



UNIVERSIDAD AUTONOMA DEL ESTADO DE MÉXICO



CENTRO UNIVERSITARIO TEXCOCO

PLATAFORMA INTEGRAL PARA EL DESARROLLO Y AUTOMATIZACIÓN DE UN  
PLUVIÓGRAFO THIES

**TESIS**

QUE PARA OBTENER EL GRADO DE

**MAESTRO EN CIENCIAS DE LA COMPUTACIÓN**

**PRESENTA**

GONZÁLEZ MANRÍQUEZ IRVING ALEJANDRO

**TUTOR ACADÉMICO**

DR. OZIEL LUGO ESPINOSA

**TUTORES ADJUNTOS**

DR. JOEL AYALA DE LA VEGA

DR. ALFONSO ZARCO HIDALGO

TEXCOCO, ESTADO DE MÉXICO

ENERO 2015



DICTÁMEN DE AUTORIZACIÓN Y OBTENCIÓN DE GRADO DE MAestrÍA

Texcoco, Méx., a 13 de ENERO del 2015

Título del proyecto:

PLATAFORMA INTEGRAL PARA EL DESARROLLO Y AUTOMATIZACIÓN DE UN PLEVIÓGRAFO THLS

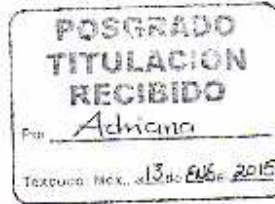
Tesisista:

IRVING ALEJANDRO GONZALEZ MANRIQUEZ

Dictamen:

No. de revisión: 5

- Rechazado
- Sujeto a modificaciones
- Aceptado, condicionado
- Aceptado



Observaciones generales:

Aceptado para la impresión.. SI

Aceptado para la defensa de grado.. SI

Tutor Adjunto	Tutor Académico	Tutor Adjunto
Dr. Joel Ayala de la Vega	Dr. Oziel Lugo Espinosa	Dr. Alfonso Zarco Hidalgo





## **AGRADECIMIENTOS**

Deseo expresar mi más sincero agradecimiento a todas aquellas personas que me brindaron su conocimiento y apoyo incondicional durante la realización de este trabajo de investigación, a todos ellos dedico el fruto de mi esfuerzo y sacrificio.

A mi padre por siempre dar su mayor esfuerzo para brindarme condiciones favorables durante toda mi formación profesional, por inculcarme el valor de la responsabilidad y por impulsarme a ser un hombre de bien.

A mi madre por acompañarme en todo momento, por sus cuidados y consejos que me guiaron y ayudaron a mantenerme firme en mis propósitos e ideales.

A mi hermano, a mi hermana, a mi cuñado y mi cuñada por estar siempre a mi lado y dispuestos a ayudarme cuando lo necesito.

A Rubí González que comparte a mi lado cada triunfo y tropiezo, y por el apoyo y comprensión que me brinda día con día.

A mi tutor académico, el Dr. Oziel Lugo Espinosa por haber confiado en mí y brindarme su apoyo, su vasto conocimiento y su amplia experiencia para la realización de este trabajo de tesis.

A mis tutores adjuntos el Dr. Joel Ayala de la Vega y el Dr. Alfonso Zarco Hidalgo por el tiempo dedicado a revisar esta tesis.

Al Consejo Nacional de Ciencia y Tecnología por haber proporcionado el apoyo económico sin el cual no hubiera sido posible el desarrollo de este proyecto.

Al Centro Universitario UAEM Texcoco por ser uno de los pilares en mi formación académica.

## RESUMEN

En este trabajo se plantea la construcción de un pluviógrafo automático con materiales de bajo costo equivalente a un pluviógrafo tipo Hellman de sifón, con el objetivo de adquirir los datos de la precipitación atmosférica; como complemento, se plantea el desarrollo de una aplicación de software para el análisis y presentación de los datos que genera el pluviógrafo. Para automatizar el pluviógrafo se implementa un sistema electrónico con base en un sensor de nivel de líquidos, un módulo de control del tiempo y una placa electrónica arduino. El sistema electrónico almacena los datos de precipitación en una tarjeta de memoria flash microSD. El software se codificó en lenguaje de programación Java y la herramienta permite interpretar la información captada por el circuito electrónico, realizar consultas de los datos de la precipitación atmosférica registrada por el pluviógrafo, desplegar tablas con la descripción de cada evento de lluvia (tormenta) y generar gráficas como pluviogramas e hietogramas. Al validar el prototipo se comprobó una exactitud total de los datos lo que permite el uso del dispositivo en ámbitos caseros o profesionales donde se requiera la lectura de la precipitación.

# Contenido

Índice de figuras.....	3
Índice de tablas.....	5
Introducción.....	7
Planteamiento del problema .....	9
Justificación del problema .....	12
Objetivo general .....	13
Objetivos específicos .....	13
Hipótesis.....	14
Antecedentes.....	15
<i>Digitalización de la información generada por los pluviógrafos.</i> .....	15
<i>Pluviógrafos electrónicos.</i> .....	17
Capítulo 1: Medición de la precipitación con pluviógrafos. ....	21
<i>Ciclo Hidrológico.</i> .....	21
<i>¿Para qué medir la precipitación?</i> .....	21
<i>Tipos de precipitación.</i> .....	23
<i>Cómo se forma la precipitación.</i> .....	23
<i>Condensación.</i> .....	23
<i>Coalescencia</i> .....	24
<i>Proceso de Bergeron.</i> .....	24
<i>Características de la precipitación.</i> .....	25
<i>Usos del agua.</i> .....	25
<i>Equipos colectores.</i> .....	26
<i>Pluviómetro.</i> .....	26
<i>Pluviógrafo.</i> .....	27
Capítulo 2: Análisis y desarrollo del hardware. ....	41
<i>Arduino</i> .....	41
<i>¿Qué es Arduino?</i> .....	41
<i>¿Para qué sirve?</i> .....	41
<i>Modelos de placas Arduino.</i> .....	42

