

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS



**ADQUISICION Y VISUALIZACION DE DATOS DE LA RED SISMICA DEL
NOROESTE DE MEXICO**

TESIS

QUE PARA OBTENER EL TITULO DE

LICENCIADO EN CIENCIAS COMPUTACIONALES

PRESENTA:

SERGIO MANUEL ARREGUI OJEDA

ENSENADA, BAJA CALIFORNIA, ENERO DE 2004

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS

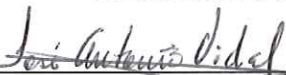
ADQUISICIÓN Y VISUALIZACION DE DATOS DE LA RED SÍSMICA DEL
NOROESTE DE MÉXICO

TESIS PROFESIONAL

QUE PRESENTA

SERGIO MANUEL ARREGUI OJEDA


APROBADO POR:



DR. JOSE ANTONIO VIDAL VILLEGAS
PRESIDENTE DEL JURADO




M. C. JOSE IGNACIO ASCENCIO LOPEZ
SECRETARIO



M. C. MIGUEL ANGEL IBARRA RIVERA
1.er. VOCAL



M. C. LUIS ENRIQUE VIZCARRA CORRAL
2do. VOCAL



M. C. MARIA VICTORIA MEZA KUBO
3er. VOCAL

AGRADECIMIENTOS

A mis padres y hermanos por el apoyo incondicional que me brindaron desde el inicio de mi carrera, fue gracias a su esfuerzo y confianza que me fue posible terminar.

A Aurora por haberme apoyado desde siempre y estar ahí conmigo. Gracias. Te quiero.

A mis compañeros de generación, en especial a Paola González y a Omar Serrato, con quienes pase momentos muy importantes tanto en el estudio, como personales, los que nunca olvidare.

A mis amigos, por brindarme su apoyo en esos momentos que más lo necesitaba, y cuando no también, gracias por estar ahí. En especial gracias a Julio, Eri, Angel, Martain, Roberto, Sukey, Gina, los del Castillo, a los del 325, los del 305, los del 1, a mis roomies y a todos aquellos con los que compartí algún momento..o cerveza.

Al Dr. Antonio Vidal, por haberme tenido la confianza y la paciencia durante el desarrollo del presente trabajo.

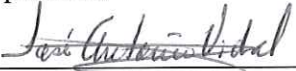
A mis compañeros de trabajo, cuya valiosa experiencia fue esencial para la realización de este trabajo.

Al CICESE por haberme dado la oportunidad de integrarme a su equipo de trabajo.

RESUMEN de la Tesis de SERGIO MANUEL ARREGUI OJEDA presentada como requisito parcial para la obtención de la Licenciatura en Ciencias Computacionales. Ensenada, Baja California, México. 19 de enero de 2004.

ADQUISICIÓN Y VISUALIZACIÓN DE DATOS DE LA RED SÍSMICA DEL NOROESTE DE MÉXICO

Resumen aprobado:



Dr. José Antonio Vidal Villegas
Director de Tesis



M.C. José Ignacio Ascencio López
Co-director de Tesis

La adquisición y visualización de datos en tiempo real con posibilidades de distribución en una red local de datos encuentra una plataforma excelente en precios y recursos de cómputo cuando se utiliza un sistema operativo de libre distribución como Linux. Con el propósito de actualizar el sistema de adquisición de datos para la Red Sísmica del Noroeste de México (RESNOM) se desarrolló un sistema de adquisición nuevo complementado con programación para la visualización de las señales sísmicas registradas. Aunque se tenía un sistema antecesor que en su tiempo fue funcional (que operaba sobre OS-9 en convivencia con MS-DOS) fue necesaria su actualización por la obsolescencia del hardware, su inestabilidad y porque eran necesarias otras funcionalidades que el viejo sistema estaba lejos de cumplir. Se escogió Linux por ser el sistema operativo que cumplía mejor los requerimientos deseados: manejo de señales en tiempo real, sistema multiusuarios y multitareas, además de ser un sistema confiable, entre otros. En el sistema de adquisición desarrollado se reciben las señales provenientes de 12 estaciones de periodo corto y de una estación de periodo largo. Estas señales se suministran a la computadora a través de los puertos serie; debido a que ésta sólo cuenta con dos puertos, fue necesario incrementar su número mediante el uso una tarjeta multipuerto. Esta tarjeta permitió incrementar la capacidad de la computadora hasta diez puertos serie. El manejo de las señales: decodificación, cambio de formato e inclusión de marcas de tiempo, fue posible gracias al desarrollo de programación en lenguaje C. Una vez que se tienen las señales en el formato apropiado, se implementó un algoritmo de detección basado en el cociente de promedios cortos entre promedios largos de la señal recibida. Este algoritmo se aplica a cada una de las señales y cuando se cumple un criterio de disparo, el sistema procede a declarar un evento y a guardarlo en disco. Los programas para la visualización de las señales en tiempo real permiten inspeccionarlas para saber el estado y la calidad de éstas (señal sin interferencia, nivel de ruido, continuidad de la señal). Los programas para la visualización de las señales grabadas permiten hacer la depuración de archivos generados por el sistema, la conversión de formato y la transferencia de archivos a otras computadoras. Con el sistema desarrollado ahora el manejo y la visualización de las señales sísmicas es más sencilla con respecto a su antecesor, por la automatización de tareas, su característica multitareas y su interfase gráfica. Por otro lado, con la programación desarrollada para homogenizar la base de datos de RESNOM se pudieron conjuntar los datos de diferentes tipos de señales (periodo corto, periodo largo y banda ancha) en un sólo formato de uso amplio en sismología.

CONTENIDO

I	INTRODUCCIÓN	
I.1	Descripción general de la Red sísmica del Noroeste de México.....	1
I.2	Características del sistema de detección de eventos sísmicos.....	2
I.3	Antecedentes y objetivos.....	4
I.4	Secuencia de presentación del trabajo.....	7
II	MARCO TEORICO	
II.1	Sistemas operativos analizados.....	10
II.1.1	Sistema operativo OS-9.....	10
II.1.2	Sistema operativo Windows.....	11
II.1.3	Sistema operativo Linux.....	12
II.2	Procesamiento digital de señales.....	14
II.2.1	¿Qué es el procesamiento digital de señales?.....	14
II.2.2	Descripción de las señales sísmicas utilizadas.....	14
II.2.3	Secuencia de obtención de los datos digitales en las estaciones de RESNOM.	16
II.3	Comunicación usando el puerto serie.....	19
II.3.1	Introducción al puerto serie.....	19
II.3.2	Conectores y funcionamiento del puerto serie.....	19
II.3.3	Tarjetas multipuerto y microcontroladores.....	21
II.4	La biblioteca GTK+.....	23

III	DESARROLLO DE UN SISTEMA DE RECEPCION DE DATOS DE PERIODO LARGO	
III.1	Introducción y Objetivo.....	26
III.2	Características y formato de transmisión de la señal de periodo largo.....	27
III.3	Señal de tiempo.....	30
III.4	Plataforma seleccionada y software desarrollado.....	32
III.5	Graficado de la señal en tiempo real.....	35
III.6	Programas adicionales.....	36
III.7	Resultados.....	38
IV	DESARROLLO DE UN SISTEMA DE RECEPCION DE DATOS DE PERIODO CORTO	
IV.1	Introducción.....	43
IV.2	Instrumentación de periodo corto y formas de onda obtenidas.....	43
IV.3	Arribo de señales en formato RS232C.....	45
IV.3.1	Señal proveniente de 1 estación.....	46
IV.3.2	Señal proveniente de 3 estaciones.....	47
IV.4	Decodificación de la señal.....	47
IV.5	Desarrollo del sistema de recepción y detección.....	49
IV.5.1	Procesos usados para la adquisición.....	49
IV.5.2	Módulos para el manejo de memoria compartida.....	49
IV.5.3	Módulos de adquisición de datos sísmicos.....	50
IV.5.4	Módulo para la adquisición de la señal de tiempo.....	52

IV.6	Estructuras de datos.....	52
IV.7	Detección automática de sismos.....	54
IV.7.1	Aplicación del algoritmo de detección.....	56
IV.7.2	Función de detección.....	58
IV.7.3	Detección y grabación de eventos sísmicos.....	60
IV.7.4	Grabación de archivos de eventos sísmicos.....	59
IV.8	Graficación en tiempo real.....	62
IV.9	Resultados.....	64
IV.9.1	Algoritmo de detección.....	64
IV.9.2	Eventos registrados.....	66
IV.9.3	Tiempos de arribo de las fases sísmicas.....	68
IV.9.4	Polaridades y amplitudes.....	70
IV.10	Programas adicionales.....	72
V	CONCLUSIONES	76
	Referencias.....	79
	Apéndice A	
I	Figuras de resultados.....	82
	Apéndice B	
I	Implementación en C de los algoritmos usados.....	86
	Apéndice C	
I	Conversión del formato de transmisión hacia el formato RESNOM.....	89
	Apéndice D	
I	Conversión de la cadena de tiempo en ASCII a formato binario.....	93

LISTA DE FIGURAS

Figura		Página
1	Estaciones de la Red Sísmica del Noroeste de México.....	2
2	Esquema del funcionamiento del sistema RESNOM.....	3
3	Ejemplo de un sismograma.....	16
4	Diagrama del funcionamiento de una estación sísmica.....	17
5	Diagrama del proceso de recepción y procesamiento de la señal.....	18
6	Esquema general del proceso de adquisición de los datos de RESNOM.....	24
7	Sismómetros de periodo largo.....	28
8	Diagrama del formato de la señal de periodo largo.....	29
9	Diagrama del formato RESNOM.....	30
10	Diagrama del formato de las marcas de tiempo.....	32
11	Diagrama del módulo que realiza la captura de la señal.....	34
12	Diagrama del módulo que se utiliza para insertar las marcas de tiempo a la señal.....	35
13	Ventana de diálogo para suministrar valores de los parámetros de graficado.....	36
14	Ventana de gráfica en tiempo real de la señal de periodo largo.....	37
15	Ventana del programa Grafica.....	38
16	Localización de los sismos que se usaron para probar el algoritmo de detección.....	41
17	Resultados obtenidos al aplicar el algoritmo de detección a una señal de periodo largo.....	42

18	Instrumentación para el periodo corto.....	44
19	Diferentes formas de onda registradas por la instrumentación de RESNOM.....	45
20	Ilustración de la ruta que siguen los datos desde que son adquiridos hasta su arribo y procesamiento en la computadora.....	46
21	Diagrama de la señal multiplexada.....	47
22	Diagrama del proceso de adquisición de la señal.....	51
23	Esquema que representa el ciclo infinito ejecutado para hacer la detección de los eventos sísmicos.....	61
24	Gráfica de la señal de periodo corto en tiempo real.....	64
25	Gráfica que ilustra el funcionamiento del algoritmo de detección para el evento del 9 de junio de 2002, M=3.2.....	65
26	Gráfica comparativa de eventos registrados en los dos sistemas durante Junio, Julio y Agosto de 2002.....	67
27	Gráfica comparativa del número de eventos detectados por estación.....	68
28	Residuales entre las lecturas de P y S obtenidas con ambos sistemas (OS-9 y Linux).....	70
29	Comparación de la polaridad y las amplitudes.....	71
30	Ventana principal del programa examina.	73
31	Gráfica de las tres componentes de un archivo.	74
32	Pantalla principal del programa monitor.....	75

LISTA DE TABLAS

Tabla		Página
I	Comparación de las características contra las necesidades del sistema y los sistemas operativos que se estudiaron.....	14
II	Conector DB9 para el puerto serial.....	21
III	Tipo de tarjeta multipuerto seleccionada.....	22
IV	Conector RJ12 para el puerto serial.....	23
V	Código de identificación de la señal de periodo largo.....	30
VI	Código de calidad de la señal de tiempo del reloj.....	31
VII	Características de la computadora en donde se implantó el sistema de recepción de periodo largo.....	33
VIII	Parámetros del algoritmo de detección aplicado a una señal de periodo largo registrada.....	39
IX	Coordenadas hipocentrales y magnitud de los sismos seleccionados para probar el algoritmo de detección.....	40
X	Parámetros usados en el algoritmo de detección.....	58
XI	Tipos de eventos seleccionados para una aplicación del algoritmo.....	65
XII	Lecturas de los tiempos de arribo registrado en los dos sistemas para el sismo del 02 de marzo del 2003 a las 00:04.....	69

I INTRODUCCION

I.1 Descripción general de la Red Sísmica del Noroeste de México.

Una de las regiones de mayor actividad sísmica en nuestro país se encuentra en el norte de Baja California, en la frontera principal entre dos placas tectónicas: la del Pacífico y la de Norteamérica. Debido a la interacción de estas dos placas existe una sismicidad continua en la región. El conocimiento actual acerca de esta sismicidad y algunas de sus características (lugares y formas de ocurrencia, por ejemplo), se ha logrado gracias a la contribución de estudios basados en información proporcionada por la Red Sísmica del Noroeste de México -RESNOM- (Vidal y Munguía, 1993). Esta red tiene como propósito registrar la actividad sísmica del norte Baja California y consiste de un conjunto de estaciones sísmicas distribuidas en la región (Figura 1). La red fue desarrollada en la División de Ciencias de la Tierra del Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE).

En la actualidad RESNOM está constituida por 12 estaciones de periodo corto ($f_n = 1$ Hz), tres estaciones de banda ancha ($f = 0.01-40$ Hz) y una estación de periodo largo ($T_n = 15$ s). Las estaciones están formadas por 3 sismómetros orientados en direcciones mutuamente ortogonales (Z, N-S, E-O), además de la instrumentación electrónica utilizada para el acondicionamiento de las señales (etapa de amplificación y filtrado y etapa de digitalización, serialización y modulación) para que puedan ser transmitidas. La información sísmica digital de las estaciones de periodo corto se transmite en forma continua por medio de enlaces de radio hacia el centro de procesamiento ubicado en el CICESE, en Ensenada, B. C. La información recibida se suministra a un sistema formado por una computadora PC con una tarjeta coprocesadora PC68K de Motorola. En el caso de los datos de banda ancha, la

coprocesadora PC68K de Motorola. En el caso de los datos de banda ancha, la transmisión se realiza a través de Internet. En el caso del periodo largo la estación se encuentra ubicada en las instalaciones del CICESE, por lo que su transmisión es directa.

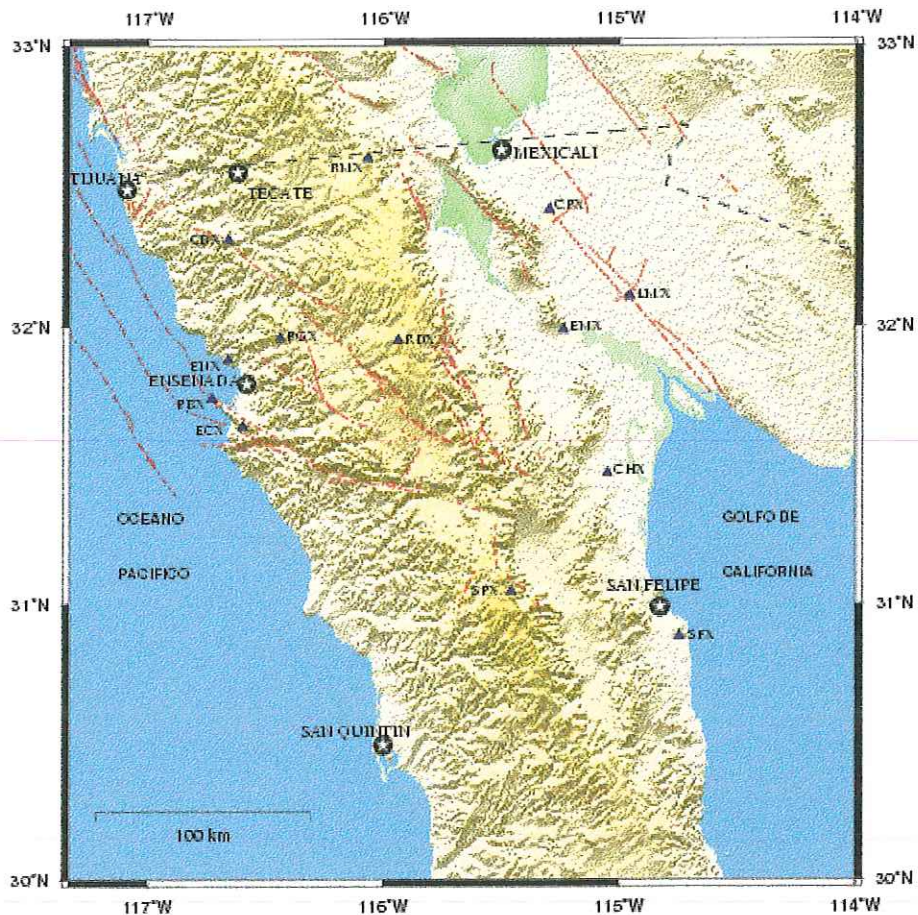


Figura 1. Estaciones de la Red Sísmica del Noroeste de México

I.2 Características del sistema de detección de eventos sísmicos

Con respecto al sistema de adquisición de datos de periodo corto, éste se puede dividir en tres subsistemas: las estaciones sísmicas, el sistema de comunicaciones y la recepción, captura y procesamiento de datos (Hinojosa, 1988). En la Figura 2 se muestra un esquema del sistema RESNOM, y a continuación se presenta una

descripción breve de su funcionamiento

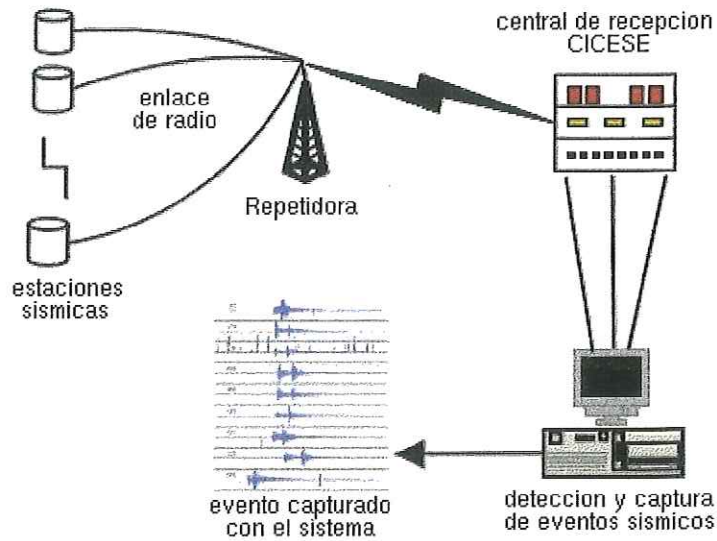


Figura 2. Esquema del funcionamiento del sistema RESNOM.

Las estaciones detectan y acondicionan la información sísmica, la cual es enviada de manera continua por medio del sistema de comunicaciones hacia el centro de procesamiento ubicado en el CICESE. En este centro la señal es demodulada y se convierte a formato RS-232 para poder suministrarla a una computadora. En ella se tiene implementado un algoritmo diseñado para incluir una marca de tiempo cada segundo a la señal entrante. Al presente se reciben las señales digitales de 12 estaciones. La tasa de muestreo de las señales de cada estación es de 120 muestras por segundo, esto es, 40 muestras por segundo por componente. Cada muestra es un entero corto sin signo y cada segundo se le agregan dos enteros cortos, para la marca de tiempo, a las 120 muestras lo que hace un total de 122 muestras por segundo. Una vez que se tiene la información con las marcas de tiempo, se aplica un algoritmo de

predeterminado se procede a grabar la información en disco (Hinojosa *et al.*, 1989).

Actualmente RESNOM está en un proceso de actualización en varias de sus áreas, entre ellas podemos mencionar la instalación de 3 estaciones de banda ancha, la actualización del subsistema de recepción, captura y procesamiento de datos de periodo corto y el diseño de un sistema nuevo para la adquisición de datos de periodo largo. En esta tesis nos ocuparemos de los sistemas de adquisición de datos de periodo corto y de periodo largo. Por las características mencionadas del problema, éste me pareció interesante por tener la oportunidad de trabajar con señales en tiempo real y poner en práctica los conocimientos adquiridos durante mi formación académica. En este caso para el desarrollo de un sistema de adquisición de datos sísmicos.

I.3 Antecedentes y objetivos.

El sistema para la captura y procesamiento de datos sísmicos en el norte de Baja California se inició en 1980 y ha evolucionado a través del tiempo (Hinojosa, 1982). El primer sistema fue desarrollado en un equipo PR1ME 350 y estuvo en funcionamiento por aproximadamente 10 años. Después se optó por mejorarlo e implantarlo sobre una PC, ya que esto reduciría los costos del mantenimiento que se daba al equipo PR1ME (Hinojosa, 1988). Otra de las ventajas de este cambio fue la portabilidad que el nuevo sistema ofrecía. Por lo tanto en 1989 se trasladó el sistema (de la manera más apegada posible al sistema anterior) hacia una PC 386 con una tarjeta coprocesadora PC68K de Motorola. El propósito de esta tarjeta fue el de incrementar en la PC el número de puertos seriales de 2 a 10 y así aumentar el número de señales a suministrar. Adicionalmente la tarjeta PC68K cuenta con cierta

capacidad de procesamiento que le permite encargarse de las tareas de adquisición, detección y almacenamiento de la información recibida. La PC se encarga de proveer al sistema de conectividad con la red interna del CICESE.

La tarjeta coprocesadora PC68K trabaja con el sistema operativo OS-9, el cual convive con el sistema operativo MS-DOS de la PC. El funcionamiento simultáneo de estos dos sistemas operativos provee al sistema tarjeta-PC con los elementos necesarios para realizar las tareas de captura, procesamiento y almacenamiento de la información además de la conectividad con la red interna del CICESE. Esta conectividad es necesaria para el traslado de la información hacia otros equipos (Sun SPARC) en donde se hace el análisis de ésta. El sistema para la adquisición y procesamiento de señales fue implantado con éxito desde 1989, pero en la actualidad (2003) el equipo (la PC y la tarjeta) en el cual fue implantado es ya obsoleto, lo cual provoca que se presenten diversos problemas que interfieren con la recepción de datos. Uno de estos problemas son las frecuentes “caídas del sistema”, ocasionando con ello que se dejen de recibir datos. Debido a lo antiguo del equipo, en ocasiones era necesario ocupar hasta un día en repararlo. Por otro lado, la tarjeta PC68K ya no existe en el mercado y se tiene saturación de los puertos serie, de tal forma que ya no es posible anexar información proveniente de más estaciones.

Otro de los problemas que se presenta con ese sistema es la lentitud con la que se realiza el graficado de la información recolectada, ocasionando pérdida de tiempo en la depuración de ésta. Un inconveniente más de este sistema se presenta en la transferencia de archivos hacia otras computadoras. Para hacer dicha transferencia es necesario interrumpir el proceso de adquisición y cambiar de sistema operativo OS-

9 a MS-DOS. Estando en MS-DOS, se sigue con el proceso de adquisición en memoria pero en caso de ocurrir algún sismo en ese momento, éste no se registraría debido a que MS-DOS tiene el control para la escritura en disco. Un problema adicional es que en ocasiones, cuando falla el sistema OS-9, es necesario formatear el disco y volver a hacer la instalación del sistema operativo y el de adquisición. Mientras se corrigen los desperfectos del equipo (formateo y reinstalación del sistema operativo) pueden ocurrir sismos que no se registran en el sistema y por tanto se presenta pérdida de información.

Un punto importante a considerar para el diseño del sistema de adquisición nuevo es que los sistemas de cómputo actuales permiten hacer mejoras sustanciales, como es el realizar varias tareas de manera simultánea. En nuestro caso ejemplos de estas tareas son: procesamiento de los datos sísmicos grabados, visualización de las señales que arriban, conexión a Internet, entre otras.

El objetivo principal de este trabajo consiste en actualizar y mejorar los sistemas de adquisición y visualización de datos sísmicos de periodo corto y periodo largo con base en el sistema operativo Linux. Este sistema operativo nos permitirá subsanar los problemas mencionados y, por otro lado, nos dará la capacidad de realizar varias tareas de manera simultánea y de agilizar el proceso de depuración de la información.

Los objetivos específicos son:

- Hacer la programación para la recepción y procesado de las señales sísmicas de

periodo corto y periodo largo.

- Implementar un algoritmo de detección, que se encargue de analizar las señales de periodo corto y periodo largo que llegan a las computadoras.
- Desarrollar una interfaz que nos permita manejar los parámetros del sistema para: modificar el umbral de disparo del algoritmo, establecer funciones de detección que determinen cuando se declara un evento, seleccionar las señales de las estaciones que van a estar en detección y las que se van a grabar en disco.
- Desarrollar herramientas para la visualización tiempo real de las señales de periodo corto y periodo largo.
- Desarrollar programas para la conversión de formatos de uso común en sismología.
- Desarrollar una aplicación que nos permita hacer la depuración de la información de una manera rápida y eficiente.

I.4 Secuencia y presentación del trabajo.

Este trabajo consta de tres partes, la primera documenta la investigación realizada para definir la solución más apropiada. Se colectó información acerca de los sistemas operativos OS-9, Windows y Linux. Después de analizar las características de cada uno de estos sistemas, optamos por trabajar con el sistema operativo Linux. Esta sección incluye además una introducción breve al procesamiento digital de señales, la cual será de utilidad para entender el manejo de las señales sísmicas. Después obtuvimos información acerca del manejo de los puertos seriales, ya que es a través de ellos por donde se realiza la adquisición de los datos. Finalmente se explica brevemente acerca de la biblioteca para gráficos GTK+,

la cual fue utilizada para la graficación de las señales.

