

reporte final: Residencia profesional

presenta: Vásquez Martínez Agustín

octubre 2018

Resumen

Los registros geofísicos de pozos (RGP) se llevan a cabo para analizar la composición del subsuelo, para obtener estos resultados se utilizan herramientas conocidas como sondas las cuales tienen un malacate el cual se introduce en el pozo emitiendo ciertas señales sobre la pared del mismo y van midiendo su respuesta física, los resultados se van dibujando en una gráfica que está en función de la profundidad del pozo y se conocen como curvas. Los RGP se obtienen en tiempo real y luego de una interpretación de las curvas obtenidas permiten detectar zonas donde sería posible la explotación de agua a mayor escala, llevando a tomar una decisión sobre el futuro del pozo, por ejemplo su expansión.

Las sondas que realizan RGP pueden tener como salida varias curvas, a su vez pueden graficar dichas curvas en papel o en formato digital, todo esto depende de qué tan sofisticadas sean.

Por lo tanto se propuso un desarrollar un programa de computadora con la capacidad de obtener las curvas y que las fuera graficando en tiempo real dentro de una interfaz de usuario en la computadora, esto elimina la necesidad de tener el papel que utiliza la sonda para hacer el dibujo de las curvas, además al obtener los datos en formato digital se permite guardarlos en archivos los cuales se pueden utilizar posteriormente en reportes o difusión.

Agradecimientos

Aquí escribir cosas bonitas sobre la gente que te ayudó en este trabajo :)

Índice general

1. Introducción	1
1.1. Antecedentes	1
1.2. Problemas a resolver	3
1.3. Justificación	4
1.4. Alcance	5
1.5. Hipótesis	5
1.6. Objetivos	6
1.6.1. Objetivo General	6
1.6.2. Objetivos Específicos	6
1.7. Metología de la investigación	7
1.8. Estructura del informe	7
1.8.1. Capítulo 2 - Estado del arte	7
1.8.2. Capítulo 3 - Marco teórico	7
1.8.3. Capítulo 4 - Prototipo del software	7
1.8.4. Capítulo 5 - Análisis de la sonda	8
1.8.5. Capítulo 6 - Sensores de movimiento	8
1.8.6. Capítulo 7 - Módulo de adquisición de datos	8
1.8.7. Capítulo 8 - Programación del software	8
1.8.8. Capítulo 9 - Conclusiones	8

2. Estado del arte	9
2.1. Equipos para RGP	9
2.2. Parámetros que registra un RGP	10
2.3. Conclusiones	12
3. Marco teórico	13
3.1. Señales	13
3.1.1. Señal Continua	15
3.1.2. Señal Discreta	15
3.1.3. Señal analógica	17
3.1.4. Señal digital	17
3.1.5. Parámetros de una señal	18
3.1.5.1. Amplitud	18
3.1.5.2. Período	19
3.1.5.3. Frecuencia	19
3.2. Arduino	21
3.2.1. Microcontrolador Atmega328	21
3.2.2. Pines de entrada/salida	23
3.2.3. Conector USB	23
3.3. HC-SR04	24
3.4. MATLAB	26
3.4.1. MATLAB Runtime	27
3.5. Git	28
3.5.1. Github	28
4. Prototipo del software	30
4.1. Programación de arduino	30
4.2. Programación de MATLAB	32

5. Analizando la sonda	34
5.1. Búsqueda de la señal	34
5.1.1. Análisis del movimiento de la pluma	36
5.1.2. Conclusión	38
6. Sensores de movimiento	39
6.1. CNY70	39
6.2. HC-SR04	42
6.2.1. Reprogramación de arduino	44
6.2.2. Reprogramación de MATLAB	46
6.3. Resultados del sensor	49
7. Módulo de adquisición de datos	50
7.1. Materiales	50
7.2. Construcción	51
8. Programación del software	55
8.1. Archivos CSV	55
8.2. Control del trabajo	58
8.3. Interfaz gráfica	58
8.3.1. Elementos	58
8.3.2. Colores	62
8.4. Manejo de excepciones	63
8.4.1. Inicio de lecturas sin el módulo conectado	63
8.4.2. Desconexión del módulo de adquisición de datos	64
8.4.3. Eventos	64
8.5. Reportes en PDF	65
8.6. Empaquetado	65

9. Resultados	67
9.1. Conclusiones	67
9.2. Trabajo a futuro	68
 A. Prueba de correo electrónico	 69
 B. Ethernet	 70
 C. Interfaz de MATLAB	 71
 D. Formato de datos en C	 72

Índice de figuras

2.1. Equipo de RGP montado en un vehículo	9
2.2. Equipo RGP enviando curvas a una computadora	10
3.1. Ejemplo de señal producida por el lanzamiento de una moneda	14
3.2. Plano cartesiano	14
3.3. Señal del lanzamiento de una moneda con $\Delta x = 2$	15
3.4. Gráfica de una señal continua	16
3.5. Gráfica de una señal discreta	16
3.6. Señal digital	17
3.7. Señales con diferentes amplitudes	20
3.8. Señal con periodo definido	20
3.9. Señal con periodo no definido	20
3.10. Señales con diferentes frecuencias	21
3.11. Placa arduino, modelo <i>UNO</i>	22
3.12. Placa arduino UNO conectada a una computadora	24
3.13. Sensor HC-SR04	25
3.14. Esquema del funcionamiento del HC-SR04	26
4.1. Potenciómetro conectado	31
4.2. Gráfica producida por el prototipo de software	33
5.1. Potenciómetros de la sonda (vista frontal)	35

5.2. Potenciómetros de la sonda (vista trasera)	35
5.3. Ejemplo 1 de span	35
5.4. Ejemplo 2 de span	36
5.5. Puente utilizando caimanes	37
5.6. Pluma de la sonda	37
5.7. Discos, alambre fino y pluma	38
6.1. Sensor CNY70	40
6.2. Prueba en laboratorio de los sensores CNY70	41
6.3. Prueba en la sonda de los sensores CNY70	42
6.4. Sensores HC-SR04 conectados al arduino	44
6.5. Midiendo las zonas de dibujo	47
6.6. Esquema del área de dibujo del potencial espontáneo	48
6.7. Interfaz del software usando los sensores HC-SR04	49
7.1. Hilos originales organizados en pares	51
7.2. Hilos utilizados en el módulo	51
7.3. Cables soldados al sensor	52
7.4. Cables para cada uno de los sensores	52
7.5. Agujero para el puerto serial del arduino	53
7.6. Interior del chasis	53
7.7. Resultado final, módulo de adquisición de datos	54
8.1. Guardando nuevo archivo CSV	56
8.2. Estructura de archivo CSV	57
8.3. Interfaz principal de anita graficando datos	59
8.4. Botones para archivos	60
8.5. Botones para el control de lecturas	60
8.6. Etiqueta que muestra el estado de las lecturas	60

8.7. Controles de span	61
8.8. Etiquetas de máximos y mínimos	61
8.9. Control de intervalo de lecturas	61
8.10. Etiquetas para visualizar el tiempo de trabajo	62
8.11. Icono de anita	63
8.12. Alerta cuando el módulo no está conectado	64
8.13. Alerta por desconexión repentina del módulo	64
8.14. Interfaz para la generación de reportes en PDF	65
8.15. Software empaquetado para su distribución	66
B.1. Puerto Ethernet y un cable	70

Capítulo 1

Introducción

1.1. Antecedentes

La geología es una ciencia que se dedica al estudio de la composición superficial e interna del planeta, una de sus disciplinas es la geofísica, que se encarga de estudiar las condiciones físicas de la tierra y sus fenómenos, por ejemplo los sismos, volcanes, oceános e incluso la atmósfera, la ingeniería especializada en geofísica abarca los estudios que permiten la exploración, evaluación y explotación de recursos del subsuelo, dentro de sus actividades están los registros geofísicos de pozos (RGP), los cuales son una manera de analizar la composición del subsuelo que rodea a un pozo para identificar tipos como el arenoso, pedregoso, húmedo, entre otros, también sirven para identificar el grosor de cada una de las capas que lo compone y la variabilidad del almacenamiento de agua subterránea. Los ingenieros geofísicos llevan a cabo estos estudios con la finalidad de conocer las capacidades que tiene la tierra y así determinar si son viables o no para ciertos trabajos. Los RGP son estudios que brindan resultados en tiempo real y para realizarlos se utilizan herramientas conocidas como sondas.

Una sonda está compuesta de un malacate el cual puede variar en longitud, es un

material parecido a un cable sin aislante al cual se le inyecta corriente eléctrica la cual viaja hasta el extremo introducido donde un electrodo es responsable de emitir esa corriente en la pared del pozo, después mide la respuesta física del subsuelo y la sonda la va graficando en la superficie, esa gráfica se conoce como *curva* y corresponde a un parámetro del subsuelo. Para llevar a cabo esta tarea la sonda puede utilizar rollos de papel o computadoras dependiendo de su nivel de fabricación y tecnología.

Dentro del Centro Interdisciplinario de Investigación para el Desarrollo Integral Regional (CIIDIR) Unidad Oaxaca cuentan con una sonda capaz de graficar dos parámetros:

- **Potencial Espontáneo (SP)** - Es un registro de fenómenos físicos, que ocurren naturalmente en las rocas de una formación geológica, esta curva registra el potencial eléctrico producido por la interacción con el agua subterránea, el voltaje generado es resultado de una corriente continua que se genera en dichos bordes por la diferencia de salinidad. [5]
- **Resistividad (R)** - Es la capacidad que tienen las rocas de oponerse al paso de corriente eléctrica inducida y varia dependiendo de la sal disuelta en los fluidos presentes en los poros de las rocas, esta curva proporciona evidencias del contenido de fluidos en las rocas. [6]

Dicha sonda utiliza rollos de papel cuadriculado dividido en dos zonas para cada una de las gráficas entregando datos en tiempo real. Fue fabricado por un empresa ya desaparecida la cual también es responsable de la elaboración de los rollos de papel, por lo que se buscó una manera de eliminar la necesidad del mismo, llevando las señales de ambas gráficas hacia una interfaz en una computadora.

Sabiendo esto se propuso diseñar un programa que desde una computadora le permitiera a una persona la capacidad de comunicar y procesar las señales de Potencial Espontáneo y Resistividad, la propuesta se entregó al Instituto Tecnológico de Oaxaca en su Departamento de Sistemas y Computación, dado que los estudiantes de su

ingeniería en Sistemas Computacionales tienen los conocimientos y habilidades esenciales para llevar a cabo el desarrollo de programas de computadora, éstos se conocen como software. Durante esta investigación se logró graficar en tiempo real las señales dentro de la interfaz de usuario del software programado, obteniendo los datos en forma totalmente digital.

Se diseñó el software tal que una vez terminadas las lecturas emita un reporte en formato PDF¹, además de generar un respaldo en un archivo de valores separados por comas (CSV), cada uno de estos archivos CSV se refiere a una sesión de trabajo y se pueden crear nuevos o abrir uno ya hecho y continuar el trabajo desde ahí.

1.2. Problemas a resolver

En esta investigación se resuelven los siguientes problemas:

1. El papel que utiliza la sonda para dibujar las gráficas de parámetros del subsuelo ya no se fabrica, es difícil conseguirlo e implica una supervisión constante para no agotar el material y quedarse a medias en el trabajo.
2. Encontrar la señal que la sonda utiliza para dibujar las curvas.
3. Diseñar un módulo de adquisición de datos que permita mandar la señal de la sonda hacia la computadora.
4. Montar el módulo de adquisición de datos en la sonda.
5. Programación de un software con la capacidad de recibir, graficar y procesar las señales que capture el módulo de adquisición de datos.

¹Siglas en inglés de Portable Document Format, es un formato de documentos que puede ser leido en los principales sistemas operativos: GNU/Linux, Windows, MacOs, Unix [3].

1.3. Justificación

El análisis de los RGP da a conocer la estructura y composición del subsuelo y permite determinar variaciones en la cantidad de agua subterránea que rodea a un pozo, durante la perforación de pozos de agua este dato es importante porque es un punto de partida a tomar en cuenta durante el trabajo de perforación, puesto que la expansión en anchura y en profundidad depende del análisis de las curvas, continuar la perforación de un pozo sin estos datos es decir, al azar, sería un gasto arriesgado de recursos económicos y de tiempo.

Los equipos de RGP actuales cuestan alrededor de 580,000 MXN², por lo que al no contar con los recursos económicos para adquirir un equipo moderno ésta es una alternativa que permite reutilizar un equipo antiguo. La captura de información en un medio digital permite reiniciar en cualquier momento el registro sin preocuparse porque el papel ya no se pueda utilizar, con esto se alcanza el beneficio de crear, eliminar o reanudar un registro cuando se desee.

²MXN = Pesos mexicanos. Se comprueba esta información en el apéndice «A».

1.4. Alcance

En este trabajo se capturan las curvas de Potencial Espontáneo y de Resistividad provenientes de la sonda para dibujar su gráfica en una computadora, por lo que se pone en marcha un software capaz de dibujar estas señales en la pantalla dentro de una interfaz gráfica de usuario, dándole al usuario la capacidad de crear nuevos registros o reanudar los ya existentes, siempre y cuando fueran generados son el software producto de esta investigación.

El usuario tambien tendrá la posibilidad de controlar el flujo de lecturas, esto quiere decir que puede iniciar el proceso, pausar, reanudar o detenerlo, tambien será posible controlar el intervalo de tiempo que ocurre entre cada una de lectura. A partir de los datos obtenidos se generarán reportes en formato PDF con el encabezado del Instituto Politécnico Nacional y del CIIDIR Oaxaca, el software se distribuirá bajo una licencia MIT [4] y sus versiones empaquetadas listas para instalar estarán disponibles en un repositorio dentro de Internet. Su código será de libre acceso y no será necesario pagar una licencia para utilizarlo. Finalmente será posible instalarlo en el sistema operativo Windows de 64 bits.

1.5. Hipótesis

1. Con la adquisición de los datos de las curvas en una computadora se tendrá una mejor perspectiva de las gráficas, puesto que cada lectura realizada quedará grabada en un archivo de valores separados por comas.
2. Al obtener un reporte de las curvas en un formato PDF será más sencillo compartir los estudios realizados.

1.6. Objetivos

1.6.1. Objetivo General

Diseñar, programar y poner en marcha un software para la comunicación y procesado de señales de Potencial Espontáneo y de Resistividad, producto de las mediciones en el subsuelo a través de un equipo de registros eléctricos.

1.6.2. Objetivos Específicos

- Estudio del Registro Geofísico de Pozos, de los equipos que utilizan para llevar a cabo los registros y de la graficación de las curvas.
- Inspección de la sonda proporcionada por el CIIDIR con el fin de hallar la señal responsable de dibujar la curva.
- Diseño de un circuito que actúe como módulo de adquisición de datos para transmitir la señal de la sonda hacia la computadora.
- Programación de un software que permita graficar las curvas en tiempo real y genere archivos con sus datos, empaquetarlo y dejarlo listo para su instalación en otras computadoras.
- Realizar pruebas con el software y el módulo de adquisición de datos conectado a la sonda.

1.7. Metología de la investigación

En este trabajo se requiere primero de una metodología aplicada con la que se buscará una manera de lograr el objetivo de transmitir las curvas desde la sonda hacia la computadora, la metodología aplicada permite enfocarse a la medida abordando un problema en específico, después se requiere de una metodología experimental ya que el software para dibujar y procesar las señales requiere de pruebas dentro de un ambiente controlado llevando a replicar este evento una y otra vez haciendo observación de los resultados obtenidos para realizar ajustes en la programación del software.

1.8. Estructura del informe

1.8.1. Capítulo 2 - Estado del arte

Describe la forma en que actualmente se llevan a cabo los registros geofísicos de pozos, mostrando ejemplos de un equipo actual y sus características, se explican los parámetros que puede registrar una sonda y se mencionan algunas empresas que brindan el servicio de RGP.

1.8.2. Capítulo 3 - Marco teórico

Expone los conceptos que se utilizaron durante la investigación cuyo dominio necesitó ser esencial.

1.8.3. Capítulo 4 - Prototipo del software

Muestra el primer acercamiento a la graficación de señales analógicas captadas por un dispositivo físico y recibidas a través del puerto serial utilizando arduino y MATLAB.

1.8.4. Capítulo 5 - Análisis de la sonda

Documenta las observaciones que se llevaron a cabo en la sonda de RGP con el fin de conocer de dónde proviene la señal que luego se dibuja en las gráficas.

1.8.5. Capítulo 6 - Sensores de movimiento

Nombra los tipos de sensores que se utilizaron así como los experimentos realizados en el prototipo del módulo de adquisición de datos.

