



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**“SISTEMA DE INFORMACIÓN PARA UNA RED
SOLAROMÉTRICA”**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

IVÁN RODRÍGUEZ RASILLA

D I R E C T O R D E T E S I S

DR. MAURO GERMÁN VALDÉS BARRÓN





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dedicatorias.

A Dios por darme la fortaleza, esperanza y fe en él.

A mi madre por su incondicional amor, por su incansable trabajo para ofrecerme una mejor vida, a su entrega y a su extenso apoyo durante todo este tiempo.

A mi hermano por su compañía, sus consejos y por su desmedida confianza en mí.

A mi hija, quien se convirtió en uno de los principales pilares de mi vida y aliciente para seguir esforzándome cada día más como mejor padre y persona.

Agradecimientos.

A la Universidad Nacional Autónoma de México por todo lo que me ha ofrecido esta honorable y magnífica Institución.

Al Dr. Mauro Valdés Barrón por haberme permitido colaborar en su proyecto y dirigir este trabajo.

Al Ing. Silvestre Ruiz Salinas por su apoyo y asesorías para realizar esta tesis en la parte de Ingeniería de Software.

Gracias a Jesús A. Pérez Santana por su apoyo durante la realización de este trabajo, a Luis Edgar Abreu Rodríguez por su colaboración en las correcciones finales, así como también a Sara Ivonne Franco Sánchez, Caridad Cárdenas Monroy y Aida Quezada Reyes por su ayuda, sugerencias y ánimo en la terminación de esta tesis.

A mí jurado de titulación, M.I. Aurelio Adolfo Millán Nájera, al M.C. Tan Li Yi, al Ing. Alejandro Velázquez Mena y al Ing. Luis Arenas Hernández por sus comentarios y contribuciones hechas para mejorar este trabajo.

A Karla Medina V., gracias porque de alguna manera me ayudaste a ver la vida de forma diferente.

A Carlos Márquez P., Pedro Jiménez N. y Luis A. Ramírez G. por su sincera amistad y descomunal apoyo durante todos estos años.

Agradezco a Héctor Tecanhuey S., Miguel A. García P., por su cálida amistad y por haber compartido sus experiencias que fueron útiles para comenzar este trabajo.

Índice.

Índice.	3
Índice de figuras.	5
Introducción.	8
Capítulo I. La Radiación Solar.	10
I.1 Definición.	10
I.2 Importancia de la medición de la Radiación Solar.	11
I.3 Medición de la radiación solar.	13
Capítulo II. Conceptos Generales.	17
II.1 Visión genérica de ingeniería de software.	17
II.2 Características de un buen sistema de software.	18
II.3 Definición de términos.	19
II.4 Proceso de Desarrollo.	20
II.4.1 Fase de definición.	21
II.4.2 Fase de desarrollo.	23
II.4.3 Fase de mantenimiento.	24
II.5 Sistema de Adquisición de datos.	26
II.5.1 Transductores.	26
II.5.2 Transductores eléctricos para medición.	27
II.5.3 Estructura general de un sistema de adquisición de datos.	28
II.6 Interfaz RS-232.	29
II.6.1. Características de la comunicación RS-232.	31
Capítulo III. Visión y ámbito del proyecto.	35
III.1 Introducción.	35
III.2 Antecedentes de operación.	36
III.3 Necesidades del Observatorio de Radiación Solar.	37
III.4 Análisis de requerimientos.	38

Capítulo IV. Análisis.	56
IV.1 Modelo de Arquitectura.	56
IV.2 Diagrama de clases.	57
IV.3 Diagramas de secuencias.	60
IV.4 Diagramas de actividad.	64
Capítulo V. Diseño.	68
V.1 Requerimientos en software y hardware.	68
V.2 Secuencia de pantallas.	70
Capítulo VI. Pruebas.	77
VI.1 Pruebas a la etapa de Adquisición.	77
VI.2 Pruebas a la etapa de Transmisión.	80
VI.3 Pruebas en la etapa de Recepción.	82
Conclusiones.	84
Apéndice A.	87
A.1 UML.	87
A.2 Diagrama de casos de uso.	87
A.3 Diagrama de clases.	88
A.4 Diagrama de secuencias.	89
A.5 Diagrama de actividades.	90
Glosario.	93
Referencias.	95

Índice de figuras.

Capítulo I

Figura I.3.1 Esquema de la constitución básica de un termopar para radiación solar.	14
Figura I.3.2 Diagrama de una termopila.	15
Figura I.3.3 La Radiación global (RG) es igual a la suma de $R_d + R_D$	16

Capítulo II

Figura II.5.1 Diagrama general de un sistema de adquisición de datos.	28
--	----

Capítulo III

Figura III.4.1.1 Operaciones realizadas en el sitio de recepción remota.	42
Figura III.4.1.2 Operaciones realizadas del lado del servidor.	42
Figura III.4.1.3 Casos de uso del control de adquisición de datos.	46
Figura III.4.1.4 Casos de uso para la comunicación por medio de la interfaz RS-232 entre la computadora y el HP39470A y el almacenamiento en disco duro de las mediciones.	47
Figura III.4.1.5 Diagrama de caso de uso para la etapa de transmisión remota de mediciones.	50
Figura III.4.1.6 Casos de uso para la recepción de datos del lado del servidor.	53

Capítulo IV

Figura IV.1.1 Modelo de dos capas.	56
Figura IV.2.1 Diagrama de clases para el modulo recepción de datos.	57
Figura IV.2.2 Diagrama de clases para el modulo de adquisición.	58
Figura IV.2.3 Diagrama de clases para el modulo de transmisión.	59
Figura IV.3.1 Diagrama de secuencias para el caso de uso Configura aparato de adquisición.	60
Figura IV.3.2 Diagrama de secuencias para el caso de uso Inicia proceso.	61
Figura IV.3.3 Diagrama de secuencias para el caso de uso Verifica existencia de datos.	62
Figura IV.3.4 Diagrama de secuencias para el caso de uso Transmisión.	63

Figura IV.3.5 Diagrama de secuencias para el caso de uso Controla recepción de datos remotos.....	64
Figura IV.4.1 Diagrama de actividad para el proceso de adquisición.....	65
Figura IV.4.2 Diagrama de actividad para el proceso de transmisión.	66
Figura IV.4.3 Diagrama de actividad para el proceso de recepción.	67

Capítulo V

Figura V.2.1 Pantalla de bienvenida del programa.....	71
Figura V.2.2 Salida en pantalla proporcionado por el comando datademon config.	71
Figura V.2.3 Salida en pantalla del comando datademon start, que muestra la lectura de datos provenientes del HP34970A.	72
Figura V.2.4 Salida en pantalla, después de haber ejecutado el comando raddemon, para la transmisión de datos.....	72
Figura V.2.5 Presentación de las mediciones el cual incluye el nombre de la estación fecha y hora de las lecturas de los canales de medición.	73
Figura V.2.6 Pantalla que muestra el envió de datos hacía el servidor.....	73
Figura V.2.7 Salida en pantalla del comando radcom cuando no se le especifica el nombre de la estación a monitorear.	74
Figura V.2.8 Salida en pantalla del comando radcom cuando no encuentra el archivo de configuración de la estación.	74
Figura V.2.9 Pantalla que muestra la recepción de datos en el servidor de una estación remota.....	75
Figura V.2.10 Listado del directorio donde se están almacenando los datos que se reciben de la estación remota.	75
Figura V.2.11 Contenido del archivo de las mediciones obtenidas en la estación remota.	76

Capítulo VI

Figura VI.1.1 Contenido del archivo mediciones presentando únicamente las lecturas de los canales de medición.	78
---	----

Figura VI.1.2 Contenido del archivo de lectura de mediciones después de haberle agregado el nombre de la estación, fecha y hora de medición.	78
Figura VI.1.3 Contenido final del archivo de mediciones.....	79

Figura VI.2.1 Contenido del archivo de mediciones recibido en el servidor, en el cual se muestra que los datos no llegan completamente.	80
--	----

Apéndice

Figura A.2.1 Diagrama de un caso de uso	88
Figura A.3.1 Diagrama de clases	89
Figura A.4.1 Diagrama de secuencia	90
Figura A.5.1 Barra de sincronización.....	91
Figura A.5.2 Bifurcación	91
Figura A.5.3 Representación de un diagrama de actividades con calles	92
Figura A.5.4 Inicio, fin y transición de un diagrama de actividad.....	92

Introducción.

Actualmente en nuestro país se tienen algunas estaciones de monitoreo de radiación solar, pero estas estaciones se encuentran limitadas en cuanto a la disponibilidad de los datos, pues en la mayoría de los sitios hay que trasladarse hacia ellos y obtenerlos de manera manual. Esto ocasiona que la información radiacional esté sujeta a periodos muy largos de tiempo para ser procesada y analizada además de que se incurre en un gasto económico y tiempo en los trayectos para la obtención de los datos.

Existen algunos sistemas los cuales permiten obtener la información de los instrumentos de medición radiacional pero surge el problema cuando éstos incurren en grandes costos para su adquisición, comenzando desde la compra de programas para manejar el instrumento de medición y procedimientos para el envío de los datos. Debido a esto el presente trabajo de tesis se enfoca en resolver esta problemática con la hipótesis de que con ayuda del software libre, el uso gratuito de herramientas de programación y con el apoyo del Internet se puede construir una Red Solarimétrica con un costo relativamente bajo y cuyos resultados son bastante confiables. En particular este trabajo se desarrolló para ser implementado en el Observatorio de Radiación Solar del Instituto de Geofísica de la Universidad Nacional Autónoma de México. Este tema de tesis está conformado de la siguiente manera:

- En el primer capítulo se describe de manera general el significado de lo que es la radiación solar, su uso e importancia y la forma en que se mide.
- En el segundo capítulo se presenta un breve marco teórico de la Ingeniería en Software, también se presenta una descripción breve de lo que es un sistema de adquisición de datos, que elementos la conforman y por último una explicación concisa de cómo está definido la comunicación RS232 que utiliza el puerto serie de 9 pines.
- El tercer capítulo nos explica el proceso de operación del Observatorio de Radiación Solar, cuáles son sus antecedentes, sus requerimientos y las necesidades del observatorio. Aquí se realiza el análisis del alcance del sistema y como va a operar.

- En el cuarto capítulo se realiza el análisis del sistema, definiendo la arquitectura del sistema, describiendo su estructura y operación mediante diagramas de clase, diagramas de secuencias y diagramas de actividades.
- El quinto capítulo se enfoca al diseño del sistema, que elementos tanto en software y hardware van hacer incluidos, además de ir describiendo algunas de las funciones de los módulos en el cual se dividió el sistema presentando algunas pantallas.
- Posteriormente en el capítulo seis, el contenido describe algunas de las pruebas más importantes a las que se sometió al sistema y como se fueron corrigiendo los errores originados por estas pruebas.
- Como último punto se presentan las conclusiones de este trabajo cuales fueron los resultados y la bibliografía utilizada para el desarrollo de esta tesis.

Es muy probable que este trabajo pueda seguir en desarrollo de actualización o mejor dicho de mantenimiento, esto debido en parte a la modularidad que presenta y la flexibilidad que este mantiene para integrar posibles cambios, originándose varias versiones del sistema, todo esto con el fin de incorporar nuevas soluciones a las necesidades que se vayan ofreciendo.

Capítulo I. La Radiación Solar.

I.1 Definición.

Se le llama radiación solar a la energía emitida por el sol en forma de radiación electromagnética en un intervalo de longitudes de onda que van desde las micro-ondas, hasta las longitudes de onda de varios kilómetros, de toda esta gama de frecuencias, las más importante para la vida en nuestro planeta es la energía comprendida entre los 0.28 μm a los 3.5 μm pues es en estas longitudes de onda, en donde se concentra la mayor parte de la energía que llega a la superficie del planeta.

El espectro solar puede dividirse en tres grandes intervalos: Ultravioleta (0.28 a 0.39 μm), el visible (0.4 a 0.77 μm) y el cercano infrarrojo (0.78 a 3.5 μm), los cuales contienen el 9, 40 y 51 % de la energía total respectivamente.

Aunque el sol emite energía al espacio de una manera constante (constante solar), a la Tierra llegan 1367 W/m² con una variación de ± 1.6 % (Iqbal, 1983), variación ocasionada básicamente por la excentricidad de la órbita terrestre, de tal manera que cuando para el hemisferio norte es verano, llega el mínimo de radiación a la tierra proveniente del sol y durante el invierno para este mismo hemisferio llega el máximo, y aunque esta radiación es muy constante, otro factor astronómico como la inclinación del eje terrestre ocasiona las diferentes estaciones del año, pero la mayor parte de la variedad en la distribución de la radiación solar en superficie, es debida básicamente a la composición de la atmósfera terrestre, que podemos agrupar en 5 sistemas:

- a. Ozono.
- b. Gases uniformemente diluidos.
- c. Atmósfera (H, O, He, etc.).
- d. Agua precipitable.
- e. Aerosoles.

Aunque ya a la altura de 40-50 km, el ozono absorbe con gran eficiencia la radiación ultravioleta en su banda "C" (menor a 0.28 μm), es la capa de los primeros 30 km desde la superficie, en donde se llevan a cabo las transformaciones energéticas más importantes de la radiación solar.

Puede considerarse que en dicha capa se encuentra casi la totalidad de los gases capaces de absorber la radiación solar, fundamentalmente estos gases son: el vapor de agua, el bióxido de carbono (CO_2) y el ozono (O_3); contiene, asimismo, el 99 % de la masa atmosférica, esto es que a éstas después de esta altitud, el aire ya no es tan denso como en los primeros 30 km, en los cuales se llevan a cabo los procesos básicos de atenuación de la radiación solar: la dispersión de la radiación por el aire y la extinción de la radiación solar por las partículas suspendidas (también conocido como aerosol atmosférico), capaces de absorber, dispersar y reflejar la radiación solar. Finalmente, en dicha capa tiene lugar la formación de los sistemas nubosos, los cuales son el principal modulador de la radiación solar en superficie.

La acción de todos estos factores generan tres tipos de flujos radiacionales en la atmósfera (ver figura 1.3.3), la Radiación Directa (RD), es decir, la que llega (atenuada por dispersión y absorción de gases y partículas), al punto en la dirección sol-observador; la Radiación Difusa (Rd), la que es dispersada por gases, partículas y nubes, hacia el punto de observación; y la suma de ambas, la Radiación Global (RG)

En la variabilidad de los flujos radiacionales en la atmósfera además de influir los factores astronómicos y atmosféricos, influyen los geográficos propiamente dichos, como es la altitud, pues mientras más alto este el sitio de observación, más radiación solar recibirá, pues los rayos solares, recorrerán una menor trayectoria en la atmósfera y por consiguiente estarán menos expuestos a procesos de absorción, dispersión y reflexión.

I.2 Importancia de la medición de la Radiación Solar.

Como ya mencionamos, la radiación solar sufre en los primeros 30 Km de la atmósfera desde la superficie de la tierra, los cambios más significativos, tanto cualitativamente como cuantitativamente, los primeros debido a que los diferentes gases y partículas suspendidas, interactúan con la radiación solar en diferentes longitudes de onda, por lo que los procesos de extinción de la radiación solar es selectiva, y la segunda es que a la superficie de la tierra no llega toda la radiación solar que penetra en un principio la atmósfera.

Lo anterior ocasiona que el régimen solar varíe en la superficie terrestre y a lo largo del año, estas variaciones serán fundamentales ocasionadas por la presencia e intensidad de los elementos climáticos y por ende determinarán el clima del mundo, así que para estudiar el clima es necesario también evaluar la radiación solar en superficie, acción que toma mayor relevancia cuando hablamos de la tendencia que tiene la elevación de la temperatura en el mundo, pues si se toma en cuenta que esta elevación y causa del Cambio Global del Clima es la acumulación de energía en la atmósfera debida al rompimiento del balance natural (relación entre energía entrante y energía saliente), deberemos entonces estudiar la radiación solar y su comportamiento en la atmósfera.

Dentro de los grandes problemas que sufre la Zona Metropolitana de la Ciudad de México, se encuentra la contaminación ambiental, teniendo al Ozono como su principal responsable, pero da la casualidad que el Ozono no es emitido por ninguno de las fuentes contaminantes, sino que es un contaminante secundario, esto quiere decir que a partir de algún proceso los contaminantes primarios son transformados y algunos de ellos forman el Ozono, este proceso en el caso del Ozono es una reacción fotoquímica que se lleva a cabo con la Radiación Solar Ultravioleta en su Banda "A".

Cualquier modelo matemático que quiera explicar el comportamiento de algún elemento atmosférico, la conjugación de varios o el comportamiento climático global, tendrá que tener en cuenta el régimen radiacional, ya que es este, el que alimentará de energía el sistema a estudiar, si no sabemos de qué energía se dispone, no se podrá conocer cuál será su comportamiento en corto, mediano o largo plazo.

Las diferentes estructuras en las que habitamos cotidianamente, deben de alcanzar el "índice de confort", que no es más que llegar a temperaturas agradables en donde sus habitantes desarrollen lo más a gusto posible sus actividades cotidianas, pero el reto no es ese, sino el de alcanzar estos índices con el menor gasto energético, y solamente se logra diseñando estructuras lo más acorde posibles al medio que les rodea, pero sobre todo acorde al régimen radiacional, pues es esta energía la disponible en el medio para lograr el objetivo principal.

La fotosíntesis es un proceso que se lleva a cabo teniendo como energía principal la radiación solar, por lo que para realizar estudios sobre diferentes temas relacionados

con las plantas (silvicultura, agricultura, etc.), es necesario primeramente evaluar el recurso solar.

Para el aprovechamiento racional de los recursos naturales primeramente se necesita llevar a cabo una evaluación espacial y temporal de éstos, la radiación solar no es una excepción, pues para su óptimo aprovechamiento es necesario saber cuánta y cómo llega a la superficie terrestre, ya que los equipos para su aprovechamiento cuentan con una característica llamada “eficiencia”, ésta, es la relación existente entre la energía que le llega y su capacidad de transformarla en energía eléctrica o en calor, si no conocemos el régimen radiacional, se corre el riesgo de utilizar un equipo que tenga una eficiencia menor al régimen de radiación, y el resultado será que la tecnología no sirve y si por el contrario se compra un equipo con una eficiencia sobrada, lo más seguro es que el costo será elevado.

Se puede continuar enumerando una gran cantidad de razones por las cuales es necesario evaluar la radiación solar en superficie, pero tal vez baste en último término con recordar que la energía solar es la fuente de vida en nuestro planeta, es la fuente de energía más importante que existe, pues si bien en este momento se utilizan los energéticos fósiles, sería conveniente recordar que éstos son de alguna manera, reservorios de energía solar, pues el carbón mineral en algún momento fueron plantas que se alimentaron con radiación solar y el petróleo, animales que en algún punto de su cadena alimenticia se encuentran las plantas.

I.3 Medición de la radiación solar.

Para cualquier estudio que implique la utilización de la radiación solar, o para el aprovechamiento de ésta, es necesario cuantificarla. Para medir la radiación solar existen sensores que se clasifican como térmicos, termomecánicos o fotoeléctricos, de acuerdo con el principio físico de medición sobre el cual se basa cada uno de ellos (Kondratyev 1983).

Una condición importante, para estos sensores es que el proceso de medición no debe de ser ni complejo, ni demasiado sensible a la influencia del medio ambiente, sobre

todo cuando se trata de sensores que deben de funcionar continuamente bajo condiciones climatológicas variables y severas.

A través de la práctica se ha visto que los sensores termoelectricos son los que presentan mayores ventajas; su funcionamiento no es complicado y los modelos más recientes presentan una gran estabilidad. El elemento básico de estos sensores es el termopar; el sensor propiamente dicho es la termopila, que está constituida por termopares conectados en serie, de tal manera que la fuerza electromotriz total es igual a la suma de las fuerzas electromotrices de cada uno de los elementos. Los termopares se agrupan en la termopila de manera que existan áreas de “juntas frías” y áreas de “juntas calientes”. La diferencia de temperatura entre ambas genera una diferencia de potencial, susceptible de ser medida y proporcional a la energía que la ocasiona (Leyva et al., 1984), en la figura 1.3.1 se puede observar un esquema.

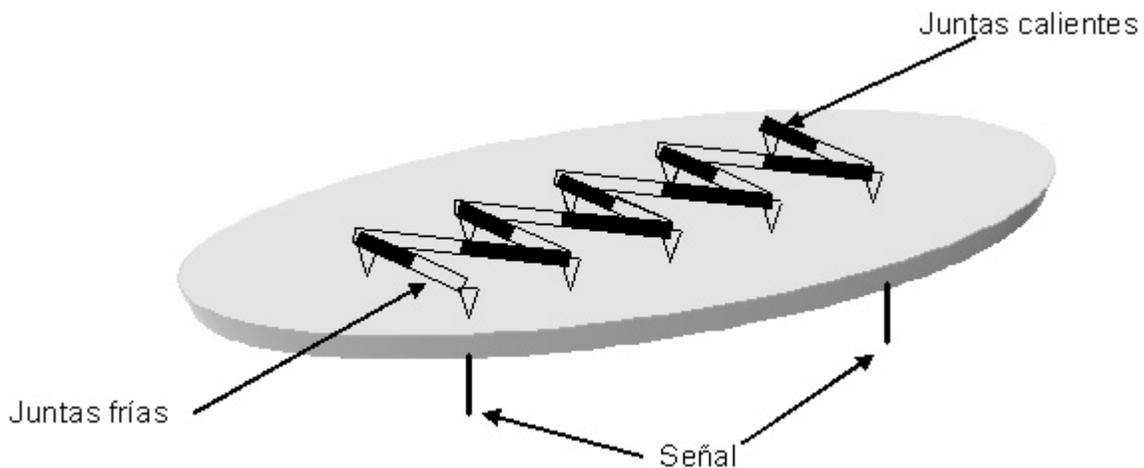


Figura I.3.1 Esquema de la constitución básica de un termopar para radiación solar.

Las “áreas calientes” se obtienen aplicando en ciertas zonas de la termopila un recubrimiento que la acerque lo más posible a un cuerpo negro, es decir que adquiera una alta absortividad, mientras más absorbente sea (lo más cercano a 100%), la calidad del sensor es mayor. Las zonas frías se consiguen en la misma termopila, pintando áreas de color blanco para que éstas sean lo más reflejante posible y no absorban energía. Otra forma de conseguir el mismo efecto es conectando las “juntas frías” directamente al cuerpo del sensor y que éste funcione como un disipador de energía; en este caso las “juntas calientes”, pintadas de negro, se exponen al flujo de radiación. Esta área de la

termopila debe tener además de gran absorptividad, poca selectividad en la absorción de radiación solar y esto depende de la calidad de la pintura con que se recubre la termopila.

La termopila debe de ser plana y colocada de tal manera que pueda recibir radiación solar en todas direcciones procurando que su campo visual esté limitado al hemisferio celeste; el efecto de viento o de cualquier tipo de precipitación se evita con la cubierta de los sensores; esta cubierta la constituyen uno o dos domos concéntricos de vidrio, de transparencia espectral uniforme en una región de 0.2 a 3.5 μm (figura 1.3.2)

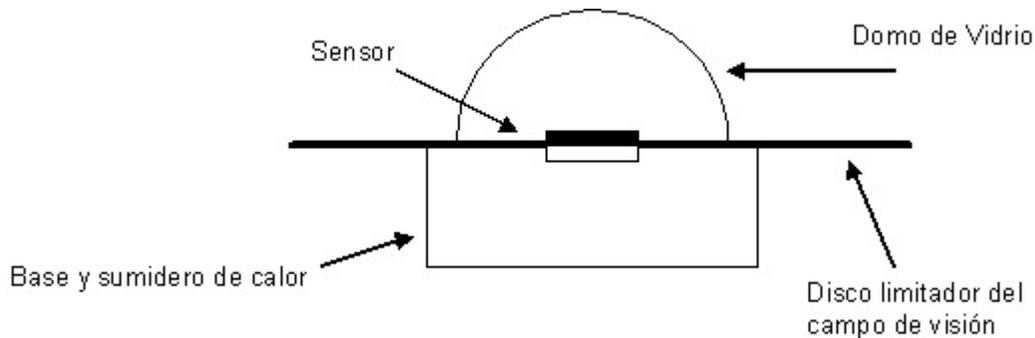


Figura 1.3.2 Diagrama de una termopila.

La termopila constituye el elemento sensible de los instrumentos para medir cualquiera de las componentes radiacionales antes mencionadas (RG, RD o Rd), a través de los instrumentos llamados Piranómetros (RG y Rd) o Pirheliómetros (RD), la diferencia entre ellos es puramente geométrica, ya que los últimos se construyen para medir exclusivamente en dirección del sol, en un ángulo sólido de 5 grados, con seguimiento manual o automático y los primeros para recibir radiación dentro de un ángulo sólido de 2π radianes (figura No. 1.3.3).

