78$$

Ecuación 2.2.2 Ecuación para obtener MMI entre grados V-VII (aceleraciones mayores a 36.0 cm/s²)

Donde:

PGA: Aceleración máxima del suelo

MMI: Intensidad Mercalli modificada

Es necesario resaltar que si la aceleración es menor a 4.9 cm/s² se considera que la intensidad sentida es de 2. Estas ecuaciones serán utilizadas posteriormente para relacionar los datos obtenidos desde acelerógrafos y realizar la simulación de intensidades de un sismo.

2.3. Protocolos de comunicación:

Para la obtención de datos desde las estaciones sísmicas se debe comprender que existen protocolos definidos para la obtención de datos en tiempo real. A continuación se describen los protocolos y términos necesarios para realizar la obtención de datos de los sensores.

2.3.1. SEED

De acuerdo con IRIS (2004), el standard de intercambio de información de terremotos (SEED en sus siglas en inglés) es “un estándar internacional para el intercambio de datos sismológicos. SEED fue diseñado para ser utilizado por la comunidad de investigación de terremotos primeramente para el intercambio entre instituciones de datos de movimientos sísmicos. Este es un formato para información digital medida en un punto en el espacio en intervalos equidistantes de tiempo. El uso de SEED permite a los sismólogos transmitir, recibir y procesar información de una forma más fácil y exacta”.

Con respecto a la información proporcionada por IRIS sobre el estándar SEED se detallan muchos elementos del estándar como lo son convenciones, versiones, especificaciones sobre cómo escribir información SEED, encabezados de control, documentación sobre los registros de datos y mucha más información referente al estándar. Para este proyecto será necesario tomar en cuenta la información referente a la nomenclatura de los canales.

Como se mencionó anteriormente, en cada estación sísmica se encuentran tanto sismógrafo como acelerógrafos. Además de esto, pueden ser verticales u horizontales. Por ello, SEED define que se debe utilizar 3 letras para cada canal sísmico. Cada canal sísmico representa un sensor digital como los descritos en la sección 2.1.1. Cada letra identifica elementos como frecuencia de muestreo, tipo de sensor y orientación.

En la tabla 6.3 en la sección de anexos, se puede observar la nomenclatura por letra de acuerdo con el tipo de banda y frecuencia de muestreo, este dato representa la primera de las 3 letras que identifican al canal. En la tabla 6.4 en la sección de anexos, se puede observar la nomenclatura de acuerdo con el código del instrumento, este dato representaría la segunda de las 3 letras. Finalmente en la tabla 6.5 se observa la nomenclatura de acuerdo con la orientación del instrumento.

Para este proyecto únicamente se hacen uso de los sensores en orientación vertical (letra z) y se utilizan únicamente sensores de alta banda ancha y banda ancha (H y B respectivamente). Además de la identificación del canal, una estación se identifica por su nombre de estación (compuesto por un string de tamaño 5) y el nombre de la red (compuesto por un string de tamaño 2), que para la red de OVSICORI es OV. De este modo de acuerdo con el estándar SEED, para consultar la estación del Cerro de la Muerte se identificaría de la siguiente manera OV_CDM:HHZ, si se quisiera consultar la información del acelerógrafo la identificación sería OV_CDM:HNZ

2.3.2.SeedLink

De acuerdo con IRIS (s.f): “SeedLink es un protocolo de red para la recolección de datos en tiempo real”. SeedLink habilita la transmisión de datos por internet vía TCP/IP, donde toda conexión es iniciada por el cliente. El protocolo permite al cliente desconectarse y reconectarse a la red sin perder datos, esto debido a que la información se encuentra ubicada en los buffers del servidor. Esta característica va ser esencial en el desarrollo de la aplicación, ya que va a permitir acceder a datos antiguos (5 o 10 minutos después de un sismo) almacenados en los buffers. El software seiscomp3 desarrollado por el programa GEOFON, utiliza el protocolo SeedLink y en su sitio web se hace una descripción detallada del protocolo.

Protocolo

Una sesión SeedLink comienza con la creación de una conexión TCP/IP y culmina cerrando la conexión TCP/IP. Durante esta sesión los siguientes pasos se llevan a cabo:

- Abrir conexión
- HandShaking
- Transferencia de paquetes SeedLink

Handshaking

Una vez se ha establecido la conexión TCP/IP, el servidor espera al cliente para realizar el handshake sin enviar ninguna información al cliente. Durante el handshake, el cliente envía comandos SeedLink al servidor. Estos comandos sirven para: establecer parámetros de conexión, establecer los selectores de flujo e inicializar la transmisión de datos. Para el caso de este proyecto, el comando más importante es el que define el arreglo de datos (string), el cual va a indicar al servidor SeedLink cuáles son las estaciones de las que se quiere obtener información. La figura 2.9, ejemplifica el handshaking en modo uni-estación o multi-estación. Para este proyecto se utiliza el modo multi-estación, ya que se requiere obtener datos de varias estaciones.

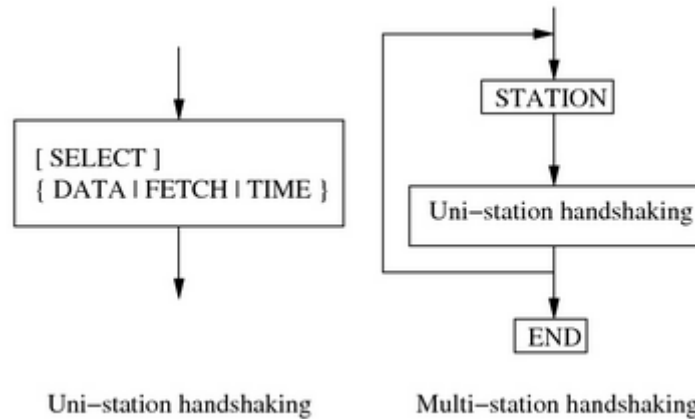


Figura 2.9 Handshake para modo uni-estación y multi-estación¹⁰

Transferencia de datos

Una vez se ha completado el handshaking, el servidor comienza a enviar paquetes de datos. Estos datos consisten de un encabezado SeedLink de 8 bytes seguido de un registro MiniSEED de 512 bytes. MiniSEED es una versión reducida de un dato SEED, el cual únicamente contiene los datos de ondas. El MiniSEED no incluye metada de la estación y el canal.

2.4.Manipulación de datos recibidos:

Considerando que el sismógrafo digital registra los datos en función a la tensión en la salida, este debe utilizar un digitalizador para hacer la conversión de la señal analógica a digital. Así mismo se deben tomar las siguientes consideraciones con respecto a los datos obtenidos de los paquetes MiniSEED.

- Los datos recibidos de acelerógrafos y sismógrafos no están en unidades de m/s y m/s^2 , de modo que se requiere hacer una conversión de unidades.
- Existen diversas fuentes de ruido que pueden alterar los registros del sensor, de forma que se requiere filtrar los datos utilizando filtros digitales para eliminar las alteraciones de la señal.

¹⁰ GEOFON program (2009). Seedlink

2.4.1. Ajuste de unidades

Para realizar los ajustes de unidades se requiere obtener los datos de sensibilidad del digitalizador por estación. En la tabla 2.1, se puede observar un ejemplo de estos datos

StaCode	Net	ChanCode	SensorSen	DigSen	StartDate
ACHA	OV	HHZ	2053.44	629330	2012,286
BARI	OV	HHZ	747.785	400000	2011,263
BATAN	OV	HHZ	1490.6	419430	2007,335

Tabla 2.1. Ejemplo de sensibilidad de tres sensores

En la tabla 2.1 se puede observar que hay 2 campos referidos a la sensibilidad: la sensibilidad del sensor y la sensibilidad digital. Para realizar la conversión de escalas únicamente se debe de seguir la ecuación 2.4.1:

$$Da = \frac{Din}{SsxSd}$$

Ecuación 2.4.1. Ecuación de ajuste de unidades

Donde:

Da: Es el dato ajustado en m/s para el sismógrafo y m/s² en el acelerógrafo

Din: Es el dato de entrada en cuentas

Ss: Sensibilidad del sensor en V/m/s para sismógrafo, V/m/s² para acelerógrafo

Sd: Sensibilidad digital en cuentas/V

2.4.2. Fuentes de ruido

A pesar de las prevenciones tomadas en la construcción de una estación sismológica¹¹, los sensores pueden verse afectados por una serie de factores que pueden alterar la señal obtenida.

Havsokv y Alguacil (2002) señalan diversas fuentes de ruido que afectan a los instrumentos de medición. Primeramente estos dividen el ruido de acuerdo con su origen, ya sea ruido generado en la instrumentación o ruido sísmico generado por las vibraciones de la tierra. El ruido sísmico se puede generar por los siguientes factores:

¹¹ Referirse a figura 2.3

- Ruido producto de la actividad humana: producido por el tráfico y maquinas, tiene frecuencias entre 2-4Hz.
- Ruido por el viento: El viento hace que cualquier objeto se mueva, de modo que genera ruido sísmico. Su frecuencia es similar a la del caso anterior.
- Ruido producido por el océano: Este es llamado ruido micro sísmico, se puede observar mundialmente. Sin embargo las zonas costeras son las que se ven más afectadas por este, ya que la energía de las olas es convertida directamente en energía sísmica, ya sea por variaciones de presión o el rompimiento de olas en la superficie. Estas ondas tienen periodos de 10 a 16 segundos

Por otro lado Stein y Wyession (2003) señalan los siguientes factores como causantes de ruido sísmico:

- Marea solar y lunar
- Fluctuaciones de temperatura
- Presión atmosférica
- Tormentas
- Actividad Humana
- Olas del mar

Stein y Wyession (2003) señalan que estas fuentes de ruido tiene un rango de frecuencias de 0.1 a 0.2 Hz. Además señalan que dependiendo de cómo un sensor sea desplegado, va a tener un gran efecto en el ruido que la estación registre.

Por otro lado el estudio de Braile (2004), señala diversos rangos de frecuencias entre las cuales se pueden encontrar diversas señales sísmicas. Particularmente señala que para sismos locales un filtro pasa banda en las frecuencias de 0.5 a 3 Hz es recomendable.

2.4.3.Filtros

Para eliminar del registro de las estaciones sísmicas todos aquellos factores que alteran la señal, es necesario diseñar filtros digitales. Estos filtros pueden ser clasificados de acuerdo con el tipo de respuesta ante entrada unitaria.

Filtros de acuerdo con el tipo de respuesta ante entrada unitaria

Filtro de respuesta infinita al impulso (IIR):

El filtro IIR es aquel que, ante un impulso en la entrada, presenta un número infinito de términos en la salida, de modo que nunca vuelve al reposo. La salida de un filtro IIR depende de las entradas actuales y pasadas además de las salidas en instantes anteriores. Un filtro IIR se puede representar por la siguiente ecuación:

$$y(n) = \sum_{l=1}^N a_l * y[n - l] + \sum_{k=0}^M b_k * x[n - k]$$

Ecuación 2.4.2. Ecuación de un filtro IIR

Donde:

- M : es el orden del filtro de pre alimentación
- b_k : son los coeficientes del filtro de pre alimentación
- N : es el orden del filtro de retro alimentación
- a_l son los coeficientes del filtro de retro alimentación
- $x[n]$: es la señal de entrada
- $y[n]$: es la señal de salida

En la figura 2.10 se puede observar el diagrama de bloques de un filtro IIR con $N=M=3$. Se puede observar como la salida depende tanto de la pre alimentación como de la retro alimentación.

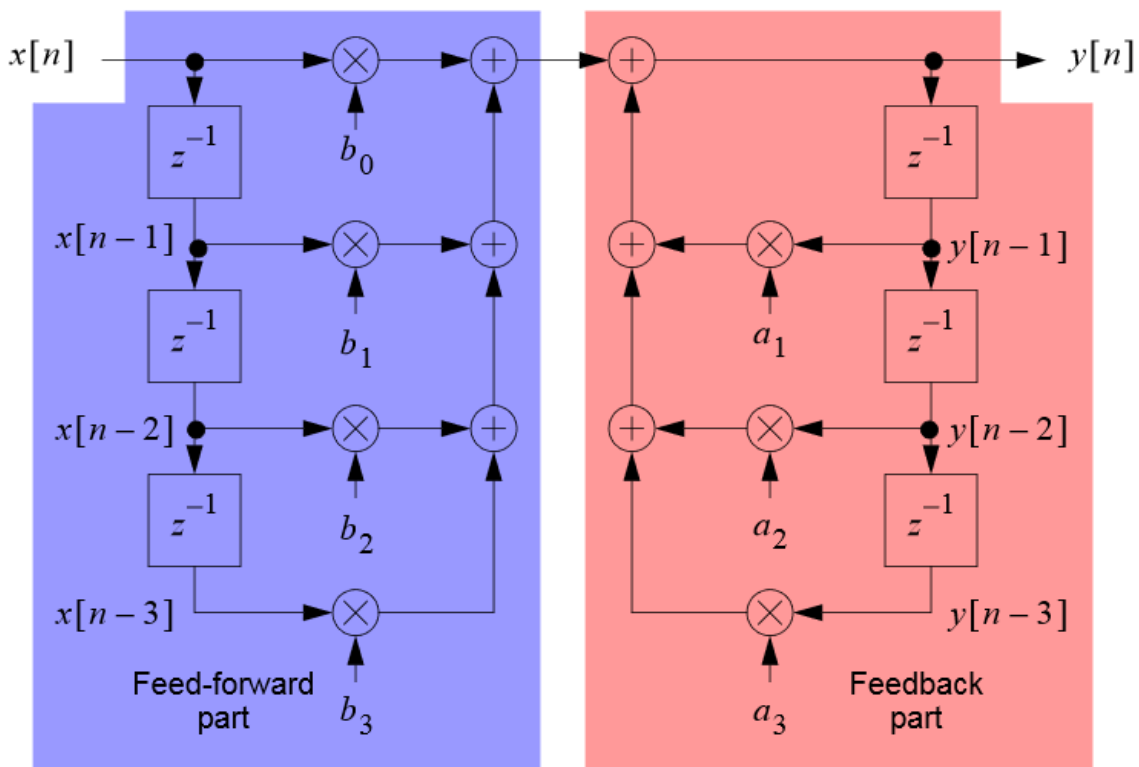


Figura 2.10 Diagrama de bloques para un filtro IIR¹²

¹² Wickert, M. (2011). IIR Filters. In ECE2610 Introduction to Signal and Systems

Filtro de respuesta finita al impulso (FIR):

De acuerdo con Wickert (2010), el término respuesta finita al impulso se debe a “que la salida del filtro es computada como una suma finita de términos pasados, presentes y futuros de la entrada del filtro”. Un filtro FIR se representa por la siguiente ecuación

$$y(n) = \sum_{k=0}^N b_k * x[n - k]$$

Ecuación 2.4.3. Ecuación de un filtro FIR

Donde:

- $x(n)$ es la señal de entrada
- $y(n)$ es la señal de salida
- N es el orden del filtro
- b_k es el valor de la respuesta al impulso en un instante k en un filtro FIR de orden N .

En la figura 2.11 se puede observar un diagrama de bloques para un filtro FIR de orden 3 el cual se representa por la ecuación:

$$y(n) = \sum_{k=0}^3 b_k * x[n - k]$$

Ecuación 2.4.4. Ecuación para el ejemplo de filtro FIR de orden 3

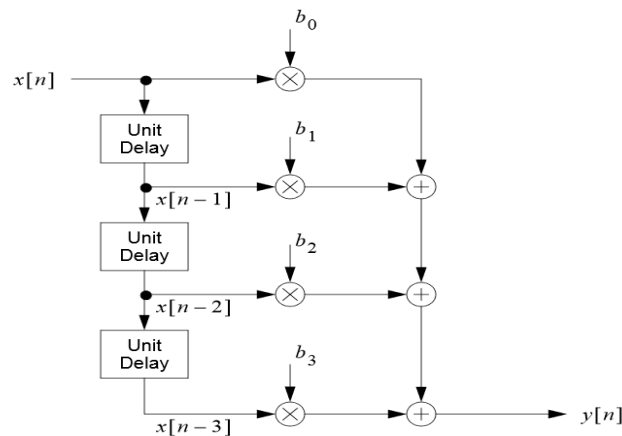


Figura 2.11. Diagrama de bloques para un filtro FIR de orden 3¹³

¹³ Wickert, M. (2011). FIR Filters. In ECE2610 Introduction to Signal and Systems

Existen diferentes métodos básicos para diseñar este tipo de filtros de los cuales se pueden mencionar los siguientes:

- Método de las ventanas, donde las más habituales son:
 - Rectangular
 - Barlett
 - Hanning
 - Hamming
 - Blackman
 - Kaiser
- Muestreo en frecuencia.
- Rizado constante
- Mínimos Cuadrados

En la sección de anexos se detalla cómo obtener los valores de b_k utilizando el método de ventanas.

Comparación de metodologías:

Punskaya (2009) señala las siguientes características entre los filtros IIR y FIR

Primeramente para el caso de filtros FIR señala lo siguiente:

- Son estables
- Pueden ser diseñados para tener fase lineal
- Hay una gran flexibilidad para controlar su respuesta en magnitud
- Son sencillos y convenientes de implementar.

En el caso de los filtros IIR se señala lo siguiente:

- Los filtros FIR suelen presentar problemas cuando una fase lineal es requerida
- Si la distorsión en fase es tolerable, el filtro IIR es recomendable:
 - Usualmente se requiere menos parámetros para obtener filtros de corte abruptos
 - Se requiere menos computaciones y memoria para ejecutarlos
- Su diseño es difícil
- Posee problemas de estabilidad

2.5.Simulación:

Para realizar una simulación de intensidades utilizando la información obtenida desde acelerógrafos, se debe de tomar en cuenta que será necesario definir los algoritmos de interpolación necesarios. Estos algoritmos servirán para mostrar un comportamiento cercanamente real a partir de un conjunto de puntos conocidos.

2.5.1.Interpolación:

Para elaborar la interpolación de un problema en el área de sistemas de información geográfica (SIG), se puede utilizar como base los algoritmos utilizados por diversos software SIG como ArcGIS y QGIS. Estas herramientas utilizan los algoritmos de distancia inversa ponderada (IDW) y Kriging para hacer un análisis de superficie

Algoritmo IDW:

De acuerdo con la documentación de QGIS 2.2. En el algoritmo de interpolación IDW, los puntos de muestra son pesados en la interpolación. De modo tal que la influencia relativa de un punto con respecto a un punto desconocido, decrece de acuerdo con la distancia entre estos dos.

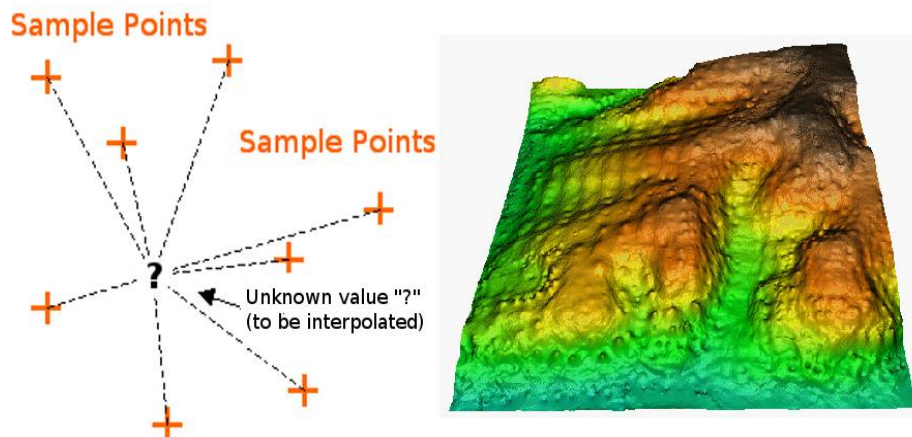


Figura 2.12 Ejemplo de superficie obtenida utilizando el algoritmo IDW¹⁴

¹⁴ Quantum GIS Development Team. (2014). Spatial Analysis (Interpolation).

La estructura general de la interpolación IDW viene definida por las siguientes ecuaciones

$$w_i(x) = \frac{1}{d(x, x_i)^p}$$

Ecuación 2.5.1 Ecuación del factor $w_i(x)$

$$u(x) = \begin{cases} \frac{\sum_{i=1}^N w_i(x) * u_i}{\sum_{i=1}^N w_i(x)} & \text{si } d(x, x_i)^p \neq 0 \text{ para todo } i \\ u_i & \text{si } d(x, x_i)^p = 0 \text{ en algún } i \end{cases}$$

Ecuación 2.5.2 Ecuación para interpolar un punto desconocido utilizando IDW

Donde:

x: Es el punto a interpolar

x_i : Es uno de los puntos conocidos

$d(x, x_i)^p$: Representa la distancia entre el punto x a interpolar y el punto x_i conocido elevado a la p conocido como el factor de poder

$w_i(x)$: Es la función de peso

u_i : Es el valor del punto conocido

$u(x)$: Es el valor del punto interpolado

N: representa la cantidad de puntos conocidos

Es necesario resaltar, que la calidad de la interpolación se puede reducir si la distribución de puntos es muy disparea.

Algoritmo de Kriging

De acuerdo con la documentación de ArcGIS, el método de Kriging está basado en modelos estadísticos que entre los cuales incluye la auto correlación, la cual es la relación estadística entre puntos conocidos. Kriging asume que la distancia o dirección entre puntos de muestra reflejan una correlación espacial que puede ser utilizada para explicar la variación en una superficie.

Kriging al igual que IDW mide la distancia entre los valores conocidos para obtener una predicción de una localización desconocida. Sin embargo IDW depende únicamente en la distancia para predecir la localización, por otro lado en el método de Kriging los pesos están basados tanto en la distancia entre los puntos conocidos como en el arreglo espacial de los puntos conocidos. Para utilizar el arreglo espacial en los pesos, la auto correlación espacial debe ser cuantificada.

De modo tal que para realizar la interpolación de Kriging se deben realizar una serie de pasos:

1. Encontrar los puntos validos de entrada
2. Determinar las distancias entre todos los puntos válidos y encontrar el valor del semi variograma para estas distancias, estos valores son computados en una matriz y luego esta es invertida
3. Para el primer punto de salida, determinar la distancias con respecto a todos los puntos de entrada y encontrar el semi variograma para estas distancias, estos valores son guardados en un vector
4. Se calculan los factores de peso (vector w) multiplicando la matriz del punto 2 con el vector del punto 3 (estos factores solo son válidos para el punto que se está calculando)
5. Se calcula los valores para el punto de salida como la suma de productos de los factores y los valores de los puntos conocidos
6. Se repite el paso 3 al 5 para cada uno de los puntos de salida

3. Desarrollo metodológico

3.1. Selección y configuración de la plataforma para desarrollar el software

Inicialmente se investigaron diferentes bibliotecas SIG que fuesen compatibles con los frameworks del laboratorio de visualización del TEC y permitieran desarrollar un servicio web. Para ello se partió con la propuesta de desarrollar el software utilizando una biblioteca llamada OSGEarth. Utilizar esta biblioteca presentaba varias ventajas como lo es poder realizar el código en C++, el cual es un lenguaje muy utilizado en la carrera, por otro lado esta biblioteca es compatible con Omegalib el cual es uno de los softwares que se utilizan para realizar la visualización distribuida en los clusters del laboratorio del CIC. Sin embargo esta biblioteca no presentaba la posibilidad de desarrollar el software para presentarlo tanto en un cluster como en un navegador de internet.

El objetivo principal del proyecto es desarrollar un software que pueda ser accedido por cualquier persona que quiera consultar la información de un sismo reciente. Por ello, es necesario que este software se pueda consultar desde la página web del OVSICORI. De esta manera se decidió buscar opciones que pudieran cumplir con este requisito. Para ello se investigaron opciones como WebGL Earth, Leaflet y Modest Maps. A continuación se describen las ventajas y desventajas de cada una de estas bibliotecas.

- WebGL Earth:
 - Ventaja:
 - Uso de la tecnología de html5 y WebGL, lo cual permitiría generar una animación atractiva y fácil de comprender para el usuario
 - Desventaja:
 - Posee poca documentación
 - Contiene pocas herramientas en comparación a otras bibliotecas
- Modest Maps:
 - Ventajas:
 - Compatible con el SAGE 2, el cual es otro de los software que se utilizan en el laboratorio de visualización del TEC
 - Permite desarrollar la aplicación en diversas plataformas como PHP, Flash, Python y JavaScript
 - Desventaja:
 - Contiene pocas herramientas en comparación de Leaflet

- Leaflet:
 - Ventajas:
 - Mucha documentación: Esta es una biblioteca muy popular escrita en JavaScript, al haber mucha documentación permite encontrar muchos ejemplos de diferentes software que cumplen condiciones "similares" a las que se pretenden en el proyecto.
 - Gran cantidad de herramientas: Leaflet permite utilizar herramientas como el time dimension el cual permite ver una especie de video de un evento, esto va a ser de gran utilidad para el usuario que quiera observar a detalle un sismo registrado por el software. Así mismo la biblioteca permite la carga sencilla de capas de información en formato GeoJSON, contiene marcadores y es compatible con WebGL

Finalmente se decide utilizar la biblioteca Leaflet ya que ofrece más facilidades para desarrollar la aplicación web. De esta forma se empezó a realizar la carga de capas y mapas.

Con respecto a la carga de capas de información se requiere únicamente las capas de puntos de los sismógrafos y acelerógrafos de OVSICORI. Para obtener estas capas se utilizó la información de la página del observatorio donde se obtuvo el nombre de la estación, la latitud y longitud. Esta información fue guardada en un archivo de texto y se utilizó el programa QGIS para pasarlo a formato GeoJSON. La carga del mapa se realizó utilizando la información de openstreetmaps.

Para desarrollar los elementos visuales como menús, botones y sliders se utilizó la biblioteca Bootstrap versión 3.3.5. La cual es muy sencilla de instalar ya que solo es necesario descargar el archivo provisto en el sitio web de la biblioteca. Esta biblioteca es fácil de utilizar y contiene mucha documentación y ejemplos en diversos sitios web lo cual facilita mucho la programación. Otra ventaja que ofrece esta biblioteca, es que permite agregar características como diseño web adaptable. Esta característica permitiría a la aplicación ser utilizada en diversos dispositivos como tablets y celulares.

Finalmente luego de tomar las decisiones sobre las herramientas a utilizar, se desarrolló el código en JavaScript utilizando Leaflet que cargara un mapa de openstreetmaps y cargara la información de las capas de puntos. Utilizando la funcionalidad de marcadores de Leaflet, se logró asignar marcadores para observar la ubicación de cada estación en el país. En la figura 3.1 se puede observar una captura de pantalla preliminar de la aplicación.

