

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica



Sistema de control automático para una antena satelital

**Informe de Proyecto de Graduación para optar por el título de Ingeniero en
Electrónica con el grado académico de Licenciatura**

Dennis Xavier Arroba Tijerino

Cartago, Agosto de 2011

INSTITUTO TECNOLOGICO DE COSTA RICA
ESCUELA DE INGENIERIA ELECTRONICA
PROYECTO DE GRADUACION
TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



Ing. Eduardo Interiano Salguero

Profesor lector



Ing. Faustino Montes de Oca Murillo

Profesor lector



Ing. Julio Stradi Granados

Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, Agosto 2011

DECLARATORIA DE AUTENTICIDAD

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, Agosto 2011



Dennis Xavier Arroba Tijerino
Cédula de residencia: 172400090616

RESUMEN

La red nacional de VSAT-ICE del Instituto Costarricense de Electricidad se encarga de proporcionar servicios de banda ancha y teléfono para zonas remotas a lo largo y ancho del país. El punto central de esta red se encuentra ubicado en la Estación Terrena de Tarbaca, en el cantón de Aserrí, provincia de San José, Costa Rica, aproximadamente a veinte kilómetros de distancia del centro de la capital.

El sistema que controla la Antena satelital que recibe la información procedente de un satélite en órbita geoestacionaria y que posteriormente reparte a la red VSAT-ICE, data de los años 80 y posee tecnología que al día de hoy se considera obsoleta. Producto de esta antigüedad, el sistema presenta una falla que obliga que, si se presenta algún problema en cuanto al posicionamiento de la antena o una interrupción en la alimentación eléctrica, se deba apersonar algún funcionario de soporte del ICE para reajustar y reposicionar la antena a la hora y fecha que se requiera.

Es por esta razón que aparece la necesidad de diseñar un prototipo para un nuevo control de la antena que pueda eventualmente llegar a ser implementado, no solamente para reemplazar el tipo de tecnología sobre el cual está construido el sistema, sino también que se pueda acceder y controlar de manera remota vía internet para evitar el traslado de personal hasta Tarbaca, y así, poder reducir costos en cuanto a transporte y horas extras.

Descripción sintética:

Reingeniería del sistema de control de la antena satelital, así como la implementación de la comunicación remota vía internet.

Palabras clave:

Antena satelital, control automático, internet, conexión remota, ICE, JAVA, Tarbaca.

ABSTRACT

The VSAT-ICE network, part of the *Instituto Costarricense de Electricidad*, is responsible for providing broad band services, including telephone service in remote areas across the country. The central core of this network is located at the Earth Station (Ground Station) in Tarbaca, Aserrí, in the province of San José, Costa Rica, about twenty miles away of the capital city.

The system that controls the Satellite Antenna that receives the information signals from a satellite in the spatial orbit, and subsequently distributed to the VSAT-ICE network, is from the 80's and has a kind of technology that today can be considered obsolete. Because of this, the system presents a failure, every time the power goes down or the control system place the antenna wrongly, an ICE operator must go to the station in person to readjust and reposition the antenna no matter the date or time of the day.

This is the main reason that it is necessary to design a prototype for a new antenna control system that someday may be implemented, not just to replace the technology of the core, but to be accessed and controlled remotely via internet to avoid the personnel going to the station, and in this way, reduce costs in transportation and overtime expenses.

Brief description:

Control system reengineering of the satellite antenna and implementation of a new remote access and control for the system via internet.

Keywords:

Satellite antenna, automatic control system, internet, remote connection, ICE, JAVA, Tarbaca, Earth Station, Ground Station, database.



A Dios Padre, a la Virgen María y a San Juan
Bosco...

Le dedico este trabajo, esta carrera a mis padres,

Jesús Arroba y Carmen Tijerino....

Y a mis hermanos

Alberto, Marisol y Roberto

A todos ellos gracias infinitas....





Gracias Dios y a la Virgen por permitirme llegar a este punto, donde se cierra una etapa pero da inicio a una nueva...

Agradezco a mis padres Jesús y Carmen, por su apoyo incondicional a lo largo de mi vida, por enseñarme siempre a no rendirme y buscar la excelencia y a vivir la vida de forma correcta moral y espiritualmente, sin ellos dos jamás lo hubiese logrado...

Agradezco a mis tres hermanos Alberto, Marisol y Roberto, porque de muchas formas me ayudaron a salir adelante en este largo caminar, apoyándome, aconsejándome, acompañándome y hasta alejándome de la tensión diaria de la vida universitaria...

Agradezco a mi bella novia Mary Schäffer, que apareció como una luz cuando más oscuro se tornó este viaje, que siempre supo darme aliento cuando todo estaba perdido y su amor que me ha llevado a más, porque hizo mucho más de lo que ella se acredita...

Agradezco a mi bella Gitanita, que me acompañó durante las tardes y noches de estudio y desvelo, y que siempre me alegraba y me distraía para tomar segundos aires...

Agradezco a los ingenieros Ricardo Neira y Julio Stradi por brindarme la oportunidad de realizar mi proyecto y así concluir mi carrera en el Instituto Costarricense de Electricidad, dándome una oportunidad única...

Agradezco al personal de DTOM en el ICE, a Elena Vargas, Magaly Solano, Rodolfo S. Toruño, Guille Matarrita, Ronny Gómez, Douglas Quesada y Mario Flores que me ayudaron, me apoyaron y me hicieron sentir como en casa...

Finalmente, agradezco a todos los amigos, compañeros y conocidos que a lo largo de la carrera tuve la oportunidad de conocer, tratar y trabajar, a aquellos que desde el primer día estuvieron presentes y los que en el camino encontré. Mención especial a un gran amigo, Manuel Rojas por su ayuda y apoyo en la recta final...

A todos, GRACIAS...



INDICE GENERAL

Capítulo 1 : Introducción.....	1
1.1. Problema existente e importancia de la solución.	1
1.2. Solución seleccionada.....	4
Capítulo 2 : Meta y Objetivos.	6
2.1. Meta.	6
2.2. Objetivo general.	6
2.3. Objetivos específicos.	7
2.3.1. Objetivos de diseño.	7
2.3.2. Objetivos de software.	7
2.3.3. Objetivos de documentación.	7
Capítulo 3 : Marco teórico.	8
3.1. Descripción del sistema o proceso a mejorar.....	8
3.2. Antecedentes bibliográficos.	8
3.3. Descripción de los principales principios físicos y/o electrónicos relacionados con la solución del problema.	9
3.3.1. Antena Satelital.....	9
3.3.2. Estación Terrena.	11

3.3.3.	Protocolo UDP.....	12
3.3.4.	Tarjeta de desarrollo SBC65EC.....	14
3.3.5.	Motor de inducción.	15
3.3.6.	Sistema de control de lazo cerrado.	17
3.3.7.	Definiciones varias de control automático.	18
3.3.8.	Compensador de adelanto.	20
3.3.9.	Lugar de las raíces.	20
3.3.10.	Control integral.	21
3.3.11.	Motor Paso a Paso.....	22
3.3.12.	Transistor Darlington.	24
Capítulo 4 : Procedimiento metodológico.....		25
4.1.	Reconocimiento y definición del problema.....	25
4.2.	Obtención y análisis de la información.....	26
4.3.	Evaluación de las alternativas y síntesis de la solución.	26
4.4.	Implementación de la solución.	27
4.4.1.	Rediseño del control.....	27
4.4.2.	Montaje del prototipo a pequeña escala.....	28
4.4.2.1.	Sitio web y base de datos.....	28
4.4.2.2.	Kit de desarrollo SBC65EC.....	28

4.4.2.3.	Aplicación JAVA® para recepción de datos UDP.....	29
4.4.3.	Motores de posicionamiento.....	29
4.5.	Reevaluación y rediseño.....	29
Capítulo 5 : Descripción detallada de la solución.....		31
5.1.	Análisis de soluciones y selección final.....	33
5.1.1.	Kit de desarrollo.....	33
5.1.1.1.	Arduino Ethernet Shield.....	33
5.1.1.2.	SBC65EC.....	34
5.1.1.3.	PICDEM.net 2.	34
5.1.2.	Manejo de la etapa de motores.	35
5.1.2.1.	Puente H.	35
5.1.2.2.	ULN2003.	36
5.1.3.	Envío de paquetes de datos.	36
5.1.3.1.	TCP:.....	37
5.1.3.2.	UDP:.....	37
5.2.	Descripción del hardware.....	38
5.2.1.	Manejo de motores.....	38
5.2.2.	Ayudas audiovisuales.....	39
5.3.	Descripción del software.....	40

5.2.1.	Envío de datos UDP.....	41
5.3.1.	Recepción de datos UDP.....	43
5.3.2.	Control de la ubicación de los motores.....	46
5.3.2.1.	Identificación del paso y posición actual.....	46
5.3.2.2.	Cálculo de pasos.....	48
5.3.2.3.	Lógica de bits para posicionamiento.....	52
5.3.3.	Sitio web.....	55
5.3.4.	Base de datos.....	56
Capítulo 6 :	Análisis de Resultados.....	59
6.1.	Control automático.....	59
6.2.	Modelo a pequeña escala del sistema.....	68
6.2.1.	Página web.....	68
6.2.1.1.	Inicio de sesión.....	68
6.2.1.2.	Agregar/Eliminar Usuario.....	70
6.2.1.3.	Registro histórico.....	72
6.2.1.4.	Búsqueda en el registro.....	73
6.2.1.5.	Posicionamiento.....	74
6.2.1.5.1.	Ángulos de posición.....	75
6.2.1.5.2.	Anclaje/Parada.....	76

6.2.2. Envío y recepción de datos UDP.....	77
6.2.3. Base de datos.....	78
6.2.4. Control sobre los motores.....	80
6.3. Funcionamiento del sistema actual de control.	80
6.3.1. Funcionamiento.	80
6.3.2. Opciones del sistema.	82
6.3.3. Unidad de Control de la Antena (ACU).....	83
6.3.4. Sistema de rastreo de señal RF.	84
6.3.5. Enlace de datos.....	84
Capítulo 7 : Conclusiones y recomendaciones.....	85
7.1. Conclusiones.....	85
7.2. Recomendaciones.....	86
Bibliografía.	87
Apéndices.....	90
A1. Abreviaturas y glosario.....	90
A2. Manual de usuario.....	92
Anexos.	97
Anexo 1. Hoja de datos del sistema de control.....	97
Anexo 2. Hoja de datos del kit de desarrollo.....	99

Anexo 3. Hoja de datos del ULN2003.....	100
Anexo 4. Fotografía del prototipo.....	102

INDICE DE FIGURAS

Figura 1.1	Representación red VSAT con <i>hub</i> central.	2
Figura 1.2	Diagrama de bloques de la solución propuesta.	4
Figura 1.3	Secuencia de la solución del problema presente.	5
Figura 3.1	Tipos de antena parabólica.	10
Figura 3.2	Secciones presentes en una Estación Terrena.	12
Figura 3.3	Encapsulado UDP.	13
Figura 3.4	Estructura de un paquete UDP.	13
Figura 3.5	Kit de desarrollo SBC65EC.	15
Figura 3.6	Motor de inducción de corriente alterna.	17
Figura 3.7	Diagrama de un sistema de lazo cerrado con realimentación unitaria.	18
Figura 3.8	Especificaciones en el dominio del tiempo para un sistema ante entrada de escalón unitario.	19
Figura 3.9	Lugar de las raíces en tiempo discreto.	21
Figura 3.10	Motor paso a paso.	23
Figura 3.11	Diagrama de un transistor Darlington NPN.	24
Figura 5.1	Diagrama general del proyecto.	31
Figura 5.2	Diagrama del funcionamiento secuencial del sistema.	32

Figura 5.3	Diagrama de conexión de los motores paso a paso.....	38
Figura 5.4	Diagrama general del software.....	41
Figura 5.5	Diagrama de flujo para el envío de paquetes UDP.	42
Figura 5.6	Extracto del código desarrollado para el envío de paquetes UDP.	43
Figura 5.7	Extracto del código realizado para la recepción de datos UDP.....	44
Figura 5.8	Diagrama de flujo para la recepción de datos UDP.....	45
Figura 5.9	Extractos del código realizado para identificar el número de pasos requeridos.	48
Figura 5.10	Diagrama de flujo para el proceso de cálculo de pasos para azimut.	50
Figura 5.11	Diagrama de flujo para el proceso de cálculo de pasos para elevación...	51
Figura 5.12	Extractos del código realizado para el cálculo del número de pasos requeridos.	52
Figura 5.13	Diagrama de flujo para posicionar el motor de elevación.....	53
Figura 5.14	Diagrama de flujo para posicionar el motor azimut.	54
Figura 5.15	Diagrama general del sitio web.	56
Figura 5.16	Tablas creadas para la base de datos.	57
Figura 5.17	Base de datos creada en ORACLE®.	57
Figura 5.18	Tablas empleadas.....	58
Figura 5.19	Tablas empleadas extendidas.....	58

Figura 6.1	Diagrama esquemático de la planta.	60
Figura 6.2	Diagrama de bloques del sistema.	61
Figura 6.3	Gráfica de la respuesta al escalón para el motor de elevación.	65
Figura 6.4	Gráfica de la respuesta al escalón para el motor azimut.....	67
Figura 6.5	Página de inicio del sitio web del proyecto.....	69
Figura 6.6	Página de inicio de sesión una vez iniciada exitosamente.....	70
Figura 6.7	Página para agregar usuarios del sistema.	71
Figura 6.8	Página para eliminar usuarios del sistema.....	72
Figura 6.9	Página de búsqueda en el registro.....	73
Figura 6.10	Página de búsqueda mostrando resultados.	74
Figura 6.11	Página web mientras se posicionan los motores.....	75
Figura 6.12	Página web para posicionar la antena.	76
Figura 6.13	Aplicación oyendo al puerto.	77
Figura 6.14	Aplicación atendiendo los datos del puerto.	78
Figura 6.15	Conexión a la base de datos.....	79
Figura 6.16	Pantalla de inicio del sistema actual de control.....	81
Figura A.1	Página web con mensaje de error de usuario.....	92
Figura A.2	Página de visualización del registro histórico.	93
Figura A.3	Página web en funcionamiento normal.	95

Figura A.4	Página web para modo STOP.	95
Figura A.5	Página web para modo STOW.	96
Figura B.1	Fotografía del prototipo montado.	102

INDICE DE TABLAS

Tabla 1	Tabla de equivalencia para motor azimutal.	47
Tabla 2	Tabla de equivalencias para motor de elevación.	47
Tabla 3	Características físicas del sistema.	62
Tabla 4	Valores obtenidos para ambos motores.	63
Tabla 5	Nuevos parámetros para el diseño de los compensadores.	64

Capítulo 1 : Introducción.

En este capítulo se describe a grandes rasgos el problema planteado, así como el procedimiento escogido para darle solución al problema. Al mismo tiempo se explican las principales razones de la importancia de encontrar e implementar la solución.

1.1. Problema existente e importancia de la solución.

Para el Instituto Costarricense de Electricidad (ICE), el compromiso y la excelencia son parte de sus valores y pilares sobre los que fue fundada esta entidad a mediados del siglo pasado, y como tal, ve en sus obligaciones no solo prestar un servicio de calidad, sino también cuidar que ante un fallo, la respuesta sea lo más pronta posible. Como parte de su portafolio de servicios, el ICE cuenta con la red VSAT-ICE, parte del Sistema Nacional de Telecomunicaciones (SNT), que brinda servicio de banda ancha y telefonía mediante antenas del tipo VSAT a comunidades y regiones del país remotas que, ante imposibilidad de brindar servicios tradicionales mediante fibra óptica, líneas de cobre y/o enlaces de microondas, brinda esos servicios de forma satelital.

Esta red de antenas VSAT tienen como punto principal de comunicación la Estación Terrena ubicada en Tarbaca, Aserrí, desde donde la información es recibida de los satélites en el espacio exterior y después es propagada hacia las VSAT, para ser distribuida a través de la red VSAT-ICE.

En la figura 1 se puede observar una representación de una red VSAT con su respectivo *hub* (Antena satelital, Estación Terrena).

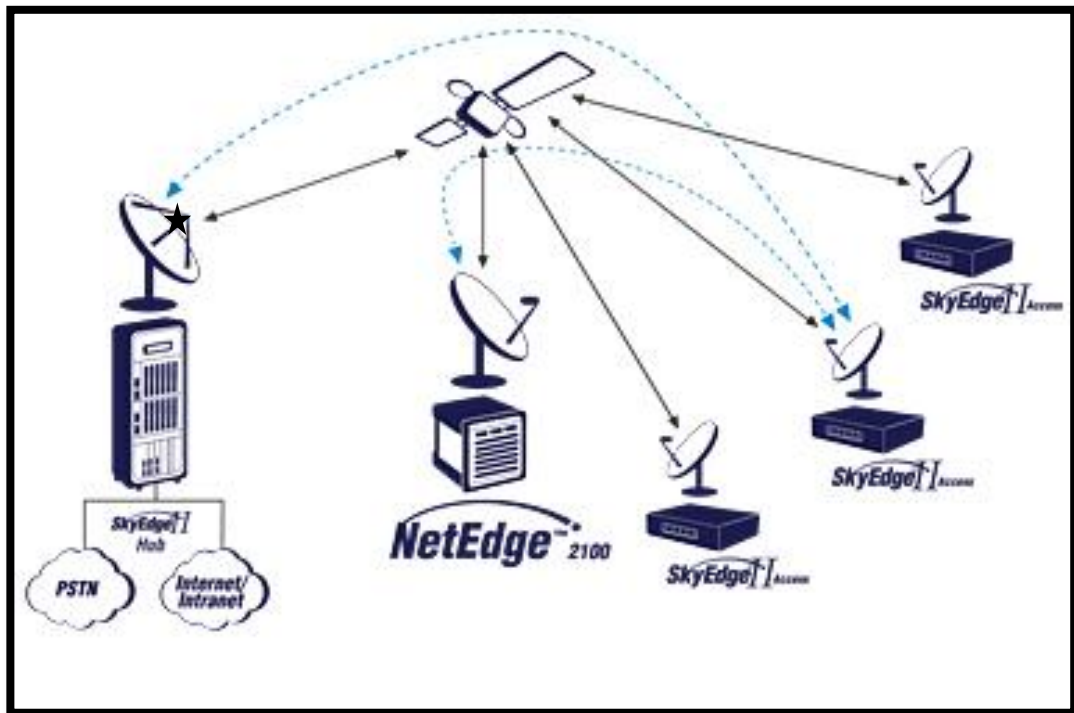


Figura 1.1 Representación red VSAT con *hub* central.

Fuente Sitio web de Gilat Satellite Networks [1]. (★, Antena satelital, Estación Terrena).

El control que gobierna el funcionar de la Antena satelital en la Estación Terrena en Tarbaca, fue implementado en la década de los años 80 y su tecnología se encuentra ya obsoleta, aparte de esto, presenta una grave falla de funcionamiento cuando se va la electricidad. En la Estación Terrena se cuenta con plantas de respaldo de diesel en caso de un corte en el fluido eléctrico, sin embargo, en el instante en que entran a funcionar las plantas y cuando se restablece la alimentación eléctrica nuevamente, el sistema de control de la antena se reinicia y pierde el posicionamiento correcto que debería tener. Debido a esto, alguien del personal de la estación debe llegar hasta allá, y reajustar el sistema para que vuelva a su correcto funcionamiento.

El traslado de personal se debe realizar en cualquier momento, en cualquier fecha de año, lo que representa gastos para la empresa en cuanto a conceptos de transporte y horas extras.

Un fallo en la comunicación entre el *hub* (Tarbaca) con la red VSAT-ICE provocaría que muchas comunidades y empresas perdiesen la señal y quedasen incomunicadas, lo que implicaría pérdidas de dinero para el ICE ante la afectación de los servicios provistos.

Actualmente, no existe comunicación remota entre el sistema de control y algún plantel del ICE, por lo tanto, la implementación de este, así como el proceso de reingeniería del control actual traería consigo beneficios, entre los que se pueden mencionar están:

- Recorte de gastos en envío de personal.
- Monitoreo en vivo del sistema.
- Control desde cualquier lugar con acceso a la red del ICE.
- Disminución física considerable de la unidad de control al reemplazar la tecnología.
- Mayor capacidad de funcionamiento debido a tecnología reciente.
- Mantener registro histórico de los posicionamientos de la antena.

Por estas razones, se decidió la realización de un prototipo de un nuevo control remoto del sistema que gobierna la antena satelital, esto con miras a que sea desarrollado e implementado a futuro.

1.2. Solución seleccionada.

El proyecto planteado permitirá posicionar la antena tanto en ángulo de elevación así como en el ángulo azimut, esto de manera remota desde cualquier lugar con acceso a la red de internet del ICE y que sea usuario activo de este. Además, debe permitir mantener un historial de los posicionamientos registrados para llevar control acerca de sus últimas ubicaciones. También debe de contar con una parada de emergencia para detener el sistema, y un enclave de la antena para situaciones donde haya presencia de ciclones o huracanes para evitar que esta sufra daños.

A continuación se muestra el diagrama de bloques de la solución propuesta.