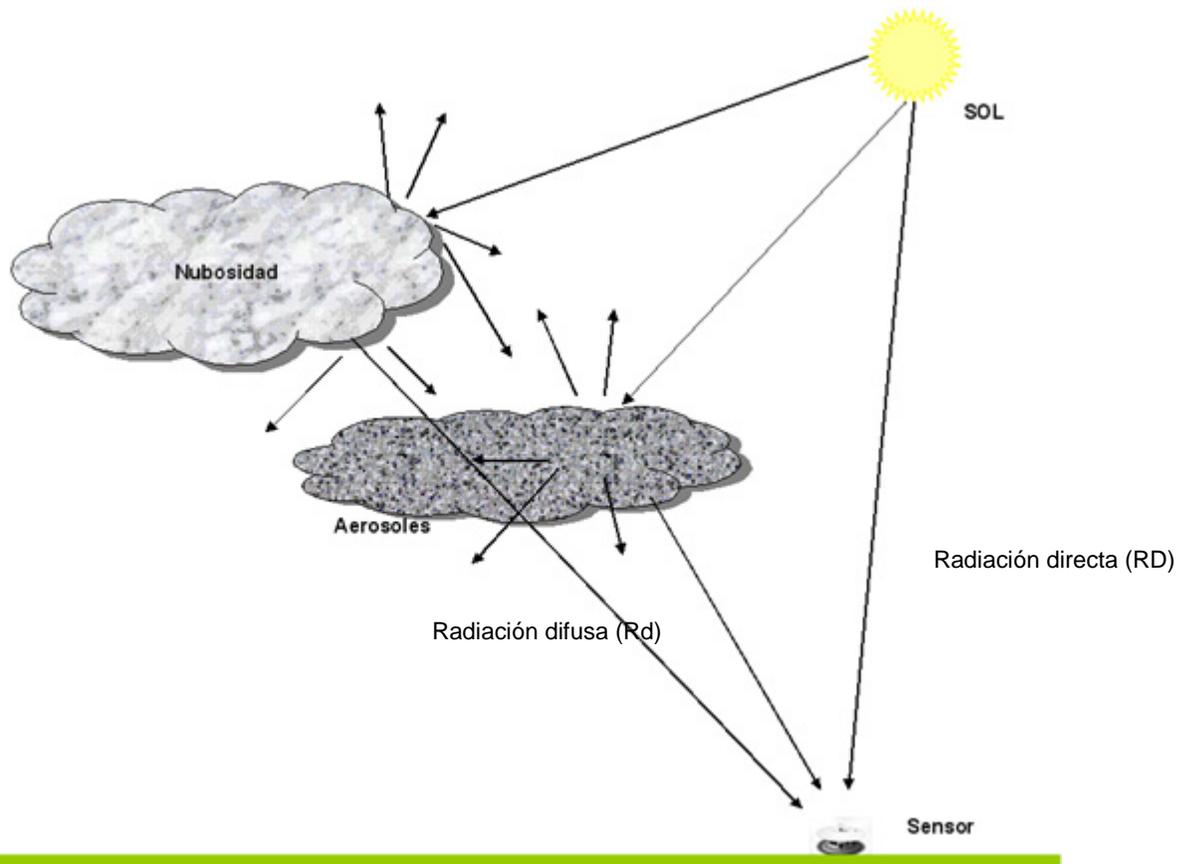


Figura I.3.3 La Radiación global (RG) es igual a la suma de $R_d + R_D$.

Para el registro de la información existen varias técnicas, casi todas ellas ya obsoletas, pues aunque aún existen piranómetros con registros termomecánicos incorporados, éstos ya no son recomendados por la Organización Meteorológica Mundial (OMM), organismo internacional que regula toda la actividad meteorológica en el mundo. Los registradores digitales son los más modernos y son aquellos en los cuales los procesos de medición o funcionamiento se lleva a cabo por medio de microprocesadores; el registro de los datos en estos sistemas es en medios magnéticos o almacenamiento digital, el alto costo de estos equipos puede verse compensado por la calidad de la información y la rapidez del procesamiento.

Capítulo II. Conceptos Generales.

II.1 Visión genérica de ingeniería de software.

Hoy en día la ingeniería de software a llegado a tomar un papel muy importante en la evolución de la computación, hasta hace algunas décadas, el objetivo principal era tener un importante desarrollo en el hardware de las computadoras, de forma que saliera más económico el procesamiento y almacenamiento de datos, con avances significativos en la reducción de los costos del hardware, tan solo en los 70's no se podía imaginar en almacenar más de 8 GB en una memoria de dimensiones de una tercera parte de una tarjeta de crédito y que esta memoria no necesitaría tener fuente externa de energía para poder mantener almacenados los datos, y que su velocidad de transmisión alcanzaría hasta los 480 Mega bits por segundo; o los procesadores que llegarían a trabajar a grandes velocidades de procesamiento con un bajo consumo de energía y que su adquisición está al alcance de la mayoría de las personas; al contar con este extraordinario avance en la microelectrónica se ha permitido tener mayor fuerza en las operaciones de cálculo.

Debido a esto, es preciso y desafiante el contar con soluciones basadas en computadora (software) a bajo costo y de calidad, que permitan aprovechar este importante avance que ha tenido el hardware, donde estas soluciones puedan satisfacer en su totalidad, las necesidades que se originaron a partir de un problema dado. El software es una potente herramienta que ayudará a simplificar en gran medida el trabajo del ser humano, aplicado a diferentes áreas del conocimiento aprovechando el potencial del avance tecnológico del hardware.

Observando la importancia que tiene el diseñar algún tipo software y dependiendo de la complejidad del problema a resolver, se debe contar con una disciplina de Ingeniería específica para el desarrollo de software, he aquí dos definiciones que parecen dar un panorama bastante claro de lo que es la ingeniería de software.

La ingeniería de software es la aplicación del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software (Pressman, 1993).

La ingeniería de software es una disciplina dentro del campo de la informática, que nos proporciona los mecanismos teórico y práctico para desarrollar y mantener software de calidad.

De aquí podemos ver que la ingeniería de software es una disciplina que por medio de sus métodos, herramientas y procedimientos permiten construir software que cumpla con las necesidades del cliente, fundamentalmente esto es construir software de calidad.

II.2 Características de un buen sistema de software.

Al hablar de ingeniería de software refiriéndose a su concepto, su función o su objetivo vemos que se menciona las palabras *software de calidad*, en diversas fuentes relacionados al tema, inclusive en el apartado anterior hacemos alusión a estas palabras, pero siendo más concisos ¿qué características o cualidades debe de tener un software que se cataloga como bueno o de calidad?, pues bien, a continuación se presenta una lista de características que dan respuesta a la pregunta anteriormente realizada.

- ✓ *Útil y eficiente:* Con base al mejor aprovechamiento de recursos en hardware y de código requeridos por un software, debe de proporcionar mecanismos y funcionalidades que le permitan al usuario facilitar o mejorar sus actividades.
- ✓ *Fiable:* Asegurar un mínimo de errores cuyo impacto no sea considerablemente tan dramático. Se debe de esperar que las funciones que realice el software se efectúen con la precisión requerida.
- ✓ *Flexible:* A medida que transcurre el tiempo, los usuarios van ampliando su panorama de necesidades debido a cambios en su ambiente de trabajo o debido a avances tecnológicos, un buen software se adaptará a los cambios de especificación, esto debido a su simplicidad en el diseño y descentralización.
- ✓ *Seguro:* La capacidad que se tiene para controlar los accesos al software o a la información almacenada por parte de personas no autorizadas.

- ✓ *Fácil de usar*: La creación de un software es con la intención de hacer la vida más fácil a los usuarios, por lo tanto un software que resulta ser muy complicado para aprender la forma en que trabaja, en cómo se preparan los datos de entrada y la forma de interpretar la salida cae en el desagrado de los usuarios.
- ✓ *Disponible*. Aquí se pueden considerar dos aspectos de disponibilidad.
 - ✓ El software, al ser transferido de un entorno en hardware y/o sistema operativo hacia otro, debe de realizarse con el mínimo esfuerzo, es decir debe ser lo suficientemente portable.
 - ✓ En un proyecto de software debe de completarse con éxito y en los tiempos de entrega prometido.
- ✓ *Reusable*: Puede existir que componentes del software, pueden ser reutilizados en otras aplicaciones, esto va relacionado con el empaquetamiento y la modularidad.
- ✓ *Facilidad de integración*: Nos referimos a esta parte a la capacidad que tiene el software de acoplarse con otros sistemas.

II.3 Definición de términos.

En varias ocasiones los términos *proceso*, *metodología*, *método* (en el ámbito de la ingeniería de software) con o sin los prefijos desarrollo o diseño y también ciclo de vida, suelen ser utilizados por muchas personas sin tener a menudo una definición que permita tener una idea clara de lo que significa una de otra, en muchos casos llegan a originar cierta confusión e inclusive muchos autores suelen utilizar estos términos indistintamente. Por tal motivo es importante conocer a que se refiere cada uno de estos términos.

El *proceso de desarrollo* o también se le puede llamar como *ciclo de vida del software*, hace referencia al conjunto de pasos o reglas que establecen como se debe llevar a cabo el desarrollo de un proyecto desde la concepción del problema hasta la entrega del sistema, en el cual se puede incluir qué documentación, modelos de análisis, modelos de diseño, y qué otros instrumentos deberían realizarse y en qué orden.

Para los términos *metodología* y *método* en la mayoría de los casos se utilizan de manera similar, pero realmente existe una diferencia entre ambos tal y como lo sugiere la longitud de las palabras; metodología se refiere al conjunto de métodos que ayudarán a cómo obtener uno o varios modelos a lo largo del desarrollo del proyecto. Como método podemos decir que son un conjunto de pasos o técnicas que forman parte de una metodología, por ejemplo, los pasos a seguir para capturar los requisitos de los usuarios o bien las técnicas para la descripción de trabajo de análisis y diseño. Ahora refiriéndonos a *metodología de desarrollo* esta nos va a proporcionar las herramientas necesarias para dar un amplio soporte de forma ordenada y sistemática en las distintas fases del desarrollo de software.

En muchas ocasiones se menciona que la metodología de desarrollo está implícita en el proceso de desarrollo, o que la metodología de desarrollo es el proceso de desarrollo, pero en base a lo que se ha escrito anteriormente podemos decir que ambos conceptos nos permiten dar un enfoque disciplinado y sistemático para desarrollar un producto de software incluyendo una colección de métodos y técnicas, ambos tienen un conjunto de componentes que especifican el *cómo* llevar el desarrollo de un proyecto de software.

II.4 Proceso de Desarrollo.

¿Cómo se construye un sistema? Tal vez es posible imaginar que la realización de un programa de computadora simple, requiera tan solo de sentarse y hacer el trabajo, empezando desde que es lo que quiero que haga, tal vez hacer un pseudocódigo, empezar a codificarlo mediante algún lenguaje de programación y por último ejecutarlo donde la última etapa será ver que realmente funcione o haga lo que se necesita. Sin embargo para sistemas que son mucho más complejos y que inclusive requiera de todo un equipo de trabajo humano, es necesario dirigir el proceso de desarrollo del sistema. Para este tipo de caso debe de ser posible planificar el desarrollo de tal manera que nos permita decir hasta donde se ha llegado y si éste va con retraso. Se debe de tener alguna forma entendible de documentar y controlar lo que se está haciendo más aun si contamos con equipo humano de desarrollo, pues se puede dar el caso de alguna persona pueda abandonar o se una al proyecto, según convenga, y que estos cambios no impliquen actos que puedan incurrir en el retraso de la entrega del proyecto, que en la vida real esto

conlleva a tener fuertes problemas económicos y de insatisfacción por parte de los clientes.

En el momento de construir un sistema de software, no es suficiente conocer el manejo de algún lenguaje de programación o tener el conocimiento de la operación de un sistema operativo, esto va más allá. Si se espera que el sistema tenga un gran alcance y que su mantenimiento sea factible, será necesario que el problema real sea llevado a un sistema de computadora, mediante una solución cuidadosamente diseñada y con un buen análisis de la problemática. Esto será posible si seguimos un proceso de desarrollo que incluya métodos, herramientas y procedimientos que se ocuparán de plantear como se realizará el análisis y diseño y como se relacionarán los resultados de ambos. Esto a su vez permitirá obtener mayor control en el desarrollo de software. Entonces la construcción de sistemas de software va a poder ser panificable e *iterativo*¹ de tal forma que será productiva y la probabilidad de obtener un sistema de alta calidad al final del proceso aumentará considerablemente.

Pressman, define que sin importar el paradigma de ingeniería de software que se utilice en el proceso de desarrollo, este último contiene de forma general tres fases las cuales son: *definición, desarrollo y mantenimiento*. Estas fases se pueden encontrar en cualquier proyecto de software sin importar a que área este aplicado este. A continuación se presenta una breve descripción de estas fases.

II.4.1 Fase de definición.

Esta fase está enfocada sobre el *qué*. Aquí es en donde el desarrollador, intenta identificar los requerimientos claves del ámbito del sistema y del software, se busca entender en su totalidad la forma en que se comporta el sistema real, identificación de sus límites, cuál es su función, qué alcance tiene, con qué otros sistemas externos a éste interactúa, en otras palabras investigar todo lo relacionado al problema. Pressman menciona que algunos métodos aplicados a esta fase varían de acuerdo al paradigma utilizado, pero éstos de alguna forma originarán tres pasos específicos:

¹ Los procesos de desarrollo que se utilizan en la actualidad tienen como elemento importante las iteraciones, y tratan de proporcionar maneras de gestionar, en vez de ignorar, los riesgos.

Análisis del sistema: Se establecen los requerimientos de todos los elementos del sistema, es decir, se identifican las necesidades del cliente, se evalúa la viabilidad del sistema de software, se realiza un análisis económico y técnico, se llegan a asignar funciones al software, al hardware, a las personas, a las bases de datos si es que se requieren, se establecen los límites de costo y tiempo y se llega a una definición del sistema de tal forma que servirá como base para el trabajo subsecuente que se llega a realizar. Como podemos ver el análisis del sistema se centra en todos los elementos del sistema no solo en el software.

Planificación del proyecto de software. Una vez que se ha determinado el ámbito del software, se procede a ver qué recursos se necesitarán, contabilizar sus costos y cuál es la agenda de actividades a seguir, cuyo objetivo será la de suministrar una estimación razonable cuantitativa de los recursos, costos y métodos sin llevarse demasiado tiempo, pero durante el progreso del proyecto estas estimaciones deben regularmente ser actualizadas.

Análisis de requerimientos. Teniendo ya un ámbito definido para la dirección del software se estudia con mayor profundidad la funcionalidad, desempeño, portabilidad etc. que tendrá la aplicación, de tal manera que con ayuda del cliente se verifique y confirme que tales funcionalidades y características cumplan las expectativas. Se realiza una clarificación de requisitos debido a que posiblemente la información que se obtiene en un principio pueda ser incierta. Toda esta información es muy útil, debido a que será utilizada por el ingeniero de software permitiendo aterrizar las necesidades del cliente dentro de un plan de software que a su vez será alimentado por la función y comportamiento del diseño de programas. El objetivo es abstraer los conceptos del sistema, reorganizarlos y sintetizarlos de tal manera que se llegue a transformar en soluciones que cumplan con los requerimientos del cliente. Para poder llegar a este objetivo se requiere llegar a comprender los entornos del usuario/cliente, aplicando el software y/o hardware a estos entornos, apoyándose en una ardua comunicación tanto escrita como verbal entre los usuarios del sistema, el cliente y el equipo técnico de desarrollo.

II.4.2 Fase de desarrollo.

Esta fase, según Pressman, se enfoca al *cómo*. Es aquí donde el ingeniero de software toma como punto de partida la documentación referente al análisis de requerimientos para comenzar el diseño del producto de software. Se realiza la construcción del sistema de software analizando el problema desde la perspectiva de los usuarios o de entidades externas que solicitarán interactuar con este mismo y el diseño de cómo va a funcionar internamente para satisfacer lo documentado en los requerimientos, especificando la forma en que ha de diseñarse las estructuras de datos y la arquitectura del sistema de manera que se pueda trasladar el diseño a un lenguaje de programación. Además se lleva a cabo una serie de pruebas para corroborar que el sistema de software elaborado opere correctamente cumpliendo con los requisitos especificados en la fase de elaboración. Pressman establece que independientemente del paradigma de ingeniería de software utilizado en esta fase, se producirán tres pasos concretos:

Diseño de software. Digamos que es la primera etapa técnica del proceso de desarrollo pues consiste en producir un modelo o representación técnica del software que se va a desarrollar tomando como punto de partida los requerimientos funcionales y de comportamiento, representándose a un alto nivel de abstracción, aquí es en donde se va a buscar y especificar la estructura y el funcionamiento del software mediante un proceso iterativo en el cual se utilizarán diversas metodologías de diseño para realizar el diseño de datos, diseño arquitectónico, diseño procedimental y diseño de la interfaz. El diseño de datos se refiere a la definición del modelo de la estructura de datos. El diseño arquitectónico define las relaciones entre los elementos de nuestro programa. El diseño procedimental transforma los elementos estructurales a una descripción de procedimientos de software en donde este último se enfocará en los detalles de procesamiento para un módulo dado. El diseño de interfaz describe la forma en que se va a comunicar el software consigo mismo y con su entorno.

Codificación. A este paso comúnmente se le suele llamar implementación o instrumentación, el cual comprende la traducción de las representaciones obtenidas en el diseño a un lenguaje de programación o bien a la elaboración del código del programa.

Prueba del software. En este paso se debe de garantizar la corrección del producto de software ya que este último haya sido puesto en marcha. Aquí se deben de crear una serie de casos de prueba que permitan descubrir los defectos que se puedan tener. Pressman menciona tres enunciados que caracterizan a una prueba:

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si se descubre un error no detectado hasta entonces.

Es importante puntualizar que a menudo se consideran validación, verificación y prueba dentro de la misma categoría. Según Stevens la validación es el proceso de asegurar que se ha construido el producto correcto, es decir, que el producto cumple su propósito, incluso puede llegar a tener cosas que no se han capturado en los requisitos establecidos. La verificación es el proceso de asegurar que se ha construido el producto correctamente, esto es, que cumple con los requisitos iniciales. La manera de poder asegurar que la validación y la verificación cumplan su cometido es probando el sistema, y la forma de poder encontrar cualquier tipo de problema en el producto es probándolo.

Por otro lado Stevens propone tres objetivos principales en los que se enfoca la prueba:

1. Ayudar a encontrar errores.
2. Convencer al cliente de que no hay errores (importantes).
3. Proporcionar información que ayudará a la evolución del sistema. Por ejemplo, el proceso de pruebas puede ayudar a recoger información sobre requisitos y prioridades futuras, así como información sobre el rendimiento actual que hasta ahora sólo ha sido estimada.

II.4.3 Fase de mantenimiento.

Teniendo como punto de partida la información obtenida en el proceso de pruebas, esta fase se centra en el cambio que va asociado con la corrección de errores, adaptaciones que se necesitan debido a evolución del entorno de software y actualizaciones originadas por los cambios en los requerimientos del cliente ya sea por

reforzar o aumentar el sistema, en otras palabras el mantenimiento va más allá de la corrección de errores y es tarea del ingeniero de software apoyándose en el proceso de desarrollo de la ingeniería de software en reducir el tiempo y esfuerzo gastado en el mantenimiento. Pressman establece que en la fase de mantenimiento se re replica los pasos de la definición y desarrollo, pero en el contexto del software existente además de que en la fase de mantenimiento se encuentran tres tipos de cambios:

Corrección. Es a veces muy difícil asegurar que al terminar un producto de software aun aplicando los mejores métodos para garantizar la calidad, no encontremos algún error en su funcionamiento y/o comportamiento del sistema, más aun cuando este último es de gran tamaño. Por eso el mantenimiento de corrección, es el proceso mediante el cual se realiza el diagnóstico y corrección de los posibles defectos que tenga el software.

Adaptación. Debido al constante cambio relacionado con el mundo de la informática y computación apareciendo nuevas generaciones de hardware donde frecuentemente hay mejoras en los equipos periféricos, dispositivos de almacenamiento y procesamiento, infraestructuras de redes entre otros elementos de sistemas y además de la aparición de nuevos sistemas operativos o en su caso nuevas versiones, protocolos de comunicación, etc., con el paso del tiempo es muy probable que el producto de software quede obsoleto para el entorno del sistema para el que fue originalmente desarrollado. De ahí que el mantenimiento de adaptación es una actividad que modifica el software para que interaccione adecuadamente con su entorno cambiante.

Aumento. Cuando el software ha tenido éxito, al transcurrir el tiempo, sucede que por el uso que se le da al paquete de software por parte de los usuarios, éstos comienzan a generar recomendaciones constructivas de manera que ayudan a tener nuevas posibilidades de modificación a sus funciones ya existentes o mejoras en general o bien surgen por parte del cliente nuevos requerimientos en cuanto a funcionalidad, rendimiento, etc., esto posiblemente debido a cambios en el área de trabajo del cliente donde el software esta funcionando, como por ejemplo, alguna evolución en su organización del cliente. Resulta importante resaltar que estos cambios en gran parte son a petición del cliente, por condiciones ajenas al proceso de desarrollo del software, por lo cual podemos decir que los nuevos requerimientos van más allá de los objetivos que inicialmente se plantearon para el sistema. Como consecuencia de esto, se requerirá un

nuevo contrato con el cliente, similar al del desarrollo original del software, ya que la intención de perfeccionar al sistema, una vez más necesitará del conocimiento y análisis de los nuevos requerimientos, rediseñar, generar código y probar el software ya terminado con las nuevas características o funcionalidades.

II.5 Sistema de Adquisición de datos.

Debido al enorme desarrollo que se ha suscitado en el campo de la electrónica, microelectrónica y computación, ha dado origen a la vertiginosa búsqueda de la aplicación de estas áreas a la solución de los problemas del entorno que envuelve al ser humano, también se ha manifestado el interés por encontrar formas de control automático en las tareas o procesos que se realizan, por ejemplo en la industria, el comercio, la salud, el hogar, la agricultura, las comunicaciones, etc., pues bien los sistemas de adquisición de datos contribuyen a llevar a cabo parte de este control automático.

La adquisición de datos toma un papel importante dentro de los procesos, ya que el adquirir datos contribuye a efectuar balances de materia y energía, calcular eficiencias y rendimientos y elaborar reportes con la información procesada.

Dentro del papel que juega un sistema de adquisición de datos es que permite revisar o captar “variables” como por ejemplo la humedad relativa, la temperatura, la iluminación, la presión, la velocidad, etc., para que posteriormente pueda ser medida, registrada y analizada para su uso en el control de algún proceso.

II.5.1 Transductores.

De forma general se puede decir que un transductor es un dispositivo mediante el cual se puede convertir energía o información determinada como entrada a otra de forma diferente en su salida. Su utilización permite realizar mediciones que difícilmente podrían ser efectuadas a simple vista, por eso es muy utilizado en el campo de las mediciones. Comúnmente se puede obtener de una forma más fácil y mejor una medición de alguna cantidad si a ésta se le puede transformar a otra que resulte mucho más sencilla de cuantificarla, o bien utilizar de forma más práctica la información que de otra manera complicaría su manejo. Por ejemplo el teclado de la computadora es un transductor

debido a que transforma la pulsación de los dedos sobre las teclas y éstas generan el código de la teclas presionadas o bien en el caso del termómetro es mucho más fácil medir los cambios de temperatura por medio de los cambios de longitud que se observan en la columna del mercurio.

II.5.2 Transductores eléctricos para medición.

Los sensores o transductores juegan un papel importante en los sistemas de adquisición de datos, pues la tarea de los transductores eléctricos es la de convertir una variable física (algunas de las magnitudes físicas que se pueden medir están: la temperatura, presión, concentración, humedad, flujo, posición, peso, iluminación, etc.) que se desea medir o registrar a una magnitud eléctrica (voltaje, corriente, resistencia, capacidad, inductancia, etc.), pues hoy en día las magnitudes eléctricas poseen muchas cualidades que son convenientes para ser utilizadas en las mediciones.

Pueden ser diversas las variables físicas que se desean medir o controlar en un determinado momento y debido a esto, existen diferentes transductores para ello; dependiendo de sus principales características como pueden ser la exactitud, fiabilidad, precisión, rango de funcionamiento, velocidad de respuesta, calibración y la forma en que estén contruidos se determina generalmente el costo del transductor a utilizar.

Para efecto de esta tesis los transductores eléctricos a utilizar serán los siguientes:

1. *Transductor de corriente:* Como su nombre lo dice, transforma una magnitud física a un nivel de corriente directa equivalente, este tipo de transductor como una de sus principales ventajas es que el nivel de corriente obtenido puede ser transportado a grandes distancias sin que se produzcan pérdidas significativas en la señal y como desventaja la corriente siempre tiene que ser convertida a voltaje antes de que esta señal llegue a un convertidor analógico-digital.
2. *Transductor de voltaje:* Este tipo de transductor entrega a su salida un nivel de voltaje equivalente a la magnitud física que se esté midiendo. Algunos transductores de este tipo están constituidos por circuitos acondicionadores, esto debido por la necesidad de acondicionar previamente la señal proporcionada por el transductor a la entrada de un convertidor analógico-digital, pues la esencia del acondicionamiento, es hacer que el rango de variación real de la señal originada por la variable física a medir, se convierta en el rango máximo de voltaje de

entrada que acepte el convertidor analógico-digital que se esté utilizando, es decir, que el valor mínimo de la variable física a medir sea transformada al valor mínimo de voltaje que acepte el convertidor y que el valor máximo de la variable a medir sea cambiada al valor máximo de voltaje que el convertidor acepte. Por otro lado a este tipo de transductores muchas veces se les incorpora resistores variables de tal forma que permita ajustar el rango de voltaje que éstos entregan al rango de voltaje que el convertidor necesita, a esto se le podría llamar como ventaja, pero a su vez su desventaja radica en que la señal proporcionada por el transductor no puede ser transmitida más allá de unos pocos metros sin que se produzcan pérdidas.

II.5.3 Estructura general de un sistema de adquisición de datos.

En el siguiente diagrama se puede observar los bloques necesarios que permiten construir un sistema de adquisición de datos de manera general.