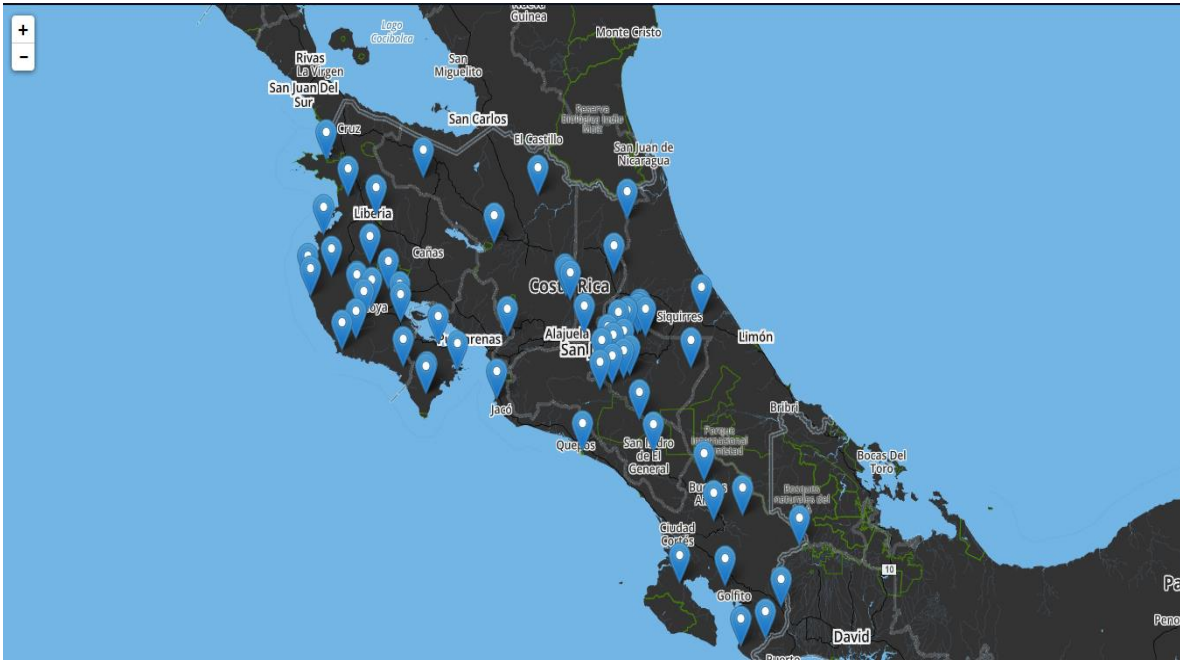


Figura 3.1. Captura de pantalla de versión inicial de la aplicación

Se puede observar los marcadores en diferentes zonas del país los cuales representan los sismógrafos de OVSICORI.

3.2.Recepción y almacenamiento de datos

Tomando en cuenta la necesidad de elaborar una aplicación que se encargue de obtener y almacenar los datos, se considera entonces que en el software van a interactuar diferentes módulos con funcionalidades específicas. Por ejemplo existen las estaciones sismológicas, las cuales envían información al centro de datos de OVSICORI, en este se almacena la información utilizando un servidor SeedLink. Para consultar la información se necesita desarrollar un módulo que obtenga los datos desde este servidor, luego procese la información recibida y finalmente guarde los datos. De modo tal que un usuario desde su navegador ingrese al sitio web de OVSICORI y ejecute la aplicación en JavaScript la cual utiliza la información que se almacenó por medio del módulo de transmisión de datos.

En la figura 3.2 se puede observar un diagrama que ejemplifica la interacción de todos estos elementos.

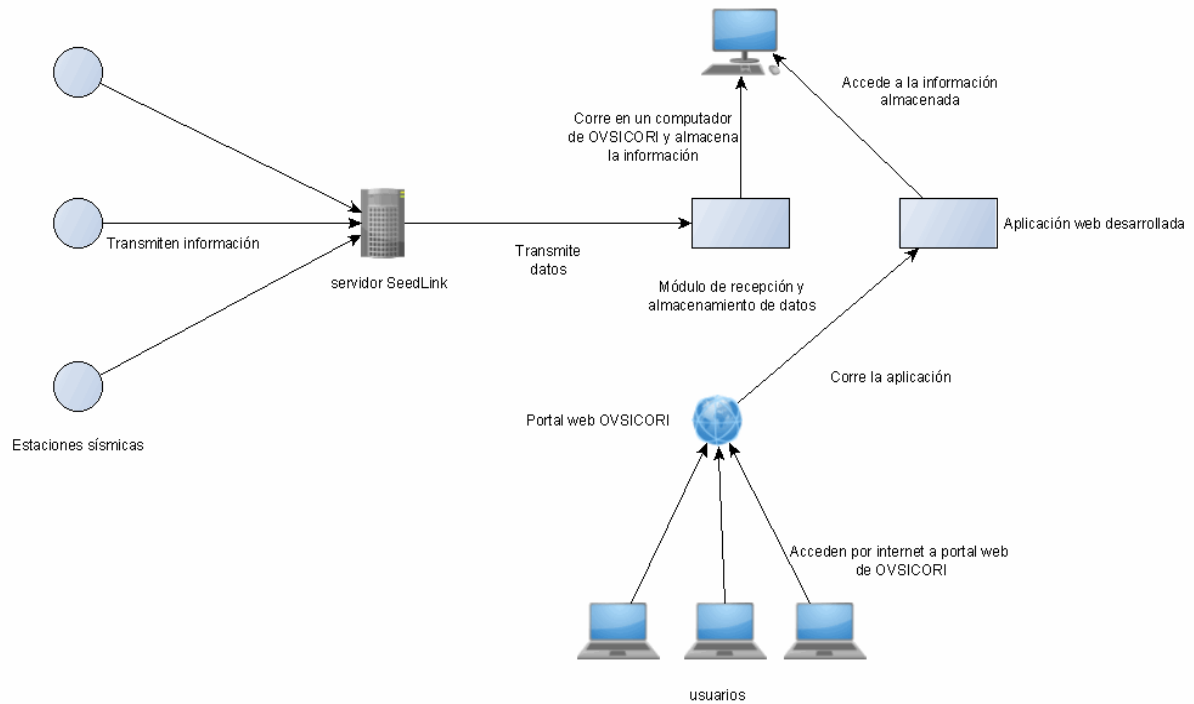


Figura 3.2. Diagrama de la herramienta desarrollada

3.2.1. Recepción de datos

Para obtener datos desde las estaciones sismológicas, se debe acceder a la información almacenada en el centro de datos de OVSICORI. Este centro de datos cuenta con un servidor SeedLink, de modo que para realizar las consultas, se debe utilizar una biblioteca que contenga todas las funcionalidades necesarias para comunicarse con un servidor SeedLink. La red IRIS provee la biblioteca libslink, la cual cumple con estas necesidades y está escrita en el lenguaje de programación C.

De este modo, se investigó la biblioteca libslink y se logró comprender cómo instalarla, conectarla al servidor, establecer las estaciones de las cuales se va a obtener la información y cómo obtener las muestras de datos de un paquete MiniSEED. La comunicación entre este módulo y el servidor SeedLink se realiza por medio del protocolo TCP, donde este módulo es un cliente que se conecta al servidor. El servidor al recibir un paquete de datos desde una estación sísmica transmite esta información a los clientes conectados.

3.2.2. Almacenamiento de datos

Para almacenar toda la información recibida desde el servidor, se decidió guardarla en archivos formato JSON. Esta decisión se tomó para mantener la simplicidad de la aplicación ya que JSON es un formato ligero para el intercambio de datos. El uso de una base de datos complicaría la aplicación que correría en OVSICORI ya que al implementar las consultas se requiere más operaciones por parte del procesador para almacenar los datos en comparación a utilizar archivos donde solo se van agregando los últimos registros. El utilizar esta estrategia facilita la migración de la aplicación, ya que solo se requeriría trasladar los códigos necesarios para ejecutar la aplicación e instalar las bibliotecas necesarias para que corra la aplicación en el computador utilizando un Makefile. Utilizar esta estrategia es favorable visto desde la perspectiva de la aplicación web (usuario de la aplicación) ya que solo requiere hacer lecturas, de forma que el acceso a los datos es fácil y rápido de implementar por la aplicación.

Para almacenar los datos, se requirió escribir un código donde al recibir la información de una estación se guarda en un archivo el cual luego va a ser leído por la aplicación en JavaScript. Para ello se requirió primero definir qué elementos iban a ser almacenados en cada objeto del JSON y cuales llaves se utilizarían para almacenarlos. A continuación se muestra un ejemplo del objeto que es almacenado en los archivos.

```
{"settings":{"Station": Nombre estación,"channel": nombre de canal},"features":[arreglo de muestras]}
```

Donde:

- Station: Es la llave que señala al string del nombre de estación
- Channel: Es la llave que señala al nombre del canal
- Features: Es la llave que señala al arreglo de muestras, es decir cuando se agrega una muestra al archivo se van agregando en un arreglo de objetos. A continuación se muestra un ejemplo de un objeto del arreglo

```
{"type":{"properties":{"Samples": cantidad de muestras}, "timestamp" :{"year": año, "day": día,"hour": hora, "minutes": minutos,"seconds": segundos,"fract": milisegundos},"amplitudes": [arreglo con muestras]}
```

Donde:

- Samples: Es la llave que señala al entero que define cuantas muestras hay en este objeto
- Timestamp: Es la llave que señala a la fecha en la cual fue registrado el paquete MiniSEED
- Amplitudes: Es la llave que señala el arreglo con las muestras obtenidas de una estación

Inicialmente el programa solo guardaba las muestras sin tomar en cuenta un límite de almacenamiento. De modo que se definió un parámetro para eliminar muestras viejas, ya que si solo se almacenan las muestras, estos archivos llegarían a ser muy grandes en poco tiempo. Por ejemplo solo almacenar los datos de una estación a lo largo de un día representaría almacenar 86400 objetos JSON en un archivo. Esto significaría que cada archivo tendría muchos datos y ocuparían mucho espacio del disco duro, lo cual complicaría la lectura del archivo tanto para la aplicación web como para el servicio que obtiene y almacena los datos.

Para solucionar esta situación, se tomaron en cuenta 2 bibliotecas escritas en el lenguaje C para manipular archivos JSON. Esto con el objetivo de poder analizar la cantidad de muestras en los archivos JSON, de modo que la cantidad de muestras en un archivo no supere las seiscientas muestras (10 minutos). Las opciones que se tomaron en cuenta fueron:

- Jsmn: Biblioteca muy sencilla, la cual no requiere ninguna instalación, está orientada a sistemas que tienen pocos recursos y su programación es muy sencilla.
- Jansson: Es una biblioteca más robusta, contiene muchos más métodos para procesar un archivo JSON y su instalación solo requería de un Makefile.

Se desarrolló un código utilizando ambas opciones, para ambos casos los datos fueron almacenados en archivos JSON utilizando el formato que se describió anteriormente. En la sección de resultados se discuten los resultados obtenidos utilizando cada una de las bibliotecas, sin embargo para el resto del proyecto se decide utilizar la biblioteca Jansson. Del mismo modo en la sección de apéndices se puede observar el diagrama de flujo para la recepción y almacenamiento de datos.

3.3.Manipulación de datos

3.3.1.Ajuste de unidades

Como se mencionó en la sección 2.4, los datos que se reciben desde las estaciones sismográficas presentan 2 situaciones: primeramente los datos están en función de la tensión de modo tal que se debe realizar un ajuste de unidades. Por otro lado los datos que se obtienen están alterados por diversas fuentes de ruido, por lo cual es necesario utilizar técnicas de filtros digitales para ajustar la señal.

Con respecto al ajuste de unidades. Se codificó las funciones necesarias para realizar el ajuste, de modo tal que al iniciar el módulo de recepción de datos, se carga la información de sensibilidad digital y sensorial de cada uno de los sensores de los cuales se va a recibir información. Una vez cargada esta información, al recibir datos desde el servidor SeedLink a cada una de las muestras del paquete recibido se les aplica la ecuación 2.4.1. De esta forma se

obtiene un arreglo de datos con valores en unidades de m/s para sismógrafos y m/s² para acelerógrafo.

3.3.2.Reducción del ruido de la señal

De acuerdo con la información de la sección 2.4, se puede observar que los estudios indican que las frecuencias de los factores que producen ruido, están entre los 0.1 a 0.2 Hz según el estudio de Stein y Wysession. Por otro lado, según los estudios de Havsokv y Alguacil, las frecuencias son más variables 2-4 Hz para el ruido humano o por viento y de 0.1 a 0.0625 Hz para el ruido producido por el océano.

Utilizando el estudio de Braile (2004), se decide implementar un filtro pasa banda entre 0.5 a 3 Hz ya que estas son las frecuencias en las que se ubican los sismos regionales. Es necesario resaltar que el filtro solo va a eliminar algunas fuentes de ruido como el caso del ruido producido por el océano, sin embargo el ruido humano va a estar presente en la señal de salida.

Es necesario tomar en cuenta que también existe ruido generado por la instrumentación, como lo es un offset en DC. Este offset puede ser eliminado utilizando filtro paso alto o calculando la media de una muestra y restarle este valor a cada elemento de la muestra.

Para realizar el filtrado se programaron diversas soluciones discutidas a continuación:

- Reducción del offset mediante el cálculo de la media
- Filtro pasa banda por el método de alisado exponencial
- Filtro pasa banda utilizando el método de ventanas
- Filtro Butterworth IIR

Reducción del offset mediante el cálculo de la media

Este método es un filtro muy sencillo el cual calcula la media de un conjunto de datos utilizando la siguiente ecuación.

$$m = \frac{1}{N} \sum_{i=0}^N x_i$$

Ecuación 3.3.1 Media aritmética

Donde m es la media y x_i representan los valores de un paquete de datos. Una vez obtenido el valor de la media se le resta a cada uno de los datos este valor. El resultado obtenido de este filtro es una señal sin offset en DC.

Filtro pasa banda por el método de alisado exponencial

El método de alisado exponencial es un tipo de filtro muy básico, el cual a partir de un factor α y utilizando la siguiente ecuación 3.3.2 se logra obtener una señal de salida más continua.

$$s_0 = x_0$$
$$s_t = \alpha x_t + (1 - \alpha)s_{t-1}$$

Ecuación 3.3.2 Ecuación del alisado exponencial para filtro paso bajo

Donde:

S: Representa la señal de salida utilizando el filtro

S_t: Representa un elemento de la señal en un momento t

α : Representa el factor de alisamiento el cual tiene valores entre 0 y 1

Para el caso de un filtro paso bajo y utilizando como modelo el filtro paso bajo de un circuito RC se tiene que:

$$\alpha = \frac{dt}{dt + RC}$$

Ecuación 3.3.3 Factor α para filtro paso bajo

Donde:

$$dt = \frac{1}{fm}$$

Ecuación 3.3.4 Factor dt para obtener factor α

$$RC = \frac{1}{fc * 2 * \pi}$$

Ecuación 3.3.5 Factor RC para obtener factor α

Para el caso de un filtro paso alto el factor α se obtiene con la siguiente ecuación:

$$\alpha = \frac{RC}{dt + RC}$$

Ecuación 3.3.6 Factor α para filtro paso alto

Finalmente el filtro paso alto tiene una diferencia con respecto a la ecuación 3.3.2 y se describe con la siguiente ecuación:

$$s_0 = x_0$$
$$s_t = \alpha(x_t + s_{t-1} - x_{t-1})$$

Ecuación 3.3.7 Ecuación del alisado exponencial

En la sección de anexos se puede observar la demostración de todas estas ecuaciones.

Con respecto a este método, para implementar el filtro pasa banda, primeramente se pasará la señal de entrada por el filtro paso bajo y luego el resultado se pasa por el filtro paso alto.

Filtro FIR pasa banda utilizando el método de ventanas

Como se mencionó en la sección 2.4.3, existen diversos métodos para diseñar un filtro FIR, entre estos el método de ventanas. Tomando la ecuación 2.4.3 el factor b_k de la respuesta al impulso se obtiene a partir de la multiplicación de la respuesta ideal al impulso del filtro con una función de ventana. Esta función de ventana puede variar y se distinguen 6 las cuales se observan en la sección de anexos. Así mismo, en la sección de anexos se puede consultar la tabla 6.1 para obtener la respuesta ideal al impulso del filtro y la metodología para diseñar un filtro de este tipo.

Con respecto a este filtro, se decide programar la ventana de Hamming ya que presenta un mejor balance entre rizado y atenuación de la banda rechazada con respecto a las otras funciones.

Filtro Butterworth IIR

Finalmente se decide codificar un filtro Butterworth IIR donde los coeficientes b_k y a_l se obtuvieron utilizando el sitio web de Tony Fisher de la universidad de York. En esta página se introducen las especificaciones de los filtros como tipo de filtro (paso bajo, paso alto, pasa banda), frecuencia de muestreo (100 canal HHZ, 200 canal HNZ), frecuencia de corte (0.5 y 3 Hz para el filtro pasa banda) y orden del filtro (4 para obtener una menor banda de transición), una vez definidos estos factores el sitio web le muestra los valores de los coeficientes.

Se utiliza este enfoque ya que para determinar los coeficientes de un filtro IIR, se requieren muchos cálculos. El uso de esta herramienta agiliza el cálculo y es más seguro ya que asegura que el resultado no tiene errores de cálculos.

Todos los filtros anteriores fueron programados y probados, estos algoritmos son ejecutados dentro del módulo de recepción de datos y se ejecutan luego de realizar el ajuste de unidades descrito en la sección 3.3.1. En la sección de resultados se discuten los resultados obtenidos para cada uno de estos filtros.

3.4.Carga de datos en la herramienta web

Después del procesamiento de la señal recibida y habiendo guardado en archivos cada muestra, se procedió a desarrollar los elementos visuales de la aplicación web utilizando la aplicación preliminar desarrollada en la sección 3.1.

Primeramente se procedió a escribir el programa en JavaScript para abrir un archivo JSON al hacer clic sobre el marcador de una estación. Una vez completada esta funcionalidad, se procedió a mostrar los datos del archivo en la pantalla en forma de un sismograma. Para mostrar el sismograma se probaron 2 bibliotecas: flot.js y canvas.js, la cuales cargan un conjunto de datos y los muestran en un gráfico.

Con respecto al sismograma se toman en cuenta los siguientes aspectos:

- El sismograma muestra los últimos 10 minutos de información registrados por una estación
- Si los datos registrados más recientes por una estación sobrepasan los 10 minutos, el sismograma no va a mostrar ningún resultado
- El sismograma se actualiza constantemente, de modo que se va a observar cuándo una estación registra nuevos datos
- Las unidades que muestra cada sismograma están de acuerdo con el tipo de sensor que representan¹⁵
- Se agregan también funcionalidades para visualizar un sismograma de los últimos 5 minutos y 1 minuto con las mismas consideraciones del caso de 10 minutos

En la siguiente sección se discuten los resultados obtenidos para cada una de las bibliotecas utilizadas. Sin embargo para el resto del proyecto se decide utilizar la biblioteca canvas.js. En la figura 3.3 se puede observar una captura de pantalla del resultado de un sismograma sin utilizar filtros usando canvas.js.

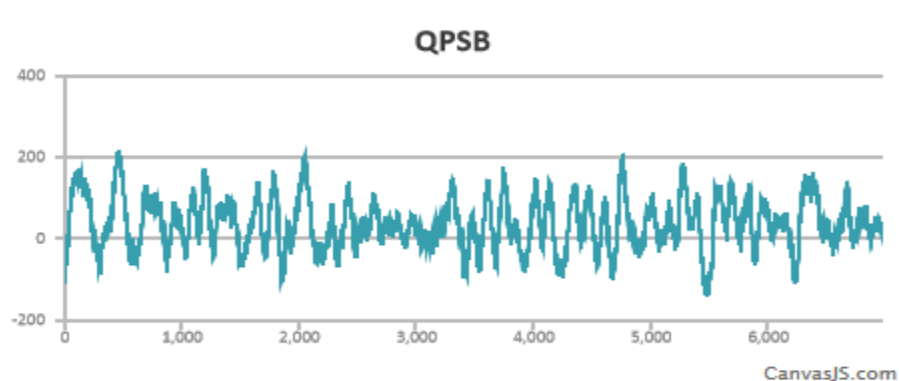


Figura 3.3. Sismograma sin utilizar filtros

¹⁵ m/s para sismógrafo m/s² para aceleración

Por otro lado, se agrega una funcionalidad visual. Esta consiste en cambiar el color de los marcadores de las estaciones de acuerdo con el máximo registro de velocidad o intensidad de una estación en el último paquete recibido. Para implementar esta funcionalidad se definió una escala tanto para velocidad como para aceleración, las cuales se pueden observar en la figura 3.3 y 3.4 respectivamente. En la figura 3.4 se puede observar una captura de pantalla de una versión preliminar de la aplicación incorporando esta funcionalidad.

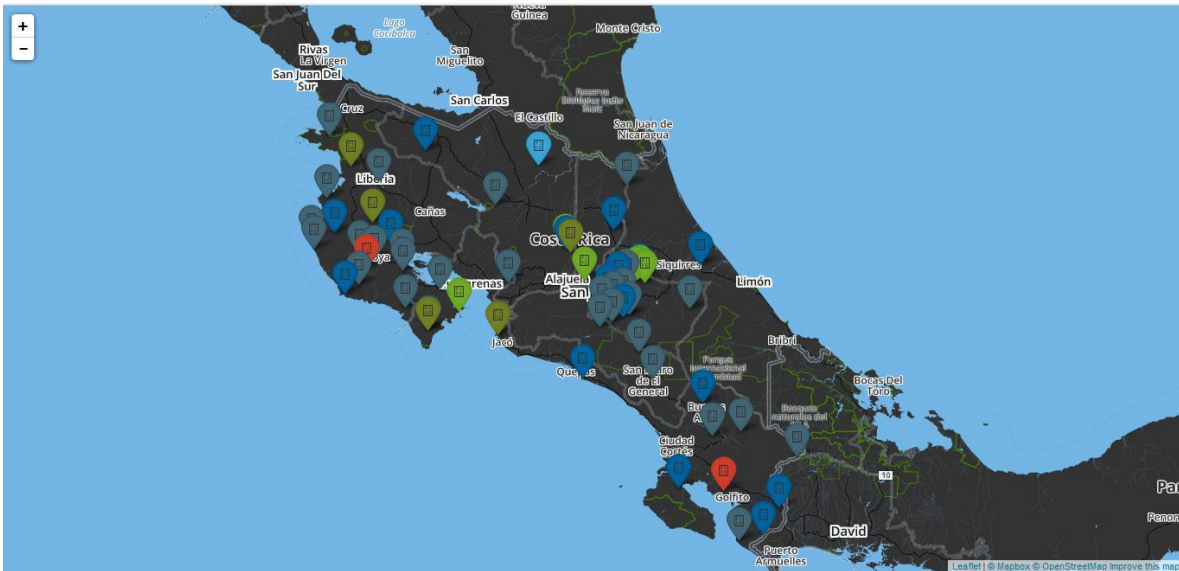


Figura 3.4. Captura de pantalla de una versión preliminar utilizando marcadores con cambio de color

Con respecto a estas escalas se deben considerar lo siguiente:

- Los colores se actualizan constantemente de acuerdo con los registros recientes de cada estación, para el caso de sismógrafos se actualiza siempre que se recibe un paquete. Para el caso de acelerógrafo se actualiza cada minuto, ya que se requiere obtener la aceleración máxima en un espacio amplio de tiempo para definir el valor de intensidad.
- Para el caso de acelerógrafos donde se muestra la intensidad se utilizan las ecuaciones 2.2.1 y 2.2.2 para convertir de aceleración máxima a un valor de intensidad.
- Una estación en color negro indica que el último registro disponible de la estación es mayor a un minuto de antigüedad.
- Una estación transparente indica que la estación está en un estado de desconexión, el cual se manifiesta cuando no hay información de la estación o su último registro almacenado es mayor a los 10 minutos.

En la siguiente sección se discuten los resultados obtenidos con respecto a esta funcionalidad.

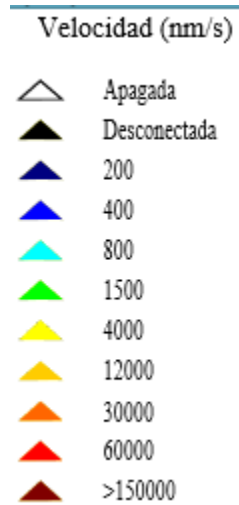


Figura 3.5 Escala de velocidad utilizada para sismógrafos



Figura 3.6 Escala de intensidades utilizando acelerógrafos

3.5.Elaboración de algoritmos de interpolación y animación de intensidades

3.5.1.Detección del sismo:

Para realizar la animación de intensidades, se tuvo que codificar varios métodos tanto en el módulo de recepción de datos como en la aplicación web desarrollada. Primeramente se tuvo que definir alguna metodología para determinar cuándo ocurrió un sismo. Las opciones para este caso se discuten a continuación:

Algoritmo de monitoreo STA-LTA:

Este algoritmo consiste en calcular un promedio en una ventana pequeña de tiempo y dividirlo entre el promedio de una ventana más grande de tiempo. Si el resultado de esta división es mayor a un factor de 3 indica que la estación registró un sismo. Utilizar este método era algo impreciso y le introducía mucho procesamiento al módulo de recepción de datos, de modo que se descartó esta opción.

Uso del epicentro:

La segunda opción para determinar cuándo ocurrió un sismo era confiar en los mismos datos que se almacenan en la red de OVSICORI, donde se utilizan herramientas y métodos precisos para determinar cuándo ocurrió un sismo y la ubicación de donde ocurrió. La información sobre un sismo es guardada en un archivo JSON, el cual almacena información del sismo sobre ubicación del epicentro, hora, localidad, profundidad, entre otros. De modo tal que para comprobar si han ocurrido sismos recientes todo lo que hay que hacer es comprobar si el archivo JSON ha sido actualizado.

Para realizar esta comprobación se escribió un código en C el cual utilizando la función `system`, permite ejecutar comandos de consola Linux. En este caso el comando que se utilizó es el siguiente:

wget -N -P [dirección donde guardar el archivo] [dirección web del archivo a descargar] 2> [dirección y nombre del archivo donde guardar el resultado del comando]

Donde:

Wget: es el comando utilizado para descargar el archivo JSON con los epicentros.

-N: comprueba la marca temporal, de modo que compara si el archivo que se va a descargar y el que se ubica en la dirección donde se va a guardar el archivo son diferentes. Si la marca temporal es diferente, el archivo se descarga, en caso contrario no se descarga el archivo. En caso de no haber un archivo en la dirección donde se guardará el archivo, se descarga el archivo de la dirección web y se salva en la dirección indicada. La comparación que se realiza con la opción -N es útil ya que ahorra ancho de banda al no tener que estar descargando en cada comprobación el archivo.

-P: Indica la dirección a donde se descargará el archivo.

2>: Redirige la salida o resultado del comando al archivo que se le indique.

El comando se ejecuta cada minuto y utilizando el archivo donde se guardó el resultado, se determina si efectivamente se descargó una nueva copia del archivo JSON o si no hay cambios en el archivo.

En caso de determinarse que hubo un cambio se procede a ejecutar las siguientes funciones:

- Se comprueba la magnitud del sismo:

Si el sismo es de una magnitud mayor a 2.5 grados en la escala Richter, se almacena la información del evento en un JSON que lee la aplicación. Este JSON se diferencia del original en que este descarta sismos de magnitudes muy bajas, sin embargo mantiene toda la información del evento de acuerdo con el archivo original. Una vez realizadas las operaciones de este paso se procede al siguiente paso

- Obtener los datos de cada acelerógrafo a lo largo de la duración del sismo:

Para obtener estos datos se debe utilizar la biblioteca libslink utilizando la opción de ventana, donde se indica el tiempo inicial y el tiempo final del cual se consultarán datos. El tiempo inicial se obtiene con el epicentro del sismo, sin embargo para obtener el tiempo final se definió unas ventanas de tiempo, las cuales de acuerdo con la magnitud del sismo definen cuanto va a durar el sismo. En la tabla 3.1 se pueden observar estas ventanas de tiempo.