<i>Ventajas sobre otros microcontroladores.</i>	44
<i>Plataforma Arduino.</i>	44
<i>Lenguaje de Programación.</i>	45
<i>Sketch de Arduino.</i>	45
<i>Áreas de investigación donde se ha utilizado Arduino</i>	47
<i>Sensor de nivel de líquidos eTape.</i>	49
<i>Descripción y modo de operación.</i>	49
<i>Características.</i>	49
<i>Sensor DS1307 Real Time Clock.</i>	51
<i>microSD Shield</i>	51
<i>Diseño y desarrollo de estructura</i>	53
<i>Funcionamiento del circuito electrónico del instrumento.</i>	61
<i>Capítulo 3: Análisis y desarrollo del software.</i>	64
<i>Personal Software Process.</i>	65
<i>Aplicación de la metodología.</i>	66
<i>Capítulo 4: Integración de la plataforma</i>	72
<i>Hardware.</i>	72
<i>Costos</i>	73
<i>Problemas en la construcción</i>	74
<i>Software</i>	75
<i>Modulo Principal.</i>	75
<i>Módulo de Eventos de Lluvia.</i>	76
<i>Módulo Excel.</i>	78
<i>Módulo Pluviogramas.</i>	78
<i>Módulo Hietogramas.</i>	78
<i>Validación</i>	79
<i>Conclusiones</i>	81
<i>Anexo A</i>	83
<i>Anexo B</i>	95
<i>Anexo C.</i>	99
<i>Anexo D</i>	100
<i>Bibliografía</i>	103

# *Índice de figuras*

Figura 1. Pluviograma de una tormenta.....	10
Figura 2. Pluviómetro Digital TFA 47.3003.....	19
Figura 3. Ciclo Hidrológico.....	22
Figura 4. Pluviómetro Tipo B. ....	27
Figura 5. Pluviógrafo de balanza. ....	29
Figura 6. Pluviógrafo de sifón. ....	31
Figura 7. Segmento de un pluviograma.....	37
Figura 8. Placa Arduino Uno .....	43
Figura 9. IDE de Arduino .....	46
Figura 10. Monitor serial. ....	47
Figura 11. eTape sensor de nivel de líquidos. ....	50
Figura 12. Sensor DS1307 RealTime Clock .....	51
Figura 13. microSD shield. ....	52
Figura 14. Pluviógrafo tipo Hellman marca Lambrecht.....	54
Figura 15. Partes que constituyen un pluviógrafo. ....	54
Figura 16. Cámara de sifonaje .....	55
Figura 17. Cámara de sifonaje y cubierta vista lateral .....	56
Figura 18. Cámara de captación. ....	56
Figura 19. Diseño del pluviógrafo.....	57
Figura 20. Estructura final del instrumento. ....	58
Figura 21. Vista interior del instrumento .....	59
Figura 22. Diagrama de bloques de la plataforma .....	60
Figura 23. Pantalla LCD. ....	61
Figura 24. Archivos de datos del instrumento .....	62
Figura 25. Diagrama de flujo de funcionamiento de circuito.....	63



Figura 26. Estructura básica de PSP.....	67
Figura 27. Diagrama UML de clase Lista.....	68
Figura 28. Interfaz del Hardware.....	72
Figura 29. Conexión de estructura física y circuito electrónico. ....	73
Figura 30. Instalación de aparato final. ....	74
Figura 31. Pantalla principal de software.....	76
Figura 32. Pantalla de eventos de lluvia.....	77
Figura 33. Descripción de un evento de lluvia. ....	77
Figura 34. Pantalla de sistema que muestra un pluviograma. ....	78
Figura 35. Pantalla de sistema donde se muestran un hietograma.....	79
Figura 36. Pluviógrafo thies .....	81

# *Índice de tablas*

Tabla 1. Datos obtenidos de un pluviograma.....	37
Tabla 2. Identificación de los 30 minutos más intensos de la tormenta .....	38
Tabla 3. Ajuste a los 30 minutos más intensos de la tormenta.....	39
Tabla 4. Componentes del Software .....	70
Tabla 5. Costos de los componentes de la estructura.....	74



# *Introducción*

La precipitación es cualquier tipo o estado de agua que cae sobre la superficie de la tierra (Wikipedia, 2013), describir la relación entre la cantidad de lluvia caída y la cantidad de agua que escurre superficialmente, es de suma importancia para determinar los diversos usos del agua, ya sea para usos domésticos, agricultura, control de inundaciones, generación de energía eléctrica, drenaje rural y urbano etc. a esta relación se le llama coeficiente de escurrimiento.

La medición de la precipitación se realiza para obtener información sobre sus características espaciales y temporales, como intensidad, frecuencia, fase, duración cantidad etc. El problema de la representatividad, que si bien es general para todas las mediciones, es particularmente importante en la precipitación, ya que presenta una gran variabilidad espacial y temporal.

La precipitación puede ser medida a través de pluviómetros, pluviógrafos o estación automatizada. El pluviómetro da información de lluvia cada 24 horas y el pluviógrafo, como su nombre lo indica, nos da una gráfica llamada pluviograma donde se registra continuamente la lluvia en las 24 horas. La lluvia puede ser reportada como lámina llovida ( $L_p$ ) en un cierto lapso de tiempo ( $t_p$ ), o bien, como intensidad de lluvia ( $I$ ). La intensidad de lluvia es la razón de incremento de la altura que alcanza la lluvia respecto al tiempo. Los pluviógrafos reportan la lámina precipitada para cualquier intervalo de tiempo y registran la lámina precipitada en mm de lluvia donde 1 mm de lluvia equivale a 1 Lt. de agua caída en  $1m^2$ .