1.8.6. Capítulo 7 - Módulo de adquisición de datos

Muestra el desarrollo y construcción con fotos del módulo de adquisición de datos.

1.8.7. Capítulo 8 - Programación del software

Documentación acerca del software anita, desarrollado para cumplir los objetivos de esta investigación.

1.8.8. Capítulo 9 - Conclusiones

Capítulo final donde se presenta los problemas resueltos con esta investigación, también se proponen mejoras hacia el software conociendo los alcances que tuvo el presente proyecto.

Capítulo 2

Estado del arte

2.1. Equipos para RGP

A ctualmente un equipo de RGP como el de Century Geophysical modelo 300M [2] en venta al año 2018 está diseñado para montarlo de forma temporal o de forma permanente en un vehículo, tal como se muestra en la siguiente foto.

En la figura 2.1 se puede apreciar un extremo en forma de tubo gris, esta parte va por delante y es la primera que va descendiendo por el pozo, está sujetada por un malacate



Figura 2.1: Equipo de RGP montado en un vehículo.

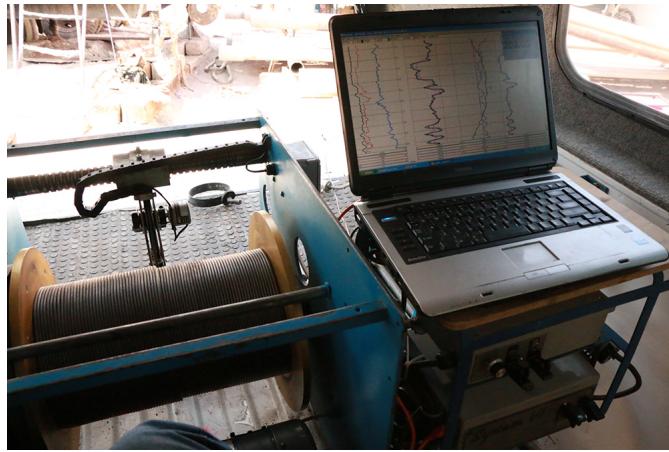


Figura 2.2: Equipo RGP enviando curvas a una computadora

que en la foto se muestra enrollado en un carrete. Este equipo viene provisto de software para el registro, procesado e impresión de las curvas en una computadora, se conecta vía Ethernet¹ y se aprecia en la figura 2.2.

Century Geophysical enlista [2] que el software provisto con este equipo es compatible con Windows, también destaca que cuenta con un servicio de ayuda online para la operación del software y manuales para todas las herramientas de registro con las que viene equipado este aparato.

2.2. Parámetros que registra un RGP

Entre las curvas de parámetros del subsuelo que las sondas pueden graficar están los siguientes: [1]

- **Registro Gamma (Gamma)** - Determina la cantidad de radiación natural que emiten los materiales que se encuentran en el subsuelo proporcionando información para la identificación de la columna litológica² registrada.
- **Registro de resistividad (R)** - Mide la resistividad eléctrica de los materiales

¹Tecnología estándar de comunicación cableada en redes de computadoras, detalle en el apéndice B.

²Litología: Parte de la geología que estudia las características de las rocas

que circundan el pozo, en función del arreglo electródico, ya sea de 8, 16, 32 o 64 pulgadas variará la profundidad de investigación o penetración hacia la formación, siendo de menor a mayor respectivamente. Este registro aporta información referente a litología y datos para estimaciones petrofísicas.³

- **Registro de resistencia eléctrica puntual (SPR)** - Mide la resistencia eléctrica del medio que rodea a la sonda en el pozo; tiene poca penetración hacia la formación pero es muy resolutivo a cambios litológicos y contrastes de materiales de bajo espesor.
- **Registro de potencial espontáneo (SP)** - Mide los potenciales eléctricos naturales originados en el interior del pozo por intercambio iónico entre los fluidos originales contenidos en la formación y el de perforación (que en el mejor de los casos deberá tratarse de un fluido cuya base sea agua dulce), reflejando cualitativamente permeabilidad y cambios litológicos.
- **Registro de temperatura** - Mide la temperatura del fluido (aire, agua o lodo) en el interior del pozo.
- **Registro de Resistividad de Fluido** - Determina la resistividad eléctrica del fluido contenido en el pozo, se define como el inverso de la conductividad eléctrica que se utiliza comúnmente junto con la temperatura como un indicador de la calidad o tipo de fluido.

Las curvas se dibujan en forma vertical de arriba hacia abajo y en función de la profundidad.

³Especialidad que conjuga conocimientos de ingeniería del petróleo, geofísica y geología

2.3. Conclusiones

La evaluación de los registros geofísicos de pozos es un punto clave durante la exploración del subsuelo, al llevar a cabo estos resultados se obtienen en tiempo real y con mucha precisión datos con los que se puede determinar la localización y la situación de los recursos naturales que se encuentren en el subsuelo, estos pueden ser yacimientos de agua, petróleo o gas. Tener el equipo necesario para realizar este tipo de registros les da a las empresas [7],[8],[10],[11] una ventaja, ya que previenen la perforación de pozos donde no es necesaria logrando con esto un ahorro en los recursos económicos de sus clientes.

Capítulo 3

Marco teórico

3.1. Señales

Se conoce como señal a un conjunto de datos que representa información relacionando dos variables; una *dependiente* y otra *independiente*. El primer tipo de variable existe y es correspondida sólo cuando la variable independiente está presente, esto se conoce como *estar en función de*. Un ejemplo básico sería el lanzamiento de una moneda, donde la variable independiente sería el lanzamiento uno, el lanzamiento dos, el lanzamiento tres, el lanzamiento N, mientras tanto la variable dependiente sería un lado u otro de la moneda. Si se colocan el número de lanzamiento en una cuadrícula y se va anotando el resultado, al final se tendrá una señal en donde el resultado está en función del lanzamiento que le corresponda. Como se puede apreciar en la figura 3.1; la variable dependiente no puede existir sin la variable independiente, en este ejemplo tiene que haber un lanzamiento para que se produzca un resultado por tanto si no existe el lanzamiento número 21, no puede haber un resultado A o B.

Tal como se observa en la figura 3.1 se escribió el número de lanzamiento en la parte más horizontal más larga mientras que los resultados se acomodaron en forma

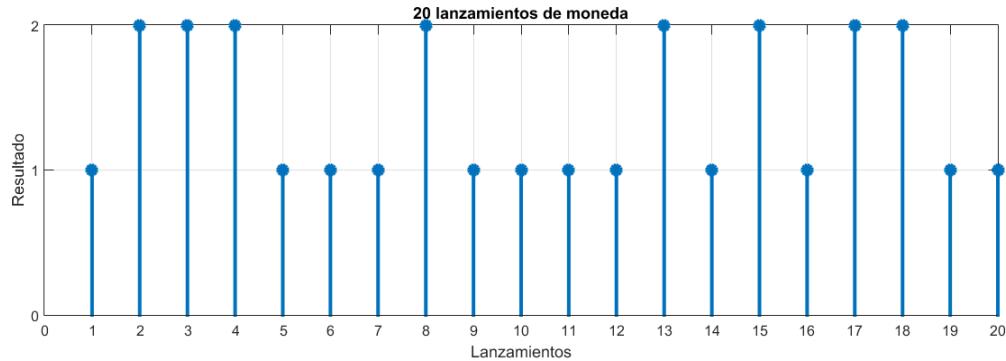


Figura 3.1: Ejemplo de señal producida por el lanzamiento de una moneda

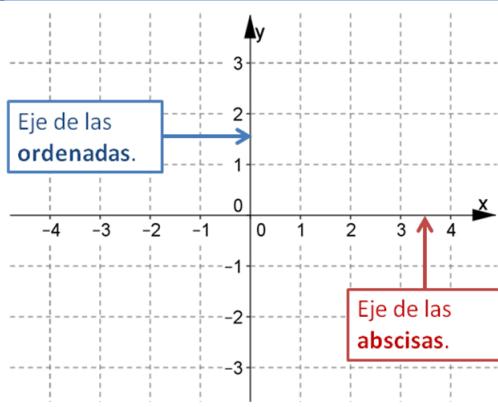


Figura 3.2: Plano cartesiano

vertical, éstos se conocen como ejes y a partir de ahora se les llamará *eje de las abscisas* o eje x al eje horizontal y *eje de las ordenadas* o eje y al eje vertical. Dichos nombres provienen del plano cartesiano el cual se compone de dos rectas perpendiculares cuyo punto que las corta es el origen o el inicio, trasladando esta idea al ejemplo de la moneda el origen sería el lanzamiento cero y el resultado cero.

Uno de las características de la variable independiente es la diferencia que existe entre cada valor que toma en el eje de las abscisas, se le conoce como *delta* y se representa con la letra griega Δ . En el ejemplo de la figura 3.1 el valor de Δ en x es de 1. ¿Qué pasaría si en lugar de registrar después de cada lanzamiento se registrara cada dos lanzamientos? En ese caso se tendría un $\Delta x = 2$ y la señal se vería como en la figura 3.3.

Los valores que una variable pueda tener y representar en sus ejes dan lugar a la

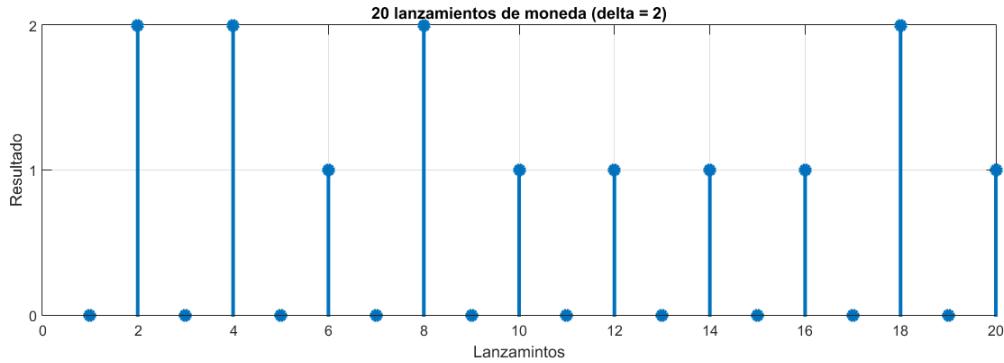


Figura 3.3: Señal del lanzamiento de una moneda con $\Delta x = 2$

clasificación de señales, las cuales se definen a continuación.

3.1.1. Señal Continua

Este tipo de señal puede tomar cualquier valor a lo largo del eje x . Su gráfica se puede apreciar como una sola línea que no se ve interrumpida a lo largo de toda su duración, ejemplo de estas señales pueden ser: una grabación de sonido, la gráfica de una función trigonométrica seno o coseno, la señal de radio que emite alguna cabina, los movimientos que produce un sismo, la figura 3.4 muestra un ejemplo de señal continua producto de la función $f(x) = \sin(2x) + \cos(3x)$.

En esta señal se puede apreciar que existen valores en el eje x con puntos decimales, sin embargo se corresponden con un valor en el eje de las ordenadas, por lo tanto existen haciendo que Δx tenga un valor muy pequeño. Existe un valor correspondiente en el eje y para el valor en $x = 3.2$ así como existe un valor en y para $x = 7.1$. Por lo tanto una señal continua puede tomar cualquier valor en el eje de las abscisas y obtener un resultado en el eje de las ordenadas.

3.1.2. Señal Discreta

Una señal discreta tiene valores definidos en el eje de las abscisas, a diferencia de las continuas y haciendo referencia a la figura 3.4 una señal discreta no podría evaluarse

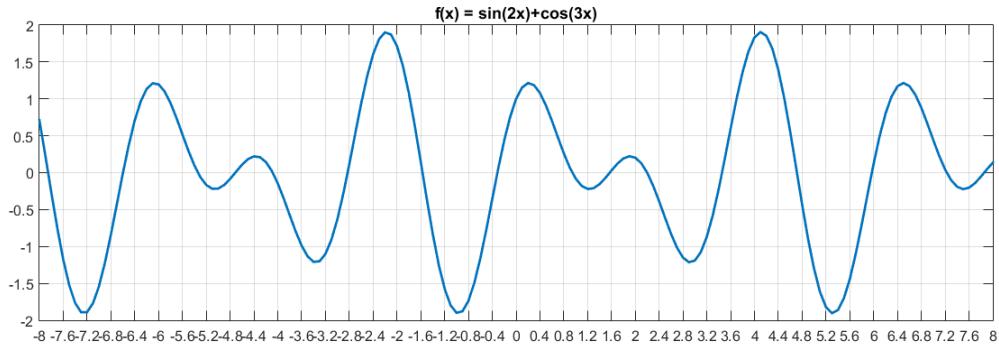


Figura 3.4: Gráfica de una señal producida por la función $f(x) = \sin(2x) + \cos(3x)$

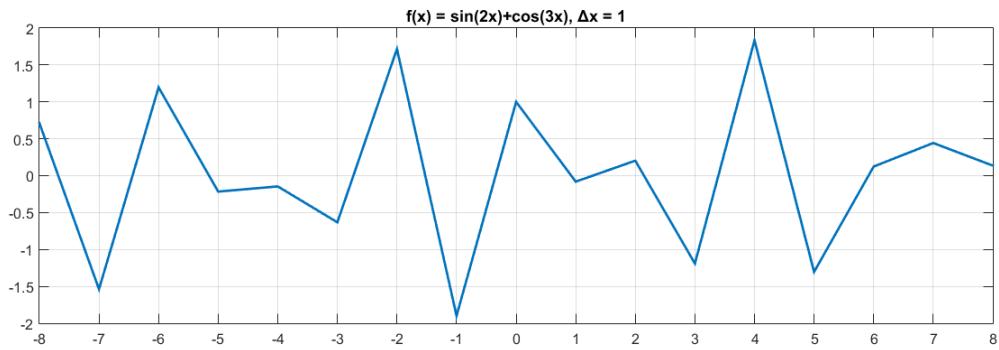


Figura 3.5: Gráfica de una señal producida por la función $f(x) = \sin(2x) + \cos(3x)$ con $\Delta x = 1$.

en una $x = 2.5$. Este tipo de señales tiene un Δ definido el cual nunca cambia a lo largo de toda la duración de la señal. Si se graficara la misma función de la figura 3.4 tomando $\Delta x = 1$ luciría como se muestra en la figura 3.5. Como se puede apreciar no toma un valor para $x = 3.2$ o $x = 7.1$, sino que sus valores en x van aumentando de uno en uno, no hay intermedios y esto es a lo que se refiere el término *discretos*. Pero no siempre los valores que una señal discreta puede tomar en su eje x son números enteros, todo depende de la Δ que se esté manejando, bien podría ir aumentando en 0.5, en 2, en 0.7, etc.

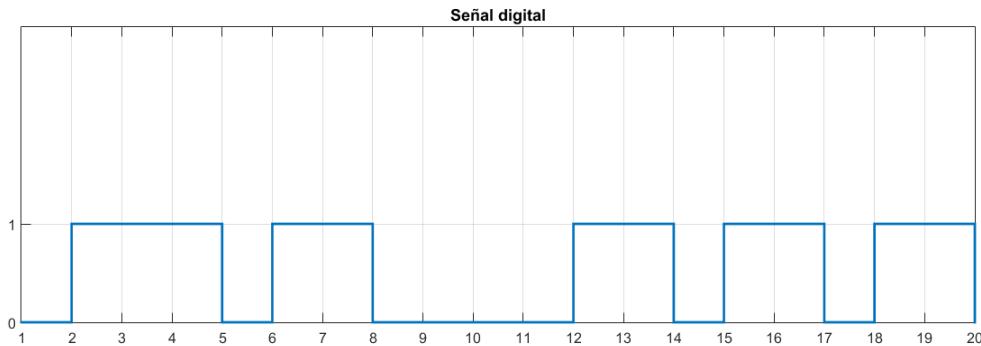


Figura 3.6: Señal digital

3.1.3. Señal analógica

Para que una señal pueda entrar en esta clasificación se requiere que sus valores en el eje de las ordenadas puedan ser cualesquiera, esto significa que en cada valor de x habrá un valor que le corresponda en y que puede ser un número entero o bien un número con puntos decimales. Las señales analógicas también pueden entrar en la categoría de las continuas o discretas dependiendo de las características que tenga su eje de abscisas. Por lo tanto ejemplos de una señal analógica se pueden apreciar tanto en la figura 3.4 como en la figura 3.5. En ambas se puede observar que para $x = 0$, $y = 1$, y que para $x = 4$, $y = \sim 1.8$.