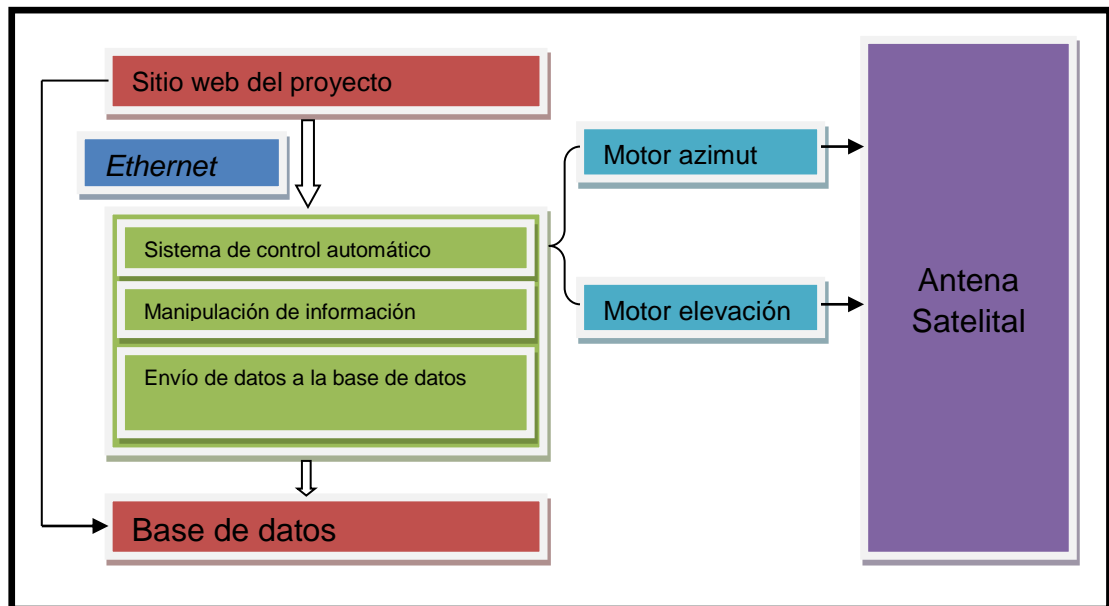


Figura 1.2 Diagrama de bloques de la solución propuesta.

Mediante el uso de un kit de desarrollo (SBC65EC) se realizará el manejo vía Ethernet del sistema de control en la Estación Terrena, mientras que en la planta del ICE, en una computadora con acceso a la red de la empresa, se podrá posicionar la antena desde el sitio web del proyecto, revisar el historial de posicionamientos, anclar la antena o detener el sistema.

Los ángulos de posición ingresados desde el sitio web son almacenados en la base de datos ORACLE® creada específicamente para este proyecto. La base de datos se encuentra en los servidores del ICE en San Pedro. La página web fue creada en el programa NetBeans® bajo la plataforma Jakarta (JAVA) Struts®, esto se da bajo el formato de trabajo requerido en la empresa.

En la siguiente figura (1.3) se detalla la secuencia de pasos seguida para la realización del proyecto en su totalidad.

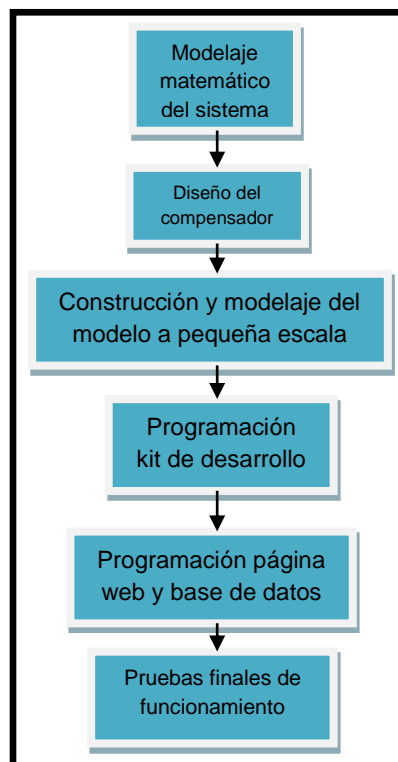


Figura 1.3 Secuencia de la solución del problema presente.

Capítulo 2 : Meta y Objetivos.

En este capítulo se detallan la meta y los objetivos planteados al inicio del planeamiento del proyecto, así como sus indicadores para determinar el cumplimiento de estos.

2.1. Meta.

Implementar un prototipo de sistema de control y comunicación de la antena satelital del ICE en la Estación Terrena de Tarbaca que permita reemplazar el utilizado actualmente.

2.2. Objetivo general.

Diseñar el control automático de una antena satelital que permita gobernar el movimiento de la antena tanto en el eje azimutal como en el de su elevación, con una interfaz de usuario a través de Internet para la comunicación, monitoreo y control de manera remota del sistema.

2.3. Objetivos específicos.

2.3.1. Objetivos de diseño.

1. Determinar el modelo matemático de la antena satelital a la cual se le va a desarrollar el nuevo control automático.
2. Diseñar el control automático de la antena satelital que sea capaz de gobernar el comportamiento de la antena para su correcto posicionamiento.
3. Simular el control de manera que se pueda verificar el correcto funcionamiento del diseño.

2.3.2. Objetivos de software.

1. Desarrollar una aplicación web que permita el monitoreo y control de posicionamiento de la antena satelital.

2.3.3. Objetivos de documentación.

1. Realizar un estudio del sistema de control de la antena que actualmente opera, de manera que se pueda realizar un acople con el nuevo diseño, para tener así un sistema redundante.

Capítulo 3 : Marco teórico.

En este capítulo se detallan las bases teóricas de los conceptos utilizados, así como del equipo que se utilizó para darle solución al proyecto aquí descrito y otros conceptos relevantes para el entendimiento del mismo.

3.1. Descripción del sistema o proceso a mejorar.

Tal y como se mencionó en la sección anterior, el ICE desea poder realizar el manejo de la antena en Tarbaca de manera no presencial para reducir costos, y lograr además reducir el tiempo de espera existente desde el momento en que se presenta la falla hasta que sea solucionada por algún funcionario del ICE.

3.2. Antecedentes bibliográficos.

Como antecedente bibliográfico, se cuenta únicamente con la información del sistema actual de control, debido a que se pretende hacer una reingeniería del sistema actual presente en Tarbaca. Esta documentación se puede encontrar en la sección de anexos al final de este documento.

3.3. Descripción de los principales principios físicos y/o electrónicos relacionados con la solución del problema.

3.3.1. Antena Satelital¹.

Una antena satelital es un tipo de antena parabólica que tiene comunicación a dos vías hacia y desde los satélites que orbitan alrededor de la Tierra en el espacio exterior.

En una antena parabólica, la información de entrada se refleja en el plato y es dirigida hacia el brazo donde se encuentra el punto focal de esta para luego ser canalizada mediante una conexión cableada, a través de, una guía de onda hasta un receptor que se encarga de manipular la información. Para cuando la información es más bien saliente, se sigue el proceso contrario, desde el transmisor, la información viaja por cable hasta el brazo de la antena que dispara el rayo al plato y este lo refleja hacia el exterior.

Se pueden citar varios tipos de antena parabólica, entre los que destacan:

- **Foco primario:** el reflector está centrado respecto al foco.
- **Foco desplazado:** el reflector se encuentra desplazado del centro del foco para evitar hacerle sombra.
- **Cassegrain:** posee un segundo reflector a la par del foco. Este reflector es convexo.
- **Gregorian:** posee también un segundo reflector a la par del punto focal, pero este es cóncavo.

¹ Referencia tomada de [2] y [3].

En la figura 3.1 se puede observar un diagrama de estos cuatro tipos de antena parabólica.

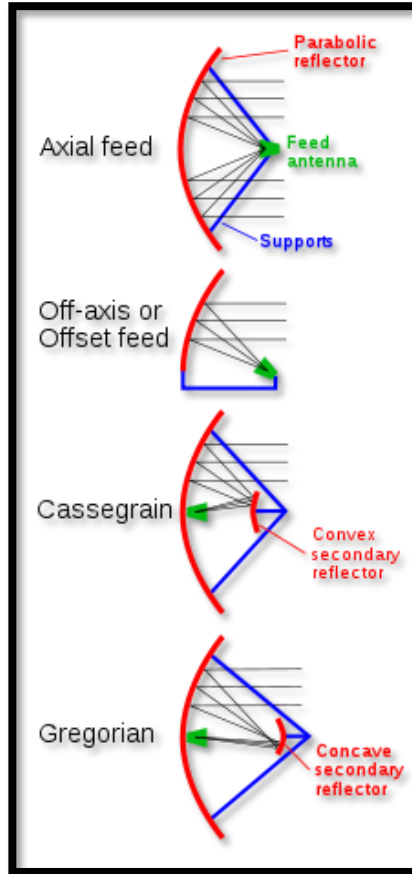


Figura 3.1 Tipos de antena parabólica. En orden de arriba hacia abajo: Foco primario, foco desplazado, Cassegrain y Gregorian.

Fuente sitio web Wikipedia [3]

3.3.2. Estación Terrena².

Una Estación Terrena es una estación cuya función principal es establecer la base terrestre para la comunicación con satélites espaciales. En todo momento debe poseer línea de vista clara y despejada con el satélite al cual pretende comunicarse.

Entre las principales funciones que se realizan en una Estación Terrena son las siguientes:

- a. Seguimiento del satélite.
- b. Telemetría para ubicar correctamente la posición del satélite.
- c. Procesamiento de los datos adquiridos.
- d. Funcionamiento como enlace central o *hub* para otras estaciones o redes de comunicación.

En la figura 3.2 se puede observar un esquema general de las secciones que existen en una Estación Terrena.

² Referencia tomada de [4] y [5]



Figura 3.2 Secciones presentes en una Estación Terrena.

Fuente documento propiedad de Universidad Politécnica Salesiana [5]

3.3.3. Protocolo UDP³.

El protocolo UDP (*User Datagram Protocol*) es un estándar TCP/IP, y como su nombre lo indica está basado en el intercambio de datagramas o paquetes de datos. No necesita de confirmación ni control de flujo para el intercambio de información, haciendo que se agilice el proceso, sin embargo, no cuenta tampoco con confirmación de entrega y recepción, lo que puede llevar a la pérdida de paquetes en el camino.

³ Referencia tomada de [6], [7] y [8]

Un mensaje UDP se encapsula dentro de un datagrama IP para ser enviado a través de la red. En la siguiente figura se ilustra lo anterior.

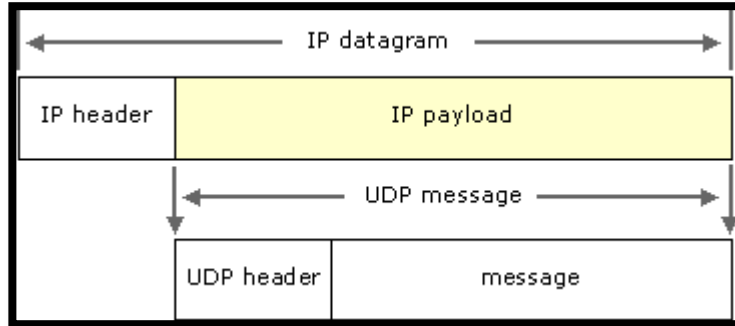


Figura 3.3 Encapsulado UDP.

Fuente sitio web Microsoft TechNet [6]

Los programas basados en el protocolo UDP tienen la capacidad de recibir varios mensajes al mismo tiempo.

En la figura 3.4 se observa la composición de un paquete UDP, formada por la dirección del puerto de origen y destino, longitud, suma de verificación y finalmente los datos. Los campos de puerto de origen y suma de verificación son campos opcionales.

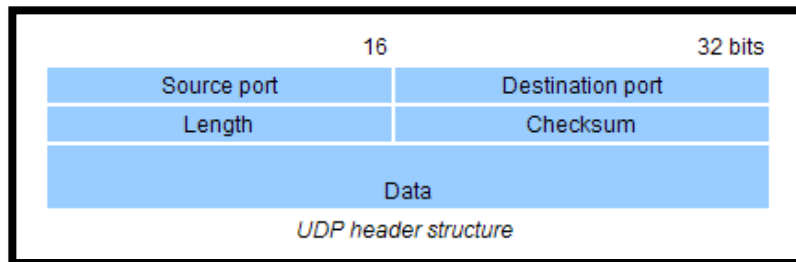


Figura 3.4 Estructura de un paquete UDP.

Fuente sitio web www.protocols.com [8]

3.3.4. Tarjeta de desarrollo SBC65EC⁴.

La tarjeta de desarrollo (o kit) SBC65EC pertenece a la empresa Modtronix®. Es un sistema embebido (empotrado) que tiene como principal agregado la capacidad de funcionar como un pequeño servidor web. Es el dispositivo que funciona como corazón del sistema. Posee un micro controlador PIC18F6627.

Entre sus principales características se encuentran:

- 32 puertos de entrada/salida para propósito general.
- 12 convertidores ADC de 10 bits programables.
- 4 salidas de PWM de 10 bits programables.
- 2 interfaces USART y 1 interfaz I2C.
- Posee capacidad de servidor web.
- Se permite la configuración y control mediante página web.
- Acepta comandos vía UDP y TCP.
- Posee conector RJ-45 para conexión Ethernet de 10Mb (10BaseT).

⁴ Referencia tomada de [9]

En la siguiente figura se muestra una imagen del kit de desarrollo SBC65EC.



Figura 3.5 Kit de desarrollo SBC65EC.

Fuente sitio web Modtronix® [9]

3.3.5. Motor de inducción⁵.

El motor asíncrono de inducción es un motor de corriente alterna, creado por Nikola Tesla, y fue presentado en el año de 1888. Su funcionamiento se detalla a continuación.

En las bobinas inductoras presentes en el estator, circula una corriente que permite la inducción de un campo magnético, envolviendo este campo al rotor del motor, tal y como lo explica la *Ley de Faraday*.

⁵ Referencia tomada de [10] y [11].

Entonces, entra en juego el *Efecto Laplace*, el cual dice que si un conductor por el que atraviesa una corriente se encuentra inmerso en un campo magnético, experimenta una fuerza que hace que este se ponga en movimiento.

La velocidad de rotación del campo magnético está dada por:

$$n_{sinc} = \frac{120 * f_e}{P} \quad (3.1)$$

Donde f_e es la frecuencia de operación, y P es el número de polos.

Cuanto mayor es la velocidad de un motor de inducción, mayor es el rozamiento que se presenta en el motor y por lo tanto, mayores son las pérdidas. Entre las pérdidas que se pueden mencionar están:

- Pérdidas en el cobre del estator.
- Pérdidas en el núcleo.
- Pérdidas en el cobre del rotor.
- Pérdidas por rozamiento propio y con el aire.
- Pérdidas misceláneas.

En la siguiente figura se pueden observar las partes básicas de un motor de inducción de corriente alterna.

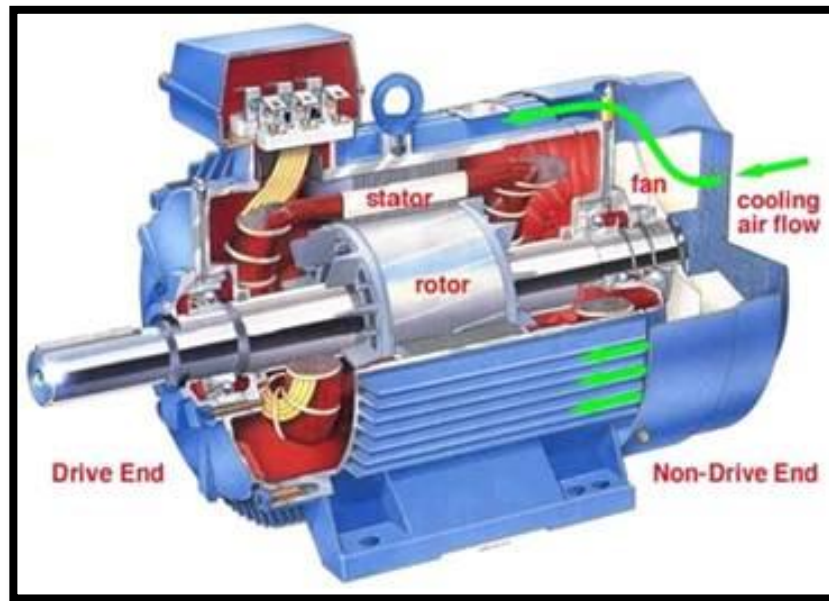


Figura 3.6 Motor de inducción de corriente alterna.

Fuente sitio web Monografías [14]

3.3.6. Sistema de control de lazo cerrado.⁶

Es un sistema donde las variables presentes en el sistema o en la salida de este, mediante la realimentación entrada-salida influyen en el control del sistema. Un sistema de lazo cerrado permite que el control sea más exacto. Una de las mayores ventajas de un sistema de lazo cerrado frente a uno de lazo abierto es que la aplicación de la realimentación permite que la respuesta del sistema sea prácticamente insensible a las perturbaciones externas.

⁶ Referencia tomada de [15]

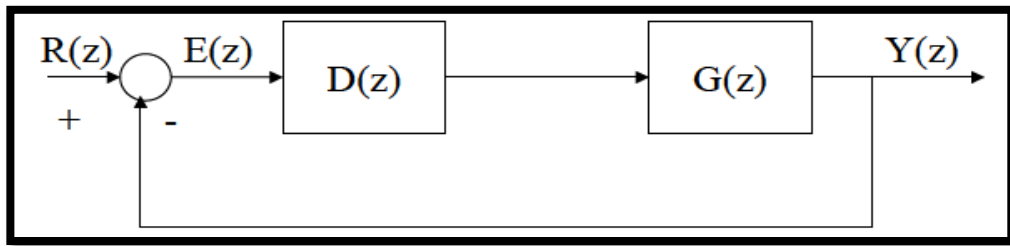


Figura 3.7 Diagrama de un sistema de lazo cerrado con realimentación unitaria.

Fuente sitio web Ing. Eduardo Interiano [15].

3.3.7. Definiciones varias de control automático⁷.

- **Respuesta al escalón:** la respuesta al impulso es la salida que permite caracterizar un sistema, cuando la entrada a este es una función escalón unitario. La función escalón unitario se caracteriza por tener un valor de uno para un tiempo positivo, mayor a cero, y un valor de salida de cero para tiempos negativos, menores que cero.
- **Sobreimpulso:** es el porcentaje extra superior al máximo valor de la respuesta de un sistema en escalón unitario.
- **Error de estado estacionario:** es la diferencia entre la salida y la referencia en estado estable. En el mundo real, debido a fricciones y pérdidas, el error de estado estacionario es prácticamente imposible que sea cero, estando presente en todos los sistemas de control. Generalmente se corrige para valores debajo de un límite tolerable.
- **Tiempo de subida t_r :** es el tiempo que tarda la respuesta al escalón en pasar de un valor de 10% al 90% del valor final de esta.

⁷ Referencia tomada de [17], [18] y [19]

- **Tiempo de estabilización t_s :** es el tiempo requerido para que la respuesta al escalón disminuya y entre a la banda de porcentaje estable cercana al valor final del escalón.

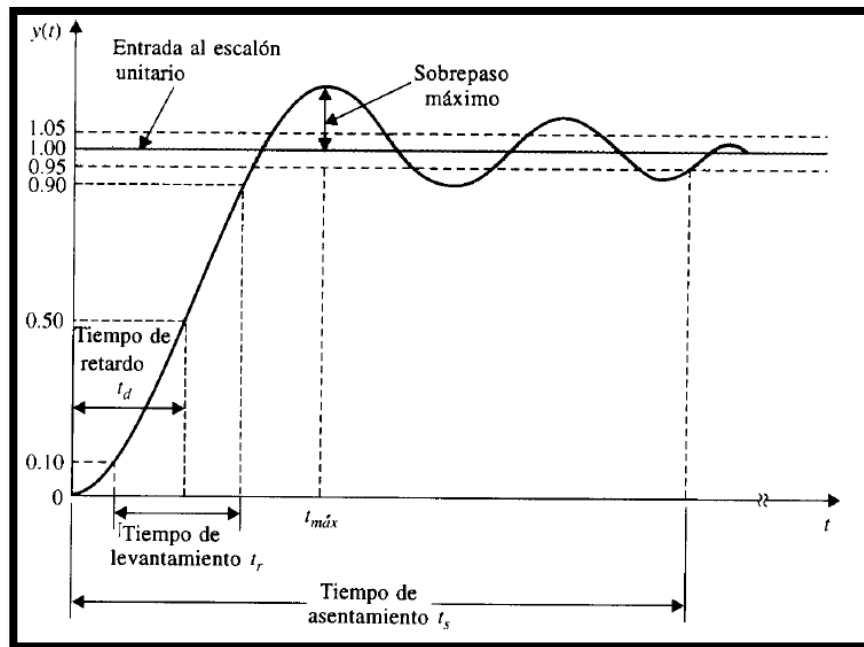


Figura 3.8 Especificaciones en el dominio del tiempo para un sistema ante entrada de escalón unitario.

Fuente Sitio web Ing. Eduardo Interiano [19].

3.3.8. Compensador de adelanto⁸.

El compensador de adelanto es una técnica utilizada para corregir el funcionamiento de un sistema en lazo cerrado, agregando ceros y polos a la función de transferencia de la planta.

Un compensador de adelanto tiene la forma de

$$G_c(j\omega) = K_c \alpha \frac{j\omega\tau + 1}{j\omega\alpha\tau + 1} \quad (3.2)$$

El compensador de adelanto se encarga de mejorar la respuesta transitoria del sistema.

3.3.9. Lugar de las raíces⁹.

El lugar de las raíces es la ubicación de los polos y ceros de la función de transferencia de un sistema. El lugar de las raíces permite realizar el diseño de compensadores mediante la ubicación de polos y ceros en el plano de las raíces. Es una técnica que fue desarrollada por primera vez por Walter R. Evans. Entre las características que permite encontrar están:

- Estabilidad del sistema.
- Frecuencia natural (w_n).
- Coeficiente de amortiguamiento (ζ).

⁸ Referencia tomada de [12]

⁹ Referencia tomada de [13]

En la siguiente figura se muestra un ejemplo del lugar de las raíces en tiempo discreto.