El desarrollo de sistemas que permitan a los pluviógrafos actuales mejorar y automatizar sus mediciones es indispensable hoy en día. De esta manera se dará mayor precisión al desarrollo de trabajos hidrológicos como:

- Controlar la descarga del agua captada por las cuencas hidrográficas.
- Definir el diseño de las obras hidráulicas que explotarán las fuentes de agua superficiales.
- Determinar la altura de puentes que cruzan los ríos.
- Determinar el diseño de drenaje parcelario y urbano.
- Diseño y operación de presas.

En este trabajo se plantea el desarrollo de un pluviógrafo automático y un sistema para procesar la información generada con el mismo con el fin de auxiliar en la interpretación de pluviogramas.

# *Planteamiento del problema*

En esta tesis se plantea el desarrollo de un pluviógrafo automático, para evitar pérdida de datos por atasco de papel, por una mala calibración del aparato, o porque el método para graficar de éste no pueda responder al trabajo intenso cuando se somete a fuertes tormentas. Además, el software que servirá como interfaz para comunicar el aparato con la computadora tendrá la función de realizar todo el procesamiento de información y un módulo encargado de generar gráficas de los eventos de lluvia, para brindar un panorama más amplio al momento de realizar estudios sobre las intensidades de las tormentas, y un módulo encargado de predicción de lluvia. Las gráficas se generarán a partir los datos obtenidos, teniendo así representaciones exactas, caso contrario a cómo se desarrolla el procesamiento actual, que a partir de la gráfica (pluviograma) se toman los datos, esto genera pérdida considerable de información, ya que si el pluviograma se ve afectado por la calibración del aparato, los datos tomados del mismo no son exactos. Con esto se logrará tener de manera digital las gráficas, ventaja que no se tiene en la actualidad ya que por la longitud del pluviograma, digitalizarlo se vuelve una tarea insostenible, por la dificultad de los medios para hacerlo, y las potentes características del equipo de cómputo necesarias para almacenar imágenes tan grandes.

La medición de la precipitación se realizan para obtener información sobre sus características espaciales y temporales, como intensidad, frecuencia, fase, duración, cantidad etc. y se obtiene mediante la lectura de los pluviogramas generados por los pluviógrafos. Los pluviogramas registran mediante una gráfica a través del tiempo las láminas precipitadas en milímetros. Un pluviograma permite registrar la precipitación de hasta un mes, por lo que un pluviograma es un rollo de papel que tiene varios metros de longitud. En muchas ocasiones los rollos de pluviograma se atascan en el aparato

ocasionando que éste no grafique y se pierdan los datos de una o varias tormentas según el tiempo que se haya tardado en estabilizar.

La Figura 1 muestra una vista parcial de un pluviograma en el cual se observa que en el eje de las X se puede leer la hora en la cual se registró la lluvia y en el eje de las Y se lee la lámina precipitada en milímetros. Se aclara que la lectura de lluvia solo cuenta cuando el pluviógrafo asciende; no se contabiliza cuando el pluviógrafo va de bajada. Por ejemplo, en la Figura 1 se puede observar que llovió aproximadamente 96 mm. Aunque se aclara que la plumilla del pluviógrafo se observa que solo la primera vez subió hasta 10 y las otras veces no. Eso podría ser causado por:

- El pluviógrafo está mal calibrado y no baja hasta 0 y no sube hasta 10.
- Cuando es una lluvia muy intensa el pluviógrafo trabaja tan rápido que no alcanza a subir hasta 10 porque ya tiene encima la otra lámina de lluvia por registrar.

Ambas situaciones causan errores muy graves en los resultados finales, ya que las estimaciones pierden exactitud.



Figura 1. Pluviograma de una tormenta

La lluvia registrada a través de las líneas ascendentes tiene pendientes diferentes, lo cual indica que la lluvia tiene intensidades diferentes. Por ejemplo, entre las 16:45 y las 18:15 la lluvia sube de una manera suave, pero entre 18:15 y 19:00 el pluviógrafo sube abruptamente. Es decir, el primer tramo de lluvia no es tan intenso como el segundo. Finalmente la tormenta terminó a las 20:10 y la plumilla del pluviógrafo queda en descanso, es decir, sigue funcionando, pero como no llueve sigue marcando de manera horizontal y volverá a subir solo cuando reinicie la lluvia. Cabe aclarar que cada vez que llueve se le conoce como un evento, y para distinguir de un evento a otro debe ocurrir un lapso de tiempo de al menos de 6 horas.



# *Justificación del problema*

Los datos generados por el pluviógrafo constituyen una base fundamental para los trabajos de manejo de cuencas e investigación en el departamento de irrigación de la Universidad Autónoma Chapingo (UACH) la cual cuenta con estaciones meteorológicas en diversos puntos del país, principalmente las zonas más lluviosas y ninguna de estas estaciones meteorológicas cuenta con algún pluviógrafo automático, en todas hay pluviógrafos thies los cuales son mecánicos y la información obtenida la representan únicamente a través del pluviograma.

Con la automatización del pluviógrafo, los datos de medición de la precipitación serán exactos, eliminando también la pérdida de información por error humano, y con el software interfaz se tendrá el soporte de una gráfica en formato digital que represente los datos y así permitir al departamento de irrigación un mejor manejo de la información de las cuencas y obtención de mejores resultados en sus investigaciones hidrográficas.

Con el desarrollo de este sistema y su implementación progresiva en las distintas estaciones meteorológicas, las investigaciones se verán también beneficiadas en la optimización del tiempo de desarrollo ya que en ocasiones los pluviogramas llegan a tardar meses en llegar al departamento de irrigación en las instalaciones de la UACH.

# ***Objetivo general***

Desarrollar una plataforma Hardware-Software que conste de la construcción de un pluviógrafo automatizado, y un software encargado de procesar los datos para generar informes y gráficas de la precipitación.