3.1.4. Señal digital

Finalmente este tipo de señal se utiliza mucho en áreas como la electrónica digital y la informática. Las señales digitales se caracterizan por tomar valores en el eje de las ordenadas únicamente de ceros o uno, estos se conocen como valores binarios o valores booleanos, siendo que el cero representa un valor falso mientras que el uno representa un valor verdadero, sus aplicaciones incluyen la representación de información dentro de una computadora, cada valor que contenga la señal representa un bit de memoria, ésta es la unidad mínima que una computadora utiliza para llevar a cabo sus operaciones.

Continua	Analógica Digital
Discreta	Analógica Digital

Cuadro 3.1: Características que pueden combinar las señales

No existen valores intermedios como el 0.2 o el 0.75, los trabajos electrónicos que requieran de una señal digital suelen redondear hacia el cero o hacia el uno para llevar a cabo la conversión. Un ejemplo de señal digital se trató al principio de esta sección, durante el ejemplo de la moneda sólo pueden existir dos valores posibles en el eje y , el lado de la moneda A o el lado de la moneda B, es como los ceros y unos de una señal digital.

Las señales digitales pueden también ser de tipo continua o discreta, como ya se mencionó anteriormente todo depende de las características que tenga su eje de abscisas, así pues se define el cuadro 3.1.

Otra forma de recordar que combinación de clasificaciones puede tomar una señal es pensar en dónde se aplican dichas clasificaciones, no se puede ser discreta y continua al mismo tiempo porque ambas actúan sobre el eje x , así como no se puede ser analógica y digital al mismo tiempo ya que esas características actúan sobre el eje y . Dejando claro que un eje sólo puede tener una de ellas.

3.1.5. Parámetros de una señal

Toda señal sin importar su clasificación tendrá una amplitud, sin embargo no todas las señales son periódicas y dado que la frecuencia está relacionada al periodo de igual manera no todas las señales tendrán una frecuencia definida.

3.1.5.1. Amplitud

Es la distancia máxima que existe desde el eje de las abscisas hasta el punto más alto que puede alcanzar una señal.

3.1.5.2. Período

Una señal es periódica cuando repite un patrón una y otra vez, inmediatamente después de terminar un patrón, empieza uno nuevo que es exactamente igual al anterior. Este patrón también se conoce como ciclo y está relacionado con la frecuencia de la siguiente manera.

$$T = \frac{1}{f} \quad (3.1)$$

Donde T es el periodo y f la frecuencia.

3.1.5.3. Frecuencia

El número de ciclos que una señal complete en una unidad de tiempo es la frecuencia, su unidad dentro del Sistema Internacional es el *hercio* y se abrevia **Hz**. Tal como lo muestra la ecuación 3.1 la frecuencia es el inverso del periodo, esto quiere decir:

$$f = \frac{1}{T} \quad (3.2)$$

Si una persona parpadea una vez por segundo, se tendrá una frecuencia de 1 Hz (un ciclo) completado en $T = 1$ segundo. Si la persona en cambio parpadea dos veces por segundo, entonces se tendrá una frecuencia de 2 Hz, (dos ciclos) completados en $T = 0.5$ segundos.

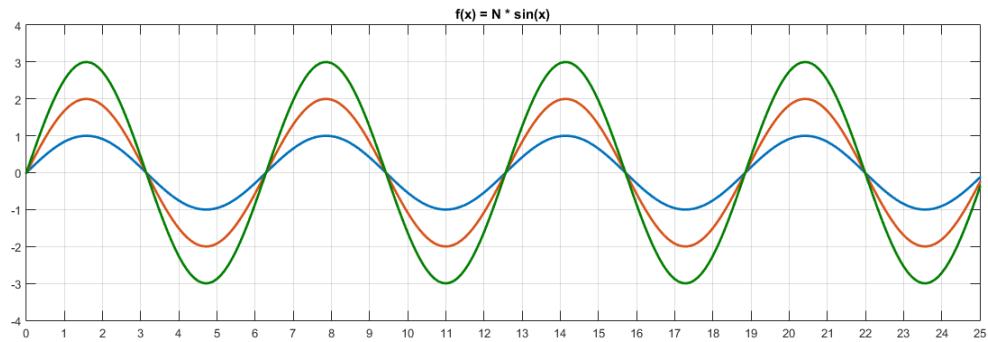


Figura 3.7: Señales con diferentes amplitudes

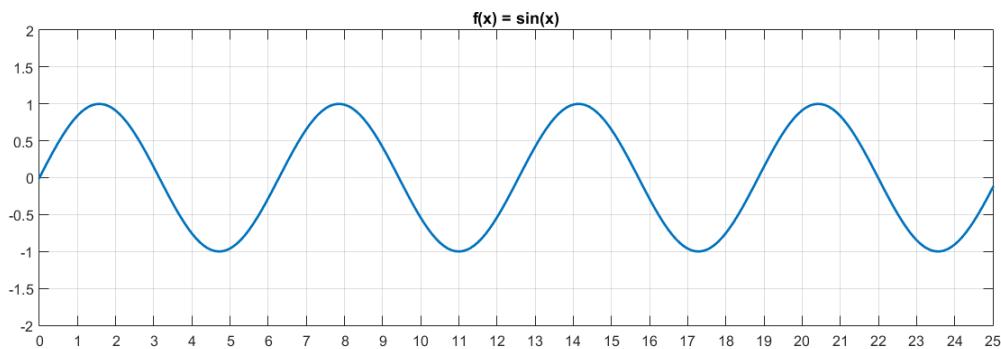


Figura 3.8: Señal con periodo definido

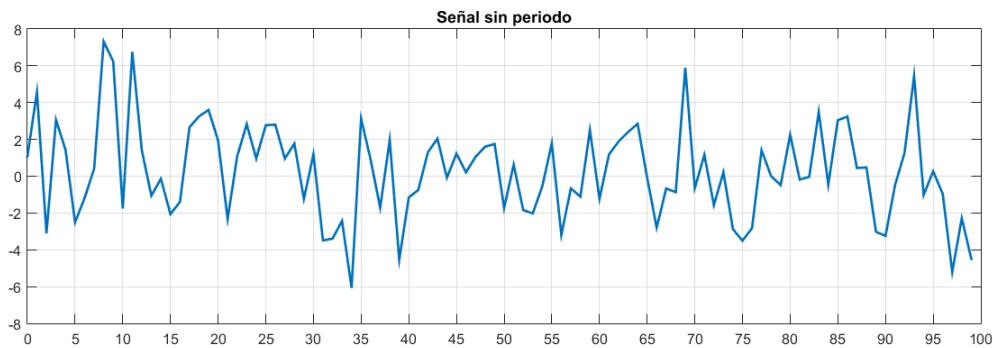


Figura 3.9: Señal con periodo no definido

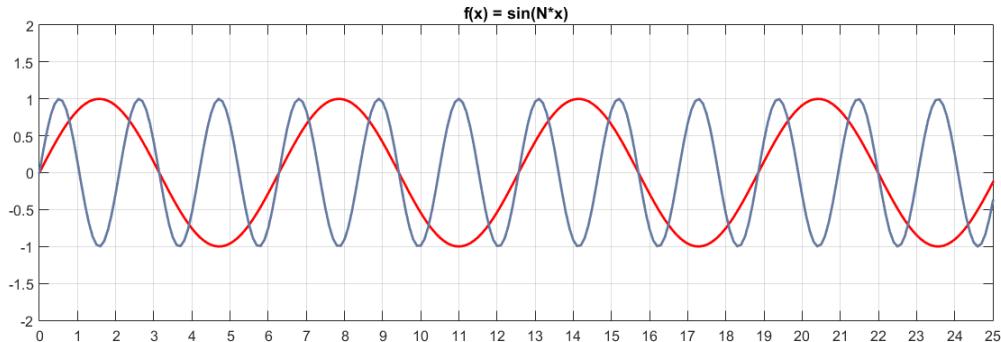


Figura 3.10: Señales con diferentes frecuencias

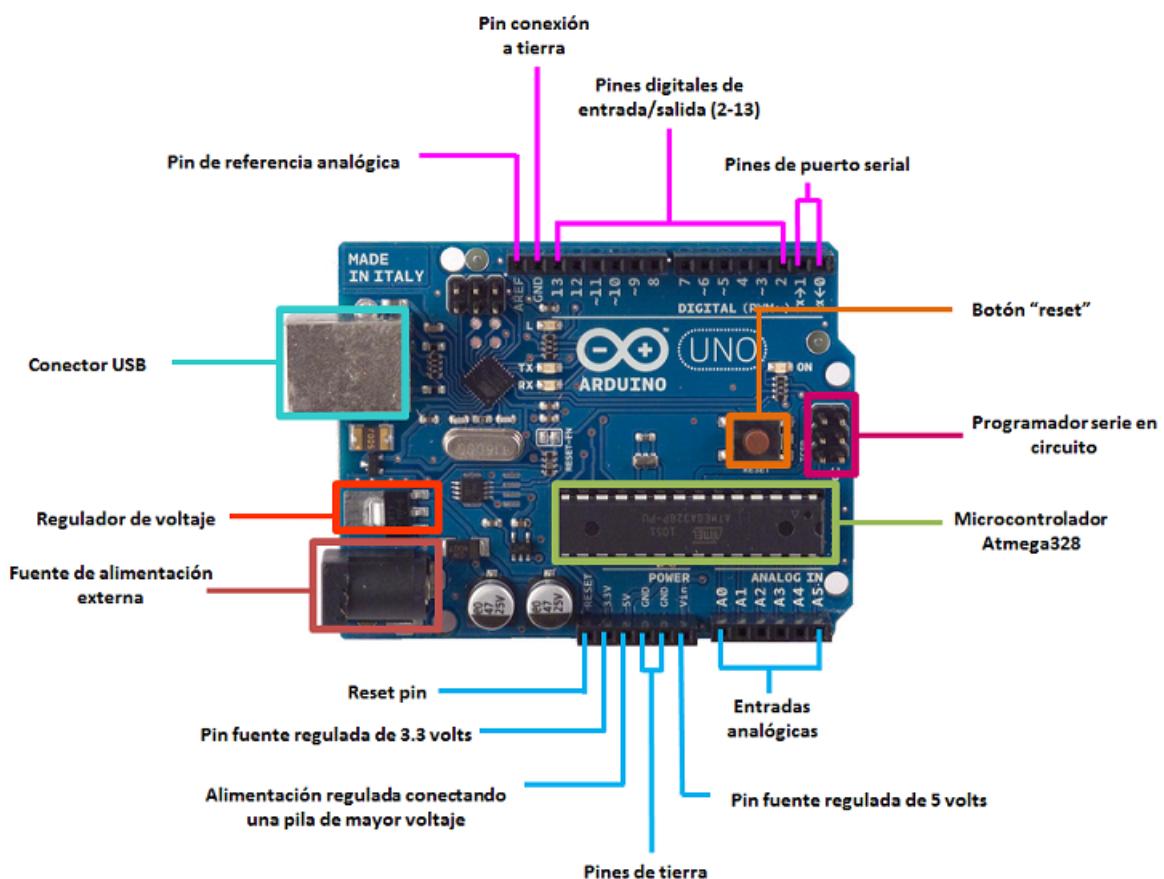
3.2. Arduino

Arduino es un conjunto de componentes electrónicos entre los que incluye un microcontrolador, pines de entrada/salida y un puerto serial, también se le conoce como *placa de desarrollo*. Su microcontrolador es útil en proyectos que requieran de procesar instrucciones sencillas, por ejemplo juguetes a control remoto, televisiones, controladores de clima automáticos, hornos de microondas. Por otro lado sus pines permiten la conexión de sensores que interactúen con el exterior y de esta manera captén lo que se necesiten para un proyecto. Las placas arduino son *open-source* o libres, esto quiere decir que su fabricación no está limitada a una sola marca, persona o empresa. Son multiplataforma llegando a funcionar tanto en Windows como en Linux y en Mac, requieren de una corriente de 5 Volts para operar la cual pueden obtener directamente desde su puerto USB. Éstas placas cuestan alrededor de 600 MXN en su versión original y alrededor de 230 MXN en una versión genérica.

3.2.1. Microcontrolador Atmega328

Es un circuito integrado con la capacidad de almacenar instrucciones en su memoria para luego ejecutarlas, éstas instrucciones se pueden borrar una y otra vez, un programador¹ puede definir las instrucciones en una computadora mediante un editor

¹Persona que desarrolla software.

Figura 3.11: Placa arduino, modelo *UNO*

de texto, en el caso de arduino se cuenta con un Entorno de Desarrollo Integrado o IDE por sus siglas en inglés² que está disponible para su descarga de manera gratuita en el sitio web de arduino [12], es multiplataforma y sencillo de utilizar. Dentro del IDE las instrucciones son escritas en el Lenguaje de programación de Arduino [13], el cual provee funciones para controlar los pines, para enviar un valor como salida por medio del puerto serial, para generar números aleatorios, entre otras.

Una vez que las instrucciones están definidas se procede a grabarlas en el microcontrolador utilizando el IDE, para esto es necesario conectar la placa arduino a la computadora utilizando el conector USB tanto de la placa como de la computadora.

3.2.2. Pines de entrada/salida

Se tratan de conectores parecidos a un enchufe que interactúan con el exterior y que pueden funcionar como *entrada* captando datos desde otro componente conectado a dichos pines, o también pueden funcionar como *salida* dando la posibilidad de enviar datos a través de ellos, los datos pueden ser de tipo analógico o de tipo digital. Dentro del lenguaje de programación de arduino se encuentran instrucciones para configurar los pines modificando su comportamiento ya sea de entrada o salida, también existen instrucciones para leer desde los pines o escribir hacia ellos.

3.2.3. Conector USB

Todas las placas arduino contienen un puerto *serial* el cual se encarga de la comunicación con la computadora, en un puerto serial los datos fluyen entre un puerto y otro en función del tiempo de manera que los bits se transmiten de forma independiente uno detrás de otro. Arduino utiliza un puerto serial con el protocolo de comunicación conocido como UART ³. Éste utiliza una línea (o un cable) para enviar información

²Integrated Development Environment.

³Universal Asynchronous Receiver-Transmitter

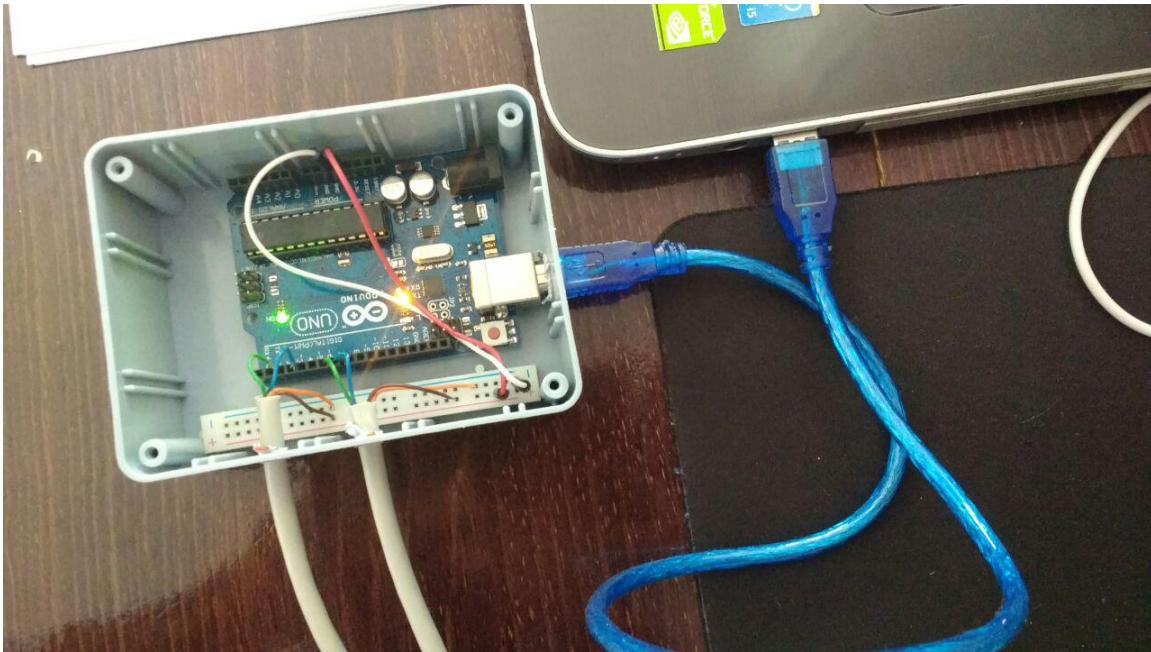


Figura 3.12: Placa arduino UNO conectada a una computadora

y otro diferente para recibirla, sin embargo en sus siglas define la palabra *asíncrono*, esto significa que UART no envía datos en función del tiempo sino que los envía por bloques, para asegurarse de que no se pierda información entre los bloques se utiliza un bit extra de información que se conoce como *bit de paridad*, este dato debe coincidir entre ambos puertos que se están comunicando.