Magnitud	Ventana (s)
2.5	160
4.0	220
4.5	330
5.5	380
6.0	500

Tabla 3.1 Ventana de tiempo de acuerdo con la magnitud del sismo

En caso de ser mayor la magnitud a 6.0 la ventana que se utiliza es de 500 segundos.

Todos los datos que se van obteniendo de la consulta al servidor SeedLink de OVSICORI son guardados en arreglos de tipo flotante, los cuales una vez terminada la consulta son utilizados en los siguientes pasos:

- Cálculo de aceleración máxima por segundo

En este paso por cada estación se utiliza el arreglo de datos obtenido del paso anterior. El objetivo de este paso es determinar por cada segundo cual fue la aceleración máxima registrada hasta ese momento. Por ejemplo se recibe el arreglo de datos del acelerógrafo de Jacó de un sismo de magnitud 3. Para las primeras 200 muestras (1 segundo), la aceleración máxima fue de 5 cm/s², este resultado se almacena en la posición 0 del arreglo. En la siguiente muestra la aceleración máxima registrada es de 10 cm/s², la posición 1 del arreglo de salida es 10 cm/s². En la tercera iteración la máxima aceleración registrada es 9 cm/s², puesto que para la animación de intensidad se debe mantener el registro de intensidad máxima sentida, se puede observar que el valor de 9 cm/s² no supera al anterior, de modo que la posición 2 del arreglo de salida va a tener un valor de 10 cm/s². Este proceso se va a repetir hasta cumplir las 160 iteraciones (160 segundos de la ventana de acuerdo con la tabla 3.1).

Una vez se hayan registrado los valores de aceleración máxima se procede a realizar el próximo paso.

- Cálculo de intensidad máxima y elaboración de arreglo de salida

En este paso se crea un arreglo final, el cual va a acomodar los datos de cada estación del paso anterior en un arreglo final en el cual los datos se puedan acceder por medio del tiempo. Por ejemplo se toman los arreglos del paso anterior de los 21 acelerógrafos de OVSICORI. Luego se crea un arreglo de 21 posiciones, por cada estación se toma la posición 0 del arreglo anterior y se llena este nuevo arreglo con los valores de aceleración en la posición 0 de cada estación. Siguiendo el ejemplo anterior se obtiene un arreglo de 2 dimensiones de 21x160, donde si se toma la posición 120 del arreglo este retorna las aceleraciones máximas registradas de los acelerógrafos para el segundo 119 del sismo que ocurrió. También es necesario resaltar que cuando se realiza este paso se hace el cálculo de intensidad utilizando las ecuaciones 2.2.1 y 2.2.2, de modo que la información que contiene este arreglo de salida no es de aceleraciones máximas sino de intensidad máxima. Una vez realizado este paso se procede a realizar la siguiente función

- Almacenamiento del evento

En este paso se utiliza la información del arreglo y se guarda en un archivo JSON, el cual va a ser leído por la aplicación cuando se quiera ver la animación del sismo. El formato de este JSON es el siguiente:

```
{"quake": [{"intensity": []}, {"intensity": []}, {"intensity": []}, ...]}
```

Donde:

Quake: Es la llave para un arreglo de un tamaño N, donde N se obtiene con la ventana de tiempo del sismo definida en la tabla 3.1

Intensity: Es la llave de un arreglo de 21 posiciones que contiene la información de la intensidad máxima registrada para un segundo n.

En la sección de apéndices se puede observar los diagramas de flujo de las funcionalidades de monitoreo del evento y obtención de intensidades del sismo.

3.5.2. Animación del sismo:

Aspecto gráfico:

Con respecto a la animación del sismo, inicialmente se desarrolló un código utilizando WebGL. Donde se utilizó un arreglo de 110x240 posiciones en el cual cada posición representa el valor de color de un vértice. Es necesario recalcar que WebGL utiliza el estándar de OpenGL, de modo que para dibujar diferentes figuras se utilizan triángulos, cada triángulo está formado por 3 vértices. Cada vértice necesita la posición x,y y el color de este vértice, de modo que el arreglo de 110x240 representa los colores de los vértices.

Tomando en cuenta este factor, se requirió definir un arreglo de las posiciones x y un arreglo de las posiciones y. Estas posiciones representan longitudes (posición x) y latitudes (posición y), las cuales están separadas por una diferencia de 0.03 grados de latitud y longitud. La posición inicial en latitud y longitud corresponden a 8.00° latitud norte y 82.54° longitud oeste.

Para crear estos arreglos se diseñó un código escrito en C, el cual de forma recursiva crea un arreglo cuyas posiciones horizontales se van agregando de la siguiente forma: 82.54, 82.54, 82.57, 82.57, 82.60, 82.60,..., hasta repetirlo 240 veces. Este mismo arreglo se repite 110 veces y se guarda en un archivo que representará los vértices de longitudes. De la misma forma se creó un código en C para el arreglo de latitudes, el cual se llena de la siguiente manera: 8.00, 8.03, 8.00, 8.03,..., hasta repetirlo 240 veces. La siguiente iteración llenaría el arreglo de la siguiente forma 8.03, 8.06, 8.03, 8.06,..., 11.27, 11.3. Este arreglo se guarda en un archivo para ser leído luego por WebGL.

A continuación se muestra un ejemplo para comprender como utilizaría WebGL los archivos:

Arreglo de vértices de longitudes: [82,54, 82,54, 82,57, 82,57, 82,60, 82,60, 82,54, 82,54, 82,57, 82,57, 82,60, 82,60]

Arreglo de vértices de latitudes: [8.00, 8.03, 8.00, 8.03, 8.00, 8.03, 8.03, 8.06, 8.03, 8.06, 8.03, 8.06]

Arreglo de vértices de colores: [1,2,3,4,5,6,7,8,9,10,11,12]

La figura 3.7 muestra cómo se formarían los triángulos en WebGL a partir de los puntos del ejemplo.

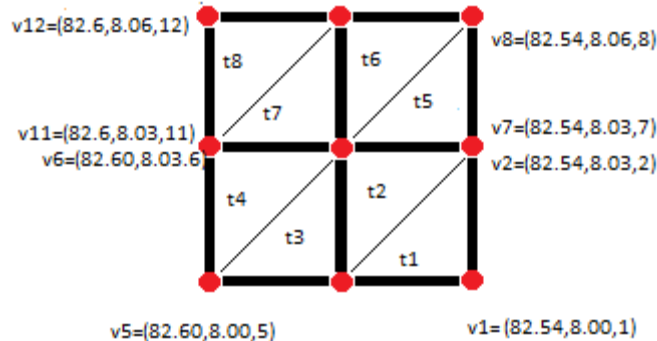


Figura 3.7 Ejemplo de uso de vértices para formar triángulos para WebGL

Se puede observar que a partir de 12 puntos se forman 8 triángulos, sin embargo la imagen no muestra los triángulos que se forman entre los puntos v5,v6,v7 y v6,v7,v8. Estos triángulos se pueden ignorar por técnicas como degeneración de triángulos o bien indicándole a WebGL que les dé una transparencia de 0, de modo tal que no sean visibles para el usuario. En la sección de resultados se discuten los resultados de este enfoque.

Interpolación:

Una vez definidos los algoritmos necesarios para dibujar los triángulos en la pantalla de la aplicación, se procedió a programar un algoritmo de interpolación el cual dado un conjunto de puntos pudiese interpolar el valor del resto de posiciones. Estos valores van a representar el color que va a tener cada uno de los vértices del triángulo.

En la sección 2.5 se presentaron 2 posibles algoritmos a utilizar. El método de Kriging y el algoritmo IDW.

Se programó tanto el algoritmo de Kriging como el algoritmo IDW, sin embargo se decide utilizar el algoritmo IDW ya que es muy rápido para generar una solución y requiere menos operaciones que el algoritmo de Kriging. Inicialmente los algoritmos se programaron en el lenguaje C, ya que se pensó en que el algoritmo de interpolación corriera en el módulo de recepción de datos para que la aplicación web no tuviese que realizar cálculos.

Partiendo de esta solución, los resultados de la interpolación se almacenaban en un archivo el cual era leído por la aplicación, sin embargo esta solución tenía el defecto que para almacenar la información de una sola interpolación se requería 2 Mb de espacio, de modo que para almacenar al menos 160 interpolaciones se iba a requerir una cantidad excesiva de espacio en el disco duro. Por esta razón se decidió pasar el algoritmo a JavaScript, para que este se ejecutara directamente en la aplicación del usuario.

Para aplicar el algoritmo se requirió utilizar las ecuaciones 2.5.1 y 2.5.2, donde los puntos conocidos eran los 21 acelerógrafos de OVSICORI. En la sección de resultados se discuten aspectos del algoritmo como tiempo de ejecución y precisión. Así mismo en la sección de apéndices se puede observar el diagrama de flujo de este algoritmo

Animación:

Con respecto a la animación, se tuvo que programar algunos elementos gráficos de control como botón de pausa, botón de siguiente y previo para detener la animación y poder observar los cambios manualmente. También se agregó una barra de desplazamiento para moverse en diferentes secciones de la animación y un cuadro que muestra el momento temporal que se está representando en la animación por ejemplo 12 de setiembre 2015, 14:04:06.

Por otro lado, a partir de los datos de la interpolación y haciendo una relación con respecto a las intensidades, se determinó una escala de colores que se puede observar en la figura 3.8. Con esta escala de colores se estableció una relación entre color e intensidad. Por ejemplo una zona que muestra un color rojizo fue donde el sismo tuvo mayor efecto y una zona con color azul oscuro o negro fueron zonas donde el sismo no fue sentido.



Figura 3.8. Escala de colores de intensidades para la animación

Una vez definidos estos elementos, se programó los métodos necesarios para leer los archivos de intensidades descritos en la sección 3.4.1. Estos datos pasan a ser utilizados por el algoritmo de interpolación, donde los resultados de interpolación obtenidos representan la intensidad de un sismo a lo largo del país para un momento de tiempo específico. En la siguiente sección se discuten los resultados obtenidos de esta animación tales como la continuidad, realidad del resultado obtenido y sus limitaciones.

4. Análisis de resultados

4.1. Elaboración de la aplicación

Con respecto a las decisiones tomadas sobre las bibliotecas utilizadas para diseñar la aplicación, se considera que Leaflet fue una buena decisión en cuanto a funcionalidades. Con esta herramienta se logró cargar diferentes mapas de acuerdo con las funcionalidades requeridas. Como se puede observar en las figuras 4.1 y 4.2 la animación de intensidad y la funcionalidad de monitoreo de estaciones muestran 2 mapas distintos. Realizar este cambio fue sencillo ya que solo significó cambiar la ruta donde se encuentra el mapa, esto demuestra la facilidad para realizar cambios en la herramienta.

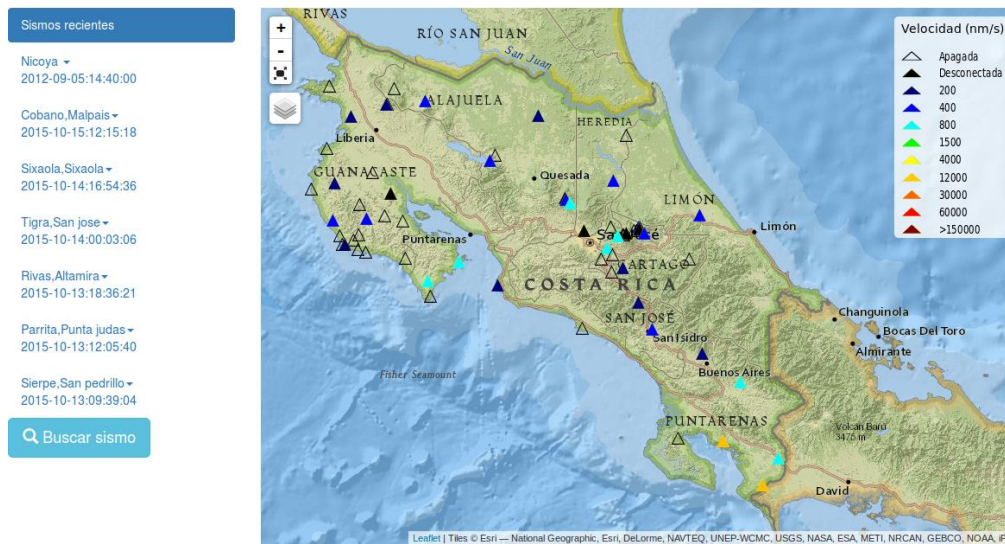


Figura 4.1 Herramienta web funcionalidad de monitoreo

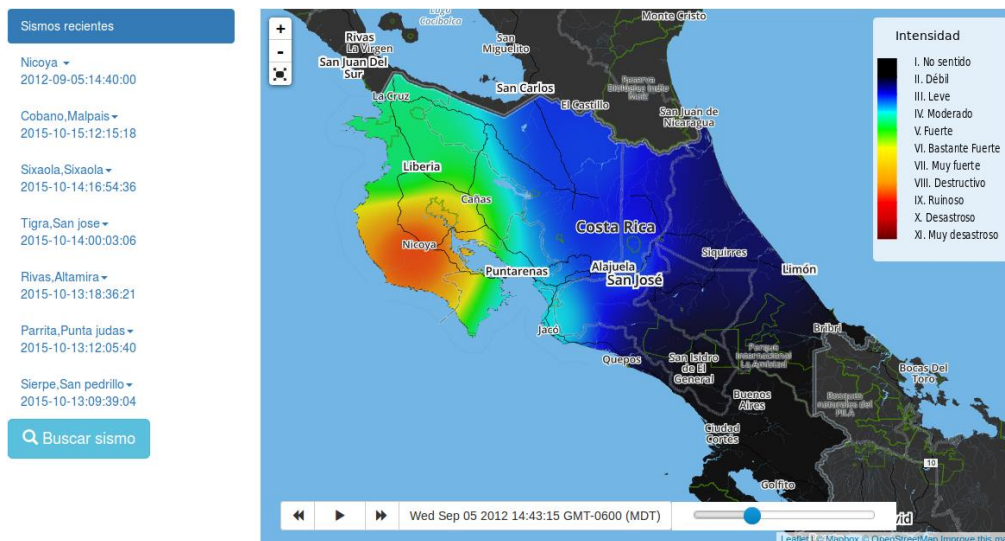


Figura 4.2 Herramienta web funcionalidad de simulación

Por otro lado la herramienta permite definir funcionalidades como zoom in y zoom out y utilizar pantalla completa, estas funcionalidades se encuentran en los controles de la esquina superior izquierda de la figura 4.1. La aplicación permite utilizar imágenes y cargarlas en el mapa tal y como se observa en las leyendas de la esquina superior derecha de las figuras 4.1 y 4.2. También se puede observar que los marcadores de las estaciones pueden editarse y se puede utilizar imágenes creadas en programas como Inkscape.

Así mismo se puede observar que la aplicación permite agregar controles para poder cargar diferentes capas del tipo GeoJSON. Para este proyecto solo se utilizó 2 capas de puntos, sin embargo la biblioteca permite cargar capas geométricas. En las figuras 4.3 y 4.4 se puede observar como en un mismo mapa utilizando el control de la esquina superior derecha se puede cambiar de la capa de sismógrafos a la capa de acelerógrafos.



Figura 4.3 Herramienta web mostrando los sismógrafos



Figura 4.4 Herramienta web mostrando acelerógrafos

Finalmente con respecto al uso de Bootstrap, la biblioteca cumplió con las expectativas. Donde se esperaba utilizar la biblioteca para elaborar controles de navegación para el usuario final como menús y controles. Se puede destacar que la herramienta también se puede integrar a los mapas de Leaflet, donde en la figura 4.2 el controlador de la esquina inferior izquierda fue diseñado utilizando las herramientas de Bootstrap y métodos como innerhtml. Esta facilidad de poder integrar las 2 bibliotecas fue muy favorable, ya que permitió mejorar las funcionalidades de la aplicación utilizando controles fáciles de entender para el usuario.

La aplicación se comprobó funcionando tanto en sistemas operativos Linux Ubuntu 14.04 y Windows 8.1 utilizando los navegadores Chrome y Firefox. De modo que la aplicación carga en diversos sistemas operativos y navegadores sin ningún problema. Por otro lado el utilizar Bootstrap y Leaflet permite que la aplicación sea visualizada en dispositivos móviles ya que las bibliotecas tienen la característica de ser adaptables (responsive).

4.2.Recepción de datos y almacenamiento

4.2.1.Recepción de datos

La biblioteca Libslink cumplió con la expectativa, donde se esperaba que al utilizar esta biblioteca se pudiesen recibir los datos desde las estaciones de OVSIORI. La biblioteca puede manejar el volumen de datos de los canales HHZ y HNZ con frecuencias de muestreo de 100 y 200 muestras por segundo respectivamente.

Es necesario resaltar que a pesar de que un canal HHZ tiene una frecuencia de 100 muestras por segundo, esto no implica que en cada paquete que se recibe, la cantidad de muestras será 100. En la figura 4.5 se puede observar parte del resultado de un paquete guardado en un archivo JSON.

```
{ "settings": { "Station": "BATAN", "channel": "HHZ" }, "features": [ { "type": { "properties": { "Samples": 209 }, "timestamp": { "year": 2015, "day": 286, "hour": 18, "minutes": 38, "seconds": 38, "fract": 158 }, "amplitudes": [-449.003571, -391.305908, -330.623627, -268.945984, -207.720245, -148.199097, -91.486343, -38.353703, 10.207454, 52.787098, 88.198586, 115.367096, 133.102448, 141.172913, 141.049698, 134.580826, 122.739616, 106.548286, 87.721558, 67.490059, 46.105347, 23.696507, 0.238406, -24.831205, -51.496639, -78.854340, -106.620613, -134.830963, -162.102631, -186.519791, -207.602570, -225.468994, -239.151611, -247.066513, -247.696991, -239.039688, -218.993088, -186.881210, -143.310608, -88.636070, -23.445911, 49.536030, 126.096069,
```

Figura 4.5 Paquete recibido de un sismógrafo guardado en formato JSON

Se puede observar que la cantidad de muestras en este paquete fue de 209, lo que indica que este paquete representa una franja de tiempo de poco más de 2 segundos. Esto se debe a que el servidor SeedLink envía información cada vez que se llenan los 512 bytes del paquete MiniSEED y no de una forma periódica cada segundo.

4.2.2.Almacenamiento

Con respecto a la estrategia de almacenamiento, anteriormente se mencionó que se utilizarían archivos con estándar JSON. Sin embargo para establecer un límite en la cantidad de paquetes que iban a guardar estos archivos se decidió utilizar 2 bibliotecas las cuales permitieran leer el archivo, parsear el objeto JSON y analizar si estos requerían eliminar un paquete del objeto JSON. Una vez realizados estos cambios, el objeto se guarda en el archivo. Para realizar esto se utilizó las bibliotecas Jsmn y Jansson, a continuación se describen los resultados obtenidos para cada uno de los casos:

- Jsmn

Inicialmente se diseñó el programa con esta biblioteca, sin embargo su utilidad era buena en archivos JSON pequeños. En un objeto JSON como el que se propuso en la sección 3.2.2 no solo están presentes las 600 muestras. Si se observa la figura 4.5 hay muchos otros elementos guardados como el nombre de la estación, nombre del canal, timestamp, cantidad de muestras y arreglo de muestras.

Utilizando esta biblioteca y basándose en el código ejemplo que brinda la biblioteca, para analizar un solo elemento del JSON se requería definir un arreglo con una cantidad fija de llaves (de tipo jsmntok_t). Para el caso de un solo objeto como el de la figura 4.5, se requería definir un arreglo 200 llaves, de modo que para analizar un archivo de 600 muestras, el arreglo debía tener un tamaño excesivamente grande, lo cual consumiría mucha memoria en el computador que corra el módulo de recepción de datos. Esta fue la razón principal por la cual se descartó el uso de esta biblioteca.

- Jansson

Usar esta biblioteca fue una buena opción ya que tenía un mejor uso de la memoria del sistema en comparación al caso de Jsmn. Usando el comando top para visualizar el consumo de recursos, se pudo observar que la cantidad de memoria que utiliza el módulo de recepción de datos utilizando esta biblioteca es del 0.1%.

```
top - 15:17:27 up 3:33, 3 users, load average: 1,03, 0,91, 0,84
Tasks: 268 total, 1 running, 267 sleeping, 0 stopped, 0 zombie
%Cpu(s): 6,4 us, 1,1 sy, 0,0 ni, 91,0 id, 1,5 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem: 8086288 total, 6354316 used, 1731972 free, 165264 buffers
KiB Swap: 8296444 total, 0 used, 8296444 free, 2602728 cached Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
8244	alejand+	20	0	1701348	629528	96452	S	25,9	7,8	13:56.12	firefox
6834	alejand+	20	0	1264968	504284	107700	S	17,6	6,2	8:09.19	chrome
4156	alejand+	20	0	2935700	924932	159760	S	4,7	11,4	12:36.84	chrome
2533	alejand+	20	0	997424	175756	82620	S	4,3	2,2	10:03.62	chrome
9983	alejand+	20	0	96012	4604	2152	S	2,3	0,1	0:11.02	slclient
1524	root	20	0	572016	92636	75640	S	2,0	1,1	7:01.91	Xorg
2269	alejand+	20	0	1542492	106716	57888	S	2,0	1,3	4:31.86	compiz
3338	alejand+	20	0	6569560	282264	26328	S	1,7	3,5	2:35.20	java
2498	alejand+	20	0	1222556	265768	105736	S	1,3	3,3	7:42.12	chrome
6857	alejand+	20	0	894488	119940	38052	S	1,3	1,5	1:10.34	chrome
1146	mysql	20	0	492652	53612	10696	S	0,3	0,7	0:06.66	mysqld
2054	alejand+	20	0	437124	8940	5556	S	0,3	0,1	0:11.40	ibus-daemon
2137	alejand+	20	0	552404	26500	16944	S	0,3	0,3	0:04.33	bamfdaemon
2288	alejand+	20	0	1405856	57184	36420	S	0,3	0,7	0:33.86	nautilus
7186	root	20	0	0	0	0	S	0,3	0,0	0:03.24	kworker/u1+
8125	alejand+	20	0	650816	29656	23240	S	0,3	0,4	0:08.06	gnome-term+
10441	alejand+	20	0	29152	3340	2744	R	0,3	0,0	0:00.02	top

Figura 4.6 Comando top, donde slclient es el módulo de recepción de datos

Por otro lado la cantidad de funciones que ofrece esta biblioteca es bastante amplia. La biblioteca permite parsear todo un objeto JSON y retornar los elementos del objeto de acuerdo con los tipos diferentes de datos como string, float, integer, arreglos, entre otros. Por otro lado la biblioteca tiene funciones para obtener el tamaño de un arreglo y remover un elemento del objeto dada una posición de un arreglo o ingresando el valor de la llave que se quiere remover. Así mismo la biblioteca ofrece un método que permite convertir un objeto JSON a un string, lo cual va a ser muy útil, ya que esto permite guardar el string en un archivo que va a ser leído fácilmente por la aplicación en JavaScript.

El uso de estas funciones fue de gran utilidad, ya que con esta biblioteca se logró mantener el tamaño de los archivos de cada estación a 600 paquetes por archivo. De modo tal que el algoritmo que ejecuta el módulo de recepción de datos al recibir un paquete es el siguiente:

- I. Primeramente se recibe un paquete de datos de una estación
- II. Se abre el archivo que contiene los paquetes guardados de la estación.
- III. El resultado es guardado en un string
- IV. El string es enviado a la biblioteca Jansson, la cual parsea el string
- V. Utilizando los métodos de la biblioteca se determina la cantidad de paquetes que hay guardados en el objeto.
- VI. Si la cantidad supera las 600 muestras (aproximación de 10 minutos), se elimina el último paquete del objeto que representa la muestra más vieja y se agrega la nueva muestra
- VII. Se utiliza el método de la biblioteca para convertir el objeto JSON en string
- VIII. El string obtenido es guardado en un archivo

Un factor negativo encontrado utilizando esta metodología, fue que al utilizar las funciones de la biblioteca en un archivo muy grande, se produce un alto consumo de recursos del procesador. Para este módulo se requiere que el consumo del procesador sea bajo. Este es un factor crítico de la funcionalidad ya que el computador que va a correr este programa va a estar ejecutando otros servicios, de modo que no es posible que la mayor parte de sus recursos se utilicen en un solo programa. En la figura 4.7 se puede observar con el comando top como el módulo de recepción de datos ocupa el 95.8% del procesador para ser ejecutado (proceso slclient).