## ***Objetivos específicos***

- Diseñar la estructura física del aparato para su desarrollo posterior
- Definir el sensor adecuado o el dispositivo electrónico cuya función sea la de obtener las mediciones del aparato.
- Diseñar y desarrollar el sistema que se encargará de procesar la información, con un módulo encargado de graficar la información y un módulo encargado para la generación de reportes.

# ***Hipótesis***

El desarrollo e implementación de un pluviógrafo automático, utilizando un sensor de nivel de líquidos, y un sistema para procesar la información generada por el mismo, permitirá tener una exactitud mayor al 90% al realizar mediciones de precipitación.

Se compararán las mediciones generadas por el instrumento propuesto con las mediciones realizadas con un pluviógrafo mecánico instalado en una estación meteorológica.

# *Antecedentes*

El pluviógrafo está diseñado para determinar la intensidad de la lluvia durante largos periodos de tiempo. Este tipo de mediciones son ampliamente utilizadas en estudios de Hidrología. Estudiar la distribución de precipitación ayuda a proponer políticas de planificación de inundaciones y generalmente para mitigar el impacto de precipitaciones extremas (Vannitsem & Naveau, 2007).

Existen estudios que han investigado la estabilidad de la distribución de los diferentes tipos de lluvia (Fiser & Kveton, 2006). Conocer la distribución de la precipitación ayuda al diseño de señales satelitales o señales electromagnéticas con el fin de disminuir la atenuación por lluvia de las señales. Se ha comprobado que la lluvia afecta gravemente la propagación de ondas electromagnéticas por encima de 10 GHz. La atenuación de la señal es un fenómeno complejo que depende del tamaño de las gotas de lluvia, la temperatura, la velocidad y la estructura vertical u horizontal de la célula de lluvia (Sen & Singh, 2007). En la Hidrografía, ciencia encargada del estudio sistemático de los cuerpos de agua planetarios, el estudio de los patrones temporales de la lluvia es un factor importante para determinar la cantidad y calidad de erosión del suelo (Mojardi Gilan, Ahmadi, Jaafari, Bohamta, & Salajegheh, 64-69). En el diseño de sistemas hidráulicos, medir la cantidad de precipitación ayuda a prever el riesgo de inundaciones (Jennings, 2006).

El desarrollo tecnológico de los pluviógrafos ha ido avanzando actualmente existen variantes de pluviógrafos que emplean los avances de la electrónica y la informática para la captación, almacenamiento y postprocesamiento del dato de lluvia.

## ***Digitalización de la información generada por los pluviógrafos.***

La medición de la intensidad de la lluvia en los pluviógrafos tradicionales emplea un mecanismo de relojería que mueve un tambor en el cual se coloca un rollo de papel donde se graba, con una plumilla, el registro de la lluvia caída de forma instantánea. Posteriormente el rollo de papel, al cual llamaremos pluviograma, debe ser procesado para obtener información de la lluvia.

Procesar la información generada por un pluviógrafo es una tarea complicada, donde la persona que la realice, debe ser un especialista o al menos tener una capacitación previa (Delitala, Deidda, Mascaro, Piga, & Querzoli, 2010) además de que la interpretación tiene que ser realizada cuidadosamente y utilizando los mismos criterios cuando cambia la información de un pluviógrafo a otro, o cuando más de una persona interpreta la información colectada en las estaciones meteorológicas. El especialista indica en cada pluviograma, y según su apreciación, donde están ubicados los puntos de inflexión de registro donde se hace evidente un cambio en la intensidad de la lluvia.

En 1987 John Sansom publicó un método para digitalizar la información generada por los pluviógrafos de su época. Basó su estudio en las mediciones de precipitación realizadas por el servicio meteorológico de Nueva Zelanda el cual contaba con 150 pluviógrafos capaces de almacenar un registro de lluvia continuo en una gráfica que tenía que ser reemplazada en algunos casos una vez al mes y en otros una vez a la semana. Los métodos utilizados para interpretar la información de precipitación no se adecuaban a las condiciones de hardware de los nuevos instrumentos, ya que estaban adaptados para medidores con diferente diámetro y resolución que además tenían que reemplazar su gráfica diariamente. La información de los pluviógrafos era recibida mensualmente de distintas estaciones meteorológicas para su procesamiento y transformación de los datos de las gráficas (Sansom, 1987).

En el centro Nacional de Investigaciones Científicas en Cuba, se desarrolló el proyecto SAPHID: un Sistema Automatizado para el Procesamiento de Datos Hidrológicos (Vinardell Grandal, Rodríguez Rubio, & Fagundo Castillo, 1992) el cual fue implementado en lenguaje

Turbo Pascal (versión 5.5) y permite automatizar la digitalización de registros de campo de limnigramas y pluviogramas.

En 2010 se desarrolló un software en la Universidad de Cagliari, basado en técnicas de tratamiento de imágenes capaz de procesar gráficas de los pluviógrafos y de manera automática procesar la señal y producir una base de datos con registros de precipitaciones continuas y una resolución de 5 a 10 minutos. El objetivo de este trabajo, además de desarrollar una técnica que facilite la interpretación de la información de los pluviógrafos, es crear un registro histórico de las precipitaciones del siglo pasado con el fin de mejorar la caracterización histórica del clima en esa región (Delitala, Deidda, Mascaro, Piga, & Querzoli, 2010).

### ***Pluviógrafos electrónicos.***

El pluviógrafo es un invento del siglo XIX. Los primeros registros que se archivan datan del pluviógrafo tipo balancín de Walravens en enero de 1893, el pluviógrafo de balancín de Hottinger en septiembre de 1896 y el pluviógrafo de flotador de Hellmann-Fuess de mayo de 1898. Sin embargo, la primera patente es del pluviógrafo de flotador presentada por Chappell en mayo de 1938. (Chappell, 1938)

En abril de 1966 fue patentado por Barker (Barker, 1966) el primer pluviógrafo con registro gráfico de mediciones automáticas. El mismo consiste en un modelo del pluviógrafo de balancín, en el que cada vaciado de las cubetas activa un interruptor magnético que cierra un circuito generador de un pulso eléctrico, el cual se registra y se muestra en un contador enmarcado en la pared del pluviógrafo. A partir de ese momento, los pluviógrafos automatizados de balancín han ido en constante desarrollo, contando cada vez con más facilidades.