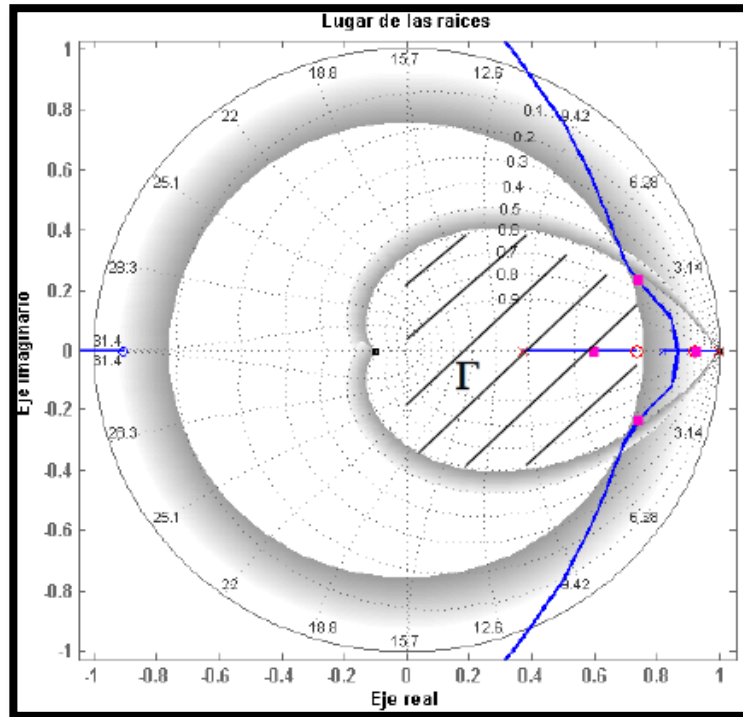


Figura 3.9 Lugar de las raíces en tiempo discreto.

Fuente sitio web Ing. Eduardo Interiano [15]

3.3.10. Control integral¹⁰.

El control integral es el encargado de disminuir y en la medida de lo posible eliminar el error de estado estacionario presente en un sistema dado al que se le quiere hacer determinada compensación de sus características estáticas y dinámicas. El error es integrado, lo que hace que sea promediado, y posteriormente es multiplicado por alguna constante K_i .

¹⁰ Referencia tomada de [16].

La ecuación del control integral está dada por:

$$I_{sal} = K_i * \int_0^t e(\tau) d\tau \quad (3.3)$$

3.3.11. Motor Paso a Paso¹¹.

Es un motor sin escobillas que convierte una serie de impulsos eléctricos en un movimiento rotacional angular discreto, es decir que un impulso en su entrada de alimentación a una bobina se convierte en un desplazamiento de una serie de grados. Este tipo de motores presenta una gran precisión.

Los distintos tipos de motores Paso a Paso que existen son:

- **Rotor de imán permanente:** permite mantener un torque diferente de cero mientras el motor está desconectado.
- **Reluctancia variable:** los dientes del rotor son atraídos a los dientes del estator por electromagnetismo, lo que permite la rotación.
- **Motor híbrido de paso:** es una combinación de los dos tipos de motores anteriores. Su principal característica es que posee una altísima precisión con un paso angular de hasta 1,8°.
- **Bipolares:** Poseen cuatro cables para conectar. Funcionan mediante el cambio de dirección del flujo de la corriente. Necesitan para poder operar un puente H por cada bobina.
- **Unipolares:** Suelen tener cinco o seis cables, y pueden tener un giro horario y anti horario simplemente variando la secuencia de pasos.

¹¹ Referencia tomada de [20].

El motor permite que se utilicen diferentes configuraciones de paso. Entre los principales se pueden mencionar:

- **Wave Drive:** en esta configuración solo una de las fases es activada a la vez.
- **Full Step:** dos fases están siempre activas. Es la operación normal de funcionamiento.
- **Half Step:** se alterna la activación de una y dos fases para incrementar la resolución angular del motor.

En la siguiente figura se encuentra una imagen de un motor paso a paso típico.



Figura 3.10 Motor paso a paso.

Fuente sitio web Wikipedia [20]

3.3.12. Transistor Darlington¹².

El transistor Darlington es un arreglo de dos transistores bipolares *NPN* o *PNP*. Fue desarrollado por Sidney Darlington, ingeniero de los Laboratorios Bell en el año de 1953.

Su principio de funcionamiento radica en que, al estar conectados el emisor del primero con la base del segundo, permitiendo así que la corriente amplificada por el primer transistor sea nuevamente amplificada por el segundo transistor, haciendo que tenga una ganancia de corriente β muy alta, típicamente de alrededor 1000. Su principal desventaja es que la tensión base-emisor V_{BE} es la suma de la de ambos transistores que lo conforman, por lo tanto, la tensión base-emisor para un Darlington de silicio será de aproximadamente 1,3V, el doble de la de un transistor común de 0,65V.

En la figura siguiente se observa el diagrama de un transistor Darlington.

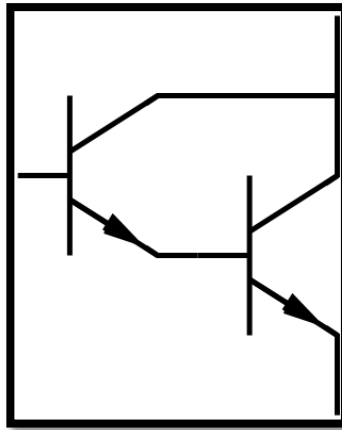


Figura 3.11 Diagrama de un transistor Darlington NPN.

Fuente sitio web Wikipedia [21]

¹² Referencia tomada de [21].

Capítulo 4 : Procedimiento metodológico.

El presente capítulo pretende ser la base para que el lector se oriente acerca de la solución escogida y pueda comprender de mejor forma esta, además se explica cómo y dónde se obtuvo la información pertinente para solucionar el problema. Se brinda también información respecto a la implementación de la solución así como el diseño del prototipo del sistema.

4.1. Reconocimiento y definición del problema.

El primer paso para el reconocimiento de la situación planteada fue la obtención de una breve documentación existente del sistema actual de control, en donde se observaban los componentes principales del sistema actual, un esquema básico del funcionamiento y el tipo de motores que se utilizan.

Posteriormente, luego de un periodo de aprendizaje del lenguaje de programación requerido por la empresa, se realizó la primera visita a la Estación Terrena en Tarbaca, Aserrí. En esta primera visita se obtuvo la documentación completa existente del sistema de la antena. Posteriormente, se definieron las funciones principales con las que debía contar el prototipo con el asesor de la empresa, Ing. Luis Guillermo Matarrita.

Para la siguiente visita a la Estación Terrena, se contó con la ayuda de uno de los encargados del sistema, explicando la forma de funcionamiento del sistema y un poco de la historia de este, así como de las limitaciones con las que cuenta el sistema.

Ya con todo el panorama esclarecido, se procedió a trazar un nuevo cronograma de actividades, sabiendo ya los puntos importantes que debían seguirse en la ruta crítica del proyecto y los que podían ser pospuestos.

4.2. Obtención y análisis de la información.

La información inicial fue provista por la empresa, los ingenieros que forman parte de la Dirección Técnica de Operación y Mantenimiento, DTOM.

Una vez definidos claramente los objetivos y pasos a seguir, se procedió a la investigación bibliográfica. Durante este proceso, se fueron descubriendo limitantes y aciertos en diferentes escenarios de soluciones, y así se determinó cuál de todas las posibles soluciones, en el caso específico del modelo a pequeña escala era la más conveniente para el proyecto.

Al ser un prototipo, y contando con la mayoría de recursos dentro de la empresa, el costo del mismo no fue un factor determinante para la escogencia de la solución.

La solución que se ajustó lo más posible a los requerimientos y objetivos planteados fue la escogida.

4.3. Evaluación de las alternativas y síntesis de la solución.

La evaluación de las soluciones para determinar su validez o su descarte se basó en primera instancia con la experiencia previa y los conocimientos adquiridos durante la universidad, como segundo punto de referencia, la investigación bibliográfica se fueron cercando las soluciones y los puntos más importantes en la elaboración de la solución.

El desarrollo del diseño del control fue realizado en el programa MATLAB®, mientras que el modelo a pequeña escala fue sometido a pruebas controladas de laboratorio para certificar su funcionamiento correcto y deseado. Se debió de hacer descartes por falta de componentes que no pudieron ser localizados en diferentes electrónicas del Gran Área Metropolitana.

Para el diseño del control, se tomaron ejemplos de casos similares resueltos en libros o foros de electrónica para obtener un lineamiento común entre todos, además se debió elegir en base al comportamiento observado en las simulaciones de la planta el o los respectivos compensadores adecuados para obtener el funcionamiento deseado.

4.4. Implementación de la solución.

La implementación de la solución del proyecto consta de dos grandes partes. La primera de ella es el estudio de la planta y el rediseño del control de esta, mientras que la segunda parte es el montaje del prototipo a pequeña escala del sistema, para verificar el funcionamiento del sitio web y la base de datos dedicada a esta.

4.4.1. Rediseño del control.

Para la solución de esta primera parte, se obtuvo información relevante de la planta bajo estudio, en este caso, la antena satelital de la Estación Terrena. Con esta información se procedió a obtener la función de transferencia para una entrada escalón.

Una vez que con la función de transferencia, se analizó el sistema y se determinó qué y cuantos tipos de compensadores eran necesarios para conseguir el comportamiento deseado.

Finalmente, se simuló la planta con los compensadores y se verificó que el sistema respondiera tal y como debía hacerlo cuando se definieron los parámetros nuevos de control.

4.4.2. Montaje del prototipo a pequeña escala.

La sección del montaje del prototipo requirió de la conclusión exitosa de varias etapas de trabajo. Estas son:

4.4.2.1. Sitio web y base de datos.

Se desarrolló en la plataforma NetBeans® bajo el lenguaje Java (Jakarta) Struts®. La base de datos fue implementada en ORACLE®. La aplicación se basó inicialmente en ejemplos realizados en la empresa y ejemplos de libros de texto del lenguaje. Los diferentes módulos fueron probados y enlazados con la base de datos para corroborar el funcionamiento correcto de la aplicación.

4.4.2.2. Kit de desarrollo SBC65EC.

Se implementó en el kit de desarrollo el envío de los datos UDP y el control de posición de los motores del modelo a pequeña escala. En la empresa ya se contaba con varias de estas tarjetas de desarrollo.

Se tomaron ejemplos disponibles en el sitio web del desarrollador del kit, Modtronix®, y también con programas propios desarrollados en los cursos de laboratorio de la universidad. El kit de desarrollo cuenta con interfaz Ethernet.

4.4.2.3. Aplicación JAVA® para recepción de datos UDP.

La aplicación fue desarrollada también en la plataforma NetBeans® y se encargó de recibir los datos UDP provenientes del kit de desarrollo, interpretarlos y almacenarlos en la base de datos ORACLE®.

4.4.3. Motores de posicionamiento.

Se utilizaron dos motores paso a paso para el montaje del sistema a pequeña escala para hacer las veces de los motores reales del sistema que ubican la antena a lo largo de un ángulo azimutal y de elevación.

4.5. Reevaluación y rediseño.

Las principales mejoras para perfeccionar el prototipo desarrollado del control de la antena satelital son las siguientes:

- Utilizar sensor óptico para hacer un sistema redundante de posicionamiento, y no solo lógico como se implementó.
- Correr el servidor Web del proyecto en un lugar remoto, puesto que fue implementado en modo local.

- Investigar el rastreo de señales de frecuencia o luminosas para agregar al prototipo, y cubrir así una de las funciones con las que cuenta el sistema real del control en Tarbaca.
- Implementar el prototipo en un circuito impreso en lugar del protoboard, a pesar de que la circuitería externa es mínima.

Capítulo 5 : Descripción detallada de la solución.

En este capítulo se le expone al lector las soluciones preliminares y la elección final de esta. Además, se detalla la solución implementada para cada uno de los bloques que conforman el diagrama total del proyecto, divididos en software y hardware, y que están conformes a los objetivos planteados al inicio de este documento.

En la siguiente figura se presenta el diagrama de bloques general del proyecto.

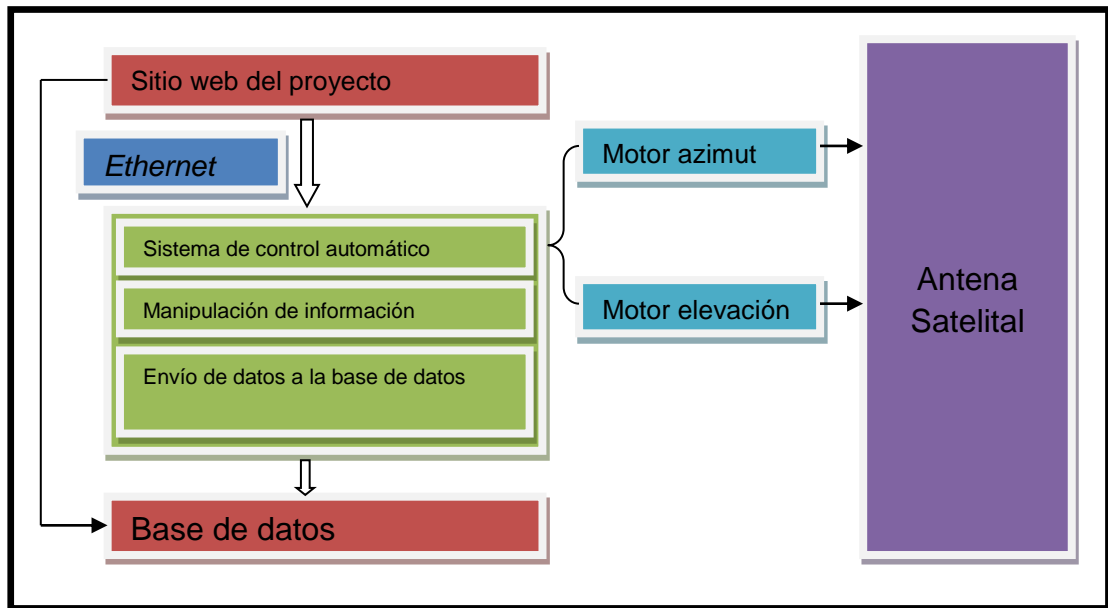


Figura 5.1 Diagrama general del proyecto.

En la siguiente figura se detalla el funcionamiento del sistema. Más adelante se explica cómo funciona el sistema.

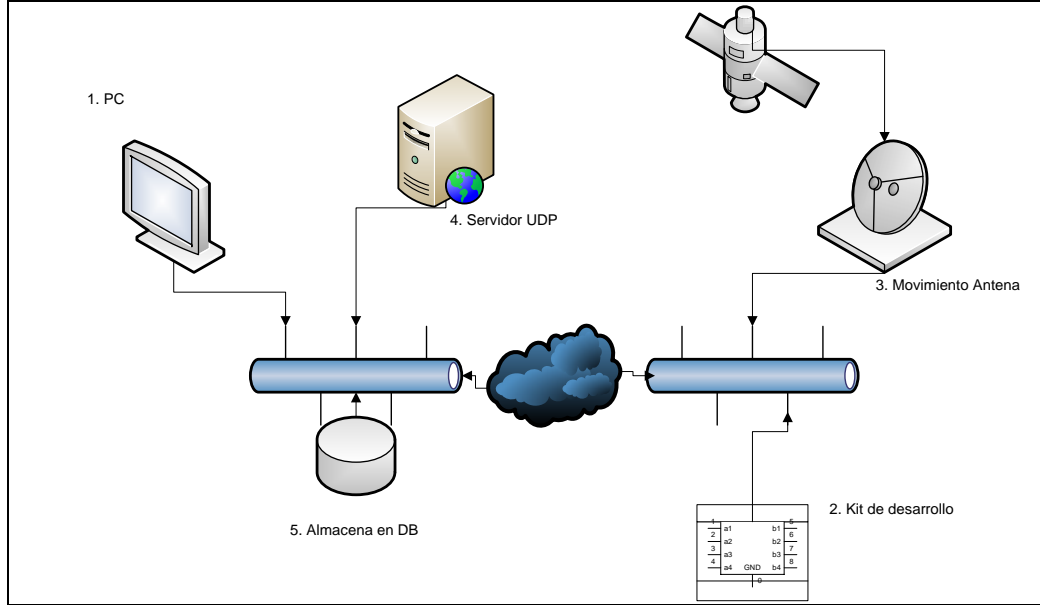


Figura 5.2 Diagrama del funcionamiento secuencial del sistema.

El funcionamiento del sistema es el siguiente:

1. Se ingresan los ángulos deseados en la página web del sistema.
2. La información viaja al kit de desarrollo.
3. Acá mediante esa información se gobiernan los motores para posicionar la antena.
4. El kit de desarrollo envía vía Ethernet por la Nube los valores de ángulos ingresados al servidor UDP.
5. El servidor decodifica la información y finalmente la almacena en la base de datos dedicada del proyecto.

5.1. Análisis de soluciones y selección final.

A continuación se detallan las opciones que fueron consideradas para la solución del proyecto.

5.1.1. Kit de desarrollo.

El kit de desarrollo es el que permite al sistema enlazar la parte física con la interfaz de usuario, puesto que a través de él se manejan los motores con los ángulos ingresados en la página web, y que también envía para almacenarlos en la base de datos. En el mercado existen muchas opciones en cuanto a kits de desarrollo. Aunque la empresa contaba ya con una determinada marca, igual se exploraron algunas otras opciones para verificar la viabilidad de su uso.

5.1.1.1. Arduino Ethernet Shield.

Es un micro controlador de código abierto. Posee un procesador Atmel AVR.

- Ventajas:
 - Posee conexión por puerto Ethernet.
 - Lector de tarjetas SD.
 - Muy bajo costo.
- Desventajas:
 - Pocos puertos digitales de E/S (14).
 - Entradas analógicas limitadas (5).

5.1.1.2. SBC65EC.

Posee un micro controlador PIC18F6627.

- Ventajas:
 - Conexión por puerto Ethernet.
 - Posee 32 puertos de E/S.
 - 12 puertos analógicos.
 - EEPROM 64Kbytes.
- Desventajas:
 - Mayor costo que el Arduino.
 - No posee protección contra interferencias electromagnéticas.

5.1.1.3. PICDEM.net 2.

De la familia Microchip, posee un micro controlador PIC18F97J60.

- Ventajas:
 - 2 interfaces Ethernet.
 - LCD alfanumérico.
 - Sensor de temperatura.
- Desventajas:
 - EEPROM de 32Kbytes.
 - Mayor precio de los tres kits.

Se decidió entonces utilizar la tarjeta de desarrollo SBC65EC de la empresa Modtronix®, en primer lugar porque ya se contaba con ella en la empresa, y en segundo lugar porque las otras opciones presentaban características muy similares y prácticamente ninguna ventaja para hacer frente a los requerimientos del proyecto en cuestión.

5.1.2. Manejo de la etapa de motores.

Para el manejo de los motores del modelo a pequeña escala se tuvieron únicamente dos opciones en consideración.

5.1.2.1. Puente H.

Circuito electrónico que permite al motor tener la capacidad de girar en ambos sentidos.

- Ventajas:
 - Permite doble sentido de giro.
- Desventajas:
 - Mayor costo.
 - Se requiere uno por cada bobina del motor.

5.1.2.2. ULN2003.

Arreglo de siete transistores Darlington y un diodo por cada transistor.

- Ventajas:
 - Posee transistores de alta corriente y tensión.
 - Posee diodos de rodamiento libre.
 - Interactúa a 5V TTL.
 - Bajo costo
- Desventajas:
 - Giro invertido debe hacerse en programación.

Se decidió escoger el circuito integrado ULN2003 puesto que el sentido de giro que provee el Puente H no era necesario al tener motores unipolares, además del costo, al necesitarse únicamente 2 IC ULN2003, mientras que si fuera el Puente H, se hubiesen necesitado 4, uno por cada bobinado de los dos motores.

5.1.3. Envío de paquetes de datos.

Para el envío de la información desde el kit de desarrollo hacia el servidor donde se aloja la página web y la base de datos del proyecto se contó con dos opciones, UDP y TCP.

5.1.3.1. TCP:

- Ventajas:
 - Proporciona un envío más confiable y ordenado de paquetes de bits.
 - Garantiza la entrega de los paquetes.
 - Brinda acuse de recibo.
 - Intercambio bidireccional de la información.
- Desventajas:
 - Mayor latencia para el envío de paquetes.
 - Puede llegar a congestionar la red por el reenvío de los paquetes perdidos.

5.1.3.2. UDP:

- Ventajas:
 - Envío más rápido de los paquetes de datos.
 - Fácil implementación.
- Desventajas:
 - No brinda información acerca de si el paquete llegó, si se desordeno o si se duplicó.
 - La información solo puede viajar en una dirección a la vez.

Se decidió utilizar el protocolo UDP por su facilidad para la implementación en el kit de desarrollo, además de la necesidad de tener un envío rápido de la información para un óptimo funcionamiento en tiempo real.

5.2. Descripción del hardware.

En esta sección se detalla el funcionamiento del módulo implementado para el manejo de los motores paso a paso del modelo a pequeña escala del sistema.

5.2.1. Manejo de motores.

En la figura a continuación se muestra el esquema de conexión para el manejo del par de motores utilizados.

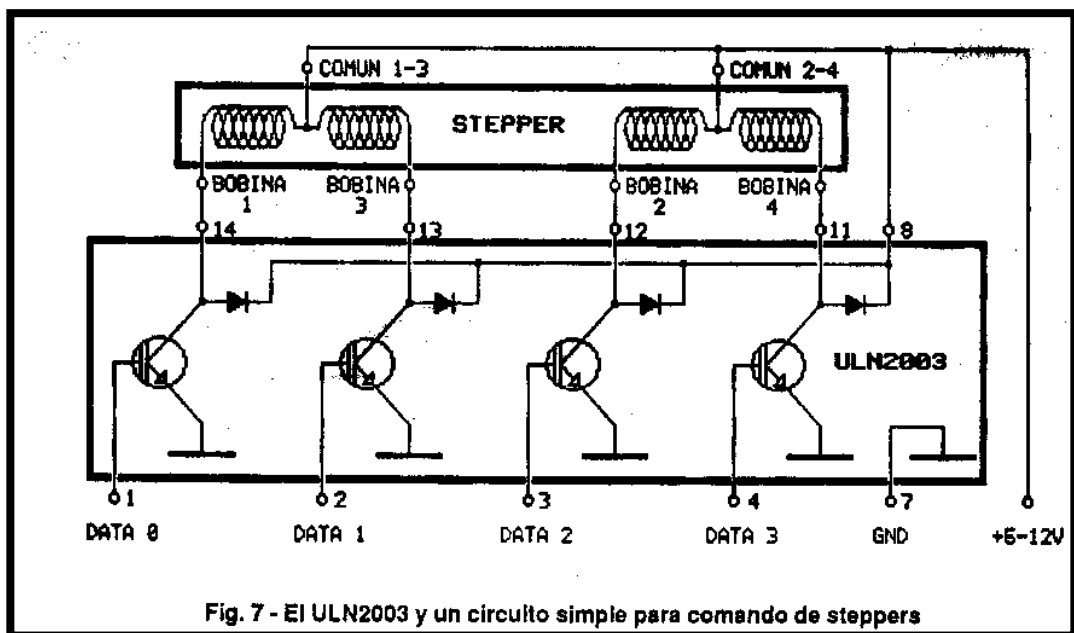


Figura 5.3 Diagrama de conexión de los motores paso a paso.