El protocolo UART puede enviar varios bits al mismo tiempo pero depende de que ambos puertos estén configurados para enviar y recibir información a la misma velocidad, este parámetro se conoce como *baud rate*.

3.3. HC-SR04

Es un sensor que basa su funcionamiento en la emisión de ondas sonoras para detectar los objetos que tenga enfrente, contiene un par de transductores; uno emisor y otro receptor, que son dispositivos capaces de transformar las señales que reciben, un ejemplo cotidiano son las bocinas que utilizan los reproductores de música ya que



Figura 3.13: Sensor HC-SR04

convierten las frecuencias escritas en una canción no importa que esté almacenada en un formato físico o en uno digital hacia sonido que las personas pueden escuchar, los megáfonos tambien son un buen ejemplo, una persona puede hablar por estas cosas y se le escuchará hasta varios metros de distancia; las ondas sonoras que conformaban su voz fueron amplificadas por un transductor para hacerlas sonar más fuerte.

El sensor HC-SR04 emite a través de uno de sus transductores ondas de sonido ultrasónicas, esto quiere decir que el oído humano no puede percibirlas ya que tienen una frecuencia de 40 KHz, el oído humano sólo puede escuchar frecuencias hasta los 20 KHz. Su trabajo se realiza de la siguiente manera:

1. El transductor emisor emite una onda sonora ultrasónica, se marca el tiempo en que fue emitida.
2. Un objeto se interpone el camino de esas ondas lo que provoca que la onda sea devuelta (se refleja, rebota) hacia donde se originó.
3. El transductor receptor percibe la onda sonora rebotada y se calcula el *tiempo* que la onda sonora tardó desde ser emitida hasta ser captada de vuelta a su origen. Este tiempo es el resultado devuelto por el sensor.

El sensor HC-SR04 se comunica a través de 4 pines:

VCC	Es el que recibe la energía de alimentación (5 Volts) o pin positivo.
Trig	Entrada que recibe la señal para iniciar la emisión de ondas sonoras.
Echo	Pin que devuelve el tiempo que la onda sonora viajó, es decir el resultado.
GND	Es el pin de tierra o negativo.

Cuadro 3.2: Funciones de los pines del sensor HC-SR04



Figura 3.14: Esquema del funcionamiento del HC-SR04

3.4. MATLAB

Es una abreviación para MATrix LABoratory y es un software diseñado para llevar a cabo el cálculo de operaciones matemáticas de forma sencilla y utilizando un lenguaje de programación (llamado lenguaje M). Además permite el procesamiento de datos mediante algoritmos ya que su lenguaje es de tipo *script*, en este tipo de lenguajes cada línea de código que se escriba se interpreta y se ejecuta antes de la siguiente línea, MATLAB se puede utilizar para el análisis de datos, procesamiento de señales, visión artificial, robótica, finanzas, sistemas de control. Es multiplataforma y de licencia propietaria.

El entorno de desarrollo de esta herramienta permite trabajar sobre una consola que interpreta comandos, archivos de scripts; que son archivos de texto donde cada línea de instrucciones de ejecuta tras otra como si se escribieran una por una en la consola. También existen archivos de funciones que a diferencia de los scripts pueden recibir

parámetros de entrada y devolver parámetros de salida.

$$f(x) = 2x \quad (3.3)$$

En la ecuación 3.3 el parámetro de entrada es la x encerrada entre paréntesis, mientras que el parámetro de salida es el resultado de operar $2 * x$, es así como también funcionan los archivos de funciones que reciben parámetros de MATLAB. Dentro del entorno tambien se encuentran herramientas para explorar archivos, para ver el historial de comandos y se cuenta con un *workspace* que básicamente es un espacio donde se muestran los valores que se han asignado a variables y los resultados de operaciones realizadas en la consola, en el apéndice «C» se puede apreciar la pantalla principal de MATLAB.

Otra de sus cualidades es la creación de gráficas de manera sencilla y sus múltiples opciones para darles formato, provee también un constructor de interfaces gráficas el cual evita todo el trabajo de crearlas a partir de líneas y líneas de código.

3.4.1. MATLAB Runtime

Se trata de un conjunto de librerías esenciales que permiten para la ejecución de aplicaciones MATLAB compiladas en computadoras que no cuentan con una instalación completa de MATLAB. Utilizando el compilador de aplicaciones todos los scripts o archivos utilizados de un proyecto quedan en un solo lugar y son empaquetados generando una versión ejecutable al instante y una versión instalable. El compilador de aplicaciones se ha utilizado en Windows por lo que sus resultados funcionan en este sistema operativo, no se sabe si existe el compilador en las versiones de MATLAB para otros sistemas operativos.

El Runtime puede se puede descargar de forma separada o se puede incluir durante la compilación del proyecto, ésta segunda opción hará que al resultado final se le

sumen alrededor de 600 mb pero es útil si no se llega a contar con una conexión a Internet. De esta forma un proyecto se puede compartir, ejecutar e incluso instalar en una computadora distinta a la que fue programado.

3.5. Git

Es un software de libre distribución y de código abierto [14] cuyo uso permite controlar los cambios que vayan realizando en un proyecto de software, fue desarrollado por Linus Torvalds. El flujo de trabajo en Git se basa en el *seguimiento* de archivos, se utiliza bajo una línea de comandos, cada acción que el usuario quiera realizar se debe escribir en un entorno previamente configurado. Existen programas con interfaz gráfica que facilitan éstas acciones y en lugar de escribir los comandos se pulsan botones o se arrastran archivos.

Cuando Git tiene el seguimiento de un archivo puede detectar si ha sido editado, cambiado de nombre o incluso eliminado, de esta forma el programador sabe que archivos han sido modificados y le permiten confirmar una nueva versión, Git puede recordar todas las cambios confirmados hechos en los archivos que está siguiendo y puede restaurarlos a su estado original aunque hayan pasado 100 versiones.

Otra de las ventajas de Git es el desarrollo paralelo de software, por ejemplo un programador puede trabajar en una función que lea un archivo de entrada, mientras que otro trabaja en una función que le muestre en pantalla al usuario una calculadora, una vez terminado el trabajo Git se puede encargar de mezclar ambas funciones en una misma versión del proyecto, esto se conoce como *ramas*.

3.5.1. Github

En Internet, Github [15] es un sitio web donde se almacenan proyectos de software cuyo sistema de control de versiones es Git. El proceso consta de crear un perfil

o cuenta personal en Github para lo cual es necesario una dirección de correo electrónico, una vez dentro del perfil personal se pueden crear *repositorios* dándoles un nombre que los identifique, por ejemplo puede ser el nombre del proyecto. Un repositorio en Github equivale a una carpeta en una computadora, los archivos que Git esté siguiendo dentro de un directorio de trabajo local pueden subirse a Github con el comando adecuado. Con esto se gana una copia del software disponible a través de Internet que se puede descargar desde cualquier parte, ya sea desde el navegador web o desde la línea de comandos de Git. El repositorio sólo puede recibir más versiones del software por parte de la persona que lo creó, a menos que él otorgue permisos a otro usuario dentro de la configuración del repositorio.

Almacenar un repositorio en Github brinda ciertas posibilidades; una de ellas se menciona con anterioridad, esa copia puede servir como respaldo en el extremo caso de que los archivos originales se pierdan o se dañe la computadora con la que se estaba trabajando. Otra posibilidad es el alcance que se gana, al estar accesible a todo público un repositorio puede obtener colaboraciones de otros desarrolladores de software, un repositorio en Github puede crecer de las siguientes maneras:

1. Programadores que trabajen en equipo y suban versiones del software al mismo repositorio, se conocen como *colaboradores*.
2. *Fork*, es un clon del repositorio que otro usuario de Github realiza pero hacia su perfil personal, de este modo puede hacer cambios en el proyecto sin alterar el original.
3. Los usuarios que hacen *fork* de un proyecto tienen la posibilidad de preguntar si sus cambios hechos pueden ser mezclados con el repositorio original, esto se conoce como *pull request* y sólo el autor original del repositorio puede autorizarlo.

Capítulo 4

Prototipo de un software para graficar señales analógicas

4.1. Programación de arduino

Utilizando el IDE de arduino se escribió un programa que leyera el valor que llegaba a uno de sus pines analógicos y transmitiera el resultado a través del puerto serial, la finalidad de este ejercicio fue abrir las puertas del módulo de adquisición de datos. Captar la señal de la corriente que genera uno de los pines de salida del arduino hubiera sido suficiente, ésta señal se dibujaría en pantalla como una línea continua con un mismo valor todo el tiempo (5 volts), sin embargo las curvas no tienen esa característica sino que van cambiando, así que se utilizó un potenciómetro el cual es un dispositivo electrónico capaz de regular la salida de la corriente que se le ingresa dándole una mayor amplitud o una menor, con esto fue posible observar los cambios que se le aplicaban a una corriente y simular los cambios que va teniendo una curva en un RGP.

Algoritmo 4.1 Primer programa de arduino, lee una entrada analógica y la transmite por serial.

```
void setup(){
    Serial.begin(9600);
}
void loop(){
    Serial.println(analogRead(0)); //número de pin a leer del arduino
    delay(100);
}
```

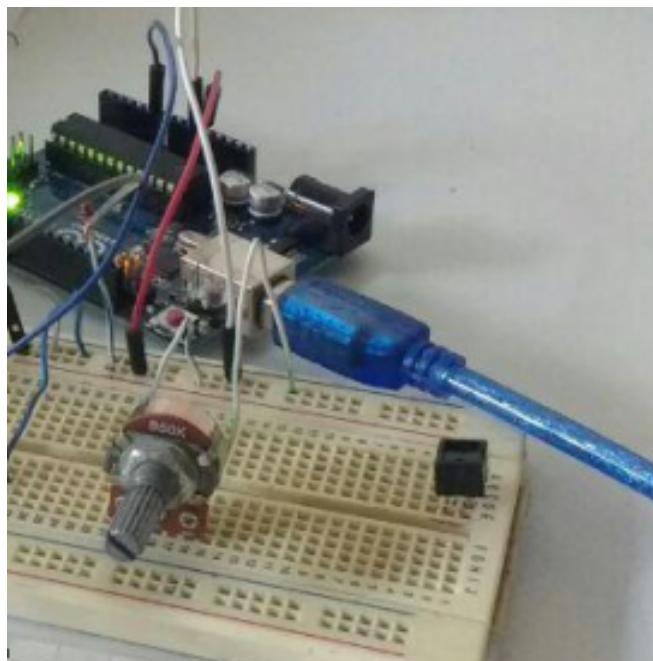


Figura 4.1: Potenciómetro conectado, guía encontrada en [16].

Algoritmo 4.2 Funciones de MATLAB para manejar el puerto serial.

```
placa = serial('COM3'); %variable que guarda la conexión serial
fopen(placa);           %abre la conexión
fscanf(placa, '%f');    %lee desde el puerto serial
fclose(placa);          %cierra la conexión serial
```

4.2. Programación de MATLAB

Ésta herramienta cuenta con las funciones necesarias para recibir datos desde un puerto serial, es necesario crear una conexión hacia el puerto y al terminar el trabajo cerrar la misma, esto es posible con tres sencillas instrucciones mostradas en el algoritmo 4.2. La primera función es un objeto que guarda los parámetros de una conexión serial, el parámetro que recibe le indica a MATLAB el puerto serial a utilizar. El nombre del puerto serial depende del sistema operativo, en Windows el puerto por default es *COM3*, pero si ya hubiera un dispositivo utilizando ese puerto el número aumentará siendo ahora *COM4*, la estructura del objeto se muestra en el algoritmo 4.3.

Por su parte la función **fscanf** recibe dos argumentos, el primero es el objeto que guarda los parámetros de la conexión y el segundo es para darle un formato a los datos leídos, en este caso se utilizó el formato de punto flotante que permite almacenar números con punto decimal. La convención utilizada para dar formato a los datos viene del lenguaje de programación C¹.

Una vez leído el valor del puerto serial se procedió a mostrarlo en una gráfica, el movimiento del potenciómetro se veía reflejado en el dibujo de la gráfica. La figura 4.2 representa una serie de 100 muestras.

¹véase el cuadro en el apéndice «D» para la lista completa de formatos admitidos por C.

Algoritmo 4.3 Estructura de un objeto serial de MATLAB

```
Serial Port Object : Serial-COM3
```

```
Communication Settings
```

```
Port: COM3
```

```
BaudRate: 9600
```

```
Terminator: 'CR'
```

```
Communication State
```

```
Status: closed
```

```
RecordStatus: off
```

```
Read/Write State
```

```
TransferStatus: idle
```

```
BytesAvailable: 0
```

```
ValuesReceived: 0
```

```
ValuesSent: 0
```

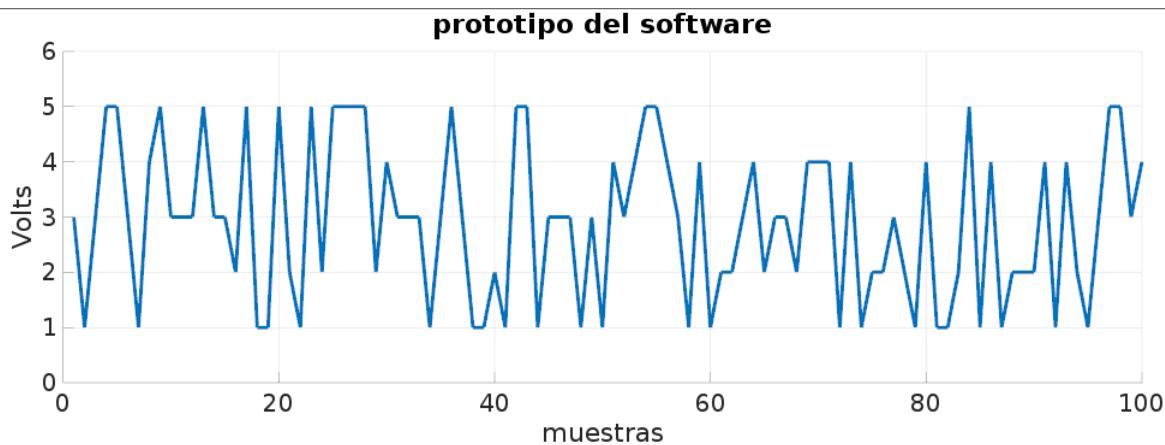


Figura 4.2: Gráfica producida por el prototipo de software

Capítulo 5

Análisis de la sonda del CIIDIR

5.1. Búsqueda de la señal

Con el objetivo de obtener las señales que la sonda produce se le hizo un análisis que consistió en abrirla y observar sus componentes electrónicos, el primer acercamiento fue captar la corriente que llegaba a los potenciómetros que la sonda utiliza para normalizar la señal. Cada curva se controla con un par de potenciómetros que regula el dibujo de la curva, uno de ellos controla el valor mínimo que se puede registrar sin salirse de la zona de dibujo y se llama *zero*, mientras que el potenciómetro de *span* controla el valor máximo o amplitud que la curva puede llegar a alcanzar. Una manera de exemplificar el *span* es observando la gráfica de una señal que tenga variaciones muy pequeñas. La figura 5.3 muestra una señal con valores menores a 10 que están en función de 100 puntos, dado el límite del eje de las ordenadas que la gráfica posee el cual va de 0 a 100 no se puede apreciar con claridad el valor exacto de cada uno. Mientras tanto en la figura 5.4 los límites de graficación están del 0 al 10, esto permite observar que los puntos se encuentran entre 3 y 6 proporcionando una mejor perspectiva de los datos, esto es lo que se conoce como *span*, vulgarmente de puede decir que es como aplicar un zoom en la gráfica.