Todavía no se ha patentado ningún modelo de pluviógrafo de flotador automatizado, sin embargo existen trabajos que implementan sistemas electrónicos adaptables a un pluviógrafo de flotador o sifón que hacen uso de un sistema de comunicación inalámbrica

y una aplicación software para el control, almacenamiento y análisis de los datos (Sandoval, Molano, Mosquera, & González, 2010).

Existen en el mercado algunos pluviógrafos automáticos capaces de registrar lecturas y almacenar la información en memorias; algunos de ellos se encuentran en costos muy elevados y los de costos más asequibles presentan limitaciones de hardware como poca capacidad de memoria o problemas en su fuente de alimentación; o problemas de interfaz como la resolución del tiempo de captación de lluvia. Algunos de ellos se presentan a continuación:

- Pluviógrafo RRG-1. Permite monitorear los datos de lluvia con una laptop conectándola directamente a la tarjeta analógica digital o a control remoto por módem. Incluye una memoria de datos que puede almacenar registros de hasta 62 días de información de lluvia por horas.
- Pluviógrafo RGR-122. Permite monitorear los datos de los niveles de lluvia con distancias no mayores a 90 m. La transmisión de los datos se hace con ondas de radio, las cuales tienen la peculiaridad de que la información se envía del radio al recipiente sin tener una línea directa de comunicación. Dicho equipo guarda los datos de lluvia de hasta 9 días.
- Pluviómetro digital TFA 47.3003. Pluviómetro digital de vaciado automático y transmisión de datos por radio sin cable; también tiene la capacidad de medir la temperatura interior y exterior. Indica la cantidad de lluvia precipitada durante la última hora, día y tormenta. El instrumento se muestra en la Figura 2.



Figura 2. Pluviómetro Digital TFA 47.3003

Existen trabajos de investigación donde se han realizado propuestas para automatizar alguno de los tres tipos de pluviógrafos existentes. A continuación se describen los más significativos.

- **Automatización de un pluviógrafo de balancín** (León Méndez, Hernández González, Garrido Monagas, & Andalia Gilbert, 2013). Desarrollado en el Centro de Investigaciones Hidráulicas (CIH) en La Habana, Cuba. Impulsado por las necesidades del desarrollo de la actividad hidráulica en ese país, desarrollaron el primer pluviógrafo automático que mide las precipitaciones con una precisión de 0.2 mm mediante una acción de conmutación. Su diseño se basa en interruptores magnéticos y sensores ópticos que detectan la conmutación de la balanza.
- **Pluviógrafo electrónico con transmisión de datos inalámbrica.** (Sandoval, Molano, Mosquera, & González, 2010). Desarrollado en la Universidad del Cauca, Colombia. Este trabajo corresponde a la implementación de un sistema electrónico adaptable a un pluviógrafo tipo Hellman de sifón



automático, con el objetivo de adquirir los datos de la precipitación atmosférica que afecta la zona de estudio, haciendo uso de un sistema de comunicación inalámbrica, y una aplicación software para el control, almacenamiento y análisis de los datos. Para llevar a cabo estos procesos se implementa un circuito electrónico, dispositivos de transmisión inalámbrica con tecnología Wi-Fi y una aplicación software diseñada en lenguaje de programación PHP, que puede ser instalada en un equipo servidor de páginas Web. La información obtenida de la precipitación atmosférica se presenta en una página Web, en la cual se pueden hacer consultas de los datos actuales e históricos de la precipitación registrada por el pluviógrafo.

# ***Capítulo 1: Medición de la precipitación con pluviógrafos.***

## ***Ciclo Hidrológico.***

Se le denomina ciclo hidrológico al movimiento general o transferencias de masas de agua, tanto de un punto del planeta a otro como entre sus diferentes estados (Comisión Nacional del Agua, 2013), en la Figura 3 se muestran los diferentes procesos que intervienen en el ciclo del agua y donde podemos observar que la precipitación es únicamente una parte del ciclo entero. El ciclo hidrológico constituye una de las condiciones naturales más importantes de las cuencas hidrográficas. En México, el río Lerma, con una longitud aproximada de 705 km. es el principal escurrimiento superficial dentro de la cuenca que lleva su nombre.

## ***¿Para qué medir la precipitación?***

Las precipitaciones se miden por la altura que alcanza el agua caída sobre una superficie plana y horizontal, en la que no existieran pérdidas por infiltración ni evaporación. Tal altura se expresa en milímetros y las mediciones se llevan a una aproximación de los décimos de mm. Es necesario tener presente que un milímetro de precipitación equivale a un litro de agua por metro cuadrado:  $1\text{mm}=1\text{L}/\text{m}^2$ .



**Figura 3. Ciclo Hidrológico.**

La medición de la precipitación se realiza para obtener información sobre sus características espaciales y temporales, como intensidad, frecuencia, fase, duración cantidad etc. El problema de la representatividad, que si bien es general para todas las mediciones, es particularmente importante en la precipitación, ya que presenta una gran variabilidad espacial y temporal. Este tipo de mediciones se realiza con un pluviógrafo, que es un aparato utilizado en estaciones meteorológicas y se le da diversos usos.

Medir la precipitación es también una tarea recurrente en los trabajos de Hidrología, que es la ciencia natural que estudia el agua, su ocurrencia, circulación y distribución en la superficie terrestre, sus propiedades físicas y químicas y su relación con el medio ambiente (Chow, 1964). Específicamente, la Hidrología mide la cantidad de precipitación para poder describir la relación existente entre el agua de lluvia y el escurrimiento, tarea importante para poder determinar la cantidad de agua que se destina para usos

consuntivos y no consuntivos, el diseño de obras públicas como la construcción de puentes, diseño de drenaje, evitar inundaciones, construcción de presas, entre otros.