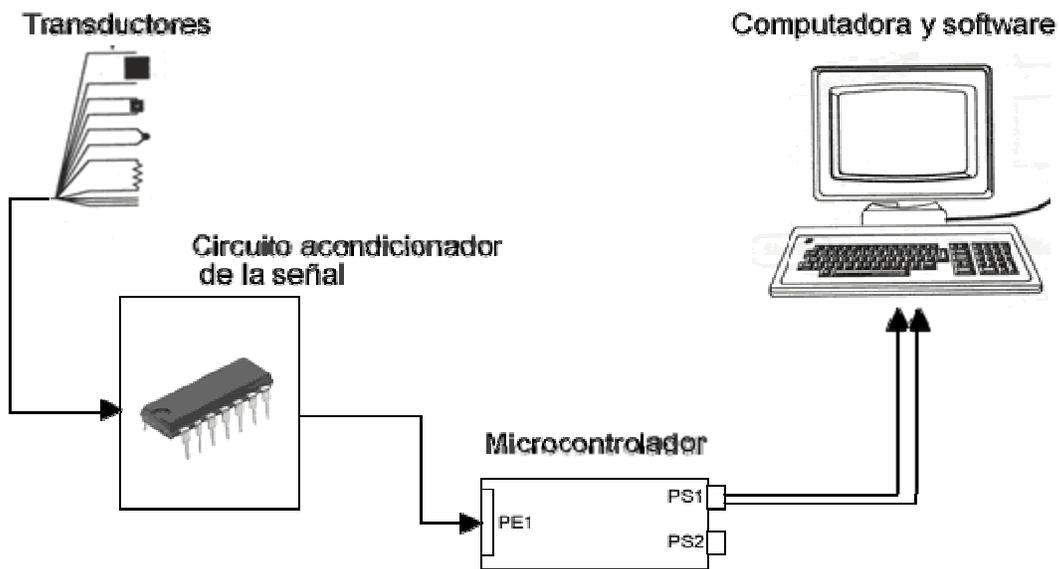


Figura II.5.1 Diagrama general de un sistema de adquisición de datos.

En primera instancia tenemos al transductor, como ya se ha visto anteriormente el propósito de este elemento es el de captar los cambios suscitados en la magnitud de la variable física que se esté midiendo. La respuesta del transductor será una señal eléctrica equivalente a la magnitud física captada.

El segundo bloque representa el circuito acondicionador cuya función será la de acondicionar la señal de salida del transductor, modificando a esta última ya sea en su amplitud o forma de onda, de tal manera que la salida de este modificador de señal permita obtener una señal eléctrica cuya forma sea apropiada a las especificaciones del convertidor A/D que este asociado a un microcontrolador.

En el tercer bloque se observa un microcontrolador, siendo este último un circuito integrado de muy alta escala de integración (millones de transistores unificados en una pequeña pastilla de silicio), cuya estructura está constituida por una unidad de procesamiento central (CPU), distintas clases de memoria (ROM, RAM, EEPROM), un convertidor A/D, periféricos que permiten la comunicación serial con algún otro dispositivo por ejemplo una computadora, está constituido también por un subsistema acumulador y una serie de puertos de entrada-salida. Gracias a las características y funciones con las que cuenta un microcontrolador, aquí es en donde se tiene escrito algún programa (lenguaje ensamblador) que será capaz de adquirir los datos, procesarlos y presentarlos adecuadamente a algunos de los periféricos de salida con lo que cuenta el microcontrolador.

Y por último se tiene el bloque registrador, el cual va a estar posiblemente formado por elementos que permitan llevar a cabo el despliegue o almacenamiento de la información proporcionada por el microcontrolador. Es importante señalar que en este bloque, también se pueden diseñar circuitos digitales, de tal manera que actúen como interfaz entre el microcontrolador y algún otro dispositivo eléctrico, dando origen al control de algún proceso determinado.

II.6 Interfaz RS-232.

Este tipo de interfaz surgió en el año de 1962 y el propósito fundamental por el cual se creó fue debido a la necesidad de comunicar la computadora (definido

generalmente como *Data Terminal Equipment*, DTE) con un modem (definido generalmente como *Data Communication Equipment*, DCE), posteriormente se extendió su uso para comunicar a la computadora con otros periféricos de entrada y salida como por ejemplo impresoras, mouses, lectores ópticos e inclusive otras computadoras. El nombre de RS-232 o también conocido como RS-232C se refiere al estándar que fue establecido por la Asociación de Industrias Electrónicas (EIA, Electronic Industries Association) en donde se definió la forma en que las señales se propagarían por un cable de 25 líneas (con conectores DB25), también se estableció la forma de utilizar los voltajes (intervalos de ceros y unos lógicos) para las señales de tal manera que los circuitos utilizados no tuvieran la posibilidad de dañarse a consecuencia de un corto circuito entre los cables o conectores.

Debido a que el estándar se definió por una asociación cuya dedicación era el ramo de la industria y por ende necesitaban cubrir todas sus necesidades, motivo por el cual utilizaron los conectores DB25, sin embargo esto no significa que se utilicen todas las líneas; de ahí el origen a dos variantes del estándar, los que utilizan los 25 conductores y los que manejan 9 líneas (conectores DB9). Hoy en día se utilizan con mayor frecuencia el uso de la versión de 9 líneas en las computadoras, debido a que se han fabricado más dispositivos de entrada-salida para este tipo de conexión.

En los conectores DB9 hay dos líneas que se utilizan para la transmisión y recepción de datos, una es la conexión a tierra y el resto son líneas de control que se emplean para implementar el protocolo.

Tabla II.6.1 Terminales importantes de la interfaz RS-232 para DB25 y DB9.

DB-9	DB-25	Línea	Acrónimo
No hay equivalencia	1	Tierra de Protección	GND
3	2	Transmisión de datos (transmit)	TXD
2	3	Recepción de datos (receive)	RDX
7	4	Petición de envío (request to send)	RTS
8	5	Dispuesto para enviar (clear to send)	CTS
6	6	Dispositivo de datos listo (data set ready)	DSR
5	7	Circuito común (signal ground)	SG
1	8	Detección de portada (data carrier detect)	DCD
4	20	Terminal de datos lista (data terminal ready)	DTR
9	22	Indicador de llamada (ring indicator)	RI

Para poder llegar a comunicar dos elementos, únicamente se necesitan 3 líneas TXD, RXD y SG. Las demás líneas se utilizan para el protocolo de comunicación.

Por lo que corresponde a las señales eléctricas, se especifica que una línea esta en *alto* si su voltaje está en el nivel de +12 V y se dice que está en *bajo* cuando su voltaje es de -12 V. Este intervalo de niveles de voltaje por lo regular son las señales de salida en una computadora personal, así como en diversos equipos de comunicaciones; mientras tanto en la entrada, las variaciones permitidas son de +3 a +15 V y de -3 a -15 V.

Las computadoras emplean la *lógica negativa* para el procesamiento de la señal recibida en la interfaz RS-232.

II.6.1. Características de la comunicación RS-232.

La transmisión de datos por medio del estándar RS-232 utiliza un bus en serie; es decir, se transmiten los datos bit a bit uno detrás del otro por línea de transmisión en un orden predeterminado y a una velocidad establecida la cual se mide en bits por segundo (bps).

La comunicación serial utilizada por las computadoras personales, usa exclusivamente la transmisión asíncrona la cual necesita de un entorno estable, debido a que si se produce algún problema en la línea de transmisión afectando a algunos de los bits de datos, posiblemente los siguientes bits pueden ser leídos incorrectamente.

La transmisión asíncrona consiste, en que el tiempo que se necesita para transmitir un carácter es independiente del resto de los caracteres; a esto se le puede llamar también ASYNC, al transmitir un dato se debe de especificar cuándo empieza éste y cuándo termina. En este tipo de comunicación cada palabra de datos (cada carácter) es enviada en el interior de un marco (frame) de tamaño variable el cual está delimitado por el bit de inicio y el bit de parada. A continuación se mencionan los diferentes tipos de bit que conforman este frame de comunicación asíncrona.

- 1) Bit de inicio (*start bit*). Es el primer bit de la comunicación asíncrona que le precede a cada palabra de datos, cuando recibe el bit de inicio el receptor, le indica a éste que va a comenzar la recepción de un dato.

- 2) Bit de datos o palabra de mensaje (*data bits*): Este es el conjunto de bits que componen el carácter a transmitir, este conjunto de bits puede ser de 5 a 8 bits de longitud.
- 3) Bit de paridad (*parity bit*): Este bit es el que se utiliza para comprobar la calidad de la transmisión, este bit es agregado al final de la serie de bit de datos. A menudo no se utiliza pero cuando se emplea la paridad puede ser *par* (“Even”) e *impar* (“Odd”), este tipo de paridad consiste en verificar el número de 1's que son incluidos en los bits de datos; cuando se recibe un carácter, se cuenta el número de 1's y se compara con el bit de paridad, de esta manera se verifica si algunos de los bits de datos sufrió algún cambio, si éste se origina entonces la paridad no coincide, originando un error que permitirá la retransmisión de los datos. Por otro lado también se encuentra la paridad llamada *marca* (“Marke”) que indica que se incluirá siempre un bit de valor “1” (marca) como bit de paridad y por último el bit de paridad o *espacio* (“Space”), el cual siempre añadirá un bit “0” al bit de paridad. Como se puede ver estas dos últimas formas de paridad no aportan ninguna información, por lo que son usados raramente.
- 4) Bit de paro (*stop bit*). Este bit es el que finaliza el frame de transmisión, este bit le indica al receptor que la palabra de datos ha sido transmitida. La duración de este bit puede ser de 1, 1.5 o 2 periodos con la finalidad de que el receptor tenga el tiempo suficiente para almacenar o procesar el dato transmitido.

En la comunicación por RS-232 se utiliza el parámetro *velocidad de transmisión* la cual es expresada en **bauds** y por lo regular suele ser confundido con **bps** los cuales se definen como los bits transmitidos por segundo, pero es importante señalar que estos bps está constituido por los bits de datos y los baudios, para esto se debe entender que un baud es igual al número de veces que cambia una señal en un segundo y en este cambio de la señal se puede transmitir más de un bit, por ejemplo, se tiene un modem de 600 baudios, lo cual significa que la señal transmitida cambia 600 veces en un segundo y sobre cada estado o cambio de la señal se transmiten 4 bits de datos (esto depende de la técnica de modulación que se utilice) por lo cual se tiene que la velocidad de transmisión medida en bps será el número de bits de datos multiplicado por los cambios de la señal en un segundo.

$$\mathbf{bps} = (\text{bauds por cada segundo}) * (\text{número de bits de datos en cada cambio de estado})$$

Siguiendo nuestro ejemplo quedaría: $600 \text{ bauds/segundo} * 4 \text{ bits/ baud} = 2400 \text{ bps}$, donde bps serían los bits transmitidos por cada segundo.

En las comunicaciones por RS-232 a menudo se utilizan el siguiente rango de bauds: 1200, 2400, 4800, 9600, 19200, 38400, etc.

Por último se tiene el parámetro de *modo de control de flujo*, que consiste en una serie de métodos para coordinar la transferencia de datos entre un instrumento de medición y una computadora o modem. Los modos que por lo regular se utilizan son los siguientes:

- *Ninguno (None)*: En este modo, los datos son enviados y recibidos sobre la interface sin utilizar ningún control de flujo, únicamente se utilizan tres líneas del RS-232 que son el TXD, RDX y GND (tierra). Cuando se utiliza este método, se recomienda utilizar un rango de bauds menor a 9600 baud y que en ambos lados, la sincronización de los datos sea a la misma tasa de baudios. Muchas veces este tipo de método es suficiente para algunas aplicaciones solo que se encuentra limitado para resolver problemas cuando los receptores están sobrecargados.
- *XON/XOFF*: Este modo de control de flujo por software, utiliza caracteres especiales incrustados en el canal de datos para el control de flujo es decir, se envían los caracteres de control sobre la misma línea donde son transmitidos los datos. Por ejemplo si un instrumento² esta direccionado para enviar datos, este continúa enviando los datos hasta que el carácter 13H (XOFF) es recibido. Cuando el 11H (XON) es recibido, el instrumento acepta el envío de datos.
- *RTS/CTS*: Este modo de control por hardware utiliza las líneas RTS y CTS del RS-232. Cuando la línea CTS está en verdadero, el remitente envía datos sobre la interfaz. Cuando esta misma línea se establece a falso el remitente detiene el envío de información. La línea RTS la utiliza el receptor para indicar que está listo para recibir datos, se pone en falso cuando el buffer de entrada esta casi por llenarse, es decir, ya no puede recibir más datos hasta que se libera el espacio y cuando ocurre esto, se libera la línea indicando que nuevamente está listo para recibir más datos.
- *DTR/DSR*: En este modo, es utilizado principalmente para la comunicación de un instrumento con un modem, en donde el instrumento monitorea el estado del la línea DSR (data set ready) del RS-232. Cuando esta línea pasa a ser verdadera, el

² Un instrumento puede ser una PC o un aparato que utilice interfaz RS-232.

instrumento envía datos sobre la interfaz. Cuando esta línea se pone en falso, el instrumento detiene el envío de información. La línea DTR la pone en verdadero en este caso el modem para especificar que se ha enlazado la comunicación por lo que el receptor debe estar listo para enviar o transmitir datos. A menudo se utiliza este control conjuntamente con el RTS/CTS de tal forma que el control DTR/DSR indica que el sistema está listo para que haya comunicación y el control RTS/CTS para el control individual de paquetes de datos.

Capítulo III. Visión y ámbito del proyecto.

III.1 Introducción.

Al iniciar un proyecto de software, es imprescindible basarse o apoyarse en el proceso de desarrollo del software, el cual nos ayudará en gran medida a planear, controlar y supervisar el proyecto. En este caso, se requiere de la elaboración de un sistema en software, que permita adecuarse a las necesidades del Observatorio de Radiación Solar (ORS), así como también en la búsqueda de una mejor utilización de los recursos en hardware y software con los que cuenta actualmente este observatorio, por tal motivo, la elaboración de este proyecto se basará en las fases del proceso de desarrollo que se mencionó en el capítulo II de esta tesis y durante el transcurso en cada una de las fases de desarrollo se utilizará como herramienta de modelación visual a UML (Unified Modelling Language) debido a que hoy en día se ha convertido en uno de los lenguajes de modelación más utilizados en el campo de trabajo del desarrollo de software, además de que cuenta con la capacidad de especificar, visualizar, construir y documentar cada uno de los elementos que se vayan originando en el sistema. En el apéndice A de esta tesis se presenta con más detalle la notación que utiliza UML. Unificando todo esto, permitirá obtener los elementos suficientes para obtener un sistema en software que cumpla con las expectativas del observatorio.

En este capítulo se describen las actividades que hasta antes del desarrollo del software se cumplían para llevar a cabo todo el proceso de adquisición y procesamiento de datos, esto con el fin de comprender el ámbito del sistema.

Se plantean las necesidades del ORS, las cuales deben de satisfacerse con la ayuda del sistema a desarrollar, en otras palabras se determina la visión del proyecto y se presenta la fase de preparación inicial para la elaboración del sistema de software, así como también se incluyen el modelo de los casos de uso, que permitirá comprender de forma intuitiva el comportamiento del sistema de software.

III.2 Antecedentes de operación.

El Instituto de Geofísica (IGF) a través del Observatorio de Radiación Solar cuenta con 4 estaciones de monitoreo de radiación solar, estas estaciones se encuentran ubicadas una en el Colegio de Posgraduados de Chapingo, dos en Ciudad Universitaria (Jardín Botánico e Instituto de Geofísica) y una en la población de Orizabita, Estado de Hidalgo. Cada estación cuenta con sensores de radiación solar, sensores meteorológicos y un sistema de adquisición de datos; por lo que corresponde a la unidad de almacenamiento central, se cuenta con una estación de trabajo (Marca Dell) la cual opera bajo el sistema operativo Linux Red-Hat en su versión (7.2), hasta el momento la forma de operar en estas estaciones se describe a continuación.

En la estación ubicada en el Instituto de Geofísica cuenta con el sistema de adquisición modelo HP 4566 el cual es controlado por una computadora HP85 donde se ejecuta un programa hecho en lenguaje de programación basic. Como se puede observar el equipo con el que se cuenta ya es demasiado obsoleto, además este equipo se encuentra en la última etapa de su vida útil. De ahí la necesidad de hacer una renovación de equipo (hardware) y de software. El programa que controla al aparato de adquisición va generando archivos en texto plano donde se guardan las mediciones minuto a minuto las 24 horas del día, 365 días al año de 12 parámetros radiacionales. La información total que contiene cada archivo son las mediciones realizadas durante el día. Posteriormente una persona tiene que darle formato y orden a la información almacenada dentro de los archivos, para que estos datos sean cargados manualmente a la base de datos con la que cuenta el observatorio.

Para la estación remota de Chapingo, el procedimiento es parecido al anterior pero con la diferencia que los datos ya formateados son enviados al observatorio por medio de correo electrónico y estos al llegar, nuevamente son ingresados manualmente a la base de datos. Esto retrasa en gran medida la llegada de la información a la base de datos del observatorio, de aquí la importancia de tener un sistema de información que ayude a realizar estas tareas en forma rápida y automatizada.

Para la estación ubicada en la población de Orizabita, Estado de Hidalgo, el sistema de adquisición tiene un modulo de memoria externa en donde se almacena la información durante un mes y medio, obligando al personal del observatorio, a realizar

una visita mensual a esta estación para recoger la información en medios magnéticos y luego ingresarlos manualmente a la base de datos.

Para la estación Jardín Botánico, su funcionamiento es idéntico a la estación anterior (Orizabita, Hgo.).

Al tener la información en la base de datos, se tiene un sitio Web pequeño, donde se presenta esta información, únicamente se muestra las integrales horarias y diarias de radiación solar, se tienen datos históricos a partir de Enero del año de 1997 a la fecha. No se cuenta con otro tipo de información a detalle más que la presentación de una tabla con números y fechas.

De los punto anteriores, podemos concluir la necesidad de homologar las estaciones en cuanto a sistemas de adquisición, computadoras y programas de computo, que hagan más eficiente el proceso de monitoreo de la radiación solar.

III.3 Necesidades del Observatorio de Radiación Solar.

Formar una Red Solarimétrica utilizando como medio de comunicación Internet, para evaluar la energía solar disponible en toda la República Mexicana.

Necesita utilizar la infraestructura disponible del Observatorio de Radiación Solar (ORS), del Instituto de Geofísica (IGF) de la Universidad Nacional Autónoma de México (UNAM) para crear una red de estaciones meteorológicas y radiométricas, que transmitan en tiempo real su información, permitiendo contar con una alta eficiencia, semejante a la obtenida con sistemas de tecnología de punta con una mínima inversión económica.

El ORS requiere organizar e integrar de forma automatizada y en tiempo real la información obtenida de los sistemas de adquisición de datos ubicados en los sitios remotos, en una sola ubicación, es decir que los datos sean almacenados y organizados en el servidor central localizado en el IGF para que posteriormente sean utilizados por otros sistemas de información.

Se requiere desarrollar un sistema de adquisición de datos en base a el equipo HP 34970A, para el proceso de adquisición de datos en tiempo real de la red solarimétrica.

Proporcionar a diferentes usuarios industriales, profesionistas, estudiantes, investigadores y público en general entre otros, información detallada de la radiación solar a través de un medio electrónico, el cual permitirá conocer la situación en tiempo real del régimen radiacional en los lugares donde se ha instalado una estación solarimétrica.

III.4 Análisis de requerimientos.

En base a las necesidades descritas anteriormente, se requiere tener un análisis de requisitos más detallado, esto con el fin de tener bien claro lo que realmente se desea obtener y que las necesidades del ORS queden cubiertas en su totalidad, además, esta clarificación de requisitos ayudará a que el sistema no requiera algún cambio en la etapa final, por causa de que no se haya considerado o previsto alguna característica o función, que pueda alterar en gran manera el desarrollo del proyecto obteniendo resultados indeseables.

De acuerdo a lo anterior, a continuación se presentan los cuatro módulos principales el cual estará constituido el Sistema de Información del Observatorio de Radiación Solar (SIORS) en el cual se explican las actividades que se llevaran acabo en cada uno de estos módulos.

Adquisición de datos: Es esencial que el sistema tenga la funcionalidad que permita comunicarse con el instrumento de adquisición de datos HP 34970A, el aparato cuenta con una interfaz RS232 el cual se puede conectar a cualquier PC portátil, escritorio o cambiador de medio. La ventaja de utilizar este tipo de interfaz es que en el mercado se puede conseguir gran variedad en marcas de tarjetas de expansión PCI para cualquier tipo de computadora que tenga este puerto, a un precio accesible, además pensando en que el equipo de medición estará en algún lugar remoto, existe la posibilidad de controlar el dispositivo de medición de forma remota, esto se puede realizar con ayuda de un cambiador de género (DIGI) cuya función es la de transformar la señal que viaja a través de un medio Ethernet a un medio RS232 (comunicación serial); en otras palabras se

puede comunicar una computadora con algún dispositivo que tenga una interfaz RS-232 a miles de kilómetros de distancia entre ellos.

Con la comunicación establecida entre el aparato de adquisición y la computadora, el programa deberá leer un archivo de configuración cuyo contenido serán las instrucciones que permitirá configurar el HP 34970A. Las instrucciones que se escriban en ese archivo de configuración podrán tener la siguiente secuencia:

- Inicialización del HP 34970A
- Configuración de los canales a leer
- Intervalo de tiempo de medición
- Ajuste de datos para determinar el valor promedio.

Cada una de estas instrucciones son propias del lenguaje de programación que reconoce el HP 34970A para su operación, llamado SCPI¹ (Standard Commands for Programmable Instruments), este lenguaje fue creado con la intención de tener una definición estándar de comandos, para ser aplicados en la operación y configuración de aparatos de medición.

Lectura y almacenamiento de datos: Una de las funciones importantes a realizar es que al quedar configurado el aparato de adquisición, permita arrancar y detener la lectura de datos, estos al ser leídos a través de la comunicación por RS232 serán almacenados físicamente en el disco duro de la computadora. La forma de almacenamiento consistirá en la creación de archivos en texto plano que alojarán las mediciones de cada minuto del día proporcionadas por las lecturas obtenidas en cada uno de los canales de medición que tiene el HP 34970A. Este archivo estará constituido por *nombre de la estación, numero de total de canales que se están midiendo, fecha, hora, minuto, medición del canal 1, medición del canal 2,...medición del canal n.*

Se debe de contar con un archivo que pueda ser modificado con cualquier editor de texto, cuyo contenido será la definición de los parámetros de configuración que puntualizará la forma de almacenamiento de las mediciones. Los parámetros básicos con lo que se debe contar son los siguientes:

¹ El consorcio SCPI estaba originalmente formado por Tektronix, Wavetek, Fluke/Philips, Racal-Dana, Keithley, Briel y Hewlett-Packard -Este último era el que pregonaba el uso de este lenguaje-

- ✓ Nombre de la estación al cual pertenece.
- ✓ Número total de canales de medición.
- ✓ Ubicación donde se almacenará temporalmente el archivo de mediciones.

Transmisión de datos: El sistema debe contar con la capacidad de transmitir las mediciones hacia el servidor central de almacenamiento y procesamiento, a través del medio de comunicación Internet en forma segura, es decir utilizar un túnel entre el sitio remoto y el servidor que permita establecer un canal de comunicación en donde la información que es transmitida y recibida sea encriptada o bien la creación de autenticaciones seguras por parte del servidor y el cliente, donde el intervalo de tiempo para ser transmitidos los datos debe ser mayor al tiempo de captura; debe de ser capaz de determinar si existe conexión con el servidor para realizar la transmisión y si esta última no la hubiere, debe contar con los mecanismos pertinentes para crear una especie de “cola”, la cual almacenará los archivos pendientes a transmitir y en cuanto la comunicación se establezca, debe tener la facultad de transmitir esos archivos en cola, para que posteriormente sea depurada. Si la comunicación con el servidor no se ha establecido por más de un día o que la cola haya excedido en tamaño al número de bytes permitidos establecido en el archivo de configuración de la transmisión, el sistema debe ser capaz de mover los datos de la cola hacia otra ubicación (dentro del disco duro de la computadora), de tal manera que en determinado momento puedan ser recuperados los datos, a los cuales se les llamará datos *históricos*. Los datos históricos ya no serán transmitidos hacia el servidor mediante este proceso.

El archivo de configuración correspondiente a la transmisión, también podrá ser modificado por cualquier editor de texto y debe contener por lo menos los siguientes parámetros básicos:

- Numero de bytes permitidos para almacenarse en la cola.
- Nombre o dirección del servidor al cual se harán llegar los datos.
- Parámetros para establecer el envío de los datos en forma segura.

Recepción de datos: El sistema debe tener del lado del servidor un programa o *demonio* el cual se encargará de estar monitoreando la llegada de los archivos, según la estación

que corresponda. Este demonio al detectar la llegada de un archivo, lo adjuntará a otro archivo ordenando los datos cronológicamente. Este segundo archivo llamado *activo*, almacenará las mediciones pertinentes al día actual del arribo de los datos, al cambiar la fecha del día, el archivo *activo* se cambiará de nombre utilizando como tal la fecha del día anterior, es decir, el año-mes-día correspondiente a la fecha de las mediciones almacenadas en ese respectivo día.

Como se puede observar el demonio debe ser capaz de verificar los cambios de día, y de realizarse este, el archivo *activo* será “vaciado” para que nuevamente sea llenado con los datos del nuevo día. Este tipo de funcionamiento se definió de esta manera debido a que los datos serán utilizados por diversos programas que utilizan en el ORS para proporcionar otro tipo de presentación y visualización.

III.4.1 Diagramas de caso de usos.

A continuación se visualizan diferentes diagramas de caso de uso los cuales ayudan a tener un panorama mucho más amplio de como se desea que trabaje el sistema en base a la información que se ha obtenido anteriormente. En algunos diagramas de caso de uso se construyen diagramas mucho más detallados que los generales debido a la complejidad que los diagramas de caso de uso generales presentan.