Figura 5.1: Potenciómetros de la sonda (vista frontal)



Figura 5.2: Potenciómetros de la sonda (vista trasera)

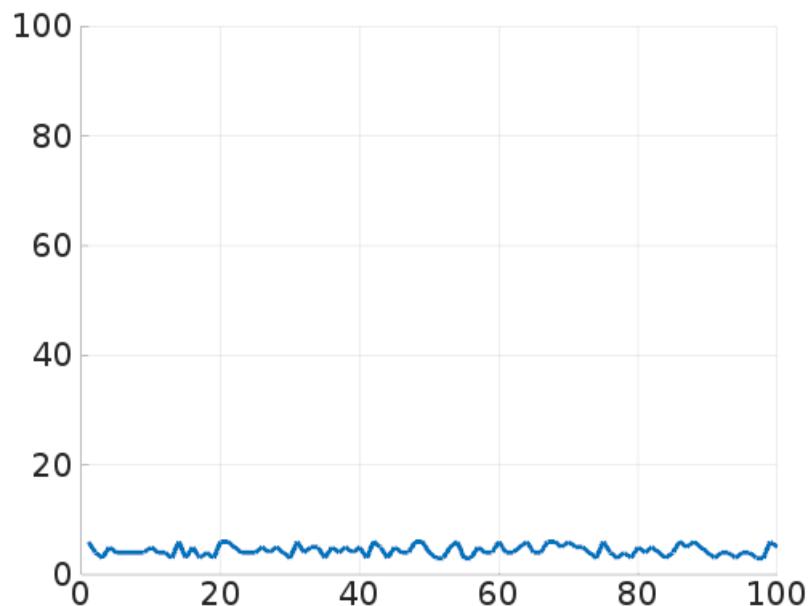


Figura 5.3: Ejemplo 1 de span

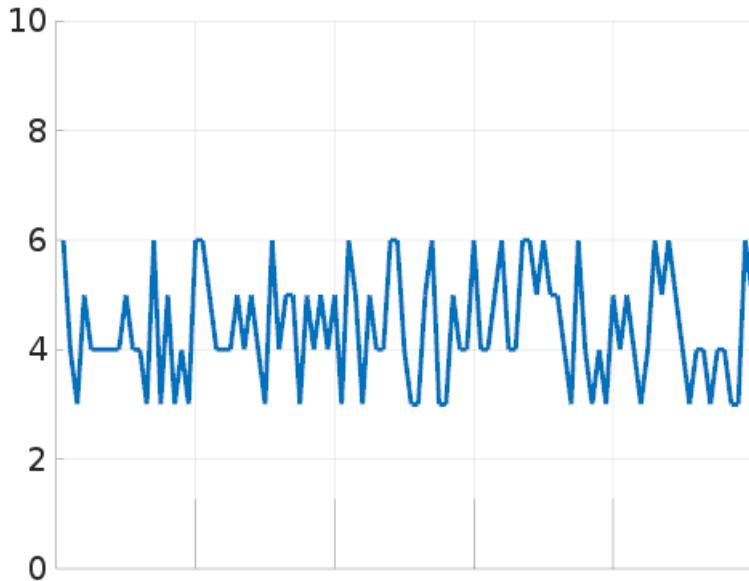


Figura 5.4: Ejemplo 2 de span

Con ayuda de conectores tipo caimán (figura 5.5) se hizo un puente entre los pines de un potenciómetro que permitiera conducir la corriente que llegaba a ellos y enviarla al arduino, tal como se esperaba el software mostraba variaciones en el dibujo de las gráficas cuando una persona movía el potenciómetro justo como en el primer ejercicio del capítulo 4, sin embargo estas mediciones no correspondían a la curva real, ya que las personas que llevan a cabo el RGP y trabajan con la sonda notificaron que el dibujo de la gráfica lo realiza un dispositivo llamado *pluma*, se muestra su fotografía en la figura 5.6. En esta pequeña especie de pinza roja se monta un plumón el cual va dibujando en el papel tal como la pinza se vaya moviendo, por lo tanto medir la corriente que llega a los potenciómetros no fue satisfactorio, así que se tomó el camino de analizar el movimiento de la pluma.

5.1.1. Análisis del movimiento de la pluma

Al explorar la sonda más a fondo se descubrió que el movimiento de las plumas está ligado a los siguientes pasos:

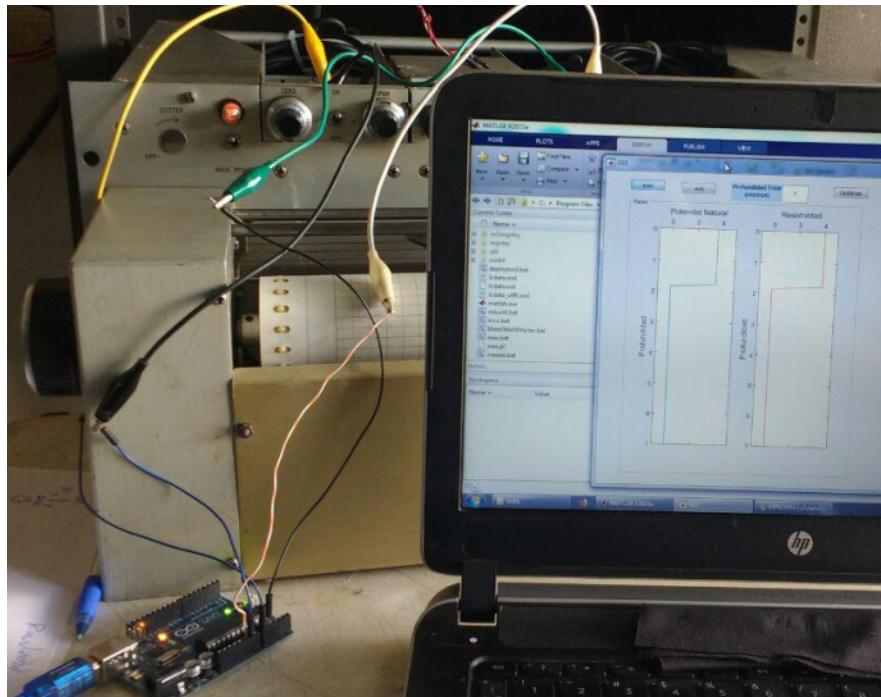


Figura 5.5: Puente de potenciómetro hacia arduino utilizando caimanes

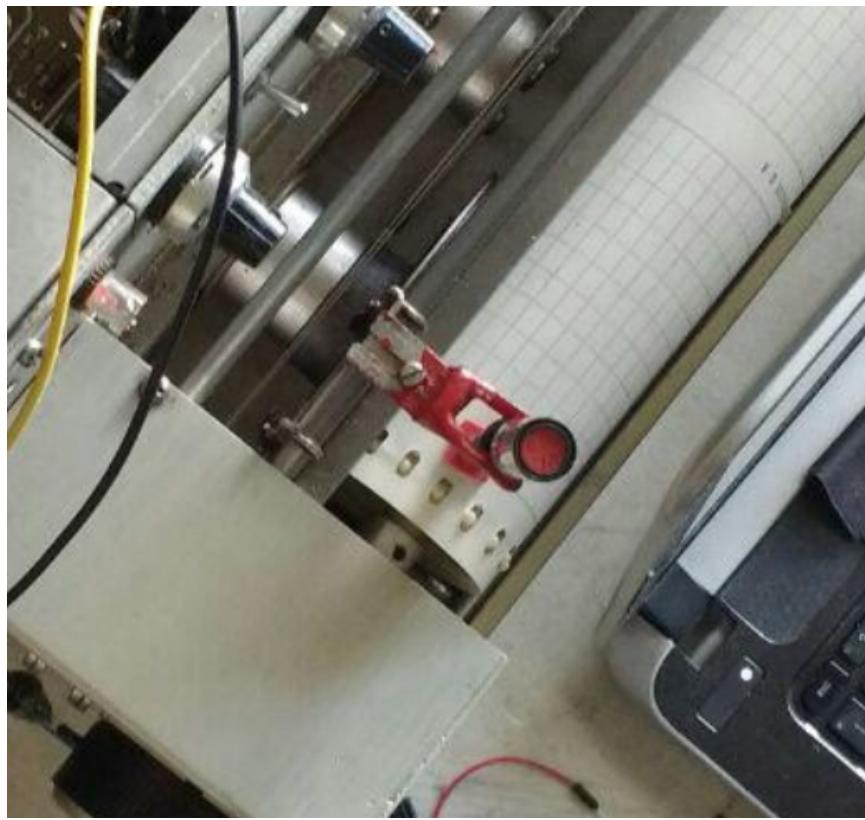


Figura 5.6: Pluma de la sonda



Figura 5.7: Discos, alambre fino y la pluma montada sobre ellos.

1. Una corriente eléctrica llega a un motor ubicado en la parte trasera de la sonda.
2. El motor hace girar unos discos de metal, ya sea a la izquierda o a la derecha.
3. Los discos están envueltos con un alambre muy fino de manera que cuando giran hacen que el alambre también se mueva.
4. Sobre el alambre existe una estructura en la cual se montan las plumas y por lo tanto el movimiento se encadena del motor a los discos, de los discos al alambre y de éste último a las plumas.

5.1.2. Conclusión

Después de analizar la situación con ayuda de los operadores de la sonda cuyos nombres son Valdemar y Horacio se decidió utilizar sensores que fueran capaces de captar el movimiento de las plumas, ya que buscar la señal dentro de la estructura electrónica de la sonda sería un camino mucho más largo y complejo.

Capítulo 6

Sensores de movimiento

Se pensó en una forma de capturar el movimiento de las plumas tal que si se movían a la izquierda el software dibujaría una desviación hacia la izquierda, lo mismo si la pluma se moviera hacia la derecha. La idea era montar sensores en alguna parte de la sonda para que se pudiera registrar ese movimiento, la placa arduino puede trabajar con sensores que detecten la proximidad así que sólo había que encontrar el sensor adecuado.

6.1. CNY70

La reflexión de la luz es un fenómeno que tiene distintos comportamientos de acuerdo a la superficie donde ocurra, un metal pulido sobre el que se incide luz provocará una reflexión diferente que si se incide luz sobre un metal oxidado, es por eso que existen algunos proyectos [17] que utilizan este sensor para detectar colores, sin embargo se encontró un ejemplo para medir distancias [18]. Era necesario trabajar detectando la distancia ya que la idea principal consistió en colocar los sensores de tal manera que registraran la distancia desde la pluma hasta algún objeto fijo de la sonda, así se iría registrando el movimiento de las plumas el cual quedaría plasmado en el dibujo de las gráficas. Las características del sensor CNY70 son las siguientes:

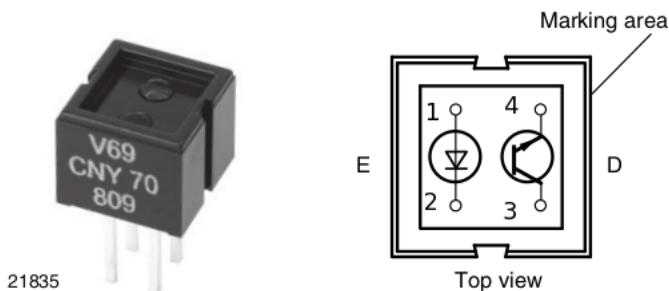


Figura 6.1: Sensor CNY70

- Led infrarrojo, responsable de la emisión de luz.
- Foto-transistor, responsable de captar la luz reflejada.
- 4 pines, numerados de la siguiente manera:
 - 1: Alimentación del led
 - 2: Tierra de led
 - 3: Alimentación del transistor (en su parte Colector)
 - 4: Salida

Funciona emitiendo un haz de luz infrarroja que el transistor percibe cuando m^Ves reflejado por alguna superficie, de este modo devuelve un valor a través del pin 4 en la escala de milivoltios (mV). En superficies que reflejan mucha luz el valor devuelto será alto (llegando a los 100 mV), mientras que para una superficie negra devolverá cerca de 50 mV. Así que se realizó la conexión de ambos sensores utilizando una protoboard¹, se modificó el programa en arduino para que leyera dos entradas desde sus pines analógicos y las transmitiera por el puerto serial. Se utilizó también una parte del papel de la sonda para tener una referencia en cuanto a las distancias medidas, la primera prueba se llevó a cabo colocando superficies cubiertas con cinta de aislar negra simulando los extremos de la sonda donde va puesto el rollo de papel.

¹Es una placa plástica que permite la comunicación de circuitos electrónicos sin utilizar soldadura

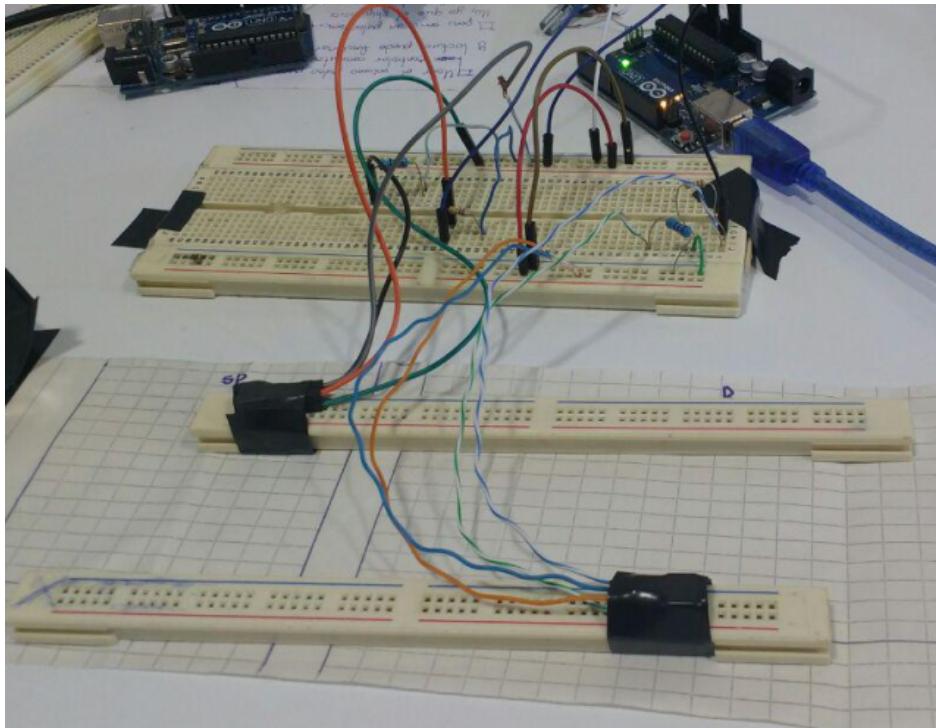


Figura 6.2: Prueba en laboratorio de los sensores CNY70

Las tiras que se observan en la figura 6.2 son un acercamiento a las plumas, se requería de una forma de imitar el movimiento de las plumas así que los sensores se montaron sobre de dichas tiras.

Los resultados fueron buenos, las gráficas en pantalla dibujaban el movimiento de las tiras, así que se llevó el prototipo del módulo de adquisición de datos a la sonda para instalarlo y realizarle pruebas con movimiento real. Utilizando cinchos de plástico se sujetaron los sensores a cada una de las plumas y se pegó cinta de aislar negra en la cavidad que soporta al rollo de papel, de este modo los sensores captarían la distancia desde la pluma hacia el extremo de la cavidad.

Sin embargo se notaron varios inconvenientes:

1. Al estar en constante movimiento los cables que conectan a los sensores serían un problema, una mejor opción sería dejar los sensores quietos.
2. Las variaciones de luz ambiental afectaban las lecturas que realizaban los sen-

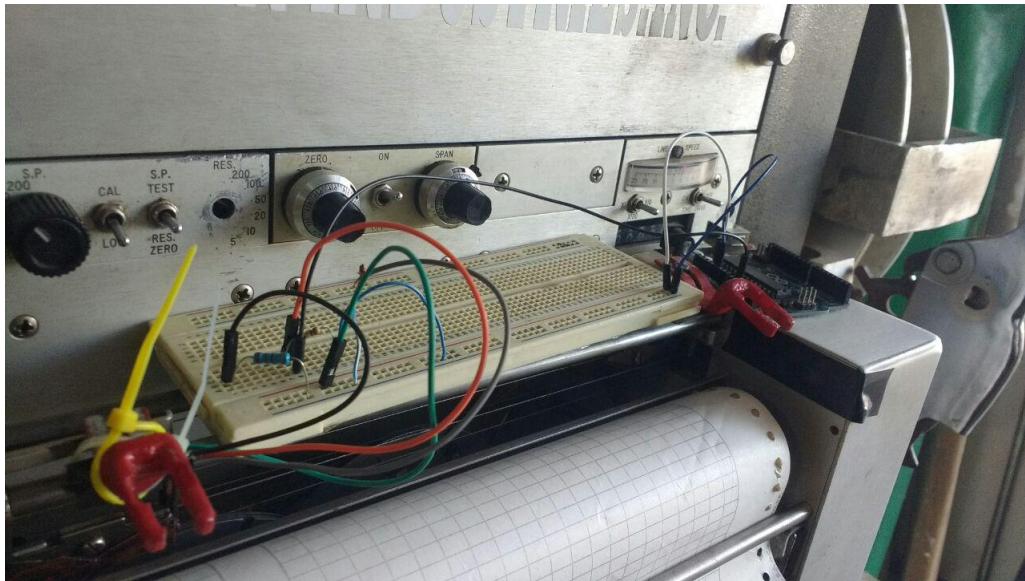


Figura 6.3: Prueba en la sonda de los sensores CNY70

sores. La incidencia indirecta de otra fuente de luz provocaba que los sensores devolvieran datos fuera del rango establecido. (más allá de los 100 mV).