En la segunda parte se describe el procedimiento seguido para resolver el primer problema, el cual consistía en desarrollar un sistema de adquisición de datos de periodo largo. La solución de este caso nos sirvió de experiencia en el manejo de señales sísmicas. Las señales de periodo largo son generadas sólo por una estación y se digitalizan a razón de una muestra por segundo por canal. Se describe además, la forma en como se incluyen las marcas de tiempo en éstas señales. Las características de las señales de periodo largo, el formato usado para grabarlas así como la programación desarrollada para su adquisición se presentan en esta sección. Finalmente, se presenta un ejemplo del sistema desarrollado, así como una discusión de los resultados obtenidos.

En la tercera parte se discute el problema de la actualización del sistema de captura y procesamiento de datos de periodo corto. La experiencia previa adquirida en la sección anterior fue de gran utilidad en el desarrollo de esta otra. En esta sección explicamos los diferentes tipos de señales recibidas y el tratamiento dado a ellas: conversión de formato, demultiplexación e identificación de cada componente. La señal del tiempo es muy importante para poder hacer la localización epicentral de los sismos registrados. Se muestran los algoritmos usados para hacer la captura de la señal y se describen los algoritmos usados para la detección automática de sismos. A continuación se muestran unas gráficas con diferentes tipos de eventos sísmicos (locales, regionales y telesismos) y la manera en como funciona el algoritmo de detección con cada uno de ellos. Se muestra además una comparación de los resultados obtenidos entre el sistema nuevo y el anterior con el fin de establecer las

ventajas y desventajas de ambos. Finalmente, se describe una utilidad hecha para hacer la depuración de la información grabada.

En la sección de conclusiones se comentan los resultados del trabajo realizado, las ventajas que se obtienen de la utilización de este nuevo sistema. Se comprueba que la elección del sistema operativo Linux fue apropiada para este caso. Se menciona que el sistema de adquisición desarrollado fue satisfactorio y que está actualmente en funcionamiento.

Se consideró conveniente incluir varios apéndices en los que se muestran gráficas complementarias que ilustran el funcionamiento del algoritmo de detección, la secuencia de conversión de formato para las señales sísmicas y para la señal de tiempo, así como la parte más importante de los programas desarrollados.

II MARCO TEORICO

II .1 Sistema operativos analizados

La primera etapa del trabajo consistió en buscar sistemas operativos como posibles candidatos para hacer el desarrollo del sistema de adquisición. Las opciones seleccionadas fueron Windows, OS-9 y Linux. Una descripción de los resultados del análisis hecho se presenta a continuación.

II . 1. 1 Sistema operativo OS-9

La opción con OS-9 parecía ser la mas apropiada, debido a que el sistema de adquisición de datos anterior se había desarrollado sobre este sistema operativo (OS-9 versión 2.4). Como toda tecnología, éste sistema operativo ha presentado avances significativos con respecto a la versión utilizada en el sistema tarjeta-PC. Sin embargo ahora tiene el inconveniente de que el desarrollo se tiene que hacer sobre una computadora huésped para después de terminada la aplicación, pasarla a una computadora objetivo.

Aún con los avances hechos en el sistema operativo OS-9 (Enhaced OS-9 for x86), se tiene un soporte limitado para hardware. Sólo se le pueden instalar ciertos tipos de tarjetas de red y no soporta discos duros mayores de 8.5 GB (Microware, 2000). El sistema operativo OS-9 se especializa en sistemas de cómputo que trabajan sólo para una aplicación o en sistemas de tiempo real y no en sistemas de propósito general como es nuestro caso. Además de la captura de los sismos, se quiere realizar su análisis en la misma computadora con el uso de programación diseñada para ello. Por otro lado, la configuración de los dispositivos periféricos del sistema OS-9 sigue

siendo complicada y sólo tiene soporte para algunas tarjetas multipuerto a diferencia de la variedad que ofrecen Linux y Windows.

El sistema operativo OS-9 sigue soportando a las tarjetas 68K de Motorola, sólo que éstas ahora vienen como una computadora completa sin tener la necesidad de compartir su espacio con una PC. No obstante si utilizáramos esta opción seguiríamos dependiendo de una tarjeta de la cual no existe certeza de tener soporte en el futuro. Por las razones previamente mencionadas se descartó el sistema operativo OS-9 para hacer la actualización del sistema de adquisición de datos sísmicos.

II.1.2 Sistema operativo Windows

La otra opción para el desarrollo del sistema de adquisición fue Windows, pero este sistema no es muy bueno manejando aplicaciones de tiempo real. De acuerdo con el documento de National Instruments (2003), “en sistemas operativos de tiempo real, las interrupciones y eventos son jerarquizados y los eventos con la mayor prioridad se ejecutan antes que los de prioridad menor. En Windows NT/98/95 los manejadores de dispositivos responden a interrupciones, aunque la tecnología multihilo ha incrementado la confiabilidad y manejo de interrupciones, aún hay situaciones en las que el manejador de dispositivo no tiene prioridad sobre otros eventos menos críticos”. Por esto Windows no puede garantizar que va a responder a un evento siempre dentro de un mismo tiempo. Para llevar a cabo el desarrollo de un sistema como el que se plantea (en el que hay que dar prioridad a la adquisición de las señales y la aplicación de un algoritmo de detección a las señales que van

arribando) implicaría trabajar con software propietario y hardware especializado, lo cual nos llevaría de nuevo al caso del sistema que se pretende reemplazar, además de que se aumentarían los costos del desarrollo puesto que aparte del sistema operativo, se tendrían que adquirir las herramientas para hacer el desarrollo. Esto no quiere decir que no existan sistemas de tiempo real que usen Windows, un buen ejemplo de un sistema de adquisición en tiempo real usando Windows lo constituye el sistema de adquisición de datos de banda ancha que tiene RESNOM. En este sistema se realiza todo el procesamiento de la información en dispositivos separados de la computadora y ésta sólo se encarga de estarla recibiendo. En este caso la computadora no procesa los datos sino que se encarga del manejo de la información.

II.1.3 Sistema operativo Linux

En 1991 Linus Torvalds desarrolló el sistema operativo Linux, el cual es distribuido gratuitamente y se comporta como el sistema operativo Unix. Linux fue diseñado específicamente para la plataforma PC y toma ventaja de su diseño para darle a sus usuarios un funcionamiento comparable con las estaciones de trabajo Unix. Linux es un sistema operativo completo, multitareas, multiusuarios. El código de Linux es abierto y la mayoría de sus aplicaciones son gratuitas, aunque también existen aplicaciones comerciales.

El *kernel* de Linux se diseñó originalmente para el procesador Intel 80386 como un sistema multitareas, siendo ésta una de sus características principales. El *kernel* de Linux es un *kernel* monolítico, esto es, todos los manejadores de dispositivos son parte propia del *kernel*. Linux hace un buen uso de las

características multitareas avanzadas que vienen prediseñadas en el conjunto de instrucciones del CPU. Linux también soporta diversos tipos de sistemas de archivos (NFS, vfat, MS-DOS, OS/2 entre otros). El sistema de archivos de Linux, *extfs2*, está diseñado para hacer el mejor uso posible del disco.

Otras ventajas que ofrece Linux radica en que es económico actualizarlo de una manera periódica y por existir un gran conjunto de herramientas para el desarrollo de aplicaciones sobre ésta. Linux es un sistema operativo robusto, que nos permite la ejecución simultánea de varias tareas, además de tener un ambiente gráfico, lo que no teníamos con el sistema anterior basado en MS-DOS y en OS-9. Con base en lo anterior seleccionamos trabajar en una computadora PC con sistema operativo Linux. Otra de las razones por las que se escogió Linux es por su compatibilidad con Unix, puesto que algunos de los programas que se utilizan en sismología para el análisis de la información son ejecutados en estaciones de trabajo Unix.

Además de estas características, Linux cuenta con compiladores, editores de texto y una amplia documentación para el uso de éstos. Todo lo anterior se encuentra en cualquier distribución (Slackware, RedHat y Mandrake son algunas de ellas). Este hecho aunado a la naturaleza de código abierto (*Open Source*) de Linux, lo hicieron el sistema operativo más apropiado para el desarrollo de nuestro sistema de adquisición de datos sísmicos. En la Tabla I se presenta una comparación de las características de cada uno de los sistemas operativos analizados.

Tabla I. Comparación de las características contra las necesidades del sistema y los sistemas operativos que se estudiaron

Necesidades	Linux	OS-9	Windows	DOS
1. Multiusuarios y multitareas	✓	✓	✓	×
2. Soporte de dispositivos	✓	No todos	✓	✓
3. Compiladores, editores, documentación.	✓	✓	Sí, pero no viene incluido	Sí, pero no viene incluido
4. Ambiente gráfico	✓	✓	✓	×
5. Acceso a Internet	✓	✓	✓	✓
6. Sistema confiable	✓	✓	Depende de la versión	✓
7. Servidor	✓	×	✓	×
8. Manejo de señales en tiempo real	✓	✓	✓	×
9. Conectividad entre sistemas operativos	✓	×	Sí, pero no viene incluido	×

II.2 Procesamiento digital de señales.

II.2.1 ¿Qué es el procesamiento digital de señales?

El procesamiento digital de señales se diferencia de otras áreas en las ciencias de la computación por el tipo único de datos que utiliza: señales. En la mayoría de los casos, estas señales se originan como datos registrados por sensores del mundo real: vibraciones sísmicas, imágenes visuales, ondas de sonido, etcétera. El procesamiento digital de señales comprende las matemáticas, los algoritmos, y las técnicas usadas para manipular estas señales una vez que han sido convertidas a una

forma digital. Esto incluye una amplia variedad de metas, tales como: mejoramiento de imágenes visuales, reconocimiento y generación del habla, compresión de datos para almacenamiento y transmisión, entre otros (Smith, 1999).

Las raíces del procesamiento digital de señales empiezan en los años 60 y 70 cuando aparecieron las primeras computadoras. Las computadoras eran caras en esa época y el procesamiento digital de señales estaba limitado a solamente unas cuantas aplicaciones. La revolución de las computadoras personales de los 80 y 90 causó una explosión de aplicaciones nuevas de procesamiento digital de señales (Smith, 1999). En nuestro caso usamos el procesamiento digital de señales en el campo de la sismología.

II.2.2 Descripción de las señales sísmicas utilizadas.

Una señal es una descripción de cómo un parámetro está relacionado con otro. Puesto que los dos parámetros pueden tomar un rango continuo de valores, llamaremos a esto una señal continua. Al pasar esta señal a través de un convertidor analógico a digital, cada uno de los dos parámetros es cuantificado. Por ejemplo, de las señales que se van a tratar en este trabajo, una de ellas se convierte a formato digital utilizando un convertidor analógico digital de 12 bits a una razón de 40 muestras por segundo. La escala máxima es de 4096 posibles niveles binarios. El segundo parámetro es el tiempo el cual está definido en incrementos de 25 milisegundos. La señal resultante de este proceso es una señal discreta o digitalizada.

En sismología se utilizan series de tiempo que representan alguno de los parámetros de movimiento del terreno. Estos parámetros pueden ser aceleración, velocidad o desplazamiento del terreno. En el caso de las señales provenientes de las estaciones

de movimiento del terreno. Estos parámetros pueden ser aceleración, velocidad o desplazamiento del terreno. En el caso de las señales provenientes de las estaciones de RESNOM trabajamos con sismogramas (registros de velocidad). Un ejemplo de un sismograma se muestra en la Figura 3.

En los capítulos siguientes se hablará del tratamiento y manejo que se hace de estas señales y la manera en como diferenciamos la señal sísmica de aquella que no lo es.

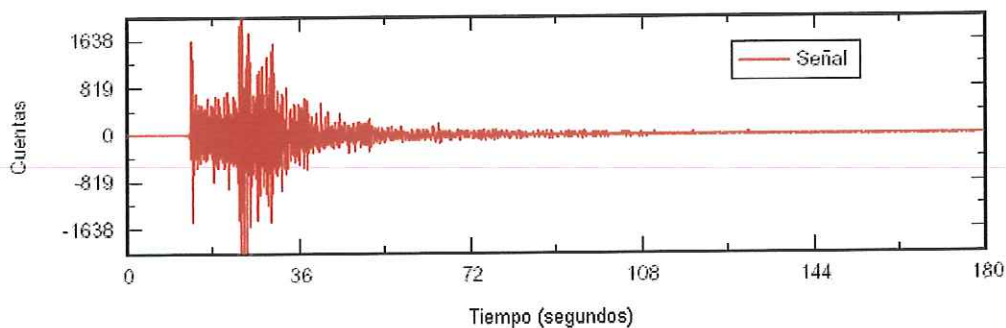


Figura 3. Ejemplo de un sismograma.

II.2.3 Secuencia de obtención de los datos digitales en las estaciones de RESNOM

Un diagrama a bloques del funcionamiento de las estaciones de RESNOM se muestra en la Figura 4. En cada estación sísmica de RESNOM se tienen tres sismómetros que generan señales analógicas de manera continua. La señal producida por los sismómetros pasa por una etapa de amplificación y filtrado. La amplificación apropiada para cada estación se selecciona en función del nivel del ruido característico del lugar. Amplificaciones típicamente usadas son: 40, 160 y 320.

Previamente a la conversión de analógico a digital, se efectúa un filtrado *antialiasing* en el cual se utiliza un filtro pasabajas tipo Butterworth de 5 polos, con una frecuencia de corte de 15 Hz. Una vez filtrada, la señal pasa por un convertidor analógico/digital (A/D) de 12 bits. (Vidal y Munguía, 1993). El intervalo de muestreo es de 40 muestras por segundo para cada componente. La señal procedente del convertidor A/D pasa a continuación por un formateador que le añade bits de identificación de canal y de sincronía. La salida para cada muestra consiste en dos palabras en serie de 8 bits cada una, en formato de transmisión asíncrona. La señal ya serializada pasa al modulador de frecuencia en el que los niveles binarios se convierten en tonos. La señal modulada se transmite por radio (VHF, *Very High Frequency* y UHF, *Ultra High Frequency*), a una velocidad de 2400 bits por segundo, hacia el centro de procesamiento ubicado en el CICESE. Las frecuencias típicas de transmisión están comprendidas en los intervalos de 150 a 22 Mhz en VHF y de 450 a 460 Mhz en UHF. (Vidal y Munguía, 1993).

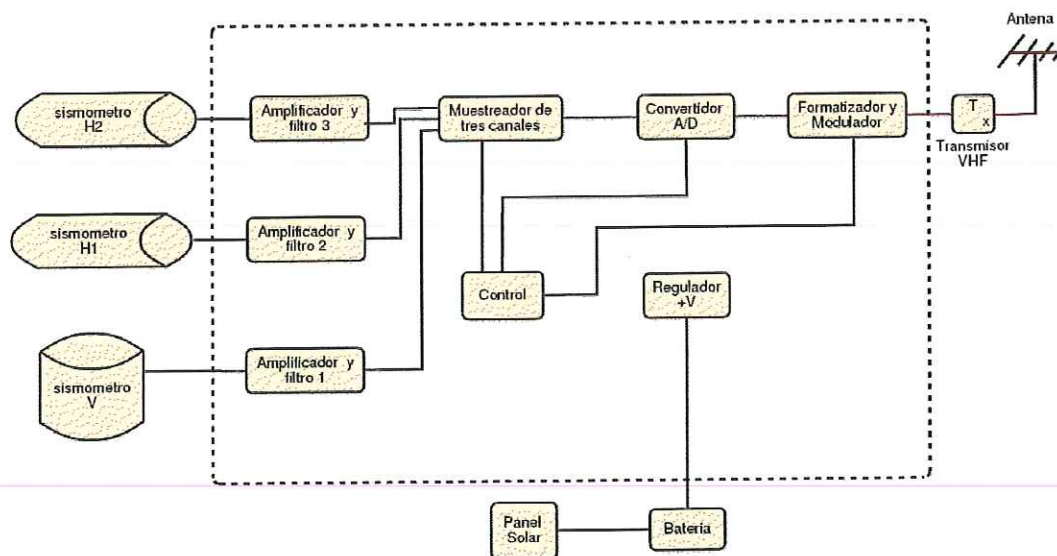


Figura 4. Diagrama del funcionamiento de una estación sísmica.

A su arribo al centro de procesamiento, las señales de todas las estaciones pasan por una etapa previa de demodulación y cambio de formato antes de entrar a la computadora. Adicionalmente a su grabación en formato digital, y como una forma de respaldo para casos de fallas en la computadora receptora, las señales de 4 estaciones se reconvierten a formato analógico y se grafican en papel (Figura 5).

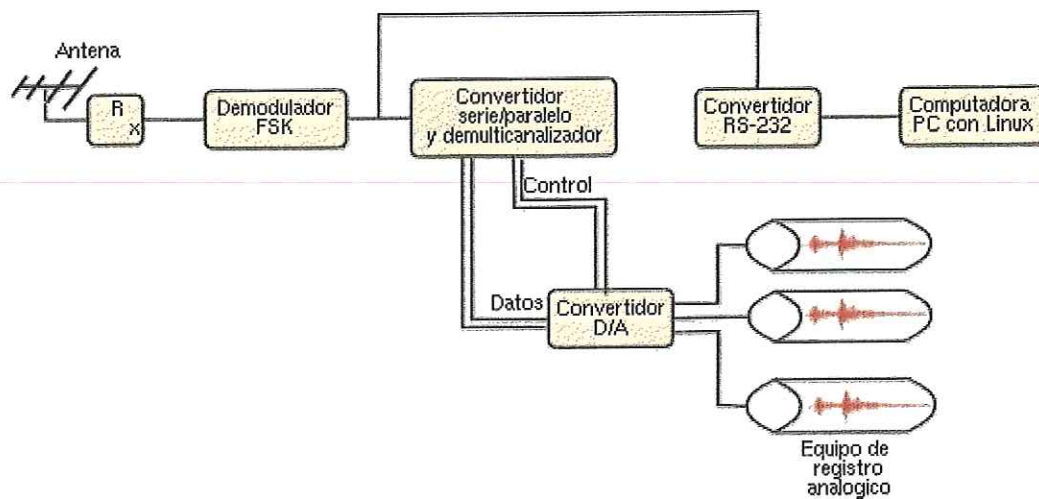


Figura 5. Diagrama del proceso de recepción y procesamiento de la señal.

Otro punto importante de mencionar es que todas las señales vienen multicanalizadas, con esto se quiere decir que tenemos tres señales en un mismo canal. La señal de las tres componentes de cada estación llega a la computadora por el mismo canal, en la computadora se tienen implementados algoritmos que se encargan de separarla en sus tres componentes. También existen otras estaciones que además de estar multicanalizadas, vienen multiplexadas. Se dice que están

intercalados, por lo que se pueden transmitir varias señales usando un mismo canal. Este proceso se discutirá con detalle en el Capítulo IV de la tesis.

II.3 Comunicación usando el puerto serie

II .3.1 Introducción al puerto serie.

Una interface es el punto de contacto entre dos ambientes diferentes, en este caso entre la circuitería de la computadora y los dispositivos externos. Una interface es llamada puerto de Entrada/Salida (E/S), o simplemente puerto. El principal objetivo de una interface es el de proveer un medio para la transferencia de datos hacia ambientes externos (Rad Data Communications, 1995).

Para recibir los datos por medio del puerto serie hay que saber las características de las señales que se reciben para poder programarlo. El puerto serie se maneja por medio de la norma RS232-C. En su forma más simple, esta interface consiste de sólo dos alambres, uno para transmitir datos, más otro para tierra. La tierra es la referencia absoluta del voltaje para toda la interface del circuito, el punto en el circuito en el que todos los voltajes son medidos.

II.3.2 Conectores y Funcionamiento del puerto serie.


Existen dos tipos básicos de comunicaciones seriales: síncrona y asíncrona. En la comunicación síncrona, dos dispositivos se sincronizan inicialmente entre ellos, y continuamente se están enviando caracteres entre sí para estar en sincronía. En la comunicación asíncrona (sin sincronía) no se requiere de estar enviando y recibiendo caracteres de sincronía. No obstante, el principio de cada byte de datos

debe ser identificado por bits de inicio y al final otro de término. El bit de inicio indica cuando está por empezar, y el bit de paro indica cuando termina. El que se necesiten estos dos bits adicionales es la causa de que la comunicación asíncrona sea más lenta que la síncrona, sin embargo tiene la ventaja de que el procesador no tiene que trabajar con caracteres adicionales de espera. Los puertos seriales en las PC son dispositivos asíncronos y por lo tanto solo soportan comunicaciones seriales asíncronas. (Taltech, 2003). Como ya se mencionó en la sección II.2.3, las señales sísmicas se transmiten en forma asíncrona. Por tanto se va a extender la explicación del manejo de la norma RS232-C para entender el suministro de las señales a la computadora.

Las especificaciones comunes para el puerto serial están en la norma RS232-C. El conector para el puerto serial se localiza en la parte trasera de la PC como uno o dos conectores de 9 *pins* (ver Tabla II). Este conector incluye electrónica asociada que debe de producir señales conforme a la norma RS-232-C. Un *pin* es usado para mandar bytes de datos y otro para recibirlos. Otro *pin* es usado como señal de tierra común. Los demás *pins* útiles son usados principalmente para propósitos de señalización con un voltaje negativo constante indicando 'apagado' y un voltaje positivo constante que significa 'prendido'. El microcontrolador UART (Transmisor-Receptor Asíncrono Universal) hace la mayor parte del trabajo que consiste en recibir y preparar los datos para que puedan ser usados por el procesador (Lawyer y Hankins, 2002).

El puerto serie es manejado por interrupciones, cada vez que el puerto serie genera una interrupción, el CPU envía otro byte. El puerto serie transmite bits a una velocidad constante (*baud rate*) seleccionada por el usuario. El puerto serie convierte los datos de serie a paralelo y cambia la representación eléctrica de los datos, siendo esta una de las funciones de UART. Las velocidades usadas dependen de los diferentes tipos de señales que recibimos y se mencionan en la sección IV.3.1.

Tabla II. Conector DB9 para el puerto serial.

Conector de 9 pins a un dispositivo DTE (conexión de PC)	
Conector macho RS232 DB9	
Número de pin	Dirección de la señal:
1	Detector de portadora (CD) Señal entrante de un módem
2	Datos Recibidos (RD) Datos que entran provenientes de un DCE
3	Datos transmitidos (TD) Datos transmitidos hacia un DCE
4	Terminal de datos lista (DTR) Señal de saludo de salida
5	Señal de Tierra. Voltaje de referencia común
6	Listo para recibir datos (DSR) Señal de saludo de entrada.
7	Petición para enviar (DTS) Señal de control de flujo de salida
8	Limpiar para envío (CS) Señal de control de flujo de entrada
9	Indicador de timbre (RI) Señal que entra proveniente de un módem.

II.3.3 Tarjetas multipuerto y microcontroladores

Una de las principales características de este trabajo es la adquisición de datos por medio de puertos serie. La mayoría de las computadoras sólo cuentan con dos puertos serie sin embargo, en nuestro caso era necesario incrementar el número de puertos de la computadora a usar. El uso de una tarjeta multipuerto suministró a la computadora puertos adicionales para manejar las señales de hasta 12 estaciones sísmicas, además de la señal proveniente de un reloj digital.

Las tarjetas multipuerto hoy en día se usan comúnmente para el control de dispositivos externos. Cada tarjeta provee de varios puertos seriales dependiendo del modelo que se elija. En nuestro caso se utiliza una tarjeta inteligente, la cual maneja internamente la mayoría de las interrupciones desde los UARTs dentro de la tarjeta. Para la tarjeta seleccionada se verificó que existiera un manejador de dispositivo compatible para Linux (ver Tabla III). Esta tarjeta usa dispositivos especiales (ttyC en el directorio /dev) y no los comunes (ttyS).

Tabla III. Tarjeta multipuerto seleccionada

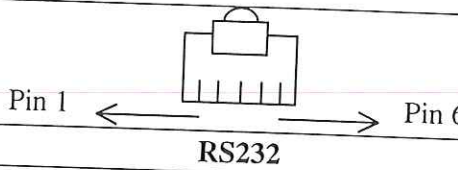
Marca	Modelo	Driver para Dispositivos Linux	UART	No. de puertos
Cyclades	Cyclom-8Ys (PCI)	Si	ttyC0-7	Cirrus Logic CD1400
				8

Las tarjetas multipuerto inteligentes usan microcontroladores con procesamiento digital de señales para control y *buffering* adicional, relevando así al CPU aún más de la tarea de manejar la entrada de datos. En el caso de la tarjeta Cyclom Cyclades seleccionada, ésta usa el microcontrolador Cirrus Logic CD1400 RISC UART. Para el caso de la tarjeta multipuerto Cyclades 8Ys, la entrada de datos se lleva a cabo usando conectores del tipo RJ12 (Tabla IV). Los conectores RJ12 y DB9 sólo usan dos *pins*, en nuestro caso, para el manejo de los datos. Estos *pins* son el 2, para entrada de datos, y el *pin* 5 como señal de tierra para el conector DB9. En el caso del conector RJ12 se usan respectivamente los *pins* 2 y 3.

En la Figura 6 se representa el camino que siguen las señales recibidas hasta llegar al proceso de detección. Las señales de todas las estaciones sísmicas (6 en

forma individual y 6 en forma multiplexada) se reciben a través de la tarjeta multipuerto y la señal de tiempo se recibe por medio del puerto serie de la computadora. Una vez que las señales se decodifican y se les incluye la marca de tiempo, las señales resultantes pasarán por un algoritmo de detección en donde se decidirá si son señales útiles para proceder a guardarlas en el disco duro de la computadora. Todos estos procesos se explican con detalle en los Capítulos III y IV del presente trabajo.

Tabla IV. Conector RJ12 para el puerto serial.