```

top - 15:34:59 up 3:51, 3 users, load average: 1,30, 0,95, 0,90
Tasks: 266 total, 2 running, 264 sleeping, 0 stopped, 0 zombie
%Cpu(s): 16,4 us, 1,6 sy, 0,0 ni, 81,0 id, 1,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem: 8086288 total, 7147904 used, 938384 free, 168212 buffers
KiB Swap: 8296444 total, 0 used, 8296444 free. 2902836 cached Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+  COMMAND
11034 alejand+  20   0   28300   14436   1880  R   95,8   0,2   0:13.68  slclient
 8244 alejand+  20   0 1691356 588888   96452  S   17,3   7,3  16:37.82  firefox
 4156 alejand+  20   0 3646608 1,315g 163216  S   11,6  17,1  14:45.56  chrome
 2533 alejand+  20   0 1053564 235396  134628  S    5,3   2,9  10:48.52  chrome
 2498 alejand+  20   0 1217496 282460  106728  S    5,0   3,5   8:19.73  chrome
 2269 alejand+  20   0 1542688 106828   58000  S    4,7   1,3   5:07.10  compiz
 1524 root      20   0   564428  86528   69532  S    3,7   1,1   7:35.97  Xorg
 6834 alejand+  20   0 1255496 492744   99148  S    3,3   6,1   9:40.11  chrome
 3338 alejand+  20   0 6569560 282148   26328  S    1,0   3,5   2:48.62  java
 6857 alejand+  20   0  894488 119920   38052  S    0,7   1,5   1:19.81  chrome
 2014 alejand+  20   0   40360   3772    2180  S    0,3   0,0   0:09.02  dbus-daemon
 2112 alejand+  20   0  575364  36200   22528  S    0,3   0,4   0:26.35  unity-pane+
 8125 alejand+  20   0  651496  30080   23240  S    0,3   0,4   0:14.19  gnome-term+
 9937 root      20   0     0     0     0  S    0,3   0,0   0:01.18  kworker/u1+
10441 alejand+  20   0  29152   3340   2744  R    0,3   0,0   0:01.55  top
 1 root      20   0   33764   4360   2736  S    0,0   0,1   0:01.05  init
 2 root      20   0     0     0     0  S    0,0   0,0   0:00.00  kthreadd

```

Figura 4.7 Comando top para una versión previa del módulo de recepción de datos utilizando las funciones de Jansson

De este modo se tuvo que definir una opción alternativa para no tener que estar utilizando constantemente los métodos del Jansson (parser, tamaño, eliminar elemento, conversión JSON a string), ya que este problema se presentaba debido al uso constante de las funciones de la biblioteca (sin la estrategia de corrección, los métodos se ejecutaban cada vez que se recibía un paquete de datos).

La opción de usar Jansson fue buena por su buen uso de memoria y cantidad de funciones para analizar un objeto, de modo que la biblioteca se mantuvo en el programa. Sin embargo se redujo el uso de funciones de la misma a solo tener que utilizar el parser para saber la cantidad de muestras guardadas de cada estación en cada archivo al iniciar el programa, de modo que las funciones de la biblioteca se utilizarán únicamente al iniciar el programa. La solución utilizada fue la siguiente:

- I. Se inicia el programa
- II. Al recibir un paquete de una estación se verifica por medio de una variable, si ya esta estación ha sido parseada con anterioridad (al inicio del programa se toma que ninguna de las estaciones ha sido parseada). Con respecto a los resultados de esta comparación se presentan las siguientes 2 situaciones:
 - a. En caso de no haber sido parseada se utiliza los métodos de Jansson para definir la cantidad de paquetes guardados en el archivo. Este valor es guardado y se utilizará como un contador
 - b. Si ya ha sido parseada se consulta el valor del contador del paso a

III. Si el contador es igual a 600 se procede a realizar lo siguiente:

Anteriormente para eliminar una muestra se utilizaba el método de eliminar elementos de Jansson, la nueva solución plantea lo siguiente: Las muestras del archivo se almacenarán en un archivo con extensión 0.json, al llegar a contener 600 muestras los datos se trasladan a un archivo con extensión 1.json y el archivo 0.json vuelve a estar vacío.

Por ejemplo la estación de Jacó cuenta con 599 muestras, estas están almacenadas en el archivo Jaco0.json. Al llegar un nuevo paquete se almacena en el archivo Jaco0.json y el contador aumenta a 600. Luego llega la muestra 601 al entrar esta, el programa se entera que la cantidad de muestras en el archivo es 600, de modo que los datos son ahora trasladados al archivo Jaco1.json y el Jaco0.json queda únicamente con el valor del paquete recién recibido.

Se puede observar que esta solución no elimina las muestras cada vez que se recibe un paquete como lo hacía la solución inicial, sino que las muestras van a ser eliminadas cuando existan tanto un archivo con extensión 0.json y 1.json. Si el archivo con extensión 0.json supera las 600 muestras sobrescribirá el archivo con extensión 1.json y el 0.json volverá a estar vacío.

Esta solución redujo el consumo de procesador de manera muy favorable, utilizando el comando top se pudo observar que el máximo que llega a utilizar la aplicación es el 14% del procesador, donde normalmente el consumo se encuentra en el 3% y 0.1% como se puede observar en la figura 4.8. Esto representa una mejora muy significativa con respecto al algoritmo inicial, donde con la nueva técnica en el peor de los casos requiere un 81.8% menos de consumo del procesador.