Por lo tanto no fueron adecuados para realizar el trabajo ya que los trabajos de RGP se llevan a cabo en el exterior donde la luz ambiental suele variar mucho, así que se probó otro tipo de sensor.

6.2. HC-SR04

El funcionamiento y la teoría de este sensor quedaron descritos en la sección 3.3 del capítulo 3, se trabajó con un par al igual que en el experimento anterior y se modificó el código arduino para interpretar los resultados devueltos por el sensor. Se encontró que devuelve un tiempo en microsegundos², existe una ecuación que relaciona la distancia y el tiempo; es la de la *velocidad*:

$$v = \frac{d}{t} \quad (6.1)$$

²Prefijo cuyo símbolo es μ , un microsegundo son 1×10^{-6} segundos, es la millonésima parte de un segundo.

Dado que el sensor utiliza ondas ultrasónicas es posible conocer su velocidad, la cual equivale a la velocidad del sonido y es $343 \frac{m}{s}$, no hay que confundir ultrasónico con la velocidad del sonido, si bien están por encima de la frecuencia que los humanos pueden escuchar siguen siendo ondas sonoras y su velocidad de propagación por el aire es igual a la de las ondas sonoras que el oído humano puede escuchar. Sabiendo esto; sólo queda una variable a conocer en la ecuación 6.1 y es la distancia, así que se realiza un despeje quedando:

$$d = v * t \quad (6.2)$$

Como primer paso se realizó la conversión de la velocidad del sonido hacia su equivalente en $cm/\mu s$, con una regla de tres se resolvió este problema:

$$\frac{(343 m (1\mu s))}{1s} = 343 \times 10^{-3} m \quad (6.3)$$

Ésta es la *distancia* en metros que el sonido recorre en $1\mu s$, después se transformó esa longitud a centímetros, quedando:

$$\frac{343 \times 10^{-3} m (100 cm)}{1 m} = 0.0343 cm \quad (6.4)$$

Con este resultado final se supo que la velocidad del sonido equivale a $0.0343 cm/\mu s$ así que se procedió a utilizarla dentro del programa de arduino.

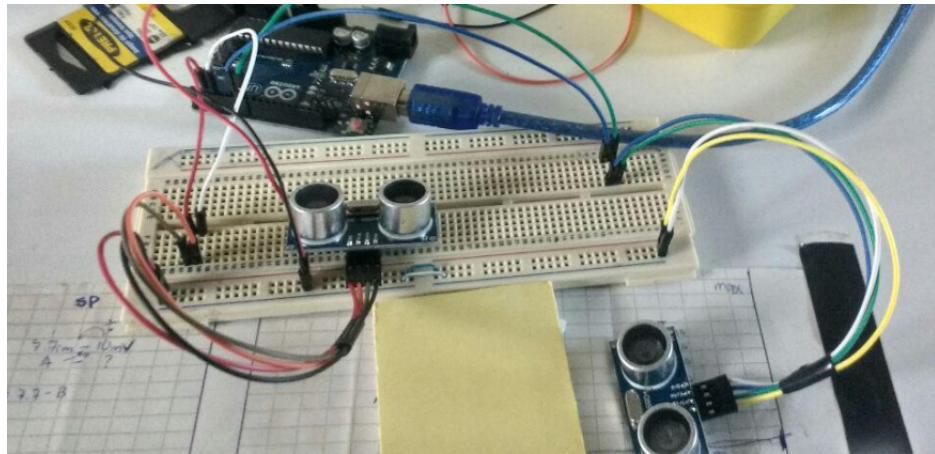


Figura 6.4: Sensores HC-SR04 conectados al arduino

6.2.1. Reprogramación de arduino

Lo primero fue definir algunas variables que indicaran el número de pin arduino a utilizar, se requieren dos para cada sensor, uno para el trigger y otro para el echo.

```
int echo = 2;  
int trigger = 3;  
int echo2 = 6;  
int trigger2 = 7;
```

Después se declara una variable que guarde el resultado de la ecuación 6.4 y otras dos que guarden los resultados finales, cada una va almacenar el resultado de un sensor.

```
double velocidadSonido = 0.0343;  
double v = 0.0;  
double r = 0.0;
```

Luego dentro del apartado de *setup* se escribió lo siguiente, en el cuadro 3.2 se mencionó que el pin de *trigger* emite las ondas así que se configura como salida mientras que el pin de *echo* recibe los resultados, por lo tanto se configura como entrada.

```
void setup() {
```

Algoritmo 6.1 Procesamiento de los datos en arduino

```

void loop() {
    digitalWrite(trigger, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigger, LOW);
    v = velocidadSonido * pulseIn(echo, HIGH) / 2;

    digitalWrite(trigger2, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigger2, LOW);
    r = velocidadSonido * pulseIn(echo2, HIGH) / 2;

    Serial.print(v);
    Serial.print(" ");
    Serial.println(r);
    delay(1000);
}

Serial.begin(9600); //indica la velocidad de baudios3 a utilizar
pinMode(echo, INPUT);
pinMode(trigger, OUTPUT);
pinMode(echo2, INPUT);
pinMode(trigger2, OUTPUT);

}

```

Por último y lo más importante, se declararon las instrucciones necesarias para el procesamiento de datos, primero la emisión de ondas con la función *digitalWrite*. Luego con la función *pulseIn* se obtiene el tiempo que las ondas sonoras tardaron en ir y venir. Esto quiere decir que devuelve el tiempo total que la onda estuvo viajando, si se calculara la distancia con este resultado se obtendría una distancia equivalente al doble de la que realmente separa al sensor del objeto. Pero la solución es simple y basta con dividir entre dos la distancia calculada.

³Velocidad con la que se transmiten los datos, debe coincidir con el objeto serial en MATLAB: [4.3]

El cálculo de v y de r está dado por la ecuación 6.2 la cual multiplica la velocidad del sonido por el tiempo que estuvo viajando con el agregado de una división entre 2 para compensar el problema que se mencionó en la página anterior de la distancia doble.

Finalmente se transmiten ambas distancias calculadas, v y r por medio del puerto serial con la instrucción *Serial.print*. La penúltima línea del algoritmo 6.1 es para pausar la ejecución del programa durante 1000 milisegundos, esto equivale a 1 segundo y quiere decir que cada distancia obtenida por los sensores se calcula cada segundo.

6.2.2. Reprogramación de MATLAB

Dentro de MATLAB se estableció la función *anita* mostada en el algoritmo 6.2 la cual recibe como parámetro de entrada el objeto que guarda la conexión serial hacia la placa arduino. MATLAB tiene una característica especial en sus funciones y es que permite la devolución de más de un parámetro como salida, esto es curioso ya que la mayoría de los lenguajes de programación sólo permiten regresar uno y sólo un parámetro, en la función *anita* los resultados quedan guardados en las variables *sensorIzq* y *sensorDer*, haciendo referencia a la colocación de los mismos; uno a la izquierda y otro a la derecha del rollo de papel de la sonda.

Dado que las funciones pueden devolver múltiples resultados es posible almacenarlos de la siguiente manera en MATLAB:

```
[izq, der] = anita(placa);
```

Después fue necesario hacer algunos ajustes al valor leído, se llevaron a cabo mediciones en la sonda y en el papel que utiliza determinando que las zonas de dibujo tienen una longitud de 7.7 cm para el potencial espontáneo y 15.2 cm para la resistividad, el primer ajuste consistió en eliminar una parte de la distancia leída la cual corresponde a una pequeña zona vacía existente entre la posición del sensor y el inicio del área

Algoritmo 6.2 Función *anita* programada en MATLAB

```
function [sensorIzq, sensorDer] = anita(placa)
    lectura = fscanf(placa, '%f');
    sensorIzq = lectura(1);
    sensorDer = lectura(2);
end
```

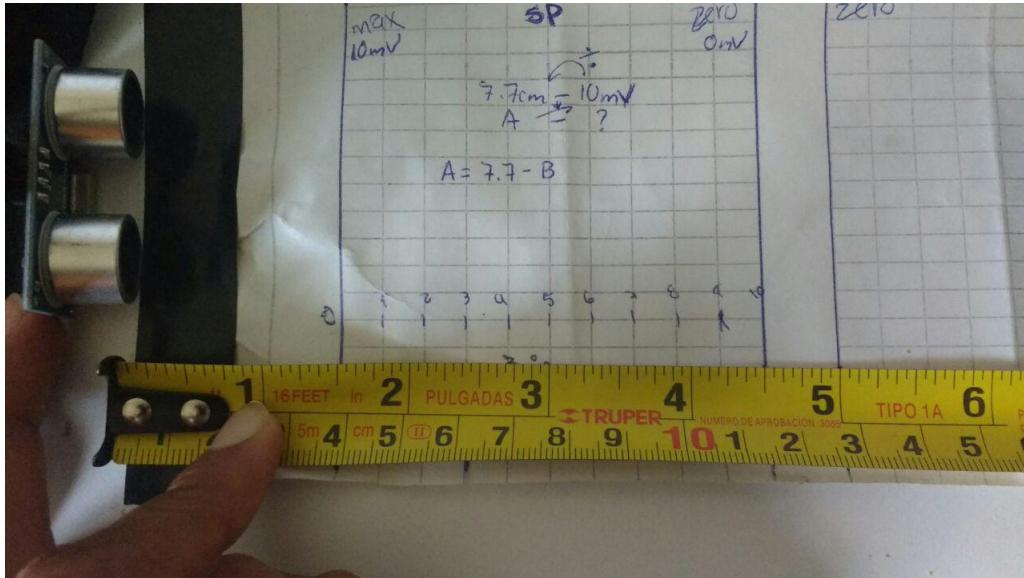


Figura 6.5: Midiendo las zonas de dibujo

para dibujar, consta de 4 cm y se muestra en el esquema de la figura 6.6.

Suponiendo que el punto central fuera la posición de la pluma el sensor estaría registrando la distancia \overleftrightarrow{B} más 4 cm extra. La curva además registra en el caso del potencial espontáneo los valores menores a la derecha y los mayores a la izquierda, esto quiere decir que el cero está en el límite lateral derecho y el máximo en el lado contrario. Este fue un problema ya que el sensor se ubicó en una zona donde la distancia que estaba midiendo ocurría de izquierda a derecha, totalmente al revés de como acomoda sus valores máximos y mínimos la curva.

Pero se encontró una solución sencilla cuyos pasos se enumeran a continuación:

1. Restar 4 cm a la distancia registrada por el sensor, con esto se obtiene el segmento \overleftrightarrow{B} .

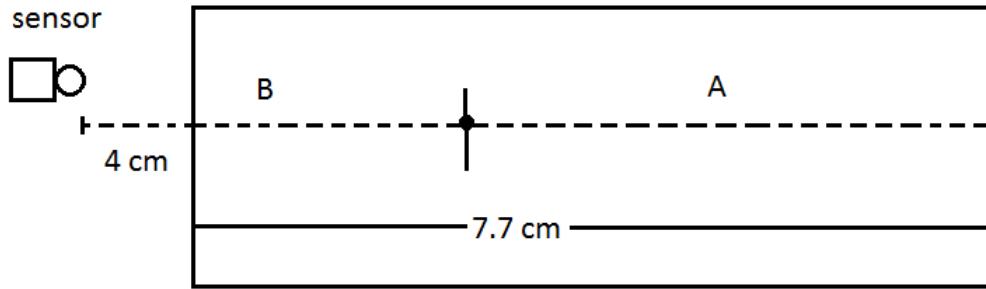


Figura 6.6: Esquema del área de dibujo del potencial espontáneo

2. Obtener el segmento \overleftrightarrow{A} el cual surge a partir de: $\overleftrightarrow{A} = 7.7 \text{ cm} - B$.
3. Utilizar esa distancia en la conversión hacia las unidades del potencial espontáneo.

Con estos simples pasos se logró obtener el valor de \overleftrightarrow{A} que va desde el extremo derecho hasta el supuesto punto central mostrado en la figura 6.6. Lo siguiente es convertirlo a sus unidades correspondientes en milivolts; donde la derecha corresponde a cero y la izquierda al valor máximo que puede variar según la escala o span mencionado en la sección 5.1.

$$? \text{ mV} = \frac{\overleftrightarrow{A} (10 \text{ mV})}{7.7 \text{ cm}} \quad (6.5)$$

Hay que tener en cuenta que el valor de 10 mV es un ejemplo ya corresponde al dado por el span y puede cambiar, se configuró el programa de MATLAB adecuadamente para que capturara este valor cada que cambiara dentro de la interfaz de usuario; los valores son ajustables tal como en el potenciómetro de la máquina y son: 5, 10, 20, 50 y 100 milivolts. Ésta solución fue implementada de igual manera para la curva de resistividad; con unas pequeñas variaciones ya que el máximo en ese caso está a la derecha y el mínimo a la izquierda.

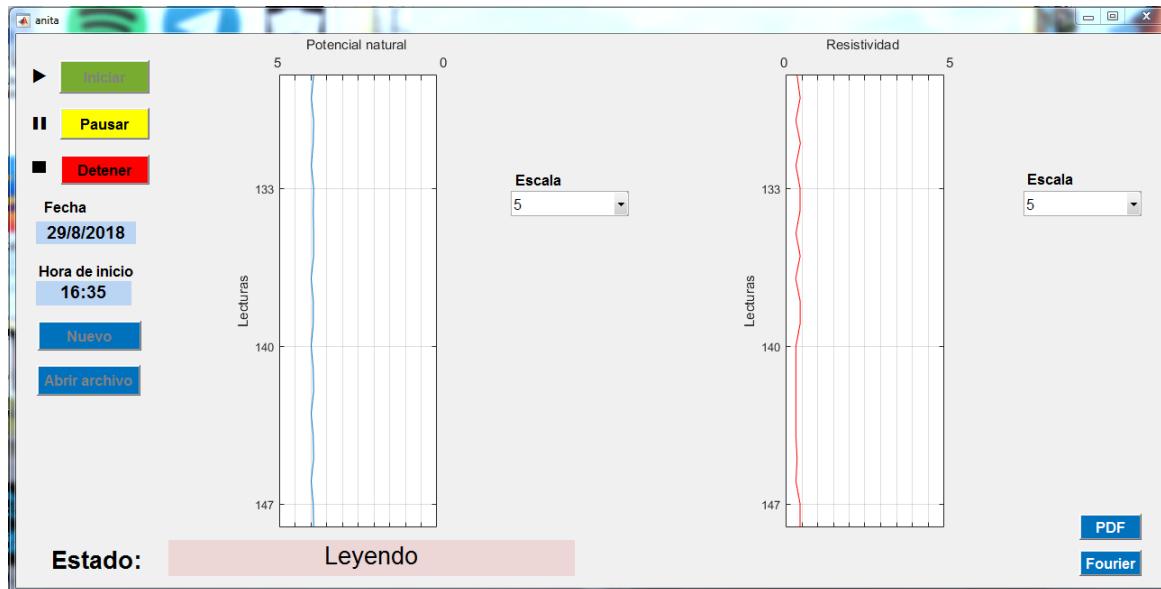


Figura 6.7: Interfaz del software usando los sensores HC-SR04

6.3. Resultados del sensor

A diferencia del CNY70 el sensor HC-SR04 brindó resultados con bastante precisión y no se veía alterado por factores ambientales, se utilizaron objetos planos en forma de tarjeta para simular el movimiento de las plumas y se concluyó que estos sensores se utilizarían para el desarrollo del módulo de adquisición de datos.

Capítulo 7

Módulo de adquisición de datos

Para eliminar el uso de la protoboard y comprimir todos los cables utilizados se procedió a buscar componentes que le dieran una mejor presentación al hardware. Fueron conseguidos en una tienda local [19] con el respaldo económico del CIIDIR.