Esquema y conector RJ12 usado en la tarjeta Ciclom Cyclades 8Ys	
Conector	
Pin	
1	Terminal de datos lista (DTR)
2	Transmisión de datos (TxD)
3	Tierra (Gnd)
4	Limpia para envío (CTS)
5	Recibir datos (RxD)
6	Detector de portadora (DCD)

II.4 La biblioteca GTK+

En Linux el modo gráfico se maneja por medio del servidor para gráficas XFree86. Dentro de este ambiente gráfico existen diversas bibliotecas para programar interfaces gráficas de ventanas. Entre ellas podemos mencionar GTK+, Qt, xforms y Xlib. Para tomar una decisión acerca de cual sería la mejor opción en nuestro caso, tomamos en cuenta el hecho de que existiera una buena documentación acerca de ella, y que su curva de aprendizaje no fuera demasiado pronunciada. Siguiendo con este criterio tomamos el camino de las librerías de GTK+, por estar

nuestro caso, tomamos en cuenta el hecho de que existiera una buena documentación acerca de ella, y que su curva de aprendizaje no fuera demasiado pronunciada. Siguiendo con este criterio tomamos el camino de las librerías de GTK+, por estar desarrolladas en lenguaje C, pero con un enfoque orientado a objetos, lo cual facilitó el aprender el manejo de ella.

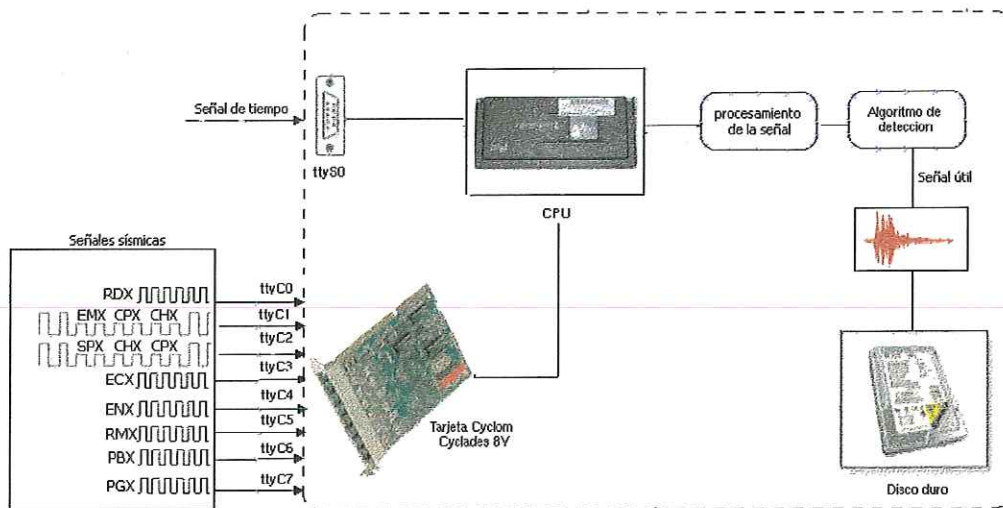


Figura 6. Esquema general del proceso de adquisición de los datos de RESNOM

GIMP ToolKit (GTK+) es una biblioteca utilizada para desarrollar aplicaciones que tengan una interfaz gráfica de usuario (GUI, Graphical User Interface). Esta biblioteca se utiliza ampliamente hoy en día para desarrollar aplicaciones GUI para Linux (Harlow, 1999). GIMP ToolKit (Graphical Image Manipulation; Manipulación de Imágenes Gráficas) fue escrito originalmente para el desarrollo de GIMP, pero GTK+ está siendo usado en un número grande de proyectos de software, incluyendo el proyecto GNOME (GNU Network Object Model Environment). GTK+ es una biblioteca orientada a objetos escrita en C que

ventanas subyacente (Xlib en el caso de X Windows). GTK+ es esencialmente una aplicación de interface de programación (API, por sus siglas en inglés) orientada a objetos. Aunque GTK+ está escrita completamente en C, está implementada usando la idea de clases y funciones *callback* (apuntadores a funciones). Existe también un tercer componente llamado Glib que contiene algunas funciones de reemplazo para llamadas estándar, así como funciones adicionales para el manejo de listas ligadas y otras estructuras de datos. Estas funciones de reemplazo son utilizadas para incrementar la portabilidad.

En este trabajo la biblioteca GTK+ se utilizó en programas para el graficado en tiempo real de las señales de periodo corto y periodo largo, además de las señales registradas en disco. Una explicación de estos programas y su puesta en funcionamiento se dará en los Capítulos III y IV.

III DESARROLLO DE UN SISTEMA DE ADQUISICIÓN DE DATOS DE PERIODO LARGO

III.1 Introducción y objetivo.

Hasta enero de 2001 funcionaba un sistema de adquisición de datos de periodo largo en una computadora PC usando el sistema operativo MS-DOS. Este sistema de adquisición presentaba el problema de un corrimiento en las marcas de tiempo en los sismogramas. Debido a este problema era necesario realizar un ajuste para poder determinar apropiadamente el tiempo de arribo de las fases sísmicas observadas. Por otro lado existía el inconveniente de que, cuando ocurría algún sismo, se tenía que interrumpir el proceso de captura, renombrar el archivo y transferirlo hacia otra computadora para su análisis posterior. Adicionalmente era necesario interrumpir el programa de captura aproximadamente cada 24 horas, guardar el archivo generado y a continuación abrir otro para que (después de volver a activar el programa de grabación) continuara con la grabación de la señal. Este proceso era muy ineficiente, además de que no se tenían rutinas para el graficado en tiempo real.

Con el propósito de subsanar los problemas mencionados, el objetivo principal de esta sección consistió en realizar el diseño de un sistema de adquisición de datos sísmicos de periodo largo. Los objetivos específicos son: diseñar un programa que incluya de manera correcta las marcas de tiempo en la señal, desarrollar herramientas para la visualización y extracción de los datos y escribir un programa para la detección de la señal de periodo largo. En la computadora se reciben simultáneamente la señal sísmica y la señal de tiempo. El programa tiene que tomar la señal que llega a la computadora a través del puerto serie, convertirla a un formato propietario (formato RESNOM), almacenarla en disco y aplicarle un algoritmo de

detección. Una vez que arriban ambas señales a la computadora, un algoritmo se encarga de incluir apropiadamente la señal de tiempo (referida al Tiempo Coordinado Universal, UTC por sus siglas en inglés) a la señal entrante y la señal resultante se graba en disco.

Con respecto a las herramientas de visualización, éstas permiten visualizar la señal en tiempo real. Los parámetros que se deben proporcionar son la escala vertical (amplitud) de la señal y la escala horizontal (segundos a graficar). El resultado será una gráfica de la señal que va arribando a la computadora. Esta herramienta es útil para revisar la señal y detectar algún problema en ella. Adicionalmente se escribieron otras herramientas que permiten hacer el cambio de formato de las señales grabadas. Uno de ellos es el cambio del formato RESNOM a código en ASCII y el segundo es de formato RESNOM a formato Seisan. Este último formato nos permite utilizar los datos con el programa Seisan de Haskov y Ottemöller (2000) para el procesamiento de la información sísmica.

III.2 Características y formato de transmisión de la señal de periodo largo

La estación de periodo largo está constituida por tres sismómetros de periodo largo (periodo natural $T_n = 15$ s) colocados en direcciones mutuamente ortogonales (Figura 7). La señal de salida de los sismómetros es proporcional a la velocidad de movimiento del terreno. Esta estación permite detectar señales de periodos largos ($2 \leq T \leq 40$ s), con amplitudes de alrededor de $\sim 10^{-5}$ cm/s. Este tipo de estaciones pueden detectar eventos sísmicos ocurridos a distancias grandes (>1000 km). Una descripción breve del acondicionamiento de las señales producidas por los

descripción breve del acondicionamiento de las señales producidas por los sismómetros se proporciona a continuación. La señal de cada sismómetro se amplifica (100 000 veces) y se filtra mediante un filtro pasabajas de 2 polos, $f_c = 0.011$ Hz. (Vidal *et al.*, 2003)

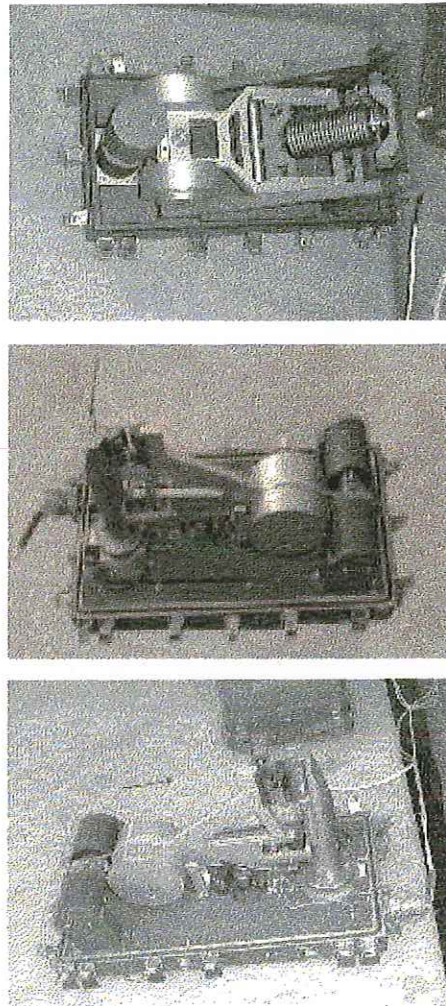


Figura 7. Sismómetros de periodo largo. En la parte superior se muestra la componente vertical (Z), en la intermedia la componente norte-sur (N-S) y en la inferior la componente este-oeste (E-W) (Figura tomada de Vidal *et al.*, 2003)

La señal filtrada se digitaliza usando un convertidor analógico/digital de 12 bits de resolución (72 db de rango dinámico) a razón de una muestra por segundo por componente. Cada muestra está representada por 12 bits que corresponden a su valor de entre 0 y 4095, además de otros cuatro bits para identificación. Dos de ellos se utilizan para identificar el canal al que pertenece la muestra (Vertical, Norte-Sur y Este-Oeste) y los otros dos bits son usados para sincronizar la señal. La Figura 8 muestra el formato de transmisión de la señal de periodo largo. La tasa de transmisión (*baud rate*) para la señal de periodo largo es de 2400 bps. En la computadora se reciben tres muestras cada segundo, esto es, una muestra por componente.

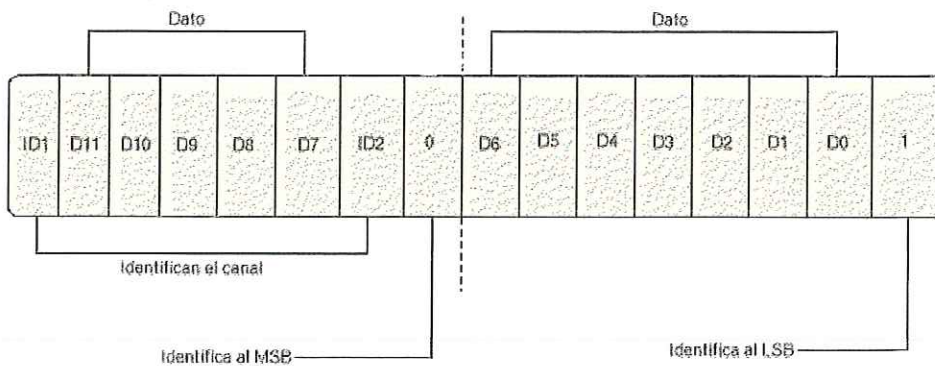


Figura 8. Diagrama del formato de la señal de periodo largo

La señal siempre llega en pares, primero el MSB (*Most Significant Byte*) y después el LSB (*Least Significant Byte*). Cada uno de estos bytes trae un identificador, el cual si está en el orden correcto indica que corresponde a una muestra. A continuación se procede a analizarla y cambiarla al formato RESNOM. El primer y el séptimo bit se utilizan para identificar a qué canal corresponde la

muestra (Hinojosa, 1988). El código que identifica el canal con la componente está definido en la Tabla V.

Tabla V. Código de identificación de la señal de periodo largo.

<i>ID1</i>	<i>ID2</i>	<i>Canal</i>	<i>Componente</i>
0	0	1	Vertical
0	1	2	Este-Oeste
1	0	3	Norte-Sur

Una vez identificado el canal al que pertenece la muestra, se toman los doce bits restantes y se unen para obtener su valor correspondiente (ver Apéndice C). Hecho lo anterior se procede a guardar la información en el formato mostrado en la Figura 9:

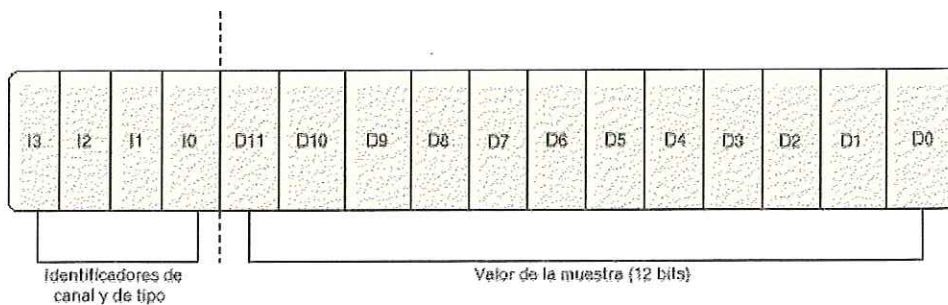


Figura 9. Diagrama del formato RESNOM

III.3 Señal de tiempo

Además de la señal sísmica, se recibe la señal de tiempo proporcionada por un reloj KINEMETRIX 468-DC a través de uno de los puertos serie de la computadora. La información de este reloj es utilizada para incluir el tiempo a la señal. El reloj está sincronizado con la señal UTC enviada por uno de los dos satélites de Observación

caracter extra que indica la calidad de la información. Cada caracter extra corresponde a un intervalo de tiempo de diferencia entre la cadena recibida y el tiempo UTC (Tabla VI).

Tabla VI. Código de calidad de la señal de tiempo del reloj (Hinojosa, 1988)

<i>Código (3 bits)</i>	<i>Símbolo ASCII</i>	<i>Error</i>
0 0 0	Espacio	< 1 milisegundo
0 0 1	.	- 1 < milisegundo < 1
0 1 0	*	- 5 < milisegundos < 5
0 1 1	#	-50 < milisegundos < 50
1 0 0	?	-500 < milisegundos < 500
1 0 1	No definido	Error en la cadena

La cadena de caracteres que envía el reloj tiene el siguiente formato:

Ctrl-a *D* *D* *D* : *H* *H* : *M* *M* : *S* *S* *Q* *CR* *LF*
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Donde :

Ctrl-a Nos indica el principio de información.

DDD Día juliano

HH Hora del día

MM Minutos

SS Segundos

Q Calidad de la hora proporcionada

CR Retorno de carro

LF Avance de línea

En el formato RESNOM, si el primer bit de una muestra está encendido (I3 de la Figura 9), éste indica que es una marca de tiempo. El formato de cada marca de

En el formato RESNOM, si el primer bit de una muestra está encendido (I3 de la Figura 9), éste indica que es una marca de tiempo. El formato de cada marca de tiempo ocupa 4 bytes (dos enteros cortos) y está codificada en octal (Figura 10).

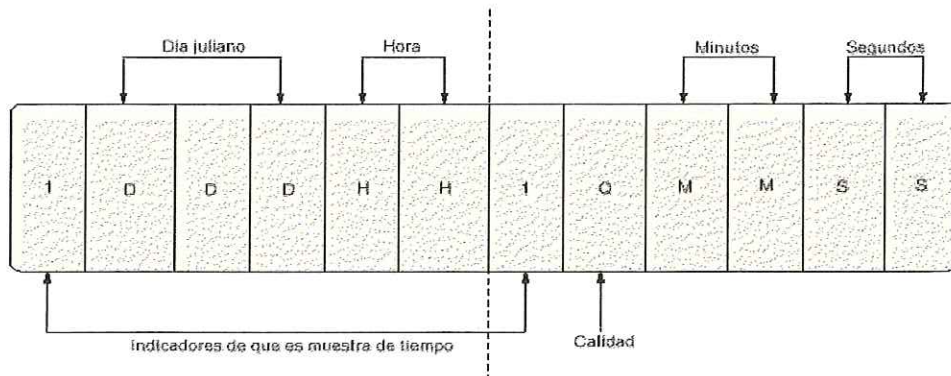


Figura 10. Diagrama del formato de las marcas de tiempo

El tiempo se incluye cada 60 segundos a la señal sísmica, es decir que por cada 180 muestras existe una marca de tiempo en formato binario (ver Apéndice D). La tasa de transmisión para la señal de reloj es de 1200 bps. Las características del formato mostradas anteriormente permiten incluir sólo hasta el día juliano, no obstante también es importante saber el año en que se grabó el registro. Este valor se toma del sistema operativo y forma parte del nombre del archivo.

III.4 Plataforma seleccionada y software desarrollado

El sistema de adquisición de la señal de periodo largo, como se mencionó anteriormente, se desarrolló en una computadora PC con sistema operativo Linux. Las características de la computadora usada se describen en la Tabla VII. Al trabajar

Tabla VII. Características de la computadora en donde se implantó el sistema de recepción de periodo largo

<i>Procesador</i>	Intel Pentium III 1000Mhz
<i>RAM</i>	128 MB
<i>Video</i>	Matrox G450 dual head AGP 32 MB
<i>Ethernet</i>	Realtek 8139C
<i>Disco duro</i>	Maxtor 120 GB
<i>CD ROM</i>	Creative CD-RW 12-10-32x

La computadora usada para el sistema de adquisición de la señal de periodo largo también se utiliza para hacer el análisis de la información sísmica grabada de periodo corto, periodo largo y banda ancha; con ello se hace un uso apropiado de las características de la computadora. Por otro lado, se aprovecha de la ventaja que ofrece Linux de ser multitareas. Además, con base en el soporte de este sistema operativo, se desarrollaron los programas en lenguaje C para realizar la adquisición y el graficado de las señales de periodo largo. Para generar los ejecutables de los programas escritos, se usó el compilador para lenguaje C (gcc ver. 2.95.3) incluido en la versión 8.0 de Slackware Linux. Los programas desarrollados para la adquisición (dos módulos) se encargan de recibir los datos, incluir las marcas de tiempo y guardar la señal resultante.

El primer módulo se encarga de recibir la señal entrante, decodificarla (la cambia de formato de transmisión a formato RESNOM) y guardarla en disco (Figura 11). Cada doce horas este módulo genera un archivo con la información recibida. El módulo recibe la información del puerto serial, revisa que sea correcto el orden de llegada de los bytes, toma muestra por muestra, cambia su formato y la guarda en un archivo. El nombre de este archivo está formado por 18 caracteres: Plargoaaaammddhhmm donde aaaa representa el año, mm el mes, dd el día, hh la

llegada de los bytes, toma muestra por muestra, cambia su formato y la guarda en un archivo. El nombre de este archivo está formado por 18 caracteres: Plargoaaaammddhhmm donde aaaa representa el año, mm el mes, dd el día, hh la hora, y mm el minuto del tiempo de grabación. El nombre del archivo se guarda en otro archivo denominado Nombre, que sirve de referencia para que el segundo módulo determine donde va a incluir las marcas de tiempo.

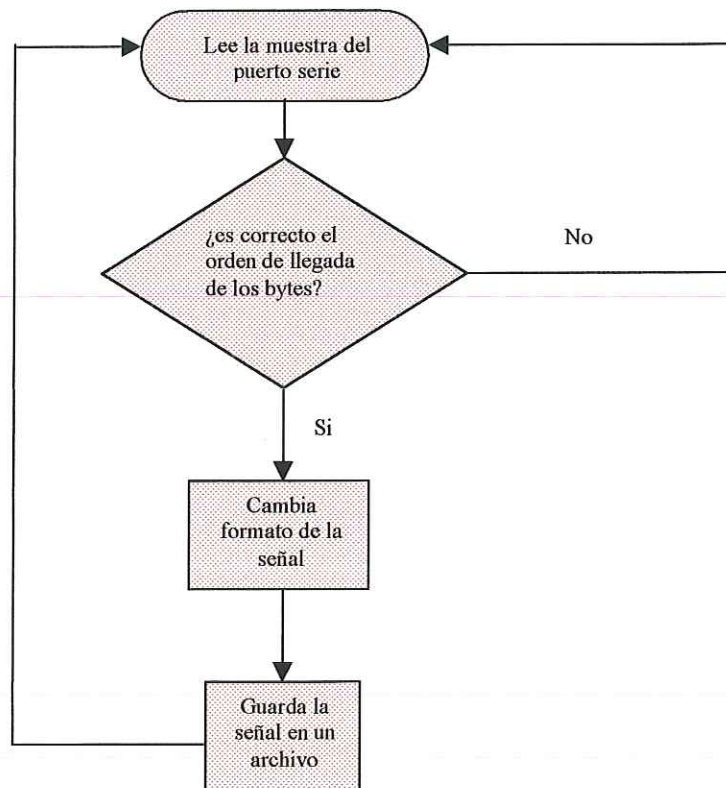


Figura 11. Diagrama del módulo que realiza la captura de la señal

El segundo módulo abre el archivo Nombre en donde se encuentra la referencia al nombre del archivo en el que va a incluir las marcas de tiempo. A continuación toma una cadena de caracteres que le envía el reloj y la guarda dentro

de un arreglo en memoria. Esta cadena se cambia primero de tipo caracter a entero y luego al formato de las marcas de tiempo (Figura 10) para incluirlas en el archivo de señal cada 60 segundos. El proceso descrito se representa en el esquema de la Figura 12. Una vez realizada la adquisición de las señales, éstas se almacenan en archivos que contienen 12 horas de señal continua. Una vez cumplido este periodo, el sistema automáticamente cierra el archivo y crea uno nuevo en donde se repite el proceso descrito anteriormente (escritura de la señal e inclusión de las marcas de tiempo).

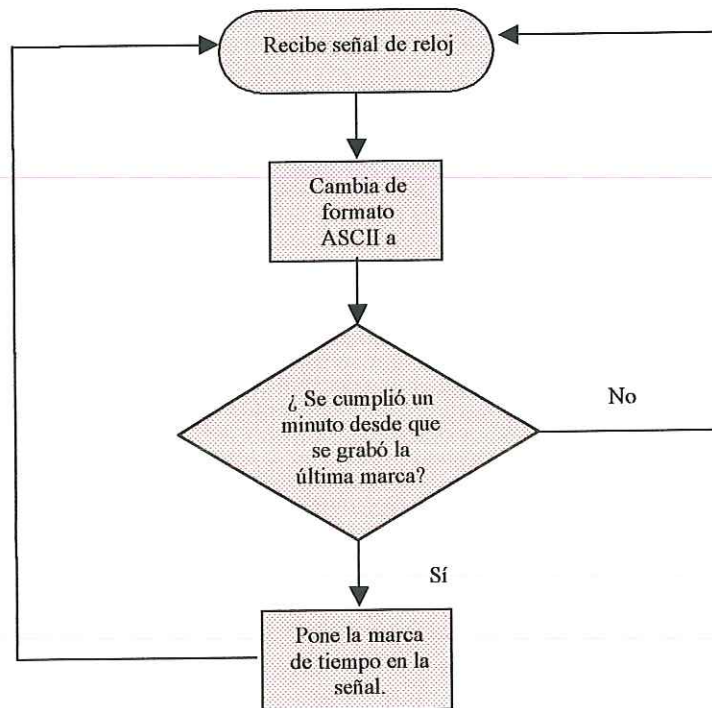


Figura 12. Diagrama del método utilizado para insertar las marcas de tiempo a la señal.

III.5 Graficado de la señal en tiempo real.

Para visualizar la información que arriba a la computadora en tiempo real se

desarrolló un programa escrito en lenguaje C. El programa utiliza las librerías para gráficos *GTK+ versión 1.2.10*, además de las librerías *extra GTK+ v0.99.14* para la creación y el manejo de la gráfica de las señales. Este programa despliega una interfaz gráfica de ventanas (Figura 13) que nos pide dos parámetros para la gráfica: la escala vertical (en cuentas) y el tiempo (en segundos). Si no introducimos ningún valor para ambos parámetros, toma 2048 para la escala vertical y 1200 para la escala horizontal como valores por omisión.

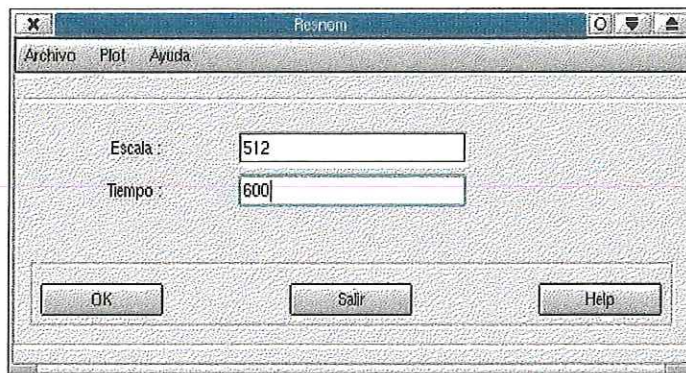


Figura 13. Ventana de diálogo para suministrar los parámetros de graficado.

Una vez que se introducen los valores de los parámetros solicitados por el programa, éste da como resultado una gráfica en donde se pueden ver las tres componentes (Z, NS, EW; [Vertical, Norte-Sur, Este-Oeste]) de la señal que arriba a la computadora en tiempo real. (Figura 14).

La ventaja del uso del programa para graficar la señal en tiempo real es que permite realizar una inspección visual de la señal recibida para saber si se está recibiendo apropiadamente.

recibiendo apropiadamente.