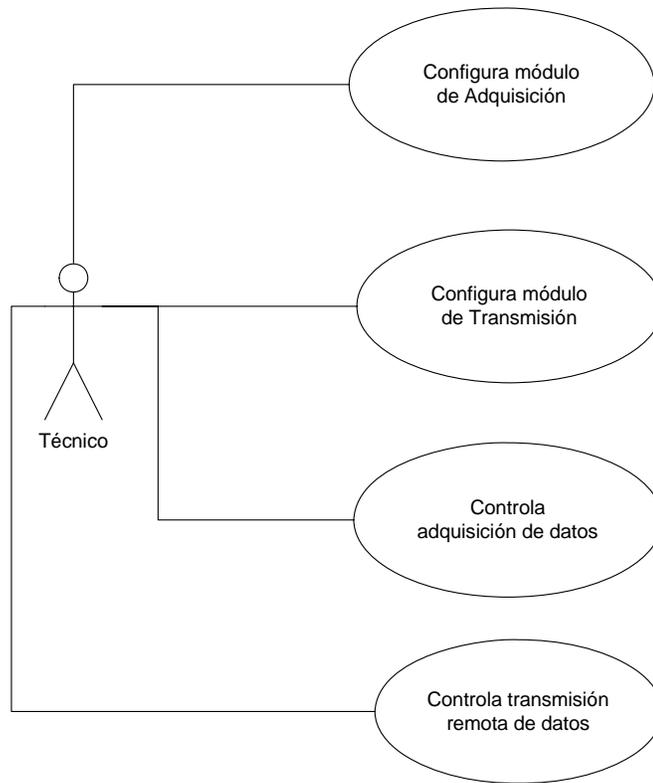


Figura III.4.1.1 Operaciones realizadas en el sitio de recepción remota.

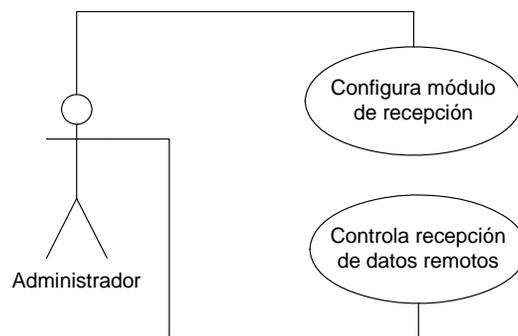


Figura III.4.1.2 Operaciones realizadas del lado del servidor.

Caso de uso	Configura módulo de adquisición
Actores	Técnico
Tipo	Básico
Propósito:	Escribir en el archivo de configuración llamado <i>rmed.conf</i> , las instrucciones SCPI para el aparato de medición.
Precondiciones	Ninguno
Flujo Principal	Por medio de un editor de texto, el técnico edita el archivo llamado <i>rmed.conf</i> , en donde escribirá las instrucciones o comandos (SCPI) que son propias del aparato de adquisición. Inicialmente el contenido de este archivo tendrá las instrucciones básicas para configurar el aparato de adquisición. Al terminar de editar el archivo, el técnico lo guarda en la misma ubicación donde lo abrió utilizando el mismo nombre.
Subflujos	Ninguno
Excepciones	Ninguno

Caso de uso	Configura módulo de transmisión
Actores	Técnico
Tipo	Básico
Propósito	Escribir en el archivo de configuración llamado <i>radcom.conf</i> , las variables o parámetros necesarios para la operación del proceso de transmisión.
Precondiciones	Ninguno
Flujo Principal	Por medio de un editor de texto, el técnico edita el archivo llamado <i>radcom.conf</i> . Este archivo contiene las variables necesarias que permiten operar el módulo de transmisión de datos hacia el servidor. Al terminar de editar el archivo, se guarda en la misma ubicación donde lo abrió utilizando el mismo nombre.
Subflujos	Ninguno
Excepciones	Ninguno

Caso de uso	Controla Sistema de Adquisición
Actores	Técnico
Tipo	Básico
Propósito	Ejecutar las operaciones de control del programa <i>Datademon</i> .
Precondiciones	Ninguno
Flujo Principal	A través de una terminal de comandos, el técnico podrá ejecutar el comando <i>Datademon</i> con las opciones con las opciones <i>start</i> , <i>stop</i> , y <i>config</i> iniciando los subflujos S-1, S-2 y S-3.
Subflujos	S-1 Se inicia el caso de uso <i>Inicia proceso</i> con la opción <i>start</i> . S-2 Se inicia el caso de uso <i>Detiene proceso</i> . S-3 Se inicia el caso de uso <i>Inicia Proceso</i> con la opción <i>config</i> .
Excepciones	E-1 El parámetro de entrada es incorrecto y/o sintaxis inválida. Se muestra un mensaje con la sintaxis correcta a utilizar y los parámetros permitidos por <i>Datademon</i> .

Caso de uso	Controlar transmisión de datos remotos
Actores	Técnico
Tipo	Básico
Propósito	Capturar los datos de mediciones y comenzar con la transmisión de las mismas hacia el servidor.
Precondiciones	Ninguno
Flujo Principal	A través de una terminal de comandos, el técnico podrá ejecutar el comando <i>Raddemon</i> , para iniciar este caso de uso. Si se desea detener el programa únicamente hay que terminar con el proceso de este programa (se puede utilizar el pulso de las teclas Ctrl-c). Este caso de uso inicia los casos de uso <i>Transmite datos</i> , <i>Verifica existencia de datos</i> y <i>Crea cola de datos</i> .
Subflujos	Ninguno
Excepciones	Ninguno

Caso de uso	Configura módulo de recepción.
Actores	Administrador
Tipo	Básico
Propósito	Almacenar dentro del archivo de configuración llamado <i>syscom.conf</i> los parámetros para el funcionamiento del demonio <i>dataserv</i> .
Precondiciones	Ninguno
Flujo Principal	Con ayuda de un editor de texto, el administrador modificará los parámetros necesarios para la operación del demonio <i>dataserv</i> . Posteriormente guardará el archivo en la misma ubicación donde lo abrió utilizando el mismo nombre.
Subflujos	Ninguno
Excepciones	Ninguno

Caso de uso	Controla recepción de datos remotos.
Actores	Técnico
Tipo	Básico
Propósito	Controlar el funcionamiento del programa <i>radcom</i> , cuyo propósito es recopilar y ordenar los datos provenientes de la estación remota, que se haya indicado.
Precondiciones	Ninguno
Flujo Principal	Este caso de uso se inicia cuando el administrador, por medio de una terminal de comandos, ejecuta el comando <i>radcom</i> seguido del nombre de la estación a capturar, esto iniciará el proceso de captura correspondiente a la estación especificada. Para terminar con este caso de uso, únicamente hay que terminar el proceso en el sistema operativo o bien en la terminal donde se inicio hay que presionar las teclas <i>Ctrl-C</i> .
Subflujos	Ninguno
Excepciones	E-2 El parámetro de entrada es incorrecto y/o sintaxis inválida. Se muestra un mensaje con la sintaxis correcta a utilizar y los parámetros permitidos por <i>radcom</i> .

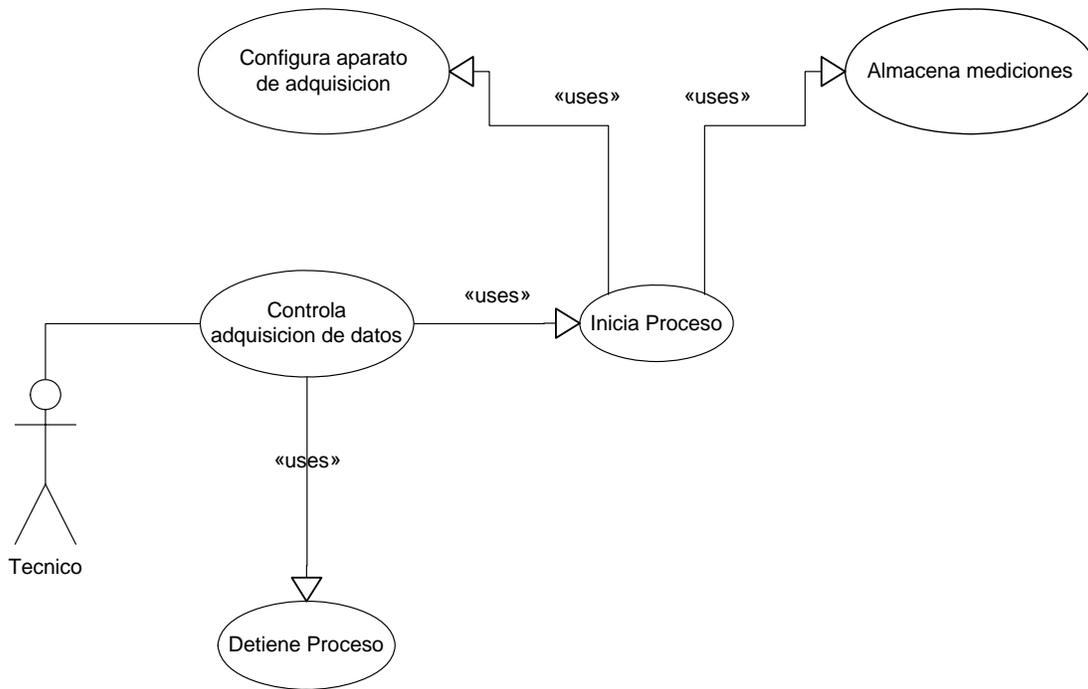


Figura III.4.1.3 Casos de uso del control de adquisición de datos.

Caso de uso	Iniciar Proceso
Actores	Técnico
Tipo	Uso
Propósito	Iniciar el proceso de captura de las mediciones y el almacenamiento temporal de estas mismas en el disco duro de la computadora. Este caso de uso inicia los casos de uso <i>Configura aparato de adquisición</i> y <i>Almacena mediciones</i> .
Precondiciones	Haber iniciado el caso de uso <i>Controla adquisición de datos</i> .
Flujo Principal	Si este caso de uso se inicia con la opción config se realiza el subflujo S-4. Si este caso de uso se inicia con la opción start, se realiza el subflujo S-5.
Subflujos	S-4 Se inicia el caso de uso <i>Configura aparato de adquisición</i> S-5 El proceso de captura de mediciones comienza iniciando los casos de uso <i>Establece comunicación RS-232</i> y posteriormente <i>Almacena mediciones</i> .
Excepciones	E-3 Si no se ha configurado previamente el aparato de adquisición muestra un mensaje de error de configuración.

Caso de uso	Detiene Proceso
Actores	Técnico
Tipo	Uso
Propósito	Detiene el proceso de almacenamiento y captura de mediciones en la computadora, así como también detiene el proceso de adquisición de datos en el dispositivo HP.
Precondiciones	Se requiere que <i>Datademon</i> se haya iniciado.
Flujo Principal	Verifica el estado de operación de <i>Datademon</i> . Si el estado de <i>Datademon</i> esta iniciado entonces se ejecuta el subflujo S-6, en caso contrario se ejecuta el subflujo S-7
Subflujos	S-6 Se envía la instrucción de STOP al aparato de adquisición de datos para detener la captura de mediciones. Termina con el funcionamiento del demonio <i>Datademon</i> esto es, detiene el almacenamiento de las mediciones en disco duro y detiene el proceso de lectura de mediciones a través del puerto serie. S-7 Genera la excepción E-4
Excepciones	E-4 No se ha iniciado el proceso de lectura de mediciones.

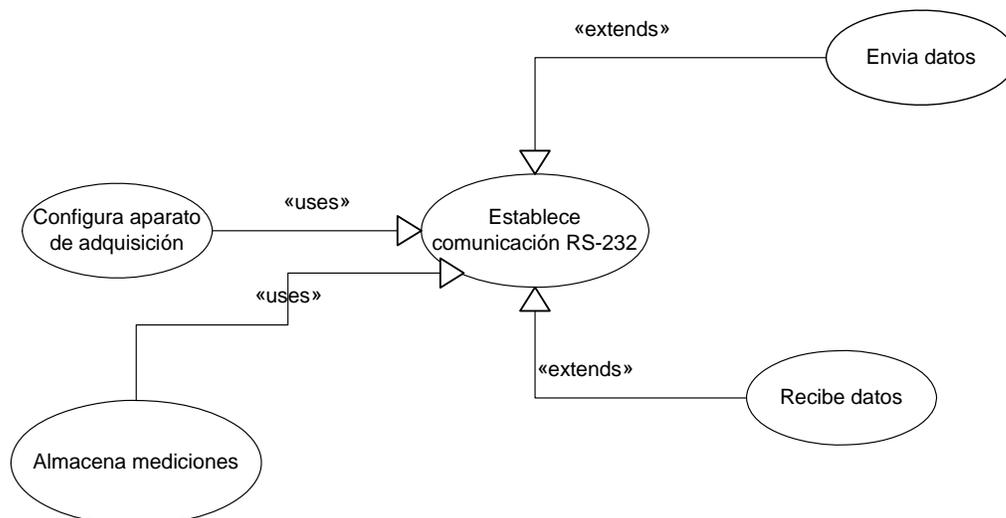


Figura III.4.1.4 Casos de uso para la comunicación por medio de la interfaz RS-232 entre la computadora y el HP39470A y el almacenamiento en disco duro de las mediciones.

Caso de uso	Configura aparato de adquisición
Actores	Técnico
Tipo	Uso
Propósito	Transmitir a través del puerto serie (RS232), las instrucciones necesarias para la operación del aparato de adquisición de datos.
Precondiciones	Ninguna
Flujo Principal	Se lee el archivo de configuración <i>rmed.conf</i> para inicializar el proceso de operación de lectura de las mediciones. Se inicia el caso de uso <i>Establece comunicación RS-232</i> con la opción de escritura.
Subflujos	Ninguno
Excepciones	E-5 No existe o no se pudo leer el archivo <i>rmed.conf</i>

Caso de uso	Establece comunicación RS-232
Actores	Técnico
Tipo	Uso
Propósito	Establecer el canal de comunicación por medio de la interfaz RS-232 entre la computadora y el HP39470A.
Precondiciones	Ninguna
Flujo Principal	Se lee el archivo de configuración <i>radcom.conf</i> para cargar los parámetros básicos para la configuración del puerto RS-232. Si se inicio este caso de uso con la opción <i>lectura</i> se ejecuta el subflujo S-8; pero si la opción fue <i>escritura</i> se ejecuta el subflujo S-9. Se cierra la comunicación con la interfaz RS-232.
Subflujos	S-8 Se inicia el caso de uso <i>Recibe datos</i> . S-9 Se inicia el caso de uso <i>Envía datos</i> .
Excepciones	E-6 No existe o no se pudo leer el archivo <i>radcom.conf</i> E-7 Error al abrir el puerto de comunicación. E-8 No existe el nombre de puerto RS-232 proporcionado por el archivo <i>radcom.conf</i> .

Caso de uso	Envía datos
Actores	Técnico
Tipo	Extendido
Propósito	Enviar datos a través de la interfaz RS-232 de la computadora hacia el aparato de adquisición HP39470A.
Precondiciones	Haber iniciado el caso de uso Establece comunicación RS-232.
Flujo Principal	Al haberse establecido adecuadamente la comunicación con el HP39470A y la computadora, comienza con la transmisión de datos desde la PC hacia el aparato de adquisición.
Subflujos	Ninguno
Excepciones	E-9 Error al escribir en el puerto serie.

Caso de uso	Recibe datos
Actores	Técnico
Tipo	Extendido
Propósito	Leer los datos a través de la interfaz RS-232, proporcionado por el aparato de adquisición HP39470A.
Precondiciones	Haber iniciado el caso de uso Establece comunicación RS-232.
Flujo Principal	Al haberse establecido adecuadamente la comunicación con el HP39470A y la computadora, comienza con la lectura de datos a través del puerto serie de la PC.
Subflujos	Ninguno
Excepciones	E-10 Error al leer en el puerto serie.

Caso de uso	Almacena mediciones
Actores	Técnico
Tipo	Uso
Propósito	Asigna un formato a los datos capturados desde el puerto serie de la computadora y los almacena en un archivo temporal.
Precondiciones	Ninguno
Flujo Principal	Carga las variables o parámetros que indican la forma de grabación de las mediciones proporcionadas por el HP39470A. Al recibir un dato obtenido proveniente del puerto serie de la computadora, utilizando los parámetros de grabación aplica el formato al dato recién llegado. Almacena el dato de forma incremental en un archivo temporal, cuyo nombre y ubicación de este archivo es especificado por alguno de los parámetros de grabación.
Subflujos	Ninguno
Excepciones	E-11 No se puede escribir el dato en el archivo temporal.

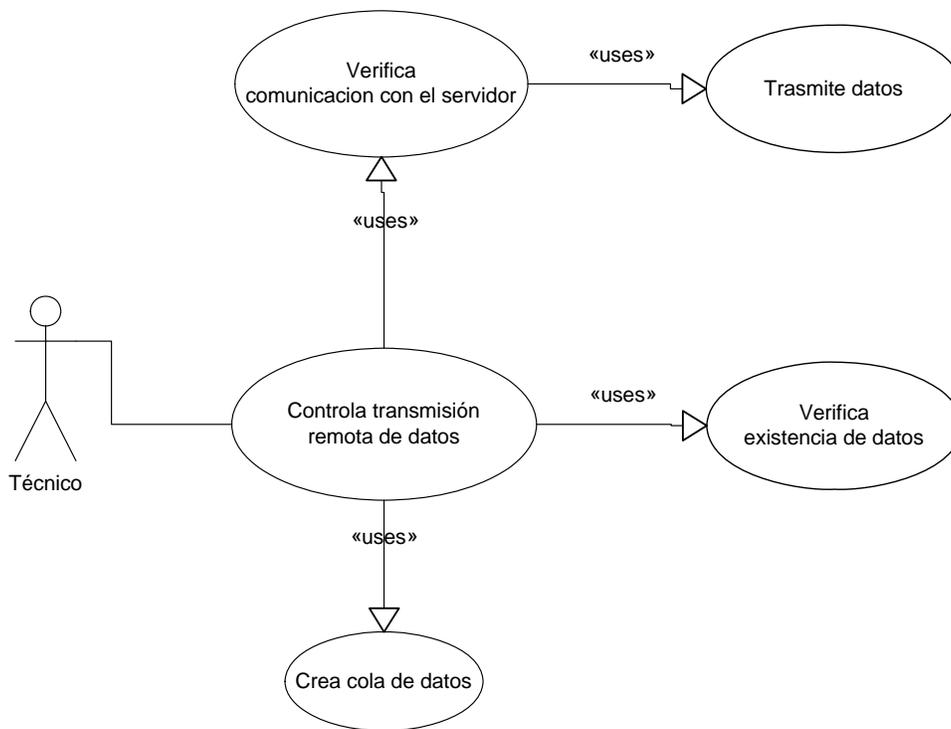


Figura III.4.1.5 Diagrama de caso de uso para la etapa de transmisión remota de mediciones.

La descripción del diagrama del caso de uso controla transmisión remota de datos ya fue detallado anteriormente.

Caso de uso	Verifica comunicación con el servidor
Actores	Técnico
Tipo	Uso
Propósito	Proporciona un estado que indica si existe comunicación con el servidor de recepción de datos.
Precondiciones	Ninguna
Flujo Principal	Realiza la conexión con el servidor, utilizando los parámetros almacenados en el archivo de configuración <i>radcom.conf</i> Si la comunicación existe se genera el subflujo S-10 en caso contrario se ejecuta el subflujo S-11
Subflujos	S-10 Se ejecuta el caso de uso transmite datos. S-11 Se genera un estado que indica que la comunicación no está disponible.
Excepciones	E-12 El nombre del host es desconocido E-13 Nombre de usuario y/o contraseña inválido. E-14 Se perdió la conexión con el servidor. E-15 No se tiene acceso a la red.

Caso de uso	Transmite datos
Actores	Técnico
Tipo	Uso
Propósito	Transmitir los datos almacenados por el caso de uso <i>Almacena mediciones</i> y la cola de datos hacia el servidor del ORS en el IGF.
Precondiciones	Que exista comunicación con el servidor.
Flujo Principal	Carga las variables que indican el origen y destino de los archivos a transmitir. Abre un canal sftp(Secure File Transfer Protocol) Posteriormente ejecuta los comandos sftp que permite transmitir los datos locales (se refiere a los datos proporcionados por el caso de uso <i>Almacena mediciones</i> y a los datos en cola si es que los hubiere) hacia el servidor. Cierra el canal sftp.
Subflujos	Ninguno
Excepciones	E-16 Permiso denegado (lectura/escritura) en el sitio remoto. E-17 El directorio remoto no existe. E-18 El directorio y/o archivo local no existe.

Caso de uso	Crea cola de datos
Actores	Técnico
Tipo	Uso
Propósito	Mantiene en una cola, los datos proporcionados por el caso de uso <i>Almacena mediciones</i> en caso de que la comunicación entre el sitio remoto y el servidor sea nula.
Precondiciones	Que existan datos a transmitir y que la comunicación con el sitio remoto no exista.
Flujo Principal	<p>Verifica la existencia de una cola de datos.</p> <p>Si existe la cola de datos, se verifica que el tamaño de esta misma no exceda el límite especificado por el parámetro de tamaño de la cola de datos.</p> <p>Si el tamaño de la cola de datos se excedió se ejecuta el subflujo S-12. En caso contrario se ejecuta el subflujo S-13.</p> <p>Si no hay una cola de datos, se crea una nueva.</p>
Subflujos	<p>S-12 Se mueven todos los datos almacenados en la cola a un archivo llamado <i>history</i>, cuyo contenido serán los datos provenientes de las colas que hayan excedido su tamaño. Posteriormente se elimina la cola.</p> <p>S-13 Agrega en la cola existente los nuevos datos proporcionados por el caso de uso <i>Almacena mediciones</i>.</p>
Excepciones	<p>E-19 Error al leer los datos de mediciones.</p> <p>E-20 Error al crear la cola de datos</p> <p>E-21 Error al escribir en la cola de datos.</p> <p>E-22 Error al escribir en el archivo <i>history</i>.</p> <p>E-23 Error al eliminar la cola de datos.</p>

En esta sección se detallan los casos de uso utilizados para la recepción de los datos en el lado del servidor, a excepción del caso de uso *controla recepción de datos remotos*, pues ya se ha descrito anteriormente.

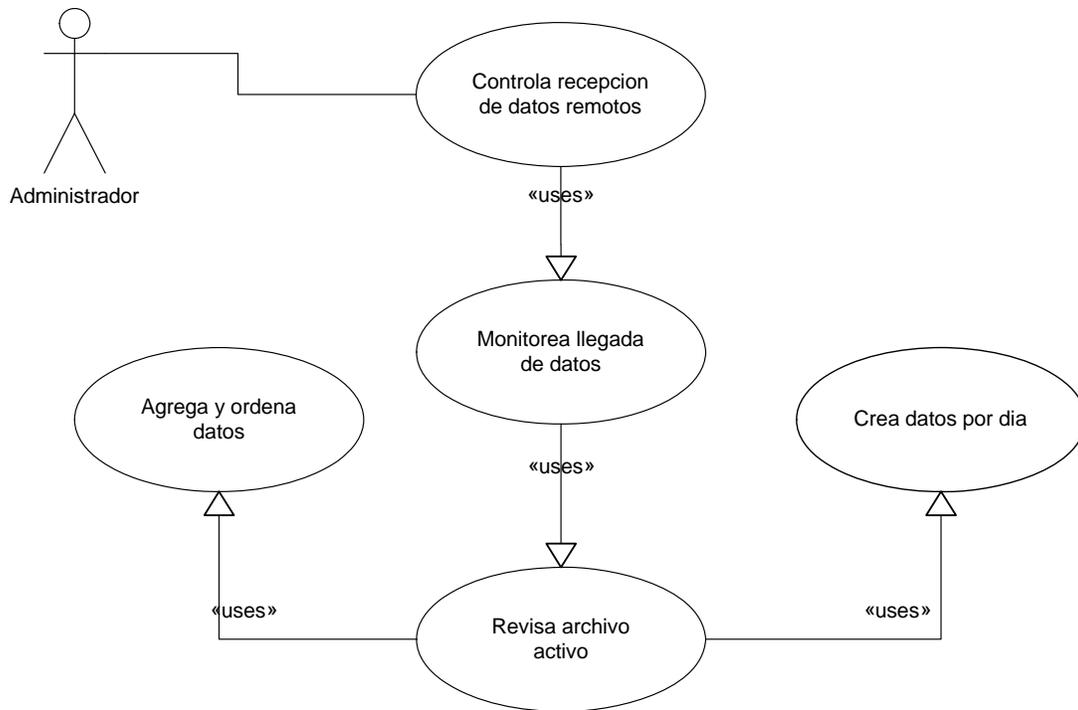


Figura III.4.1.6 Casos de uso para la recepción de datos del lado del servidor.

Caso de uso	Monitorea llegada de datos.
Actores	Administrador
Tipo	Uso
Propósito	Detectar la llegada de datos, provenientes de la estación remota especificada en el caso de uso <i>Controla Recepción de datos remotos</i> .
Precondiciones	Ninguno
Flujo Principal	Carga los parámetros de configuración desde el archivo syscom.conf. Comienza con el monitoreo de llegada de datos de acuerdo al parámetro que especifica la ubicación de arribo de los datos y al nombre de la estación que se desea capturar. Al percatarse de la llegada de datos se ejecuta el subflujo S-14.
Subflujos	S-14 Se ejecuta el caso de uso <i>Revisa archivo activo</i>
Excepciones	E-24 El nombre de la estación no existe. E-25 No se puede leer la ubicación de llegada de archivos E-26 No existe la ubicación de llegada de archivos.