Fuente sitio web Facultad de Ingeniería, Universidad de la Plata [22]

Como se puede observar, el circuito es muy simple, las salidas digitales del kit de desarrollo se conectan a las entradas TTL del circuito integrado ULN2003 (DATA 0-3), y sus correspondientes salidas se conectan a las bobinas de los motores.

La salida digital del kit es una señal de 5V que entra a la base de cada uno de los transistores Darlington. Los siete Darlington están arreglados con colector abierto y emisor común. Los diodos de rodamiento libre se encuentran conectados a los emisores.

A la salida del ULN2003 se agregaron resistencias de *Pull-Down* para asegurar la salida lógica de cero y evitar que algún valor intermedio bloquee el motor activando alguna de las bobinas.

5.2.2. Ayudas audiovisuales.

Se implementaron dos ayudas para el usuario, una visual y otra auditiva. La ayuda visual se activa única y exclusivamente cuando el sistema está en el modo de Parada (STOP), con un LED encendido fijo, y para cuando el sistema está en modo de Anclaje (STOW) con un LED parpadeante.

La ayuda auditiva fue implementada con un parlante que permite al usuario, mediante una alarma sonora, darse cuenta cada vez que la antena cambia de posición, sea en modo normal, STOW o STOP.

5.3. Descripción del software.

Esta sección brinda información detallada acerca del software implementado en la solución del proyecto, incluidas explicaciones y diagramas de flujo de los algoritmos desarrollados.

El desarrollo del software se dividió en cuatro categorías, la página web creada en JAVA Struts®, la aplicación para recibir los paquetes UDP creada en JAVA®, el firmware del kit de desarrollo para el control de los motores y envío de paquetes UDP, creada en MPLAB® y finalmente la base de datos creada en ORACLE®.

En la figura 5.4 de la página siguiente se presenta el diagrama general del funcionamiento del software.

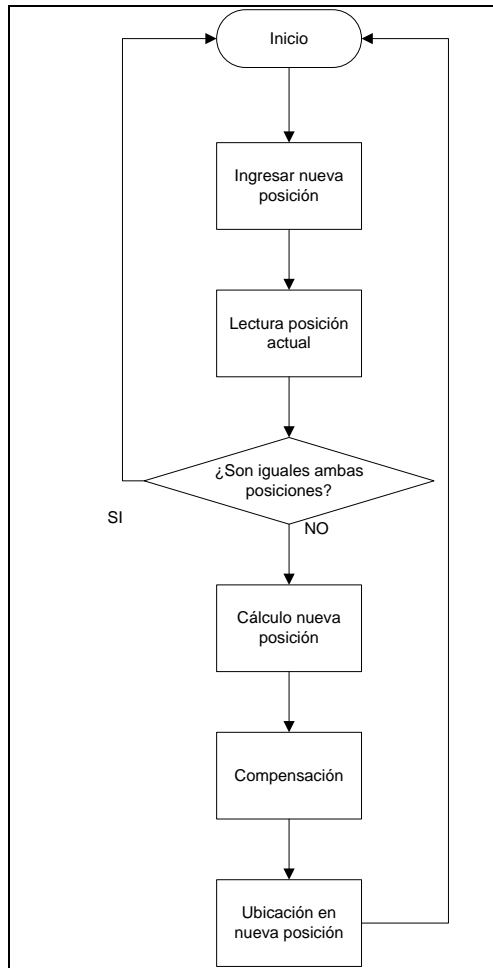


Figura 5.4 Diagrama general del software.

5.2.1. Envío de datos UDP.

El envío de los datos UDP se realizó en el kit de desarrollo utilizado, mediante el uso de la librería UDP que posee. En la figura 5.5 se muestra el diagrama del algoritmo para esta tarea, y en la figura 5.6 un extracto del código.

El primer paso para el envío de un paquete UDP es la creación del socket UDP, que indica que puerto envía y recibe los datos. El siguiente paso es la creación del nodo UDP en donde se especifican la dirección IP y MAC del servidor remoto.

Finalmente, se verifica que los datos a enviar sean válidos y que el socket esté listo para enviar. De ser positivo, se envían los datos UDP al destino indicado en el nodo UDP y se limpia el buffer de datos.

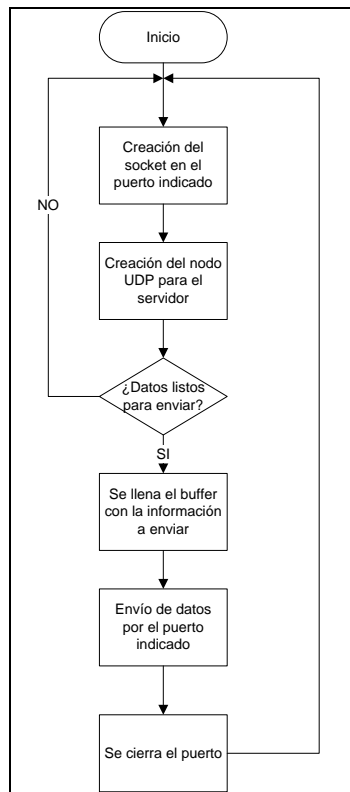


Figura 5.5 Diagrama de flujo para el envío de paquetes UDP.

```

////////////////////////////////////
void UDPServer(UDP_SOCKET udpSocketUser){
    if(LATGO == 0 && LATG1 == 0){
        valorLeido = appcfgGetc(VALOR1);
        valorLeido2 = appcfgGetc(VALOR2);
        if((valorAnt != valorLeido)|| (valorAnt2 != valorLeido2)) {

            if (UDPIsPutReady(udpSocketUser)) {
                if(MAY == 1){
                    envio2B = valorLeido2;

                    UDPPut(temporal_1);
                    UDPPut(valorEnviado);
                    UDPPut(envio2B);
                    UDPPflush();

                    valorAnt = valorLeido;
                    valorAnt2 = valorLeido2;
                    StackTask();
                } else {
                    envio2B = valorLeido2;

                    UDPPut(temporal_1);
                    UDPPut(valorLeido);
                    UDPPut(envio2B);
                    UDPPflush();

                    valorAnt = valorLeido;
                    valorAnt2 = valorLeido2;
                    StackTask();
                }
            }
        }
    }
}

```

Figura 5.6 Extracto del código desarrollado para el envío de paquetes UDP.

5.3.1. Recepción de datos UDP.

La recepción de los datos UDP provenientes del kit de desarrollo se realizó en el programa NetBeans®. Esta aplicación se encarga de aceptar los datos por el puerto estipulado, se interpretan y se manipulan y finalmente son almacenados en la base de datos del proyecto. En la figura 5.8 se observa el diagrama de flujo de esta aplicación, y en la 5.7 un extracto del código realizado.

El primer paso para crear el servidor UDP consiste en la asignación del puerto por el cual se escuchará y se crea también un socket asignado al número de puerto anterior.

Del kit de desarrollo se envían 3 bytes, conteniendo dos de ellos el ángulo azimut y el tercero el ángulo de elevación. Una vez recibido por la aplicación, se concatenan los dos primeros para formar el número de 10 bits, esto porque para azimut, los ángulos llegan hasta 360° y posteriormente se concatena este número de 10 bits con el ángulo de elevación, separados uno de otro por un carácter especial, es este caso fue un punto y coma, “;”. Previo a esta sección los datos se convierten de entero a string decimal.

Se separan mediante la instrucción *StringTokenizer* en dos strings cada vez que encuentra el carácter especial. Cada uno de los strings separados se convierte a su valor entero. Se asigna entonces la hora y fecha del evento, y se procede a almacenar en la base de datos, para lo que se debe conectar a esta primero, y posteriormente se ingresan los datos de los dos ángulos ingresados, la fecha y la hora en que fueron ingresados.

```
switch(control){
    case 0:
        concatenar = concatenar.concat(variable);
        control = 1;
        break;
    case 1:
        concatenar = concatenar.concat(variable);
        concatenar = concatenar.concat(";");
        control = 0;
        break;
}
}
s.close();
double[] DB = new double[recibido.getLength()];
int[] convertido = new int[recibido.getLength()];
String Halfway = " ";
StringTokenizer ST = new StringTokenizer(concatenar, ";");
while (ST.hasMoreTokens()) {
    convertido[contador] = Integer.parseInt(ST.nextToken());
    Halfway = Integer.toString(convertido[contador]);
    DB[contador] = Double.parseDouble(Halfway);
    contador++;
}
```

Figura 5.7 Extracto del código realizado para la recepción de datos UDP.

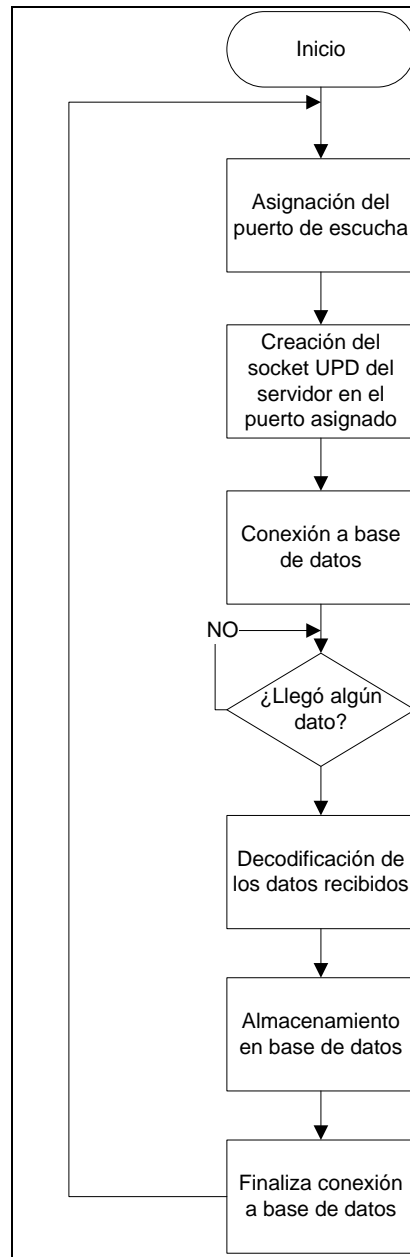


Figura 5.8 Diagrama de flujo para la recepción de datos UDP.

5.3.2. Control de la ubicación de los motores.

Esta sección detalla los tres diferentes algoritmos que requirió cada uno de los motores para poder realizar el posicionamiento. El primero de los tres procedimientos es común para ambos motores, el segundo procedimiento el cual se encarga de calcular la cantidad de pasos a realizar es muy similar entre ambos y el tercer y último procedimiento, el encargado de enviar la lógica de bits para ubicar el motor difiere uno de otro.

5.3.2.1. Identificación del paso y posición actual.

Este proceso lo que realiza es una tabla de conversión, entre el valor del ángulo ingresado desde la página web con el número de pasos requeridos que debe dar el motor para llegar hasta ese punto. Esto permite también saber en qué posición se encuentra ubicado el motor anteriormente al estar almacenado en memoria el valor inmediato anterior obtenido de esta tabla. Es un proceso muy simple basado en un *switch-case*. Para los ángulos azimutales se agrega una variable extra (MAY) que permite indicarle al programa que el valor ingresado requiere de dos bytes para su manejo, esto para evitar algún mal funcionamiento del sistema.

En la figura 5.9 se puede observar a la izquierda un extracto del código utilizado para el ángulo azimutal, y a la derecha para el ángulo de elevación para este procedimiento.

En la tabla 1 y 2 se indica la equivalencia utilizada para la relación ángulo ingresado-número de pasos a dar por el motor de azimut y elevación respectivamente.

Solo se realizó una muestra de 40 ángulos y no los 360° completos para el eje azimutal, y para el de elevación la muestra de 22 ángulos y no los 180°. Esto por ser solo un prototipo. El agregar el resto de ángulos no difiere en lo realizado hasta ahora.

Tabla 1 Tabla de equivalencia para motor azimutal.

Ángulo ingresado	Número de pasos	Ángulo ingresado	Número de pasos
9°	5	189°	105
18°	10	198°	110
27°	15	207°	115
36°	20	216°	120
45°	25	225°	125
54°	30	234°	130
63°	35	243°	135
72°	40	252°	140
81°	45	261°	145
90°	50	270°	150
99°	55	279°	155
108°	60	288°	160
117°	65	297°	165
126°	70	306°	170
135°	75	315°	175
144°	80	324°	180
153°	85	333°	185
162°	90	342°	190
171°	95	351°	195
180°	100	0°	0

Tabla 2 Tabla de equivalencias para motor de elevación.

Ángulo ingresado	Número de pasos	Ángulo ingresado	Número de pasos
0°	0	99°	14
9°	1	108°	16
18°	3	117°	17
27°	4	126°	18
36°	6	135°	19
45°	7	144°	20
54°	8	153°	21
63°	9	162°	22
72°	10	171°	23
81°	11	180°	24
90°	12		

```

////////////////////////////////////
static void MotorAZI(void){          static void MotorELE(void){
if(LATGO == 0 || LATG1 == 0){      if(LATGO == 0 || LATG1 == 0){
MAY = 0;                          switch(appcfigGetc(VALOR2)){
switch(appcfigGetc(VALOR1)){      case 0: pasoEle = 0;
case 9: pasoAzi = 5;              movEle = 1;
MAY = 0;                          break;
movAzi = 1;                       case 9: pasoEle = 1;
temporal_1 = 0;                  movEle = 1;
break;                            break;
case 18: pasoAzi = 10;           case 18: pasoEle = 3;
MAY = 0;                          movEle = 1;
movAzi = 1;                       break;
temporal_1 = 0;                  case 27: pasoEle = 4;
break;                            movEle = 1;
case 27: pasoAzi = 15;           break;
MAY = 0;                          case 36: pasoEle = 6;
movAzi = 1;                       movEle = 1;
temporal_1 = 0;                  break;
break;                            case 45: pasoEle = 7;
case 36: pasoAzi = 20;           movEle = 1;
MAY = 0;                          break;
movAzi = 1;                       case 54: pasoEle = 8;
temporal_1 = 0;                  movEle = 1;
break;                            break;
case 45: pasoAzi = 25;           case 63: pasoEle = 9;
MAY = 0;                          movEle = 1;
movAzi = 1;                       break;
temporal_1 = 0;                  case 72: pasoEle = 10;
break;                            movEle = 1;
break;                            break;
break;                            break;
}
}
}
}

```

Figura 5.9 Extractos del código realizado para identificar el número de pasos requeridos.

5.3.2.2. Cálculo de pasos.

El proceso para realizar el cálculo de la cantidad de pasos que debe realizar el motor para ubicarse en la posición deseada consiste en ir verificando la ubicación anterior y así no tener riesgo de mal ubicarse.

El primer paso es verificar si el funcionamiento es en modo normal o está detenido o anclado. Si es alguno de estos dos últimos funcionamientos, se gira la instrucción de no moverse, caso contrario, se inicia el posicionamiento de cada uno de los motores.

Del proceso de identificación del paso (sección anterior) se obtiene una variable con la cantidad de pasos requeridos. Esta variable pasa a ser el *Paso_futuro* en este proceso.

Para el caso de azimut, se verifica si la cantidad de pasos es igual a cero, de ser así, el sistema no se mueve y se regresa al inicio a esperar un nuevo ángulo. Para elevación esto no se implementa puesto que el paso cero regresa al motor al ángulo cero.

El siguiente paso se encarga de ver si la nueva posición deseada es un valor de ángulo mayor o menor al actual. Para azimut, si es un valor mayor, el motor simplemente se mueve la cantidad de pasos faltantes desde donde esta para llegar al destino ($Paso_{fut}-Paso_{act}$); si es menor se determina la cantidad de pasos faltantes ($PasoA_A$) para llegar a 360° ($Paso_{Final}$) desde la ubicación actual, y posteriormente este valor de $PasoA_A$ se agrega al número de pasos que se requieren para llegar a la posición deseada ($PasoA_A+Paso_{fut}$). Finalmente se actualiza el valor de la ubicación actual con el valor ingresado previamente ($Paso_{act}=Paso_{fut}$).

Se implementó la opción para determinar que si la cantidad de pasos a dar es mayor a 110 hacia adelante, se invierte el sentido de giro para que el motor recorra la menor distancia posible, de la misma manera, si la cantidad de pasos es mayor mientras está activado el sentido de giro invertido, este se desactiva para que el motor se mueva de manera horaria para recorrer la menor cantidad de pasos posible. Se estipuló que fuese la mitad más dos de la cantidad total de pasos, es decir 110 pasos.

En la figura 5.10 se observa este diagrama de flujo.

Para el motor de elevación, el proceso es igual excepto al momento en que la posición deseada sea menor que la actual. Al entrar a este subproceso, se activa la bandera para indicar al sistema que se debe invertir el sentido de giro del motor (*Paso_atras*). Posteriormente, se determina la cantidad de pasos faltantes restando el valor actual menos el valor deseado ($Paso_{act} - Paso_{fut}$) y nuevamente se actualiza el valor actual tomando el valor de la ubicación deseada previamente ($Paso_{act} = Paso_{fut}$).

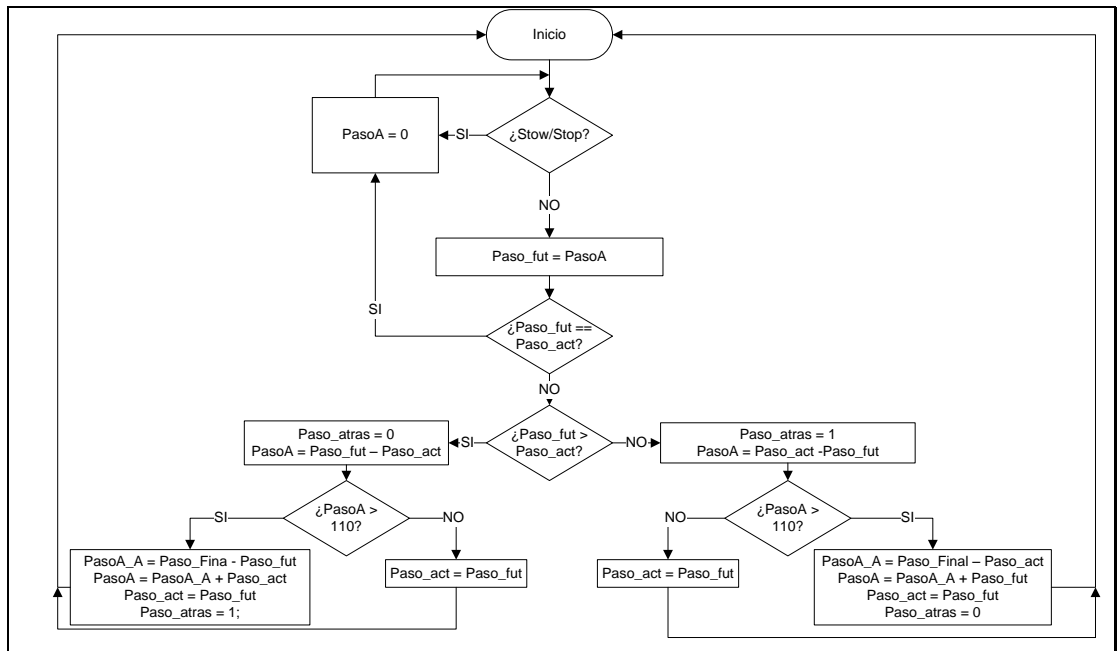


Figura 5.10 Diagrama de flujo para el proceso de cálculo de pasos para azimut.

En la figura 5.11 se puede observar el diagrama de flujo para el cálculo del posicionamiento del motor de elevación.

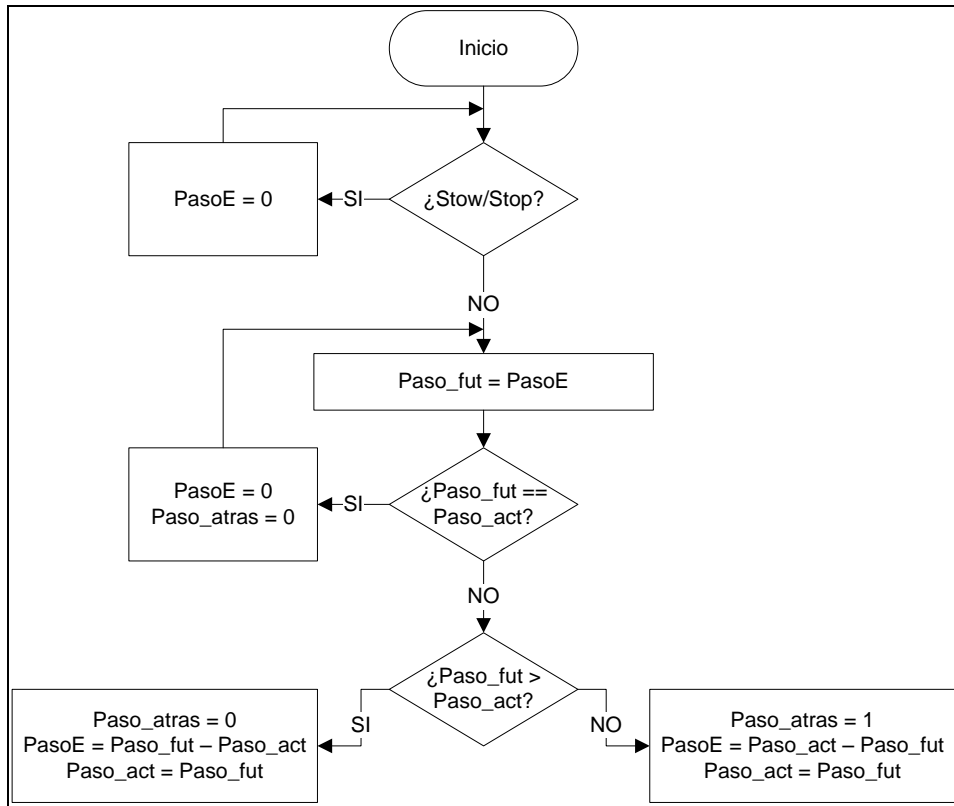


Figura 5.11 Diagrama de flujo para el proceso de cálculo de pasos para elevación.