III.6 Programas adicionales.

Además de los programas mencionados, se escribieron otros programas adicionales con el propósito de manejar la señal registrada. Estos programas se describen a continuación.

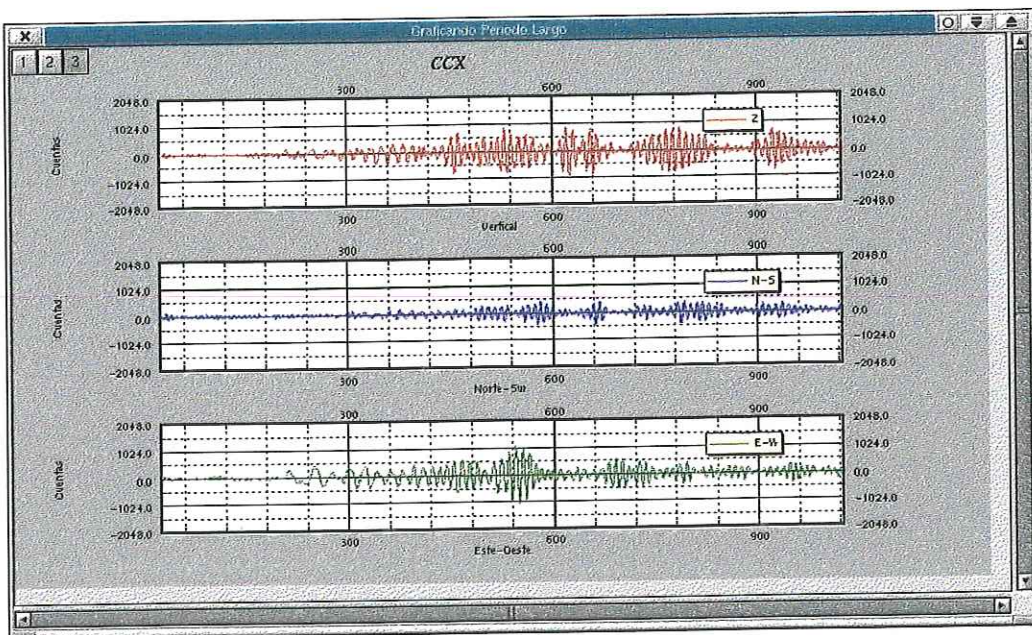


Figura 14. Ventana que ilustra la gráfica de la señal de periodo largo en tiempo real.

El programa **grafica** permite visualizar la señal contenida en los archivos grabados (Figura 15), con este programa podemos variar las escalas vertical (amplitud) y horizontal (tiempo) y hacer una revisión detallada de la señal. Este programa se hizo con el fin de hacer una inspección de la señal almacenada en los archivos. El programa **extrae** permite extraer una determinada cantidad de señal del

total guardada en los archivos. El programa **detecta** aplica un algoritmo de detección a la señal guardada en los archivos. Este algoritmo está basado en el cociente de promedios cortos (STA) y promedios largos (LTA) de la señal sísmica. Cuando este cociente rebasa un umbral determinado, se procede a declarar un evento. Una descripción con detalle acerca de este algoritmo se proporcionará en el Capítulo IV. El programa **plargoaseisan** convierte los datos del formato RESNOM al formato que maneja el programa Seisan (Haskov y Ottemöller, 2000). Este programa es el que se utiliza para hacer el análisis de la información sísmica. (por ejemplo lecturas de tiempos de arribo, localización hipocentral y cálculo de magnitudes)

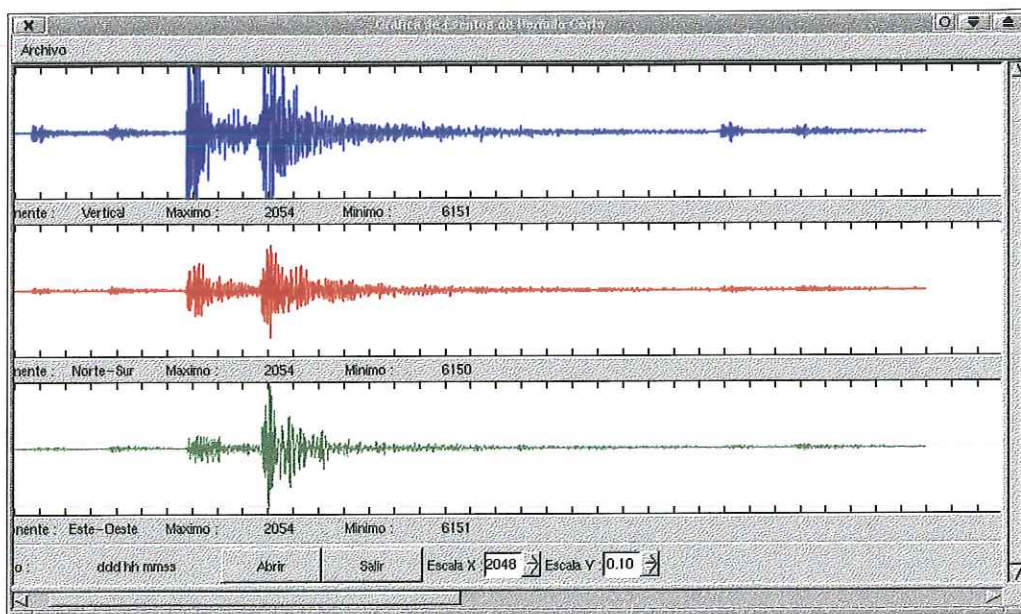


Figura 15. Ventana del programa Grafica

III.7 Resultados

Una vez que se puso en operación el sistema, éste ha registrado actividad sísmica continua obteniéndose registros digitales de sismos ocurridos desde agosto

con la de banda ancha y la de periodo corto.

A continuación se presentan los resultados obtenidos con el uso del programa **detecta** en el cual está programado el algoritmo de detección que se aplica a la señal de periodo largo registrada. Para ello, se eligieron 3 registros de sismos localizados a diferentes distancias epicentro-estación (epicentrales). Para determinar los valores apropiados para nuestro sistema de detección, variamos los parámetros STA y LTA entre 5 y 50 segundos y 150 y 300 segundos, respectivamente. Los parámetros con los que se obtuvieron resultados satisfactorios son los mostrados en la Tabla VIII.

Tabla VIII. Parámetros del algoritmo de detección aplicado a la señal de periodo largo registrada

Promedios cortos (STA)	Promedios largos (LTA)	Umbral
50	300	5

Como se mencionó anteriormente, la selección de los sismos se hizo en función de la distancia epicentral. Para ello se seleccionaron: un sismo local registrado a 166 km, un sismo registrado a 2292 km y un telesismo, registrado a 4275 km. En la Tabla IX se proporciona información de las localizaciones epicentrales de los sismos seleccionados.

La Figura 16 muestra la ubicación de cada uno de estos sismos, así como la ubicación de la estación de periodo largo (CCX). En la Figura 17 presentamos un conjunto de tres gráficas por cada evento. En cada conjunto mostramos en las

Tabla IX. Coordenadas hipocentrales y magnitud de los sismos seleccionados para probar el algoritmo de detección.

Fecha ddmmaaaa	Origen hh:mm:ss	Latitud °N	Longitud °W	Prof. (km)	Mag.	Región
08/12/2001	23:30:00	32.04	-114.91	10	5.8	Norte de Baja California
18/04/2002	05:05:00	16.99	-100.86	24	6.3	Costa de Guerrero
25/05/2002	05:43:00	53.81	-161.12	33	6.5	Península de Alaska

gráficas superiores la señal original de cada sismo. Esta señal está contenida en los archivos grabados rutinariamente cada 12 horas. El propósito del algoritmo de detección es el de declarar un evento sísmico para extraerlo de los archivos originales. En las gráficas intermedias se muestra la señal filtrada, y en las gráficas inferiores los resultados obtenidos de aplicar el algoritmo de detección a las señales grabadas con los parámetros indicados en la Tabla VIII.

Con base en los resultados mostrados en la Figura 17 se demuestra que el algoritmo cumple con su función de hacer la detección de sismos. Este algoritmo también se aplica para la señal de periodo corto, cuyas características se proporcionaran en el siguiente capítulo.

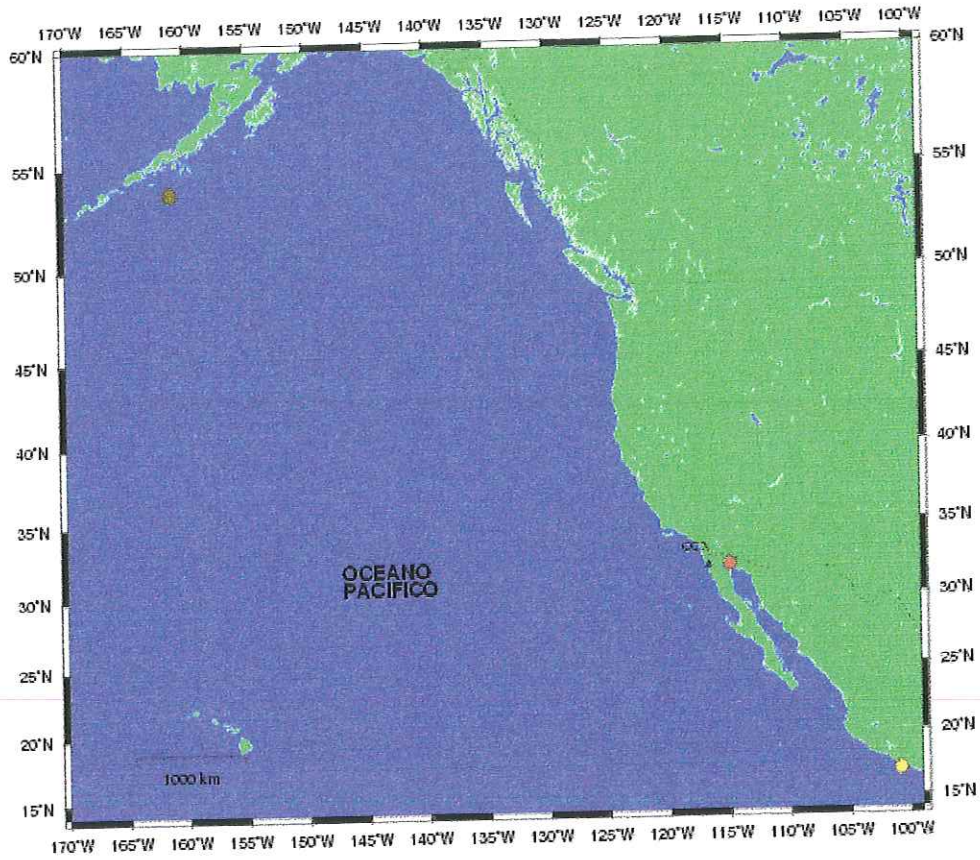
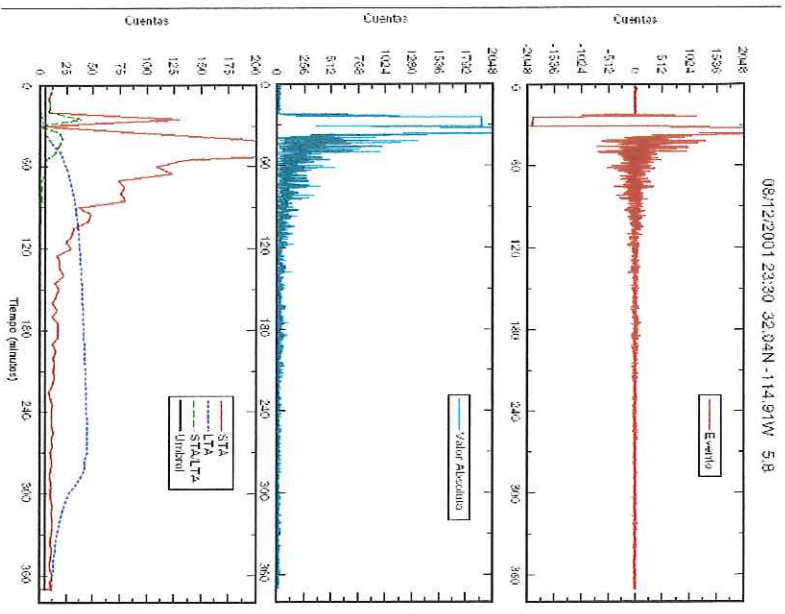
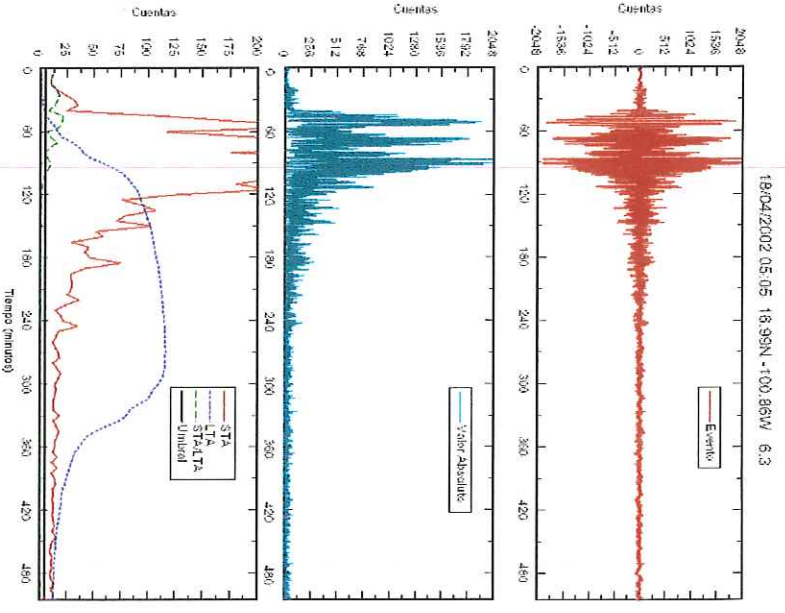


Figura 16. Localización de los sismos utilizados para probar el algoritmo de detección.

Sismo del Norte de B.C.



Sismo de la Costa de Guerrero



Sismo de la Península de Alaska

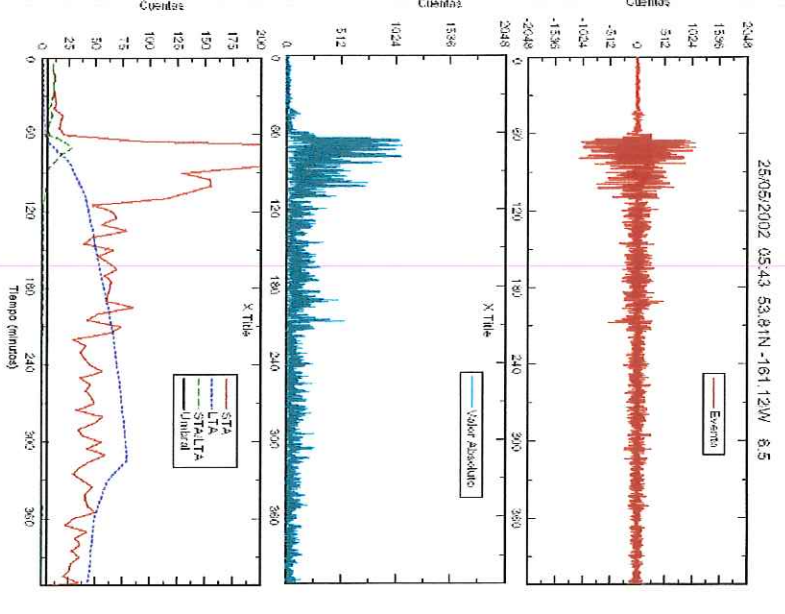


Figura 17. Resultados obtenidos al aplicar el algoritmo de detección a una señal de periodo largo

IV DESARROLLO DE UN SISTEMA DE ADQUISICIÓN DE DATOS DE PERIODO CORTO

IV.1 Introducción

En este capítulo se presenta el proceso de diseño y la implantación de un sistema nuevo de adquisición de señales sísmicas de periodo corto ($T_n = 1s$). Actualmente (Agosto 2003) se encuentra en funcionamiento un sistema de adquisición de este tipo de señales desarrollado por Hinojosa (1988). Como se mencionó en la sección I.3 el sistema de adquisición desarrollado por Hinojosa funciona sobre el sistema operativo OS-9, sin embargo a pesar de su buen funcionamiento, éste no se actualizó y en consecuencia tampoco el sistema de adquisición. Debido a la falta de actualización del sistema de adquisición con el tiempo se presentaron los problemas descritos en la sección I.3 y cuya solución motivó el presente trabajo.

El objetivo de esta parte de la tesis fue el de migrar el sistema de adquisición de señales de periodo corto basado en el sistema PC/OS-9 hacia una computadora PC con Linux. Dentro de este proceso, un punto importante fue el implementar un algoritmo para la detección de señales sísmicas en tiempo real. También fue necesario hacer programas para la visualización y el procesamiento preliminar de las señales sísmicas (depuración, cambio de formato, entre otros).

IV.2 Instrumentación de periodo corto y formas de onda obtenidas.

Como se mencionó en la sección I.1 RESNOM está constituida en parte por doce estaciones sísmicas de periodo corto ($T_n = 1s$) distribuidas en la región norte de Baja California. Cada estación consiste de tres sismómetros que registran dos

Baja California. Cada estación consiste de tres sismómetros que registran dos componentes mutuamente ortogonales de movimiento horizontal y una componente de movimiento vertical, equipo electrónico para el acondicionamiento de las señales y un radiotransmisor para el envío de las señales al centro de procesamiento (Figura 18).

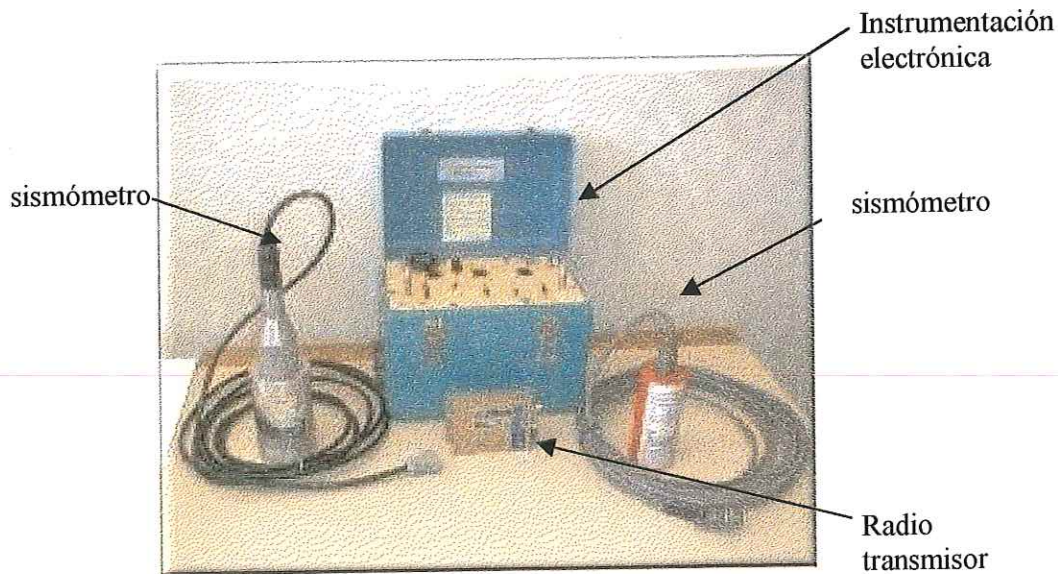


Figura 18. Instrumentación típica de una estación de periodo corto.

En lo que respecta a los sismogramas obtenidos con el tipo de instrumentación mostrada en la Figura 18, éstos corresponden de acuerdo al tipo de sismo registrado. Los sismos se clasifican de acuerdo a la distancia epicentral (Δ). La mayoría de los sismos registrados por RESNOM corresponden a eventos locales ($\Delta < 300$ km.) aunque también se tienen registros de eventos regionales ($300 < \Delta \leq 1000$ km), telesismos ($\Delta > 1000$ km) y explosiones nucleares. La Figura 19 nos muestra los diferentes tipos de formas de onda generadas por el tipo de sismos mencionados.

IV.3 Arribo de señales en formato RS232C

Para obtener señales como las mostradas en la Figura 19, se tiene que hacer un procesamiento previo, descrito a continuación. En el centro de procesamiento las señales arriban al puerto serie de la computadora de dos formas: señal proveniente de una estación, o señal proveniente de tres estaciones (señal multiplexada). En todos los casos la señal de cada estación viene multicanalizada es decir la señal que arriba contiene información de tres canales (Z, NS, EO). En el caso de la señal multiplexada se tiene información de nueve canales (3 por cada estación).

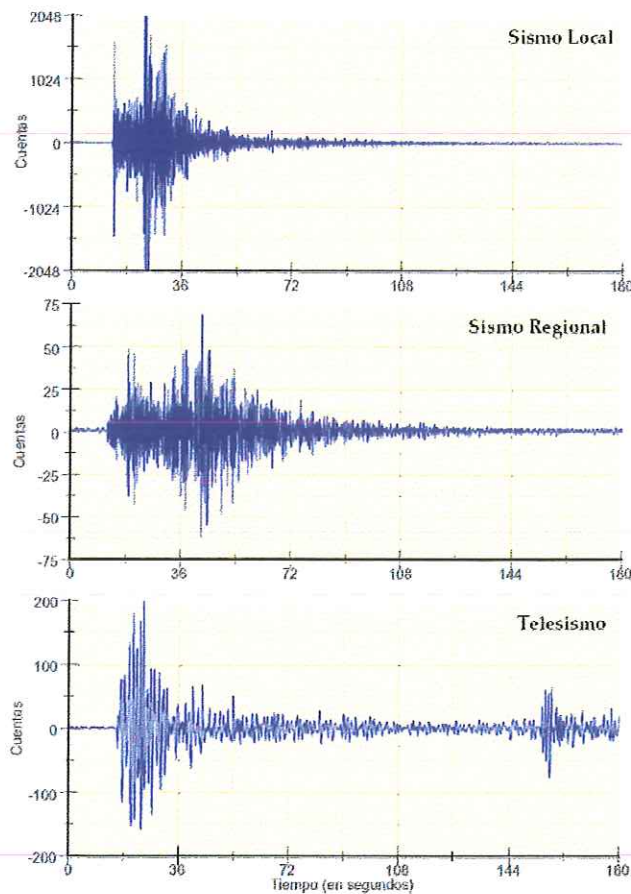


Figura 19. Diferentes formas de onda registradas con la instrumentación de periodo corto de RESNOM

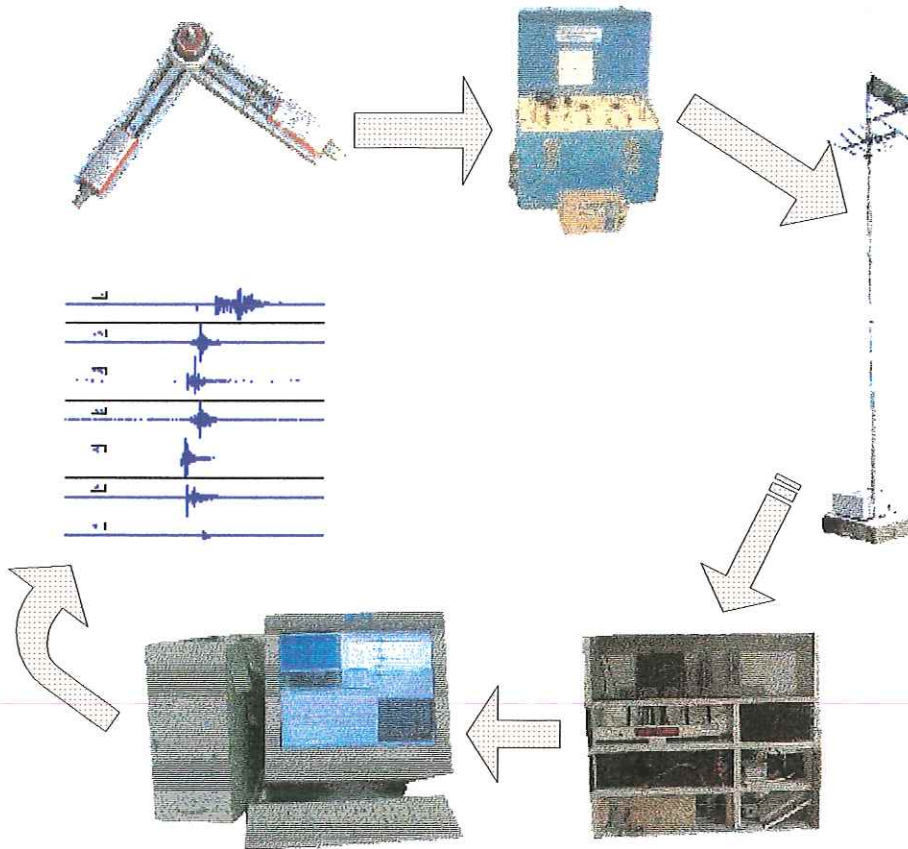


Figura 20. Ilustración de la ruta que siguen los datos desde que son adquiridos hasta su arribo y procesamiento en la computadora.

IV 3.1 Señal proveniente de una estación.

El formato de la señal de periodo corto es similar al descrito para la señal de periodo largo. La diferencia consiste en que ahora tenemos 40 muestras por segundo, por cada canal (120 muestras en total) en vez de una muestra por segundo por canal para el periodo largo (3 muestras en total). La Figura 8 (sección III.2) muestra el formato de transmisión de la señal de periodo corto. Para poder capturar las 120 muestras por segundo hay que programar el puerto serie para que reciba las muestras

a una velocidad de 2400 bps. Una vez que se programó el puerto se procede a la decodificación de la señal, proceso que se describe en la sección IV.4.5.