En México existen alrededor de 3,817 estaciones meteorológicas oficiales en operación, encargadas de medir variables climatológicas e hidrométricas como precipitación, temperatura, evaporación, velocidad y dirección del viento.

Se estima que caen anualmente sobre el planeta aproximadamente 505000 km<sup>3</sup> de agua, y de ellos 398000 km<sup>3</sup> caen sobre los océanos. En México llueve cada año aproximadamente 1489 km<sup>3</sup> de agua. De esa cantidad, se estima que el 73.2% se evapora, otro 22.1% se incorpora a ríos o arroyos y el restante 4.7% se infiltra al subsuelo y recarga los acuíferos (Comisión Nacional del Agua, 2013).

### ***Tipos de precipitación.***

Existen tres categorías en las cuales se pueden clasificar cualquier tipo de hidrometeoro que cae de la atmósfera y llega a la superficie terrestre (Pérez, 2014).

- Precipitación congelada. Dentro de esta categoría se clasifica todo tipo de precipitación que se presente en estado sólido, por ejemplo: cristales de hielo, granizo o copos de nieve.
- Precipitación glacial: En esta categoría se clasifican las lloviznas congeladas y la lluvia congelada mejor conocida como aguanieve.
- Precipitación líquida: En esta categoría se clasifican la lluvia y llovizna.

### ***Cómo se forma la precipitación.***

#### ***Condensación.***

La precipitación comienza a formarse cuando asciende el aire cálido y húmedo. Al enfriarse el aire, el vapor de agua comienza a condensarse en núcleos de condensación, formando nubes. Después de que las gotas de agua se ponen lo bastante grandes, pueden ocurrir los siguientes dos procesos (Pérez, 2014):

### ***Coalescencia***

La coalescencia ocurre cuando las gotas de agua se fusionan para crear gotas más grandes, o cuando las gotas se congelan en un cristal de hielo. La resistencia del aire hace que las gotas de agua en una nube permanezcan inmóviles. Cuando se produce una turbulencia del aire, las gotas de agua chocan, produciendo gotas más grandes. Cuando estas gotas descienden, la fusión continua, de modo que las gotas se hacen lo bastante pesadas como para vencer la resistencia del aire y caer como lluvia (Pérez, 2014).

### ***Proceso de Bergeron***

El proceso de Bergeron ocurre cuando los cristales de hielo adquieren moléculas de agua de las gotas de agua superfrías cercanas. Cuando estos cristales de hielo ganan bastante masa, comienzan a caer. Esto generalmente requiere más masa que la fusión entre el cristal y las gotas de agua vecinas. Este proceso es dependiente de la temperatura, ya que las gotas de agua superfrías sólo existen en una nube por debajo de la congelación. Además, debido a la gran diferencia de temperaturas entre la nube y el nivel de tierra, estos cristales de hielo pueden derretirse cuando caen y convertirse en lluvia. (Pérez, 2014)

## ***Características de la precipitación.***

Las gotas de lluvia tienen tamaños en los límites de 0.1mm hasta los 9 mm de diámetro, y por encima de ese tamaño tienden a romperse. Las gotas más pequeñas tienen forma esférica. Cuando una gota de lluvia aumenta de tamaño, su forma se hace más redondeada, con un corte transversal más grande.

La intensidad y la duración, por lo general se encuentran inversamente relacionadas, es decir, las tormentas de intensidades más altas serán de duración corta, y las tormentas de intensidad baja generalmente tienen una duración más larga.

## ***Usos del agua.***

Se define al uso como la aplicación del agua a una actividad. Cuando existe una diferencia entre el volumen suministrado y el volumen descargado, se trata de un uso consuntivo. Existen otros usos que no consumen agua como la generación de energía eléctrica, que utiliza el volumen almacenado en presas. A estos se les denomina usos no consuntivos.

En el registro Público de Derechos de Agua (REPDa) se tienen registrados los volúmenes concesionados (o asignados en el caso de volúmenes destinados al uso público y urbano) a los usuarios de aguas nacionales. En dicho registro se tienen clasificado los usos del agua en 12 rubros, mismos que para fines prácticos se han agrupado en cinco grandes grupos; cuatro de ellos: agrícola, abastecimiento público, industria autoabastecida y las termoeléctricas, corresponden a usos consuntivos mientras que el último grupo, el hidroeléctrico se contabiliza a parte por ser de uso no consuntivo.

Del agua destinada para usos consuntivos, destaca el volumen considerado para actividades agrícolas como el riego, que corresponden al 77% del total destinado a usos consuntivos; seguido del 14% destinado para el abastecimiento público, el 5% para

termoeléctricas y el 4% para actividades de industria auto establecida (Comisión Nacional del Agua, 2013).

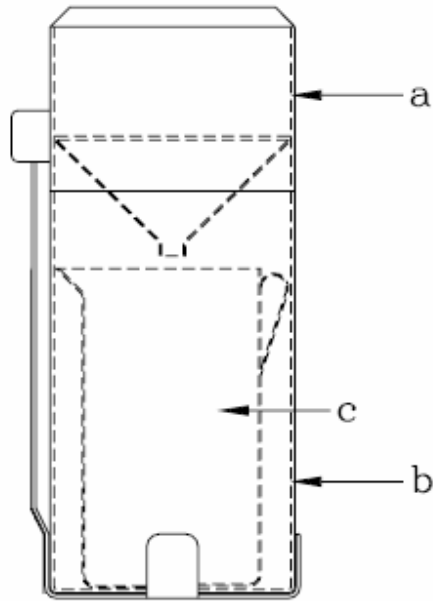
## ***Equipos colectores.***

### ***Pluviómetro***

Al no ser físicamente posible obtener o coleccionar la totalidad de la lluvia que llega a la tierra desde la atmósfera, el dato de la precipitación sólo puede ser obtenido a partir de muestras tomadas por instrumentos específicamente diseñados para ese fin, siendo los pluviómetros y pluviógrafos, los aparatos más empleados.