Caso de uso	Revisa archivo activo.
Actores	Administrador
Tipo	Uso
Propósito	Determinar la existencia del archivo <i>active</i> cuyo contenido sean mediciones correspondientes a la fecha actual del sistema, y revisa la fecha de las mediciones contenidas en los archivos de arriba.
Precondiciones	Ninguno
Flujo Principal	Determina la existencia del archivo active. En caso de que no exista este archivo, se ejecuta el subflujo S-15, en caso contrario, determina la fecha y hora de modificación del archivo <i>active</i> y si esta fecha es del día anterior a la del sistema se ejecuta el subflujo S-19. Posteriormente se revisa que la fecha del dato a almacenar este dentro de la fecha actual del sistema, si esto es verdadero se ejecuta el subflujo S-16 y si es falso se ejecuta el subflujo S-17.
Subflujos	S-15 Se crea el archivo active y se ejecuta el caso de uso <i>Agrega y ordena datos</i> . S-16 Se ejecuta el caso de uso <i>Agrega y ordena datos</i> . S-17 Se ejecuta el caso de uso <i>Crea datos por día</i> y posteriormente se ejecuta el caso de uso <i>Agrega y ordena datos</i> . S-19 Se renombra el archivo <i>active</i> por el nombre correspondiente a la fecha de ese día.
Excepciones	E-27 No se puede crear el archivo <i>active</i> . E-25 No se puede crear el archivo correspondiente al día.

Caso de uso	Crea datos por día.
Actores	Administrador
Tipo	Uso
Propósito	Agrega y ordena las mediciones en su correspondiente archivo determinado por su fecha.
Precondiciones	Ninguno
Flujo Principal	Dependiendo de la fecha correspondiente a la medición, se introduce ese dato en su correspondiente archivo. Posteriormente ordena el archivo por medio de la fecha de los datos que lo contiene.
Subflujos	Ninguno
Excepciones	E-25 No se puede agregar los datos.

Caso de uso	Agrega y ordena datos.
Actores	Administrador
Tipo	Uso
Propósito	Agrega y ordena las mediciones en el archivo <i>active</i> .
Precondiciones	Ninguno
Flujo Principal	Obtiene la medición con su correspondiente fecha e introduce este dato en el archivo <i>active</i> . Posteriormente ordena el archivo por medio de la fecha de los datos que lo contiene.
Subflujos	Ninguno
Excepciones	E-26 No se puede agregar los datos.

Capítulo IV. Análisis.

IV.1 Modelo de Arquitectura.

De acuerdo al sistema que se intenta desarrollar, en el capítulo anterior se menciona que el sistema va a estar conformado por cuatro módulos, de los cuales uno de estos estará funcionando de forma independiente y remota de los otros tres. Este tipo de modelo a seguir encaja en el modelo de arquitectura de dos capas cliente/servidor, debido a que por un lado se tiene al cliente, que en este caso se identifica como la entidad la cual va a estar encargada de realizar la adquisición y el procesamiento de los datos, donde acto seguido será la petición de comunicación y enlace con el servidor, para poder almacenar los datos obtenidos durante la adquisición.

Por otro lado el servidor estará continuamente esperando a que el cliente se conecte y que este último transmita el conjunto de datos para que el servidor los capture, almacene y ordene la información, en un lugar específico. En el siguiente diagrama se visualiza la lógica que persigue el modelo de dos capas para esta aplicación a desarrollar.

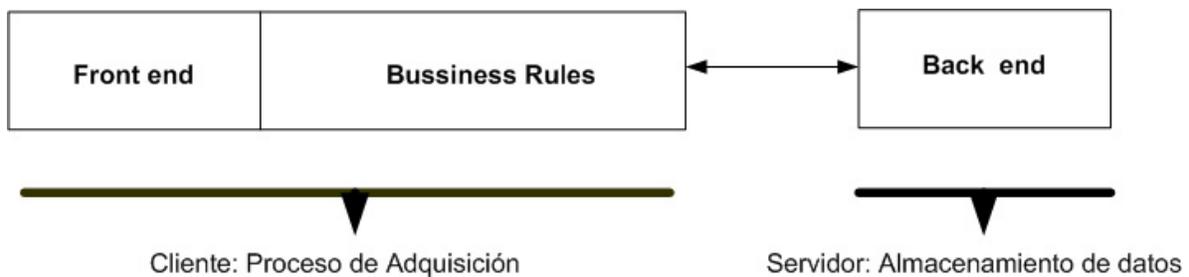


Figura IV.1.1 Modelo de dos capas.

Las ventajas que se tiene al seguir en este tipo de arquitectura son las siguientes:

- Se tiene un alto control de las aplicaciones que se van a ejecutar de forma independiente dentro del cliente y la aplicación de almacenamiento en el servidor.
- Debido a su modularidad, facilita en gran manera alguna adaptación o modificación en algunos de sus procesos.
- Se puede integrar a otro tipo de sistema.

Ya definido el tipo de arquitectura que se va a utilizar, en lo que resta de este capítulo se van a presentar los diagramas de clases, los diagramas de secuencia apoyándose en los casos de uso descritos en el capítulo anterior y por último los diagramas de actividades que detallan la secuencia de acciones implicadas en los casos de uso.

IV.2 Diagrama de clases.

En los siguientes esquemas se ilustra los diagramas de clases pertenecientes al módulo de adquisición de datos y transmisión de datos, los cuáles son la base del sistema de adquisición, sin olvidar la parte de recepción de datos que a continuación es la que se presenta primero debido a su relativa simplicidad. Estos diagramas de clase son la parte fundamental de un desarrollo de software orientado a objetos y nos sirve para identificar los elementos (objetos¹) que van a existir en el sistema en los cuales va a recaer el comportamiento que dicho sistema necesita.

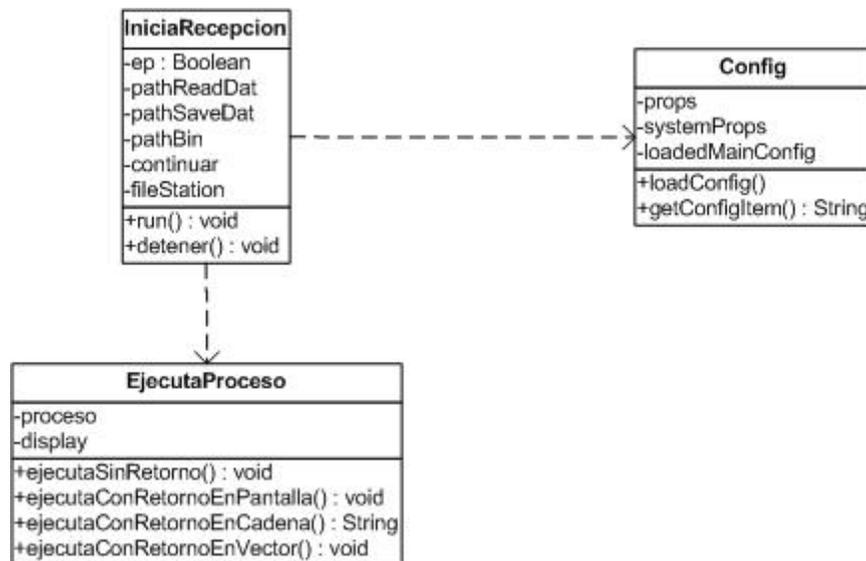


Figura IV.2.1 Diagrama de clases para el modulo recepción de datos.

1 Un objeto se puede definir como una cosa tangible, algún evento, interacción o role que ejecuta una persona o cosa.

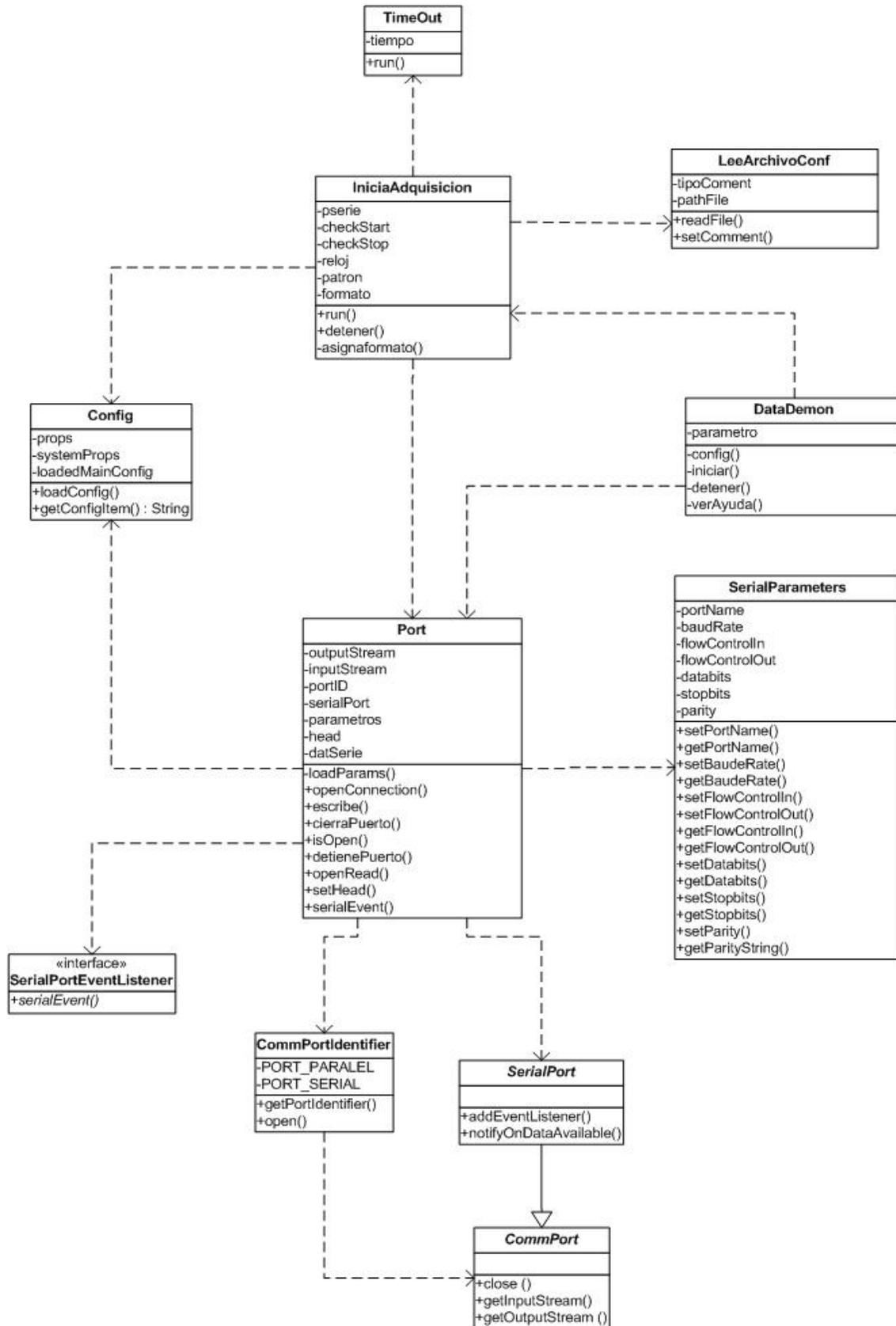


Figura IV.2.2 Diagrama de clases para el modulo de adquisición.

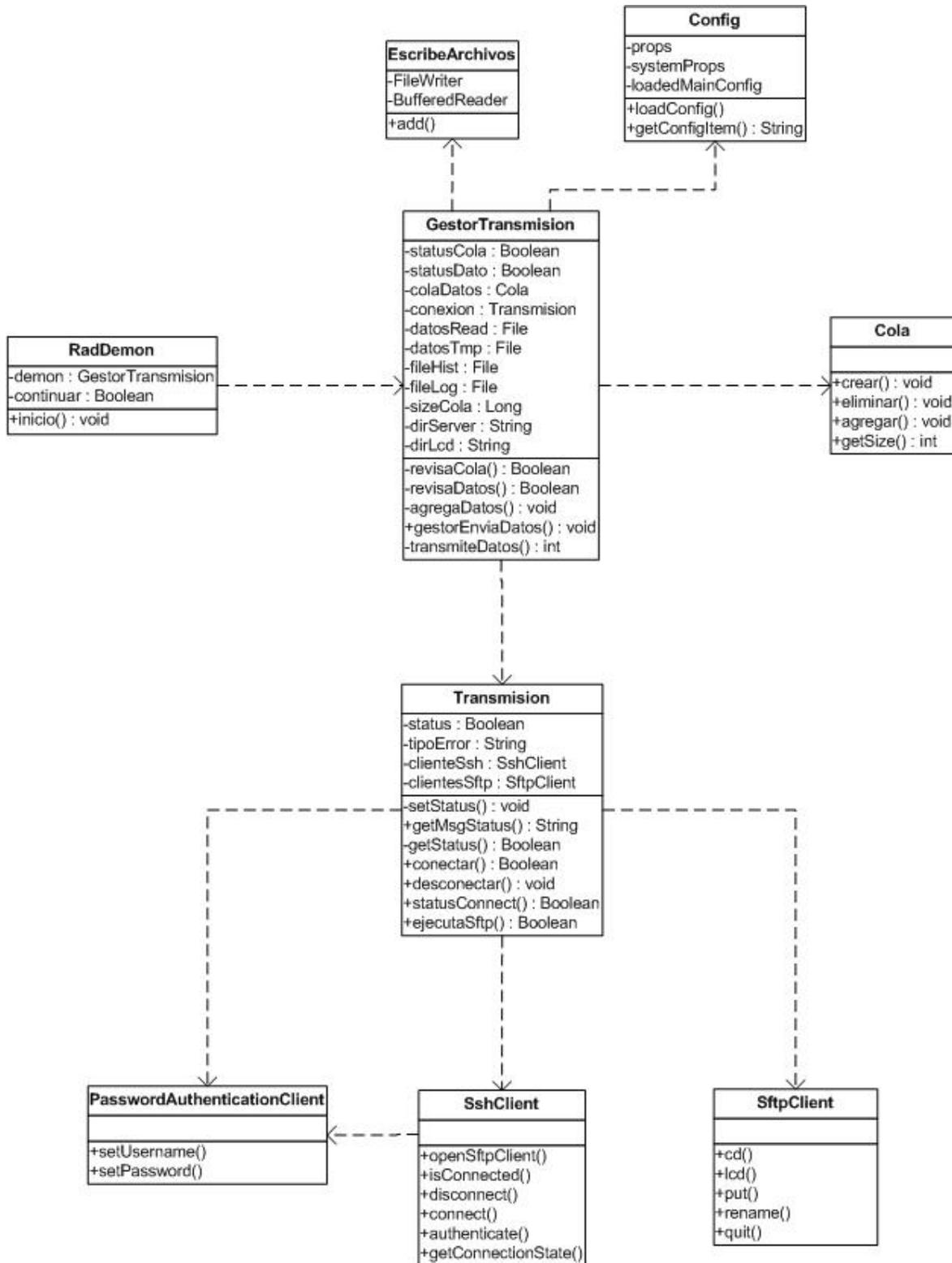


Figura IV.2.3 Diagrama de clases para el modulo de transmisi3n.

IV.3 Diagramas de secuencias.

Los diagramas de secuencia nos serán útiles para visualizar la interacción que a través del tiempo tienen los objetos y actores para llevar a cabo alguna colaboración. En esta colaboración, la mayoría de las veces está sustentada por la definición de un caso de uso. A continuación se muestran los diagramas de secuencias.

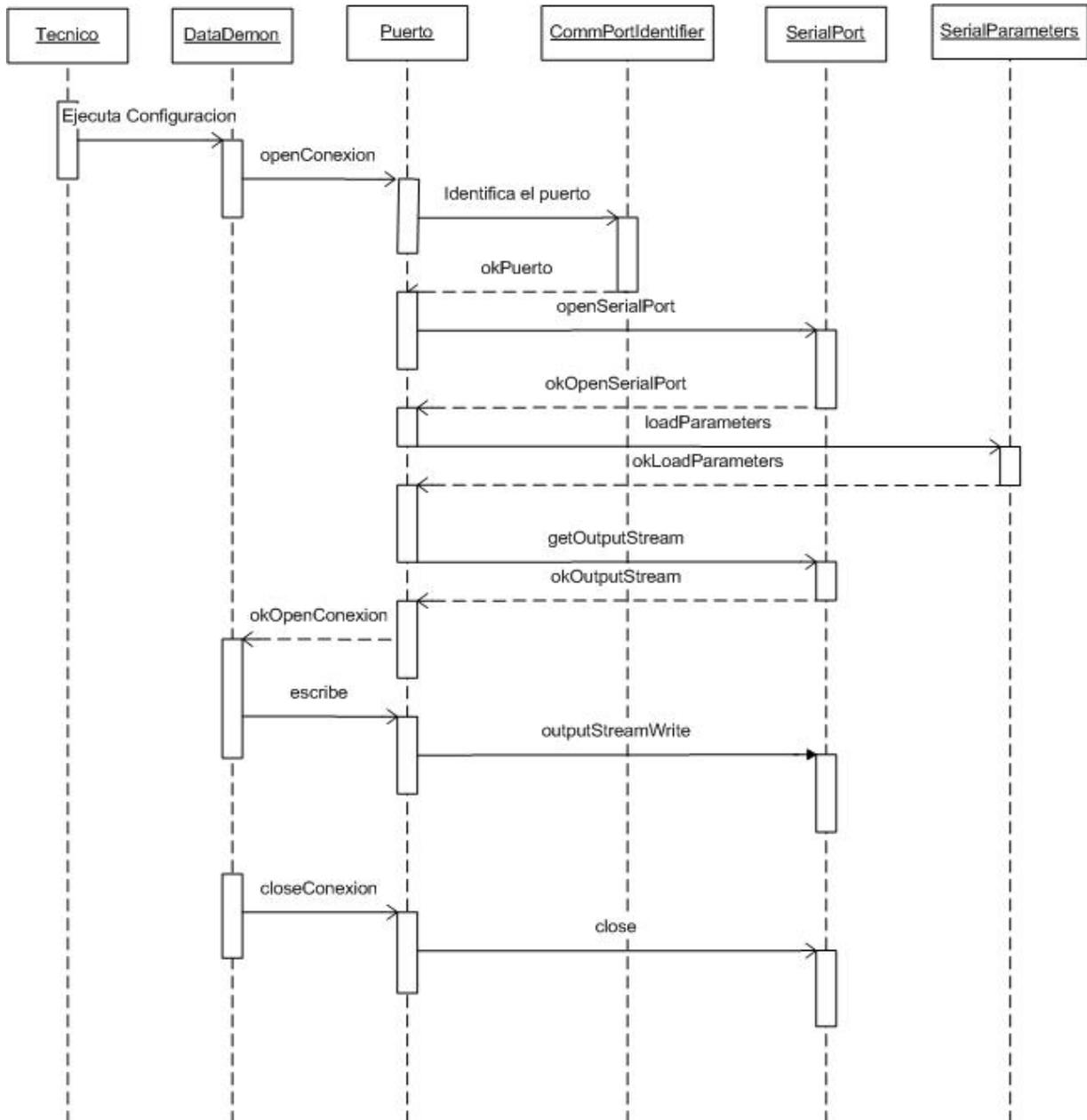


Figura IV.3.1 Diagrama de secuencias para el caso de uso *Configura aparato de adquisición*.

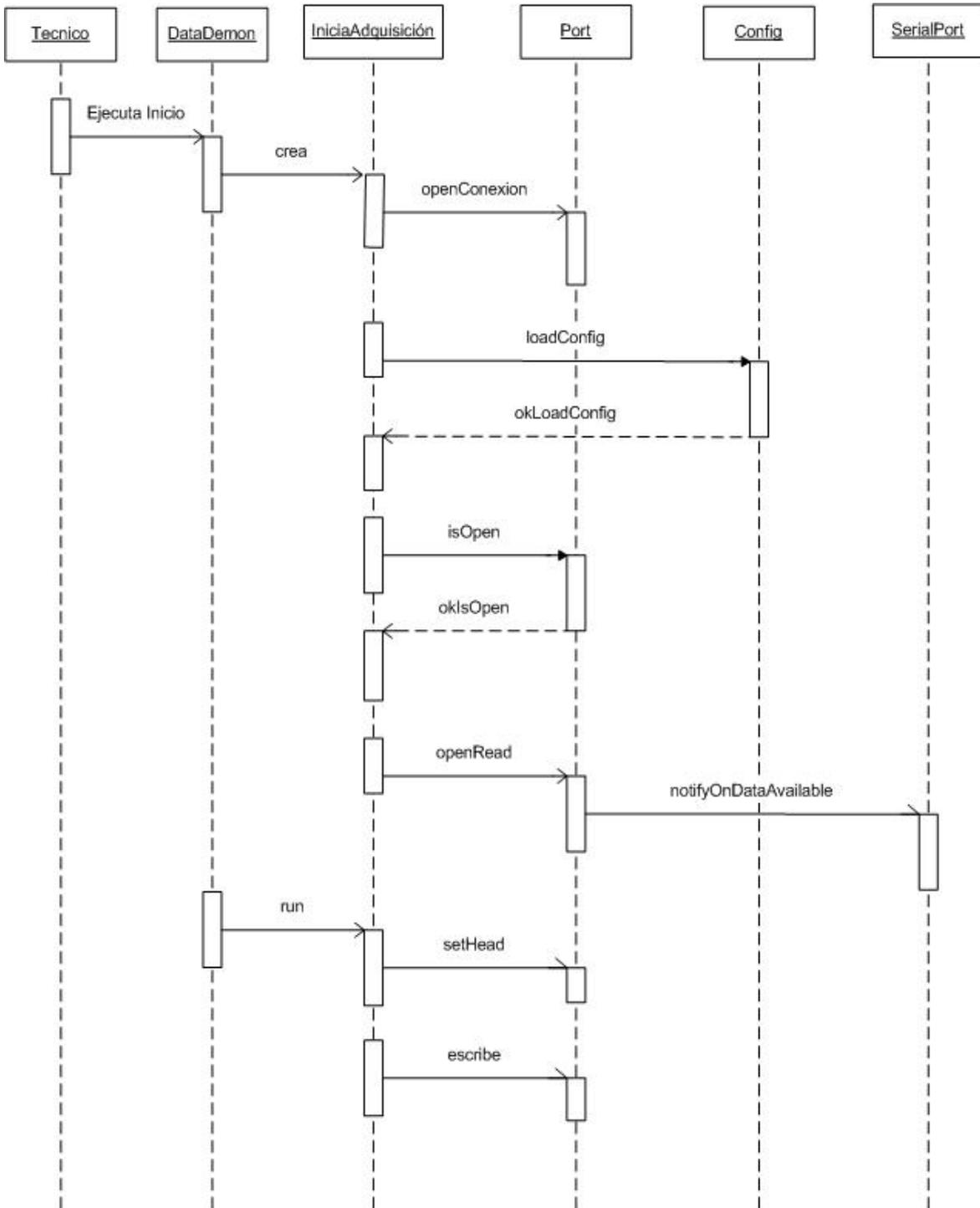


Figura IV.3.2 Diagrama de secuencias para el caso de uso *Inicia proceso*.

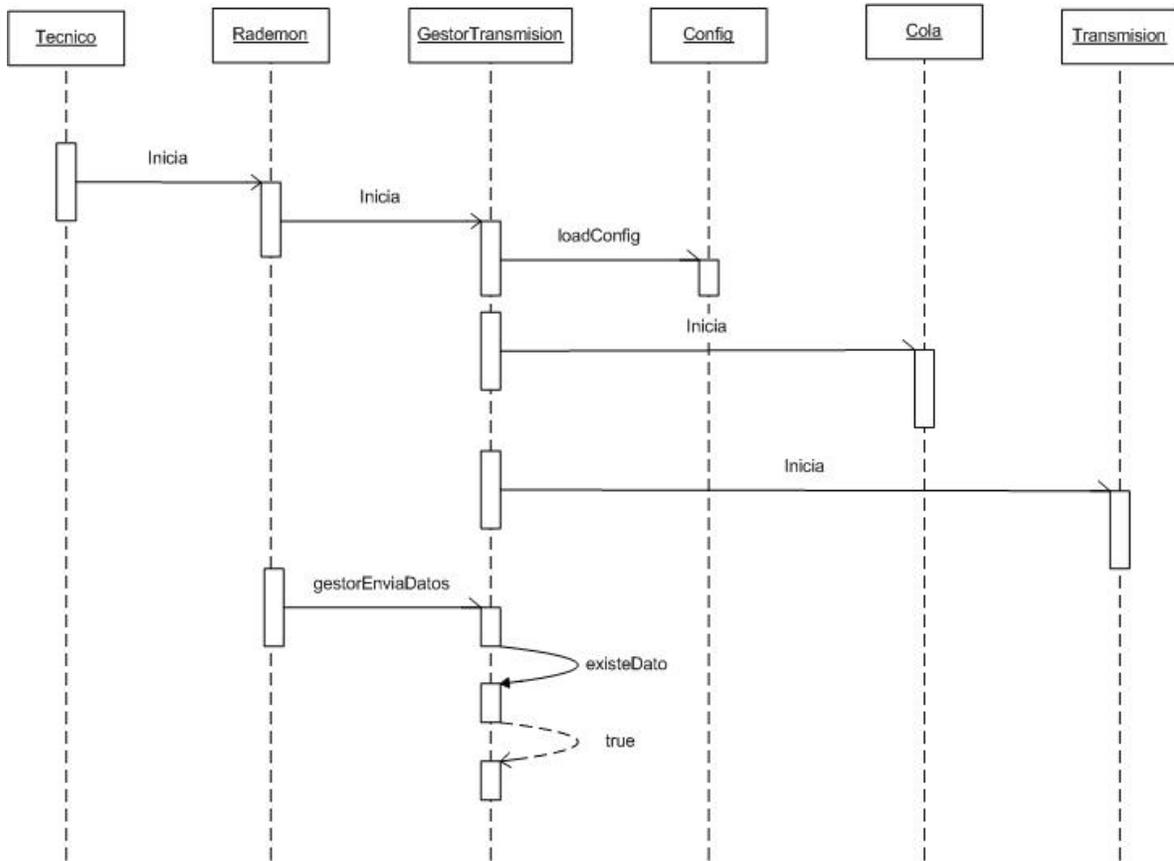


Figura IV.3.3 Diagrama de secuencias para el caso de uso *Verifica existencia de datos*.