IV 3.2 Señal proveniente de tres estaciones.

El segundo caso que se presenta es cuando la señal llega multiplexada (señal de 3 estaciones) a un sólo puerto serial. En este caso se programa el puerto para recibir datos a una velocidad de 9600 bps. Como ya sabemos, la señal llega en pares, primero el LSB y después el MSB. En el caso de la señal multiplexada se utilizan dos bytes marcados con unos para señalar el inicio de un paquete de datos. Por medio de estos bytes de sincronía sabemos que los siguientes seis bytes serán de las muestras de cada una de las estaciones (Figura 20). El proceso para la decodificación de la señal multiplexada se describe a continuación

IV.4 Decodificación de la señal.

Con la experiencia adquirida en el desarrollo del sistema de adquisición de datos de periodo largo, se procedió a desarrollar los algoritmos para hacer la decodificación de las señales de periodo corto. Para decodificar la señal que proviene

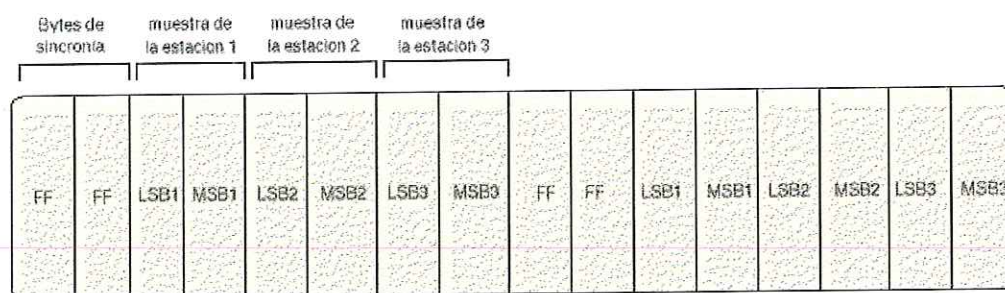


Fig. 21. Diagrama de la señal multiplexada.

señal, es decir, asegurarnos que siempre estemos recibiendo el LSB primero y el MSB después. El siguiente paso es identificar el canal al que pertenece la muestra, y finalmente tomar el valor de la muestra. Una descripción detallada de este proceso se encuentra en el Apéndice C.

El proceso para hacer la decodificación de la señal multiplexada (tres estaciones) es parecido, la diferencia consiste en que primero debemos establecer la sincronía de la señal: hay que esperar primero por los dos bytes de sincronización, posteriormente etiquetar las muestras para saber a que estación pertenece cada una y finalmente realizar el proceso descrito para el caso de una estación.

El resultado de los pasos anteriores se guarda en un entero corto sin signo. Esto significa que se trabaja con 16 bits de los cuales se usan dos para identificación de canal, otro para saber cual es el byte más significativo y otro para saber cual es el menos significativo, con lo que se tiene una resolución de muestra de 12 bits (para un valor máximo de hasta 4095 cuentas) y se usan 4 bits para guardar la información acerca del canal, o para indicarnos si la muestra pertenece a una marca de tiempo. Cada segundo hay que insertar una marca de tiempo. La señal de tiempo llega a la computadora a través de otro puerto serie. La señal de tiempo es una cadena de caracteres con información acerca de la fecha, hora, minuto y segundo (sección III.3). Una vez que se tiene esta cadena, se le aplica el procedimiento descrito en el Apéndice D para ponerla en un formato binario y así poder incluirla, cada segundo, en la información sísmica.

IV.5 Desarrollo del sistema de recepción y detección.

IV.5.1 Procesos usados para la adquisición.

Para hacer la adquisición de las señales provenientes de cada estación existe un módulo o proceso que realiza esta tarea para cada uno de los puertos seriales que las reciben. Estos procesos deben guardar las señales en algún lugar común para que puedan estar accesibles para otros módulos, los cuales se encargan de su procesamiento posterior. Este lugar es un segmento de memoria compartida del sistema, de esta manera es posible que varios procesos accedan a la información de manera simultánea. En esta sección se dan detalles acerca de la implementación de los diferentes módulos hechos para el sistema de adquisición.

IV.5.2 Módulos para manejo de memoria compartida

Las librerías de C en Linux (Loosemore *et al.*, 1999) permiten el manejo de memoria compartida. Con este objetivo se diseñaron varios módulos para hacer la gestión de la memoria y hacer referencia a ella por medio de un apuntador. Para simplificar el proceso de acceso a la memoria, éste se dividió en cuatro módulos que contienen las funciones necesarias para su gestión. Los módulos son: *memoria.c*, *crea_memoria.c*, *datos_memoria.c*, *libera.c*. En el primer módulo existen varias funciones que permiten el manejo del segmento de memoria compartida como son: la creación del segmento, la conexión con él, y otra función para borrarlo. El segundo módulo hace una llamada a la función que crea el segmento de memoria. El tercer módulo se encarga de poner cada muestra (de la señal decodificada) en la cola circular que le corresponde a cada muestra. El último módulo tiene la función de liberar el segmento de memoria compartida en caso de que se decida interrumpir el

proceso de adquisición.

IV.5.3 Módulos de adquisición de datos sísmicos.

Una vez que está disponible el segmento de memoria, se procede a ejecutar los módulos para la adquisición de los datos. Existen dos tipos de módulos, uno para hacer la adquisición de la señal de una estación y otro para manejar señales de hasta tres estaciones. En la Figura 21 se explica de una manera gráfica el proceso de adquisición de datos.

Los módulos para la adquisición de datos reciben de manera continua la señal de las estaciones, a través de los puertos seriales de la computadora. Estos módulos funcionan de la siguiente manera: primero abren el descriptor de archivo que controla el puerto serie (en linux todos los dispositivos son tratados como archivos), revisa que éstos (los puertos serie) existan y los configura con las características de la señal que se recibe (*baudrate* = 2400 bps, control de flujo, máscara de tamaño de carácter [CS8]) y se habilita sólo para recepción. También se le indica que se van a tomar muestras de 2 caracteres de longitud. Una vez configurado el puerto, los módulos se conectan con el segmento de memoria compartida. Para el caso de la recepción de señales de tres estaciones, se cambia la configuración del *baudrate* = 9600 bps, y se le indica que vamos a tomar muestras de un caracter de longitud.

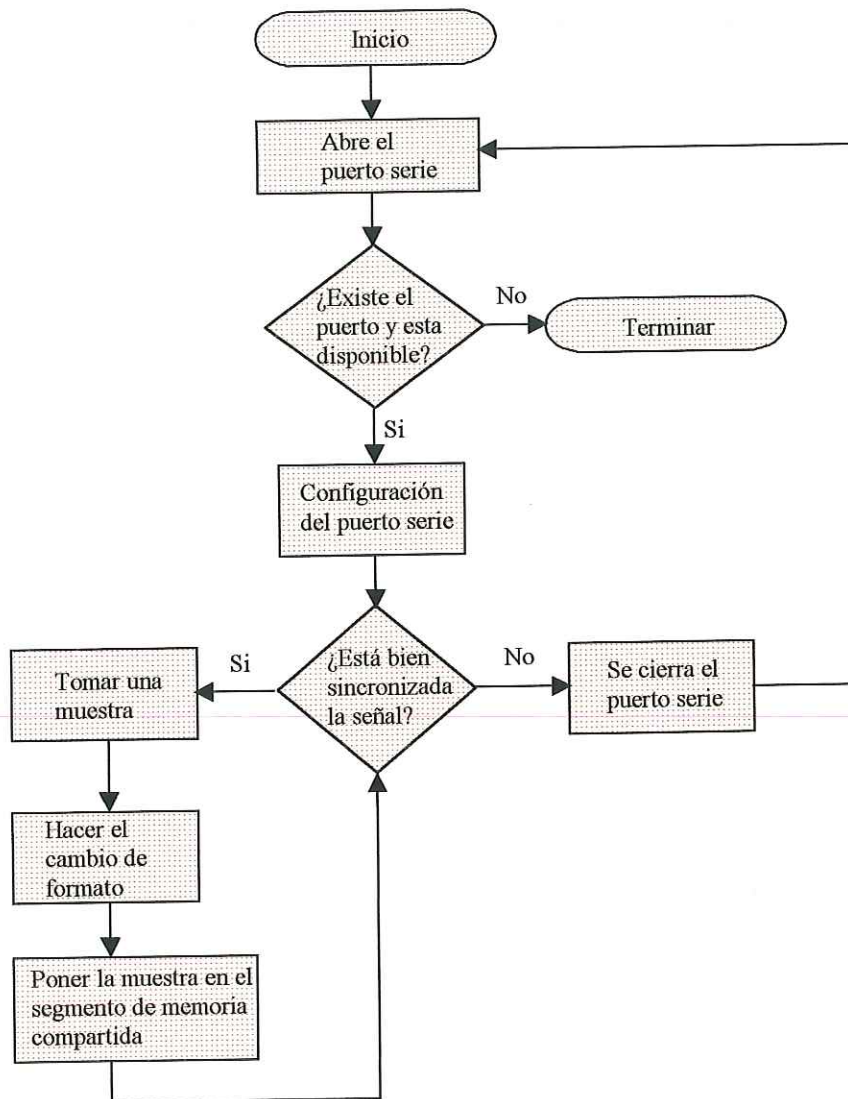


Figura 22. Diagrama del proceso de adquisición de las señales.

Una vez hecho lo anterior se inicia la recepción de los datos verificando primero que el puerto esté bien sincronizado con la señal. Para poder verificar la sincronización hay que revisar que siempre esté en la muestra primero el LSB y después el MSB. En caso de que la señal no esté sincronizada, se cierra el puerto serie y se vuelve a abrir hasta que se pueda sincronizar con la señal. Una vez

formato de la señal (ver Apéndice C) y se manda al segmento de memoria. En el caso de la recepción de tres señales, primero hay que armar la muestra a partir de los dos caracteres que llegan al puerto, etiquetar la muestra con el número de estación a la que pertenece (Apéndice B), y finalmente cambiarla de formato.

IV.5.4 Módulo para la adquisición de la señal de tiempo.

Para el proceso de la adquisición de la señal de tiempo, existe otro módulo encargado de realizar esta tarea. Este módulo recibe como entrada una cadena de 16 caracteres ASCII (ver sección III.3) por el puerto serial. La configuración del puerto serial tendrá ahora las siguientes características: *baudrate* = 1200 bps y la máscara para el tamaño de carácter será de 7 bits. Esta máscara tiene el propósito de revisar la paridad en la entrada de datos. También se programa el puerto para que reciba grupos de 16 caracteres de longitud cada vez. Al igual que con la señal entrante, en este proceso también se revisa que esté sincronizado el puerto con la cadena de entrada. En caso de que no lo esté, se cierra el puerto y se abre de nuevo hasta lograr la sincronización. Una vez sincronizado el puerto se procede a darle el formato apropiado a la señal de tiempo (Apéndice D) para incluirla a todas las señales que están arribando.

IV.6 Estructuras de datos

El proceso de adquisición de los datos es aplicado a cada muestra que llega. Una vez que éste se completa, toda la información se mantiene en una estructura de datos, en una cola circular por cada estación. Estas colas se van sobrescribiendo cada vez que se llenan, es decir, que solo se maneja el índice trasero de la cola. Cuando detectamos algún evento entonces si tomamos en cuenta el índice frontal de la cola,

para evitar que se sobrescriba información que pudiera ser útil.

La estructura de datos esta compuesta de la siguiente manera:

```
typedef struct modulo{
    int hora;
    int minuto;
    int segundo;
    int dia;
    int anio;
    int mes;
    int umbral;
    int muestras;
    float secs;
    int estado;
    unsigned short atras_resultados;
    unsigned short bloqueador_det;
    unsigned short bloqueador_esc;
    unsigned short funcion_deteccion[ESTACIONES];
    int promedios_largos[ESTACIONES];
    int frente_datos[ESTACIONES];
    unsigned short atras_pcortos[ESTACIONES];
    unsigned short cola_resultados[ESTACIONES];
    unsigned short indice_datos[ESTACIONES];
    unsigned short pcortos[ESTACIONES][MAXCORTOS];
    unsigned short buffer[ESTACIONES][MAXQUEUE];
};
```

donde los primeros 7 campos de esta estructura guardan la fecha. El campo *umbral* es el valor que al ser rebasado por el cociente *STA/LTA* hace que algoritmo de detección declare un evento. El campo *muestras* guarda el valor del número de muestras que conforman un promedio corto (STA). El campo *estado* es una variable booleana que indica el estado actual del sistema. Si es verdadero se empieza a grabar datos y si es falso el sistema se encuentra analizando las señales. El campo *atras_resultados* es el apuntador trasero de la cola de resultados de la aplicación del algoritmo de detección. El campo *bloqueador_det* es un arreglo de bits que indica las estaciones que se van a bloquear para la detección. El campo *bloqueador_esc* es otro arreglo de bits usado en este caso para bloquear estaciones para el proceso de escritura. La variable *funcion_deteccion* es un arreglo en donde se guardan los

términos de la función de detección (ver Ecuación 8). El campo *promedios_largos* guarda el último valor de los promedios largos por estación. La variable *frente_datos* es un arreglo que guarda los apuntadores frontales de todas las colas de datos. El campo *atras_pcortos* es un arreglo que contiene los índices traseros de las colas de promedios cortos. El arreglo *cola_resultados* sirve para guardar los últimos resultados de la aplicación del algoritmo de detección. El apuntador *indice_datos* indica el frente de las colas de datos que contienen las muestras que están arribando. El arreglo bidimensional *Pcortos* contiene una lista de promedios cortos de cada estación habilitada para detección. Por último *buffer* es otro arreglo bidimensional en donde se almacenan las muestras que arriban a los puertos seriales en colas circulares.

IV.7 Detección automática de sismos

Para la adquisición de datos digitales, se volvió un problema técnico el poder guardar en archivos de manera continua las señales sísmicas. Una red sísmica, o incluso sólo una estación, operando de manera continua con una frecuencia alta de muestreo, produce una enorme cantidad de datos difíciles de guardar y analizar. Por ejemplo en nuestro caso la cantidad de información que producen las 12 estaciones de periodo corto de RESNOM es muy grande (alrededor de 240 MB de datos por día). Para un disco con capacidad de 15 GB se tendría espacio para guardar menos de una semana de datos. Por tanto fue necesario tener un algoritmo de detección para guardar sólo la señal sísmica. Una vez que el algoritmo detectó un evento sísmico, se graba la señal de todas las estaciones que están arribando al centro de procesamiento. La grabación se detiene una vez de que el algoritmo 'declara' el final de la señal sísmica.

En este trabajo se usó el algoritmo de "promedios cortos contra promedios largos" (Short Time Average / Long Time Average, STA/LTA) el cual es ampliamente utilizado en sismología. La forma en como se definen los promedios cortos y largos se presenta a continuación. Sea $A(\tau)$ la amplitud de la señal que arriba a la computadora y τ el tiempo, el promedio corto $\alpha(\tau)$ está definido por (Lee y Stewart, 1981):

$$\alpha(\tau) = (\tau_1)^{-1} \int_{t-\tau_1}^t |A(\tau)| d\tau, \quad (1)$$

donde τ_1 es la longitud de la ventana de tiempo para el promedio corto (típicamente un segundo o menos). Esto permite una respuesta rápida a cambios en la amplitud de la señal que caracteriza a un sismo. Similarmente, el promedio largo $\beta(\tau)$ puede definirse como:

$$\beta(\tau) = (\tau_2)^{-1} \int_{t-\tau_2}^t |A(\tau)| d\tau, \quad (2)$$

donde τ_2 es la longitud de la ventana de tiempo para el promedio largo, el cual puede variar desde unos segundos hasta unos minutos. Debido a que τ_2 es usualmente mucho mayor que el periodo predominante de la señal entrante, $\beta(\tau)$ representa el promedio largo del nivel de la señal detectada por el sismómetro (Lee y Stewart, 1981). El siguiente paso consiste en efectuar el cociente del promedio corto entre el promedio largo ($\gamma[\tau]$):

$$\gamma(\tau) = \alpha(\tau) / \beta(\tau). \quad (3)$$

Se declara un evento cuando:

$$\gamma(\tau) > \gamma_d, \quad (4)$$

donde γ_d es el nivel de umbral seleccionado para la detección de eventos. Si este nivel es muy bajo y la señal es ruidosa, ocurrirán detecciones falsas. Si el nivel de umbral es muy alto, el número de detecciones falsas será menor pero también lo será el número de sismos detectados.

IV.7.1 Aplicación del algoritmo de detección.

A las señales de las estaciones sísmicas que arriban al centro de procesamiento, después de pasar por las etapas de demodulación y decodificación, se les incorpora una marca de tiempo cada segundo. Una vez finalizado este proceso, las señales de todas las estaciones pasan a una área de almacenamiento temporal en donde son accedidas para aplicarles el algoritmo de detección. Para hacer esto se tiene que implementar un sistema de colas circulares (First In, First Out, FIFO), una por cada estación. En estas estructuras de datos es donde se hacen las operaciones para calcular los promedios cortos y largos.

El algoritmo descrito por Lee y Stewart (1981) se modificó para poder llevar a cabo su implementación (Hinojosa, 1988). Una de las cosas que cambian con respecto a lo contenido en la ecuación (1) es que aquí se trabaja con el valor absoluto de las primeras diferencias para hacer resaltar las características de la señal sísmica. De esta manera el promedio corto se calcula así:

$$sta = \frac{1}{nm} \sum_{i=2}^{nm} |x_i - x_{i-1}|, \quad (5)$$

donde sta es el promedio corto, nm es el número de muestras para un promedio corto y el valor absoluto de las diferencias el cual es un filtro para reducir el ruido de baja frecuencia de la señal entrante (Lee y Stewart, 1981). Con los promedios cortos calculados, el siguiente paso es calcular los promedios largos de acuerdo a la siguiente fórmula:

$$lta = \frac{1}{npc} \sum_{j=1}^{npc} sta_j, \quad (6)$$

donde lta es el promedio largo, npc es el número de promedios cortos requeridos para calcular un promedio largo y sta_j son los promedios cortos. Una vez obtenido el primer promedio largo, usamos la siguiente fórmula para calcular los promedios largos subsecuentes:

$$lta_n = lta_{n-1} + sta_n - sta_1, \quad (7)$$

donde lta_n es el promedio largo más reciente, lta_{n-1} es el promedio largo anterior, sta_n es el promedio corto más reciente y sta_1 es el promedio largo más antiguo que se tiene. Recordemos que este algoritmo trabaja con ventanas móviles de tiempo. Debido a esto, es posible calcular los promedios largos de la manera planteada en la ecuación (7).

El procedimiento descrito se programó para el nuevo sistema de adquisición. Los parámetros usados en esta implementación son 64 muestras (nm) para los

promedios cortos y 64 promedios cortos (*npc*) para cada promedio largo. Considerando que la razón de muestreo utilizada es de 40 muestras por segundo, los valores de STA y LTA (en segundos) están dados en la siguiente tabla:

Tabla X . Parámetros usados en el algoritmo de detección

Promedios Cortos (STA)	Promedios Largos (LTA)	Umbral
1.6 segundos	102.4 segundos	5

El algoritmo descrito en esta sección se aplica constantemente a todas las señales que arriban a la computadora.

IV.7.2 Función de detección

Como se tienen varias señales llegando al mismo tiempo, hay que definir un esquema para seleccionar las estaciones que pueden declarar un evento. Este proceso se lleva a cabo por medio de un método de votación denominado función de detección. En esta función están declaradas varias estaciones pero sólo se declara un evento cuando éste se haya detectado en una o más estaciones predeterminadas. Para hacer esto, se toma el resultado de la aplicación del algoritmo de detección para cada estación, y se guarda dentro de un arreglo de bits. Una vez hecho esto se aplica una operación OR lógica con la función de detección predeterminada, y si cumple con ella entonces se procede a grabar las señales de todas las estaciones programadas para ello.

La función de detección es una función booleana. Esta función permite

especificar situaciones bajo las cuales se inicie el proceso de grabación (condición de disparo). Por ejemplo: la función se podría configurar de tal manera que se iniciara el proceso de grabación de información sísmica cuando las estaciones involucradas en la función de detección sobrepasen el umbral y además satisfagan la siguiente relación:

$$Disparo = (Estación1) \text{ o } (Estación2 \text{ y } Estación3) \text{ o } (Estación2 \text{ y } Estación5 \text{ y } Estación6) \dots(8)$$

Esto quiere decir que *Disparo* va a ser verdadero si:

- En la Estación1, el cociente $STA/LTA > \gamma_d$.
- Si en la Estación2 y en la Estación3 $STA/LTA > \gamma_d$.
- Si en la Estación2 y en la Estación5 y en la Estación6 $STA/LTA > \gamma_d$.

Esta función implica que, de las 12 estaciones en detección, las estaciones 4, 7, 8, 9 y 10 no se están considerando en nuestra relación booleana. También implica que todas las otras situaciones que se presenten no van a ser consideradas, por ejemplo cuando en la estación 3 el umbral sea mayor que el cociente de los promedios (STA/LTA). El uso de la función de detección es para darle flexibilidad a la condición de disparo ya que al revisar las señales de varias estaciones, puede haber casos en los que sea necesario observar con mayor atención una estación o deshabilitar otra que no esté funcionando correctamente (Hinojosa, 1988).

IV.7.3 Detección y grabación de eventos sísmicos.

El programa que aplica el algoritmo de detección (*stalta.c*) a las señales que provienen de las estaciones funciona de la siguiente manera. Primero se conecta con el módulo compartido de datos y, una vez establecida esta conexión, se inicializan los parámetros que maneja el algoritmo. A continuación, se lee e implementa la función de detección como se describió en la sección IV.7.2 y se leen y asocian los nombres de las estaciones con los puertos seriales correspondientes. Ya que se tienen los valores iniciales, entra en un ciclo infinito que se va a encargar de estar revisando y aplicando el algoritmo de detección cada vez que así lo indiquen los parámetros (Figura 22).

Cada vez que se aplica el algoritmo a una de las estaciones, el resultado de éste se guarda en un arreglo. A continuación los resultados del arreglo son comparados con todos los términos de la función de detección (ver Ecuación [8]). Si alguno de los disparos llega a coincidir con alguno de los términos que están dentro de la función, entonces se declara un evento y el sistema pasa a estado de grabación. Durante este proceso se graban archivos de todas las estaciones habilitadas para escritura, independientemente de la que haya declarado el evento.

IV.7.4 Grabación de archivos de eventos sísmicos

Una vez iniciado el proceso de grabación se procede a generar archivos con los

IV.7.4 Grabación de archivos de eventos sísmicos

Una vez iniciado el proceso de grabación se procede a generar archivos con los

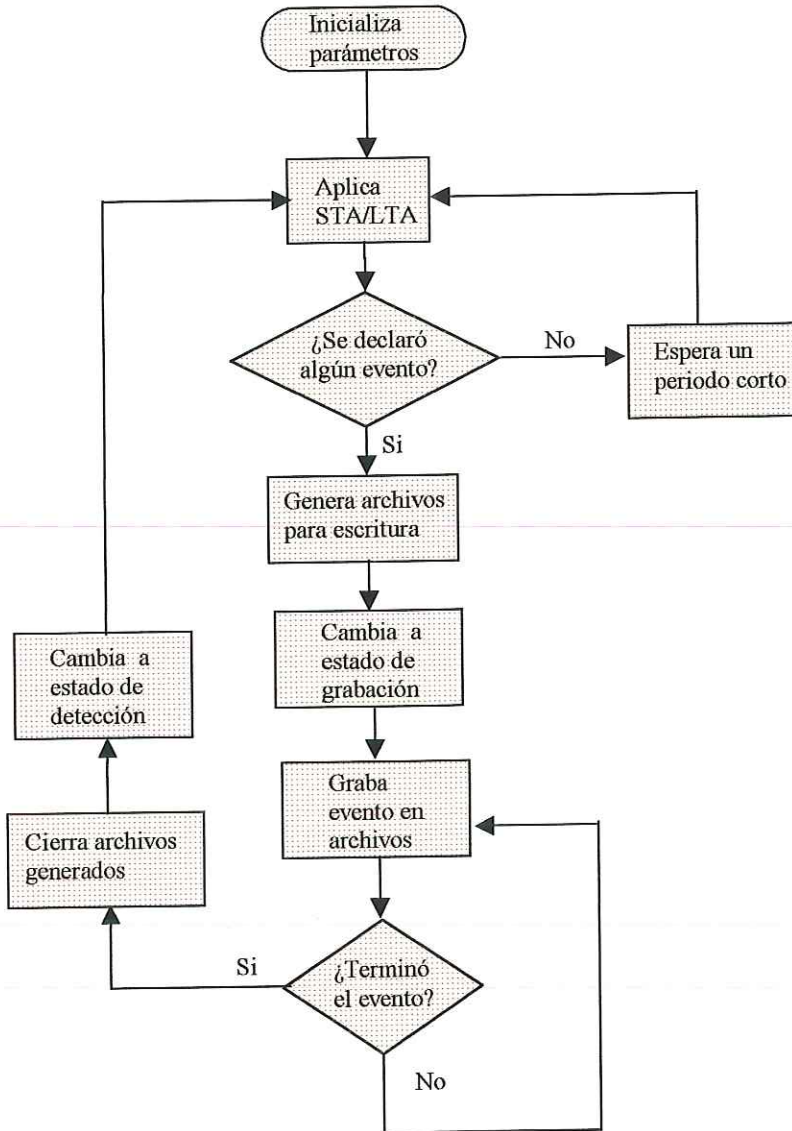


Figura 23. Esquema que representa el ciclo infinito ejecutado para hacer la detección de los eventos sísmicos

minutos de señal. Cuando el algoritmo de detección determina que hay un evento, se empieza a grabar quince segundos antes del tiempo de disparo. A este tiempo se le conoce como memoria pre-evento y su objetivo es evitar que se pierdan los primeros arribos (ondas P).