```
top - 15:38:30 up 3:55, 3 users, load average: 0,97, 0,95, 0,91
Tasks: 269 total, 3 running, 266 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5,0 us, 1,6 sy, 0,0 ni, 91,5 id, 1,9 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem: 8086288 total, 7183676 used, 902612 free, 168920 buffers
KiB Swap: 8296444 total, 0 used, 8296444 free. 2940576 cached Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 8244 alejand+  20   0 1687388 589920 96452 S   16,3   7,3   17:09.59 firefox
 4156 alejand+  20   0 3657760 1,327g 163924 S   12,3  17,2   15:27.60 chrome
 1524 root      20   0 566492 87184 70188 S    5,3   1,1    7:47.08 Xorg
 2269 alejand+  20   0 1542992 107176 58024 S    5,3   1,3    5:18.30 compiz
 2533 alejand+  20   0 1039904 235756 120932 S    5,3   2,9   11:00.59 chrome
 2498 alejand+  20   0 1217496 282696 106696 S    4,7   3,5    8:31.14 chrome
11139 alejand+  20   0 96252 4868 2180 S    2,7   0,1    0:04.79 slclient
 2614 alejand+  20   0 901164 177504 44328 R    1,7   2,2    1:55.71 chrome
 3338 alejand+  20   0 6569560 282148 26328 S    1,3   3,5    2:51.14 java
 6857 alejand+  20   0 894488 119920 38052 S    0,7   1,5    1:21.61 chrome
 2205 alejand+  20   0 553140 12388 8144 S    0,3   0,2    0:00.26 indicator+
 2592 alejand+  20   0 875432 142152 44340 R    0,3   1,8    1:01.47 chrome
 6834 alejand+  20   0 1255496 492256 99148 S    0,3   6,1    9:41.29 chrome
 8125 alejand+  20   0 651496 30080 23240 S    0,3   0,4    0:15.11 gnome-term+
 9757 root      20   0 0 0 0 S    0,3   0,0    0:01.54 kworker/u1+
10441 alejand+  20   0 29152 3340 2744 R    0,3   0,0    0:01.85 top
 1 root      20   0 33764 4360 2736 S    0,0   0,1    0:01.05 init
```

Figura 4.8 Resultado del comando top con modulo de recepción de datos (slclient) en su versión final

4.3.Manipulación de datos:

4.3.1.Ajuste de unidades

Con respecto a la manipulación de datos, las funcionalidades principales fueron dadas por la elaboración de algoritmos para ajustar la escala y filtrar los datos. Esto con el objetivo de obtener un sismograma donde la señal estuviese liberada del ruido producido por las fuentes mencionadas con anterioridad en la sección 2.4.2.

Con respecto al ajuste de unidades, la funcionalidad cumplió con las expectativas, donde recibido un dato desde el servidor SeedLink, se utilizaron los datos provistos de OVSICORI sobre la sensibilidad del digitalizador utilizando la ecuación 2.4.1. Con respecto a la ecuación 2.4.1 se realizó un ajuste más, se determinó que en vez de mostrarse m/s o m/s^2 , se mostraría un valor de nm/s y nm/s^2 . Este cambio se implementó ya que los valores de los registros de las estaciones son muy bajos utilizando la unidad de metro, esto se puede observar en el registro de la figura 4.16, donde el valor máximo de velocidad registrada es de 5000 nm/s.

4.3.2.Filtrado

Con respecto al filtrado se utilizaron diversas opciones, donde cada uno de los filtros mostró distintos resultados. Para evaluar cual filtro presentó los mejores resultados, el criterio utilizado fue comparar diversos sismogramas y comprobar la continuidad de la gráfica mostrada.

Reducción del offset mediante el cálculo de la media

Este tipo de filtro es muy básico y utilizando la ecuación 3.3.1 se logró eliminar el offset de las señales de las estaciones. En la figura 4.9 se puede observar un sismograma de una estación con offset y en la figura 4.10, se puede observar como la señal vuelve a estar centrada en el 0 luego de utilizar esta estrategia. Se puede observar que estrategia únicamente elimina un factor que altera la señal, de modo que las otras fuentes de ruido aún están presentes en el sismograma de la figura 4.10. Por lo tanto las otras estrategias de filtrado muestran mejores resultados con respecto a la reducción de ruido.

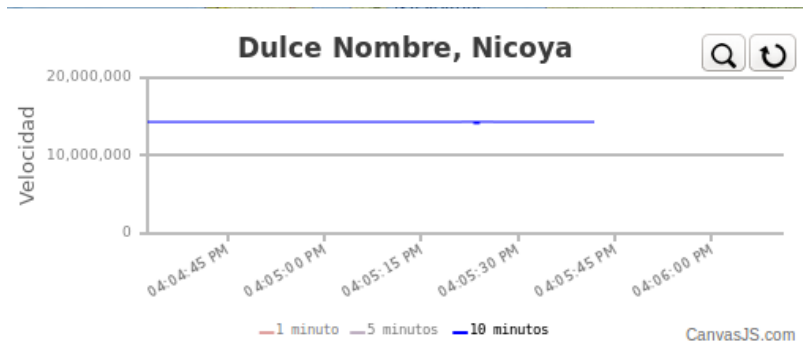


Figura 4.9 Sismograma de una estación con offset

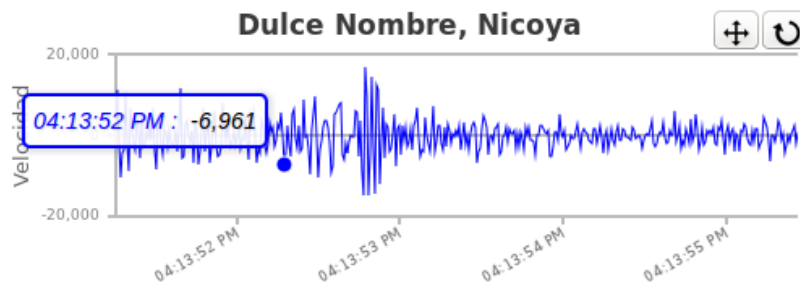


Figura 4.10 Estación ajustada luego de restar la media a la señal

Filtro pasa banda por el método de alisado exponencial

Este método tuvo condiciones favorables en remover el offset, sin embargo el ruido solo se redujo muy poco. La principal razón por la cual este filtro no cumple con las expectativas de un filtro se debe a varias razones:

- El filtro está basado en el modelo del circuito RC, el cual no es un circuito ideal para aplicar un filtro
- Este tipo de filtros no pretenden reducir ruido de una señal en frecuencias específicas, su orientación es más que todo reducir las variaciones entre un conjunto de puntos de modo tal que se obtenga una señal de salida con un aspecto más continuo.

En la figura 4.11 se observa el sismograma utilizando la técnica de alisado exponencial.

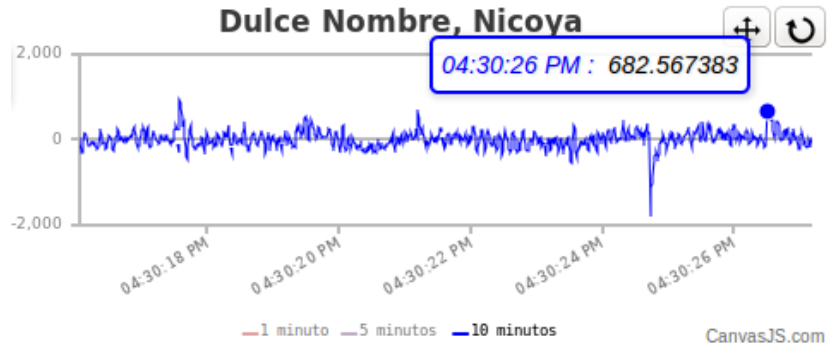


Figura 4.11 Sismograma de una estación utilizando alisado exponencial

Se puede observar comparando las figuras anteriores, que el filtro logra eliminar el offset de la señal de la estación. En comparación en el caso anterior, esta estrategia es mejor para reducir el offset, ya que lo hace de una manera más suavizada debido a que por medio del factor α , se toman en cuenta valores de muestras anteriores.

El caso anterior remueve el offset de un modo más abrupto, donde por cada paquete se calcula la media y se reduce el valor a cada una de las muestras del paquete. En el siguiente paquete se calcula una nueva media, la cual no tiene ninguna relación con el paquete anterior, de forma que entre 2 paquetes el ajuste que se aplicó no fue el mismo.

De este modo se determina que el método anterior de calcular la media es válido de utilizar si se aplica únicamente a una señal completa. Por ejemplo el método es válido si se aplica a los registros del sismo de Nicoya, en el cual se tiene un punto de inicio y un punto final, por lo cual a toda la señal se le aplicaría el mismo ajuste.

Filtro pasa banda utilizando métodos FIR e IIR

Se logró implementar un filtro FIR con la ventana de Hamming. Sin embargo los resultados obtenidos no fueron tan satisfactorios como en el caso del filtro IIR. Para el caso del filtro FIR se puede observar en la figura 4.12 como la respuesta es menos suavizada que la de la figura 4.13 la cual corresponde al resultado del filtro IIR.

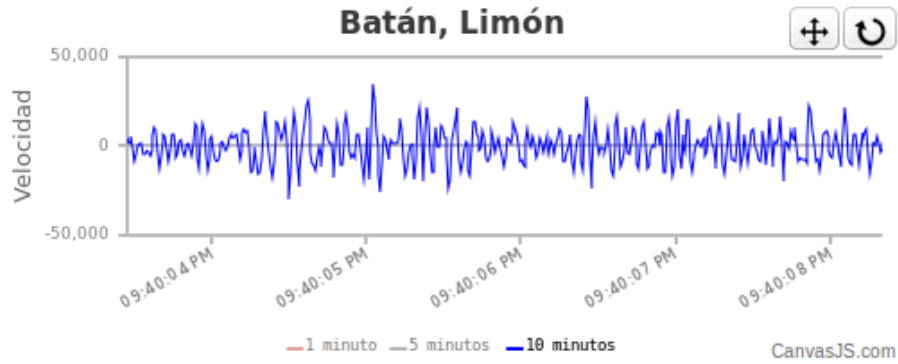


Figura 4.12 Sismograma de la estación Batán con filtro FIR

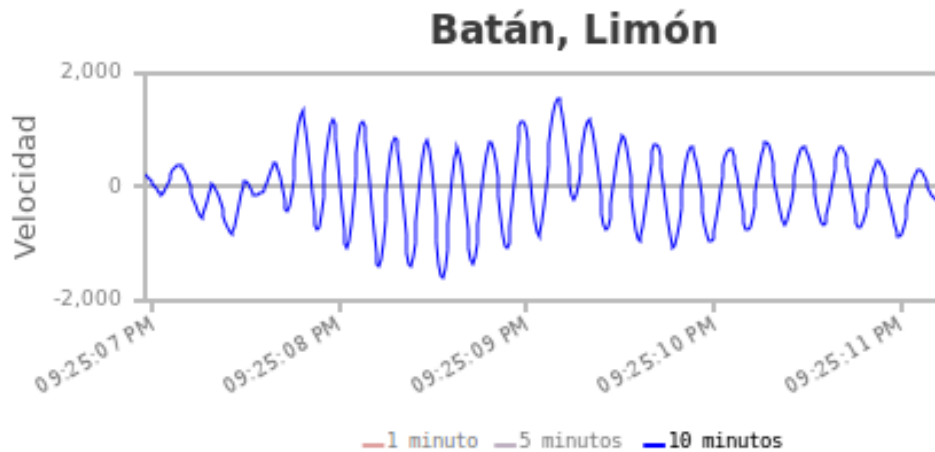


Figura 4.13 Sismograma de la estación Batán utilizando filtro IIR

Por otro lado en la figura 4.14 y 4.15 se puede observar los sismogramas para la estación de Heredia. La cual al estar ubicada en una ciudad, su registro se ve afectado por la actividad humana. Se puede observar en ambos casos que la señal obtenida está más distorsionada en comparación al resultado de las figuras 4.13 y 4.12.

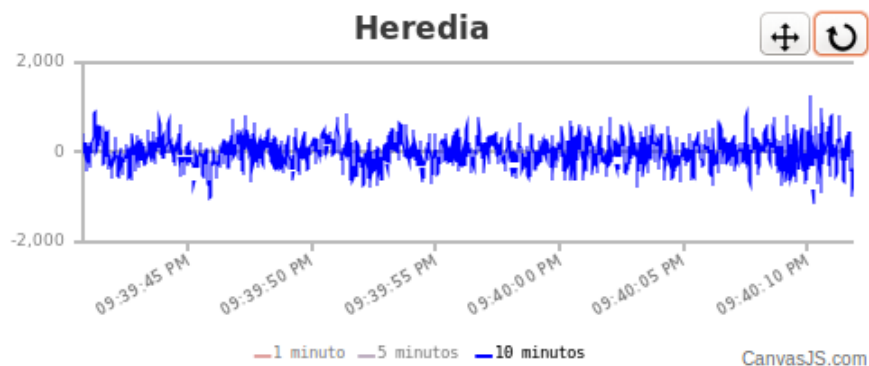


Figura 4.14 Sismograma de la estación de Heredia utilizando filtro FIR

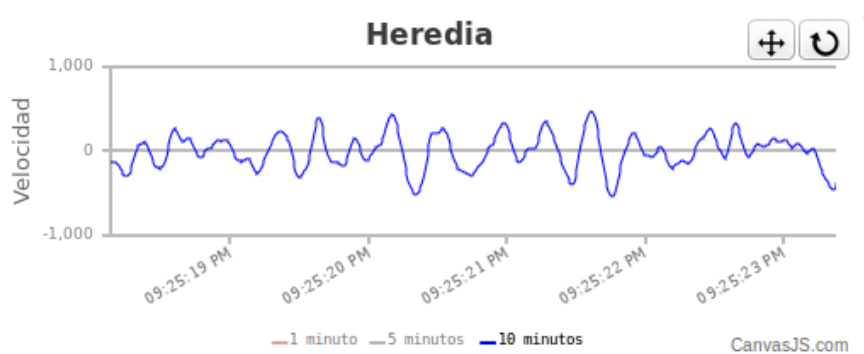


Figura 4.15 Sismograma de la estación de Heredia utilizando filtro IIR

De modo tal que se comprueba que a pesar de las técnicas implementadas para filtrar la señal, en algunas estaciones se logra obtener una señal suavizada, mientras que para otras estaciones el resultado sigue siendo alterado por la actividad humana. Se puede observar tanto en la figura 4.14 y 4.15, como nuevamente el filtro IIR obtuvo un resultado más satisfactorio en comparación al filtro FIR para filtrar la señal.

Con base en los resultados obtenidos, se decide utilizar el filtro IIR para eliminar el ruido de la estación, por otro lado se utiliza el filtro paso alto con alisado exponencial para eliminar el offset de la señal. De modo que el resultado del sismograma es la aplicación de 2 filtros, primeramente un filtro paso alto de alisado exponencial para remover el offset y luego un filtro IIR Butterworth para eliminar algunos factores de ruido sísmico que alteran la señal.

4.4. Carga de datos en la aplicación

Con respecto a la funcionalidad de monitoreo de acelerógrafos y sismógrafos, para el caso de mostrar sismogramas a partir de los registros almacenados en archivos con formato JSON se decidió utilizar 2 bibliotecas `flot.js` y `canvas.js`

Inicialmente se utilizó la biblioteca `flot.js`. Con esta biblioteca se logró cargar el dato de un paquete almacenado en los archivos JSON. Sin embargo en el momento que se comenzó a cargar archivos con gran cantidad de paquetes almacenados, se observó que la biblioteca duraba mucho tiempo para mostrar la información y paralizaba la aplicación. Se determinó que la biblioteca no era capaz de manejar el volumen de datos de la aplicación y no podía ser utilizada para este proyecto.

Se procedió a utilizar la biblioteca `canvas.js`. La biblioteca `canvas.js` presentó mejores resultados con respecto a `flot.js` ya que esta es capaz de renderizar 60000 puntos en 60 ms. Por esta razón esta biblioteca fue la utilizada como solución final para mostrar los sismogramas.

La biblioteca tiene muchas funcionalidades y permitió mejorar el aspecto gráfico del sismograma, de modo que le permite al usuario ver diferentes opciones de sismogramas en 10, 5 o 1 minuto como se puede observar en la figura 4.16. Además la biblioteca permite hacer zoom en el sismograma, lo que permite ver con mayor detalle el movimiento sísmico, permite realizar paneo para navegar en el sismograma y también permite ver el valor en el eje y de un punto, lo cual permite saber con precisión el valor de velocidad o aceleración registrado en un sismograma para un punto específico. En las figuras 4.16, 4.17 y 4.18 se pueden observar sismogramas en 10, 5 y 1 minutos.

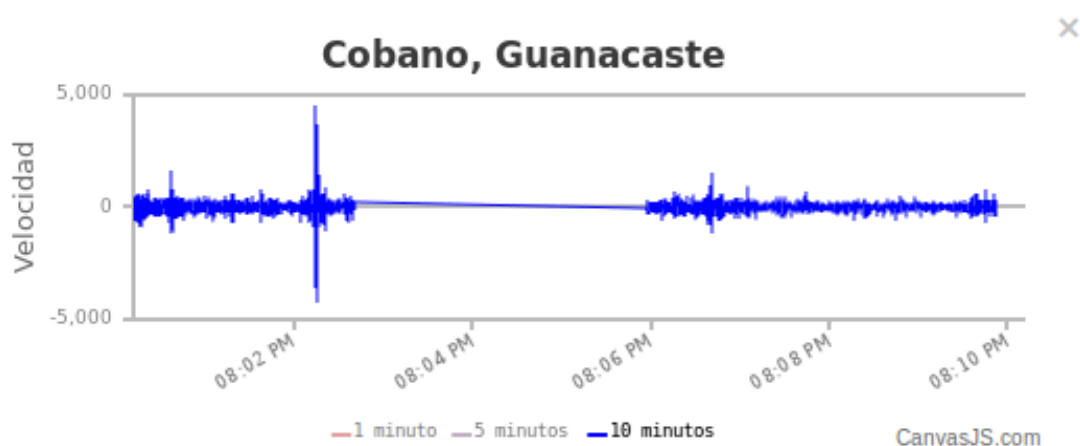


Figura 4.16 Sismograma de una estación para 10 minutos

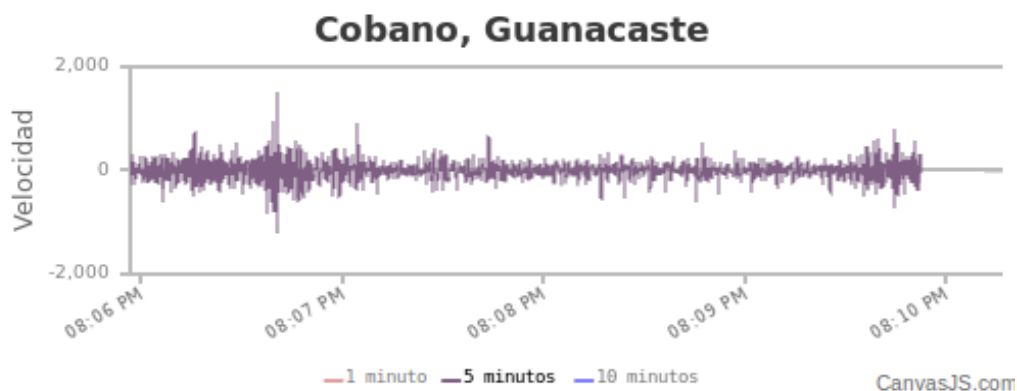


Figura 4.17 Sismograma de una estación de 5 minutos

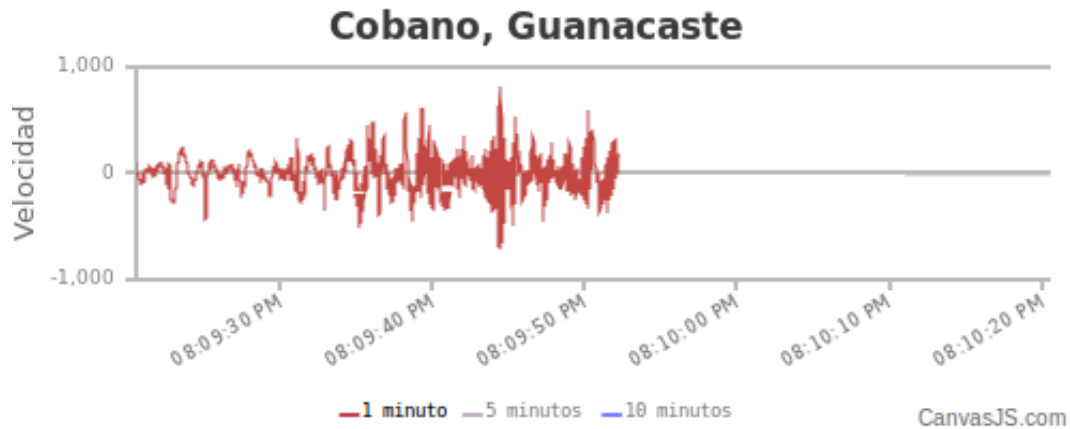


Figura 4.18 Sismograma de una estación de 1 minuto

La funcionalidad del sismograma permite comprender con mayor detalle el movimiento sísmico que sucede en el país. Todos estos sismogramas se actualizan de forma constante, de modo tal que al momento que se recibe un paquete de una estación, se puede observar el cambio en el movimiento sísmico de la estación. En la figura 4.16 se puede observar que hay un retraso entre el último punto del sismograma y el tiempo actual, el cual es representado por un espacio sin datos entre el último punto y el final del eje x del gráfico. Esto se debe a que el servidor SeedLink no envía de forma periódica datos cada segundo, sino que envía paquetes de forma asincrónica y el paquete es enviado cuando el buffer del servidor haya llenado los 512 bytes del paquete MiniSEED. En el caso de la figura 4.18, se observa un retraso de aproximadamente 30 segundos, esto se debe a que el servidor al cual se le consultaron los datos era el de IRIS, el cual presenta retrasos significativos con respecto al tiempo real.

Por otro lado la funcionalidad de cambiar el color de las estaciones cumplió con las expectativas. Utilizando la escala de colores de las figuras 3.5 y 3.6, se puede observar como el color de una estación cambia de acuerdo con el registro de las estaciones sísmicas. Con esta funcionalidad se logra dar una característica de monitoreo en tiempo real de los sensores, donde el color sirve como una advertencia sobre el movimiento sísmico que ocurre en el país. En la figura 4.19, se puede observar una captura de pantalla de esta funcionalidad.

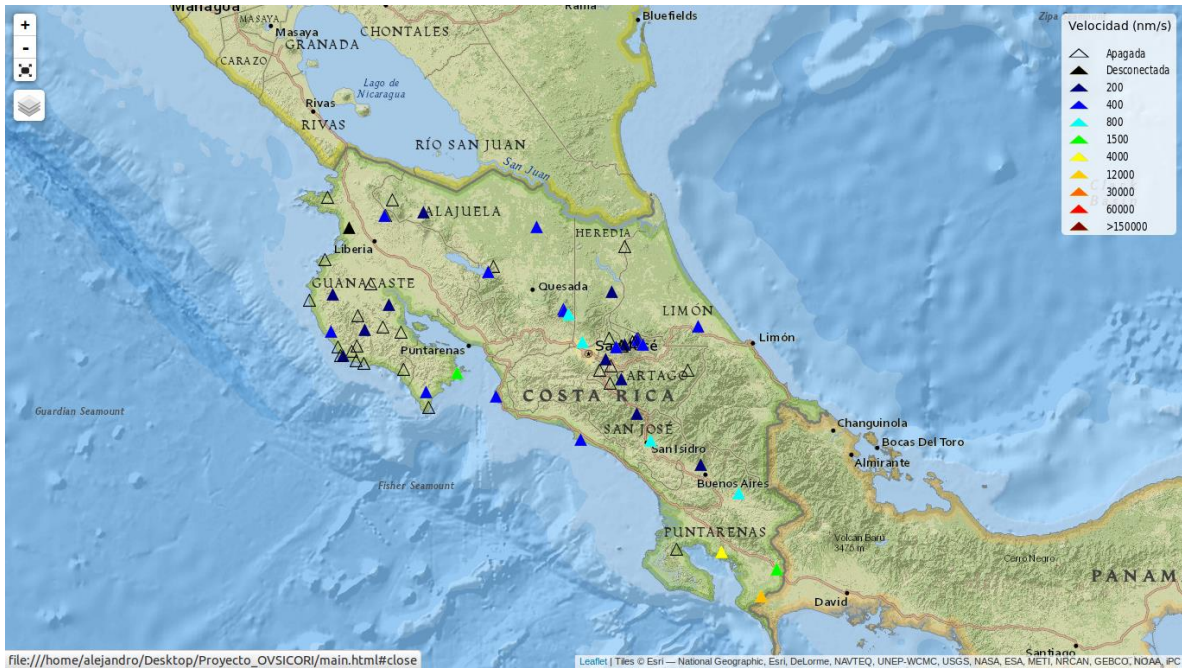


Figura 4.19 Captura de pantalla de funcionalidad de alertas por color

En la figura 4.19, se puede observar que la mayoría de las estaciones registran valores bajos de velocidad (colores azules). Otras estaciones muestran colores verdes o naranja posiblemente por estar expuestas a maquinaria que altera el registro (ruido humano). Sin embargo en caso de suceder un sismo las estaciones alcanzarían un valor de velocidad mayor a los 150000 nm/s. Tomando los datos del sismo de Nicoya, se puede observar en la figura 4.20 que la velocidad registrada por el sismógrafo supera en gran medida los 150000 nm/s, de forma que en un sismo el color de la estación de acuerdo con la escala utilizada sería del color rojo oscuro (supera 150000 nm/s). Considerando el movimiento de la onda sísmica, el comportamiento esperado de esta funcionalidad sería que una vez que el color de una estación cambia a rojo, otras estaciones alrededor y de forma radial cambien a un color rojo indicando el movimiento de las ondas P y S.

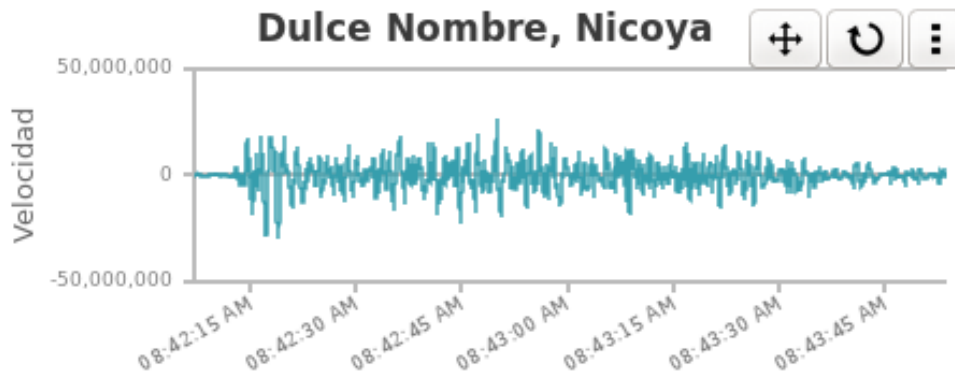


Figura 4.20 Sismograma para una estación durante el sismo de Nicoya

Con esta funcionalidad se podría observar en tiempo real cuándo sucede un sismo, y de acuerdo con la secuencia de encendido de las estaciones se podría notar cómo se va propagando la onda sísmica a lo largo del país. Si bien la funcionalidad no ha sido puesta a prueba en el momento que sucede un sismo, a lo largo del proyecto la funcionalidad fue utilizada para observar el movimiento sísmico del país y observar el comportamiento mientras no ocurre ningún evento. De este modo se comprobó la validez del color mostrado con el registro del sismograma. Por otro lado utilizando los datos del sismo de Nicoya, se pudo comprobar como las velocidades registradas por un sismo superan el límite de 150000 nm/s y utilizando esta información se puede afirmar que la funcionalidad va a presentar el comportamiento descrito.

4.5. Elaboración de algoritmos de interpolación y animación de intensidades

4.5.1. Detección del sismo:

Con respecto a la estrategia definida para detectar el sismo. Utilizar la información de los archivos de OVSICORI fue un elemento favorable, ya que facilitó la forma de determinar cuándo se presentó un sismo. En la figura 4.21 se puede observar en el menú de la aplicación los últimos sismos registrados.

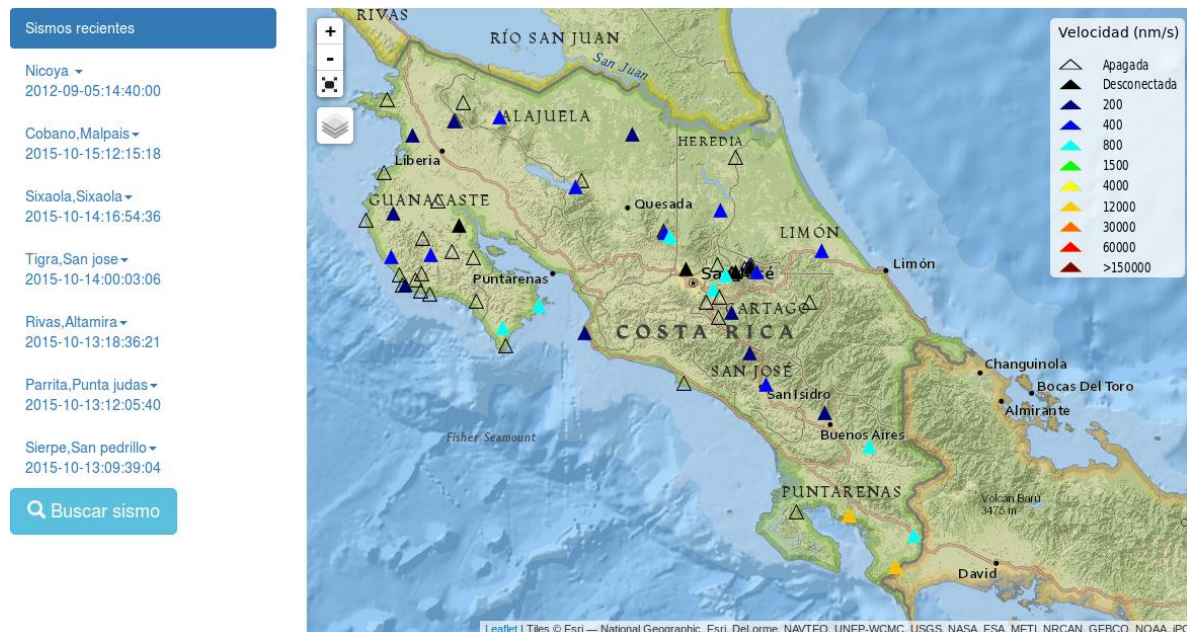


Figura 4.21 Captura de pantalla de la aplicación mostrando los últimos sismos registrados.

Los sismos que muestra la aplicación no son todos los que están registrados en el archivo JSON de OVSICORI. La información se filtró de modo tal que los sismos que va a presentar la aplicación son magnitud mayor a 2.5 grados en la escala Richté. Por otro lado los registros de sismos también abarcan sismos de países de la región centroamericana, de modo tal que si el nuevo registro de sismo en el archivo JSON no está entre latitudes 8.00°-11.15° norte y longitudes 82°-86° oeste, el sismo no es registrado para la aplicación. Por otro lado también el archivo registra eventos que son registrados manualmente, los cuales pueden ser del día de hoy o de días atrás, de modo que si el registro del evento supera los 30 minutos, el evento es descartado.

Con respecto al retraso entre el tiempo en que ocurre el sismo y cuándo es registrado en el archivo, el retraso es de algunos minutos de modo tal que la actualización de la página no es en tiempo real. Esto se debe a que para obtener el epicentro, el software utilizado en OVSICORI tarda varios minutos para obtener esta información.

Con respecto a la obtención de datos del sismo, por medio del método de ventanas utilizando la biblioteca libslink, a pesar de que se elaboraron los algoritmos necesarios para obtener la información, esta funcionalidad no pudo ser comprobada, ya que para acceder a la información del sismo se requiere la información de acelerógrafos. A lo largo del proyecto se estuvo utilizando información del servidor SeedLink de IRIS, el cual envía información de algunos sismógrafos de OVSICORI, sin embargo este servidor no provee información de acelerógrafos. La razón de utilizar este servidor, fue debido a restricciones de seguridad, donde no se puede acceder al servidor de OVSICORI desde una red externa, de modo que para poder comprobar los resultados de esta funcionalidad, será necesario trabajar directamente en OVSICORI o bien permitir acceso desde la red del TEC al servidor de OVSICORI.

De esta forma con respecto a la funcionalidad de detección de sismos, aún faltan elementos por verificar para poder completar la aplicación.

4.5.2. Animación

Aspecto gráfico:

Con respecto al aspecto gráfico el uso de WebGL cumplió con la expectativa, donde se logró elaborar un algoritmo capaz de representar por medio de una relación de colores un conjunto de datos sobre el mapa de Costa Rica. Se puede observar en la figura 4.22 un mapa con colores en diversas zonas del país, donde las zonas con color azulado representan datos con valores bajos y zonas con color más amarillo representan valores más altos.

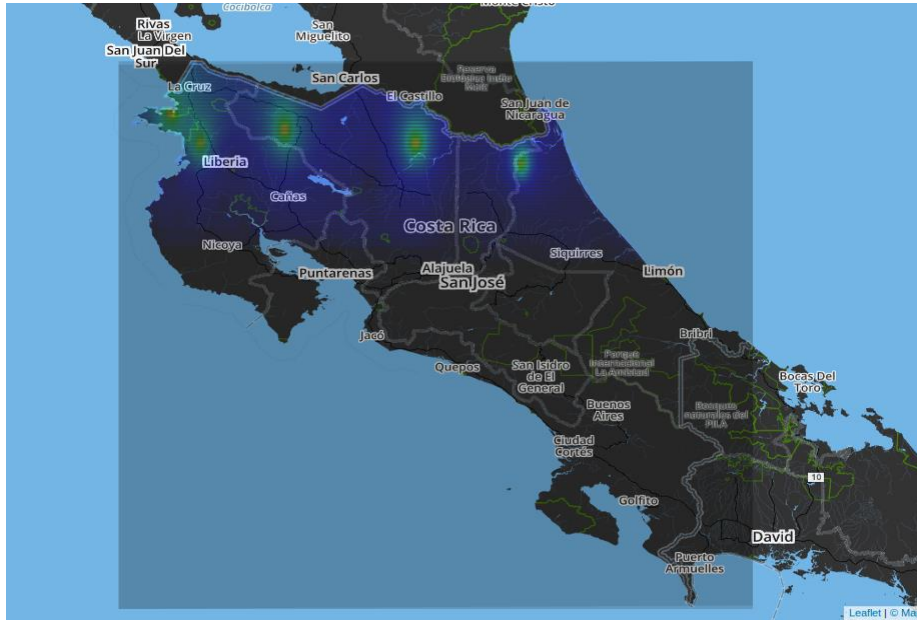


Figura 4.22 Captura de pantalla de versión previa de la representación gráfica

Con respecto a esta funcionalidad se hicieron mejoras tales como solo pintar posiciones que representan el área geográfica, de modo tal que el espacio marítimo se ignora. Para descartar las posiciones del espacio marítimo se tuvo que utilizar el programa QGIS, donde utilizando el lenguaje C, se creó un archivo de puntos raster que representan los vértices de los triángulos que utiliza WebGL. Luego utilizando comandos de consola en QGIS, se cargó el archivo de puntos y se creó una capa de puntos como se observa en la figura 4.23 representada como el área en negro ya que esta sobrecargada de puntos.

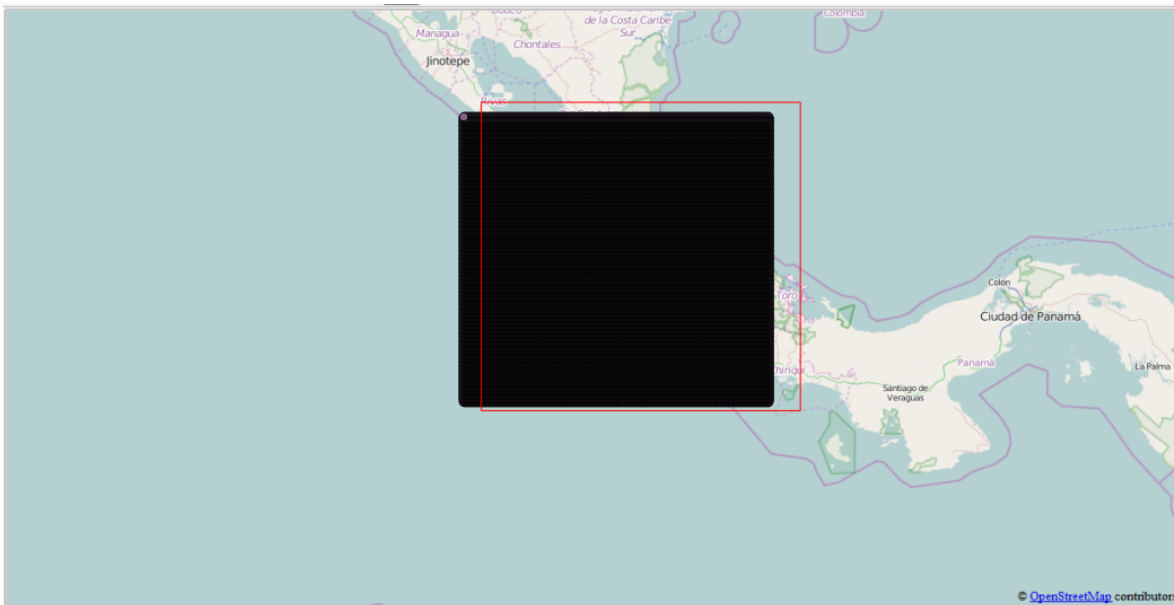


Figura 4.23 Carga de capa de puntos en QGIS

Luego utilizando una capa vectorial del área de Costa Rica como se observa en la figura 4.24, se utilizó la funcionalidad de cortar de QGIS, la cual compara las 2 capas y retorna todos los puntos que se encuentran dentro del área de la capa vectorial.

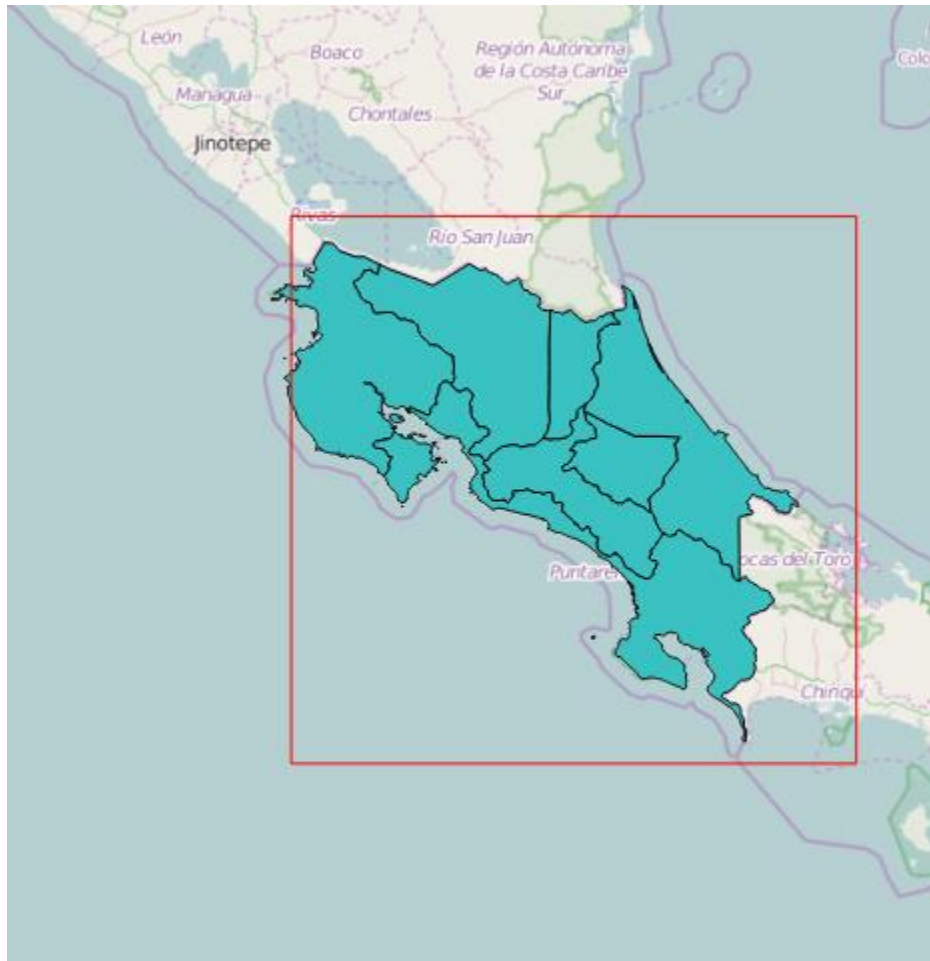


Figura 4.24 Capa vectorial de Costa Rica

Finalmente el resultado es la capa de puntos que se observa en la figura 4.25, donde esta capa es guardada en un archivo GeoJSON, el cual luego utilizando un programa en C, retorna un vector de posiciones válidas representadas por un 0 y posiciones no válidas representadas por un -1. Este archivo será leído por la aplicación de JavaScript, donde en el método de renderizar de WebGL, toda posición no válida no se le asigna color ni transparencia, de modo tal que la posición no podrá ser vista en la aplicación. En la figura 4.26 se puede observar el resultado de estas operaciones, donde es visible que la zona marítima no tiene color generado por WebGL.



Figura 4.25 Capa de puntos resultante de la operación de cortar

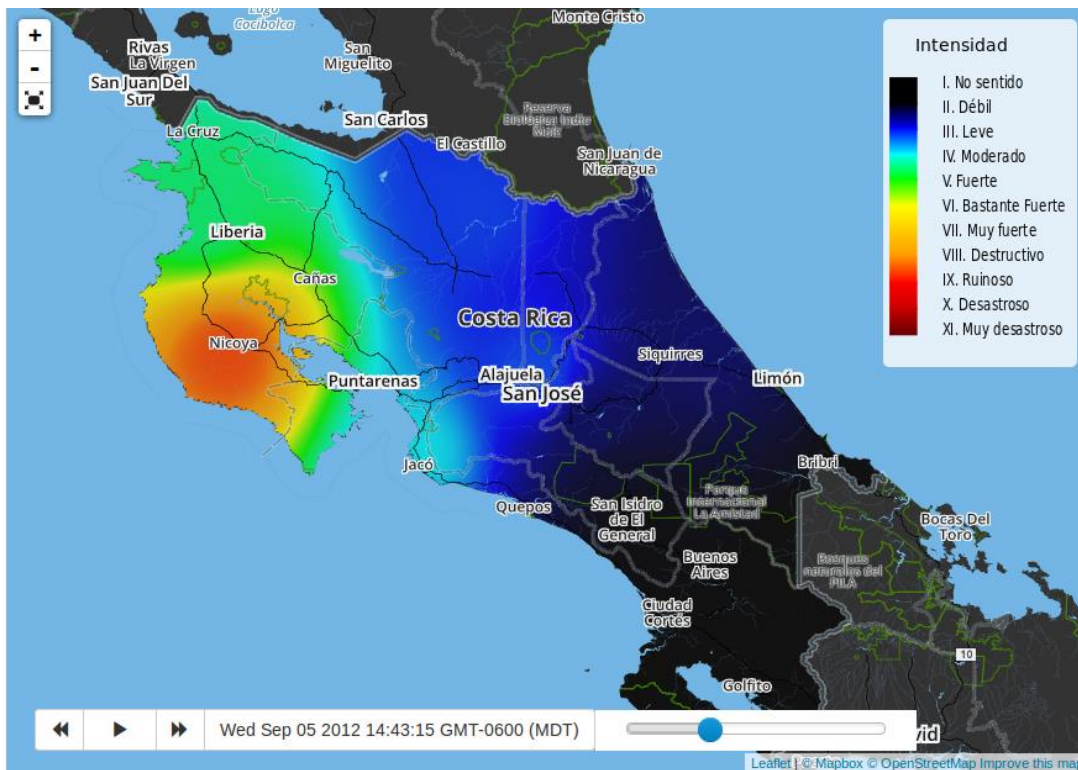


Figura 4.26 Captura de pantalla de la visualización con únicamente área terrestre de color

Por otro lado se observó que el uso del vector de 110x240 utilizando puntos con diferencia de 0.03°, producía un resultado en que la pantalla se veía algo pixelada, ya que el tamaño de los triángulos que formaban la imagen eran muy grandes. Por esta razón se decidió utilizar un vector de 322x700, donde cada punto tenía una diferencia de 0.01°. Con este cambio se logró ajustar la visualización sin que se manifestara mucho pixelado. En la figura 4.28, se puede observar este resultado, donde el pixelado debido al tamaño de los triángulos es bajo en comparación a la figura 4.27 donde el tamaño de los triángulos es más grande.

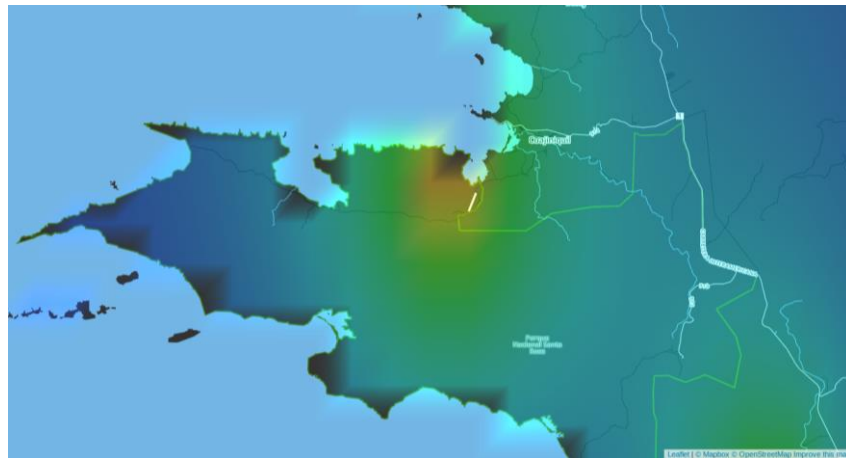


Figura 4.27 Captura de pantalla de la aplicación con pixelado debido a la distancia de los vértices

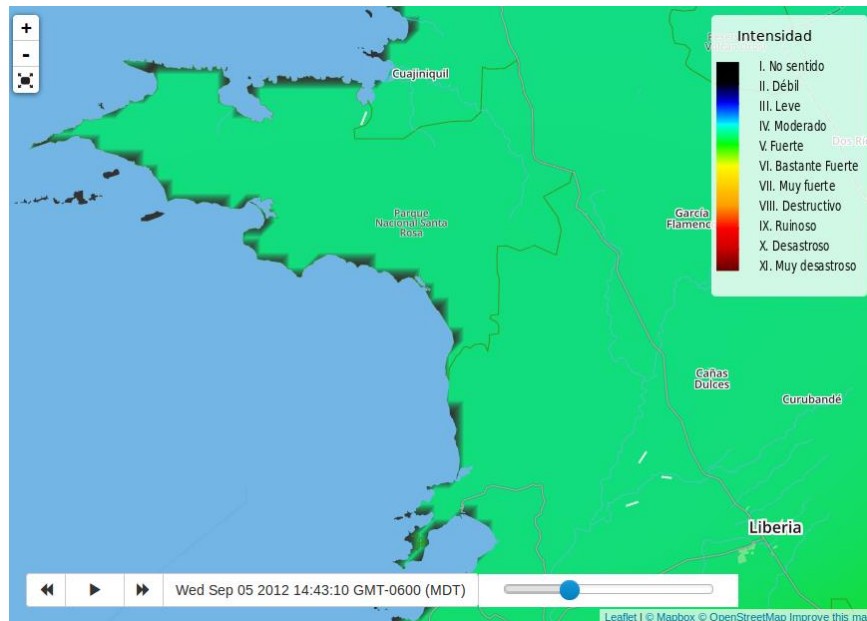


Figura 4.28 Captura de pantalla de la aplicación con pixelado reducido

Interpolación

Con respecto a la interpolación los aspectos que se evaluaron fueron tiempo de ejecución para decidir que algoritmo utilizar y la precisión del algoritmo escogido.

Para el caso del algoritmo IDW utilizando la función `console.time()` de JavaScript, se pudo observar que el tiempo requerido para obtener un resultado fue de 9ms, en el caso del algoritmo de Kriging el tiempo de duración para obtener una respuesta fue de 3743 ms. Se puede observar que hay una diferencia considerable entre ambos casos, esto se debe a que el algoritmo de Kriging requiere de más operaciones para obtener el resultado que el algoritmo IDW. Estas operaciones son utilizadas para determinar una mejor relación que el algoritmo IDW, ya que el resultado depende tanto de los pesos basados en la distancia entre los puntos conocidos como en el arreglo espacial de los puntos conocidos. En el caso de IDW el resultado depende únicamente de la distancia entre los puntos conocidos. Considerando este aspecto, el algoritmo de Kriging presentará resultado más preciso que el algoritmo IDW, ya que asume más criterios para obtener el resultado, sin embargo estos criterios hacen que aumente el tiempo requerido para obtener una respuesta del algoritmo. Puesto que se necesita un algoritmo de ejecución rápida para la aplicación en la cual el usuario no observe algún retraso producido por el procesamiento, el algoritmo de Kriging no es viable, por lo cual el algoritmo IDW es una opción recomendable si se considera únicamente tiempo de ejecución.

Para verificar la precisión del algoritmo IDW se utilizaron los datos del sismo de Nicoya. Estos datos contenían la información de los acelerógrafos que lograron registrar el evento. De modo que utilizando los códigos descritos en la sección 3.5.1, se obtuvo la aceleración máxima de cada una de las estaciones. Una vez obtenidos estos datos se cargaron en la aplicación de JavaScript y se obtuvo el resultado de la interpolación, el cual se puede observar en la figura 4.29. Para comparar la precisión de este algoritmo se comparó con los resultados del mapa de intensidad del Laboratorio de Ingeniería Sísmica (LIS). En la figura 4.30 se puede observar el resultado de LIS.

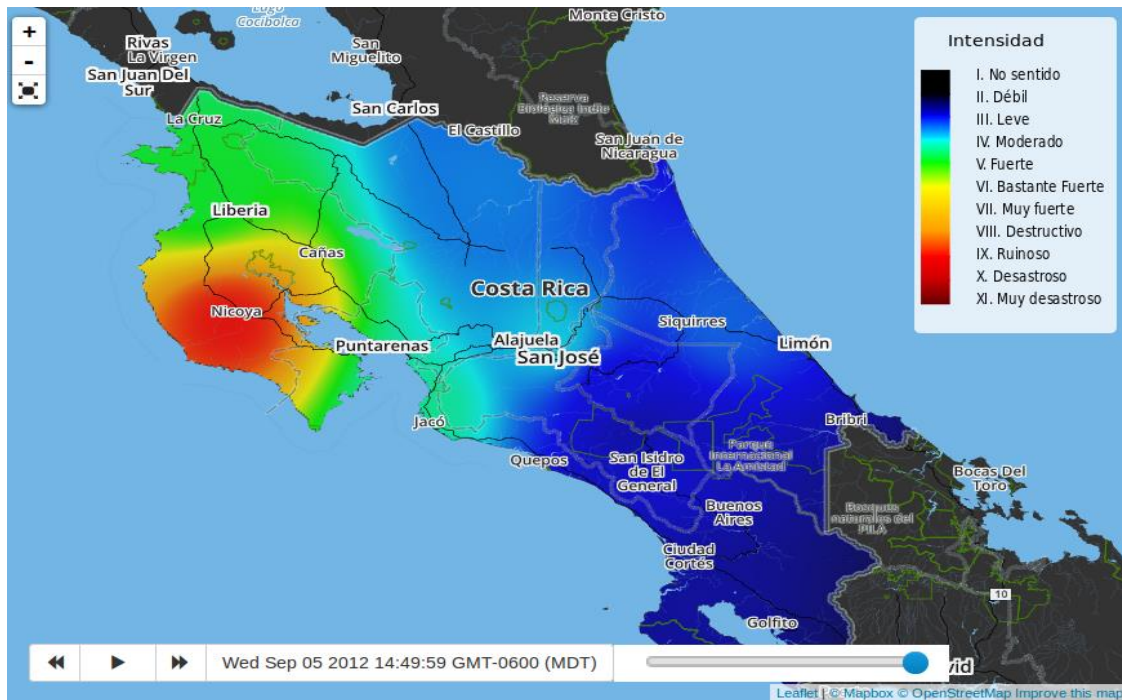


Figura 4.29 Resultado del mapa de intensidades utilizando interpolación IDW para el sismo de Nicoya

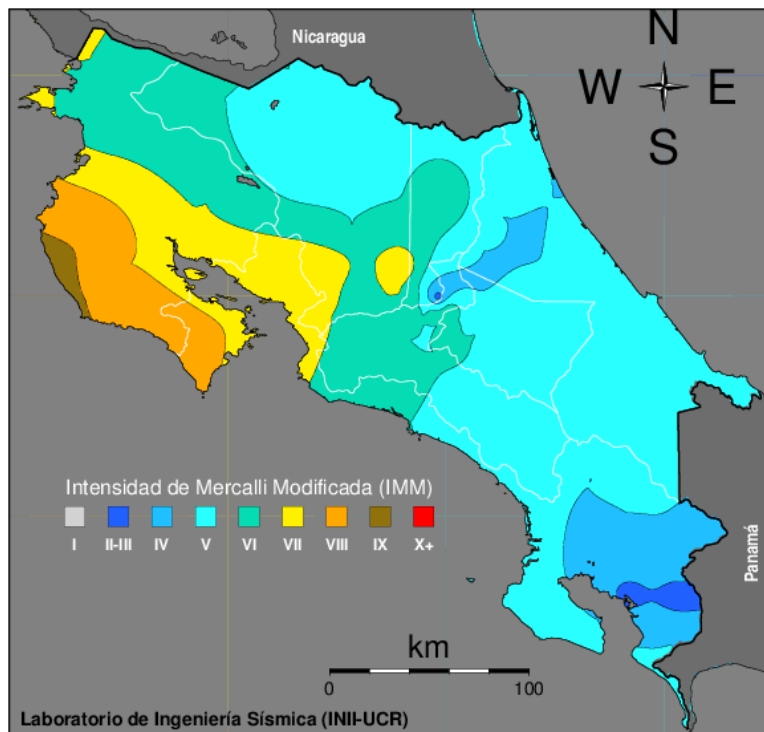


Figura 4.30 Resultado de mapa de intensidades de LIS para el sismo de Nicoya¹⁶

¹⁶ Laboratorio de Ingeniería Sísmica. (2012). Terremoto del 05 de setiembre del 2012

Se puede observar que hay semejanzas en el resultado obtenido con respecto al de LIS. En la zona de Nicoya se observa una mayor intensidad con respecto al resto del país y en la zona de la península de Osa la intensidad es de un valor entre II y III (débil y leve). Se observan diferencias en el área metropolitana donde los valores de intensidad marcados fueron de grado III y IV, sin embargo en el caso de LIS este efecto solo se presenta en una zona y el resto del área metropolitana fue de grado V. Estas diferencias están dadas por las siguientes razones:

- La cantidad de estaciones que utiliza LIS para sus cálculos. En la figura 4.31 se puede observar que la cantidad de estaciones utilizadas por LIS es mucho mayor que las 11 que se utilizaron de OVSICORI.
- La ubicación y el suelo donde se ubican las estaciones son otros factores que pueden determinar la diferencia entre los 2 casos.
- El algoritmo de interpolación que utiliza LIS para realizar su simulación probablemente no sea el mismo que se utilizó para desarrollar la aplicación de OVSICORI
- Finalmente la principal razón entre los cambios en los datos, es por el uso de únicamente el canal vertical de las estaciones. Las ondas S por su tipo de movimiento registran mayor movimiento en los canales horizontales que en el canal vertical de la estación

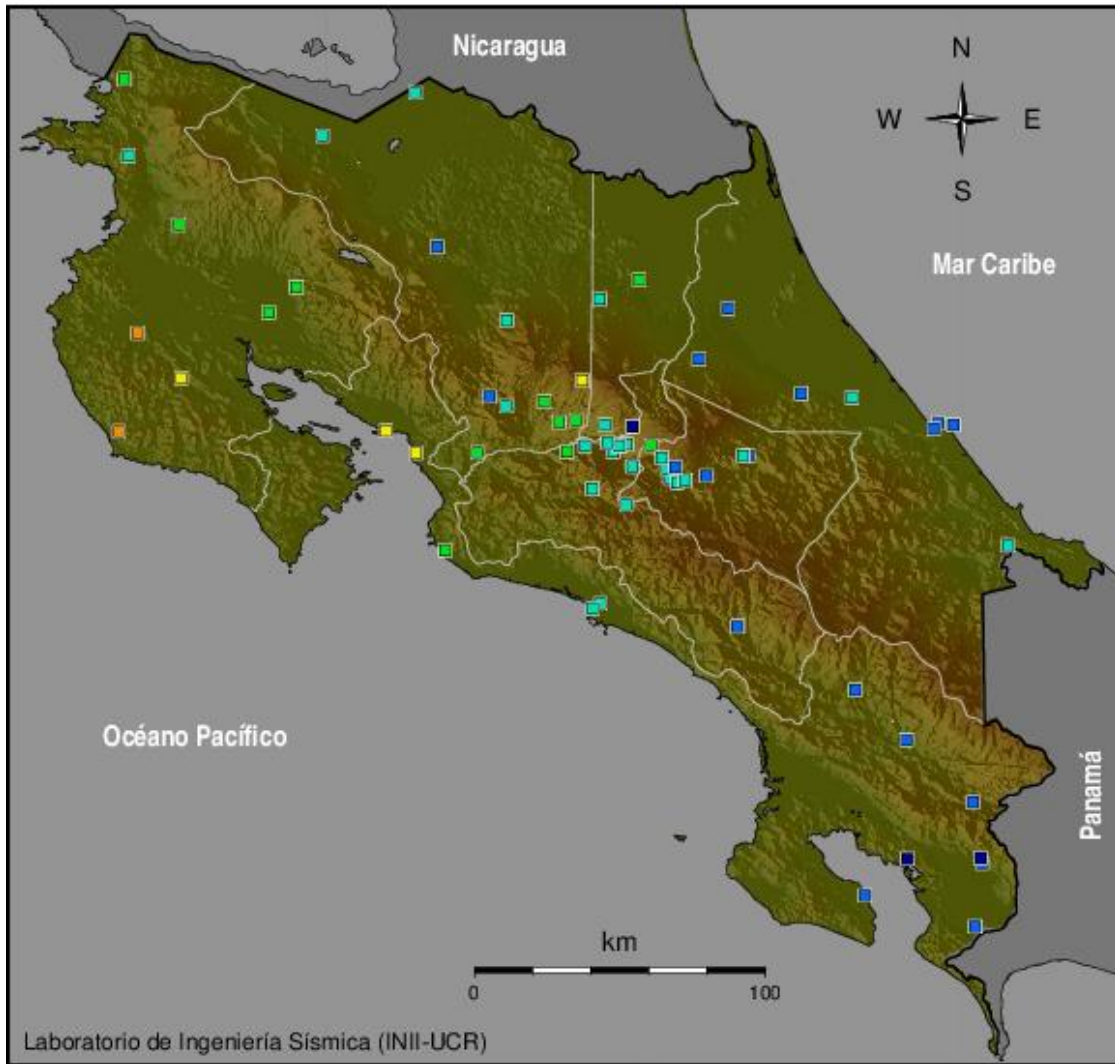


Figura 4.31 Mapa de acelerógrafos de LIS con colores de acuerdo con la intensidad utilizando escala MM¹⁷

De modo tal que observando el resultado final del sismo de Nicoya, se puede determinar que el algoritmo presenta un resultado real. Este resultado puede ser afectado por elementos como cantidad de estaciones y terreno en el que estén ubicados los sensores.

Por otro lado observando el resultado de las interpolaciones en distintos momentos, en la figura 4.32 se puede observar una anomalía. Esta se presenta ya que el algoritmo al tomar en cuenta la relación de distancia para cada estación, permite que estaciones más lejanas ejerzan una influencia sobre un punto lejano. Esto se puede observar en la figura 4.32 donde en la zona de Liberia, a pesar de que presenta un color negro, zonas aledañas y más lejanas como La Cruz muestran un color azulado. Esta es una anomalía que se presenta por la influencia que ejerce la estación de Nicoya en estas zonas.

¹⁷ Laboratorio de Ingeniería Sísmica. (2012). Terremoto del 05 de setiembre del 2012

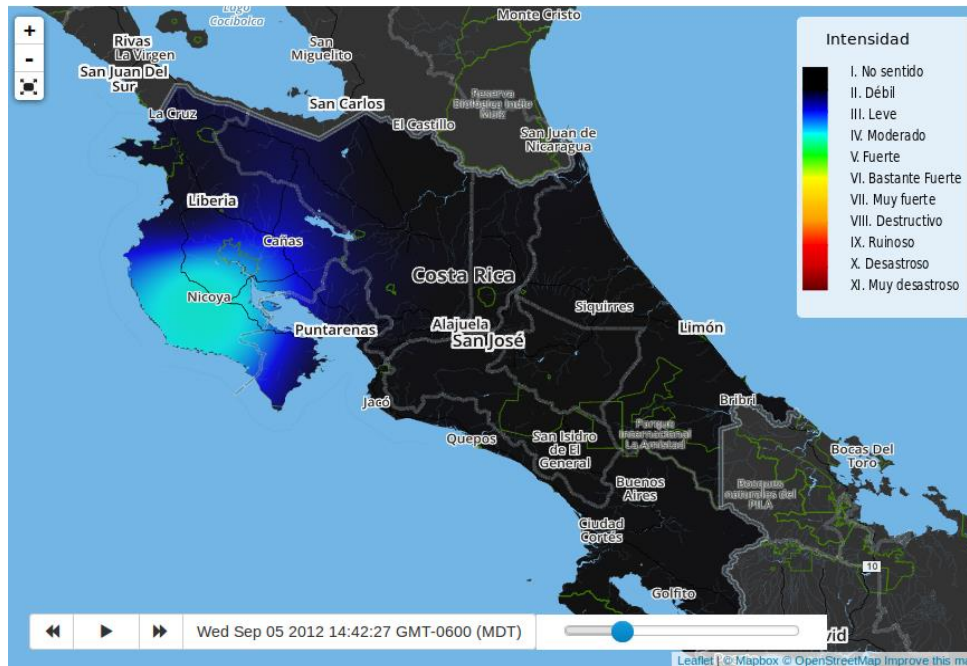


Figura 4.32 Anomalia presentada por influencia de estaciones lejanas con respecto a un punto
 Para corregir esta situación se establecieron una serie de alternativas:

- I. Definir un radio para ignorar las estaciones que superan el valor del radio.

Esta solución se implementó en el código, sin embargo el resultado no fue satisfactorio, ya que como se observa en la figura 4.33 hay zonas donde el cambio de color es más abrupto. Con base en estos resultados esta solución fue descartada.

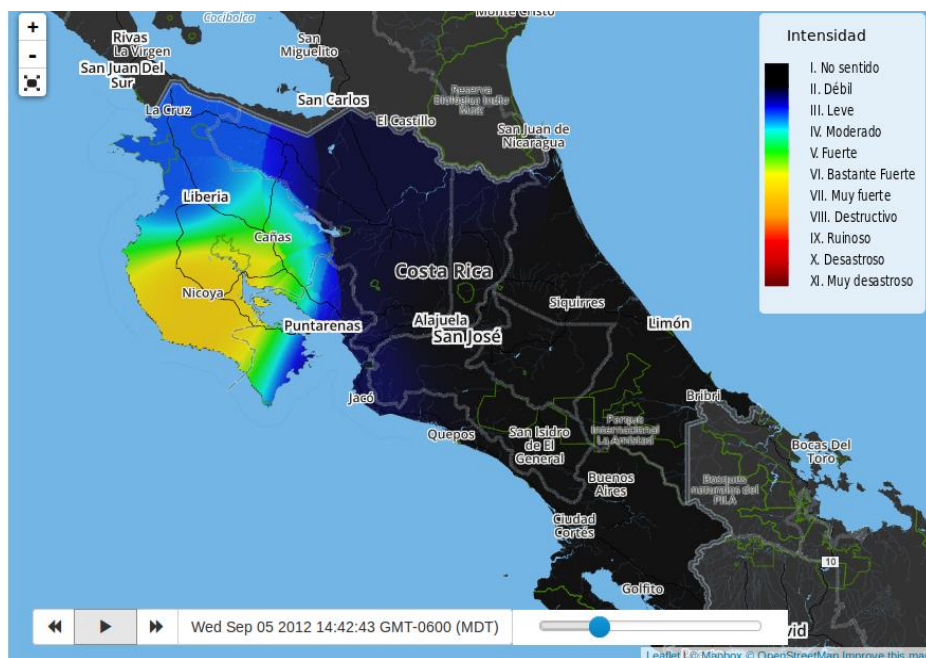


Figura 4.33 Anomalia obtenida al utilizar un radio como criterio de ajuste

II. Multiplicar cada resultado de la ecuación 2.5.2 por una función exponencial

La idea de esta solución es tomar la ecuación 2.5.2 y modificarla de la siguiente forma

$$u(x) = \begin{cases} \frac{\sum_{i=1}^N w_i(x) * u_i * e^{-\frac{d(x,x_i)^2}{r^2}}}{\sum_{i=1}^N w_i(x)} & \text{si } d(x,x_i)^p \neq 0 \text{ para todo } i \\ u_i & \text{si } d(x,x_i)^p = 0 \text{ en algún } i \end{cases}$$

Ecuación 4.5.1. Ecuación de ajuste para la interpolación

Donde:

$d(x, x_i)$: Es la distancia del punto a la estación

Utilizando esta función la idea es que se establezca una relación exponencial tal que utilizando la ecuación del círculo $\frac{(x-h)^2+(y-k)^2}{r^2}$, si la distancia entre un punto y la estación supera al radio, utilizando la función exponencial decreciente el valor va a ser cercano a 0. Si la distancia es pequeña el valor del exponente va a ser cercano a 1. De modo tal que la influencia de una estación decrece de forma exponencial usando como base un círculo. Esta función es una estrategia mejorada del caso anterior, donde el cambio es más suave y la influencia de la estación disminuye exponencialmente conforme aumenta la distancia. En la figura 4.34 se puede observar el resultado de esta estrategia.

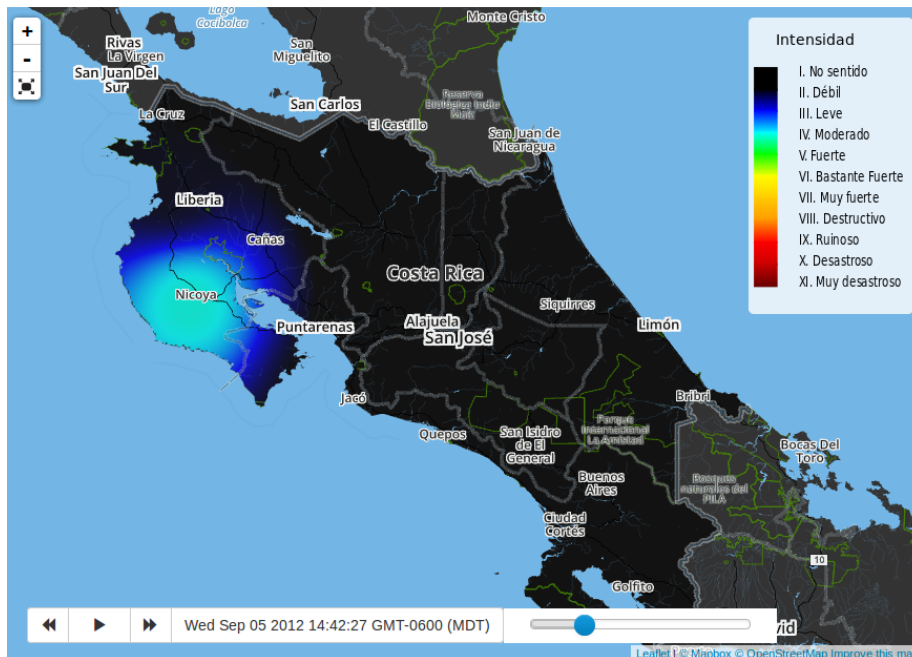


Figura 4.34 Captura de pantalla de la aplicación con el ajuste de la ecuación 4.5.1

Comparando el problema que se observaba en la figura 4.32 y el resultado obtenido en la figura 4.34. Se puede observar que se logró solucionar el problema con esta estrategia.

Animación

Con respecto a la animación debido a la situación descrita en la sección 4.5.1, no se pudo comprobar la animación para los sismos recién registrados. Sin embargo para comprobar la animación se utilizaron los datos obtenidos del sismo de Nicoya.

Con respecto a los resultados obtenidos, se utilizaron los datos de aceleración de las estaciones por cada segundo y estos se pusieron a correr utilizando la interpolación y WebGL. De modo que la animación corresponde a la evolución de las intensidades sentidas en Costa Rica por cada segundo.

Con respecto a la fluidez entre cada salto de tiempo (1 segundo), inicialmente se observaba en algunos momentos discontinuidad en la animación producidos por variaciones de gran diferencia en las intensidades registradas para un tiempo t y un tiempo $t+1$. En figura 4.35 se puede observar una captura de pantalla de la animación para un tiempo t , en la figura 4.36 se puede observar el tiempo t más un segundo. Se puede observar que hay una variación significativa entre los dos espacios de tiempo en la zona de El Castillo.

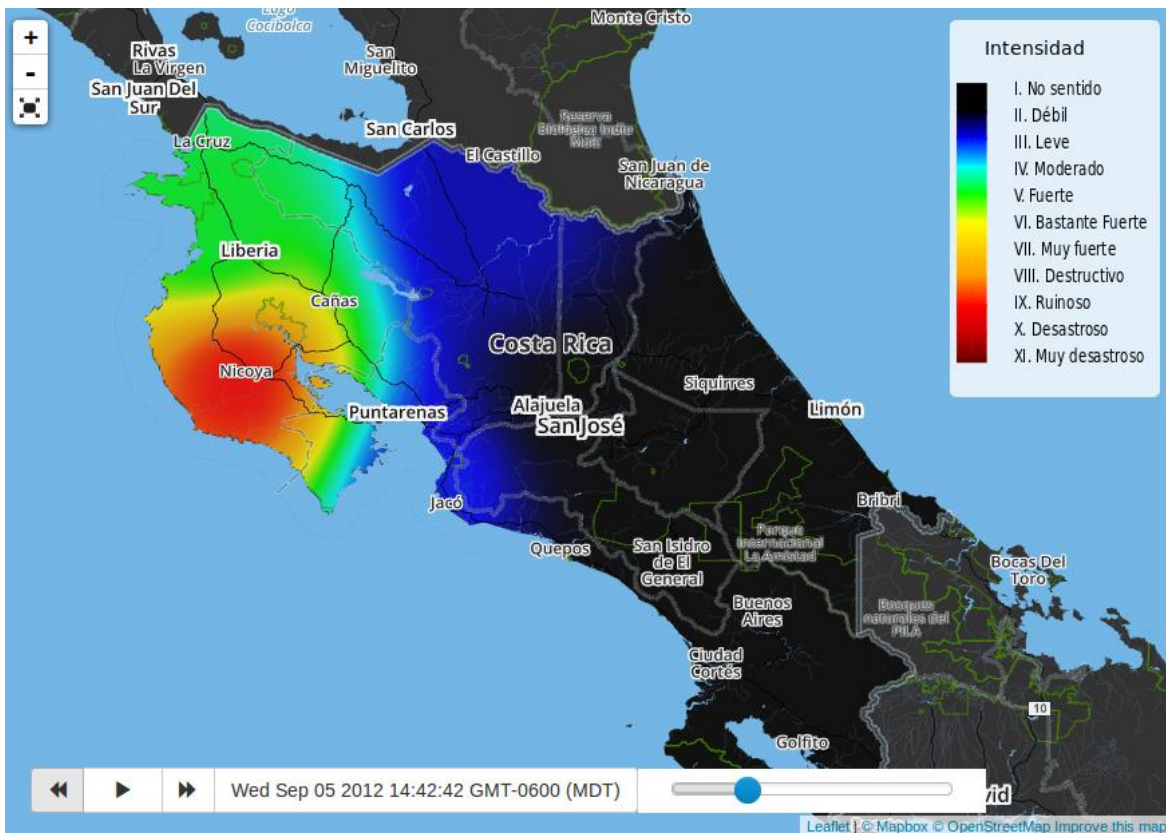


Figura 4.35 Captura de pantalla de la animación en un previo a un cambio abrupto

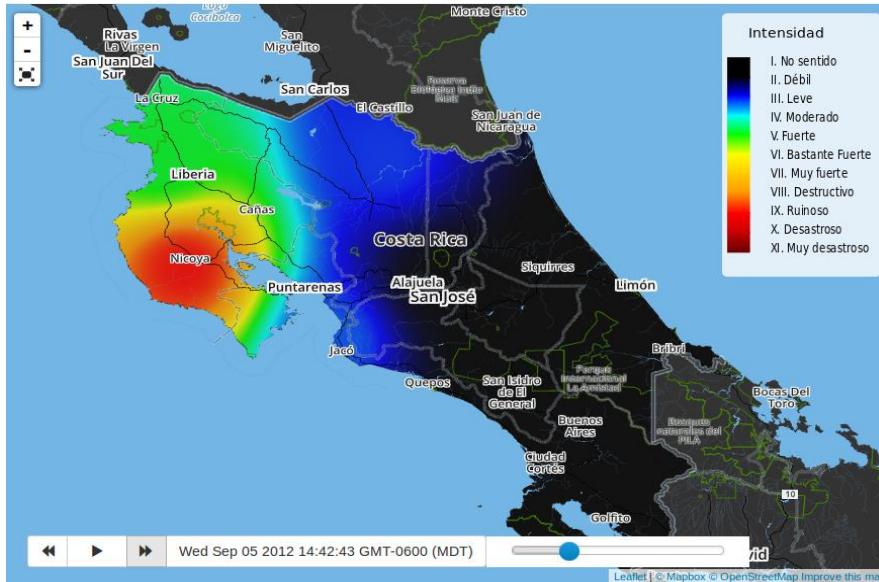


Figura 4.36 Captura de pantalla de la animación con luego de un salto

Para corregir la situación se utilizó la técnica de alisado exponencial. Utilizando la ecuación 3.3.2, se logró reducir la diferencia de intensidades entre cada espacio temporal. Utilizar esta técnica fue favorable, ya que la variación de intensidad de un punto a otro fue más suave. De este modo se puede observar en la figura 4.37 y la figura 4.38, la forma corregida del caso anterior. Se puede observar que ahora el cambio es menos notorio, ya que la variación de colores es más suave entre la figura 4.37 y 4.38 en la zona de El Castillo.

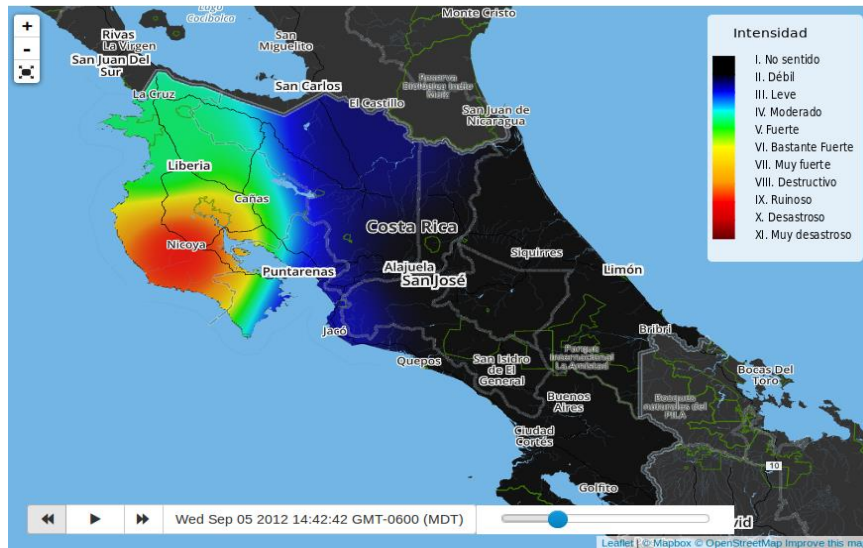


Figura 4.37 Captura de pantalla de la aplicación con ajuste previa al cambio abrupto

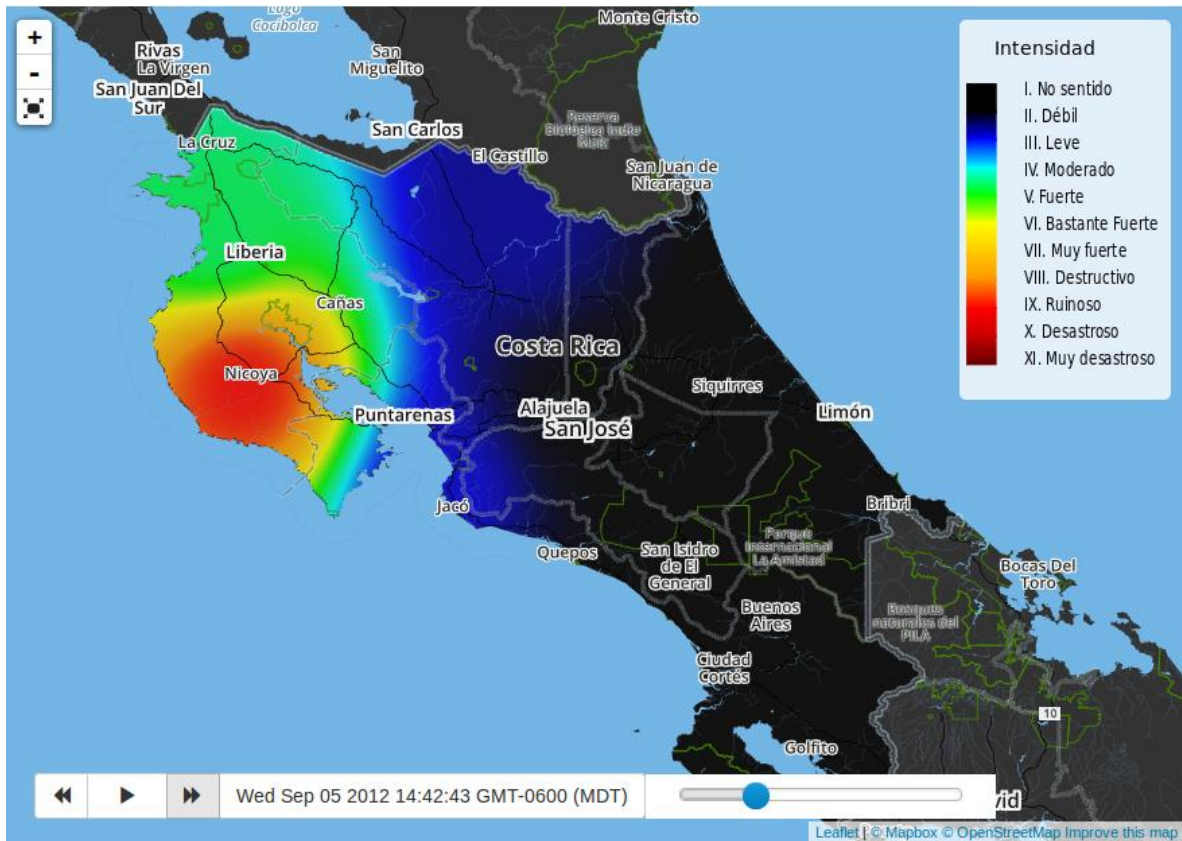


Figura 4.38 Captura de pantalla de la aplicación con ajuste en donde ocurría la discontinuidad

5. Conclusiones y recomendaciones

5.1. Conclusiones

Existen diferentes tipos de instrumentos sísmicos como lo es el sismógrafo y el acelerógrafo, los cuales tienen diferentes grados de sensibilidad. Cada sensor puede tener una orientación vertical u horizontal, donde la intención de cada uno es registrar el movimiento de los diferentes tipos de onda (S,P, Rayleigh y Love). Un sensor vertical puede registrar los movimientos de la onda P y Rayleigh, mientras que el sensor horizontal registra mejor los movimientos de la onda S y Love. Por esta razón para realizar el monitoreo en tiempo real de estaciones sísmicas, el canal vertical es el más indicado de utilizar, ya que al ser la onda P la más veloz de los diferentes tipos de onda, va a ser la primera en ser registrada por las estaciones.

Para poder recibir datos desde sismógrafos y acelerógrafos se utilizó la biblioteca libslink y se desarrolló un módulo de recepción de datos. La biblioteca libslink implementa el estándar SEED y permite la comunicación con un servidor SeedLink habilitando una conexión TCP/IP. Esta biblioteca implementa todas las funcionalidades necesarias para comunicarse con el servidor. Por otro lado, el servidor SeedLink envía paquetes MiniSEED de 512 bytes al cliente siempre que reciba datos desde una estación sísmica, de modo tal que la recepción de datos se genera en tiempo real. Con esta funcionalidad se logra el objetivo de obtener datos en tiempo real desde las estaciones sismológicas de OVSICORI

Para almacenar los datos desde las estaciones se utilizaron archivos aplicando el formato JSON. Esta opción permitió crear un módulo de recepción de datos que utiliza pocos recursos del sistema. Donde el módulo utiliza un 0.1% de memoria y 3% del procesador. Utilizar una base de datos hubiese generado un mayor consumo de recursos ya que se hubiese tenido que estar accediendo de forma continua a los datos de la base de datos utilizando consultas complejas para acceder a información de las diferentes tablas.

Se logró desarrollar una aplicación web la cual presenta 2 funcionalidades principales: Monitoreo en tiempo real de sismógrafos y acelerógrafos y animación de sismos a partir de la intensidad.

A partir de los datos obtenidos de las estaciones, se logró mostrar en un mapa los marcadores que representan las estaciones. Cada marcador cambia de acuerdo con el valor registrado en la estación, donde el color rojo alerta un movimiento sísmico fuerte en la zona donde se ubica la estación. Debido a que los datos se están obteniendo en tiempo real desde las estaciones, un usuario que esté visualizando la herramienta web podrá observar con el cambio de color de las estaciones si se está presentando un sismo en una zona. Con esta funcionalidad se logra generar un monitoreo en tiempo real de las estaciones sismológicas del OVSICORI

Para comprender de mejor manera el movimiento que registró la estación se desarrolló la funcionalidad de mostrar sismogramas. Se pudo observar que todos los sensores se ven afectados por diferentes fuentes de ruido. Para eliminar estos factores que distorsionan la señal se implementaron diversas técnicas de filtros digitales. El filtro Butterworth IIR mostró los mejores resultados, sin embargo observando el resultado obtenido con respecto a la señal original se determinó que en gran medida se logró restaurar la señal del sensor, no obstante aún hay fuentes de ruido que siguen afectando la señal de la estación sísmica. El caso específico es el del ruido humano que se encuentra entre los 2 y 4 Hz, este ruido no se pudo eliminar ya que la frecuencia de los sismos regionales está entre los 0.5 a 3 Hz.