La figura 5.12 muestra el código realizado para los procesos de cálculo del posicionamiento de los motores azimut (izquierda) y elevación (derecha).

```

////////////////////////////////////
static void CalculoAZI(void){
    if(LATG1 == 0 || LATG0 == 1){
        p_futuroAzi = pasoAzi;
        if(pasoAzi == 0) movAzi=0;
        if(p_futuroAzi == p_actualAzi){
            pasosAzi = 0;
        }
        if(p_futuroAzi<p_actualAzi){
            pasosAAzi = pasoFinalAzi - p_actualAzi;
            pasosAzi = pasosAAzi + p_futuroAzi;
            p_actualAzi = p_futuroAzi;
        }
        if(p_futuroAzi>p_actualAzi){
            pasosAzi = p_futuroAzi - p_actualAzi;
            p_actualAzi = p_futuroAzi;
        }
    }
    else {
        pasosAzi = 0;
    }
}

static void CalculoELE(void){
    if(LATG1 == 0 || LATG0 == 1){
        p_futuroEle = pasoEle;
        if(p_futuroEle == p_actualEle){
            pasosEle = 0;
            paso_atras = 0;
        }
        if(p_futuroEle<p_actualEle){
            paso_atras = 1;
            pasosEle = p_actualEle - p_futuroEle;
            p_actualEle = p_futuroEle;
        }
        if(p_futuroEle>p_actualEle){
            pasosEle = p_futuroEle - p_actualEle;
            p_actualEle = p_futuroEle;
            paso_atras = 0;
        }
    }
    else {
        pasosEle = 0;
    }
}
}

```

Figura 5.12 Extractos del código realizado para el cálculo del número de pasos requeridos.

5.3.2.3. Lógica de bits para posicionamiento.

La última fase para el posicionamiento de los motores corresponde al envío de la lógica de bits correcta para avanzar o retroceder el motor al lugar indicado.

En la figura 5.13 se observa el diagrama de flujo del proceso para el motor de elevación.

El proceso inicia verificando del número de pasos requeridos que fueron calculados previamente y seguido se inicializa el contador general en cero, esto es igual para ambos motores; a partir de este punto se explicará la rutina para cada uno de los motores de manera independiente.

Primero el motor de elevación. Cabe recordar que este motor posee la capacidad de invertir su sentido de giro, moviéndose de 0° a 180° y viceversa. Una vez identificada la cantidad de pasos a realizar, se verifica si la bandera de paso atrás esta activada; si es positivo se empieza a verificar el puerto de salida correspondiente siguiendo una lógica inversa.

Se cambia el puerto de salida en congruencia con la posición a la que debe de llegar y se aumenta el contador. Este proceso se continúa repitiendo verificando el puerto con diferentes valores (0110, 0101, 1001, 1010, en este orden) hasta que el contador iguale el número de pasos requeridos. Una vez igualado, se termina el proceso de posicionamiento. Si por el contrario, no se requiere la inversión de giro, se realiza el mismo proceso anteriormente descrito, pero se verifica el valor del puerto con un orden diferente, lo que representa que el motor se moverá en sentido horario (0110, 1010, 1001, 0101).

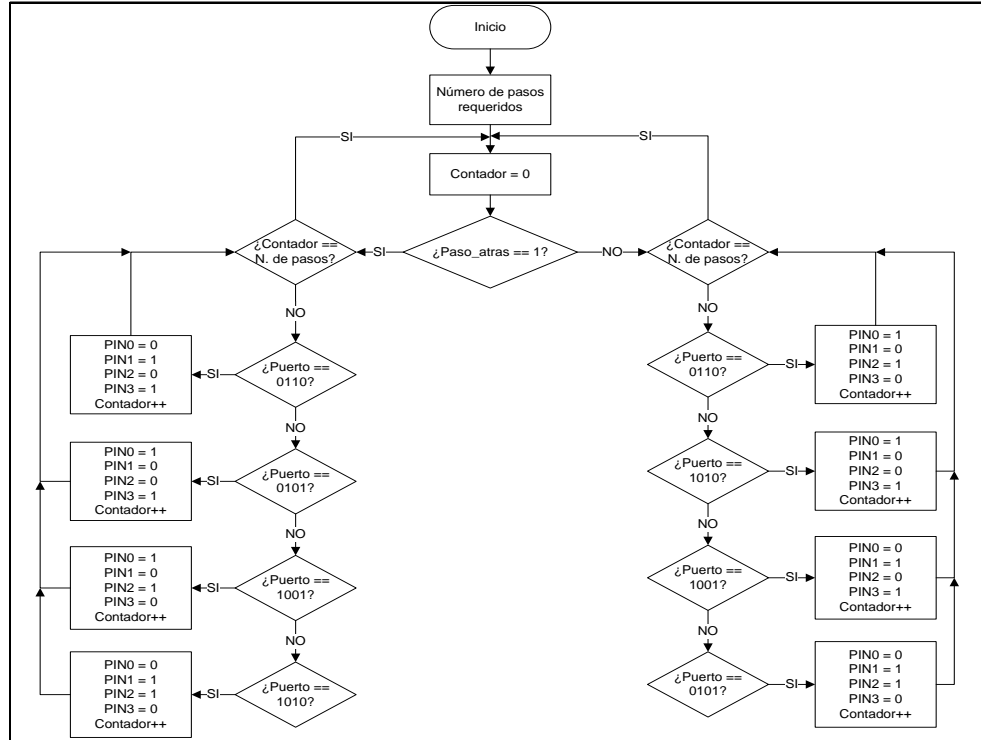


Figura 5.13 Diagrama de flujo para posicionar el motor de elevación.

Para el motor azimut se verifica también el estado del puerto correspondiente de salida que está asociado a este motor, sin embargo, para azimut se aplica como configuración de pasos el tipo *Wave Drive*, que solo activa una de las bobinas a la vez.

En la figura 5.14 se muestra el diagrama de flujo correspondiente para este proceso.

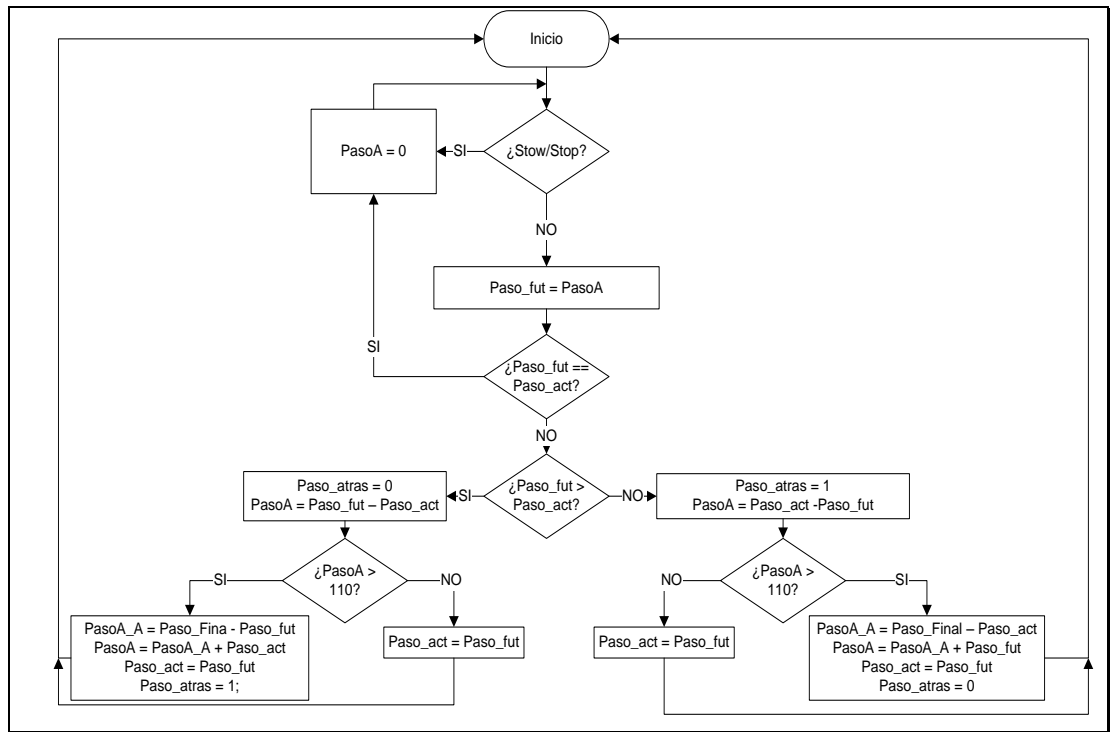


Figura 5.14 Diagrama de flujo para posicionar el motor azimut.

Por lo tanto, la verificación que se requiere del puerto debe ser secuencial buscando que bit está activado (1000, 0100, 0010, 0001; LSB a MSB).

Encontrado el bit activo, se activa la salida correspondiente, que por ser secuencial, sería la que le sigue al bit actual, es decir, si la secuencia encontrada fue 1000, la activada en la salida sería 0100 y así sucesivamente. Activada la salida, se aumenta el contador y nuevamente esto se repite la cantidad de veces necesaria hasta que el contador iguale al número de pasos correspondientes y se termine entonces el posicionamiento a la espera de una nueva ubicación.

5.3.3. Sitio web.

El sitio web del proyecto es la que permite la interfaz del usuario con el sistema electrónico, brindando un acceso controlado y rápido, y permitiéndole a este realizar los desplazamientos de la antena, anclarla o detenerla y verificar la base de datos.

El sitio ofrece la facilidad de agregar y eliminar usuarios con acceso al control, posicionar la antena, modificar su funcionamiento de modo normal a modo detenido o anclado, visitar la base datos y buscar por fecha de posicionamiento de la antena en la base de datos.

El diagrama de la figura 5.15 muestra el diagrama general del sitio web.

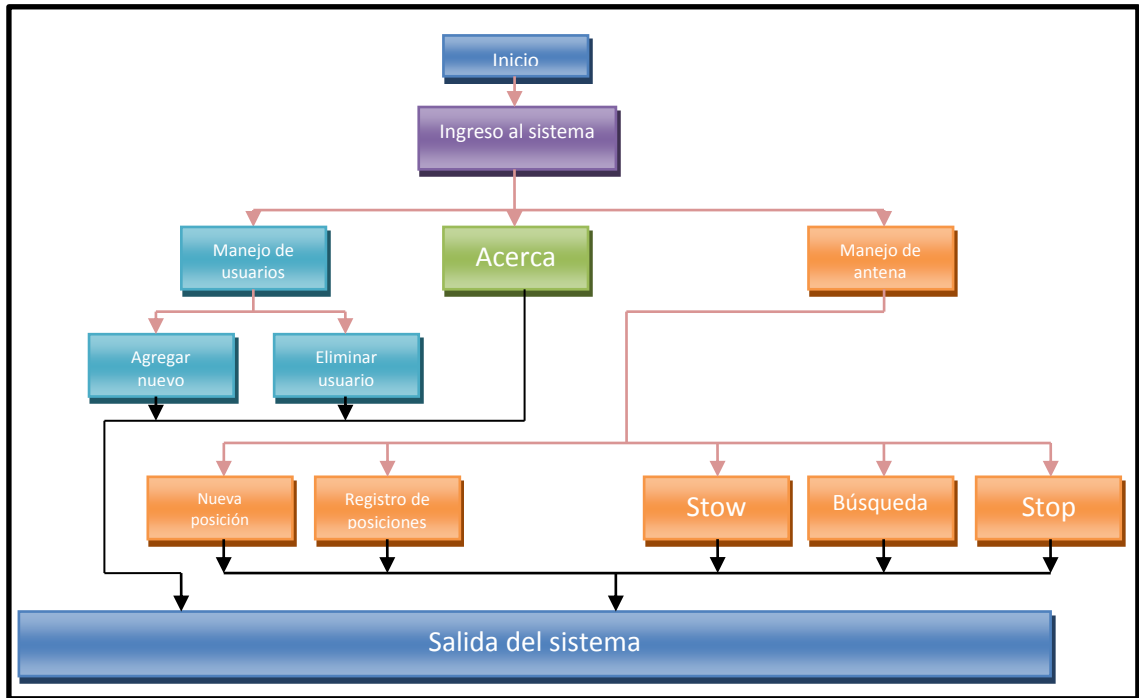


Figura 5.15 Diagrama general del sitio web.

5.3.4. Base de datos.

La base de datos del proyecto fue creada en ORACLE® (figura 5.17) y consta de tres tablas, dos de ellas interconectadas. La primera de ellas es la tabla del perfil, donde se define si los usuarios son visitantes, usuario o administrador. La segunda tabla es la de usuarios, en donde se define qué tipo de perfil tiene, nombre, correo electrónico y cédula. La tercera y última tabla es la de eventos, que es la que registra los ángulos ingresados, la fecha y la hora.

La base de datos se alojó dentro del servidor de la empresa, reservando un espacio en ella para este proyecto.

En la figura 5.16 se muestran estas tablas y la forma en que están interconectadas la tabla de perfil y la de usuario.

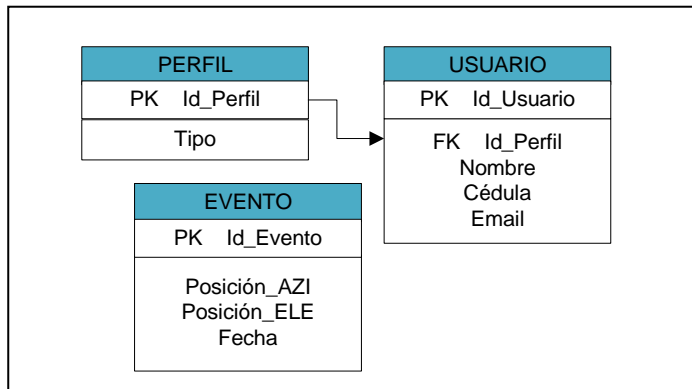


Figura 5.16 Tablas creadas para la base de datos.

El manejo de estas tablas estuvo enlazado con la plataforma NetBeans® mediante el driver JDBC de ORACLE®. En las siguientes figuras (5.18, 5.19) se muestran imágenes de las tablas en NetBeans®.

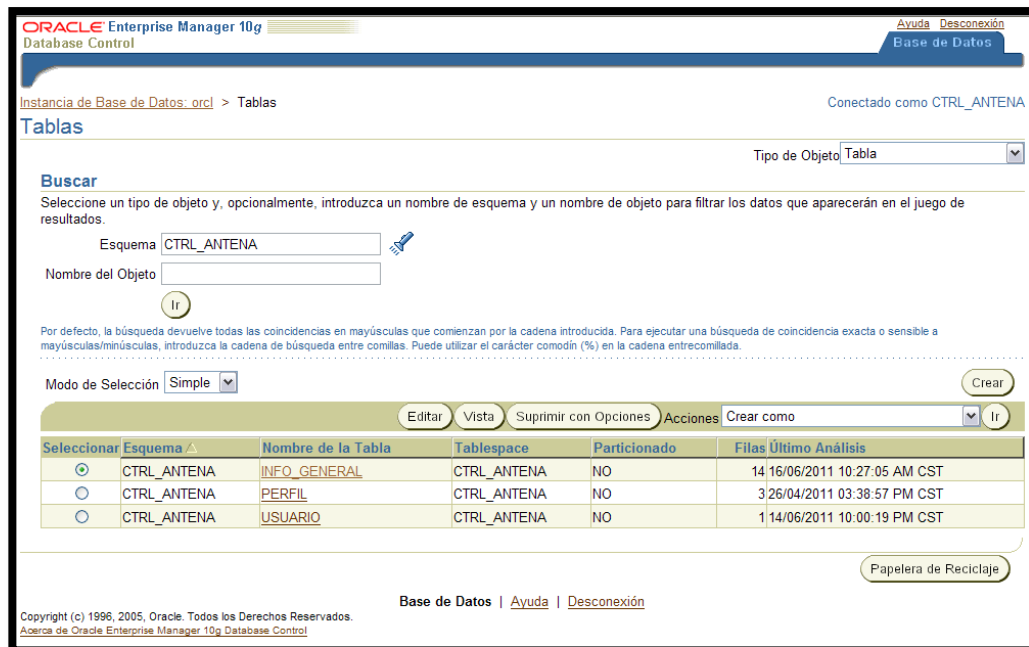


Figura 5.17 Base de datos creada en ORACLE®.

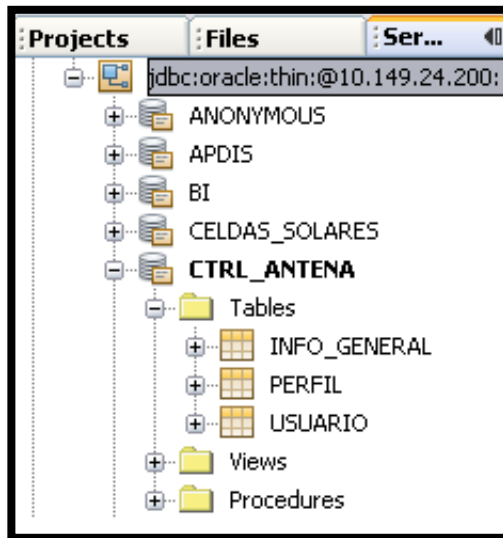


Figura 5.18 Tablas empleadas.

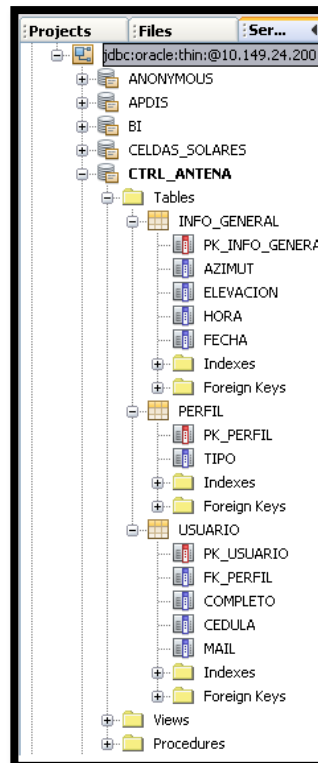


Figura 5.19 Tablas empleadas extendidas.

Capítulo 6 : Análisis de Resultados.

El siguiente capítulo muestra los resultados obtenidos así como un análisis de estos, de manera que los objetivos planteados al inicio pueden ser corroborados y determinar que han sido cumplidos a cabalidad.

Se expondrán primero los resultados y posteriormente se dará el análisis de estos. El capítulo se subdivide en las dos partes fundamentales del proyecto, y estos a su vez en los bloques necesarios para su solución.

6.1. Control automático.

Para elaborar el nuevo control automático del sistema primero se procedió a realizar el análisis de la planta en cuestión, es decir, la antena satelital. Ante la imposibilidad de detener el sistema para realizar las pruebas rutinarias y así determinar los parámetros necesarios para el modelado matemático se procedió entonces a la búsqueda de datos relevantes del sistema mediante documentos, manuales, hojas de datos, etc., disponibles en la empresa.

La información más relevante encontrada en esta documentación fue el peso de la antena satelital, la longitud de su radio, los datos principales de placa de los motores y la ganancia del control automático actualmente en funcionamiento. El tren de engranes presente en el motor de elevación se debió estimar debido su gran dimensión, superior a los cuatro metros, y el de azimut se midió para poder determinar así la relación de radios correspondiente a cada uno.

A partir de estos datos, y con la ayuda de las referencias bibliográficas¹³ en donde se detalla un ejemplo de un motor AC de inducción ensamblado a un tren de engranes se obtuvo el modelo matemático de la planta, la antena satelital, los motores que la mueven y los engranes de estos.

En la siguiente figura se observa un diagrama del sistema al que se le obtuvo el modelo matemático.