El pluviómetro es un instrumento que sirve para medir las cantidades de precipitación (Valdivieso, 2006). Están oficialmente reconocidos dos tipos de pluviómetros, de los cuales el que se haya en uso en la actualidad es el denominado Tipo B. Está compuesto de tres secciones principales, como se muestra en la Figura 4:

- La sección superior (a), que es la receptora, tiene una boca circular, formada por un aro de bronce reforzado, con la cara inclinada hacia afuera. En su interior tiene un embudo con orificio para la salida del agua. El borde superior de dicho embudo está soldado a las paredes del pluviómetro, a 10 cm por debajo de la boca, a fin de que las gotas que caigan sobre el mismo, no puedan volver al exterior por rebote.
- La sección inferior (b) destinada a retención.
- El recipiente (c), denominado colector, que sirve para trasvasar a la probeta el agua recogida, a fin de efectuar su medición.



**Figura 4. Pluviómetro Tipo B.**

El pluviómetro únicamente proporciona la altura de precipitación total en intervalos previamente establecidos, comúnmente 24 horas, medidas a partir de las 8 de la mañana de cada día.

### ***Pluviógrafo***

El pluviógrafo es un instrumento meteorológico empleado en el estudio y análisis de las precipitaciones. La base teórica que sustenta su funcionamiento y su uso está estrechamente relacionada con la del pluviómetro. El pluviógrafo tiene por función registrar de forma continua la cantidad de precipitación caída en un período de tiempo determinado, aunque se diferencia del pluviómetro en que además es capaz de registrar la intensidad de las precipitaciones, así como su variación en el tiempo.



## ***Tipos***

Para obtener registros continuos de las precipitaciones y determinar las intensidades de lluvia producidas en intervalos de tiempo predeterminados, se recurre al empleo del pluviógrafo, que consiste básicamente en un pluviómetro con un dispositivo de registro cronológico de las alturas de agua precipitadas.

Existen tres tipos fundamentales de pluviógrafos: de peso, de balanza y de flotador o sifón. Se realizó un estudio de precisión que involucró un pluviógrafo de flotador y un pluviógrafo de balanza y resultó que las cantidades totales de precipitación en todo el periodo de prueba medidas por ambos instrumentos estaban muy cercanas sin embargo la mayor suma de precipitación fue registrada por el pluviógrafo de flotador, con un total de 99.5% de precisión mientras que el valor registrado por el pluviógrafo de balanza fue de 98.9%. (Kotowski, Dancewicz, & Kazmierczak, 2011).

En el pluviógrafo de contrapeso, el peso de un recipiente junto con la precipitación acumulada en él, se registran continuamente, bien por medio de un mecanismo dotado de un muelle o bien con otro sistema de balanza de pesas. Este pluviógrafo no tiene un mecanismo que le permita vaciarse por sí mismo, pero existe un sistema de palancas que facilita que la pluma recorra la banda las veces que sea necesario. Este tipo de pluviógrafo está diseñado para evitar al mínimo las pérdidas por evaporación.

El pluviógrafo de balanza, mostrado en la Figura 5 cuenta, bajo la boca del embudo, con un compartimiento en el que hay dos cubetas gemelas, una de las cuales recibe el agua precipitada y al llenarse, se produce un desequilibrio que hace que la cubeta vuelque la cantidad de agua que contiene (equivalente a 0,1; 0,2 o 0,5 mm de lluvia según los modelos), moviendo a la segunda cubeta al lugar de recolección del agua. En ese momento se acciona un circuito eléctrico que marca o produce el registro correspondiente.

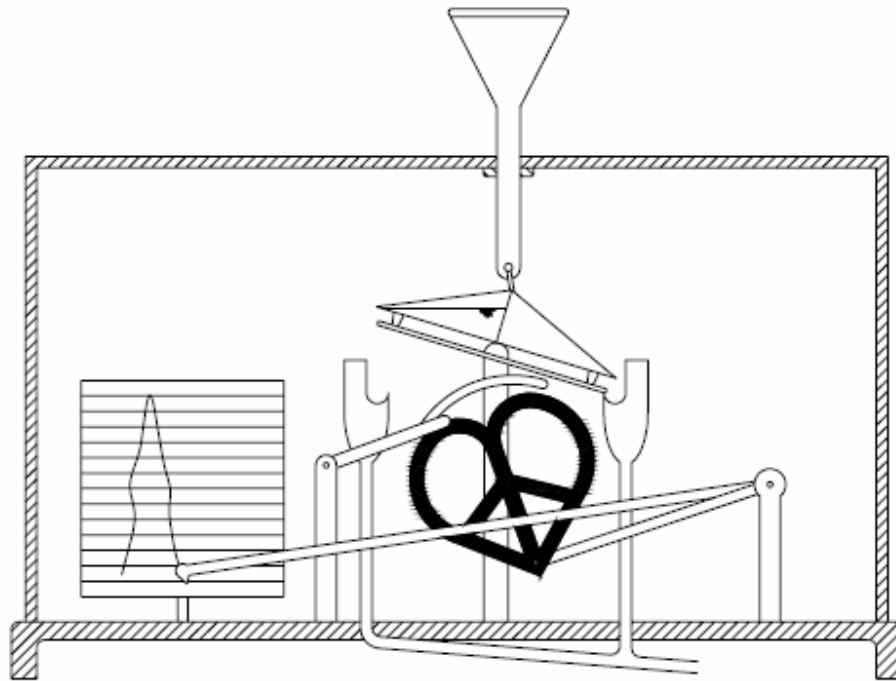


Figura 5. Pluviógrafo de balanza.