7.1. Materiales

- ✓ Chasis plástico
- ✓ Cautin Steren
- ✓ Pasta para soldar Soldek (roja)
- ✓ Placa fenólica CL-003
- ✓ 2 metros de cable categoría 5
- ✓ 1 metro de soldadura
- ✓ Placa arduino UNO R3 genérica

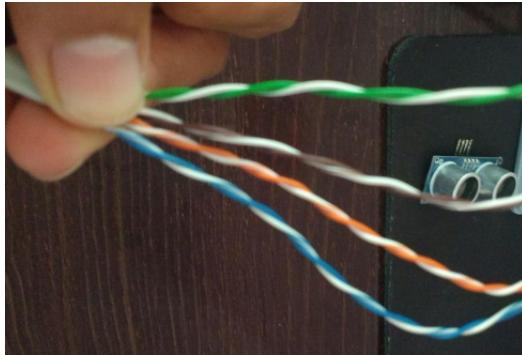


Figura 7.1: Hilos originales organizados en pares

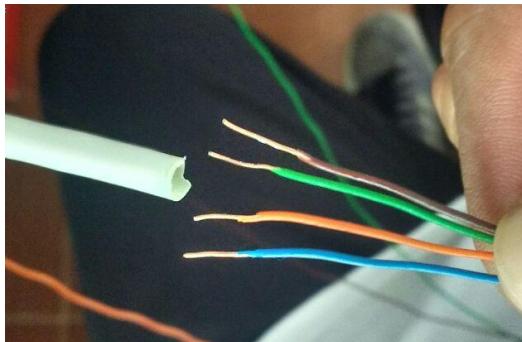


Figura 7.2: Hilos utilizados en el módulo

7.2. Construcción

El primer paso consistió en dejar sólo cuatro de los ocho hilos que tiene por dentro el cable categoría 5, de esta forma no estaría tan grueso y sería más fácil doblarlo y adaptarlo a la forma de la sonda. Para ello es necesario extraer los cuatro pares, desenredar los hilos a utilizar y volver a insertarlos en la funda aislante.

Después se soldaron en los pinos de ambos sensores las puntas de cada hilo utilizando un pedacito de placa fenólica, soldadura, pasta y el cautín, el color que corresponde a cada pin está anotado en el cuadro 7.1.

Mientras tanto en el chasis se realizaron tres agujeros que funcionaran como sujetadores de dos tramos de cable categoría 5 y del cable que utiliza la placa arduino para comunicarse con la computadora.

Dentro del chasis la placa arduino se fijó con pegamento y para la alimentación de

Color de cable	Pin del HC-SR04
Naranja	GND
Café	VCC
Verde	Echo
Azul	Trigger

Cuadro 7.1: Cableado de los sensores



Figura 7.3: Cables soldados al sensor

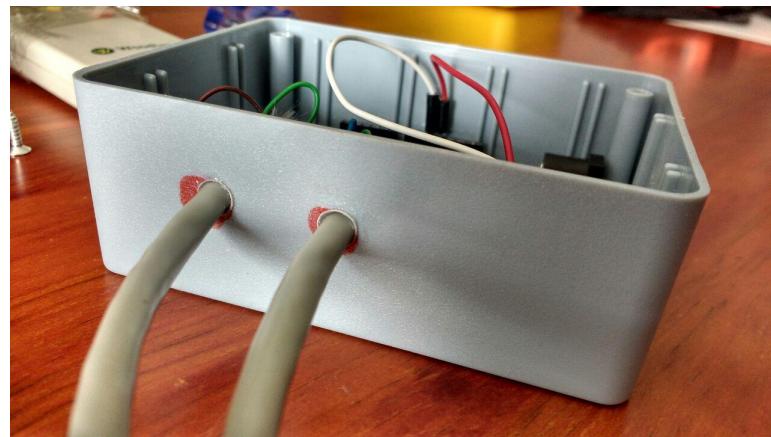


Figura 7.4: Cables para cada uno de los sensores



Figura 7.5: Agujero para el puerto serial del arduino



Figura 7.6: Interior del chasis

ambos sensores se utilizó una pequeña tira de protoboard de únicamente dos canales; a esta tira llegan la corriente de 5 volts producida por la placa así como su aterrizaje a tierra (figura 7.6).

Finalmente se construyó un par de cajitas con cartón que sirvieran de contenedores para cada sensor, el chasis se cierra utilizando tornillos y el resultado final queda en la figura 7.7.



Figura 7.7: Resultado final, módulo de adquisición de datos

Capítulo 8

Programación del software

El software desarrollado durante presente investigación se denomina *anita*, su finalidad es la visualización y procesado de señales geofísicas del subsuelo de la sonda explicada en el capítulo 5 utilizando para ello el módulo de adquisición de datos construido en el capítulo 7. Fue diseñado y programado en el entorno de desarrollo de MATLAB ya que los investigadores del CIIDIR tienen familiaridad, conocen esta herramienta y la utilizan a menudo. Se continúa el desarrollo a partir del prototipo de la sección 6.2.2; el cual ya contaba con la capacidad de dibujar las curvas correspondientes a dos valores recibidos desde los sensores HC-SR04 a través del puerto serial. Posee control de versiones usando Git y repositorio en Github [20] para su almacenamiento seguro.

8.1. Archivos CSV

Son archivos que guardan datos y separan cada uno mediante comas (,); de ahí el nombre en inglés *Comma-Separated Values*, este tipo de archivos se organiza en estructuras llamadas columnas y cada una puede agrupar a una categoría; es decir que pueden tener nombre.

Los datos obtenidos en cada lectura se programaron para ser escritos en un archivo

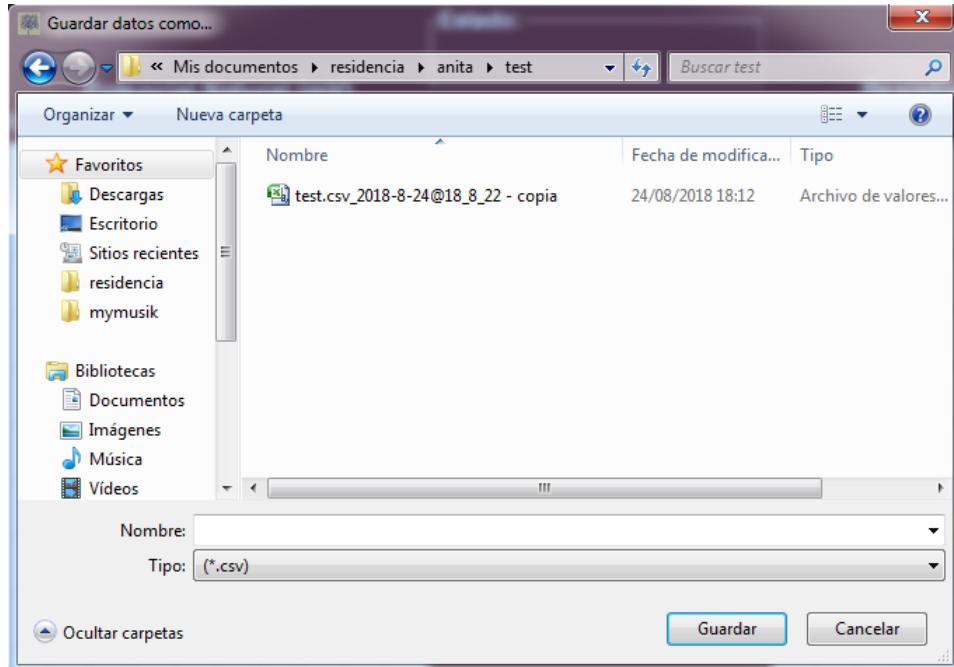


Figura 8.1: Guardando nuevo archivo CSV

CSV guardado de modo local, preguntando al usuario qué nombre debería tener ese archivo, esto se logra con la instrucción:

```
uiputfile('*.csv', 'Guardar datos como...');
```

La cual abre un cuadro de diálogo y espera recibir un nombre de archivo para guardar ahí los datos. Hecho esto y una vez que inicia el ciclo de trabajo se guardaron en una columna los datos del sensor izquierdo; correspondientes a la curva de potencial espontáneo y en otra columna los datos del sensor derecho; correspondientes a la curva de resistividad. Para escribir cada uno de los valores leídos se utiliza la instrucción de MATLAB:

```
disV = 7.7;
disI = 15.2;
fprintf(archivo, '%.3f,\t', (disV - (izq - 4))*escalaV / disV);
fprintf(archivo, '%.3f\r\n', (disI - (der - 4))*escalaR / disI);
```

```

2.442, 3.783
2.474, 3.816
2.506, 3.816
2.506, 3.816
4.883, 3.816
4.883, 3.776
4.818, 7.566
4.883, 7.553
4.883, 7.553
5.013, 15.263
5.013, 15.250
4.883, 15.250
1.857, 15.132
2.429, 14.053
2.584, 10.908
2.455, 9.632

```

Figura 8.2: Estructura de archivo CSV generado por anita

Donde *archivo* es un apuntador que está conectado al archivo local y le permite saber dónde escribir los datos, *%.³f* significa que únicamente va a guardar el valor entero junto con tres decimales, */t*, */r*, */n* son instrucciones especiales que indican dejar una tabulación de espacio y bajar un renglón respectivamente.

La variable *izq* es el valor leído, nótense que se aplican algunas operaciones matemáticas, esto corresponde a la resta de *4 cm* del espacio entre el sensor y el inicio de la zona de lecturas; luego *disV* y *disI* son longitudes correspondientes a la zona de dibujo de las curvas. Finalmente se aplica la regla de 3 tal como se hizo en la ecuación 6.5.

Después de grabar los datos se programaron las instrucciones necesarias para dibujar las gráficas, MATLAB contiene la función *plot* cuyo parámetro de entrada es un conjunto de datos y sirve para graficarlos en pantalla; ya se tenía el conjunto de datos y sólo hacia falta ocuparlo dentro del programa, esto se logró con la instrucción:

```
dataset = load(archivo);
```

Donde *dataset* es la variable que guarda todos los datos que estén contenidos en *archivo*, se espera que ésto último sea un nombre de archivo existente ya que de otra forma se producirá un error.

Cada vez que un nuevo dato se graba; se ejecutan las funciones necesarias para que

se vuelva a cargar el archivo dentro de MATLAB, se actualice la variable *dataset* y dibuje el nuevo conjunto de datos, con esto se alcanzó la característica de graficar los datos en tiempo real.

8.2. Control del trabajo

Se implementó una forma en la que se pueden crear nuevos archivos o reanudar los que han sido guardados con anterioridad, el flujo de las lecturas también se puede controlar mediante tres botones colocados en la interfaz gráfica de usuario. permitiendo iniciar, pausar/reanudar y detener la captura de nuevos datos.

8.3. Diseño de la interfaz gráfica de usuario

8.3.1. Elementos

Se diseñó de tal manera que se mostrara el dibujo de las gráficas en tiempo real en dos conjuntos de ejes designados; uno para la curva de potencial espontáneo colocado a la izquierda y otro para la curva de resistividad colocado a la derecha, cada que se toma una nueva lectura el dibujo en pantalla se actualiza, se agruparon los botones respecto a su categoría:

- Archivos: Permite crear nuevos archivos CSV o abrir archivos existentes
- Control: Son los botones que manejan la toma de lecturas
- Reportes: Pueden generar un documento en PDF del archivo actual o de otro ya existente

Anita contiene un letrero cambiante donde es posible verificar el estado actual de las lecturas, éste letrero puede contener tres valores posibles: *Leyendo*, *Pausado* y *Detenido*. Cuando se llega a éste último estado el archivo actual se cierra y al mismo tiempo

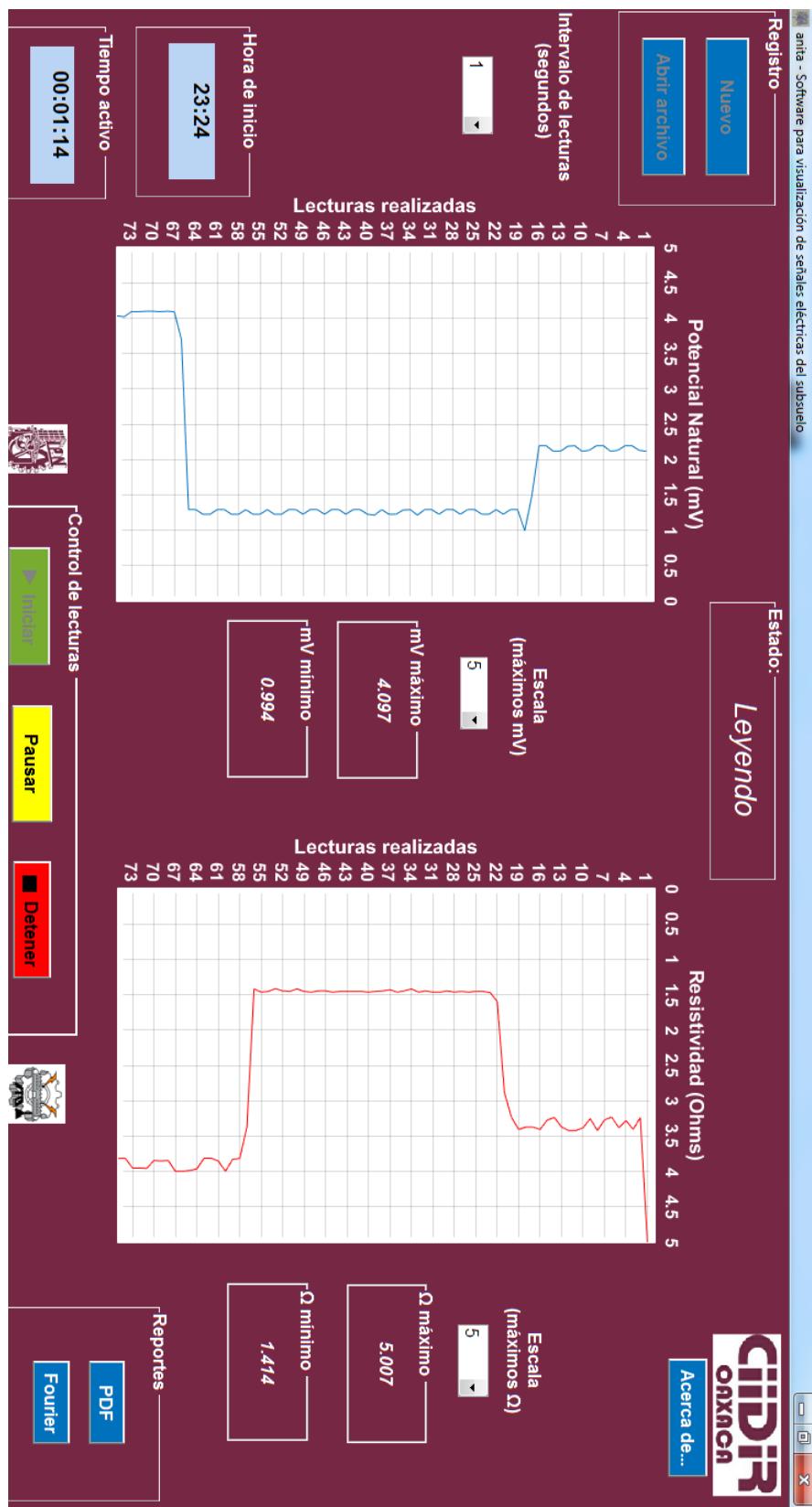


Figura 8.3: Interfaz principal de anita graficando datos



Figura 8.4: Botones para archivos



Figura 8.5: Botones para el control de lecturas

los botones que controlan el flujo de lecturas se bloquean, esto quiere decir que no es posible escribir más en ese archivo a menos que se vuelva abrir. Los controles de lecturas se desbloquearán únicamente con la creación de un nuevo archivo o con la apertura de uno existente.

También se colocaron controles en forma de lista desplegable que permiten modificar el límite máximo (span) de cada una de las curvas, el cuadriculado de las gráficas no se ve influido y siempre serán diez divisiones a lo largo del eje horizontal.

Las etiquetas en el lado derecho de las gráficas indican el valor máximo y mínimo leído de entre todos los valores ya tomados.

Otro control que se configura mediante una lista desplegable es el intervalo que hay entre una lectura y otra, puede variar desde uno hasta diez segundos.