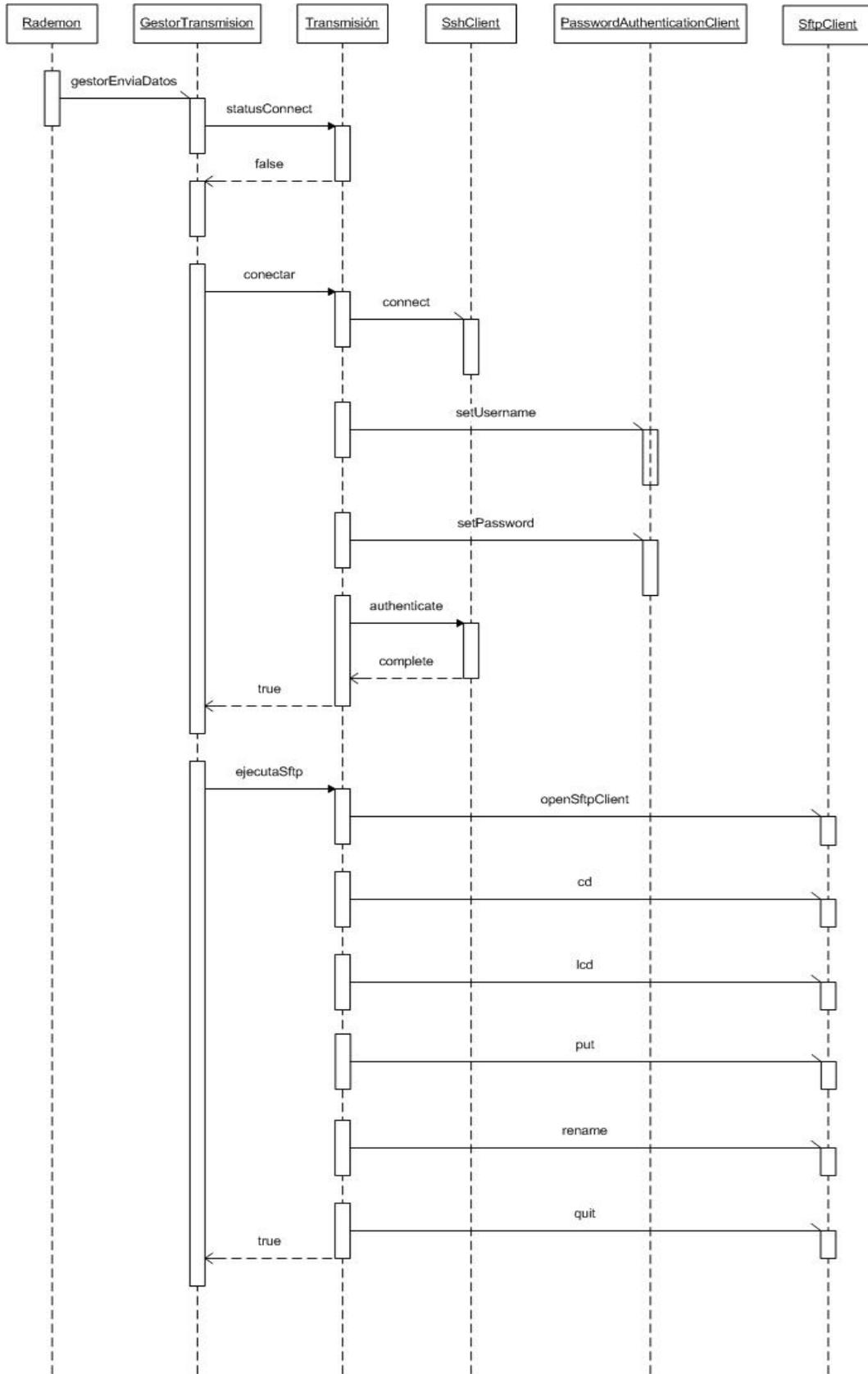


Figura IV.3.4 Diagrama de secuencias para el caso de uso *Transmisión*.

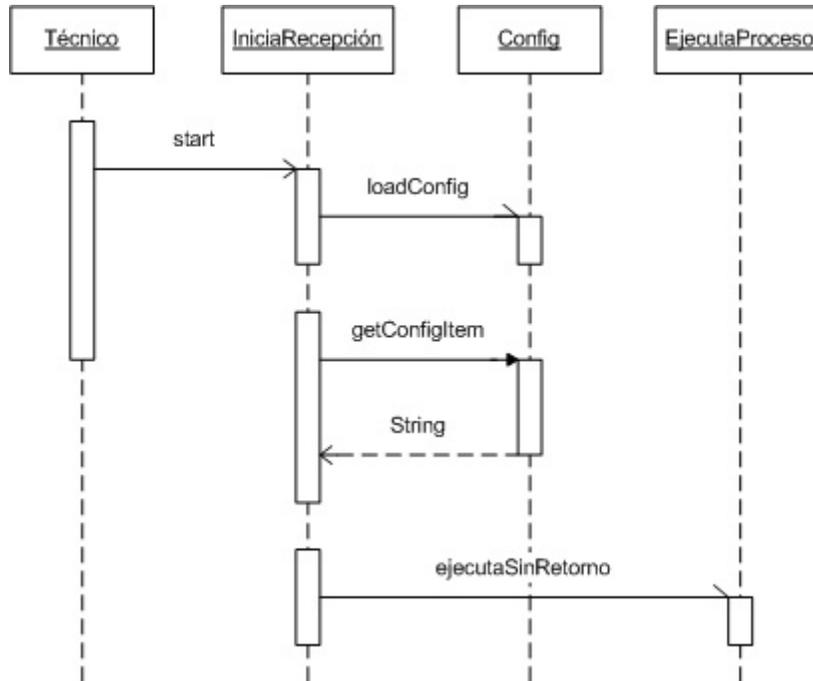


Figura IV.3.5 Diagrama de secuencias para el caso de uso *Controla recepción de datos remotos*.

IV.4 Diagramas de actividad.

En los diagramas de actividad vamos a observar el flujo de trabajo que se sigue para concluir una colaboración y de qué forma se puede implementar alguna operación, en otras palabras, estos diagramas describen como se coordinan las actividades. Es importante señalar que los diagramas de actividad también son útiles para describir como es la presentación de un conjunto de casos de uso.

A continuación se presentan los diagramas de actividad, para el flujo de trabajo que se realiza en la adquisición de datos, transmisión de datos y recepción de datos.

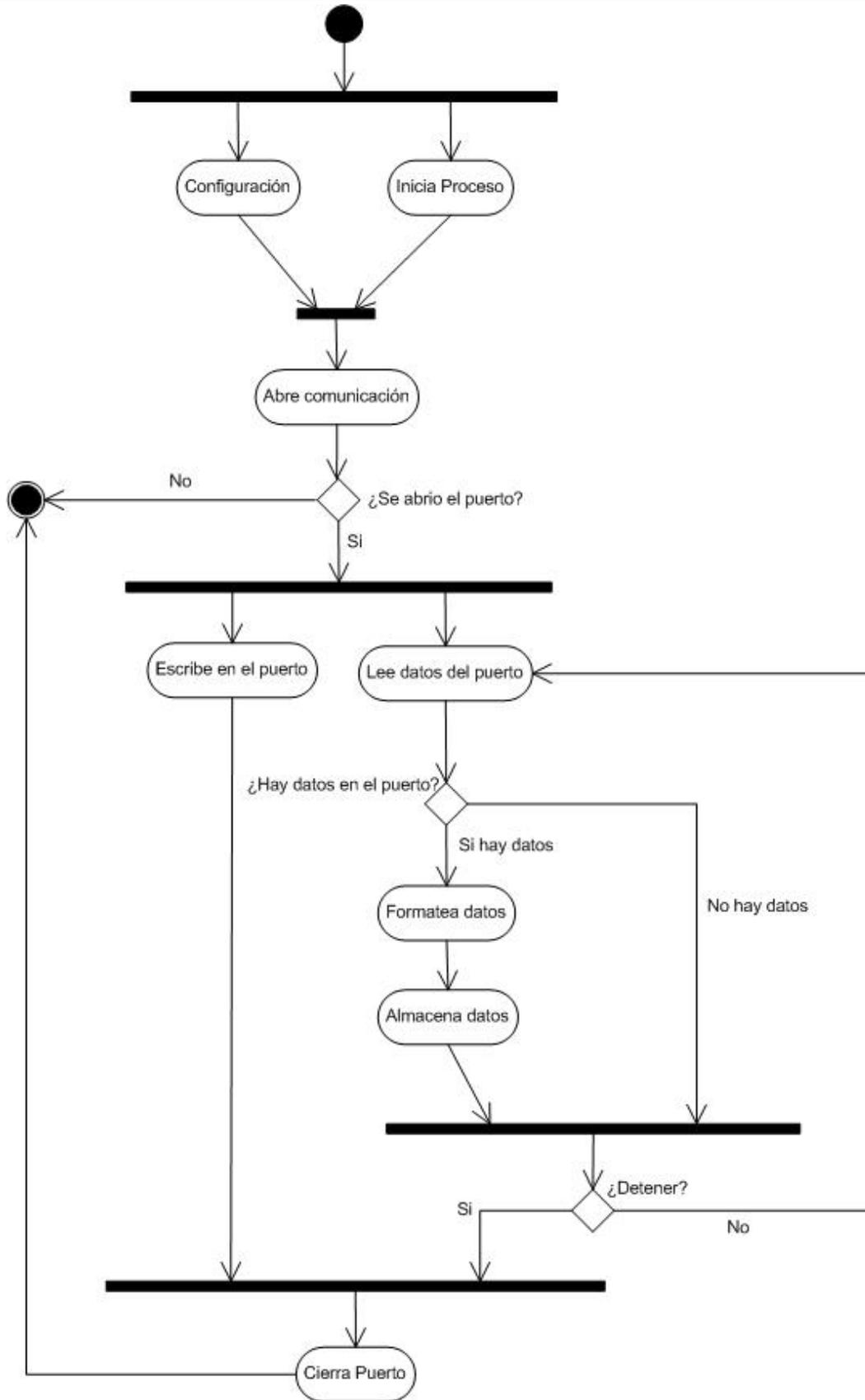


Figura IV.4.1 Diagrama de actividad para el proceso de adquisición.

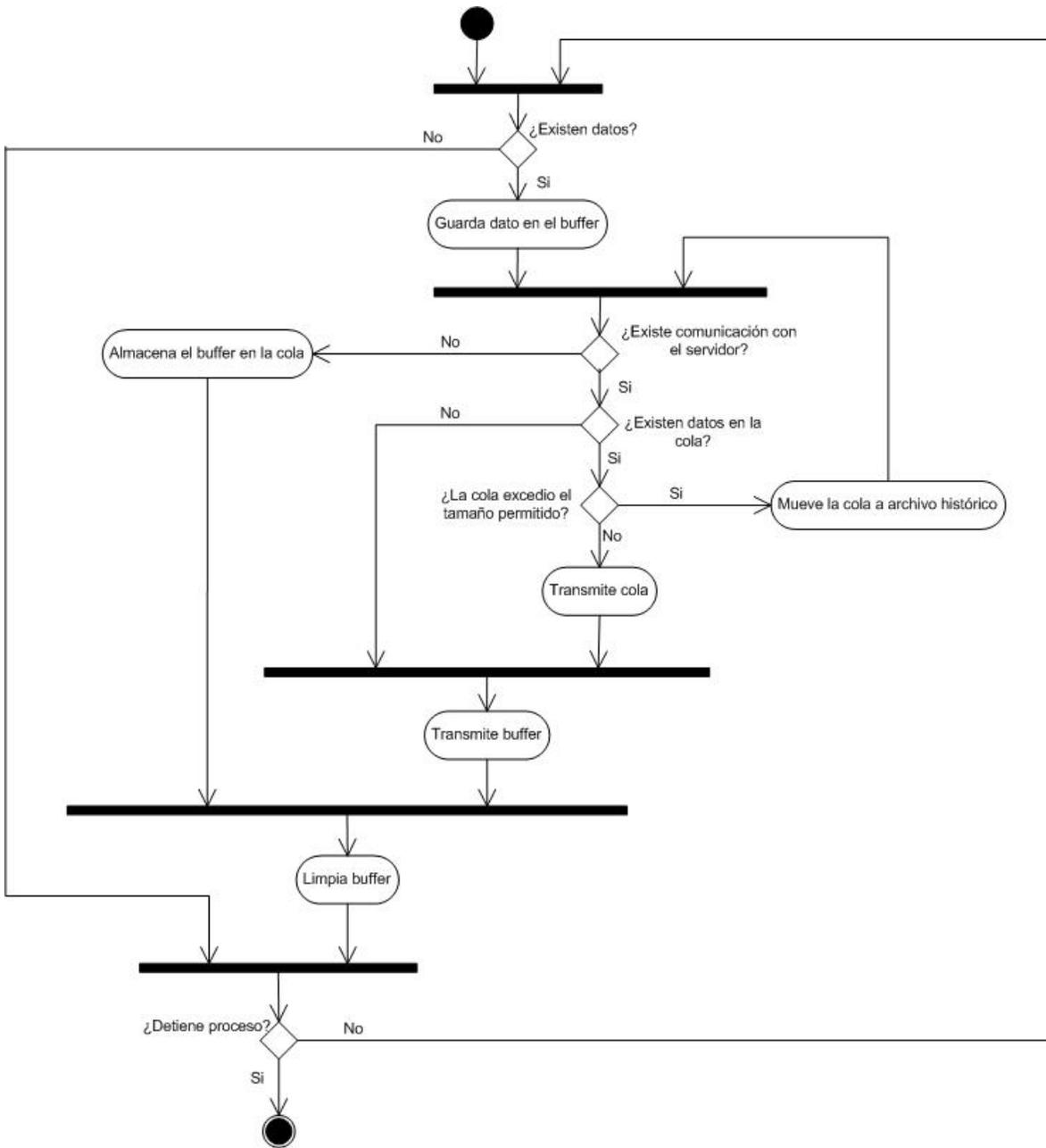


Figura IV.4.2 Diagrama de actividad para el proceso de transmisión.

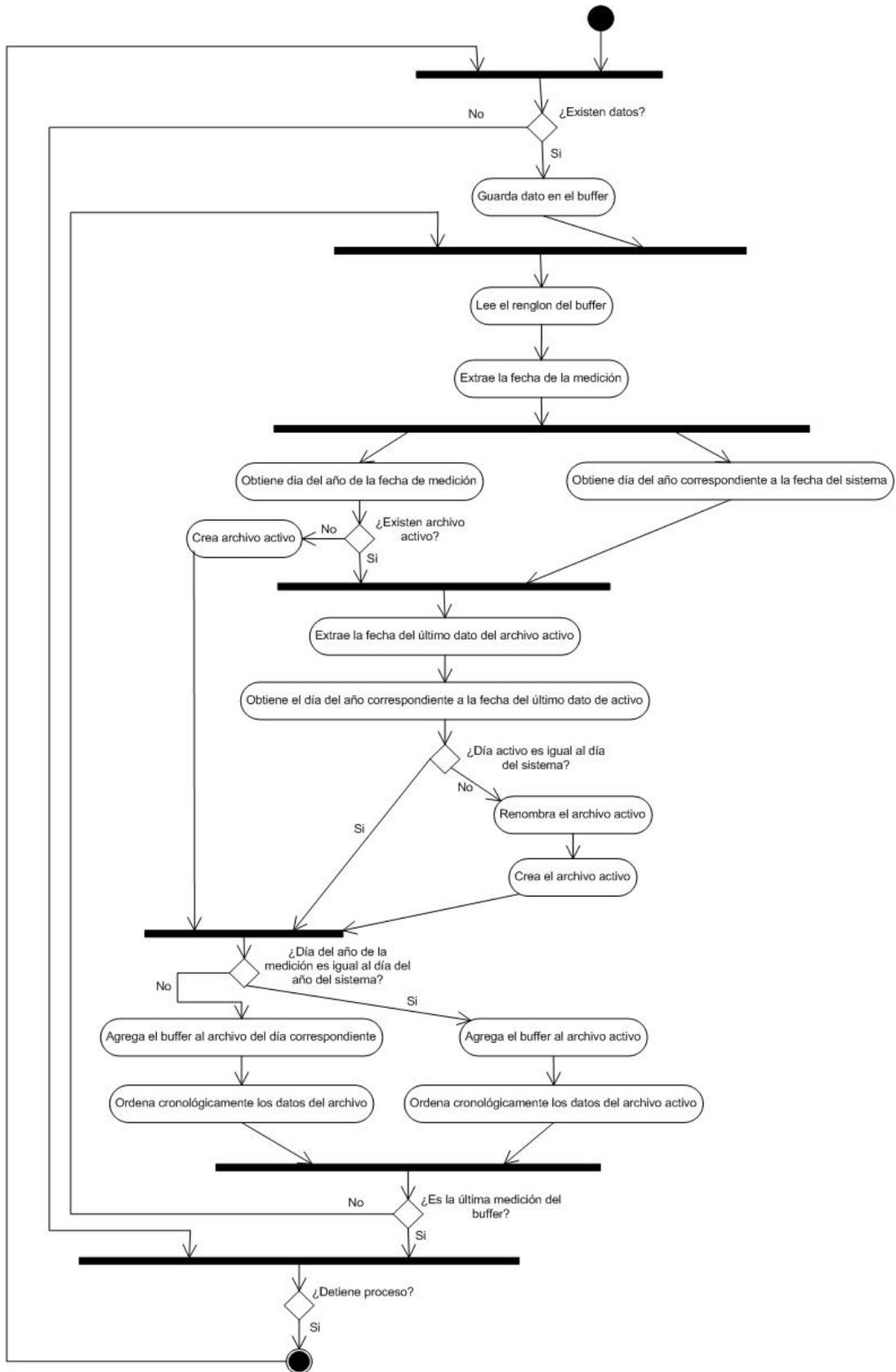


Figura IV.4.3 Diagrama de actividad para el proceso de recepción.

Capítulo V. Diseño.

V.1 Requerimientos en software y hardware.

En este capítulo se muestra la forma en que se va a desarrollar el sistema dentro de un modelo de software con sus respectivos componentes de hardware, considerando las especificaciones, los requerimientos funcionales y el comportamiento que se estableció en los dos capítulos anteriores.

En la etapa de adquisición y transmisión de los datos, se opta por utilizar el lenguaje de programación en Java, debido a que cuenta con las siguientes características:

- Es un lenguaje independiente de la plataforma, pues los programas una vez que han sido compilados, se generan archivos (ByteCode) los cuáles son interpretados por una máquina virtual. Esta máquina virtual hoy en día está disponible de forma gratuita para la mayoría de los sistemas operativos, mencionando los más conocidos como Windows, Mac, Linux y Solaris.
- Existe un paquete de forma gratuita, que incorpora clases que permiten realizar conexiones con puertos de entrada/salida, como es el caso del puerto serie, el cual en esta tesis se va a utilizar.
- También se encuentra disponible de forma gratuita, un conjunto de clases que son utilizadas para la transmisión de datos a través de Internet, utilizando SSH como protocolo de comunicación entre diferentes computadoras de manera segura.

En esta misma etapa se desarrollará bajo el sistema operativo Linux, en su distribución Fedora Core4. Se eligió esta plataforma ya que cuenta con licencia GNU que permite su uso sin costo alguno y además ofrece soporte para el lenguaje de programación que se va a utilizar.

Como se va a trabajar con java, necesitamos del lado del cliente, el siguiente conjunto de herramientas de software:

1. JDK1.5.0
2. Java Communication 3.0 o superior API para Linux
3. Sshools - Java SSH2 API

Trasladándonos hacia el otro lado del modelo de arquitectura elegido (recepción y almacenamiento de los datos), se intenta manipular los archivos de mediciones que se van a estar recibiendo, así como su contenido, sin olvidar que la plataforma en la cual se van a recibir los datos, tiene que ser bajo un servidor con Linux, esto debido a la infraestructura con la que cuenta el ORS., por lo tanto, se decide utilizar también el lenguaje de programación con Java conjuntado con el lenguaje de programación Perl. A continuación se exponen algunos puntos por lo cual se decidió a utilizar Perl como lenguaje de programación para esta etapa:

- Perl al igual que Java es de tipo Open Source o de código abierto y puede trabajar bajo plataformas como Windows, Mac, Solaris y Linux, donde este último sistema operativo es el que nos interesa.
- Perl cuenta con una gran variedad de características que permiten el fácil uso de manipulación de archivos de texto y procesos, que en este caso es lo que se requiere.
- Por su facilidad de uso, Perl permite que sea utilizado por muchos desarrolladores, para la realización de programas que requieran desarrollarse de manera rápida.

En el caso de esta capa del modelo de arquitectura elegido, como ya se había mencionado anteriormente, el ORS cuenta con un servidor Linux, que tiene la distribución Fedora-Core4.

Por último en cuanto al software se refiere, del lado del servidor se necesita tener instalado el servicio de SSH. Para este caso se tiene instalado OpenSSH en su versión 7.0. Este servicio por lo general ya viene dentro de la distribución de Linux que se instaló, su uso es libre, pues también está bajo los términos de licencia GNU.

También, se requiere del paquete de Java Standar Edition (Java SE) en su versión 4 o superior.

En cuanto a los requerimientos en hardware para la etapa de adquisición y recepción se tiene en cuenta que una de las necesidades del observatorio, es la de utilizar el aparato de adquisición modelo HP 34970A. Este dispositivo cuenta con dos tipos de interfaz para poder conectar una computadora a estos puertos de entrada/salida, los cuales son el HP-IB y el RS-232. En este caso se va a utilizar la interfaz RS-232 y no la interfaz HP-IB por lo siguiente:

- Se tienen las herramientas en software que van a permitir la lectura y escritura de datos por medio del puerto serie (estándar RS-232).
- En la actualidad podemos encontrar fácilmente y a bajo costo en el mercado, una tarjeta de puerto serie que una tarjeta para PC con la interfaz HP-IB.
- Si en un futuro se requiere controlar el aparato de adquisición por medio de un cambiador de medio Ethernet a serial, con la interfaz RS-232 se puede realizar.

Se requiere tener una computadora personal como medio de respaldo para almacenar todas las lecturas de datos que se vayan registrando por el aparato de adquisición, y la otra razón de esta elección es que la computadora será la terminal encargada de transmitir esos datos hacia el servidor de almacenamiento.

A continuación se especifican los requerimientos mínimos que se necesitan para operar el modulo de adquisición y recepción.

- Pentium III a 800MHZ, 128 MB RAM, disco duro de 10GB, un puerto serie, tarjeta de red o modem dependiendo de la forma de conexión a Internet.

Del lado de la etapa de recepción y almacenamiento de datos, el servidor cuenta con las siguientes características de hardware:

- Pentium III, 512 MB RAM, tarjeta de red con una dirección IP homologada.

V.2 Secuencia de pantallas.

En el modulo de adquisición, abriendo una terminal de comandos se puede ejecutar el siguiente comando:

datademon

Escrito este comando en la terminal sin ningún parámetro o con un parámetro inválido, se despliega la siguiente pantalla como lo muestra la figura V.2.1.

```

[ivan@rori serie]$ datademon
#####
#
#
#           Parametros de Operacion del sistema SIORS           #
#   Favor de ejecutar:                                           #
#
#   datademon [ config | start | stop ]                          #
#
#   config: Enviar configuracion al HP34970A                     #
#   start:  Iniciar proceso de adquisicin                        #
#   stop:   Detener proceso de adquisicion                       #
#
#
#####
No hay argumentos
[ivan@rori serie]$ █

```

Figura V.2.1 Pantalla de bienvenida del programa.

Utilizando el comando con la opción de configuración se muestra la siguiente pantalla.

```

[ivan@rori serie]$ datademon config
Transmitidas las intrucciones de configuracion al HP34970A

config

```

Figura V.2.2 Salida en pantalla proporcionado por el comando *datademon config*.

Al realizar el inicio del proceso de adquisición se muestra la siguiente pantalla (figura V.2.3), donde cada línea de datos contiene las mediciones obtenidas por el aparato de adquisición, cada medición de un canal va separado por comas.

```
[ivan@rori serie]$ datademon start
Iniciando proceso de adquisicion
-2.44802100E-01,+1.04021000E-02,+9.98920000E-03,-1.18405000E-02,-2.62383000E-02
-3.17523000E-02,-6.77930000E-03,-9.13680000E-03,-1.27728000E-02,-2.75169000E-02
-3.44294000E-02,-1.98052000E-02,-1.88596000E-02,-1.68351000E-02,-2.25356000E-02
-3.41364000E-02,-2.12836000E-02,-2.16566000E-02,-2.01116000E-02,-2.08974000E-02
-3.02206000E-02,-2.22293000E-02,-2.32415000E-02,-2.27088000E-02,-2.11771000E-02
-3.04337000E-02,-2.32282000E-02,-2.36944000E-02,-2.34280000E-02,-2.10705000E-02
-3.32573000E-02,-2.41339000E-02,-2.48131000E-02,-2.32682000E-02,-2.02581000E-02
```

Figura V.2.3 Salida en pantalla del comando *datademon start*, que muestra la lectura de datos provenientes del HP34970A.