La grabación continúa hasta completar tres minutos que sirven como memoria post-evento de esta manera se evita el tener registros incompletos. En caso de que el algoritmo de detección determine que todavía existe señal sísmica, continúa con la grabación. La grabación se termina cuando el cociente STA/LTA es menor que el umbral de disparo y se completan los tres minutos de registro.

Con el propósito de preparar las señales grabadas para efectuar su procesamiento se diseñaron dos programas: *pone_enc.c* y *res2sei.c*. El primer programa se encarga de incluir un encabezado (fecha de grabación, número de muestras, valores máximos y mínimos) a los archivos grabados. El segundo programa se encarga de cambiar el formato RESNOM al formato Seisan (Haskov y Ottemöller, 2000). Como se mencionó anteriormente el programa Seisan es el que se utiliza para el procesamiento estándar (lectura de tiempos de arribo, localización y cálculo de magnitudes) de las señales sísmicas. Adicionalmente se hicieron unos *scripts* que se encargan de manejar de manera rápida y eficiente grandes cantidades de archivos, evitando con esto hacer la conversión individual de los eventos generados.

IV.8 Graficación en tiempo real.

Para visualizar la señal que arriba a la PC en tiempo real se hizo el programa

monitorgtk. Esta aplicación se escribió en lenguaje C usando las librerías para gráficos *GTK+ 1.2.10*. Como resultado el programa presenta una ventana con la gráfica de las señales que están arribando. Una opción que maneja **monitorgtk** consiste en hacer un acercamiento de la señal de hasta 8 cuentas como mínimo en la escala vertical. El programa recibe los datos de entrada de un archivo que contiene el valor más reciente de la componente vertical de cada estación. Cada 50 milisegundos el programa lee estos archivos y los escala para hacer un graficado apropiado de la señal.

El programa **monitorgtk** está conformado por tres módulos: *frontend.c*, *graph.c* y *misc.c*. Cada módulo se encarga de hacer una tarea diferente. El primer módulo se usa para la creación de la ventana principal y de algunos de sus componentes como la barra de herramientas que utiliza dentro de ella y la llamada a la función que crea el área de dibujo. En este módulo también se establece la conexión entre los elementos de la ventana y las acciones que se realizan cuando alguno de sus botones (extremo superior izquierdo de la ventana, Figura 23) es presionado. El segundo módulo contiene las funciones que dibujan en la ventana los componentes solicitados en la creación de la ventana. El tercer módulo se encarga de crear el área de dibujo y conectarla con las señales encargadas de su manejo (limpieza y actualización de datos). En este módulo se leen los datos de los archivos y se guardan en unas colas de datos usadas para hacer la animación de la gráfica (Figura 23). Para la escritura este programa, parte del código se obtuvo del libro escrito por Harlow (1999).

(limpieza y actualización de datos). En este módulo se leen los datos de los archivos y se guardan en unas colas de datos usadas para hacer la animación de la gráfica (Figura 23). Para la escritura este programa, parte del código se obtuvo del libro escrito por Harlow (1999).

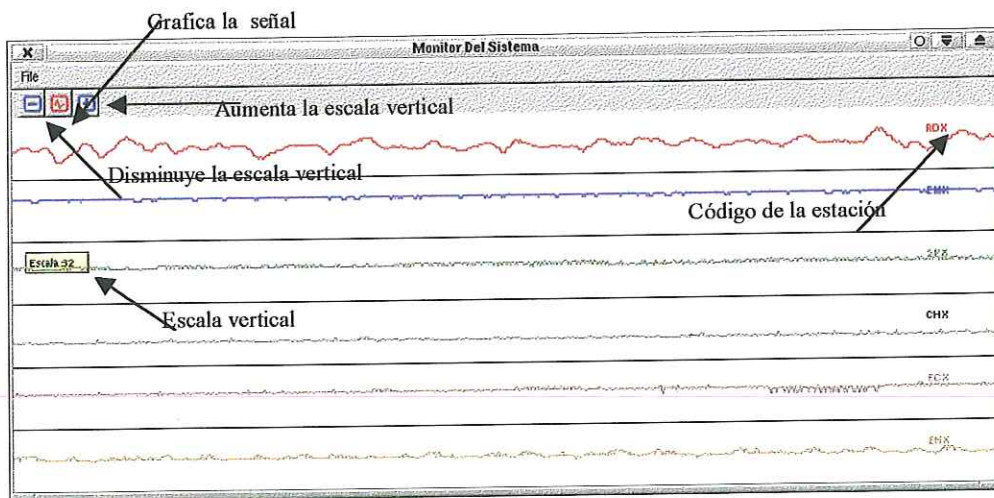


Fig. 24. Gráfica de la señal de periodo corto en tiempo real.

IV.9 RESULTADOS

IV.9.1 Algoritmo de detección

Con el propósito de ilustrar el funcionamiento del algoritmo de detección se eligieron eventos con diferentes características en la señal sísmica (Tabla XI). Cuando se registra un evento, la relación entre la señal sísmica y el ruido del sitio de observación (S/R) nos será de utilidad para determinar la calidad del sismograma obtenido. Si el ruido es alto, la relación S/R será baja y se tendrá un sismograma en el que será difícil apreciar la señal sísmica. En contraste, si el ruido es bajo la relación S/R es alta y la señal sísmica se apreciará con claridad. Con respecto a las fases sísmicas, primero arriba a una estación de registro las ondas P. Estas pueden ser impulsivas (de gran amplitud) o emergentes (de menor amplitud) que en algunos

casos puede ser comparable con el ruido ambiental. El sistema de adquisición diseñado se probó con cada uno de los casos mencionados y los resultados se ilustran a continuación.

Tabla XI Tipos de eventos seleccionados para probar el algoritmo de detección.

<i>Características de la señal</i>	<i>Fecha</i>	<i>Hora</i>	<i>Localizacion</i>		<i>Magnitud</i>
Impulsivo	9 de Junio 2002	10:27:00	32-15.2 N	115-06.8 W	3.2
Emergente	22 de Julio 2002	10:08:00	29-59.4 N	115-36.8 W	3.5
Ruidoso	28 de Agosto 2002	22:32:00	31-56.7 N	115-40.0 W	2.5
Regional	31 de Agosto 2002	16:24:00	33-01.3 N	115-29.0 W	3.8
Picos	11 de Sept. 2002	15:54:00	31-52.8 N	116-22.2 W	3.1
Telesismo	19 de Agosto 2002	11:12:00	29-42.0 S	179-30.6 E	7.7

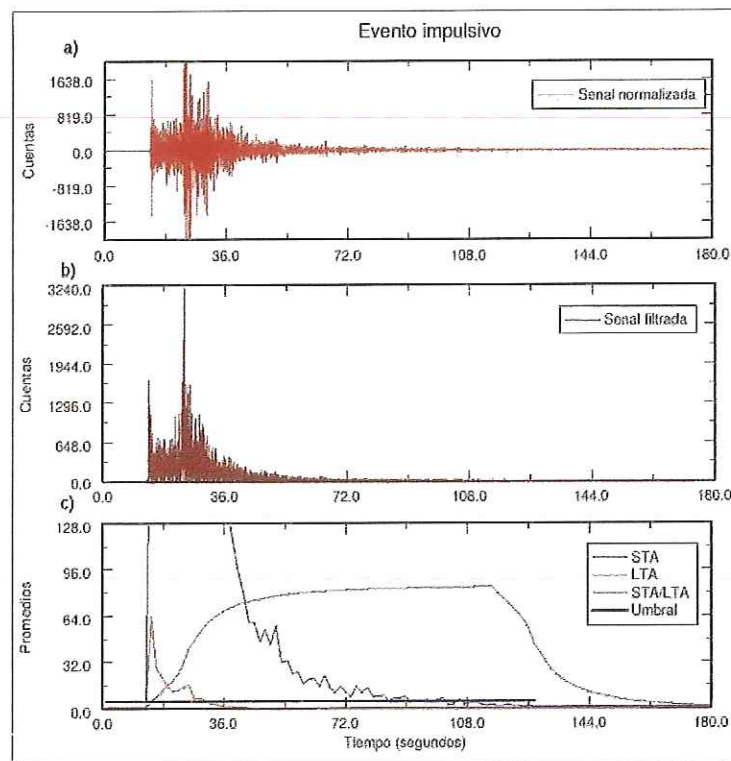


Figura 25. Gráfica que ilustra el funcionamiento del algoritmo de detección para el evento del 9 de junio de 2002, M=3.2

En la gráfica superior se puede ver la señal del evento con la media removida. La eliminación del valor medio es importante para eliminar el “*offset*” (corrimiento hacia arriba o hacia abajo) de la señal original y lograr un buen funcionamiento del algoritmo. La gráfica intermedia ilustra la señal filtrada. Nótese en esta gráfica cómo se hacen resaltar las características de la señal sísmica, lo cual no ocurre con el ruido previo a la onda P ni cuando ha decaído la amplitud (~70s). La gráfica inferior muestra los resultados de aplicar el algoritmo de detección. Con la línea azul se puede apreciar los promedios cortos de la señal (STA), con línea verde tenemos los promedios largos (LTA) y la línea roja representa el cociente STA/LTA. La línea negra representa el umbral que debe rebasar el cociente de los dos promedios. En este caso el umbral seleccionado es de 5 unidades. En el Apéndice A podemos encontrar el resto de las gráficas obtenidas para los diferentes tipos de eventos mostrados en la Tabla XI.

IV.9.2 Eventos registrados

Una prueba importante para el sistema nuevo fue el comparar los resultados obtenidos con los del sistema antecesor. Este periodo de prueba se llevó a cabo durante tres meses del año 2002 y los resultados obtenidos se muestran en la Figura 25. De esta figura podemos observar que el sistema nuevo registra un número mayor de eventos que el anterior. La diferencia es más notable a partir de la segunda mitad del primer mes analizado, ya que antes la detección se hizo con la componente (Norte-Sur). En los meses siguientes se observa una clara mejoría en la cantidad de eventos registrados por el sistema.

del primer mes analizado, ya que antes la detección se hizo con la componente (Norte-Sur). En los meses siguientes se observa una clara mejoría en la cantidad de eventos registrados por el sistema.

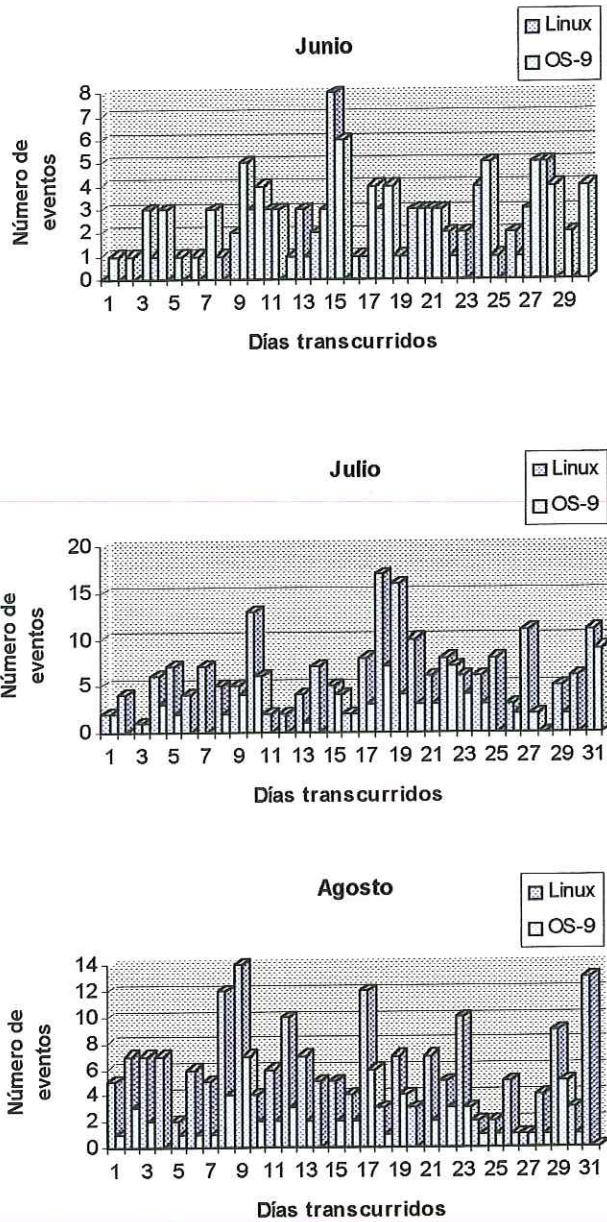


Figura 26. Gráfica comparativa de eventos registrados en los dos sistemas durante Junio, Julio y Agosto de 2002.

Otro de los resultados consistió en que, aparte de que se detectan más sismos, en algunos casos el sismo se detectó en un mayor número de estaciones.

Comparación de eventos registrados por estación

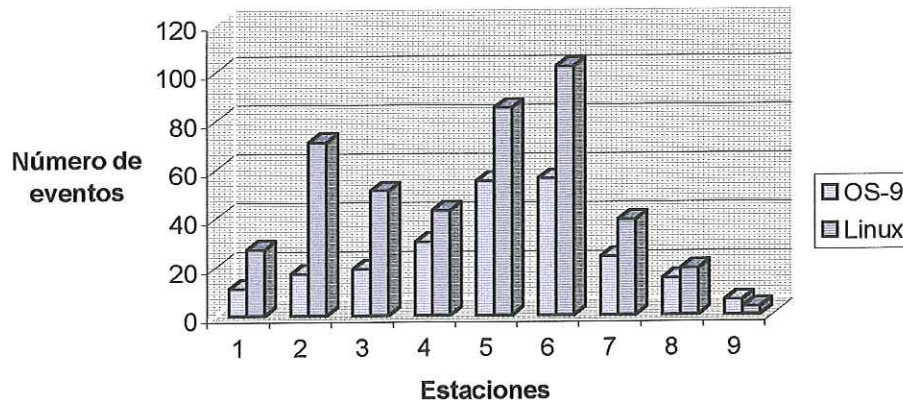


Figura 27. Gráfica comparativa del número de eventos detectados por estación.

La razón exacta acerca de porque se registró un número mayor de eventos en el sistema basado en Linux, puede deberse a dos causas: a pesar de ser sistemas iguales que se manejan con los mismos parámetros, existió una diferencia en los términos de la función de detección usada en uno y otro. La otra razón puede ser debido a que ahora se hace una depuración más rápida de los datos, lo que permite analizar cantidades grandes de datos en una cantidad pequeña de tiempo, lo que permite que se revisen todos los archivos generados, en vez de ir directamente solo sobre lo sobresaliente, como se hacia cuando se saturaba el sistema antecesor.

IV.9.3 Tiempos de arribo de las fases sísmicas.

Otro aspecto muy importante fue asegurarse que los tiempos de arribo de las

la localización epicentral del evento. Para la revisión de los tiempos se tomaron varios sismos al azar y se leyeron los tiempos de arribo de las fases P y S con el uso del programa Seisan. Un ejemplo de los resultados obtenidos con ambos sistemas se muestran en la Tabla XII. De esta tabla se aprecia que las diferencias son de máximo 3 centésimas de segundo entre los tiempos obtenidos con uno y otro sistema. Estas diferencias pueden ser atribuibles a errores de apreciación.

Tabla XII. Lecturas de los tiempos de arribo registrado en los dos sistemas para el sismo del 02 de marzo del 2003 a las 00:04

Sistema OS-9				Sistema Linux			
Est.	Canal	Fase	Tiempo	Est.	Canal	Fase	Tiempo
Diferencia							
ECX	SZ	IP	005 7.44	ECX	SZ	IP	005 7.41
							0.03
ECX	SN	ES	005 27.18	ECX	SN	ES	005 27.16
							0.02
ECX	SE	ISG	005 49.32	ECX	SE	ISG	005 49.29
							0.03
ENX	SZ	IP	005 14.64	ENX	SZ	IP	005 14.65
							-0.01
ENX	SN	ES	005 25.98	ENX	SN	ES	005 25.99
							-0.01
ENX	SE	ISG	006 17.04	ENX	SE	ISG	006 17.02
							0.02
PBX	SZ	IP	005 16.24	PBX	SZ	IP	005 16.24
							0.00
PBX	SN	ES	005 31.83	PBX	SN	ES	005 31.84
							-0.01
PBX	SE	ISG	006 12.90	PBX	SE	ISG	006 12.88
							0.02
PGX	SZ	IP	005 9.34	PGX	SZ	IP	005 9.35
							-0.01
PGX	SN	ES	005 23.94	PGX	SN	ES	005 23.94
							0.00
PGX	SE	ISG	005 37.44	PGX	SE	ISG	005 37.42
							0.02
RDX	SZ	IP	005 5.71	RDX	SZ	IP	005 5.72
							-0.01
RDX	SN	ES	005 11.45	RDX	SN	ES	005 11.44
							0.01
RDX	SE	ISG	005 10.74	RDX	SE	ISG	005 10.76
							-0.02

Siguiendo un procedimiento similar al descrito anteriormente, se escogieron 15 sismos en cuyos sismogramas se leyeron un total de 182 fases (P y S). La media de las diferencias entre las lecturas de uno y otro sistema (Anterior – Nuevo, Figura 27) es de 0.018 segundos y la desviación estándar es de 0.03 segundos. Este resultado

las diferencias entre las lecturas de uno y otro sistema (Anterior – Nuevo, Figura 27) es de 0.018 segundos y la desviación estándar es de 0.03 segundos. Este resultado comprueba que la diferencia entre ambos sistemas en los tiempos de arribo es pequeña (< 1 muestra) por lo que se tiene confianza de que los tiempos de arribo obtenidos con el sistema nuevo son correctos.

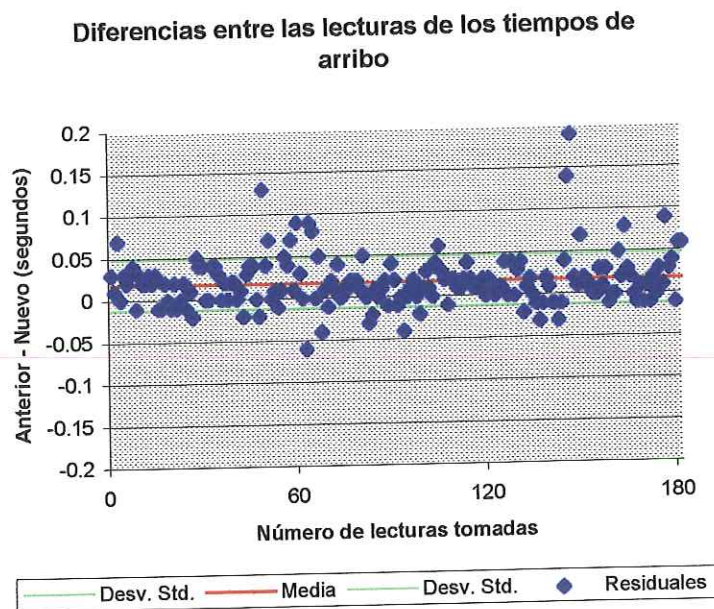


Figura 28. Residuales entre las lecturas de P y S obtenidas con ambos sistemas (OS-9 y Linux)

IV.9.4 Polaridades y amplitudes

Además de los tiempos de arribo, otro aspecto importante de las señales sísmicas son la polaridad y la amplitud. Debido a esto se realizaron pruebas para comprobar que tanto la amplitud como la polaridad fueran correctas. En las gráficas de la Figura 28 se muestra el evento registrado el 20 de marzo de 2003 tanto por el sistema anterior (Figura 28 a) como por el sistema nuevo (Figura 28 b). Se puede

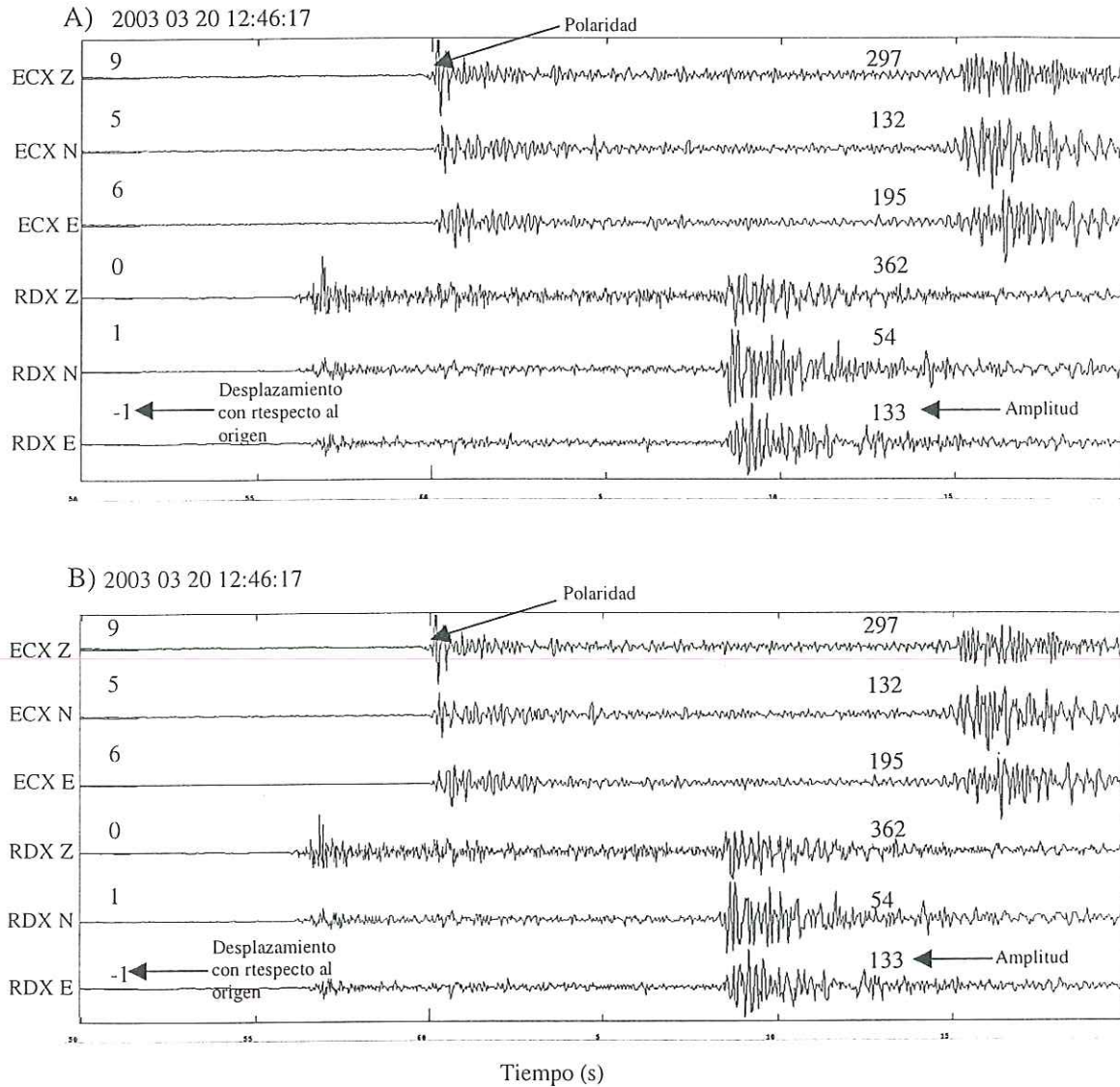


Figura 29. Comparación de la polaridad y las amplitudes del sismo ocurrido el 20 de marzo de 2003 a las 12:42 ($M = 3.5$) y registrados por los sistemas de adquisición basados en OS-9 (a) y Linux (b).

En sismología la polaridad de la señal es útil para, por ejemplo, obtener una localización aproximada del epicentro del sismo. Por otro lado, la amplitud es importante para calcular su magnitud.

Se hicieron más pruebas con otros eventos obteniéndose los mismos resultados en ambos sistemas, por lo que se llegó a la conclusión de que tanto las polaridades de los sismos registrados en el sistema basado en Linux, como sus amplitudes son correctas.

IV.10 Programas adicionales

Debido a que no toda la información que se registra en el sistema son eventos sísmicos si no que también se graba una cierta cantidad de disparos en falso, es necesario hacer una depuración de los datos. Para ello se diseñó el programa **examina** con el propósito de revisar de una manera rápida y eficiente toda la información grabada y seleccionar sólo la información útil (señal sísmica). Esta aplicación selecciona los eventos por fecha y despliega en pantalla una gráfica de la componente vertical de todas las estaciones de las que se tenga información (Figura 29). Una vez que se tiene el evento graficado en pantalla, el usuario tiene las opciones de borrar todos los archivos (en caso de que no contengan información sísmica) o eliminar solamente aquellos registros sin señal.

El programa **examina** fue escrito en lenguaje C usando las librerías para gráficos GTK+, consta de 5 módulos: *archivos.c*, *listas.c*, *menu.c*, *grafica.c* y *main.c*. Cada módulo realiza tareas diferentes como se explica a continuación. El primer módulo se encarga de leer los archivos de eventos en el formato propietario de RESNOM, y de poner la información en un arreglo global para que pueda ser accedido por los otros módulos. El segundo módulo se encarga de generar las listas

primer módulo se encarga de leer los archivos de eventos en el formato propietario de RESNOM, y de poner la información en un arreglo global para que pueda ser accedido por los otros módulos. El segundo módulo se encarga de generar las listas con las fechas de los eventos y también buscar los archivos que corresponden a cada fecha para desplegarlos. El tercer módulo se encarga de manejar las opciones de la barra del menú y el *status bar*. El cuarto módulo se encarga de dibujar las señales en el área de dibujo además de desplegar información acerca del evento (fecha, valores máximo, mínimo y número de segundos graficados). Por último, el quinto módulo se encarga de inicializar el ciclo de GTK+ y de hacer la llamada para la creación de la ventana principal.