Finalmente se colocaron campos que indican la hora en que se inició el trabajo; este dato se toma desde la hora que tenga el sistema operativo, se colocó además un cam-



Figura 8.6: Etiqueta que muestra el estado de las lecturas

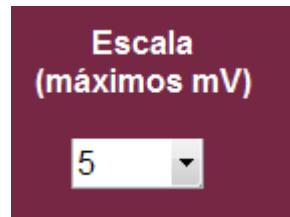


Figura 8.7: Controles de span

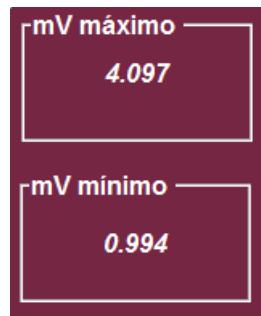


Figura 8.8: Etiquetas de máximos y mínimos



Figura 8.9: Control de intervalo de lecturas



Figura 8.10: Etiquetas para visualizar el tiempo de trabajo

Color	Valor
Red	118
Green	39
Blue	68

Cuadro 8.1: Combinación RGB del color guinda

po que indica en formato de horas, minutos y segundos el tiempo que ha transcurrido mientras se toman lecturas, si el trabajo se llegase a pausar; éste tiempo también se detiene y se reanuda en caso de retomar el trabajo.

8.3.2. Colores

Se eligieron los colores representativos del Instituto Politécnico Nacional¹ para el fondo de la interfaz gráfica de usuario principal, también se agregaron los emblemas del CIIDIR, IPN e ITO. Los colores fueron puestos mediante formato RGB² en la escala establecida en el cuadro 8.1.

En cuanto al ícono del software, representa al programador y autor de esta investigación; es un dibujo de su autoría.

¹Guinda y blanco

²Es un formato que combina los colores rojo (R), verde (G) y azul (B) en una escala de 1 a 255.



Figura 8.11: Icono de anita

8.4. Manejo de excepciones

Existen ocasiones en las que por descuido se olvide el flujo de trabajo y se quiera realizar una acción no programada, MATLAB puede manejar esto de la siguiente manera:

```
try
    conjunto de instrucciones que puedan generar una excepcion
catch
    instrucciones a ejecutar en caso de producirse un error
end
```

8.4.1. Inicio de lecturas sin el módulo conectado

Otra causa común de excepción es querer iniciar la toma de lecturas sin haber conectado el puerto serial del módulo a la computadora, cuando se crea un archivo nuevo o se abre uno existente; el botón *Iniciar* se desbloquea permitiendo la ejecución de las funciones para llevar a cabo la lectura del puerto serial, sin embargo si el módulo no está conectado se le muestra una alerta al usuario para proceda.

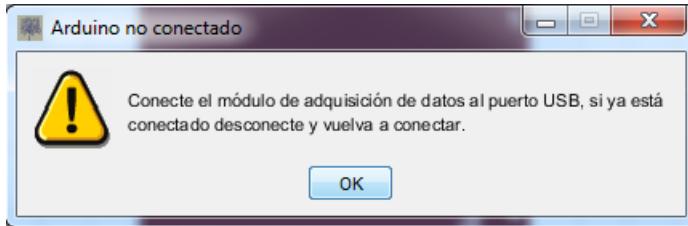


Figura 8.12: Alerta cuando el módulo no está conectado

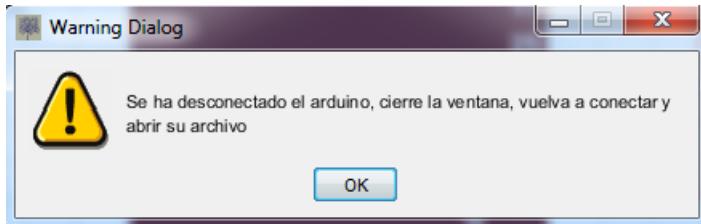


Figura 8.13: Alerta por desconexión repentina del módulo

8.4.2. Desconexión del módulo de adquisición de datos

Una de las excepciones con las que cuenta anita es la suspensión del trabajo y cierre de archivos con los que se estén trabajando en caso de que no se encuentre una respuesta por parte del puerto serial, principalmente porque ha sido desconectado y no existen datos en transmisión.

8.4.3. Eventos

Éstos últimos consisten en la correcta ejecución de los pasos para el inicio de un nuevo trabajo:

1. Conectar el módulo de adquisición de datos.
2. Crear o abrir un archivo
3. Iniciar la toma de lecturas

Iniciar un nuevo ciclo de lecturas bloquea los controles para abrir o crear un nuevo archivo y desbloquea los botones que controlan la pausa y el detenimiento de lecturas.

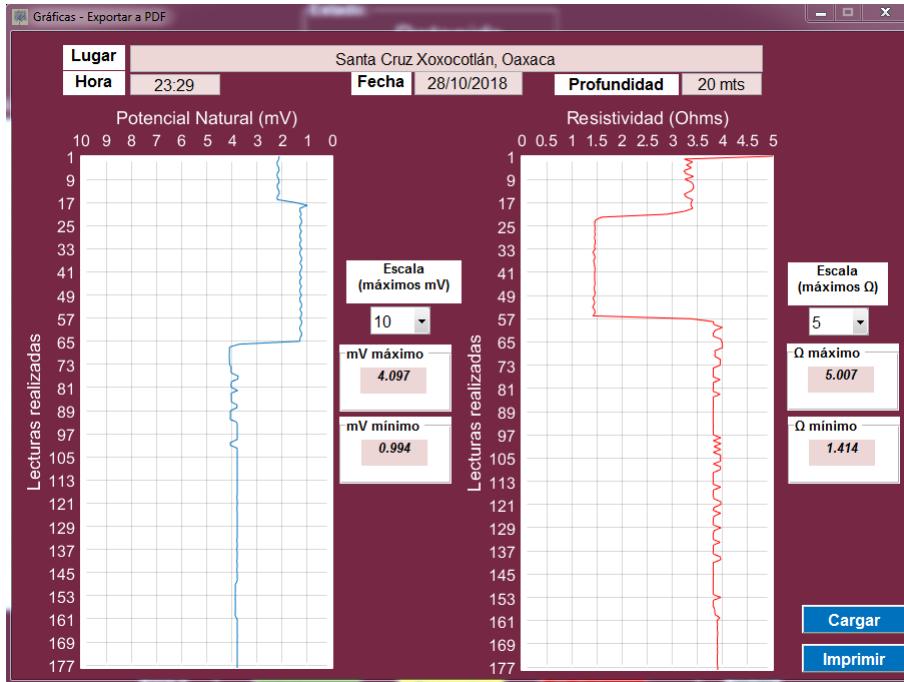


Figura 8.14: Interfaz para la generación de reportes en PDF

8.5. Reportes en PDF

Se colocó un botón que abre una nueva interfaz; en ésta es posible cargar un archivo de datos CSV generado por anita para exportarlo en un documento PDF, se agregó un encabezado con la decoración de los documentos oficiales del CIIDIR.

La interfaz permite visualizar el dibujo de las gráficas antes de generar el reporte, también es posible ajustar su span y escribir datos como el lugar donde se realizó el RGP, la profundidad que el pozo tiene, la fecha en que fue realizado y la hora. El documento se genera en una sola página y se anexa un ejemplo de un reporte al final de este documento.

8.6. Empaquetado

Para la distribución del software se utilizó la herramienta de compilación de aplicaciones de incluida en MATLAB, se generó un archivo que es instalable en sistemas

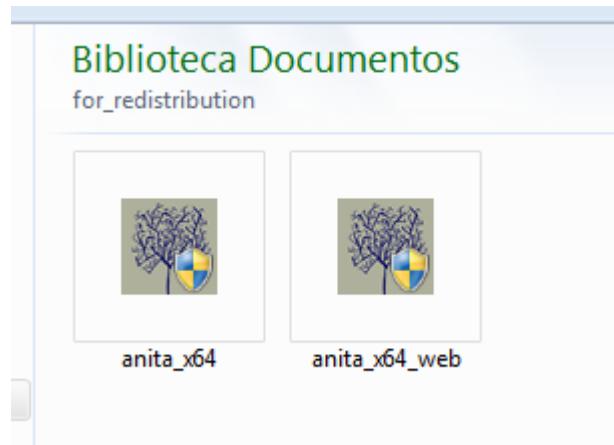


Figura 8.15: Software empaquetado para su distribución

operativos Windows 7 o superior en versiones de 64 bits. Se compiló una versión que incluye el MATLAB Runtime mencionado en la sección 3.4.1 y una que no lo incluye, ambas quedan disponibles para su descarga en la sección *releases* de su repositorio en Github [21].

Capítulo 9

Resultados

9.1. Conclusiones

Se ha resuelto la situación para dibujar las gráficas en un entorno digital eliminando así la necesidad de papel, el software programado para esta tarea es capaz de recibir, graficar y procesar las señales que genera la sonda de RGP, el software va dibujando en tiempo real y además indica la duración el trabajo, algo con lo que la sonda no cuenta. Se cumple el objetivo de poner e marcha un software para la comunicación y procesamiento de señales de Potencial Espontáneo y de Resistividad, producto de las mediciones en el subsuelo a través de un equipo de registros eléctricos.

Se acepta la primera hipótesis la cual menciona que se tendrá una mejor perspectiva de las gráficas al grabar cada lectura en un archivo CSV, ahora no sólo se pueden observar las gráficas sino que además se pueden consultar sus archivos fuente para saber cuánto vale cada punto de ellas.

Se acepta también la segunda hipótesis que habla de la facilidad para compartir los estudios realizados, el documento generado en formato PDF puede imprimirse o ser enviado mediante correo electrónico.

9.2. Trabajo a futuro

Se propone las siguientes mejoras en el software:

- Respaldar los archivos en una base de datos.
- Programar el envío de los estudios a través de Internet en un esquema de cliente (toda aquella persona que utilice el software anita) y servidor (la base de datos).
- Establecer una API (Interfaz de Programación de Aplicaciones) que permita explotar la información obtenida en la base de datos para generar conocimiento a través de minería de datos.

Apéndice A

Prueba de correo electrónico

Utilizando el formulario encontrado en la página web¹ de este distribuidor de equipos para estudios geofísicos se preguntó acerca del equipo [2], obteniendo la respuesta unas horas después.

 **Brian Peterson** <peterson@centurywirelineservices.com>
to me ▾ 2:36 PM (13 minutes ago) ☆ ↶ :

Hello Vama, in regards to your question, you would need about \$ 30,000 USD for a complete system with 300 meters winch, electric logging tool, interface box and computer.

Hope that helps and regards,
Brian P.

On Fri, Oct 19, 2018 at 2:08 PM <no-reply@parastorage.com> wrote:
Tienes un mensaje nuevo:
Via: <https://www.century-geo.com/>

Detalles del Mensaje:

Company / Organization Centro Interdisciplinario de Investigación para el Desarrollo Integral Regional (CIIDIR)
Name Agustin Vasquez Martinez
Phone 951 102 3676
Email vama.agus@gmail.com
Subject Price of one of your products

Message Greetings, I'm an engineering student from México, I'm doing my thesis and I worked with an old Drawwork machine which stil uses paper, I did the job to log, process and print the signal in a computer.

I'd like to justify why I did this in an old machine instead of buying a new one and I think the price could be a good factor, so I'd like to know the price of this product: <https://www.century-geo.com/drawworks-300m>

Thanks in advance, all of your responses will be used only for research purposes.

¹<https://www.century-geo.com/>

Apéndice B

Ethernet

Es una tecnología para la comunicación entre dispositivos en redes de computadoras, utiliza un cable llamado *categoría 6*, aunque también se encuentra la categoría 5 que es un cable menos robusto que el 6 y la categoría blindada que cuenta con protección en los 8 hilos internos. Dentro del aislante se pueden encontrar ocho cables de cobre organizados en pares cuyos colores son:

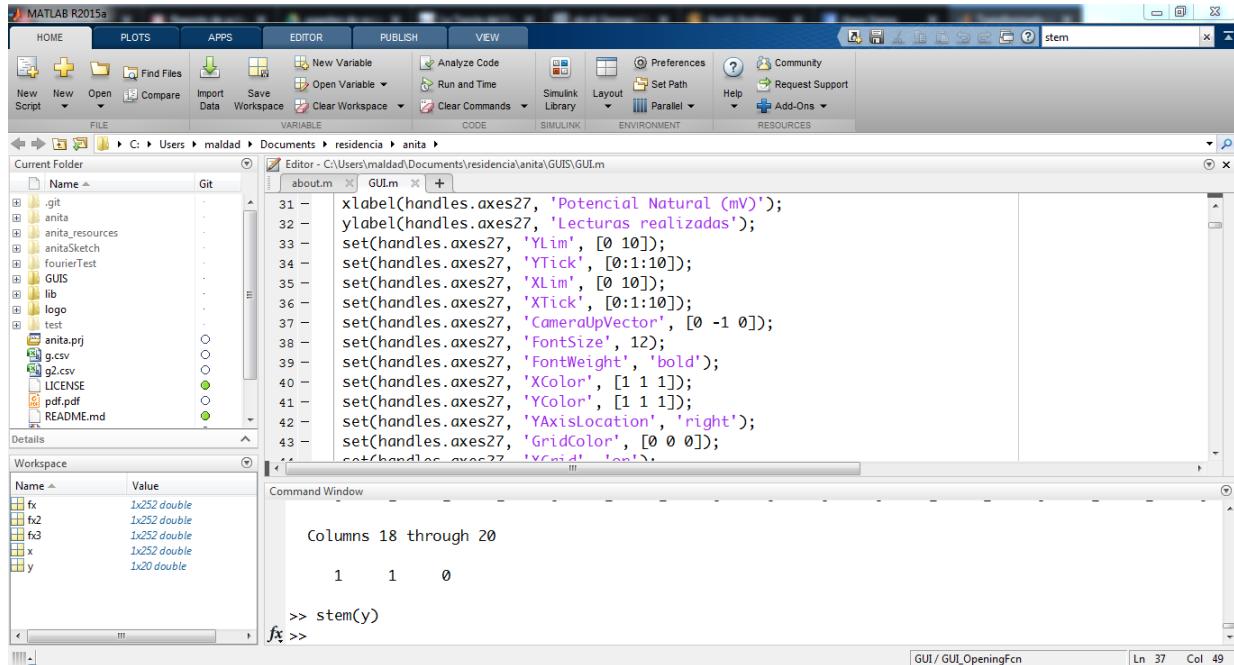
- Verde y Blanco con verde
- Azul y Blanco con azul
- Café y Blanco con café
- Naranja y Blanco con naranja



Figura B.1: Puerto Ethernet y un cable

Apéndice C

Interfaz de MATLAB



Apéndice D

Formato de datos en C

%c	un char (caracter)
%d	un entero con signo en notación de base decimal
%i	un entero con signo
%e	reales en notación científica indicando el exponente con "e"
%E	reales en notación científica indicando el exponente con "E"
%f	formato de punto flotante
%o	un entero sin signo en notación de base octal
%s	una cadena de caracteres
%u	un entero sin signo
%x	entero sin signo en notación de base hexadecimal, minúsculas
%X	un entero sin signo en notación de base hexadecimal, mayúsculas
%p	un puntero
%n	un puntero a un entero

Bibliografía

- [1] Registro Geofísico de Pozos - <http://www.geotem.com.mx/registro.php>
- [2] Equipo RGP de Century Geophysycal - <https://www.centurygeo.com/drawworks-300m>
- [3] PDF - <https://es.wikipedia.org/wiki/PDF>
- [4] Licencia del MIT (Massachusetts Institute of Technology) - <https://choosealicense.com/licenses/mit/>
- [5] Ingeniería de petróleo - <http://www.ingenieriadepetroleo.com/registro-sp/>
- [6] Ingeniería de petróleo 2 - <http://www.ingenieriadepetroleo.com/registros-de-resistividad/>
- [7] Exploración Geofísica y Estudios del Subsuelo del Centro S.A. de C.V. - <https://www.egyes.mx/nosotros/>
- [8] Geoelec - <http://www.geoelec.com.mx/nosotros.html>
- [9] cppReference - https://es.cppreference.com/w/cpp/io/c/printf_format
- [10] Geotem - <http://www.geotem.com.mx/nosotros.php>
- [11] Exploración, Perforación y Estudios del Subsuelo, S.A. de C.V. - <http://epyesa.com/quienes-somos/>

- [12] Arduino IDE - <https://www.arduino.cc/en/Main/Software>
- [13] Arduino Lenguaje - <https://www.arduino.cc/reference/en/>
- [14] Repositorio de Git - <https://github.com/git/git>
- [15] Github - <https://github.com/>
- [16] GeekyTheory - <https://geekytheory.com/matlab-arduino-serial-port-communication/>
- [17] Detección de colores - <https://www.youtube.com/watch?v=DKhTtNc3eoQ&t=11s>
- [18] Detección de distancias - https://www.youtube.com/watch?v=NseZfzIwO_s
- [19] Electrónica Aragón (ubicación) - <https://goo.gl/maps/knEwGfg12Qk>
- [20] anita repo - <https://github.com/maldad/anita/>
- [21] anita releases - <https://github.com/maldad/anita/releases>