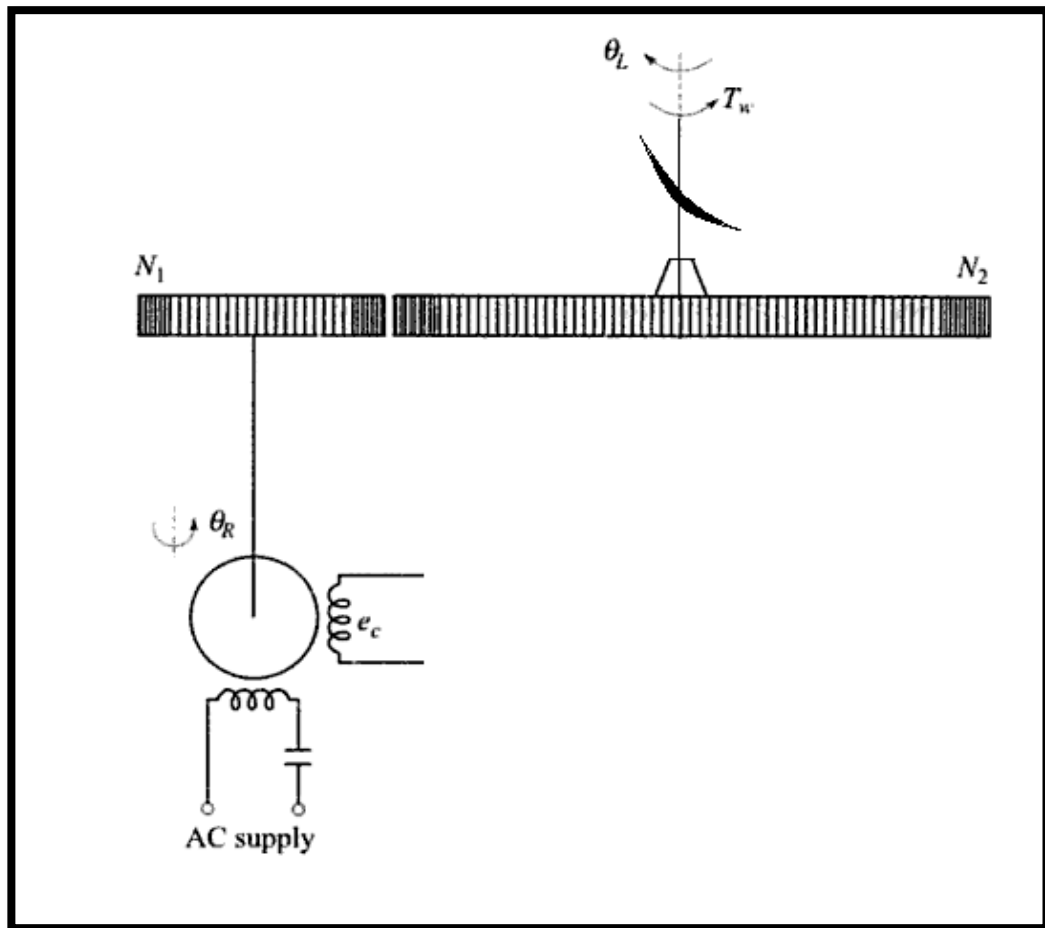


Figura 6.1 Diagrama esquemático de la planta. [23]

¹³ Referencia tomada de [23]

Este diagrama se reconfiguró como un diagrama de bloques para lograr obtener la función de transferencia del sistema. A continuación se muestra el diagrama de bloques correspondiente, el cual cabe mencionar que fue el mismo para ambos motores, con la salvedad de que los valores de los radios de los engranes fueron diferentes, variando así el parámetro N del diagrama. El sistema se manejó como dos SISO, tal y como se explica en la literatura puesto que el movimiento no depende uno de otro.

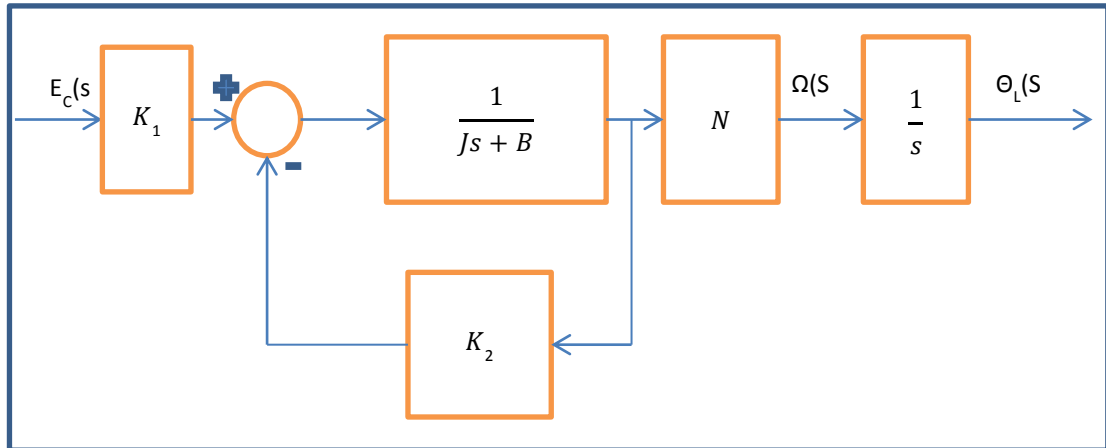


Figura 6.2 Diagrama de bloques del sistema.

En la siguiente tabla se detallan algunas de las características del sistema que, como se mencionó antes, fueron obtenidos de la documentación correspondiente.

Tabla 3 Características físicas del sistema.

Característica	Valor
J (Kg*m ²)	1.23x10 ⁵
B (N*m)	22.9x10 ³
N _{AZI}	0.053
N _{ELE}	0.59
K ₁	5000
K ₂	0.21

El valor de la inercia del sistema fue estimado mediante la fórmula para la semiesfera hueca.

$$\frac{1}{3} * M * r^2 \quad (6.1)$$

El valor del coeficiente de fricción B fue tomado de los datos de placa de los motores del sistema. La antena rueda sobre rieles engrasados.

Los valores N_{AZI} y N_{ELE} fueron estimados mediante la medición de los engranes que mueven la antena satelital tanto en azimut como en elevación.

Finalmente, las ganancias K_1 y K_2 fueron tomadas del manual de los drivers de los motores que poseen un control de compensación y tenía ajustados los valores utilizados.

Así pues, se procedió a insertar los valores de la tabla 3 en el diagrama de bloques del sistema de la figura 6.2 para obtener las dos funciones de transferencia, correspondiente para cada uno de los motores.

Para el motor azimut, la función de transferencia fue:

$$\frac{0.0239}{s^2 + 0.1287 s} \quad (6.2)$$

Para el motor de elevación, la función de transferencia fue:

$$\frac{0.002153}{s^2 + 0.1287 s} \quad (6.3)$$

De esta forma, se cumplió con el objetivo número uno del proyecto.

En la tabla 4 se detallan los valores observados de las gráficas. A continuación se enumeran. Las gráficas de la respuesta al escalón de las funciones de transferencias se presentan más adelante junto con los sistemas compensados para comparaciones.

Tabla 4 Valores obtenidos para ambos motores.

Parámetro	Azimut	Elevación
Tiempo de subida 2% (s)	9	113
Tiempo de estabilización (s)	69	206
Sobreimpulso (%)	28	0

Con los valores de las tablas 4 y 5 se escogieron los valores para mejorar el sistema mediante el control automático y a partir de estos desarrollar el o los compensadores necesarios. En la siguiente tabla se detallan dichos valores para ambos controles.

Tabla 5 Nuevos parámetros para el diseño de los compensadores.

Parámetro	Azimut	Elevación
Tiempo de estabilización (s)	55	120
Sobreimpulso (%)	5	1
Error de estado estacionario (%)	0	0

Con la herramienta SISOTOOL® de MATLAB® se diseñaron los compensadores para cada uno de los motores. Para el motor de elevación se diseñó un compensador de adelanto para corregir el tiempo de estabilización del sistema, que pasó de 206 segundos a 106 segundos, sin embargo, el error de estado estacionario aumentó a 40%, por lo que fue necesario agregar un control integral con el que se consiguió corregir el E_{SS} a 0%.

La función del compensador de adelanto que se diseñó es la siguiente:

$$\frac{2.9659*(z-0.938)}{(z-0.909)} \quad (6.4)$$

La función del compensador de adelanto y del control integral que fueron aplicados a la planta es la siguiente:

$$\frac{0.18966*(z-0.938)}{(z-0.892)*(z-0.909)} \quad (6.5)$$

En la siguiente figura se muestran tanto el sistema sin compensar así como el sistema ya compensado para el motor de elevación.

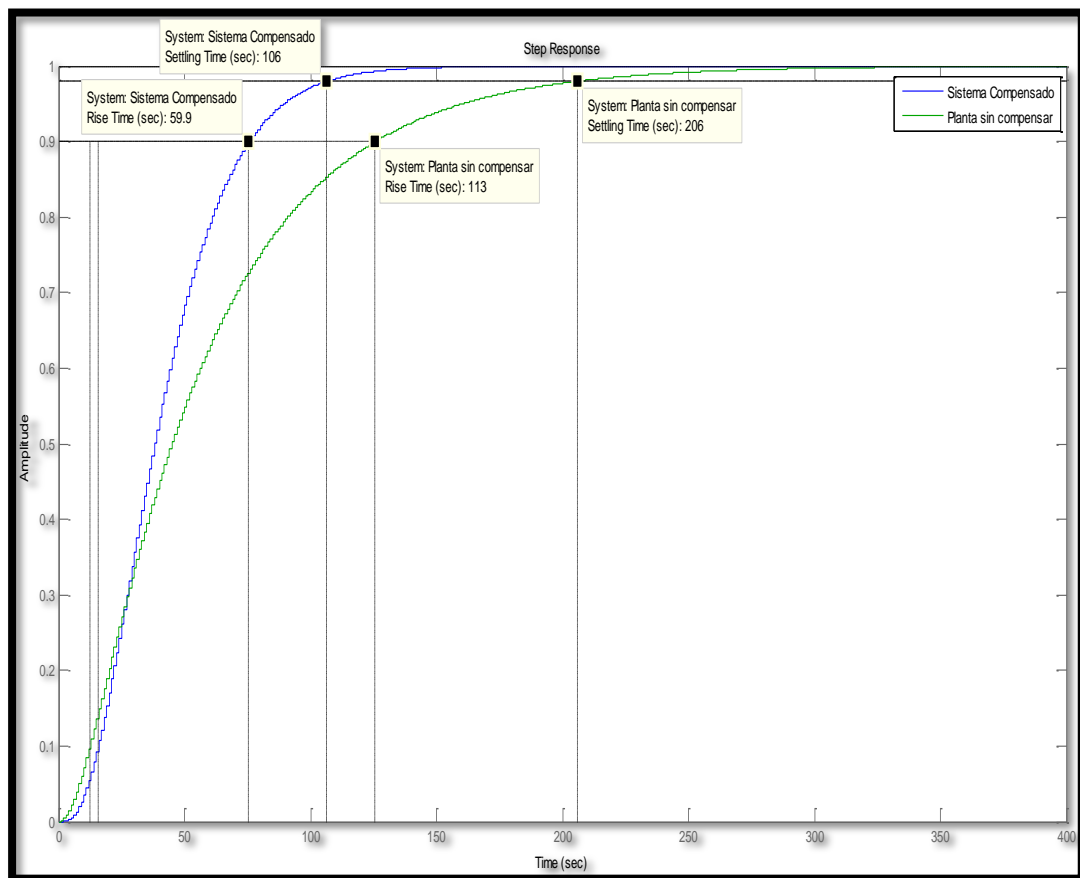


Figura 6.3 Gráfica de la respuesta al escalón para el motor de elevación.

Se debe explicar que en el denominador de la ecuación 6.5, para ser un control integral exacto debería tener un valor de $(z - 1)$, sin embargo, al utilizarse la herramienta SISOTOOL® y al haber realizado aproximaciones para estabilizar el sistema, el valor de $(z - 1)$ se aproxima al valor encontrado de $(z - 0.909)$.

Para el motor de azimut únicamente fue necesario el diseño de un compensador de adelanto para mejorar sus características de tiempo y de sobreimpulso. Así, el tiempo de estabilización se corrigió de 69 segundos a 53 segundos, y el sobreimpulso se mejoró a solamente un 4%, pasando de un 28% originalmente. Para este motor, nunca se presentó el problema de error de estado estacionario (E_{SS}).

La función del compensador de adelanto empleado en este motor es la siguiente:

$$\frac{0.81117*(z-0.79)}{(z-0.618)} \quad (6.6)$$

La gráfica del sistema para el motor azimut sin compensar y compensado ante una entrada escalón se presenta en la siguiente figura.

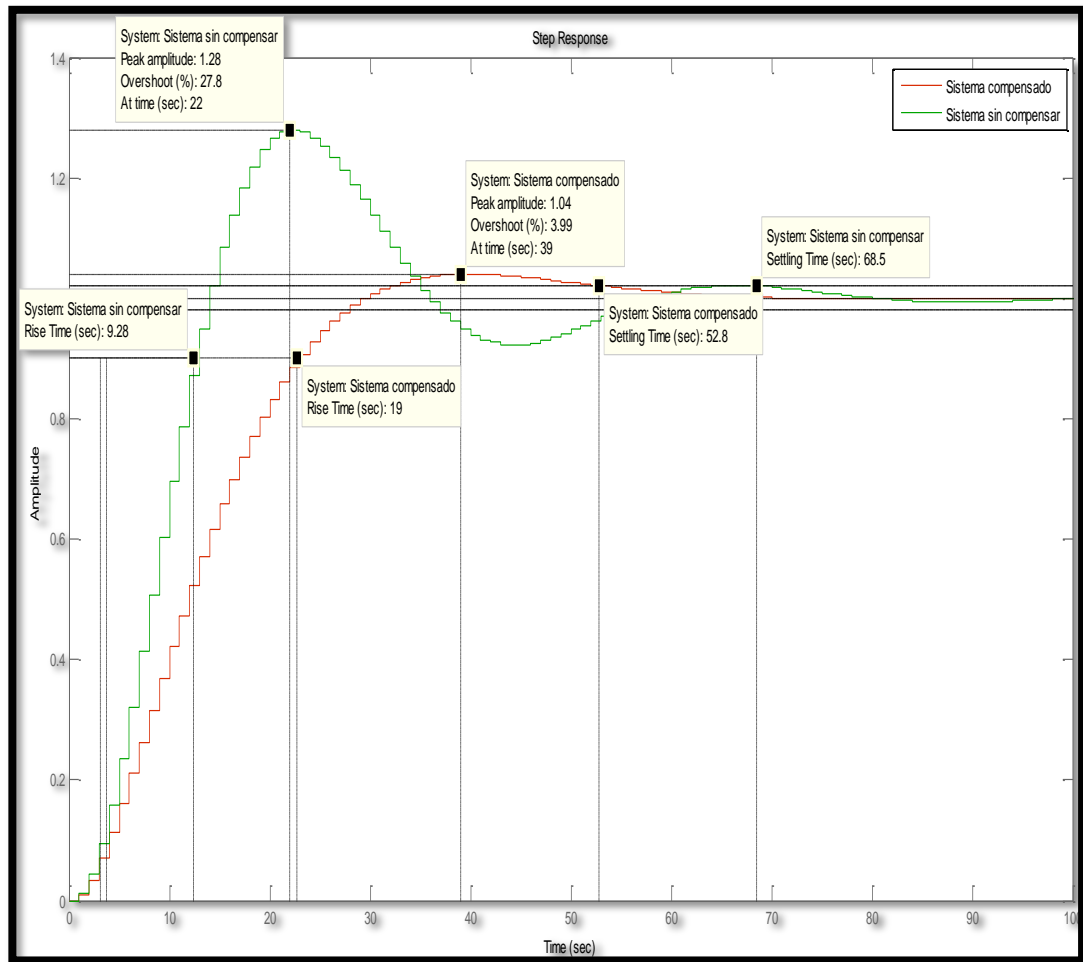


Figura 6.4 Gráfica de la respuesta al escalón para el motor azimut.

Por lo tanto de esta manera se rediseñó el nuevo sistema de control para la antena satelital, permitiendo que se mejorase en su tiempo de respuesta tanto para el eje azimutal como para el eje de elevación y se redujo el sobreimpulso presente en el motor azimut, de esta manera, se cumplió con el segundo y tercer objetivo de diseño planteados en el proyecto.

6.2. Modelo a pequeña escala del sistema.

El modelo a pequeña escala, a pesar de no estar explícito en el capítulo de objetivos, fue requerimiento necesario de implementar para comprobar el funcionamiento de los objetivos de software, como lo son el sitio o página web, la base de datos y la comunicación entre ellos, por lo tanto, se decidió manejar dos motores paso a paso, los cuales fueron escogidos por su gran precisión, para simular el movimiento de la antena bajo el funcionamiento del sitio web; esto debido a que por estar en continuo funcionamiento el sistema actual no existía la posibilidad de detenerlo y acoplarlo al sistema prototipo.

A continuación se detallan cada una de las partes requeridas para su funcionamiento.

6.2.1. Página web.

Se desarrolló mediante la plataforma NetBeans® en el lenguaje JAVA Struts®. La página en sí consta de varias secciones, a continuación se detalla acerca de cada una de ellas.

6.2.1.1. Inicio de sesión.

En la página de inicio de sesión el usuario requiere de ingresar su nombre de usuario y contraseña del sistema del ICE para poder ingresar a las demás opciones presentes en el sitio web. En caso de no ser usuario, se le presenta un mensaje de error.

A continuación se muestra la imagen correspondiente.

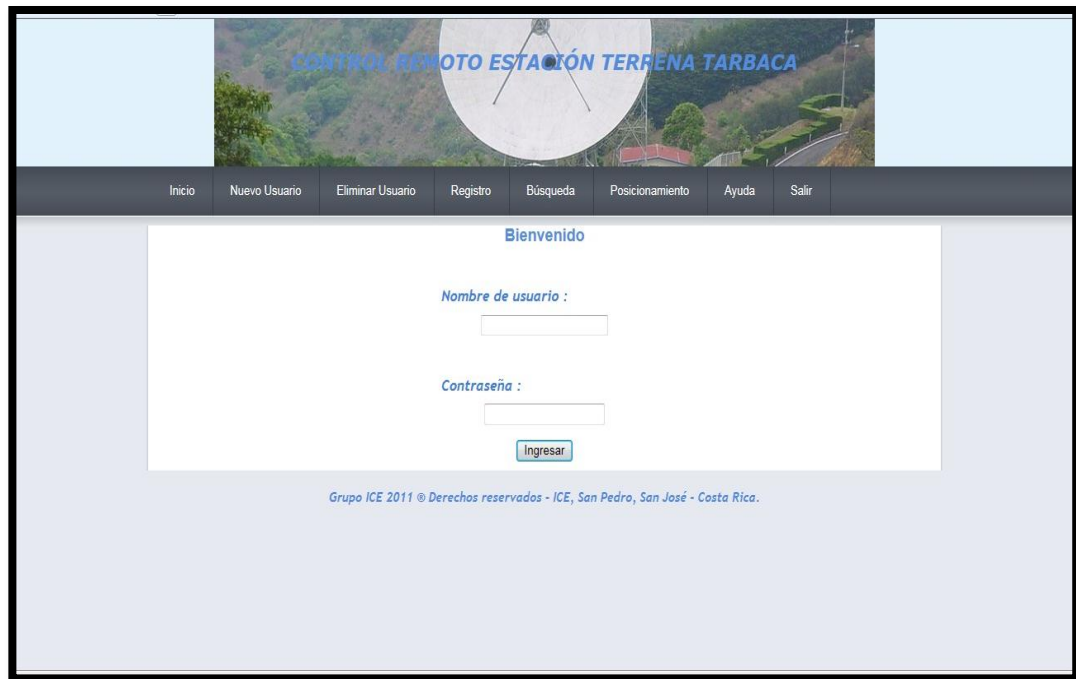


Figura 6.5 Página de inicio del sitio web del proyecto.

Una vez identificado el usuario este es bienvenido y se le hace una corta explicación acerca del proyecto en desarrollo. A continuación se muestra una imagen.

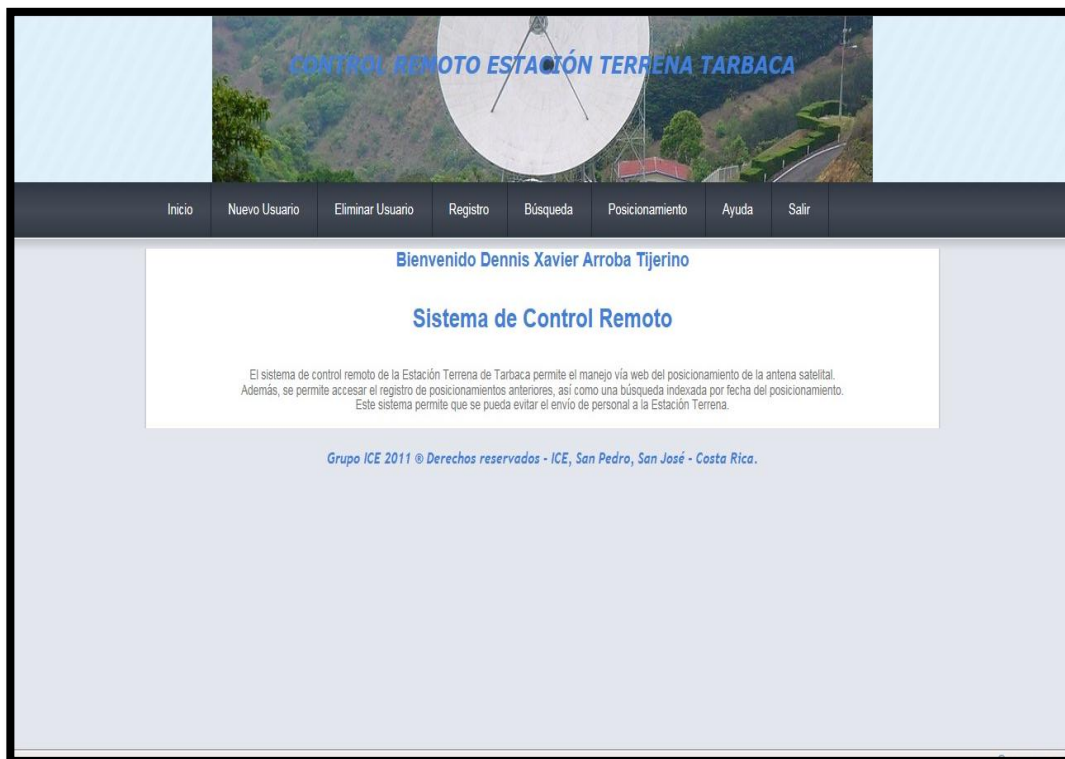


Figura 6.6 Página de inicio de sesión una vez iniciada exitosamente.

6.2.1.2. Agregar/Eliminar Usuario.

Las opciones de agregar y eliminar usuario fueron implementadas en dos páginas diferentes. Para agregar usuario (figura 6.7) se debe indicar nombre completo, correo electrónico, número de cédula y que tipo de perfil posee el usuario. El perfil puede ser visitante, usuario y administrador, y dependiendo del nivel de perfil el usuario puede tener habilitada ninguna, algunas o todas las funciones del sitio.

Inicio Nuevo Usuario Eliminar Usuario Registro Búsqueda Posicionamiento Ayuda Salir

Bienvenido Dennis Xavier Arroba Tijerino

INTRODUZCA LOS PARÁMETROS

Nombre:	Cédula:
<input type="text"/>	<input type="text"/>
Email:	*Perfil:
<input type="text"/>	<input type="text"/>

1:administrador
2:usuario
3:visitante

Nueva Usuario

Grupo ICE 2011 © Derechos reservados - ICE, San Pedro, San José - Costa Rica.