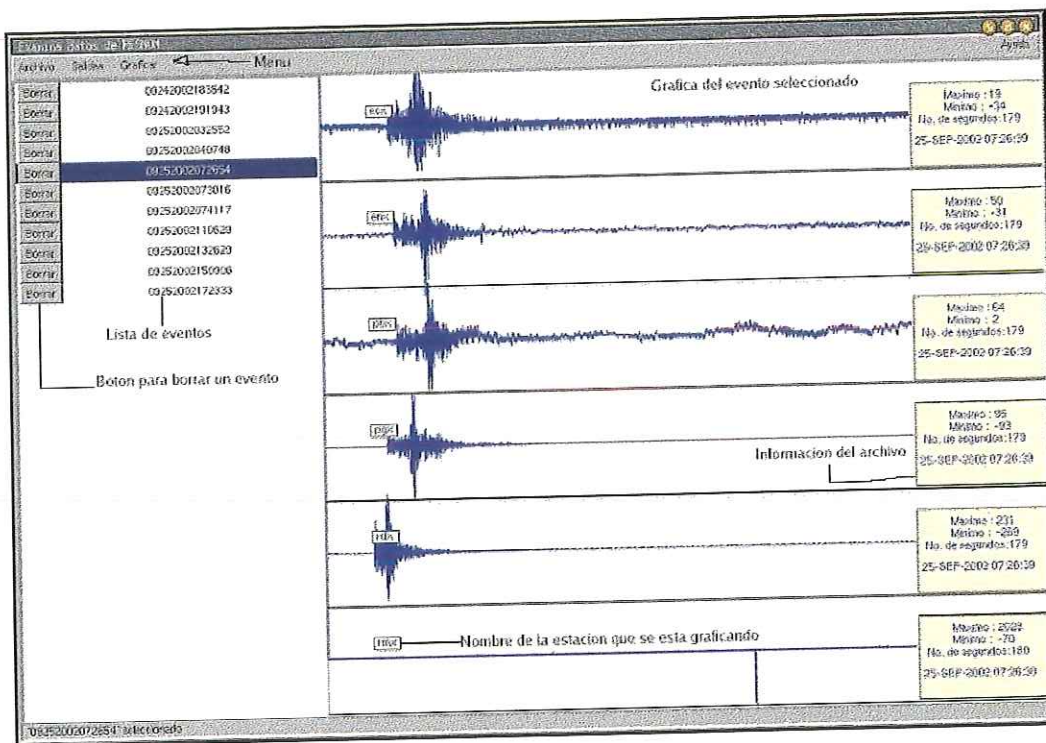


Figura 30. Ventana principal del programa examina.

Una vez que se hizo la depuración con **examina** y se tiene una lista con los

archivos útiles, éstos se convierten al formato que maneja el programa Seisan, el cual se utiliza para hacer el análisis de la información (lectura de tiempos de arribo, medición de amplitudes, localización, entre otros). El programa hace una llamada a un *script* que se encarga de hacer el trabajo de la conversión de formatos, de una manera transparente para el usuario. El siguiente paso consiste en transferir los archivos, vía ftp, hacia la computadora en donde radica la base de datos de los sismos registrados.

Otras opciones que nos ofrece **examina** es la de visualizar las tres componentes de las señales que conforman a cada archivo (Figura 30). Una opción más nos permite conectarnos con el programa que grafica la señal de las estaciones en tiempo real. Estas opciones son complementarias y se hicieron con el propósito de revisar el funcionamiento correcto de las tres componentes en cada estación.

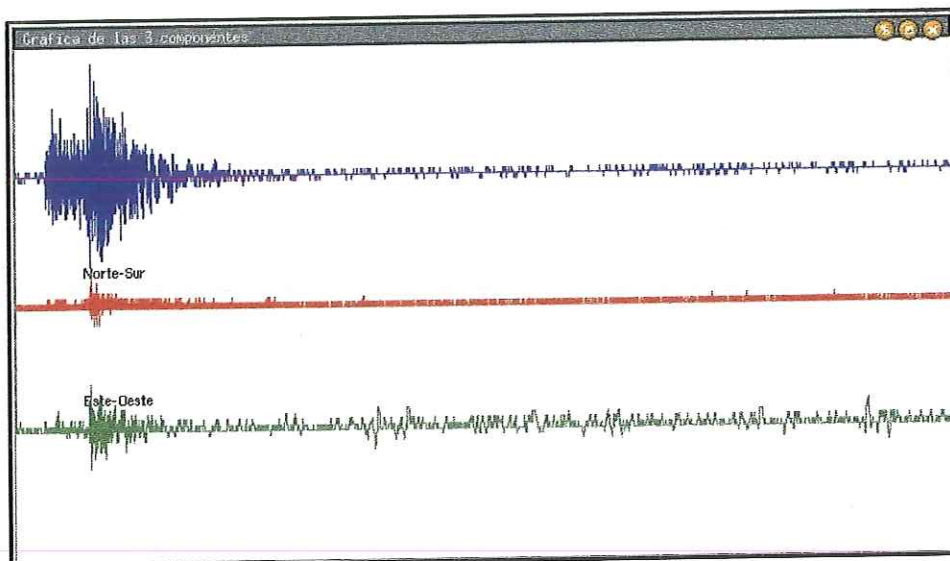


Figura 31. Gráfica de las tres componentes de un archivo.

Adicionalmente se desarrolló otro programa llamado **monitor** (escrito en

lenguaje C), el cual tiene el propósito de permitirnos la interacción con los parámetros del algoritmo de detección por medio de una interfaz en modo texto. Este programa permite modificar el umbral de detección, las funciones de detección y habilitar estaciones para escritura y detección (Figura 31). Además de estas características tiene opciones para monitorear el estado de las señales, el funcionamiento del algoritmo de detección y presentar estadísticas de las señales que arriban a la computadora.

Los resultados discutidos en las secciones IV.9 y IV.10 del presente capítulo demuestran que el sistema desarrollado para la adquisición de las señales de periodo corto de RESNOM es funcional y cumple satisfactoriamente con los objetivos planteados para este sistema (ver sección I.3). Actualmente (agosto de 2003) tanto el sistema desarrollado (objeto del presente trabajo de tesis) como el sistema antecesor están funcionando de forma simultánea, no obstante, en los próximos días el sistema nuevo reemplazará en su totalidad al sistema antecesor.

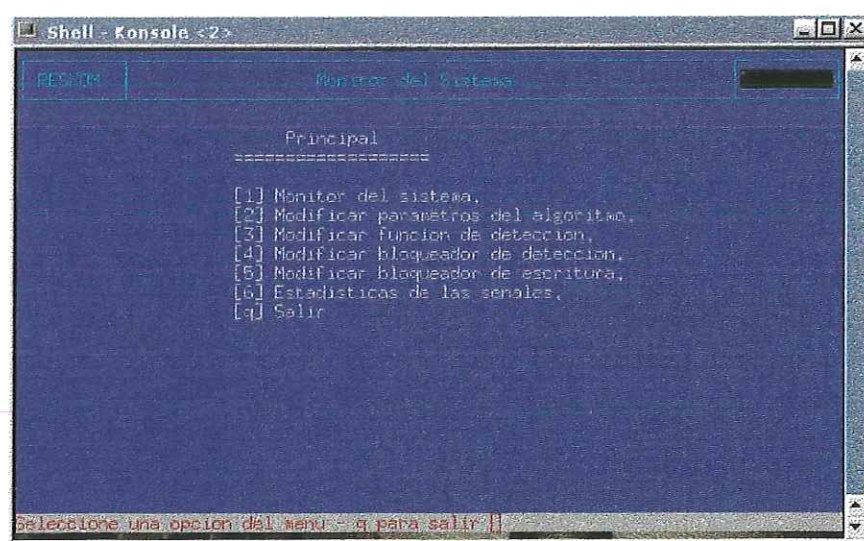


Figura 32. Pantalla principal del programa monitor

IV CONCLUSIONES

Con el desarrollo y la implementación de dos sistemas para la adquisición de datos de periodo largo y de periodo corto, se cumplieron los objetivos planteados en el presente trabajo de tesis. El sistema operativo Linux fue una buena elección para la implementación de los sistema de adquisición. El sistema operativo Linux demostró ser muy estable y tener la capacidad suficiente tanto para manejar simultáneamente la adquisición de datos, como para hacer el procesamiento de las señales registradas. La información generada por estos sistemas conforma la base de datos de RESNOM la cual es de utilidad para el estudio de la sismicidad del norte de Baja California. Esta información es usada para la realización de estudios sísmicos de la región.

El sistema de adquisición de datos de periodo largo fue rediseñado en su totalidad. Si bien existía un sistema de adquisición para este tipo de datos, el sistema desarrollado permitió mejorar sustancialmente el sistema anterior. Este sistema graba de manera continua la señal recibida y mediante un algoritmo de detección, se extrae la señal sísmica de los archivos grabados. Una de las ventajas que tiene este sistema con respecto al anterior es que cuenta con los medios para la visualización de la señal, tanto en tiempo real, como la registrada en los archivos generados.

El sistema de adquisición de datos de periodo corto desarrollado esta basado en una versión previa anterior desarrollada por Hinojosa (1988). Esta versión, implementada sobre el sistema operativo OS-9, empezaba a tener fallas continuas, debidas a problemas de hardware. Por lo tanto fue necesaria una actualización y mejora del sistema de adquisición de datos de periodo corto. Por este motivo, y

basado en la documentación existente, se realizó programación nueva adaptada a las características de Linux, con base en el cual se hizo el desarrollo. El sistema de adquisición previo tenía más de quince años en funcionamiento y durante ese tiempo había probado su efectividad, por lo que el nuevo sistema tenía que ser de la misma calidad y mejorar el aspecto de la visualización de las señales grabadas. El sistema desarrollado cumple de manera satisfactoria con los requisitos mencionados y ha sido sometido a varias pruebas (revisión de tiempos de arribo de las fases sísmicas, polaridad y amplitudes de las señales), las cuales ha cumplido satisfactoriamente.

En lo que respecta al proceso de la depuración de los archivos registrados, éste es más eficiente. Esto se logró por medio de programación en un ambiente gráfico de ventanas, lo que facilita muchas de las tareas que se llevaban a cabo anteriormente desde la línea de instrucciones. Estas tareas ahora se realizan de una manera transparente para el usuario. También aprovechando las capacidades del ambiente gráfico de Linux ahora es posible la visualización simultánea de varias estaciones en tiempo real por medio de un programa que presenta en una ventana las trazas de las estaciones que están arribando.

Para el sistema de adquisición de señales de periodo corto el algoritmo de detección basado en el cociente de promedios cortos entre promedios largos funciona de manera correcta. Este algoritmo se encarga de analizar las señales entrantes y seleccionar sólo la señal sísmica. En el caso del sistema de adquisición de señales de periodo largo, la grabación de la señal es continua y a partir de los registros grabados se extrae la señal de interés. Por otro lado se realizó programación adicional para el

manejo de la señal grabada tanto para el sistema de periodo largo como para el de periodo corto. Dicha programación se encarga del manejo de la señal para cambiarla a un formato homogéneo (formato SeisAn) que permite trabajar con los diferentes tipos de señales que se manejan (periodo largo, periodo corto y banda ancha).

Al empezar este proyecto yo contaba con conocimientos sobre el análisis, diseño y programación de sistemas de cómputo. No obstante, al realizar el desarrollo de los sistemas de adquisición, aprendí nuevas técnicas de programación, amplié mis conocimientos sobre el sistema operativo Linux, y sobre todo aprendí a enfrentar y resolver una situación del mundo real. Algo que me ayudó mucho para desenvolverse como profesionalista fue el aprender a trabajar con un equipo interdisciplinario de personas con conocimientos en áreas diferentes a la mía. La comunicación establecida con ellos y su cooperación y disposición fue muy importante para la realización de este trabajo.

Referencias

Barlow, Diane, A. Robbins, P. Rubin, R. Stallman, 1989, The GAWK manual,
Edition 0.11 Beta

Frerking, Gary y P. Berm, 2001, Serial Programming HOWTO, 2001, Revision 1.01

Gale, T. y Main I. 2000. GTK+ v1.2 Tutorial.

Goldt Sven, S. Van der Meer, S. Burkett, M. Welsh, 1995, The Linux programmers
guide, Sven Goldt, sachsendamm, 10829, Berlin Germany.

Harlow, E. 1999, Guia avanzada desarrollo de aplicaciones Linux con GTK+ y
GDK, Prentice Hall Iberia, Madrid, España, 496 p.

Havskov, J. y L. Ottenmøller, 2000, Seisan: The Earthquake analysis software for
windows, solaris and linux, version 7.1. Institute of solid Earth Physics,
University of Bergen, Norway. 236 p.

Hinojosa, A. 1988. Sistema de captura y deteccion de eventos sísmicos de la red
sismológica RESNOR, basado en una computadora personal IBM-PC.
Tesis de maestría, CICESE, Ensenada Baja California, 119 pp.

Hinojosa, A. C. Duarte y L. Inzunza. 1989. Sistema de captura y procesamiento de la

red sísmológica RESNOR. Memorias de la reunión 1987 de la Unión Geofísica Mexicana, 38-45.

InstrumenTAL software, 2003. Introduction to RS232 serial communications.

www.taltech.com/TALtech_web/resources/intro-sc.html,

taltech software.

Lawyer David y G. Hankins, Serial HOWTO, Marzo 2002, version 2.16

Lee W. H. K., S. W. Stewart. 1981. Principles and applications of microearthquake networks. En B. Saltzman, Ed., Advances In Geophysics, Academic Press. 293 pp.

Loosemore, Sandra, R. Stallman, R. McGrath, A. Ovam y U. Drepper, 1999, The GNU C library reference manual, Edition 0.08 DRAFT, Version 2.1, Beta

Marshall, A. 1999. Programming in C UNIX system calls and subroutines using C. <http://www.cs.cf.ac.uk/Dave/C/CE.html>. Cardiff University.

Martin, D. 2000. Teach yourself GTK+ programming in 21 days. Sams Publishing. 936 p.

Microware, 2000. Getting started with enhanced OS-9 for x86, Microware.

National Instruments, 2003, Desarrollo de sistemas de adquisición de datos y control de tiempo real con tecnologías estándar, www.ni.com. National Instruments Corporation

Rad Data Communications, 1995. Interface RS232-C,
www2.rad.com/networks/1995/rs232/back.html

Smith, S. 1999. The Scientist and Engineer's guide to Digital Signal Processing. Second Edition. California Technical Publishing. 650 p.

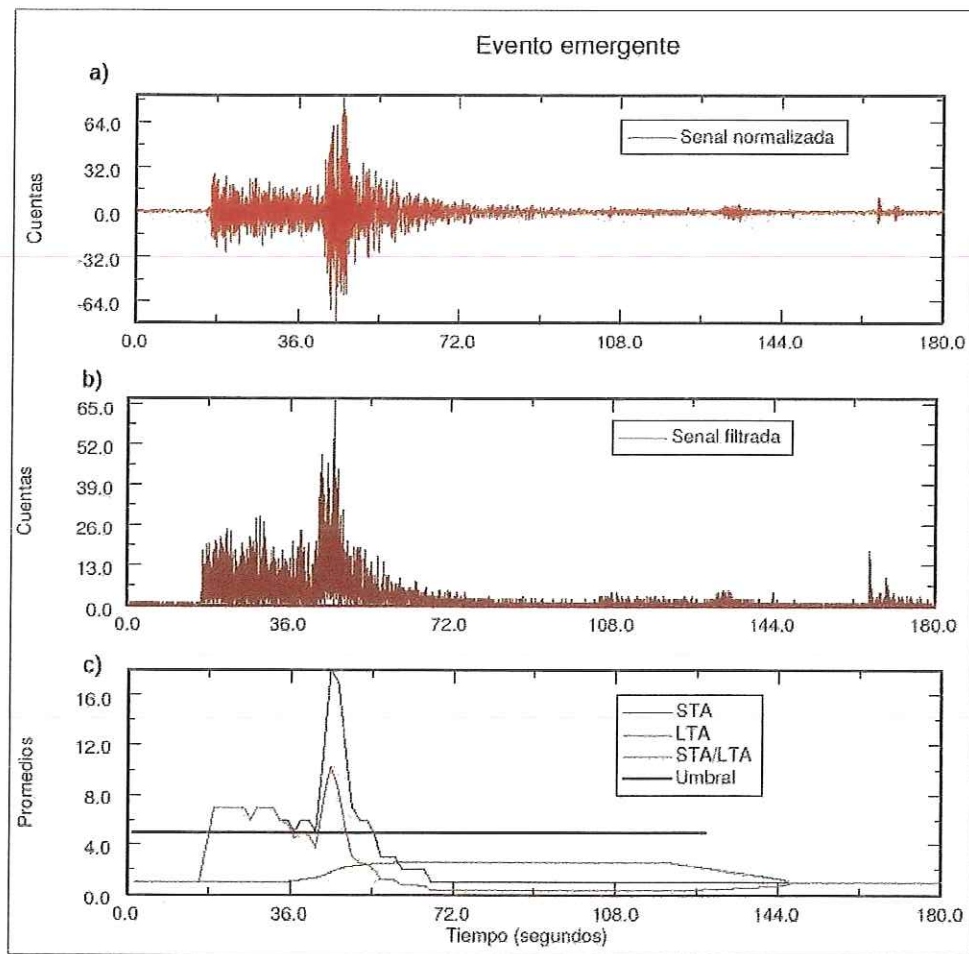
Trnkoczy A. 2001. Understanding and parameter setting of STA/LTA trigger algorithms, En P. Borman, Ed., IASPEI New manual of seismological observatory practice, GeoForschungsZentrum Postdam.

Vidal, A. y L. Munguía, 1993, Diez años de funcionamiento de la red sísmica del noroeste de México. Ciencia y Desarrollo, 18, 108, 77-85.

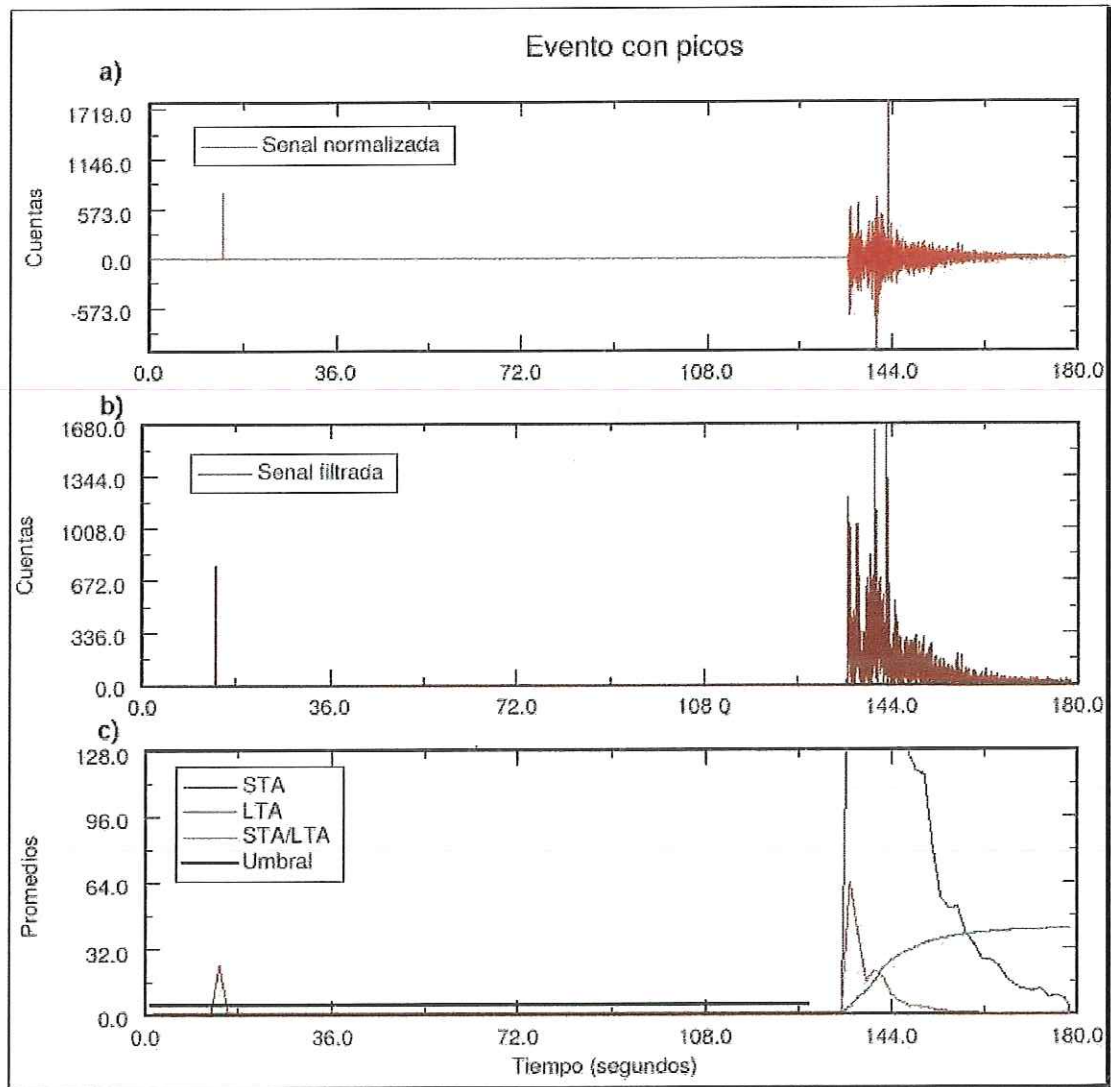
Vidal, A., L. Orozco, S. Arregui, J. Sánchez, O. Gálvez, I. Méndez, y L. Inzunza. 2003. A note on upgrading long period seismographs. Geofísica Internacional, 42, 1, 53-59.

Apéndice A. Figuras de resultados de la aplicación del algoritmo de detección

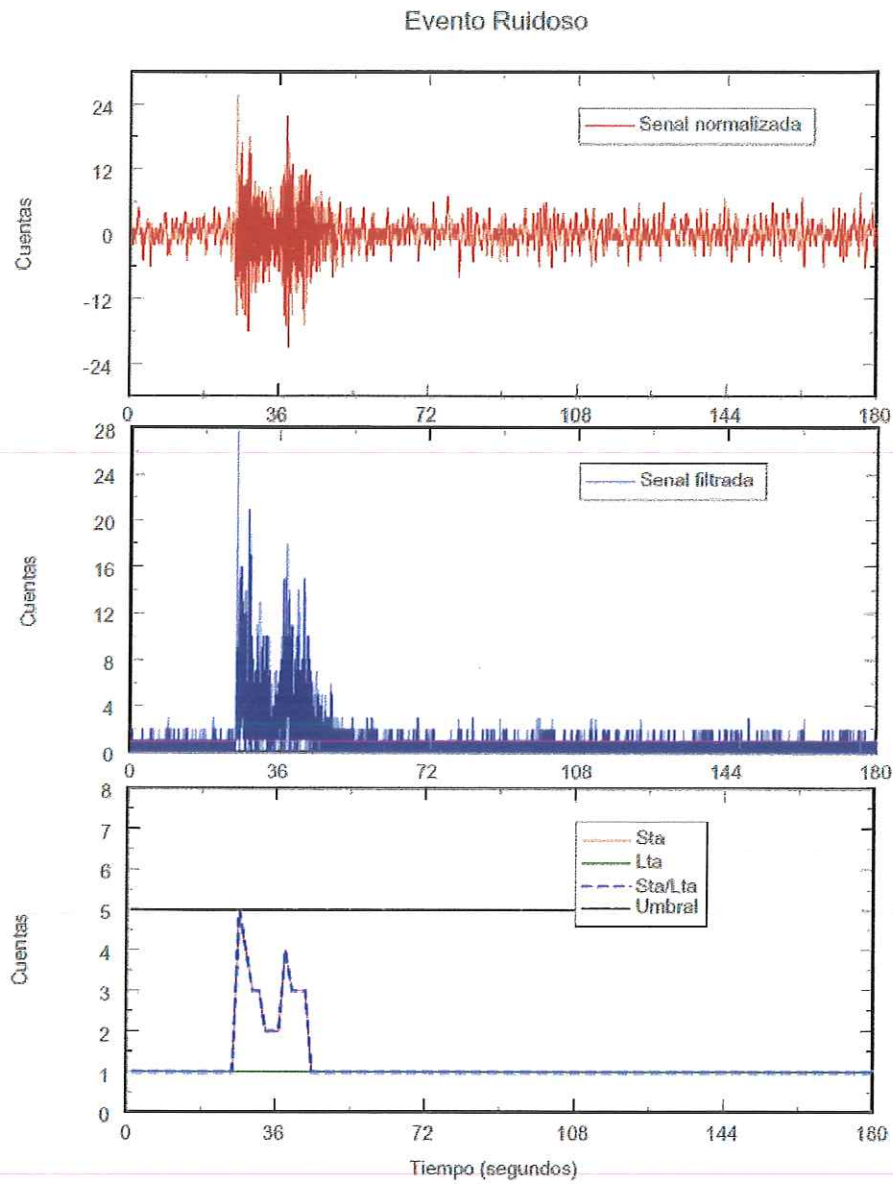
A continuación se presentan las figuras restantes de la aplicación del algoritmo de detección a diferentes tipos de eventos. La primer figura muestra el caso de un evento emergente, la segunda muestra los resultados para un evento con picos, la tercera muestra un evento regional, y por último se muestra el caso de un evento lejano.