Para el proceso de transmisión de datos, al igual que el proceso de adquisición, hay que abrir una terminal de comandos y ejecutar la siguiente instrucción:

raddemon

```
[ivan@rori transmision]$ raddemon
Revisando llegada de datos.....
```

Figura V.2.4 Salida en pantalla, después de haber ejecutado el comando *raddemon*, para la transmisión de datos.

Como se puede apreciar en la siguiente imagen, cada línea de las mediciones tienen incorporadas el nombre de la estación, día, mes, año y la hora en que se realizó dicha medición.

```

Se esta escribiendo jarbo,12/02/2007,21:34:58,-3.14326000E-02,-2.26155000E-02,-2.53193000E-02,-2.51062000E-02,-2.06710000E-02
Se esta escribiendo
Se esta escribiendo jarbo,12/02/2007,21:35:03,-3.52552000E-02,-2.37476000E-02,-2.21893000E-02,-2.31616000E-02,-2.04445000E-02
Se esta escribiendo
Se esta escribiendo jarbo,12/02/2007,21:35:08,-3.66004000E-02,-2.65446000E-02,-2.39607000E-02,-2.26821000E-02,-1.96188000E-02
Se esta escribiendo
Se esta escribiendo jarbo,12/02/2007,21:35:13,-3.13394000E-02,-2.35878000E-02,-2.38675000E-02,-2.38675000E-02,-2.08707000E-02
Se esta escribiendo
Se esta escribiendo jarbo,12/02/2007,21:35:18,-3.09798000E-02,-2.38675000E-02,-2.54791000E-02,-2.40540000E-02,-2.03380000E-02
Se esta escribiendo

```

Figura V.2.5 Presentación de las mediciones el cual incluye el nombre de la estación fecha y hora de las lecturas de los canales de medición.

Al completarse la transmisión de los datos hacia el servidor se visualiza en la figura V.2.6.

```

INFO: Negotiating protocol version
13-feb-2007 21:10:45 com.ssh.tools.j2ssh.transport.TransportProtocolCommon negotiateVersion
INFO: Protocol negotiation complete
13-feb-2007 21:10:45 com.ssh.tools.j2ssh.transport.TransportProtocolCommon beginKeyExchange
INFO: Starting key exchange
13-feb-2007 21:10:45 com.ssh.tools.j2ssh.transport.kex.DhGroup1Sha1 performClientExchange
INFO: Starting client side key exchange.
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.transport.AbstractKnownHostsKeyVerification verifyHost
INFO: Verifying 132.248.6.83 host key
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.transport.TransportProtocolClient verifyHostKey
INFO: The host key signature is valid
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.transport.TransportProtocolCommon completeKeyExchange
INFO: Completing key exchange
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.transport.cipher.SshCipherFactory newInstance
INFO: Creating new blowfish-cbc cipher instance
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.transport.cipher.SshCipherFactory newInstance
INFO: Creating new blowfish-cbc cipher instance
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.connection.ConnectionProtocol onServiceInit
INFO: Registering connection protocol messages
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.transport.Service start
INFO: ssh-connection has been requested
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.transport.AsyncService onStart
INFO: Starting ssh-connection service thread
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.connection.ConnectionProtocol openChannel
INFO: Channel 0 is open [session]
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.connection.ConnectionProtocol openChannel
INFO: Channel 1 is open [sftp]
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.sftp.SftpSubsystemClient initialize
INFO: Initializing SFTP protocol version 3
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.subsystem.SubsystemChannel startSubsystem
INFO: Starting sftp subsystem
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.sftp.SftpSubsystemClient initialize
INFO: Server responded with version 3
13-feb-2007 21:10:46 com.ssh.tools.j2ssh.sftp.SftpSubsystemClient getOKRequestStatus
INFO: Received response

```

Figura V.2.6 Pantalla que muestra el envío de datos hacia el servidor.

En el proceso de recepción de datos, se utiliza también una terminal de comandos del sistema, para poder ejecutar el comando siguiente:

radcom jarbo

En el comando anterior se puede observar que se le especifica un parámetro, donde en este caso es *jarbo*, determinando este último el nombre o clave de la estación a monitorear.

En la siguiente pantalla (figura V.2.7) se muestra la acción del programa *radcom* cuando no se le introduce ningún parámetro.

```
[ivan@niebo-gps recepcion]$ radcom
Sintaxis erronea, Utilizar:
radcom [nombre_de_estacion]
[ivan@niebo-gps recepcion]$ █
```

Figura V.2.7 Salida en pantalla del comando *radcom* cuando no se le especifica el nombre de la estación a monitorear.

En la figura V.2.8 se muestra el resultado del comando *radcom* cuando se le especifica una estación que no existe o no está configurada.

```
[ivan@niebo-gps recepcion]$ radcom xyz
No existe el archivo de configuracion de la estacion xyz
El proceso de recepcion no pudo ser completado

[ivan@niebo-gps recepcion]$ █
```

Figura V.2.8 Salida en pantalla del comando *radcom* cuando no encuentra el archivo de configuración de la estación.

La operación de recepción de datos, por el programa *radcom* se ilustra en la siguiente figura.

En la figura V.2.11 se visualiza parte del contenido del archivo *jarbo.20070208.039*. Las mediciones tienen un intervalo de 5 segundos entre cada medición.

```

ivan@niebo-gps:~/rt/2006/jarbo
Archivo Editar Ver Terminal Solapas Ayuda
[ivan@niebo-gps jarbo]$ more jarbo.20070208.039
jarbo,08/02/2007,14:35:34,-1.46641500E-01,-5.86832000E-02,-4.15683000E-02,-3.91709000E-02,-3.87980000E-02
jarbo,08/02/2007,14:35:39,-5.94157000E-02,-4.32332000E-02,-3.04337000E-02,-3.17656000E-02,-1.32124000E-02
jarbo,08/02/2007,14:35:44,-4.57238000E-02,-3.16857000E-02,-2.70241000E-02,-2.91951000E-02,-1.64089000E-02
jarbo,08/02/2007,14:35:49,-3.67602000E-02,-2.82894000E-02,-2.55990000E-02,-2.66512000E-02,-1.96054000E-02
jarbo,08/02/2007,14:35:54,-3.74128000E-02,-3.01141000E-02,-2.47066000E-02,-2.40406000E-02,-1.91659000E-02
jarbo,08/02/2007,14:35:59,-3.62008000E-02,-2.34413000E-02,-2.26555000E-02,-2.42404000E-02,-2.05378000E-02
jarbo,08/02/2007,14:36:04,-3.43229000E-02,-2.57987000E-02,-2.43337000E-02,-2.30550000E-02,-2.00317000E-02
jarbo,08/02/2007,14:36:09,-3.64672000E-02,-2.32149000E-02,-2.26022000E-02,-2.28952000E-02,-1.99651000E-02
jarbo,08/02/2007,14:36:14,-3.19787000E-02,-2.30950000E-02,-2.29885000E-02,-2.35212000E-02,-2.04978000E-02
jarbo,08/02/2007,14:36:19,-3.09265000E-02,-2.34946000E-02,-2.38009000E-02,-2.34813000E-02,-2.12437000E-02
jarbo,08/02/2007,14:36:24,-3.56681000E-02,-2.61051000E-02,-2.28153000E-02,-2.25356000E-02,-1.98718000E-02
jarbo,08/02/2007,14:36:29,-3.18455000E-02,-2.26022000E-02,-2.20961000E-02,-2.27754000E-02,-2.03247000E-02
jarbo,08/02/2007,14:36:34,-2.96479000E-02,-2.31882000E-02,-2.34280000E-02,-2.35212000E-02,-2.19629000E-02
jarbo,08/02/2007,14:36:39,-3.44028000E-02,-2.42005000E-02,-2.31882000E-02,-2.21094000E-02,-2.02448000E-02
jarbo,08/02/2007,14:36:44,-3.25781000E-02,-2.07908000E-02,-2.21760000E-02,-2.34546000E-02,-2.12437000E-02
jarbo,08/02/2007,14:36:49,-3.22451000E-02,-2.44003000E-02,-2.37210000E-02,-2.23358000E-02,-2.11105000E-02
jarbo,08/02/2007,14:36:54,-3.20986000E-02,-2.19629000E-02,-2.12836000E-02,-2.23491000E-02,-2.06177000E-02
jarbo,08/02/2007,14:36:59,-3.05802000E-02,-2.37077000E-02,-2.41072000E-02,-2.27088000E-02,-2.13635000E-02
jarbo,08/02/2007,14:37:04,-3.45626000E-02,-2.30151000E-02,-2.19762000E-02,-2.18697000E-02,-2.02581000E-02
jarbo,08/02/2007,14:37:09,-3.05403000E-02,-2.06843000E-02,-2.26954000E-02,-2.33880000E-02,-2.20029000E-02
jarbo,08/02/2007,14:37:14,-3.24982000E-02,-2.46267000E-02,-2.32682000E-02,-2.20428000E-02,-2.09240000E-02
jarbo,08/02/2007,14:37:19,-3.30442000E-02,-2.17764000E-02,-2.08574000E-02,-2.21627000E-02,-2.06843000E-02
--More-- (19%)

```

Figura V.2.11 Contenido del archivo de las mediciones obtenidas en la estación remota.

Capítulo VI. Pruebas.

En este capítulo se dará a conocer las diferentes pruebas que contribuyeron a la verificación y validación del software, estas pruebas fueron aplicadas a los tres módulos que conforman el sistema, explicando de forma general el comportamiento y resultado de la aplicación de las pruebas, así como también se comenta la forma en que se encontró la solución a la falla. Para cada uno de los módulos desarrollados en este sistema (adquisición, transmisión y recepción) se presentan las pruebas por separado de acuerdo al modulo que se probó.

Como se menciona en el Capítulo II, las pruebas son una de las herramientas que nos permiten reducir la probabilidad de catalogar a un sistema como poco funcional, ya que si a un sistema no se le aplican pruebas de manera minuciosa y dirigida, los resultados del sistema en determinado momento pueden ser erróneos originando que el usuario o cliente quede insatisfecho en el mejor de los casos, ya que puede suceder que una falla grave, desencadene una serie de eventos que originen consecuencias de muy alto costo y/o que el daño sea irreparable.

VI.1 Pruebas a la etapa de Adquisición.

El primer objetivo que se tenía que cumplir, era el de poder enviar al aparato de adquisición a través del puerto serie, un conjunto de instrucciones (comandos SCPI) para configurar el instrumento. En un principio se mandaba solo una instrucción por ejecución y el resultado era satisfactorio, pero al querer incluir todo un set de instrucciones en una sola ejecución dentro de un ciclo, el instrumento arrojaba mensajes de error de tipo *error de sintaxis*, *carácter inválido* y *separador inválido*. Resultaba poco práctico tener que realizar una ejecución por cada instrucción para configurar el instrumento. La solución a este problema fue el de agregar al final de cada instrucción el carácter de nueva línea ($\backslash n$), con esto el instrumento puede discernir entre cada instrucción que se le suministra.

El segundo objetivo que había que realizar, era el de obtener las mediciones que arrojaba el aparato de adquisición y almacenarlas en un archivo en forma secuencial, en esta parte todavía no se consideró el formato que iban a tener los datos, es importante

señalar que la salida de los datos por parte del instrumento era en código ASCII, lo que facilitó la obtención de estos mismos. Para realizar esta prueba se configuró el instrumento para que cada 10 segundos tomara una muestra de 7 canales de medición. Los resultados fueron satisfactorios, la PC almacenaba correctamente cada lectura del instrumento en un solo renglón del archivo, es decir, las siete mediciones por cada 10 segundos eran separadas por comas todas en un solo renglón. En la siguiente figura se muestra el resultado de la prueba.

```
-5.22768000E-02,-3.63340000E-02,-2.49463000E-02,-2.45468000E-02,-2.51594000E-02,-2.41738000E-02,-1.91260000E-02
-2.99676000E-02,-2.07109000E-02,-2.36944000E-02,-2.24557000E-02,-2.37876000E-02,-2.36011000E-02,-2.25756000E-02
-3.20586000E-02,-1.93391000E-02,-2.25756000E-02,-2.14967000E-02,-2.38009000E-02,-2.28686000E-02,-2.26155000E-02
-3.23916000E-02,-2.01515000E-02,-2.20428000E-02,-2.24024000E-02,-2.22026000E-02,-2.34413000E-02,-2.29485000E-02
-3.03139000E-02,-2.19363000E-02,-2.31616000E-02,-2.35612000E-02,-2.26954000E-02,-2.22959000E-02,-2.28419000E-02
-3.22717000E-02,-2.32016000E-02,-2.29751000E-02,-2.20561000E-02,-2.26155000E-02,-2.24823000E-02,-2.27887000E-02
-3.18189000E-02,-2.13103000E-02,-2.29485000E-02,-2.17365000E-02,-2.25489000E-02,-2.24157000E-02,-2.26288000E-02
-2.48531000E-02,-1.00025000E-02,-2.10838000E-02,-2.19096000E-02,-2.28819000E-02,-2.25489000E-02,-2.42271000E-02
-3.74395000E-02,-3.00608000E-02,-2.30018000E-02,-2.15500000E-02,-2.08441000E-02,-2.15101000E-02,-2.20295000E-02
-3.12595000E-02,-2.35478000E-02,-2.33614000E-02,-2.14834000E-02,-2.23891000E-02,-2.16566000E-02,-2.29219000E-02
-3.10597000E-02,-2.24024000E-02,-2.34546000E-02,-2.37343000E-02,-2.29219000E-02,-2.25356000E-02,-2.29352000E-02
-3.24848000E-02,-2.23625000E-02,-2.29751000E-02,-2.18297000E-02,-2.30018000E-02,-2.28952000E-02,-2.19096000E-02
-3.05669000E-02,-1.66486000E-02,-2.24690000E-02,-2.17764000E-02,-2.09240000E-02,-2.29885000E-02,-2.40939000E-02
-3.14593000E-02,-2.24024000E-02,-2.19629000E-02,-2.18697000E-02,-2.24557000E-02,-2.07908000E-02,-2.27354000E-02
-1.82070000E-02,-2.78099000E-02,-2.66378000E-02,-2.54391000E-02,-2.90885000E-02,-2.79165000E-02,-3.01407000E-02
-3.81187000E-02,-2.51728000E-02,-2.35345000E-02,-2.32149000E-02,-2.44402000E-02,-2.41605000E-02,-2.28153000E-02
-3.79989000E-02,-2.35478000E-02,-2.30550000E-02,-2.25356000E-02,-2.42804000E-02,-2.31350000E-02,-2.12170000E-02
-3.88646000E-02,-2.57588000E-02,-2.34679000E-02,-2.24690000E-02,-2.31749000E-02,-2.40406000E-02,-2.34413000E-02
-3.72930000E-02,-2.46933000E-02,-2.37343000E-02,-2.32548000E-02,-2.31216000E-02,-2.32282000E-02,-2.20961000E-02
```

Figura VI.1.1 Contenido del archivo mediciones presentando únicamente las lecturas de los canales de medición.

Al incorporarle al principio de la línea de mediciones el nombre de la estación, la fecha y hora de medición y al final de ésta el carácter de nueva línea (“\n”), los resultados ya no fueron los mismos, pues se visualizaba lo siguiente en el archivo de recepción de datos.

```
jarbo 02/02/2007 20:45:53 -1.59414
300E-01
,-4.5657
3000E-02
,-4.0795
8000E-02
,-3.9224
2000E-02
,-3.7079
9000E-02
,-4.2340
8000E-02
,-3.5921
1000E-02
,-3.6334
0000E-02
```

Figura VI.1.2 Contenido del archivo de lectura de mediciones después de haberle agregado el nombre de la estación, fecha y hora de medición.

Como se puede observar, los datos de cada medición ya no eran almacenados en una sola línea. El problema se debía a que al llenarse el buffer de datos del puerto serie, inmediatamente se mandaba a ejecutar el método *add* de la clase *EscribeArchivos*, el cual escribe los datos del buffer al archivo, por lo tanto, si se utilizaba inmediatamente otra vez el método *add* para adjuntar el carácter de nueva línea sin esperar a que terminara toda la secuencia de datos provenientes del puerto, pues esta nueva línea era escrita en el archivo, posteriormente se vacía el buffer y nuevamente comenzaba a leer los datos del puerto; se llenaba nuevamente el buffer, se escribían los datos y una vez más se ejecutaba el método *add* para escribir el carácter de nueva línea y así sucesivamente.

La solución más rápida y confiable fue que por medio de otra clase generar el encabezado (nombre de la estación, fecha y hora) que debe tener la línea, por otro lado en el método que lee los datos del puerto serie, se unió al encabezado con el primer flujo de datos provenientes del puerto, acto seguido se utiliza el método *add* para escribir los datos al archivo, posteriormente se inicializa a una cadena vacía el encabezado, para que en el siguiente ciclo de arribo de datos, se vaya escribiendo en el archivo únicamente las mediciones ya que el encabezado ahora estará vacío. Realizando esta corrección los resultados fueron muy satisfactorios, en la figura VI.1.3 se muestra la presentación de los datos.

```
jarbo 12/10/2006 21:05:01 -5.22768000E-02,-3.63340000E-02,-2.49463000E-02,-2.45468000E-02,-2.51594000E-02,-2.41738000E-02,-1.91260000E-02
jarbo 12/10/2006 21:05:06 -2.99676000E-02,-2.07109000E-02,-2.36944000E-02,-2.24557000E-02,-2.37876000E-02,-2.36011000E-02,-2.25756000E-02
jarbo 12/10/2006 21:05:11 -3.20586000E-02,-1.93391000E-02,-2.25756000E-02,-2.14967000E-02,-2.38009000E-02,-2.28686000E-02,-2.26155000E-02
jarbo 12/10/2006 21:05:16 -3.23916000E-02,-2.01515000E-02,-2.20428000E-02,-2.24024000E-02,-2.22026000E-02,-2.34413000E-02,-2.29485000E-02
jarbo 12/10/2006 21:05:21 -3.03139000E-02,-2.19363000E-02,-2.31616000E-02,-2.35612000E-02,-2.26954000E-02,-2.22959000E-02,-2.28419000E-02
jarbo 12/10/2006 21:05:26 -3.22717000E-02,-2.32016000E-02,-2.29751000E-02,-2.20561000E-02,-2.26155000E-02,-2.24823000E-02,-2.27887000E-02
jarbo 12/10/2006 21:05:31 -3.18189000E-02,-2.13103000E-02,-2.29485000E-02,-2.17365000E-02,-2.25489000E-02,-2.24157000E-02,-2.26288000E-02
jarbo 12/10/2006 21:05:36 -2.48531000E-02,-1.00025000E-02,-2.10838000E-02,-2.19096000E-02,-2.28819000E-02,-2.25489000E-02,-2.42271000E-02
jarbo 12/10/2006 21:05:41 -3.74395000E-02,-3.00608000E-02,-2.30018000E-02,-2.15500000E-02,-2.08441000E-02,-2.15101000E-02,-2.20295000E-02
jarbo 12/10/2006 21:05:46 -3.12595000E-02,-2.35478000E-02,-2.33614000E-02,-2.14834000E-02,-2.23891000E-02,-2.16566000E-02,-2.29219000E-02
jarbo 12/10/2006 21:05:51 -3.10597000E-02,-2.24024000E-02,-2.34546000E-02,-2.37343000E-02,-2.29219000E-02,-2.25356000E-02,-2.29352000E-02
jarbo 12/10/2006 21:05:56 -3.24848000E-02,-2.23625000E-02,-2.29751000E-02,-2.18297000E-02,-2.30018000E-02,-2.28952000E-02,-2.19096000E-02
jarbo 12/10/2006 21:06:01 -3.05669000E-02,-1.66486000E-02,-2.24690000E-02,-2.17764000E-02,-2.09240000E-02,-2.29885000E-02,-2.40939000E-02
jarbo 12/10/2006 21:06:06 -3.14593000E-02,-2.24024000E-02,-2.19629000E-02,-2.18697000E-02,-2.24557000E-02,-2.07908000E-02,-2.27354000E-02
jarbo 13/10/2006 20:16:29 -1.82070000E-02,-2.78099000E-02,-2.66378000E-02,-2.54391000E-02,-2.90885000E-02,-2.79165000E-02,-3.01407000E-02
jarbo 13/10/2006 20:16:34 -3.81187000E-02,-2.51728000E-02,-2.35345000E-02,-2.32149000E-02,-2.44402000E-02,-2.41605000E-02,-2.28153000E-02
jarbo 13/10/2006 20:16:39 -3.79989000E-02,-2.35478000E-02,-2.30550000E-02,-2.25356000E-02,-2.42804000E-02,-2.31350000E-02,-2.12170000E-02
jarbo 13/10/2006 20:16:44 -3.88646000E-02,-2.57588000E-02,-2.34679000E-02,-2.24690000E-02,-2.31749000E-02,-2.40406000E-02,-2.34413000E-02
jarbo 13/10/2006 20:16:49 -3.72930000E-02,-2.46933000E-02,-2.37343000E-02,-2.32548000E-02,-2.31216000E-02,-2.32282000E-02,-2.20961000E-02
```

Figura VI.1.3 Contenido final del archivo de mediciones.

Como último punto se realizó la prueba de ejecución al programa que opera todo el proceso de adquisición de datos, obteniendo los siguientes resultados de acuerdo a la siguiente tabla.

Tabla VI.1.1 Resultados finales obtenidos por el comando *datademon* con sus diferentes parámetros.

Comando	Parámetro	Resultado
Datademon	ninguno	El sistema muestra una pantalla con los diferentes parámetros que son aceptados por éste, así como su respectiva sintaxis.
Datademon	erróneo	Lo mismo que en el punto anterior.
Datademon	config	Muestra el aviso que los parámetros de configuración han sido cargados con éxito.
Datademon	start	Muestra en la pantalla el registro de datos que se están leyendo por el puerto serie. Si ya se ha ejecutado el proceso de adquisición muestra un mensaje de error en la pantalla diciendo que la adquisición de datos se está ejecutando.
Datademon	stop	Si el proceso no se ha iniciado el programa muestra un mensaje en la pantalla diciendo que el proceso de adquisición no se ha iniciado. En caso contrario se detiene el proceso de adquisición.

VI.2 Pruebas a la etapa de Transmisión.

El primer objetivo en esta etapa, fue el de lograr que la PC transmitiera los datos recibidos por el proceso de adquisición hacia el servidor donde serán almacenados los datos.

Durante esta prueba los datos eran transmitidos, pero al ser recibidos éstos en el servidor, se encontraban algunos datos que no llegaban de forma completa, en la figura VI.2.1 se muestra la forma en que se almacenaban algunos datos.

```
jarbo 12/10/2006 21:05:46 -3.12595000E-02,-2.35478000E-02,-2.3361
jarbo 12/10/2006 21:05:51 -3.10597000E-02,-2.24024000E-02,-2.34546000E-02,-2.37343000E-02,-2.29219000E-02,-2.25356000E-02,-2.29352000E-02
jarbo 12/10/2006 21:05:56 -3.24848000E-02,-2.23625000E-02,-2.29751000E-02,-2.18297000E-02,-2.30018000E-02,-2.28952000E-02,-2.19096000E-02
jarbo 12/10/2006 21:06:01 -3.05669000E-02,-1.66486000E-02,-2.24690000E-02,-2.17764000E-02,-2.09240000E-02,-2.29885000E-02,-2.40939000E-02
jarbo 12/10/2006 21:06:06 -3.14593000E-02,-2.24024000E-02,-2.19629000E-02,-2.18697000E-02,-2.24557000E-02,-2.07908000E-02,-2.27354000E-02
jarbo 13/10/2006 20:16:29 -1.82070000E-02,-2.78099000E-02,-2.66378000E-02,-2.54391000E-02,-2.90885000E-02,-2.79165000E-02,-3.01407000E-02
jarbo 13/10/2006 20:16:34 -3.81187000E-02,-2.51728000E-02,-2.35345000E-02,-2.32149000E-02,-2.44402000E-02,-2.41605000E-02,-2.28153000E-02
jarbo 13/10/2006 20:16:39 -3.79989000E-02,-2.35478000E-02,-2.30550000E-02,-2.25356000E-02,-2.42804000E-02,-2.3135000
jarbo 13/10/2006 20:16:44 -3.88646000E-02,-2.57588000E-02,-2.34679000E-02,-2.24690000E-02,-2.31749000E-02,-2.40406000E-02,-2.34413000E-02
jarbo 13/10/2006 20:16:49 -3.72930000E-02,-2.46933000E-02,-2.37343000E-02,-2.32548000E-02,-2.31216000E-02,-2.32282000E-02,-2.20961000E-02
```

Figura VI.2.1 Contenido del archivo de mediciones recibido en el servidor, en el cual se muestra que los datos no llegan completamente.

El problema se encontró durante el proceso por el cual la transmisión revisa los datos disponibles que proporciona el proceso de adquisición. En otras palabras debía de haber sincronización en la petición de datos para ser transmitidos y la respuesta del proceso de adquisición para que este pueda liberar los datos.

Como solución a esta falla, se implementó del lado de la adquisición un mecanismo que permite bloquear el archivo mientras se estén almacenando los datos leídos por el puerto serie. En el momento que se ha completado la lectura de datos, entonces, por medio de una *bandera* se habilita el archivo para que puedan ser extraídos los datos. Del lado del proceso de transmisión se implementó la funcionalidad la cual permite conocer el estado de las banderas para poder extraer los datos del archivo que genera el proceso de adquisición. Cuando se requiere transmitir algún dato, antes de extraerlo del archivo de adquisición, primero consulta la bandera que indica que pueden ser extraídos los datos, posteriormente se asegura que el archivo esté disponible, técnicamente se detiene el proceso alrededor de un segundo y por último se realiza la extracción de los datos para ser transmitidos.