Figura 6.7 Página para agregar usuarios del sistema.

Para eliminar algún usuario del sistema se requiere que se ingrese el número de cédula de éste. Una vez eliminado, el usuario queda fuera de la base de datos.



Figura 6.8 Página para eliminar usuarios del sistema.

6.2.1.3. Registro histórico.

El registro histórico permite visualizar todos los posicionamientos almacenados en la base de datos del sistema donde se indica fecha, hora y ambos ángulos de posicionamiento realizados. Como opción extra, los datos de la base de datos se pueden exportar desde la página web a un archivo de Excel, PDF, XML y CSV (*Comma Separated Values*). Más detalles en la sección de anexos.

6.2.1.4. Búsqueda en el registro.

El sitio web también permite la búsqueda indexada en el registro con la fecha de posicionamiento como referencia. El usuario solo debe ingresar la fecha (en este formato: 22/4/2011) y seleccionar la opción de búsqueda por fecha para que se le presenten los resultados en una tabla.

A continuación en la figura 6.9 se muestra la imagen de la página donde se debe ingresar la fecha que se desea buscar.

CONTROL REMOTO ESTACIÓN TERRENA TARBACA

Inicio Nuevo Usuario Eliminar Usuario Registro Búsqueda Posicionamiento Ayuda Salir

Bienvenido Dennis Xavier Arroba Tijerino

SELECCIONAR PARÁMETRO

FECHA DE POSICIONAMIENTO

Introducir dato: 24/5/2011

Consulta

Para fecha usar el formato dd/m/aaaa

Figura 6.9 Página de búsqueda en el registro.

En la figura 6.10 se muestra el resultado de la búsqueda de la figura anterior.

CONTROL REMOTO ESTACION TERRENA TARBACA

Inicio Nuevo Usuario Eliminar Usuario Registro Búsqueda Posicionamiento Ayuda Salir

Bienvenido Dennis Xavier Arroba Tijerino

9 items found, displaying 1 to 3.[FirstPrev] 1, 2, 3 [NextLast]

°AZIMUT	°ELEVACIÓN	HORA	FECHA
90	180	11:15	24/5/2011
180	180	11:30	24/5/2011
0	180	11:32	24/5/2011

Export options: CSV | Excel | XML | PDF

Grupo ICE 2011 © Derechos reservados - ICE, San Pedro, San José - Costa Rica.

Figura 6.10 Página de búsqueda mostrando resultados.

6.2.1.5. Posicionamiento.

La página para controlar la ubicación de la antena permite posicionar la antena mediante el ingreso de los ángulos azimut y de elevación y también tiene la capacidad de anclar la antena en una posición predeterminada o simplemente detener la antena para evitar el ingreso de algún otro ángulo.

Al ingresar un nuevo set de ángulos o anclar/detener la antena, la página web se detiene y no permite ingresar a ninguna otra función hasta que los motores terminen de posicionarse, tal y como se muestra en la siguiente figura.



Figura 6.11 Página web mientras se posicionan los motores.

6.2.1.5.1. Ángulos de posición.

En esta sección es donde se ingresan los dos valores de los ángulos deseados al que se desea mover la antena. Se puede ingresar uno o los dos valores.



Figura 6.12 Página web para posicionar la antena.

6.2.1.5.2. Anclaje/Parada.

La opción de anclaje y parada permite detener la antena en la posición predeterminada y en la posición en la que se encuentre actualmente, respectivamente. Se detalla más en el manual de usuario en la sección de apéndices.

6.2.2. Envío y recepción de datos UDP.

El envío de los datos ingresados en la página web, es decir los ángulos de posicionamiento, se realizó desde el kit de desarrollo, tal y como se explicó en el capítulo anterior, mientras que la recepción de los mismos se realizó en una aplicación JAVA® desarrollada en la plataforma NetBeans®.

El sistema se encuentra oyendo siempre el puerto determinado para aceptar los datos una vez que lleguen. En la figura 6.13 se muestra la imagen del servidor oyendo el puerto.

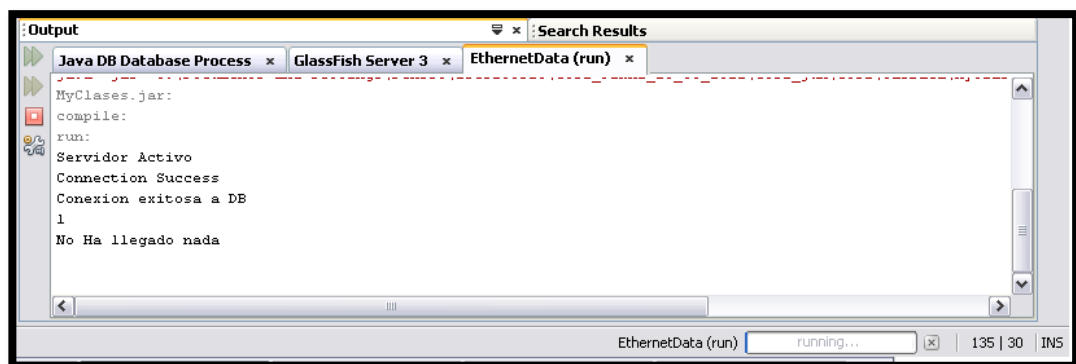


Figura 6.13 Aplicación oyendo al puerto.

Una vez que llegaron los datos, el sistema determina de donde provienen y la hora en que se atiende. A partir de este momento entra en escena la conexión a la base de datos que se explica en la sección siguiente.

En la figura 6.14 se muestra la imagen del servidor atendiendo la llegada de datos al puerto de escucha.

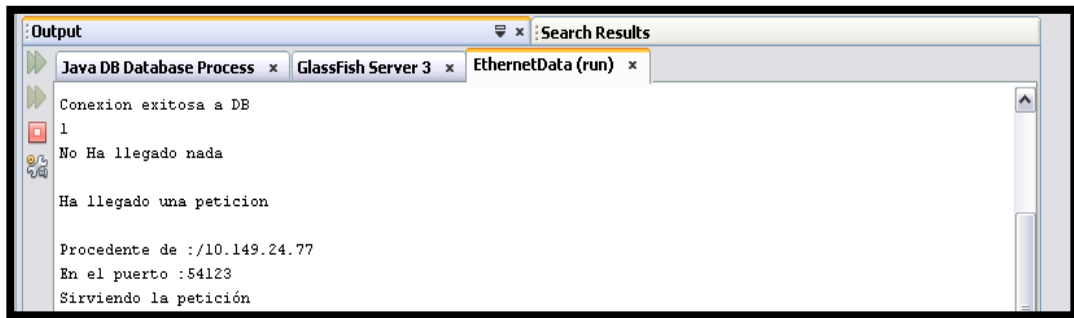


Figura 6.14 Aplicación atendiendo los datos del puerto.

6.2.3. Base de datos.

Una vez que los datos UDP son captados por la aplicación JAVA®, se realizó la conexión a la base de datos. Posteriormente se presentan en la consola los datos decodificados provenientes del paquete UDP.

El penúltimo paso es insertar los datos ya correctamente en la base de datos, incluyendo la hora y fecha del posicionamiento. Finalmente, cuando terminó de ingresar los datos, la aplicación se desconecta de la base de datos.

En la siguiente figura se muestra el proceso anteriormente descrito.

```
Output
Java DB Database Process x GlassFish Server 3 x EthernetData (run) x
Conexion exitosa a DB (1)
1
No Ha llegado nada
Ha llegado una peticion
Procedente de :/10.149.24.77
En el puerto :54123
Sirviendo la peticion
1:0
2:0180; (2)
1:0180;180
HORA DEL SERVIDOR Wed Jun 29 08:13:18 CST 2011
180 (3)
180
29/5/2011
8:13
1
Datos ingresados correctamente
Conexion terminada (4)
Connection Success
Conexion exitosa a DB
1
No Ha llegado nada
Ha llegado una peticion
Procedente de :/10.149.24.77
En el puerto :54123
```

Figura 6.15 Conexión a la base de datos.

En (1) se conectó la aplicación a la base de datos.

En (2) se detallan los valores decodificados provenientes de los datos ingresados en la página web.

En (3) se ingresan los valores y la fecha y hora a la base de datos. Finalmente en (4) se termina la conexión.

6.2.4. Control sobre los motores.

Los motores fueron manejados por el circuito integrado ULN2003 que corresponde a un empaquetado de siete transistores Darlington con diodos de rodamiento libre. La señal de control, proveniente del kit de desarrollo entró al CI antes mencionado y la salida de este fue dirigida a cada una de las entradas de control de los motores.

El uso de los transistores permitió un mayor manejo de corriente para los motores, la cual el kit de desarrollo no fue capaz de proveer por sí solo. A las salidas de cada uno de los CI se agregó una resistencia de Pull-Down para bajar a cero lógico los valores flotantes y evitar que el motor tuviera un comportamiento no deseado.

6.3. Funcionamiento del sistema actual de control.

A continuación se detalla la forma de funcionamiento del sistema que actualmente opera.

6.3.1. Funcionamiento.

Al encender el equipo, este toma varios minutos para poder inicializar el sistema operativo (propio de la empresa VertexRSI®, escrito en C y utiliza QNX como sistema operativo.). Una vez que este entra en funcionamiento, la pantalla de inicio es la que aparece en la siguiente figura. En esta se puede observar las diferentes partes o submenús que lo conforman.

Estas son:

- Fuerza de la señal recibida
- Modo de operación.
- Posición Elevación y Azimut.
- Estado del sistema.
- Menú de funciones.
- Mensajes de falla.



Figura 6.16 Pantalla de inicio del sistema actual de control.

Tomado de manual provisto por la empresa.

A través del menú de funciones es que se puede navegar por el sistema. Puesto que el sistema es un poco antiguo, este presenta cierta dificultad para moverse entre ventanas. Por ejemplo, para moverse a la ventana superior, se debe apretar Alt + flecha de arriba, para regresar al menú se hace con Alt + tecla Home, para la ventana izquierda es Alt + flecha izquierda y para la derecha Alt + flecha derecha.

6.3.2. Opciones del sistema.

Las principales opciones del sistema son las siguientes:

- **Modo Stop:** el sistema deshabilita los ejes de control de los motores, fijando los frenos. La antena se desacelera, y al alcanzar el 10% de la velocidad, se aplican los frenos.
- **Posicionamiento:** en esta opción se ajustan los valores de los ángulos de elevación y de azimut, así como la longitud y latitud del sistema. El sistema de control revisa que el set de datos ingresados esté dentro del rango de operación.
- **Preset:** acá se encuentran almacenadas hasta 40 posiciones de ángulos para ser escogidas y utilizadas por el control para posicionar la antena.
- **Optrack:** es la opción que permite el rastreo de la señal RF proveniente de los satélites que permite a la antena ubicarse al encontrar la señal más fuerte.
- **INTELSAT:** esta opción permite la selección de los datos de posicionamiento mediante los datos provistos por INTELSAT. INTELSAT es una empresa que posee una red de 52 satélites comerciales, la flota más grande del planeta.

- **Posicionamiento manual:** esta opción lo que permite es que se acceda a la consola manual de posicionamiento, que asemeja un potenciómetro que permite ubicar a la antena en la posición deseada.
- **Anclaje (Stow):** la opción de anclaje permite llevar a la antena a una posición segura. Esta opción es utilizada generalmente bajo condiciones ambientales con fuertes vientos para proteger a la antena de sufrir algún daño.
- **Control externo:** esta opción permite que se maneje la antena desde otra computadora conectada al sistema de control principal mediante RS-232.

6.3.3. Unidad de Control de la Antena (ACU).

La unidad de control consta de un panel frontal, en donde se encuentra ubicada la parada de emergencia, la pantalla LCD donde se navega por los menús. Cuenta además como sistema de almacenamiento una unidad de disquete de 1.44MB y un disco duro de 16MB donde reside el sistema operativo. La tarjeta madre del sistema permite la conexión del teclado necesario para el manejo del sistema operativo.

La ACU está conectada con las otras partes del sistema mediante una tarjeta especial. Esta tarjeta consta de cuatro puertos seriales, dos de ellos dedicados para el envío de los datos para el manejo de los motores. Además, permite la conexión del sistema de búsqueda de RF y con la Unidad Central de Control (CCU).

Posee además un convertidor analógico-digital (ADC) de 14 bits para el monitoreo de la señal RF. Como salida de este sistema está la conexión al sistema de relés.

6.3.4. Sistema de rastreo de señal RF.

El receptor de la señal RF convierte esta en una señal CD. La entrada es una señal en la frecuencia de bajada (Down Link). La salida es una señal CD proporcional a la calidad de la señal recibida. La conexión entre los receptores y el sistema es mediante RS-232. Las frecuencia de rastreo de RF es de entre 3.4 GHz y 4.2 GHz. La idea de este rastreo es ubicar la antena hacia el satélite mediante la localización del punto más alto de la señal de rastreo.

6.3.5. Enlace de datos.

Los protocolos soportados para el enlace de datos son RS-232 y RS-422. La configuración del puerto serial es de una velocidad de transmisión de datos de 4800 baudios, 8 bits de datos, un bit de inicio y dos de parada, con paridad impar.

Capítulo 7 : Conclusiones y recomendaciones.

En este capítulo se exponen las principales conclusiones a las que se llegó una vez concluido el proyecto de graduación, además de las recomendaciones pertinentes del caso para futuras referencias en cuanto a mejoras, expansiones o implementaciones.

7.1. Conclusiones

1. El tren de engranes presentes en la planta modifican sustancialmente el numerador en ambas funciones de transferencia de los sistemas.
2. El compensador de adelanto produce, sobre la planta, que se disminuyan los tiempos de subida y estabilización, contribuyendo a la mejora de la respuesta del sistema.
3. El compensador integral disminuye o elimina por completo el error de estado estacionario en el sistema.
4. El uso de *beans* en Struts® para la programación del sitio web permite un uso más eficiente de los recursos y al ser un modelo de programación funcional, permite detectar errores de forma más expedita.
5. El abrir y cerrar los sockets de los puertos UDP garantizan un funcionamiento óptimo de estos; mantener abierto uno de estos puede llevar a un fallo crítico del sistema.
6. En el programa NetBeans®, se implementaron el programa de recepción de datos UDP, la conexión a la base de datos ORACLE® y el sitio web.

7.2. Recomendaciones

1. Identificar cuál usuario modificó la posición de la antena al ser almacenada ésta en la base de datos.
2. Agregar un sistema de potencia capaz de acoplar el prototipo con el manejo de los motores de la antena.
3. Agregar al prototipo el rastreo de señal RF para posicionamiento de la antena.
4. Implementar el envío de paquetes mediante el protocolo TCP/IP en lugar de UDP, esto para tener mayor confiabilidad a la hora de la transmisión y recepción de los datos.
5. Implementar sensor de posicionamiento físico y no solo lógico para tener así información redundante para corroborar exitosamente la ubicación.

Bibliografía.

- [1] Gillat. [En línea]. Disponible en: <<http://www.gilatnetworks.com>>. [Última consulta 6 de mayo 2011]
- [2] Wikipedia, *Antena parabólica*. [En línea]. Disponible en: <http://es.wikipedia.org/wiki/Antena_parab%C3%B3lica>. [Última consulta 6 de mayo 2011]
- [3] Wikipedia, *Satellite dish*. [En línea]. Disponible en: <http://en.wikipedia.org/wiki/Satellite_dish>. [Última consulta 6 de mayo 2011]
- [4] Wikipedia, *Earth station*. [En línea]. Disponible en: <http://en.wikipedia.org/wiki/Earth_station>. [Última consulta 6 de mayo 2011]
- [5] Universidad Politécnica Salesiana. [En línea]. Disponible en: <<http://dspace.ups.edu.ec/bitstream/123456789/201/3/Capitulo%202.pdf>>. [Última consulta 6 de mayo 2011]
- [6] Microsoft Tech Net. [En línea]. Disponible en: <[http://technet.microsoft.com/es-es/library/cc785220\(WS.10\).aspx](http://technet.microsoft.com/es-es/library/cc785220(WS.10).aspx)>. [Última consulta 6 de mayo 2011]
- [7] Wikipedia, *UDP*. [En línea]. Disponible en: <<http://es.wikipedia.org/wiki/UDP>>. [Última consulta 6 de mayo 2011]
- [8] Protocols, *UDP*. [En línea]. Disponible en: <<http://www.protocols.com/pbook/tcpip2.htm#UDP>>. [Última consulta 6 de mayo 2011]

- [9] Modtronix, *SBC65EC*. [En línea]. Disponible en: <http://www.modtronix.com/product_info.php?products_id=149>. [Última consulta 16 de mayo 2011]
- [10] Wikipedia, *Motor asíncrono*. [En línea]. Disponible en: <http://es.wikipedia.org/wiki/Motor_as%C3%ADncrono>. [Última consulta 16 de mayo 2011]
- [11] Chapman S., ***Máquinas eléctricas***. 4ta edición. Mc Graw Hill, 2006, México. [Última consulta 16 de mayo 2011]
- [12] Wikipedia, *Lead compensator*. [En línea]. Disponible en: <http://en.wikipedia.org/wiki/Lead%E2%80%93lag_compensator>. [Última consulta 16 de mayo 2011]
- [13] Wikipedia, *Lugar de las raíces*. [En línea]. Disponible en: <http://es.wikipedia.org/wiki/Lugar_de_ra%C3%ADces>. [Última consulta 16 de mayo 2011]
- [14] Monografías. [En línea]. Disponible en: <<http://www.monografias.com>>. [Última consulta 16 de mayo 2011]
- [15] Sitio web Ing. Eduardo Interiano. ***Control automático***. [En línea]. Disponible en: <<http://ie.itcr.ac.cr/einteriano/control/>>. [Última consulta 10 de junio 2011]
- [16] Wikipedia, *Control integral*. [En línea] Disponible en: <http://es.wikipedia.org/wiki/Proporcional_integral_derivativo>. [Última consulta 10 de junio 2011]
- [17] Kuo, Benjamín C. ***Sistemas de Control Automático***. 7ma edición. Prentice Hall, 1996, México. [Última consulta 10 de junio 2011]

- [18] Ogata, Katsuhiko. **Ingeniería de Control Moderna**. 4ta edición. Pearson Education, 2003, Madrid. [Última consulta 10 de junio 2011]
- [19] Sitio web Ing. Eduardo Interiano. **Análisis de sistemas**. [En línea]. Disponible en: <<http://ie.itcr.ac.cr/einteriano/analisis/>>. [Última consulta 10 de junio 2011]
- [20] Wikipedia, *Motor Paso a Paso*. [En línea]. Disponible en: <http://es.wikipedia.org/wiki/Motor_paso_a_paso>. [Última consulta 17 de junio 2011]
- [21] Wikipedia, *Transistor Darlington*. [En línea]. Disponible en: <http://es.wikipedia.org/wiki/Transistor_Darlington>. [Última consulta 17 de junio 2011]
- [22] Facultad de Ingeniería, Universidad de la Plata, Argentina. [En línea]. Disponible en: <www.ing.unlp.edu.ar/islyd/Presentacion_steppers.ppt>. [Última consulta 21 de junio de 2011]
- [23] Gopal, M. **Control Principles and Design**. 2da edición. Tata McGraw-Hill, 2002. Nueva Delhi. [Última consulta 27 de junio 2011]

Apéndices.

A1. Abreviaturas y glosario.

Ángulo de Azimut: medida angular utilizada en sistema de coordenadas esféricas.

Bit: acrónimo para dígito binario. Es un dígito del sistema binario.

Byte: 8 bits.

Datagrama: unidades principales de información lógica.

Estación Terrena: estación ubicada en la Tierra que permite el intercambio de información con estaciones espaciales o satélites.

E_{ss}: error de estado estacionario

Ethernet: estándar de comunicación de redes.

HUB: dispositivo que permite centralizar una red y ampliarla.

ICE: Instituto Costarricense de Electricidad.

ITCR: Instituto Tecnológico de Costa Rica.

JAVA®: lenguaje de programación orientado a objetos.

NetBeans®: plataforma de desarrollo para diferentes lenguajes de programación.

ORACLE®: plataforma para creación y administración de bases de datos.

RF: Radio Frecuencia.

Socket: mecanismo para intercambio de paquetes de datos.

TCP: protocolo de comunicación que es orientado a conexión y fiable.

T_r: tiempo de subida

T_s: tiempo de estabilización.

UDP: protocolo basado en el intercambio de datagramas.

VSAT: Terminal de apertura muy pequeña (Very Short Aperture Terminal).

A2. Manual de usuario.

En la página de inicio de la página web se solicita el nombre de usuario y contraseña, tal y como se muestra en la figura 6.5 de la sección de análisis de resultados. Si el usuario es encontrado en la base de datos se le da la bienvenida y se muestra la página de la figura 6.6, caso contrario el usuario no puede avanzar más y se le muestra la página de la figura A.1.