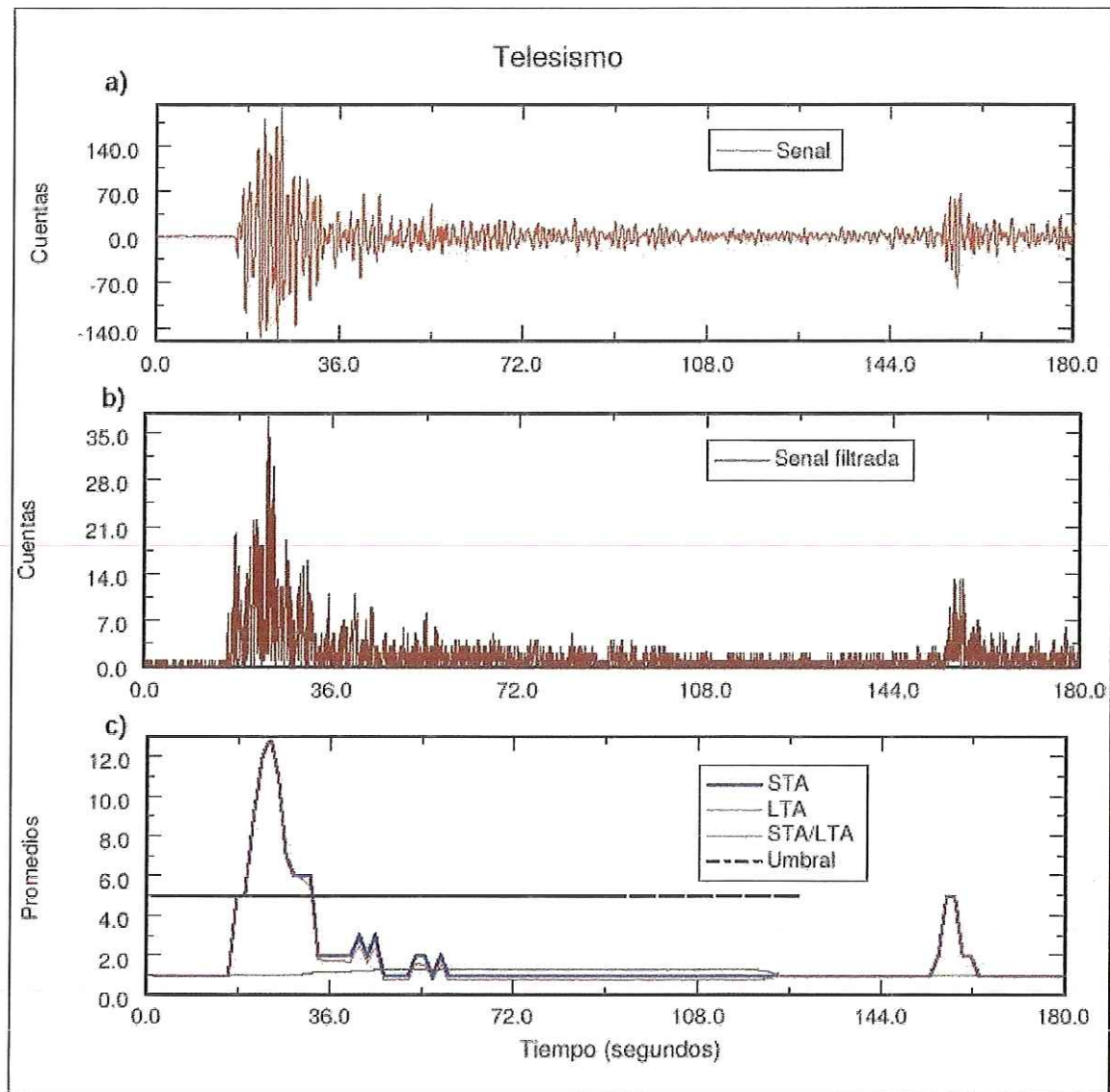
Evento ocurrido el 22 de Julio de 2002 a las 10:08 (UTC) localizado en 29-59.40N, 115-36.80W con magnitud de 3.5



Evento ocurrido el 11 de Septiembre de 2002 a las 15:54 (UTC) localizado en
31-52.80N, 116-22.20W con magnitud de 3.1



Evento ocurrido el 28 de Agosto de 2002 a las 22:32 (UTC) localizado en
31-56.70N, 115-40.00W con magnitud de 2.5



Evento ocurrido el 19 de Agosto de 2002 a las 11:12 (UTC) localizado en 29-42.00S, 115-30.63W con magnitud de 7.7

Apéndice B Implementación en C de los algoritmos usados

En este apéndice se mostrará la implementación que se hizo en lenguaje C de los algoritmos descritos en la tesis. Para la adquisición de datos se realizaron tres algoritmos que se describen en las secciones III.2, III.3 y III.4 del Capítulo III. A continuación se muestra cada uno de ellos. En primer lugar vamos a ver la implementación del cambio de formato de la señal que llega del puerto serie al formato propietario en que se guarda la señal.

```
canal = muestra & CANAL; /*Sacamos el canal*/
muestra = muestra >> 1; /*tiramos el ultimo bit del LSB*/
palta = muestra & ALTA; /*Preparamos los datos para armarlos*/
palta = palta >> 2;
pbaja = muestra & BAJA;
valor = palta | pbaja; /*Armamos el dato*/
bituno = muestra & PBITID; /*Sacamos los bits de identificacion*/
bitdos = muestra & SBITID;
bitdos = bitdos << 4; /*Aqui se hace la rotacion de bits*/
id = bituno | bitdos; /*Acomodamos los bits de id en su lugar*/
muestraf = id | valor; /*Armamos la muestra con formato resnor*/
```

Ahora veremos el algoritmo usado para la separación de las estaciones que vienen multiplexadas, este algoritmo se explica en la sección IV.3.2 del capítulo IV :

```
while((i=read(fd,&muestra,1))) /* Leemos 1/2 muestra del puerto serie */
{
    if(muestra != 0xff) /* si no son bytes de sincronia entonces */
    {
        tren[cuenta] = muestra; /* ponemos las 3 muestras en un arreglo */
        cuenta++;
        if(cuenta == 6) /* cuando ya tenemos completas las 3 muestras */
        {
            for(j = 1; j <= 5; j+=2)
            {
                /* armamos las muestras y les ponemos una etiqueta */
                /* para saber de que estacion es cada muestra */
                muestra_completa = tren[j-1]|(tren[j]<<8);
                /* las mandamos a darle el formato de resnom */
                Da_Formato3est(muestra_completa, estacion);
                estacion++;
            }
            estacion = 1; /* limpiamos contadores */
            cuenta = 0;
        }
    }
    else
        cuenta = 0; /* si son bytes de sincronia volvemos a empezar la
cuenta */
}
```

Para la descripción hecha en la sección III.4, Capítulo III del algoritmo por medio del cual se insertan las marcas de tiempo a la señal se usa el siguiente código.

```

res=read(tiempo, &buf, 16); /* Leemos la cadena de entrada */
buf[res]=0;
dia = atoi(&buf[1]); /* asignamos los valores de la */
hora = atoi(&buf[5]); /* cadena a las variables que */
minuto = atoi(&buf[8]); /* vamos a usar */
segundo = atoi(&buf[11]);
calidad = buf[13];
retcar = buf[14];
enter = buf[15];
if(enter != '\n')/* si la cadena recibida no esta sincronizada */
return(0); /* regresamos un valor para indicarlo asi */
tdia = dia << 6; /* recorremos el valor dia 6 pos. a la izq. */
/* juntamos el dia con la hora para la primera marca de tiempo */
primer_marca = tdia | hora;
final1 = primer_marca | PRENDEBIT; /* para marcarlo como tiempo */
/* recorremos el valor de calidad en 12 pos a la izq */
tcalidad = (conv_cal(calidad)) << 12;
/* recorremos el valor del minuto 6 pos a la izq */
tminuto = minuto << 6;
/* juntamos calidad con minuto */
segunda_marca_1 = tcalidad | tminuto;
/* juntamos lo anterior con los segundos */
segunda_marca = segunda_marca_1 | segundo;
/* lo marcamos como estampa de tiempo */
final2 = segunda_marca | PRENDEBIT;
/* pone la marca de tiempo a la senal */
EstampaTiempo(final1, final2);

```

Para la implementación del algoritmo de detección (sección IV.5, Capítulo IV

) se realizó el siguiente código:

```

for(i=0; i<ESTACIONES; i++)
{
if((estaciones & j) && ( i >= PRIMERA) && ( i <= ULTIMA))
{
if((promedio=calcula_promedios_cortos(i))
{
shmptr->atras_pcortos[i] = ++shmptr->atras_pcortos[i] %
perlargo;
shmptr->promedios_largos[i] += promedio -
shmptr->pcortos[i][shmptr->atras_pcortos[i]];
plargo = shmptr->promedios_largos[i]/perlargo;
if(!plargo)
plargo=1;
shmptr->pcortos[i][shmptr->atras_pcortos[i]] = promedio;
shmptr->sta[i] = promedio;
shmptr->lta[i] = plargo;
if((plargo * shmptr->umbral) < promedio)

```

```

        shmptr->cola_resultados[shmptr->atras_resultados] |= j;
    }/* if(promedio=...*/
    }//if(estaciones ..
    j <= 1;
} /*for(i=0... */

resultado = 0;
for(i=0; i<EPOCAS; i++)
    resultado |= shmptr->cola_resultados[i];

evento = FALSE;
i=0;
while((shmptr->funcion_deteccion[i] != 0) && (i < ESTACIONES) &&
(!evento))
{
    evento = (shmptr->funcion_deteccion[i] ==
        (shmptr->funcion_deteccion[i] & resultado)) ? TRUE :
FALSE;
    i++;
}

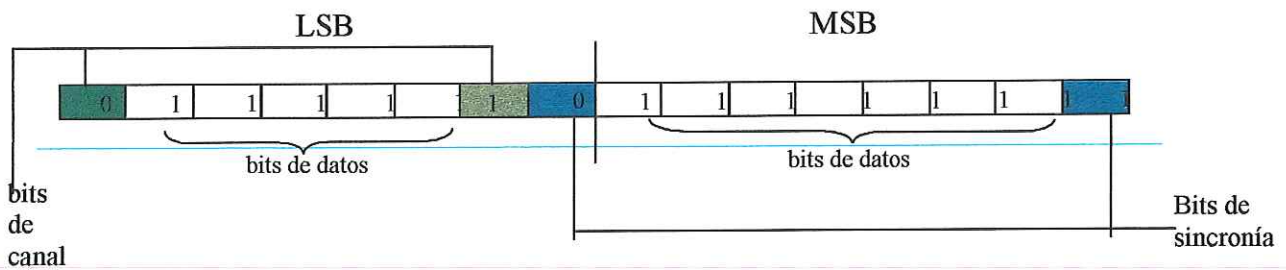
return(evento);

```

Para la conversión del formato propietario de resnom hacia el formato de forma de onda de Seisan, se puede consultar el manual de Seisan (referencia), a partir de la página 221, en el apéndice 2 que explica la forma en que se deben de escribir los encabezados de los archivos y como se deben de escribir los valores de la forma de onda.

Apéndice C. Conversión del formato de transmisión hacia el formato RESNOM

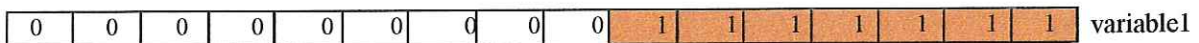
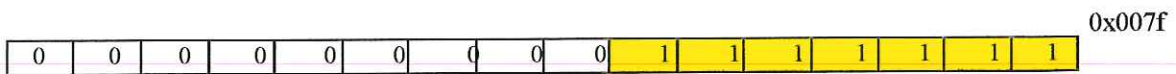
A continuación se va a presentar una explicación gráfica del proceso que se aplica a cada muestra que arriba al sistema para cambiarla al formato RESNOM. Empezamos con una muestra en el formato de transmisión en que llega a la computadora:



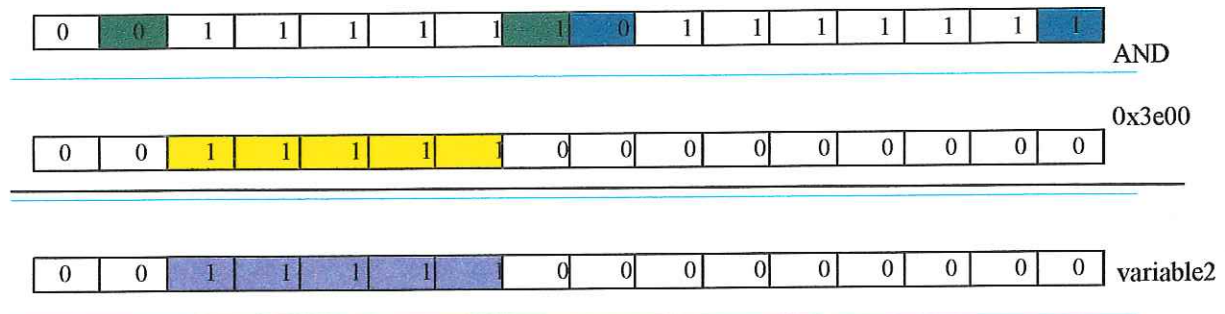
1. Se recorre la muestra un bit hacia la derecha para eliminar uno de los bits de sincronización:



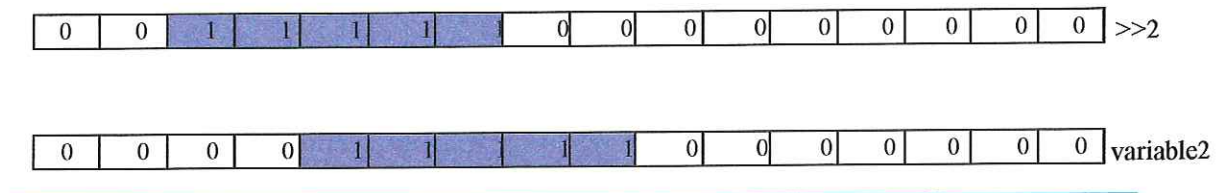
2. Se aplica dos operaciones AND lógicas para obtener solamente los bits de datos de la muestra en dos variables. Para obtener la primer parte de la muestra hacemos la primera operación lógica:



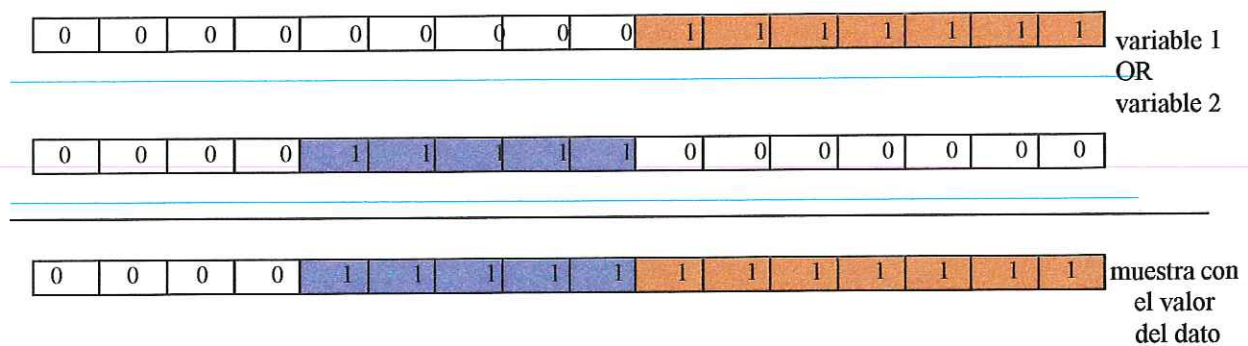
donde 0x007f es una máscara binaria que usamos para tomar sólo los bits que requerimos. Como se ilustra a continuación la segunda operación lógica permite obtener la segunda parte de la muestra:



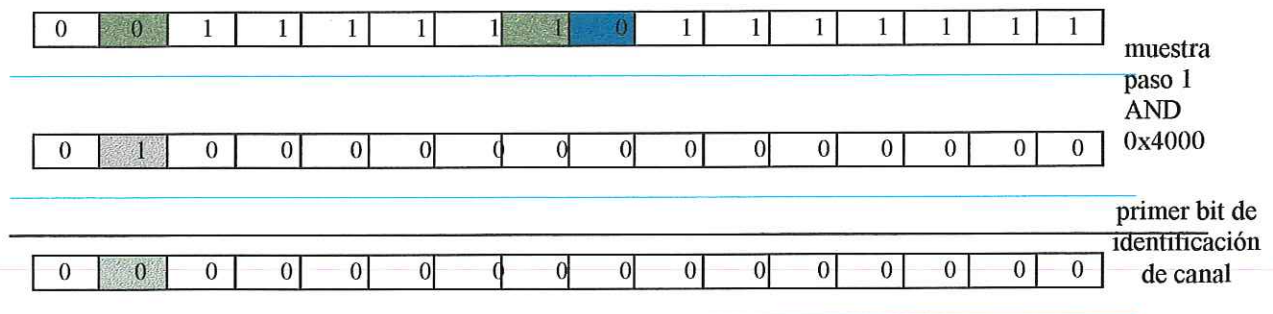
Donde 0x3e00 representa la máscara utilizada en esta operación. A la variable 2 hay que recorrerla 2 lugares hacia la derecha, de manera que al tiempo de juntarla con la variable 1, se tengan los 12 bits del dato.



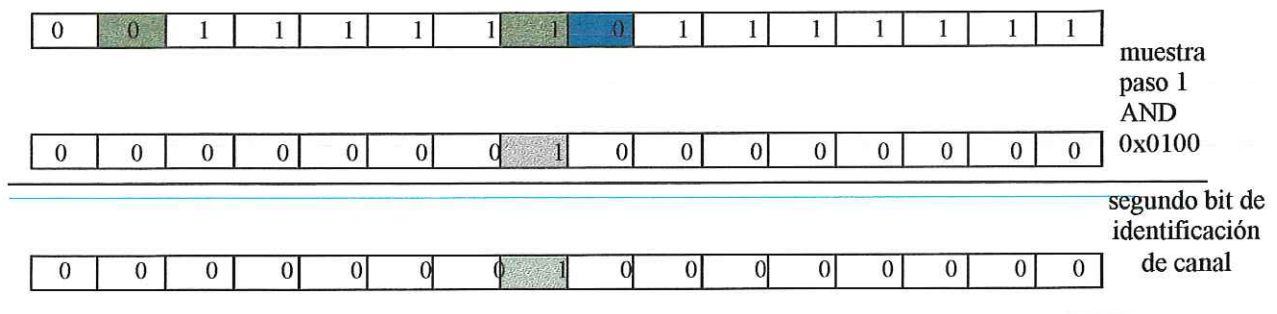
3. Finalmente se unen las dos variables mediante una operación OR lógica, para así obtener el valor de la muestra:



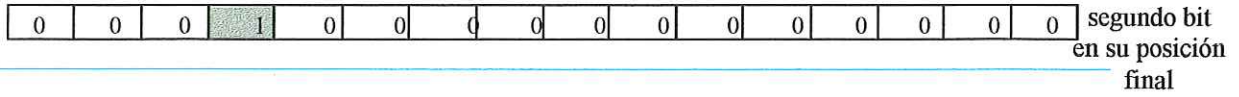
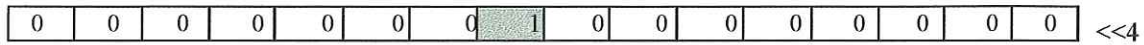
4. Para obtener el canal al que pertenece la muestra, realizamos las siguientes operaciones: primero se toma el valor del canal de la muestra que queda después de aplicar el paso 1. Mediante una operación AND similar a la realizada para obtener la primera mitad de la muestra se obtiene el primer bit del canal.



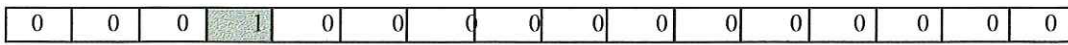
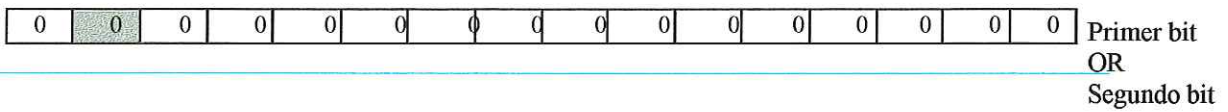
Donde 0x4000 es la máscara usada para esta operación. Para obtener el segundo bit de identificación de canal se realiza otra vez la operación AND pero ahora en la posición que le corresponde al segundo bit.



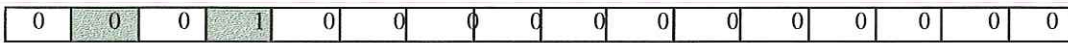
0x0100 es la máscara binaria usada en este caso. Una vez que obtuvimos el segundo bit de identificación de canal, lo recorremos 4 lugares hacia la izquierda. Este es el lugar que le corresponde en el formato final.



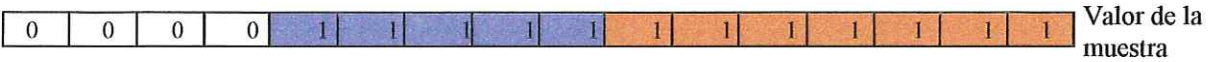
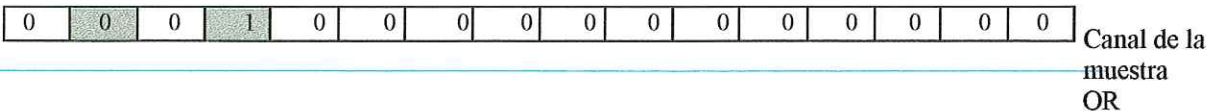
5. Ya que se extrajeron los bits para identificar el canal, ahora se unen en una sola variable usando una operación OR.



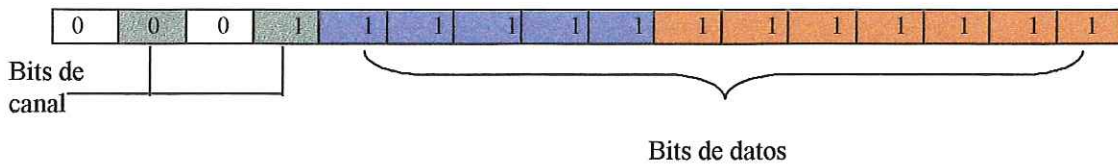
Canal de la
muestra



6. Finalmente, se une el dato de la muestra con el canal a la que pertenece, también mediante el uso de una operación OR

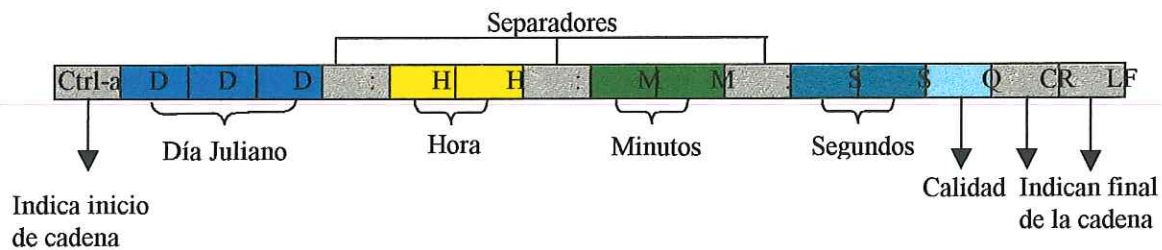


Muestra final
en formato
RESNOM



Apéndice D Conversión de la cadena de tiempo en ASCII a formato binario

En este apéndice se va a mostrar una explicación gráfica del proceso que se le aplica a la cadena que se recibe con el tiempo, para poder guardarla junto con la señal que se recibe de las estaciones sísmicas. Al principio se recibe como entrada una cadena de 16 caracteres ASCII, los cuales contienen información del día juliano y la hora. Esta cadena trae además indicadores de inicio y de finalización. En el siguiente esquema se muestra dicha cadena.



1. Se toman únicamente los valores numéricos y el caracter de la calidad de la señal, se convierten a su valor numérico y se guardan en unos enteros cortos.

0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	Día juliano
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-------------

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	Hora
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	------

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	Minutos
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---------

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	Segundos
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----------

Para guardar el valor de la calidad de la señal se tiene que asignar el valor que le corresponda al carácter de acuerdo con la Tabla VI

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Calidad

2. Se recorre el valor del día juliano seis posiciones hacia la izquierda

0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<<6

Y se aplica una operación OR lógica a este valor y al valor de la hora para juntarlos en un entero corto

0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Día juliano

OR

0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Hora

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Se enciende el primer bit para indicar que esto es señal de tiempo, y de esta manera obtenemos la primera marca de tiempo.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Primera marca de tiempo

3. Para la segunda marca de tiempo se recorre el valor de la calidad doce posiciones hacia la izquierda.

0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<<12

el valor de los minutos se recorre seis posiciones hacia la izquierda.

0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

<<<6

0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Calidad

OR

0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Minutos

0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

4. Se hace otra operación similar para unir los datos anteriores con el valor de los segundos.

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Segundos

OR

0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Minutos +
Calidad

0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Segunda marc:
tiempo

el siguiente paso consiste en señalar esta muestra como marca de tiempo, para hacer esto encendemos el primer bit de ésta.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Ahora ya se tienen las dos marcas de tiempo que se estampan a la señal que arriba a los puertos seriales. Éstas se insertan cada segundo para el caso de las señales de periodo corto y cada 60 segundos para el periodo largo.