Al realizar estas incorporaciones, el arribo de los datos en el servidor es satisfactorio.

Como parte de los requerimientos en la etapa de transmisión descritos en el capítulo III, se deben cumplir ciertos comportamientos que permitan asegurar que los datos no se pierdan cuando la comunicación no exista. Por medio de la tabla VI.2.1 se muestra las diversas pruebas realizados al proceso de transmisión y los resultados obtenidos.

Tabla VI.2.1 Pruebas realizadas, comportamiento y resultados al proceso de transmisión.

Prueba realizada	Comportamiento del sistema	Resultado
Cable de red desconectado o conexión a Internet nula.	Se lanza una Excepción mostrando el error. <i>Se desconecto de la Red Connect to host ERROR:UnknownHostException?</i>	Los datos a transmitir son almacenados en la cola de espera.
Conexión con el servidor nula, el servidor no responde	Se lanza una Excepción mostrando el error. <i>NO Connect to host?ERROR: NoRouteToHostException</i>	Los datos a transmitir son almacenados en la cola de espera.
Nombre de usuario o contraseña invalida, el servidor interrumpe el acceso.	Se lanza una Excepción indicando el siguiente error en pantalla: <i>Falló la conexión en el Cliente Sftp, nombre de usuario inválido o contraseña.</i>	Los datos a transmitir son almacenados en la cola de espera.
Comunicación y acceso satisfactorio, pero la cola de datos de datos excede al tamaño permitido para ser transmitido.	Almacena en un archivo log el acontecimiento sucedido	Mueve los datos de la cola hacia un archivo, destinado a guardar estos datos en la PC, hasta que se extraigan de forma manual. Vacía la cola.
Comunicación y acceso satisfactorio, pero la cola de datos de datos no está vacía.	Realiza la transmisión.	Se trasmite la cola de datos y los datos recientemente adquiridos.

VI.3 Pruebas en la etapa de Recepción.

En esta etapa el objetivo principal es el de poder recibir los datos del sitio remoto de tal manera que al recibirlos, el programa de recepción permita almacenarlos en un directorio de acuerdo al nombre de la estación y discriminando los datos por medio de dos archivos, el archivo *active* el cual como ya se ha explicado en el Capítulo III, almacenará las mediciones del día actual y por otro lado se tiene el archivo cuyo nombre del archivo es: *sigla_de_la_estacion.yyyymmdd.día_del_año* (yyyy=año en formato de 4 dígitos, mm=mes del año empezando por 01, dd=día del mes empezando por 01), para fines de explicación se nombrará a este archivo como *archivo histórico*.

En la primera fase de ejecución del programa se tuvo problemas en el monitoreo de llegada de datos, pues el proceso de transmisión entrega los datos utilizando un archivo que es enviado por sftp; al percatarse el programa de recepción de datos que el archivo existía, comenzaba el proceso de almacenamiento, al realizar esto, tomaba los datos que encontraba en el archivo de arriba y eliminaba dicho archivo, esto provocaba la pérdida de datos.

Para corregir este problema se implantó en el proceso de transmisión un procedimiento que permite renombrar el archivo de datos cuando este ha sido transmitido satisfactoriamente, de esta manera el programa de recepción tomará los datos del archivo que ya ha sido completado evitando de esta manera la pérdida de datos.

Al realizar otra prueba de ejecución del programa, al generarse el cambio de día, el programa no podía realizar el cambio del archivo *active* por el correspondiente al día anterior y generar nuevamente el archivo *active* para el día actual, por lo cual se tuvo que realizar cambios en la programación, de tal manera que al haber un cambio de día, antes de que iniciará el cambio de los archivos, primero identifica la fecha del último dato del archivo *active* (se tiene la certeza de utilizar este dato ya que los datos son almacenados cronológicamente según su fecha de medición), si esta fecha ya no correspondía a la fecha actual, entonces era indicador de que ese archivo *active* debe de ser remplazado por el *archivo histórico*.

Al probar nuevamente la ejecución del programa este cumple con el requisito de almacenar los datos en forma adecuada como lo expresa los requerimientos para la etapa de recepción.

Conclusiones.

Durante la elaboración de este trabajo, se pudo demostrar que la implantación de sistemas de información propietario y de alto costo, aunque éstas sean soluciones que permitan resolver algunas necesidades de un cliente, no son siempre la mejor solución, muchas veces estas soluciones están sobradas o carecen de alguna función en particular que necesita el cliente y si a esto se le agrega que el capital del cliente es bastante limitado como para poder adquirir esta tecnología, la implantación de un sistema de información a su problemática será difícil de llevar a cabo, sin embargo, al realizar un buen estudio de las herramientas que hay hoy en día y que no tienen costo alguno más que el tiempo en instalar y desarrollar las aplicaciones, se logra tener una excelente solución a los requerimientos del cliente. Debido a esto, el sistema SIORS fue desarrollado totalmente bajo la plataforma Linux y el uso de herramientas de programación de uso libre, así como también el uso de Internet como medio de comunicación pues en estos tiempos es relativamente sencillo contar con este servicio ya sea por línea telefónica, celular e inclusive satelital, aunque es importante señalar que la elección de algún medio de comunicación entre un sitio remoto y el Observatorio de Radiación Solar estará sujeto al presupuesto de la organización, en este trabajo de tesis no se sujeta a algún tipo de medio de comunicación en especial, pero si deja claro el uso de una red con protocolo TCP/IP el cual es utilizado para la comunicación entre computadoras por Internet.

También uno de los objetivos principales a cumplir en el desarrollo del sistema SIORS, fue el de brindarle al grupo de trabajo de Radiación Solar una herramienta en software que les permitiera reducir en tiempo y costo la adquisición de datos del régimen radiacional, de lo cual se logró finiquitar en buen término, además, la recopilación de los datos en tiempo real que proporciona el sistema de adquisición le permite al Observatorio de Radiación Solar obtener la información de forma organizada y centralizada cubriendo la disponibilidad de las mediciones obtenidas por las estaciones en las cuales se instale este sistema.

Debido al tipo de arquitectura que fue elegido para desarrollar el sistema (cliente/servidor), proporciona la capacidad de poder instalar un conjunto de estaciones a lo largo de la República Mexicana, las cuales estarán enviando la información de las mediciones en tiempo real, hacia el servidor central ubicado en el observatorio, con esto

se llega a conformar una red de estaciones solarimétricas para el país, logrando cumplir con otro de los objetivos que se habían planteado en la realización de este proyecto.

Un poco más a detalle, en el módulo del sistema concerniente a la comunicación con el instrumento de adquisición de datos (HP 34970), se logró crear una interfaz, la cual le permite al operador configurar y lanzar el proceso de adquisición de forma sencilla, realmente solo se requiere ingresar en un archivo de texto algunos parámetros de configuración para poder operar el HP 34970, así como también parámetros de configuración para la comunicación RS232 (puerto serie) y parámetros que definen el comportamiento de almacenamiento de datos y envío de éstos hacia el servidor central. Para futuras versiones de los aparatos de adquisición HP el sistema seguirá funcionando ya que el sistema configura al instrumento de adquisición utilizando el conjunto de instrucciones SCPI el cual es un estándar para dichos instrumentos, sin embargo no se asegura el funcionamiento del sistema para otro tipo de instrumentos, ya que éstos pueden incluir un conjunto de instrucciones que sea específico de la marca y que difiera del estándar SCPI.

También es importante mencionar que el mecanismo para la operación del sistema de adquisición de datos en los sitios remotos y en el servidor de almacenamiento, es mediante el uso de comandos en una terminal de Unix, posiblemente como una nueva versión de este trabajo se pueda incluir una interfaz gráfica que permita al usuario de forma más amigable operar las funcionalidades del sistema; la interfaz gráfica no fue uno de los objetivos planteados en este desarrollo, por lo cual no se incluyó en este tema de tesis, pero aun así se establecieron los medios adecuados que permiten el fácil uso del sistema, sin tener que recurrir a un conocimiento avanzado por parte del usuario que va a operar dicho software, lo único que necesita saber el usuario es conocer los parámetros de configuración y los comandos que se utilizan.

Por el lado del servidor de almacenamiento de datos, ya sea el administrador del sistema o bien otro usuario que cuente con los privilegios suficientes para operar el sistema, cuenta con la posibilidad de lanzar el proceso de recepción de datos de alguna estación en particular, los datos al ser recibidos y almacenados de forma organizada son fuente de información para otros sistemas que requieren utilizarla para generar algún tipo de análisis, reporte u otro tipo de proceso, éste fue otro de los objetivos planteados en este trabajo que se concluyó satisfactoriamente.

El uso de herramientas UML para el análisis del sistema, ayudó en gran medida a conocer en forma detallada los requerimientos que necesitaba el cliente, la manera en que debería operar y los resultados que se deberían obtener, gracias a esto se logró cumplir las expectativas que se tenían planteadas. El uso de UML facilitó el proceso de desarrollo del sistema además de proporcionar información para posibles desarrolladores que requieran mejorar y mantener esta aplicación.

Este sistema tiene la capacidad de ser adaptado y modificado de acuerdo a alguna necesidad en especial, esto gracias a la modularidad con la que se desarrollo, pues puede servir ya sea como para el uso exclusivo de configuración y adquisición de los instrumentos HP 34970 o bien para utilizarlo como un sistema de transmisión de datos de manera automatizada, las posibilidades son varias y lo mejor de todo es que no se requiere contar con gran capital monetario para operar y mantener esta aplicación.

Apéndice A.

A.1 UML.

UML (Unified Modeling Language) por sus siglas en inglés, es un lenguaje de modelación gráfica el cual nos permite analizar construir y documentar a los elementos que intervienen en el desarrollo de software orientado a objetos. Este lenguaje surgió de la necesidad de unificar e incorporar las principales ventajas de las diferentes herramientas de modelación para el desarrollo de software orientado a objetos como lo era Booch, OMT, OOSE y otras técnicas, esta unificación también implicaba ayudar a los ingenieros en software orientado a objetos a tener una herramienta visual compartida.

Es importante tener claro que UML puede ser utilizada en cualquier proceso de desarrollo que se elija y puede aplicarse en diferentes etapas, sin importar el ámbito del sistema, pues esta herramienta ofrece un conjunto de diagramas determinados por una semántica y notaciones que nos van a poder permitir generar diferentes vistas de los aspectos dinámicos y estáticos del sistema, entendiendo como aspectos dinámicos el comportamiento del sistema a lo largo del tiempo y aspectos estáticos como la descripción de los elementos del sistema y sus relaciones. A continuación se describen brevemente algunos diagramas utilizados en UML.

A.2 Diagrama de casos de uso.

Estos diagramas nos ayudan a identificar quienes interactúan en el sistema y cuáles son sus funciones o actividades dentro del mismo, logrando con esto establecer las relaciones en ambos casos. En la figura A.2.1 se muestra un ejemplo de caso de uso. Los elementos que constituyen un diagrama de casos de uso son los *actores* los cuales representan a una entidad externa que va interactuar con el sistema y los casos de uso que está representado gráficamente mediante una elipse cuyo nombre es la descripción de una tarea específica a realizar ya sea por parte de un actor o puede ser de otro caso de uso. También en un diagrama de este tipo puede haber las siguientes relaciones entre sus elementos:

- Extiende. Se utiliza cuando un caso de uso especializa a otro extendiendo su funcionalidad.
- Usa. Se utiliza cuando un caso de uso ocupa la funcionalidad de otro.

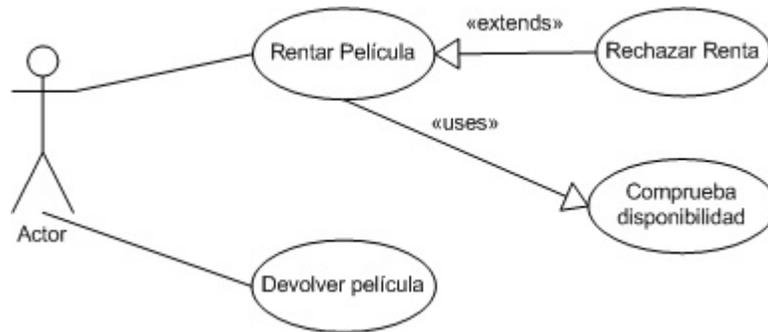


Figura A.2.1 Diagrama de un caso de uso.

A.3 Diagrama de clases.

Los diagramas de clase son aquellos que nos van a permitir modelar la estructura, contenido y relación que hay entre las clases del sistema, en otras palabras son los utilizados para documentar la estructura estática del sistema. Una clase va a estar constituida de tres características:

- Nombre
- Atributo o propiedad
- Función

El diagrama de clases va a estar constituido por clases y asociaciones entre estas, en donde una asociación define la relación entre las clases indicando la multiplicidad que tiene asociada.

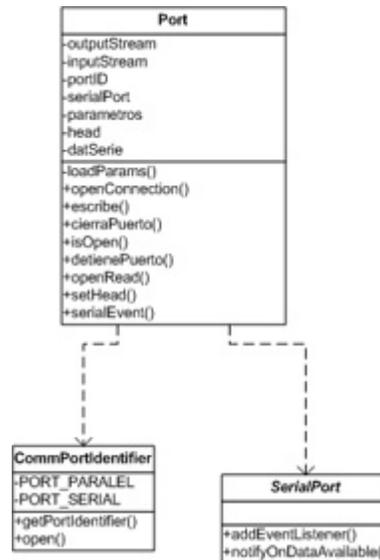


Figura A.3.1 Diagrama de clases.

A.4 Diagrama de secuencias.

Estos diagramas nos van a permitir identificar la interacción de forma ordenada entre los actores y objetos durante la ejecución de alguna tarea. Las interacciones entre los objetos están representadas mediante los mensajes que son intercambiados entre ellos de forma ordenada durante un lapso tiempo. El diagrama de secuencia está conformado de la siguiente manera, la línea vertical representa el tiempo también conocida como la línea de vida y en el eje horizontal se van colocando los objetos que interactúan entre sí, no es necesario tener un orden en la aparición de los objetos aunque se recomienda colocar los objetos que participan antes lo más a la izquierda posible, esto con el fin de que el diagrama sea lo más legible posible. Al rectángulo que se muestra sobre la línea de vida se le conoce como el tiempo de activación del objeto en donde el tiempo fluye de arriba hacia abajo. Los mensajes son representados mediante flechas que van desde la línea de vida del emisor hacia la línea de vida del receptor.

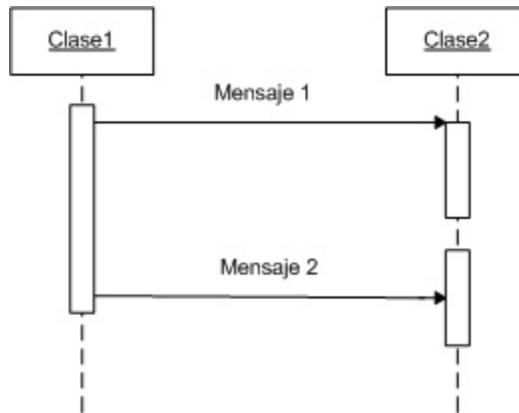


Figura A.4.1 Diagrama de secuencias.

A.5 Diagrama de actividades.

En los diagramas de actividad se va documentando como se coordinan las actividades para llegar a un fin común, de tal manera que se comienza a modelar el flujo de control o de trabajo entre las actividades. Los diagramas de actividad van almacenando las dependencias entre las actividades mostrando con esto que tareas se pueden realizar en paralelo y/o que se necesita culminar para comenzar otra actividad. Los elementos que intervienen en un diagrama de actividad se listan a continuación:

- **Actividad.** Está representada mediante un rectángulo con esquinas redondeadas cuyo nombre representa una actividad o una acción.
- **Transición.** Se representa mediante una flecha, la cual nos indica la finalización de una actividad previa y el comienzo de otra, ver figura A.5.4
- **Barra de sincronización.** Es una línea gruesa horizontal, la cual nos indica la coordinación entre las actividades proporcionando una manera de expresar que tareas hay que esperar a que terminen antes de continuar otra (unión) o bien comenzar con la ejecución de varias tareas en paralelo (división).

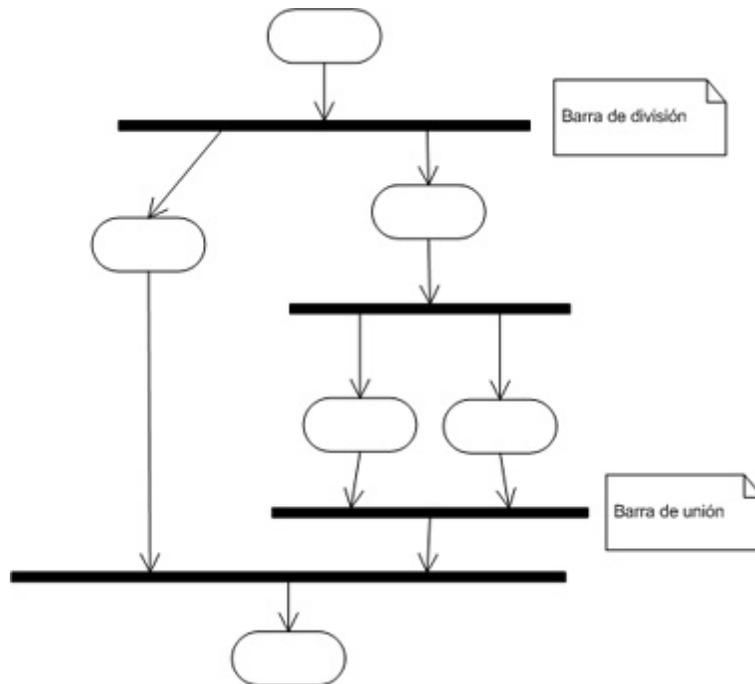


Figura A.5.1 Barra de sincronización.

- **Diamante de decisión.** Representa la forma de elegir varios caminos alternos al término de una actividad, también llamada como bifurcación, esta tendrá una transición de entrada y dos o más de salida. A la salida de cada transición será evaluada una expresión booleana que será evaluada una vez que haya llegado a la bifurcación.



Figura A.5.2 Bifurcación.

- **Calles.** Se utiliza cuando se modelan flujos de trabajo de organizaciones, a las actividades que se encuentran dentro de una calle nos indica que organización es la responsable de la realización de estas.

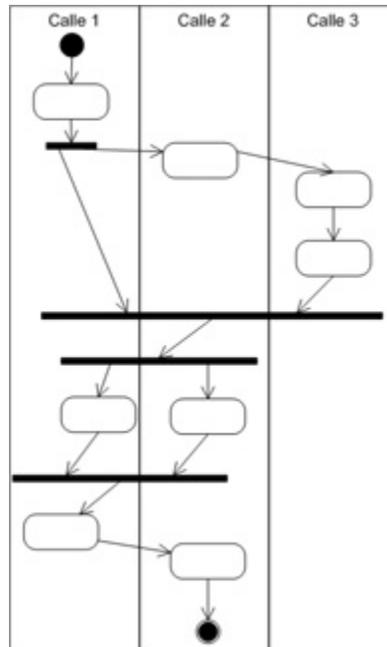


Figura A.5.3 Representación de un diagrama de actividades con calles.

- **Marcas de creación y destrucción.** Son representados mediante un círculo de color negro indicando el inicio y fin del diagrama, en este último se dibuja una circunferencia alrededor del círculo de color negro.

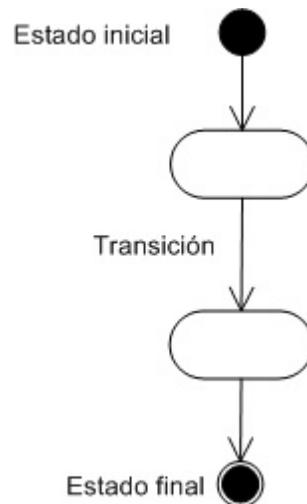


Figura A.5.4 Inicio, fin y transición de un diagrama de actividad.

Glosario.

API:	Es una interfaz de aplicaciones (Application Programming Interface por sus siglas en inglés), la cual contiene un conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizada por otro software.
Back End:	Es el conjunto de procesos que se encargan del almacenamiento de los datos de las aplicaciones y de ejecutar las tareas necesarias para responder a las solicitudes de los clientes.
Bussiness Rules:	La lógica del negocio en donde se realiza el procesamiento de la información.
Byte Code:	Cuando se compila un programa escrito en Java, el compilador genera un código de bytes que se almacenan en un archivo, este archivo byte-code tiene que ser interpretado por la maquina virtual de Java para que sean ejecutadas las instrucciones contenidas en el programa.
Cliente:	Es la persona la cual va hacer uso del sistema dependiendo del rol que a éste se le asigne.
Convertidor A/C:	Dispositivo electrónico el cual convierte un voltaje determinado a un valor binario, en otras palabras se encarga de transformar señales analógicas a digitales.
Digi:	Vea Port Server.
Dirección IP:	Es un número único, que identifica de manera lógica y jerárquica a un dispositivo dentro de una red que utilice el protocolo IP, por ejemplo la dirección 192.168.1.23 esta asignada a un servidor de impresión para que pueda ser identificado en la red.
EEPROM:	Memoria de lectura y escritura no volátil, en donde la información puede ser borrada o modificada eléctricamente.
Ethernet:	Es el nombre de la tecnología en redes de computadoras basada en el envío de paquetes de datos.
Front End:	Conjunto de procesos enfocados a la presentación del usuario, con las entradas de datos y visualización de resultados.
GNU:	Licencia pública general de software desarrollado para distribución

sin fines de lucro.

Lenguaje ensamblador:	Lenguaje de programación que utiliza códigos nemotécnicos para indicarle al hardware las operaciones que tiene que realizar.
Open Source:	Significa que puede obtenerse el código fuente de un programa para copiarlo, ejecutarlo, distribuirlo, usarlo ó modificarlo para mejorarlo. El open source es un asunto de libertad, no de precio.
Port Server:	Dispositivo electrónico el cual permite obtener comunicación serial ya sea con uno o más puertos a través de un enlace Ethernet. Físicamente el equipo cuenta de un lado con puertos RS-232 y del otro tiene una interfaz Ethernet. En la computadora se instala un driver el cual asigna al equipo uno o más puertos seriales (COM4, COM5, etc.) y el medio de acceso a estos puertos es por medio de la dirección IP del PortServer. La marca de estos dispositivos es DIGI, existen otros en el mercado.
RAM:	Memoria de acceso aleatorio de lectura y escritura volátil.
Red solarimétrica:	Conjunto de estaciones interconectadas entre sí, las cuales cuantifican la radiación solar, utilizando como medio un canal de comunicación para el flujo de información.
ROM:	Memoria de solo lectura no volátil. Después de que se ha sido almacenada información en ella, dicha información no se puede alterar.
SFTP:	Protocolo para la transferencia de archivos y la funcionalidad de manipulación de datos fiables sobre cualquier secuencia ya que los datos circulan cifrados por la red.
SSH:	Nombre del protocolo y del programa que lo implementa. Este protocolo sirve para lograr el acceso remoto entre computadoras a través de una red de manera segura, gracias a las técnicas de cifrado de la información que utiliza, con el objetivo de impedir a un atacante descubrir el nombre del usuario, contraseña o la información que fluye durante toda la sesión de acceso.
TCP/IP:	Fundamental conjunto de protocolos utilizado en Internet, que sirve para enlazar computadoras o dispositivos de red. Garantiza que los datos serán entregados a su destino sin errores y en el mismo orden en que se transmiten ya sea desde equipos dentro de una área local (LAN) como de una red de área extensa (WAN).

Referencias.

Ceballos Fco. Javier, "Java 2 Curso de Programación", AlfaOmega, México 2003.

Coughlin, Driscoll, Villanuci, "Data Acquisition and Process Control with the M68HC11 Microncontroller", Merrill, U.S.A. 1994.

Dvorak – Anis, "Telecomunicaciones para PC", Borland – Osborne/McGraw-Hill, México 1992.

Ferré Grau X., Sánchez Segura Maria I., "Desarrollo orientado a objetos", Facultad de Informática, UPM.

Garcimartín Montero A., "Sistema de medida y adquisición de datos", Universidad de Navarra, España.

Hewlett-Packard Company, "HP 34970A Data Acquisition/Switch Unit", USA, 1997.

Kondratyev, L. Ya. "Radiation in the atmosphere", Academic Press, New York, 1983.

Leyva A., Muhlia A., "Radiación Solar", en Memorias del curso de actualización en energía solar, 26 de marzo- 6 de abril de 1984, UNAM, Unidad Académica de Ciclos Profesionales y de Posgrado, C.C.H., pp 3-43, 1984.

Muhammad Iqbal, "An introduction to solar radiation", Academia Press, New York, 1983.

Perdita Stevens, "Utilización de UML en ingeniería de software con objetos y componentes", Addison Wesley, España, 2000.

Pressman Roger S., "Ingeniería de Software un enfoque práctico", McGrawHill, México, 1992.

Sánchez Prieto S, García Población O., "UNIX y Linux", Ra-Ma, España, 2005.

Zang Tony & Till David, "Perl 5 in 21 days", SAMS, Indiana, 2000.

Sitios web consultados:

Apache Ant, <http://ant.apache.org/>, Abril 2007.

Java Communications API,
<http://java.sun.com/products/javacomm/reference/api/javax/comm/package-summary.html>,
Enero 2007.

Java sstools API, <http://sourceforge.net/projects/sstools>, Mayo 2007.