Comprendiendo que las personas no siempre van a estar conectadas a la aplicación y que estas accederán al sitio web luego de un evento para consultar la información del terremoto, se desarrolló una animación del progreso de la intensidad de un sismo en todo el país a lo largo de su duración. Para realizar esto se investigaron los algoritmos de interpolación de Kriging e IDW, donde el seleccionado fue IDW, ya que este algoritmo requiere menos tiempo y procesamiento para generar un resultado en comparación del algoritmo de Kriging. Esto fue un aspecto necesario para la aplicación, ya que era necesario que el usuario no notara retrasos en la animación producto del procesamiento requerido para realizar las interpolaciones.

Se desarrollaron los algoritmos para generar la información necesaria para realizar la animación. Se desarrolló un código que define si sucedió un terremoto a partir de consultas a información de OVSICORI. En caso de definir qué sucedió un evento se obtienen los datos de cada estación sísmica que registró el evento para obtener la aceleración máxima. Una vez obtenida esta aceleración máxima, se utilizaron las ecuaciones de LIS de la UCR para hacer una relación intensidad-aceleración máxima. Sin embargo los algoritmos para capturar la información de las estaciones por medio de la consulta de una ventana de tiempo utilizando libslink, no pudieron ser verificados ya que no se tuvo acceso directo al servidor de OVSICORI, el cual era el único que podía brindar la información de acelerógrafos. No obstante esto no fue una limitante para generar la animación y comprobar su funcionamiento, ya que se utilizaron los datos del sismo de Nicoya para verificar la validez de la animación.

Con respecto a la animación utilizando el algoritmo de interpolación IDW, se logró a partir de los datos de 11 de los 21 acelerógrafos de OVSICORI generar un mapa de intensidades del sismo de Nicoya utilizando WebGL y una escala de colores. Los resultados de la interpolación fueron favorables en cuanto a tiempo de ejecución se refiere, ya que para generar el mapa solo se requieren 9 ms implementando el algoritmo. Por otro lado el resultado obtenido es bastante acertado en comparación al mapa generado por LIS para el sismo de Nicoya. Se puede resaltar que la interpolación tiene limitaciones, donde áreas con muy pocas estaciones cercanas pueden afectar la precisión del resultado obtenido. Por otro

lado se puede observar que hay diferencias entre el resultado obtenido y el resultado de LIS, esto es generado principalmente por utilizar únicamente el canal vertical de la estación, donde LIS utiliza estaciones con el canal horizontal. Otro factor que altera los resultados obtenidos es que LIS utiliza muchas más estaciones para realizar el cálculo, por lo cual para mejorar el resultado de la interpolación se requiere utilizar una mayor cantidad de estaciones.

Finalmente para implementar la animación se utilizaron los datos de intensidad máxima para cada segundo del sismo y se les aplicó la interpolación, de modo tal que se logró observar cómo evolucionó la intensidad del sismo en cada segundo en un tiempo de 5 minutos. Se logró observar que la animación una vez aplicadas las modificaciones para generar una suavidad entre los datos de intensidad evoluciona de una forma constante sin cambios abruptos. La funcionalidad logra cumplir con el objetivo de permitir a la población comprender un sismo de una forma más robusta, ya que permite observar en qué grado afectó el sismo a diferentes zonas del país. Es a través de esta funcionalidad que se le permitirá a OVSICORI contar con una herramienta que les permita mostrar a la población de forma más detallada un evento sísmico.

En síntesis se logró desarrollar una herramienta web para OVSICORI, la cual cuenta con 2 funcionalidades principales, monitoreo en tiempo real de las estaciones sísmicas del observatorio y simulación de sismos. Para ello se tuvo que comprender términos y estándares del área de sismología para comprender el funcionamiento de las estaciones y datos obtenidos de las mismas. Se desarrolló una funcionalidad de monitoreo la cual muestra en el mapa de Costa Rica las estaciones y se estableció una relación de color con su registro más reciente, así mismo esta funcionalidad permite ver los sismogramas y acelerogramas de las estaciones. Con estos gráficos se evidencia la actividad sísmica en una región del país y tienen la limitante de que su registro puede verse alterado por la actividad humana. Por otro lado se logró realizar una funcionalidad de simulación de intensidad para un sismo, la cual solamente pudo ser verificada para el sismo de Nicoya. La funcionalidad cumple con el objetivo de mostrar con mayor detalle la evolución de un sismo y su efecto en diversas zonas del país. Se logró desarrollar una herramienta para OVSICORI que permite brindar más información de la que se brindaba inicialmente sobre eventos sísmicos del país a la población en general.

5.2.Recomendaciones

A continuación se mencionan algunas recomendaciones y funcionalidades que se pueden implementar para extender la usabilidad de la herramienta:

- Todas las bibliotecas utilizadas para la herramienta web son responsive, de modo tal que se pueden elaborar las modificaciones para permitir que la aplicación sea consultada desde dispositivos móviles
- La aplicación tiene un espacio que se quiere desarrollar el cual es la simulación de ondas sísmicas, donde el objetivo es ver el desplazamiento de las ondas P y S cuando se presenta un evento sísmico. Es por esto que se pretende seguir extendiendo este proyecto en los próximos meses
- Las técnicas de interpolación utilizadas sacrifican un poco la precisión para obtener resultados en poco tiempo. De modo que si se quiere obtener un resultado de mayor precisión, se debe utilizar equipos de supercomputación como los clusters para obtener una simulación más real. Utilizar los equipos de supercomputación implicaría que la forma en la que se ejecuta actualmente la simulación no podría presentarse directamente en la aplicación, la funcionalidad deberá cambiar a brindarle la opción al usuario de descargar un video. Utilizar esta estrategia permite una precisión más real, pero elimina aspectos como poder navegar por el mapa y realizar zoom sobre el mismo para acercarse a ver una zona en específico, así mismo el hecho de que el usuario tenga que descargar un video puede causar disconformidad para el mismo.
- Aún queda la posibilidad de ofrecer a OVSICORI el servicio de montar un clúster de visualización en las oficinas de OVSICORI. Este equipo sería de utilidad para dar una conferencia a los medios de comunicación luego de un evento y poder mostrar en las noticias las simulaciones de la aplicación generadas. De ser posible esto, se deberá lograr correr la aplicación dentro de un clúster, para ello se requerirá la investigación de como mostrar la aplicación web de manera distribuida en un cluster de visualización
- Finalmente es necesario verificar la funcionalidad para obtener los datos de acelerógrafos luego de registrar que sucedió un sismo, ya que esta funcionalidad no pudo ser verificada por la limitación de seguridad para acceder al servidor de OVSICORI desde una red externa.

6. Anexos y Apéndices

6.1. Apéndices

6.1.1. Apéndice A.1 Descripción de herramientas utilizadas

A continuación se detallan las diversas herramientas que utilizadas a lo largo del proyecto, así como sus versiones y el uso que se le dio en el proyecto.

- **Bootstrap** (versión 3.3.5): Bootstrap es un framework, para el desarrollo de sitios web y aplicaciones web. Permite la creación de elementos gráficos de una página web como menús, botones, entre otros elementos. Bootstrap permite desarrollar sitios web adaptables, de modo tal que la aplicación web es compatible tanto con dispositivos móviles, laptops, computadores de escritorio. El framework fue utilizado en el proyecto para el desarrollo de componentes visuales como menús, botones, sliders, los cuales mejoraron el aspecto gráfico y usabilidad de la aplicación para el usuario.
- **Canvas.js** (versión 1.7): Es una biblioteca de gráficos para HTML5 escrita en JavaScript. La biblioteca corre en diversas plataformas como iPhone, iPad, Android, Windows Phone, PC, entre otras. La biblioteca permite crear tableros que funcionan en todos los dispositivos sin comprometer la funcionalidad de la aplicación web. Esta biblioteca fue utilizada para generar los sismogramas de la aplicación
- **Jansson** (versión 2.7): Es una biblioteca escrita en C, la cual permite la codificación y decodificación de y manipulación de información en formato JSON. La biblioteca fue utilizada para leer y escribir los datos desde las estaciones sísmicas en formato JSON. También fue utilizada para leer la información de las capas de puntos de estaciones sísmicas para obtener información como localización y nombre de la estación. Finalmente la biblioteca también fue utilizada para obtener la información de un sismo de los archivos de OVSICORI como lo es epicentro, localización, hora, entre otros.
- **Leaflet** (versión 0.7.3): Es una biblioteca de código libre escrita en JavaScript, la cual permite el desarrollo de mapas interactivos. Esta biblioteca fue diseñada considerando aspectos como simplicidad, rendimiento y usabilidad. Trabaja eficientemente a través de todas las plataformas móviles y de escritorio y puede ser extendida combinándola con otras bibliotecas. Esta biblioteca fue utilizada a lo largo del proyecto como la base para desarrollar la aplicación web, donde todos los mapas, capas de puntos y animaciones fueron elaboradas utilizando esta biblioteca.

- **Inkscape** (versión 0.91): Inkscape es un editor profesional de gráficos vectoriales para Windows, Mac OS X y Linux. Es libre y de código abierto. Este programa fue utilizado para generar las imágenes de las leyendas de la aplicación, las cuales se pueden observar en las figuras 3.5,3.6 y 3.8.
- **Libslink** (versión 2.4b): Es una biblioteca escrita en C proporcionada por la red IRIS. La biblioteca provee todas las funcionalidades para comunicarse con un servidor SeedLink de forma generalizada. Esta biblioteca fue utilizada para poderse comunicar con el servidor de OVSICORI y la red IRIS para obtener datos desde las estaciones sísmicas de OVSICORI.
- **Qgis** (versión 2.10.1): Este es un sistema de información geográfica libre y de código abierto, el cual permite crear, editar, analizar y publicar información geoespacial. Esta herramienta fue utilizada para crear capas de información como los puntos de los sismógrafos, puntos de los acelerógrafos y capa de puntos válidos del mapa de Costa Rica utilizado en la animación de WebGL.

6.1.2. Apéndice A.2 Diagramas de flujo

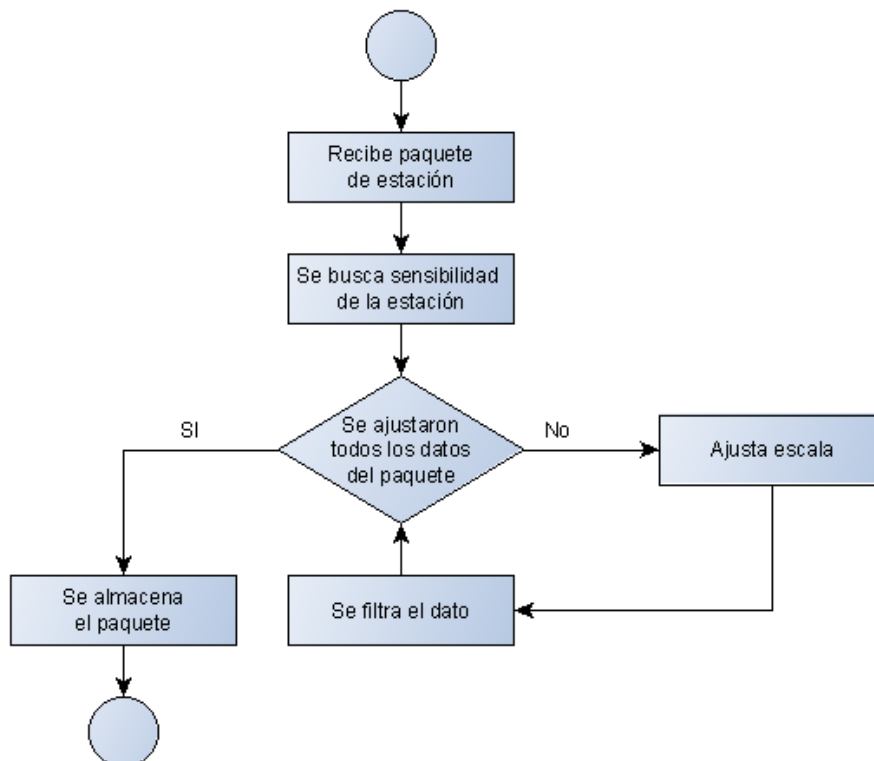


Figura 6.1 Diagrama de flujo para algoritmo de recepción de datos

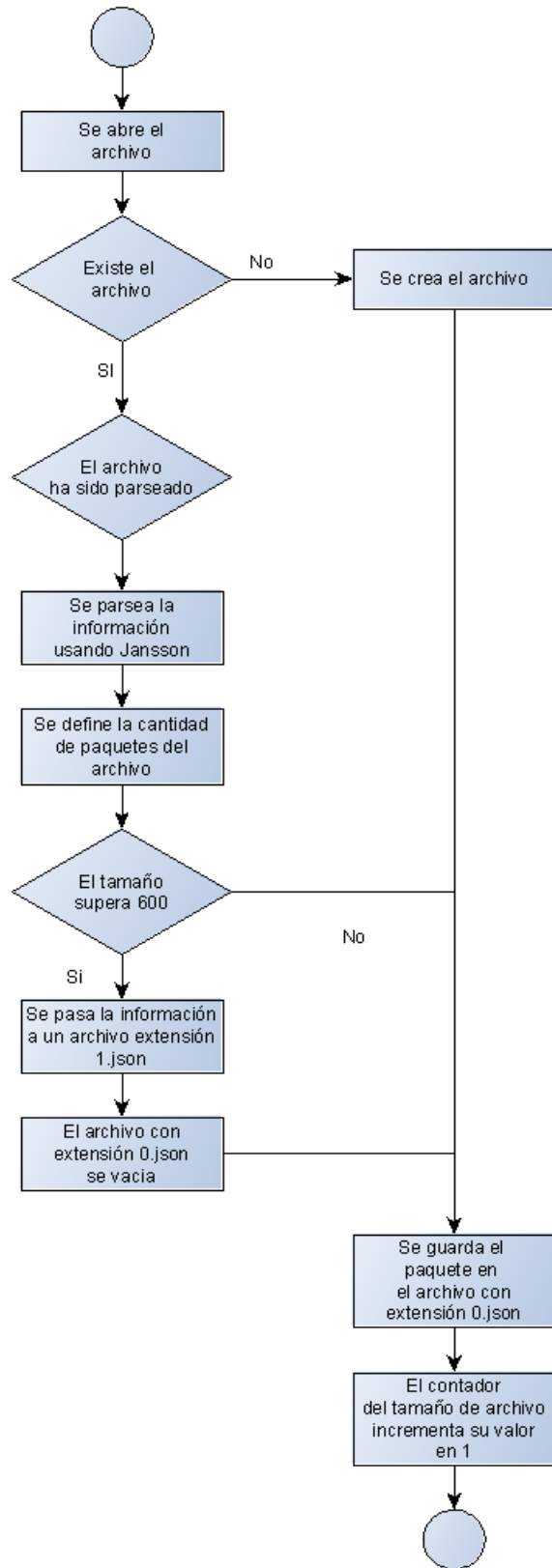


Figura 6.2 Diagrama de flujo para el algoritmo de almacenamiento de datos

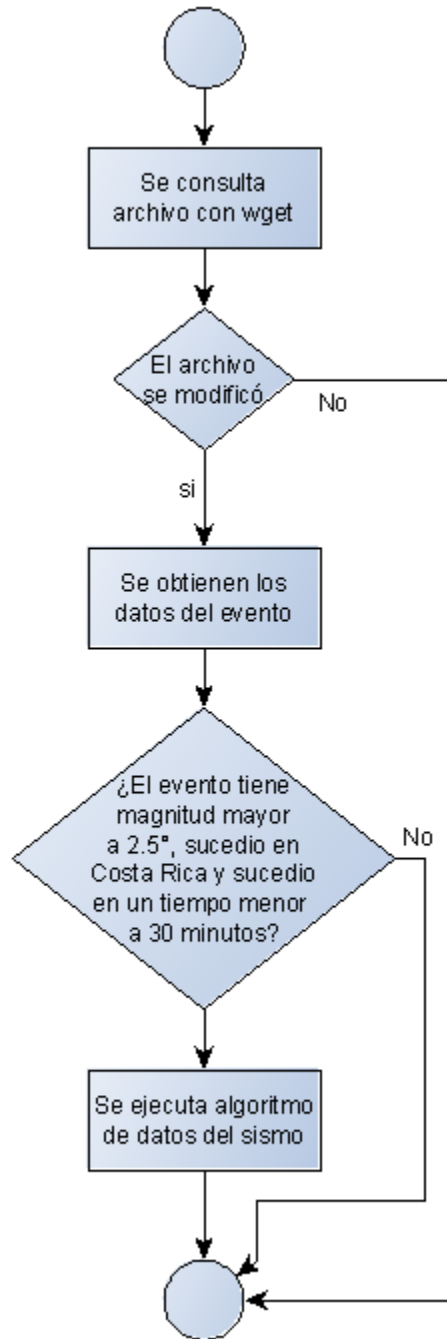


Figura 6.3 Algoritmo de monitoreo de eventos

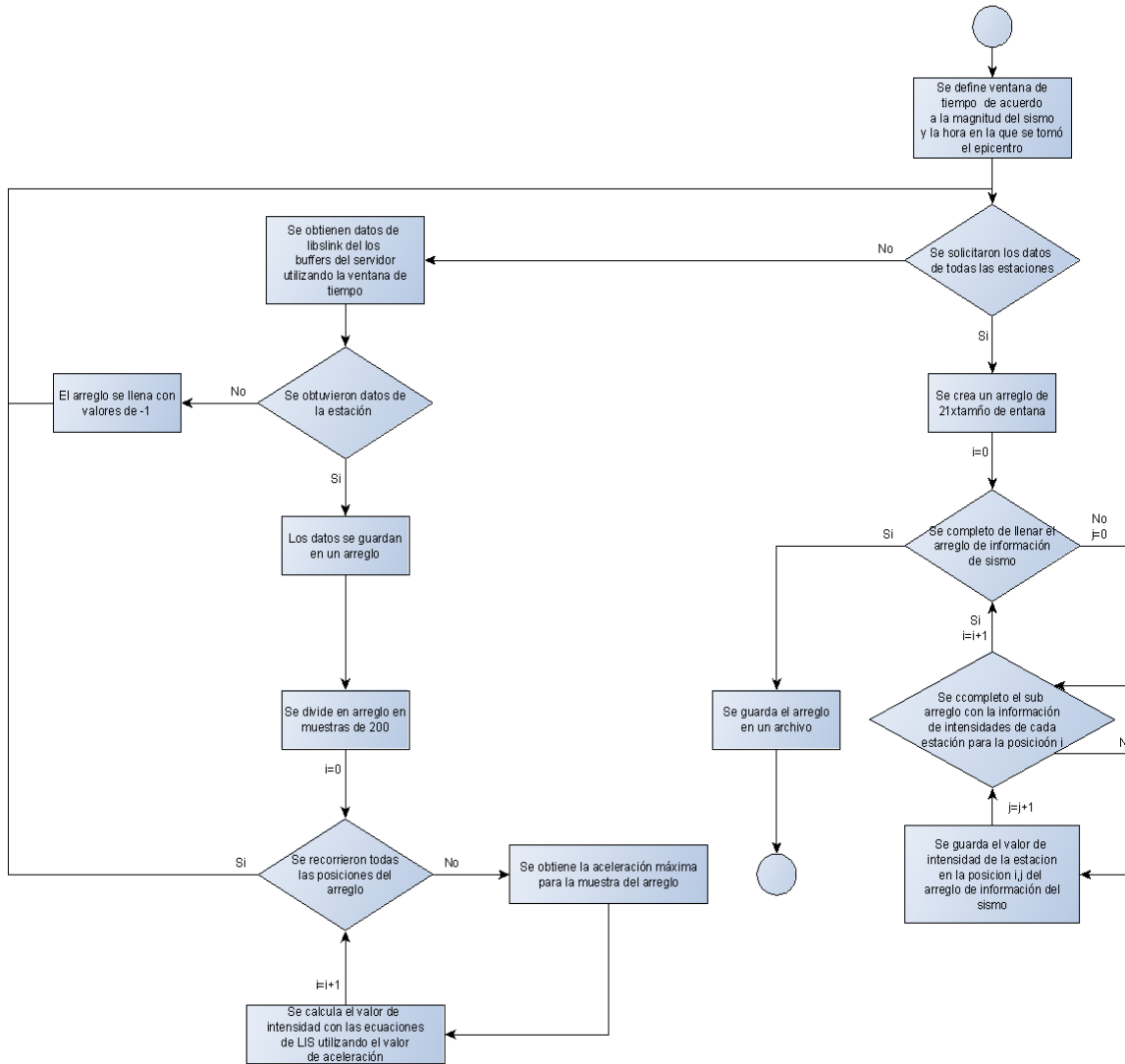


Figura 6.4 Diagrama de flujo para algoritmo de obtención de datos del sismo

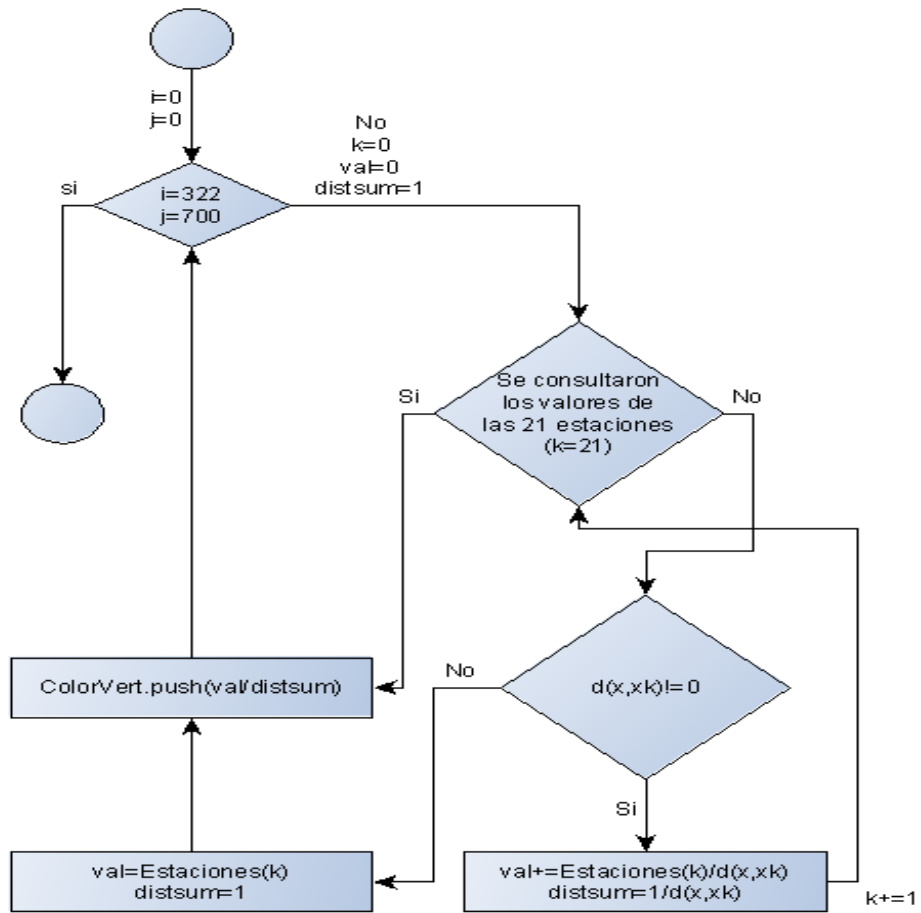


Figura 6.5 Diagrama de flujo para interpolación IDW

6.2. Anexos

6.2.1. Anexo B.1 Escalas para medición del tamaño de un terremoto

A continuación se presentan las escalas de medición de intensidades y relación entre magnitud, aceleración e intensidad. Estas tablas fueron tomadas de los sitios web de LIS y la Red Sísmica de Puerto Rico respectivamente.

Intensidad	Significado
I	<ul style="list-style-type: none">• Sacudida sentida por muy pocas personas
II	<ul style="list-style-type: none">• Sacudida sentida sólo por pocas personas en reposo, especialmente en los pisos altos de los edificios.• Los objetos suspendidos pueden oscilar.
III	<ul style="list-style-type: none">• Sacudida sentida claramente en los interiores, especialmente en los pisos altos de los edificios, muchas personas no lo asocian con un temblor.• Los vehículos de motor estacionados pueden moverse ligeramente.• Vibración como la originada por el paso de un carro pesado.• Duración estimable.
IV	<ul style="list-style-type: none">• Sacudida sentida durante el día por muchas personas en los interiores, por pocas en el exterior.• Por la noche algunas despiertan.• Vibración de vajillas, vidrios de ventanas y puertas; los muros crujen.• Sensación como de un carro pesado chocando contra un edificio, los vehículos de motor estacionados se balancean claramente.
V	<ul style="list-style-type: none">• Sacudida sentida por mucha gente; muchos despiertan.• Algunas piezas de vajilla, vidrios de ventanas, etcétera, se rompen; pocos casos de agrietamiento de aplanados; caen objetos inestables.• Se observan perturbaciones en los árboles, postes y otros objetos altos.• Se detienen de relojes de péndulo.
VI	<ul style="list-style-type: none">• Sacudida sentida por muchas personas; muchas personas atemorizadas huyen hacia afuera.• Algunos muebles pesados cambian de sitio; pocos ejemplos de caída de aplanados o daño en chimeneas.• Daños ligeros.

VII	<ul style="list-style-type: none"> • Advertido por todos. • La gente huye al exterior. • Daños sin importancia en edificios de buen diseño y construcción. • Daños ligeros en estructuras ordinarias bien construidas; daños considerables en las débiles o mal planeadas; rotura de algunas chimeneas.
VIII	<ul style="list-style-type: none"> • Daños ligeros en estructuras de diseño; considerable en edificios ordinarios con derrumbe parcial; grande en estructuras débilmente construidas. • Los muros salen de sus armaduras. • Caída de chimeneas, pilas de productos en los almacenes de las fábricas, columnas, monumentos y muros. Los muebles pesados se vuelcan. • Arena y lodo proyectados en pequeñas cantidades. • Cambio en el nivel del agua de los pozos. • Pérdida de control en las personas que guían vehículos motorizados.
IX	<ul style="list-style-type: none"> • Daño considerable en las estructuras de diseño bueno; las armaduras de las estructuras bien planeadas se desploman; grandes daños en los edificios sólidos, con derrumbe parcial. • Los edificios salen de sus cimientos. • El terreno se agrieta notablemente. • Las tuberías subterráneas se rompen.
X	<ul style="list-style-type: none"> • Destrucción de algunas estructuras de madera bien construidas; la mayor parte de las estructuras de mampostería y armaduras se destruyen con todo y cimientos; agrietamiento considerable del terreno. • Las vías del ferrocarril se tuercen. • Considerables deslizamientos en las márgenes de los ríos • Invasión del agua de los ríos sobre sus márgenes. • Casi ninguna estructura de mampostería queda en pie. • Puentes destruidos. • Anchas grietas en el terreno. • Las tuberías subterráneas quedan fuera de servicio. • Hundimientos y derrumbes en terreno suave. Gran torsión de vías férreas. Destrucción total. • Ondas visibles sobre el terreno. • Perturbaciones de las cotas de nivel (ríos, lagos y mares). • Objetos lanzados en el aire hacia arriba.

Tabla 6.1. Escala Mercalli original

Escala de Intensidad Rossi-Forel	Escala de Intensidad Mercalli Modificada	Magnitud (Escala Richter)	Aceleración Máxima del Terreno(G's)
I	I No sentido.	< 2.3	< 0.002
II	II. Sentido solamente por algunas personas en posición de descanso, especialmente en pisos altos. Objetos suspendidos oscilan un poco.	2.3 - 2.9	0.002 - 0.003
III	III Sentido en el interior. Muchas personas no lo reconocen como un temblor. Automóviles parados se balancean. Vibraciones como el paso de un camión pequeño. Duración apreciable.	3.0 - 4.1	0.004 - 0.007
IV	IV Sentido en el interior por muchos, en el exterior por pocos. Ventanas, platos, puertas vibran. Las paredes crujen. Vibraciones como el paso de un camión grande; sensación de sacudida como de un balón pesado. Automóviles parados se balancean apreciablemente.	3.7 - 4.2	0.015 - 0.02
V	V Sentido por casi todo el mundo; muchos se despiertan. Algunos platos, ventanas, etc. se rompen; algunas casas de mampostería se agrietan. Objetos inestables volcados. Los péndulos de los relojes se detienen. Las puertas se balancean, se cierran, se abren. Árboles, arbustos sacudidos visiblemente.	4.3 - 4.9	0.03 - 0.04