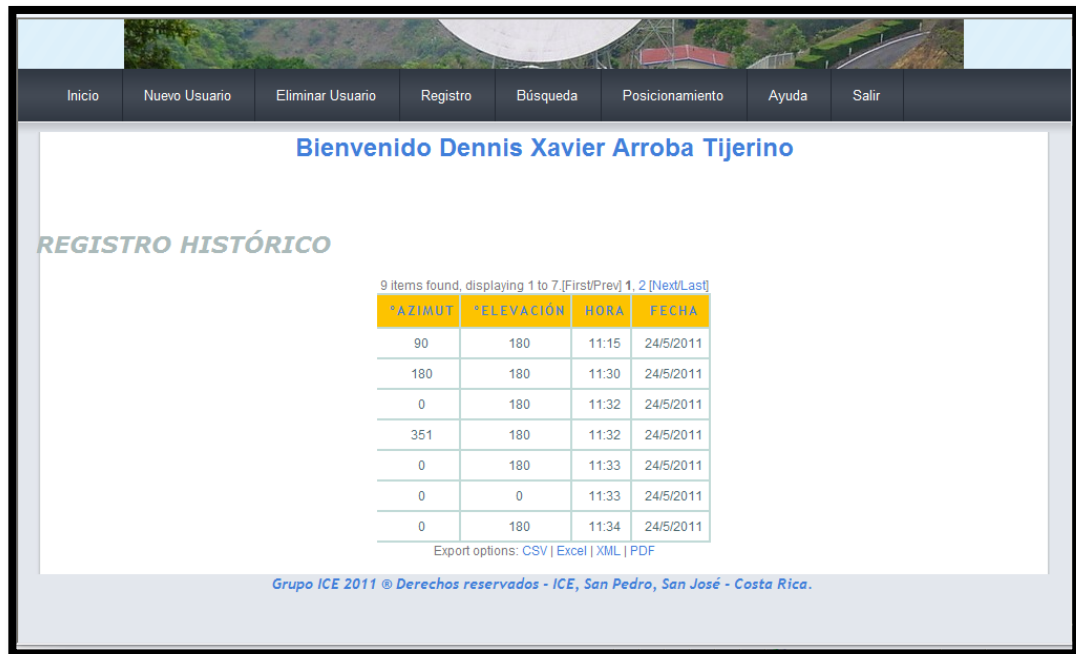
The screenshot displays a web interface for 'CONTROL REMOTO ESTACIÓN TERRENA TARBACA'. At the top, there is a navigation menu with the following items: Inicio, Nuevo Usuario, Eliminar Usuario, Registro, Búsqueda, Posicionamiento, Ayuda, and Salir. The main content area features a 'Bienvenido' heading, followed by a red error message: 'USTED NO ES USUARIO DEL ICE'. Below this, there are two input fields: 'Nombre de usuario :' with the text 'DeArr' and 'Contraseña :'. An 'Ingresar' button is positioned below the password field. At the bottom of the page, a footer reads: 'Grupo ICE 2011 © Derechos reservados - ICE, San Pedro, San José - Costa Rica.'

Figura A.1 Página web con mensaje de error de usuario.

Una vez dentro del sistema, se puede agregar o eliminar usuario. Para agregar se requiere de nombre completo, número de cédula, correo electrónico y el tipo de perfil que tendrá, ya sea visitante, usuario o administrador. Esto se ilustró en la figura 6.7.

Al eliminar un usuario de la base de datos, se requiere primero, tener perfil de administrador y posteriormente, ingresar el número de cédula del usuario a eliminar. Esto se mostró en la figura 6.8.

El registro histórico permite observar todos y cada uno de los posicionamientos que se le han realizado al sistema, incluyendo hora y fecha del mismo. (Figura A.2)



AZIMUT	ELEVACIÓN	HORA	FECHA
90	180	11:15	24/5/2011
180	180	11:30	24/5/2011
0	180	11:32	24/5/2011
351	180	11:32	24/5/2011
0	180	11:33	24/5/2011
0	0	11:33	24/5/2011
0	180	11:34	24/5/2011

Figura A.2 Página de visualización del registro histórico.

También existe la opción que permite buscar algún registro por fecha de posicionamiento, tal y como se mostró en la figura 6.9 del capítulo 6.

La sección de posicionamiento tiene dos opciones dentro, una es el posicionamiento normal, y la otra el modo Anclaje/Parada.

Para posicionar la antena se deben especificar uno o los dos ángulos deseados. Si es el mismo ángulo, el sistema ni guarda en base de datos los valores repetidos ni mueve los motores puesto que es donde se encuentran.

El modo Parada permite que el sistema no se mueva a ninguna nueva posición. Para ingresar a este modo, se le da clic sobre la opción del menú izquierdo sobre Anclaje/Parada. Una vez dentro, se presiona el botón de valor "0" bajo el título STOP.

Un LED rojo fijo indica que se encuentra en este modo. Para ingresar al modo Anclaje, habiendo presionado anteriormente el botón bajo el título STOP, se presiona el que está debajo de STOW. Automáticamente los motores mueven la antena a la posición predefinida y el LED que antes estaba fijo ahora parpadea para indicar que la antena está anclada.

Para salir del modo Anclaje, se presiona nuevamente el botón bajo la columna STOW, y para salir del modo Parada se procede a realizar lo mismo con el botón bajo STOP. En modo STOP, los botones deben estar como se muestra en la figura A.3. Para el modo STOW, deben estar como se indica en la figura A.4.



Figura A.3 Página web en funcionamiento normal.



Figura A.4 Página web para modo STOP.



Figura A.5 Página web para modo STOW.

Finalmente, la opción de Ayuda brinda una pequeña información adicional acerca del proyecto.

Para salir del sistema, se presiona en el menú la opción Salir, que devuelve al usuario a la pantalla de ingreso (Figura 6.5).

Anexos.

Anexo 1. Hoja de datos del sistema de control.



Model 133 Antenna Control System

Precision control of satellite tracking

The Vertex RSI control system can be used with almost any antenna for precision satellite tracking, telemetry and control, radar, radio and optical telescope applications.

- Pointing, Steptrack, Optrack, Monopulse systems
- Fixed or transportable earth stations
- Accommodates AZ/EL, X-Y, HA/Dec, AZ/EL/Tilt mounts
- Single or multiple linear variable speed antenna motor drives
- Stable or Inclined GEO, MEO, LEO and celestial targets
- L, S, C, X, Ku, Ka, V or multiband operation



System

The Antenna Control Unit is the primary control and monitor interface point for the system, featuring PC based processing and a friendly windowing interface.

The Antenna Control Unit, Tracking Receiver and Power Drive Unit are linked via serial connections. This provides the flexibility to configure the key system components without concern about separation distances or electrical ground plane transients.

Tracking Accuracy - Optrack

Normally better than 5% of the receive beamwidth in winds of 30 mph gusting to 45 mph, satellite inclination of up to 15° and signal scintillation of up to 2 dB.

Pointing Accuracy

Normally better than 0.005 RMS in winds of 30 mph gusting to 45 mph. This includes all drive train errors, but excludes structural errors between the position transducers and RF beam.

Operational Modes			*Optional
Tracking	Pointing	Acquisition	Other
Optrack	Intelsat 11	Box Scan*	Maintenance
Steptrack	Memtrack	Spiral Scan*	Manual
Monopulse*	StarTrack	Geo Scan*	Stop
	Preset	Raster Scan*	Computer
	Designate		Simulator
	NORAD*		Polarization*
	TableTrack*		Auto Stop*

GENERAL DYNAMICS
C4 Systems



Specifications

Weights and Dimensions

PDU:

High Power, multiple motor per axis

Shipping - 89H x 77W x 31D, 1600 lbs.
 Net - 81H x 71W x 20D, 1095 lbs.
 120/220 VAC, 1Ø, 3A/1.5A

Single DC motor per axis

Shipping - 79H x 40W x 27D, 800 lbs.
 Net - 71H x 32W x 16D, 400 lbs.
 120/220 VAC, 1Ø, 3A/1.5A
 208/380/415 VAC, 3Ø, motor dependent
 208/380/415 VAC, 3Ø, motor dependent

Transducers

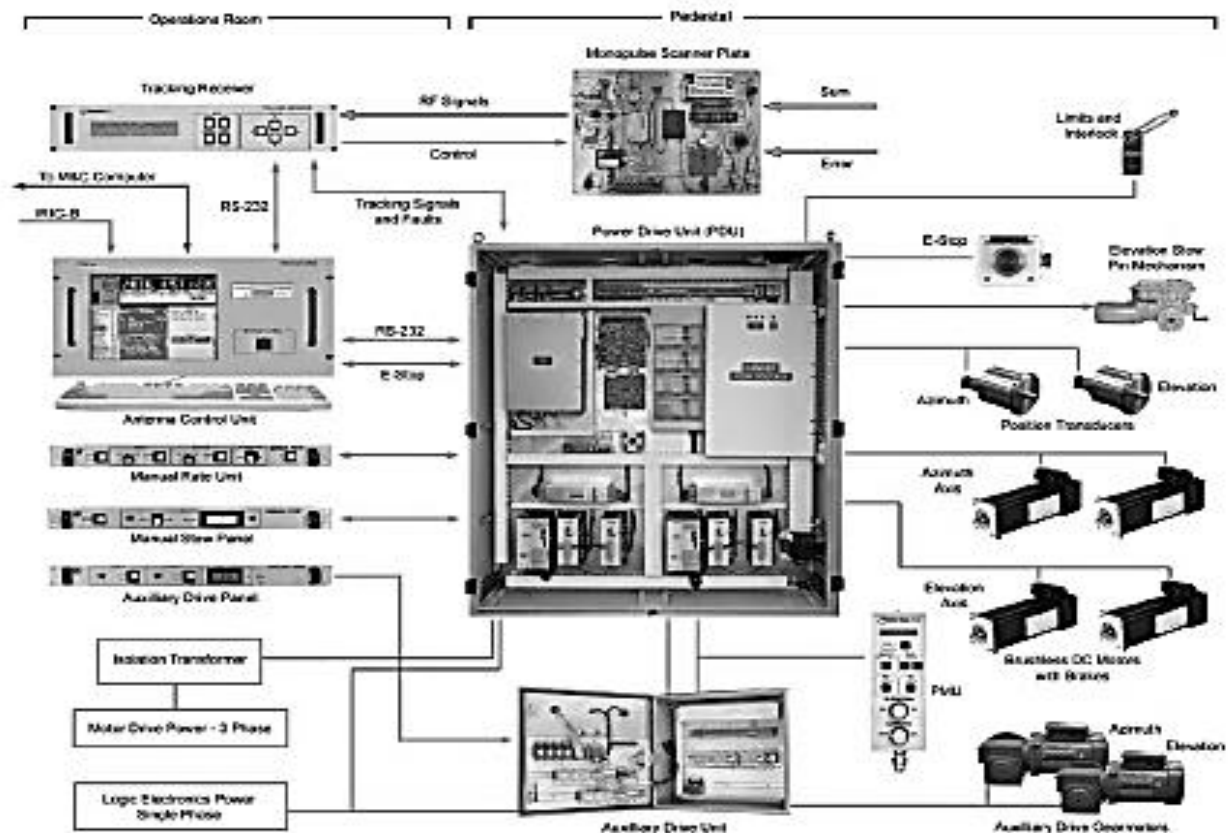
High Accuracy Position Transducer

- 0.0003° Resolution, 0.003° RMS accuracy
- 20-bit, 16:1 design, absolute



Optical Encoders

- Available with resolution to 29 bit, and accuracy to <1 arc second.



GENERAL DYNAMICS

1219 Digital Drive • Richardson, Texas 75081 • Tel: 972-907-9599 • Fax: 972-907-0027 • www.gdsatcom.com

© 2006 General Dynamics. All rights reserved. General Dynamics reserves the right to make changes in its products and specifications at any time and without notice.

GDSatcomTech 004 Rev A Oct 2006

Anexo 2. Hoja de datos del kit de desarrollo.

Product Documentation

SBC65EC

2 Features

- Has 32 general purpose user programmable I/O ports that can be monitored, configured and updated via web pages, HTTP and UDP. Twelve inputs can be configured as 10 bit Analog Inputs, and 4 as 10-bit PWM outputs.
- Programmed with Modtronix SBC65EC Web Server, for details [click here](#).
- Can be configured and controlled via web pages, for details [click here](#).
- Accepts commands via UDP or HTTP for configuring the board and controlling outputs, for details see modtronix.com/products/sbc65ec/doc/commands
- Implements tags that can be placed on web pages to display dynamic information and current input status, for details see modtronix.com/products/sbc65ec/doc/tags
- Programmed with a bootloader for updating firmware over the network or internet.
- Is part of our MicroX product range, meaning you can upgrade or expand it with any of the other MixroX products. For details see modtronix.com/microx
- Can be used as a daughter board to Ethernet enable any product.
- Diode protected 2.1mm power connector for standard DC transformer. Center is positive.
- 98KBytes FLASH, 3840 Bytes SRAM and pluggable EEPROM. Default TCP/IP stack uses less than half the available memory, which leaved heaps of code space for custom code.
- Has space for a 8 pin Ramtron SPI FRAM chip (32Kbyte FM25256 chip for example) to be assembled on the bottom of the board.
- Wide operating voltage range from 7 – 30V.
- Default operating frequency of 40MHz, software configurable low power mode that runs at 10MHz.
- Red 3mm User programmable LED.
- RJ45 connector with built in LEDs to indicate link and activity status.
- Micro Match connector for connecting a LCD2S serial LCD display with keypad decoder. For details on LCD2S range of LCD displays, see modtronix.com/products/lcd2s
- RS232 interface via 3 pin Molex type connector or Daughter Board connector.
- Has a 40 pin Daughter Board connector. For details see modtronix.com/microx/expansion.
- Assembled with brand name, quality components. For example, electrolytic capacitors used are extra long life rated (5 times more than standard), Industrial rated semiconductors (not commercial).
- Has an ICSP (In Circuit Serial Programming) connector (ICPC1 type) - CPU can be programmed and debugged in circuit. For details see modtronix.com/picboards/prog.
- Programmed with free Modtronix TCP/IP stack that features:
 - Includes MAC, IP, ARP, ICMP, TCP, UDP, HTTP, HTTP Compression, FTP, DHCP, NetBIOS, DNS, MXFS
 - Socket support for TCP and UDP
 - Portable across PIC18 MCUs
 - Out-of-box support for Microchip MPLAB C18 and Hi-Tech PICC-18 compilers
 - RTOS independent
 - Full TCP state machine
 - Modular Design

Anexo 3. Hoja de datos del ULN2003.



ULN2001A-ULN2002A ULN2003A-ULN2004A

SEVEN DARLINGTON ARRAYS

- SEVEN DARLINGTONS PER PACKAGE
- OUTPUT CURRENT 500mA PER DRIVER (800mA PEAK)
- OUTPUT VOLTAGE 50V
- INTEGRATED SUPPRESSION DIODES FOR INDUCTIVE LOADS
- OUTPUTS CAN BE PARALLELED FOR HIGHER CURRENT
- TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY LAYOUT

DESCRIPTION

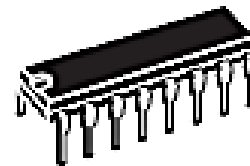
The ULN2001A, ULN2002A, ULN2003 and ULN2004A are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 800mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout.

The four versions interface to all common logic families :

ULN2001A	General Purpose, DTL, TTL, PMOS, CMOS
ULN2002A	14-25V PMOS
ULN2003A	5V TTL, CMOS
ULN2004A	5-15V CMOS, PMOS

These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays filament lamps, thermal print-heads and high power buffers.

The ULN2001A/2002A/2003A and 2004A are supplied in 16 pin plastic DIP packages with a copper leadframe to reduce thermal resistance. They are available also in small outline package (SO-16) as ULN2001D/2002D/2003D/2004D.



DIP18

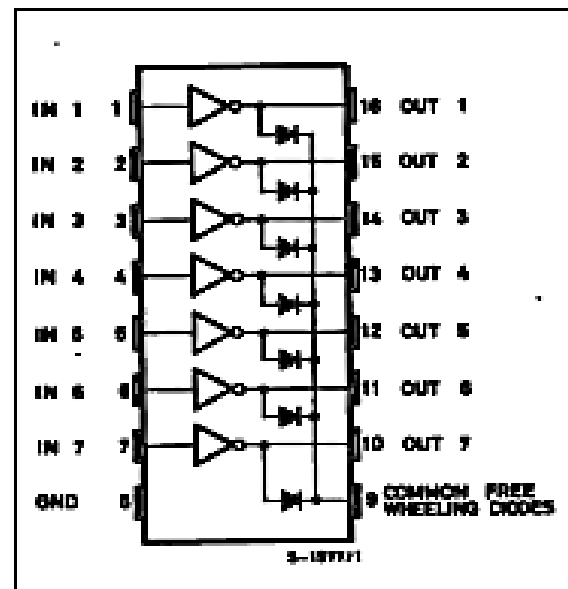
ORDERING NUMBER: ULN2001A/2A/3A/4A



SO18

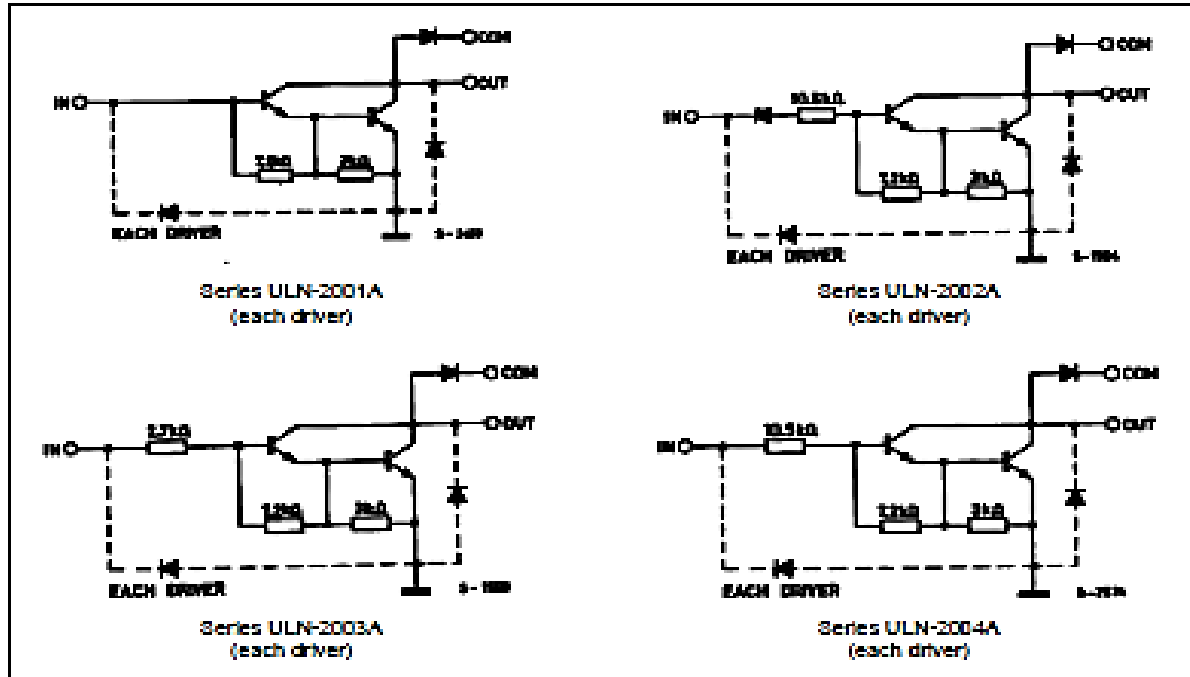
ORDERING NUMBER: ULN2001D/2D/3D/4D

PIN CONNECTION



ULN2001A - ULN2002A - ULN2003A - ULN2004A

SCHEMATIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V
V_{in}	Input Voltage (for ULN2002A/D - 2003A/D - 2004A/D)	30	V
I_c	Continuous Collector Current	500	mA
I_b	Continuous Base Current	25	mA
T_{amb}	Operating Ambient Temperature Range	-20 to 85	°C
T_{stg}	Storage Temperature Range	-55 to 150	°C
T_j	Junction Temperature	150	°C

THERMAL DATA

Symbol	Parameter	DIP18	SO18	Unit
$R_{\theta(j-a)}$	Thermal Resistance Junction-ambient	Max. 70	120	°C/W

Anexo 4. Fotografía del prototipo.

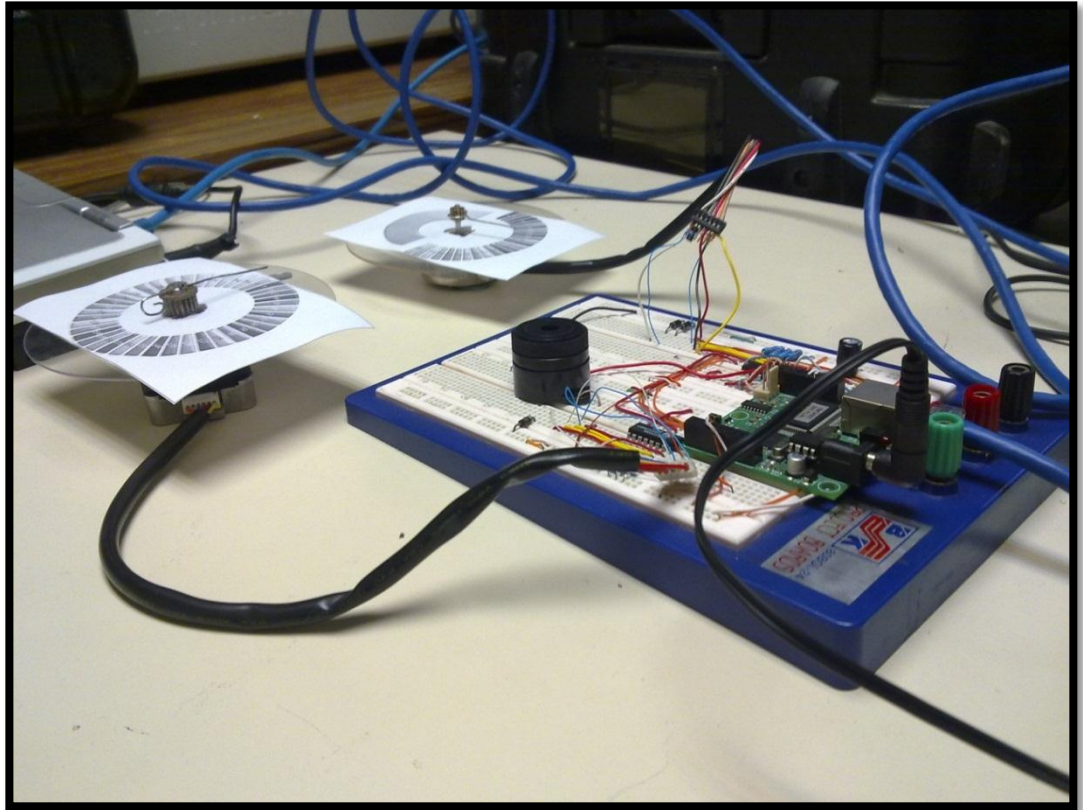


Figura B.1 Fotografía del prototipo montado.