VI	VI Sentido por todos; muchos se asustan y corren al exterior. Es difícil andar. Ventanas, platos y objetos de vidrio se rompen. Algunos muebles pesados se mueven; se caen algunas casas de mampostería; chimeneas dañadas. Daños leves.	5.0 - 5.6	0.06 - 0.07
VII	VII Todo el mundo corre al exterior. Daños muy pequeños en edificios de buen diseño y construcción; leve a	5.7 - 6.2	0.1 - 0.15

	moderado en estructuras bien construidas; considerable en las mal construidas; algunas chimeneas se rompen. Sentido por conductores.		
VIII	VIII Daño leve en estructuras especialmente diseñadas para terremotos; considerable hasta con colapso parcial en edificios corrientes; mayor en estructuras pobremente construidas. Los paneles de las paredes se salen de los marcos. Se caen chimeneas, monumentos, columnas y paredes. Se viran muebles pesados. Pequeños corrimientos de arena y fango. Cambios en el caudal de fuentes y pozos. Difícil conducir.	6.3 - 6.9	0.25 - 0.3
IX	IX Daño considerable en estructuras de diseño y construcción buena, estructuras bien diseñadas, desplazadas de sus cimientos; mayor en edificios corrientes con colapso parcial y total. Amplias grietas en el suelo. Eyección de arena y barro en áreas de aluvial. Tuberías subterráneas rotas.	7.0 - 7.6	0.5 - 0.55
X	X Algunas estructuras bien construidas en madera y puentes destruidos, la mayoría de las construcciones y estructuras de armazón destruidas con sus cimientos. Grietas grandes en suelo. Deslizamientos de tierra, agua rebasa las orillas de canales, ríos, lagos, etc. Arena y barro desplazados lateralmente.	7.7 - 8.2	> 0.6
	XI Colapso de la mayoría de las estructuras de cemento y hormigón. Puentes y otras vías de transporte seriamente afectadas.	8.3 - 9.0	

	XII Pérdida total en la infraestructura. Grandes masas de rocas desplazadas. Objetos pesados lanzados al aire con facilidad.	> 9.0	
--	--	-------	--

Tabla 6.2 Relación entre magnitud, intensidad y aceleración

6.2.2. Anexo B.2. Nomenclatura del canal de acuerdo con el estándar SEED

A continuación se presentan las tablas de nomenclatura de acuerdo con el estándar SEED, estas tablas son una adaptación del manual del protocolo SEED. Para más detalles sobre esta nomenclatura referirse al documento “SEED reference manual”.

Código de banda	Tipo de banda	Frecuencia de muestreo (Hz)
F	...	≥ 1000 a < 5000
G	...	≥ 1000 a < 5000
D	...	≥ 250 a < 1000
C	...	≥ 250 a < 1000
E	Periodo extremadamente corto	≥ 80 a < 250
S	Periodo corto	≥ 10 a < 80
H	Alta banda ancha	≥ 80 a < 250
B	Banda ancha	≥ 10 a < 80
M	Periodo Medio	> 1 a < 10
L	Periodo largo	≈ 1
V	Periodo muy largo	≈ 0.1
U	Periodo ultra largo	≈ 0.01
R	Periodo extremadamente largo	≥ 0.0001 a < 0.001
P	En orden de 0.1 a 1 día	≥ 0.00001 a < 0.0001
T	En orden de 1 a 10 días	≥ 0.000001 a < 0.00001
Q	Sobrepasa los 10 días	< 0.000001
A	Canal de instrumento administrativo	NA
O	Canal de instrumento opaco	NA

Tabla 6.3 Nomenclatura de la primera letra del canal de acuerdo con la banda y frecuencia de muestreo

Código de instrumento	Tipo de instrumento
H	Sismómetro de alta ganancia
L	Sismómetro de baja ganancia
G	Gravímetro
M	Sismómetro de posición de masa
N	Acelerómetro

Tabla 6.4 Nomenclatura de la segunda letra del canal de acuerdo con el tipo de instrumento

Código de orientación	Orientación
Z N E	Tradicional (Vertical, Norte-Sur, Este-Oeste)
A B C	Triaxial
T R	Transversal, Radial
1 2 3	Componentes ortogonales pero orientaciones no tradicionales
U V W	Componentes opcionales

Tabla 6.5 Nomenclatura de la tercera letra del canal de acuerdo con la orientación del instrumento

6.2.3. Anexo B.3. Filtro de alisado exponencial

A continuación se observa la demostración de las ecuaciones de filtro de alisado exponencial obtenidas para un filtro paso bajo y filtro paso alto, utilizando como modelo un circuito RC.

El filtro paso bajo utilizando un circuito RC se representa de la siguiente forma:

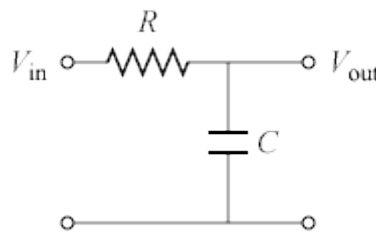


Figura 6.1 Filtro paso bajo

Partiendo del circuito anterior se tiene entonces que

$$f_c = \frac{1}{2\pi RC}$$

Ecuación 6.1 Frecuencia de corte

$$V_{in}(t) - V_{out}(t) = Ri_c(t)$$

Ecuación 6.2 Análisis de malla de circuito

$$i_c(t) = C \frac{dV_c(t)}{dt}$$

Ecuación 6.3 Corriente de un capacitor

$$V_{in}(t) - V_{out}(t) = RC \frac{dV_{out}(t)}{dt}$$

Ecuación 6.4 Sustitución de ecuación 6.2 utilizando la ecuación de corriente en capacitor

De modo que: $V_{in}(t)$ representa las muestras de entrada, $V_{out}(t)$ representa la señal filtrada. Ajustando la ecuación, se toma que $dt = \Delta t$ y $dV_{out}(t) = V_{out}(t) - V_{out}(t - 1)$ de forma tal que se obtiene:

$$V_{out}(t) = V_{in}(t) - RC \frac{V_{out}(t) - V_{out}(t - 1)}{\Delta t}$$

Ecuación 6.5 Ecuación 6.4 ajustada

$$\frac{\Delta t + RC}{\Delta t} V_{out}(t) = V_{in}(t) + RC \frac{V_{out}(t - 1)}{\Delta t}$$

Ecuación 6.6 Ecuación 6.5 ajustada

$$V_{out}(t) = \frac{\Delta t}{\Delta t + RC} V_{in}(t) + \frac{RC}{\Delta t + RC} V_{out}(t - 1)$$

Ecuación 6.7 Ecuación 6.6 ajustada

De modo tal que:

$$\alpha = \frac{\Delta t}{\Delta t + RC}$$

Ecuación 6.8 valor de alpha para el filtro paso bajo

Sustituyendo alpha:

$$V_{out}(t) = \alpha V_{in}(t) + (1 - \alpha) V_{out}(t - 1)$$

Ecuación 6.9 Ecuación de filtro paso bajo para alisado exponencial

De modo que conociendo el valor de la frecuencia de corte y utilizando la ecuación 6.1. Se puede determinar el valor de RC

$$RC = \frac{1}{2\pi f_c}$$

Ecuación 6.10 Valor de RC a partir de la ecuación 6.1

Finalmente el valor de dt, se puede obtener a partir del valor de la frecuencia de muestreo. El cual para el caso de estaciones sísmicas se puede obtener de la tabla 6.3

Para el caso del filtro paso alto se toman las siguientes consideraciones:

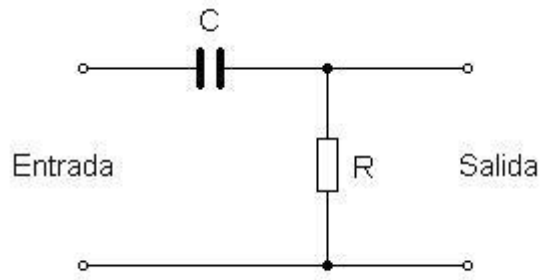


Figura 6.2 Circuito RC para filtro paso bajo

$$V_{out}(t) = Ri_c(t)$$

Ecuación 6.11 Valor del voltaje de salida del circuito

$$V_{out}(t) = \frac{RCdV_c(t)}{dt}$$

Ecuación 6.12 Sustitución de corriente utilizando la ecuación de voltaje de capacitor

$$V_{out}(t) = \frac{RC * d(V_{in}(t) - V_{out}(t))}{dt}$$

Ecuación 6.13 Sustitución del voltaje de capacitor

$$V_{out}(t) = \frac{RC * (V_{in}(t) - V_{in}(t - 1) - V_{out}(t) + V_{out}(t - 1))}{\Delta t}$$

Ecuación 6.14 Sustitución de ecuación 6.13

$$\Delta t * V_{out}(t) + RC * V_{out}(t) = RC * (V_{in}(t) - V_{in}(t - 1) + V_{out}(t - 1))$$

Ecuación 6.15 Sustitución de ecuación 6.14

$$V_{out}(t) = \frac{RC * (V_{in}(t) - V_{in}(t - 1) + V_{out}(t - 1))}{\Delta t + RC}$$

Ecuación 6.16 Sustitución de ecuación 6.15

De modo que se toma que:

$$\alpha = \frac{RC}{\Delta t + RC}$$

Ecuación 6.17 valor de alpha para el filtro paso bajo

$$V_{out}(t) = \alpha * (V_{in}(t) - V_{in}(t - 1) + V_{out}(t - 1))$$

Ecuación 6.18 Ecuación de filtro paso alto

6.2.4. Anexo B.4. Filtro FIR con el método de ventanas

A continuación se presenta la información necesaria para elaborar un filtro FIR por el método de ventanas. La siguiente información está basada en las notas de Hinton (2001).

Primeramente se debe definir qué tipo de filtro se quiere modelar, paso alto, bajo, pasa banda, recorta banda. Una vez definido el tipo de filtro se debe obtener la respuesta ideal del impulso. Para definir la respuesta ideal del impulso se puede utilizar la tabla 6.6

Tipo de filtro	$H_D[n], n \neq 0$	$H_D[n], n=0$
Paso bajo	$2F_c \frac{\sin n\Omega_c}{\Omega_c n}$	$2F_c$
Paso alto	$1 - 2F_c \frac{\sin n\Omega_c}{\Omega_c n}$	$1-2F_c$
Paso banda	$2F_2 \frac{\sin n\Omega_2}{\Omega_2 n} - 2F_1 \frac{\sin n\Omega_1}{\Omega_1 n}$	$2F_2-2F_1$
Rechaza banda	$1 - (2F_2 \frac{\sin n\Omega_2}{\Omega_2 n} - 2F_1 \frac{\sin n\Omega_1}{\Omega_1 n})$	$1-(2F_2-2F_1)$

Tabla 6.6 Respuesta ideal al impulso según el tipo de filtro

Para la tabla 6.6 se toman las siguientes consideraciones:

F_c : Frecuencia de corte del filtro

F_2 : Frecuencia de corte del filtro paso bajo

F_1 : Frecuencia de corte del filtro paso alto

Ω_c : Frecuencia angular de corte normalizada

Para el caso del filtro paso alto se utiliza la respuesta de un filtro pasa todo y se le resta la respuesta de un filtro paso bajo. Esta misma estrategia es utilizada en el filtro rechaza banda donde se resta la respuesta de un filtro pasa banda.

Para obtener la frecuencia de corte angular normalizada se utiliza la ecuación 6.19.

$$\Omega_c = 2\pi(F_p + 0.5\Delta F)$$

Ecuación 6.19 Frecuencia angular de corte normalizada

Donde:

$$F_p = \frac{fp}{fs}$$

Ecuación 6.20 Frecuencia de corte normalizada

$$\Delta F = \frac{\Delta f}{fs}$$

Ecuación 6.21 Ancho de transición normalizado

Para determinar los coeficientes del filtro FIR $h[n]$ se debe de multiplicar los coeficientes de la ventana con la respuesta ideal al impulso.

$$h[n] = w[n] * h_D[n]$$

Ecuación 6.22 Coeficientes $h[n]$ para filtro FIR por método de ventanas

Para obtener la función de la ventana se utiliza la información de la tabla 6.7

Nombre de la función de ventana $w(n)$	Definición matemática
Rectangular	1
Hanning	$0.5 - 0.5 \cos \frac{2\pi n}{N-1}$
Hamming	$0.54 - 0.46 \cos \frac{2\pi n}{N-1}$
Blackman	$0.42 - 0.5 \cos \frac{2\pi n}{N-1} + 0.08 \cos \frac{2\pi n}{N-1}$
Kaiser	$I_0 \frac{\beta \sqrt{1 - \left(\frac{ 2n-N+1 }{N-1}\right)^2}}{-I_0(\beta)}$ Donde: $I_0(x) = \sum_{k=0}^{\infty} \left(\frac{x^k}{2^k k!}\right)^2$

Tabla 6.7 Funciones de ventana

El factor N representa la cantidad de coeficientes del filtro, el cual se obtiene a partir de la tabla 6.8

Nombre de la función de ventana w(n)	Ancho de transición normalizado	
Rectangular	$\frac{0.9}{N}$	
Hanning	$\frac{3.1}{N}$	
Hamming	$\frac{3.3}{N}$	
Blackman	$\frac{5.5}{N}$	
Kaiser	$\beta=4.54$	$\frac{2.93}{N}$
	$\beta=5.65$	$\frac{3.63}{N}$
	$\beta=6.76$	$\frac{4.32}{N}$
	$\beta=8.96$	$\frac{5.71}{N}$

Tabla 6.8 Relación entre cantidad de coeficientes y ancho de transición

Pasos para el diseño del filtro FIR utilizando el método de ventana:

- I. Se toma la respuesta ideal al impulso utilizando la tabla 6.6
- II. Se selecciona un tipo de ventana w(n) según la tabla 6.7
- III. Se calcula la cantidad de coeficientes requeridos N a partir de ΔF y la tabla 6.8
- IV. Se calculan los coeficientes del filtro FIR multiplicando $H_D[n]$ con w(n)

Se debe recordar que el filtro FIR está definido por la ecuación 2.4.3

$$y(i) = \sum_{n=0}^N H(n) * x[i - n]$$

Ecuación 2.23 Ecuación 2.4.3 ajustada para un filtro FIR por método de ventanas

De modo que en el paso IV se ha calculado H(n). Por otro lado el valor de N de la ecuación 2.4.3 es el mismo que se calculó en el punto 3. Este valor N será el que definirá el retraso y número de saltos para el filtro.

7. Referencias bibliográficas

- Agafonkin, V. (2014). Leaflet an open-source JavaScript library for mobile-friendly interactive maps (Versión 0.7.3) [Software]. Recuperado de <http://leafletjs.com/>
- Braile, L. (2014). *The AS-1 Seismograph Operation, Filtering, S-P Distance Calculation, and Ideas for Classroom Use*. Recuperado de <http://web.ics.purdue.edu/~braile/edumod/as1mag/as1mag2.pdf>
- Earth Scope (s.f) Transportable Seismic Station Design Diagram. [Figura]. Recuperado de: <http://www.earthscope.org/assets/uploads/pages/transportable-seismic-station-design.jpg>
- Fisher, T. (1999). *Interactive Digital Filter Design*. Recuperado de <https://www-users.cs.york.ac.uk/~fisher/mkfilter/>
- GEOFON program. (2009). *Seedlink*. Helmholtz Centre Potsdam, GFZ German Research Centre for Geosciences. Recuperado de <http://www.seiscomp3.org/wiki/doc/applications/seedlink>
- Havskov, J., & Alguacil, G. (2002). *Instrumentation in Earthquake Seismology*. Recuperado de <http://www.ipgp.fr/~brunet/Seismometers.pdf>
- Hernández Castro, F., & Monge Fallas, J. (2013). *iReal posibilidades*. Recuperado de http://revistas.tec.ac.cr/index.php/investiga_tec/article/view/751
- Hernández Castro, F. (2015). *iReal 3.0: visualization of Costa Rica's tectonic plates*. Recuperado de <http://skizata.com/ireal-3.0.html>
- Hinton, O. (2001). Design of FIR Filters. “EEE305”, “EEE801 Part A”: *Digital Signal Processing*. Recuperado de <https://www.staff.ncl.ac.uk/oliver.hinton/eee305/Chapter4.pdf>
- Incorporated Research Institutions for Seismology. (s.f.). *¿Cómo Trabaja un Sismógrafo?*. Recuperado de https://www.iris.edu/hq/files/publications/brochures_onepaggers/doc/SP_1pager_7.pdf
- Incorporated Research Institutions for Seismology. (2008). *How do seismographs work?* Recuperado de http://www.iris.edu/hq/programs/education_and_outreach/animations/8
- Incorporated Research Institutions for Seismology. (2013). *Libslink* (Versión 2.4b) [Software]. Washington, DC. Recuperado de <http://ds.iris.edu/ds/nodes/dmc/software/downloads/libslink/>
- Incorporated Research Institutions for Seismology. (s.f). *SeedLink*. Recuperado de <http://ds.iris.edu/ds/nodes/dmc/services/seedlink/>

- International Federation of Digital Seismograph Networks, Incorporated Research Institutions for Seismology, & United States Geological Survey. (2012). *SEED Reference Manual*. Recuperado de http://www.fdsn.org/seed_manual/SEEDManual_V2.4.pdf
- Lehtinen, P. (2014). Jansson a C library for encoding, decoding and manipulating JSON data. (Versión 2.7) [Software]. Recuperado de <http://www.digip.org/jansson/>
- Laboratorio de Ingeniería Sísmica. (2011). *Instrumentación: El Sismógrafo*. Recuperado de <http://www.lis.ucr.ac.cr/index.php?id=10>
- Laboratorio de Ingeniería Sísmica. (2011). *Magnitud e Intensidad*. Recuperado de <http://www.lis.ucr.ac.cr/index.php?id=11>
- Laboratorio de Ingeniería Sísmica. (2007). *Relationship Between Peak Ground Acceleration and Modifies Mercalli Intensity in Cost Rica*. San Pedro Montes de Oca, Costa Rica.: Revista Geológica de América Central. Recuperado de www.redalyc.org/articulo.oa?id=45437345006
- Laboratorio de Ingeniería Sísmica. (2012). *Terremoto del 05 de setiembre del 2012*. Recuperado de <http://www.lis.ucr.ac.cr/pdf/set052012/>
- Laboratorio de Ingeniería Sísmica. (2011). *Tipos de onda*. Recuperado de <http://www.lis.ucr.ac.cr/index.php?id=7>
- Laursen, O., & Schnur, D. (2014). Flot Attractive JavaScript plotting for jQuery (Versión 0.8.3) [Software]. Retrieved from <http://www.flotcharts.org/>
- Otto, M. (2015). Bootstrap (Versión 3.3.5) [Software]. Recuperado de <http://getbootstrap.com/>
- Observatorio Vulcanológico y Sismológico de Costa Rica. (s.f) *¿Cómo nace?*. Recuperado de <http://www.ovsicori.una.ac.cr/index.php/ovsicori/como-nace>
- Monge Fallas J. (s.f). *Proyecto iReal*. Recuperado de <http://ic-itcr.ac.cr/cic/index.php/proyectos/proyectos-actuales/ireal>
- Punskaya, E. (2009). *Design of IIR Filters*. Cambridge, Reino Unido. Recuperado de <http://fisika.ub.ac.id/web/sites/default/files/L08-Design%20of%20IIR%20Filter.pdf>
- Punskaya, E. (2009). *Design of FIR Filters*. Cambridge, Reino Unido. Recuperado de http://www.vyssotski.ch/BasicsOfInstrumentation/SpikeSorting/Design_of_FIR_Filters.pdf
- Quantum GIS Development Team. (2015). QGIS Un Sistema de Información Geográfica libre y de Código Abierto (Versión 2.10.1) [Software]. Recuperado de <http://www.qgis.org/es/site/>

- Quantum GIS Development Team. (2014). *Spatial Analysis (Interpolation)*. Recuperado de https://docs.qgis.org/2.2/en/docs/gentle_gis_introduction/spatial_analysis_interpolation.html
- Red Sísmica de Puerto Rico. (s.f.). *Información sísmica*. Recuperado de http://redsismica.uprm.edu/Spanish/informacion/sisnotas_tam.php
- Red Sísmica de Puerto Rico. (s.f.). *Ondas Sísmicas*. Recuperado de http://redsismica.uprm.edu/Spanish/educacion/sisnotas_ondas.php
- Stein, S. & Wysession, M. (2003). *An Introduction to Seismology, Earthquakes, and Earth Structure*, Capítulo 6: Seismograms and Signals, Wiley-Blackwell Publishing
- Urs, S. (2015). Canvasjs (Versión 1.7) [Software]. Recuperado de <http://canvasjs.com/>
- U.S. Geological Survey. (2012). *Ground Shaking Animations*. Recuperado de <http://earthquake.usgs.gov/regional/nca/simulations/>
- Wickert, M. (2011). FIR Filters. In *ECE2610 Introduction to Signal and Systems*. Recuperado de http://www.eas.uccs.edu/wickert/ece2610/lecture_notes/ece2610_chap5.pdf
- Wickert, M. (2010). IIR Filters. In *ECE2610 Introduction to Signal and Systems*. Recuperado de http://www.eas.uccs.edu/wickert/ece2610/lecture_notes/ece2610_chap8.pdf
- Zaitsev, S. (2015). JSMN [Software]. Recuperado de <http://zserge.com/jsmn.html>