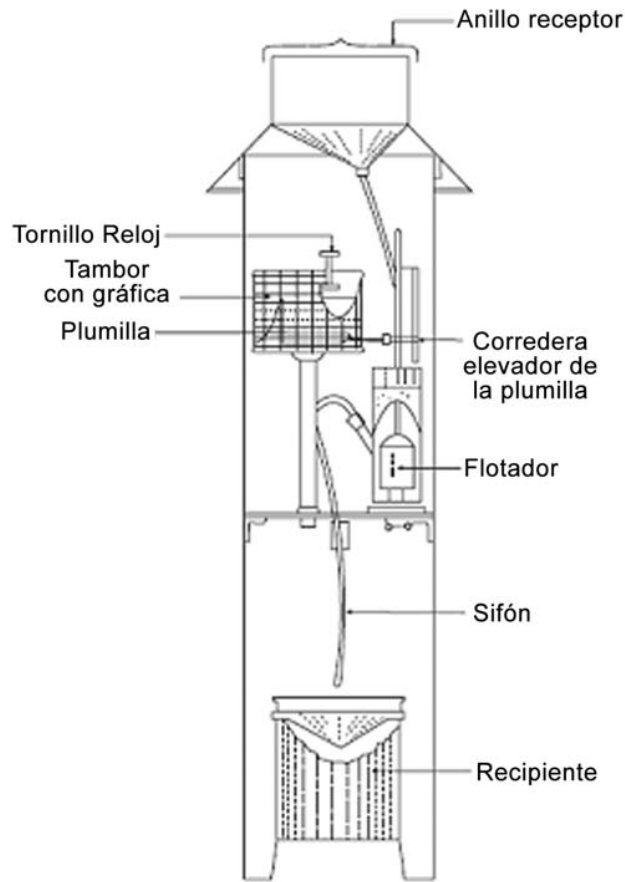
### ***Pluviógrafo de flotador o sifón.***

El pluviógrafo de flotador o sifón en el cual se centra este trabajo por ser el de uso común en México se muestra en la Figura 6, es un cilindro terminado en su parte superior en una boca circular de 16 cm. de diámetro. La lluvia, recogida por el receptor, pasa a través de un embudo y un tubo al recipiente cilíndrico que contiene un flotador. A medida que el nivel del agua en el depósito sube, lo hace también el flotador, que se encuentra vinculado al sistema registrador. La capacidad del recipiente es igual al volumen de agua correspondiente a 10 mm de lluvia, de modo que al llenarse, se accione un sifón que desagua el recipiente a un depósito y el flotante retorna a su posición inicial, para luego volver a subir si la lluvia continúa.

El sistema inscriptor está formado por un elevador con una plumilla unida al flotador, cuando este sube o baja lo hace también la plumilla. Los cambios de altura del agua son registrados en el pluviograma el cual se coloca sobre el tambor con sistema de relojería.

El pluviograma constituye la gráfica sobre la cual la plumilla del pluviógrafo registra la lluvia acumulada (León Méndez et al, 2013). La lectura del mismo indica la cantidad de precipitación acumulada cada determinado período de tiempo.

Los ejes de los pluviogramas marcan el tiempo (eje X) y la cantidad de lluvia (eje Y). El eje de tiempo puede ser de un día, una semana o un mes, de acuerdo con el mecanismo de relojería del pluviógrafo y las necesidades de precisión; mientras que el eje de la cantidad de lluvia tiene un máximo de 10 mm con una resolución de 0,1 mm. Sobre el pluviograma es que la plumilla va registrando en todo momento, de forma que la ausencia de precipitación queda reflejada como una recta horizontal y la presencia de lluvia, como una recta inclinada. A medida que aumenta la intensidad de la lluvia, esta recta se hace más inclinada y tiende a convertirse en una recta vertical, sin llegar a serlo. El pluviograma presenta como limitación la amplitud de registro de 10 mm de lluvia acumulada, que corresponde al punto de vaciado del volumen acumulado de lluvia en el recipiente. Este vaciado queda representado en la cartilla como una recta vertical o casi vertical pero de pendiente negativa, lo que se ilustra en la Figura 1 para el caso de una tormenta real.



**Figura 6. Pluviógrafo de sifón.**

Los pluviogramas son gráficas continuas que pueden representar la información correspondiente a una semana o a un mes dependiendo del pluviógrafo. Para poder interpretar un pluviograma, se requiere de personal capacitado que examina visualmente las gráficas y extrae los totales de precipitación diaria y las intensidades máximas de precipitación en periodos cortos de tiempo (de unos pocos minutos a horas). En la Universidad de Cagliari desarrollaron un software que realiza el procesamiento de pluviogramas de manera automática basado en procesamiento de imágenes, el cual obtiene las imágenes escaneadas y digitaliza la señal para obtener una base de datos de registros de precipitación. El desarrollo de este software tuvo lugar dentro de un proyecto financiado por el programa operativo de la Unión europea "Italia-Francia Marítimo", la

Agencia Regional de Protección Ambiental de Cerdeña y la Universidad de Cagliari (Delitala, Deidda, Mascaro, Piga, & Querzoli, 2010)

### ***Usos del pluviógrafo.***

El pluviógrafo es un instrumento meteorológico empleado en el estudio y análisis de las precipitaciones. Con él, se puede determinar la intensidad de la lluvia durante largos periodos de tiempo (León Méndez, Hernández González, Garrido Monagas, & Andalia Gilbert, 2013). El pluviógrafo de sifón, también llamado tipo Hellman, fue instalado por primera vez en Uccle en el año de 1898 (Vannitsem & Naveau, 2007). El pluviógrafo es utilizado principalmente para analizar tormentas, este análisis está íntimamente relacionado con los cálculos o estudios previos al diseño de obras de ingeniería hidráulica como pueden ser: estudio de drenaje, Determinación de caudales máximo, que deben pasar por el aliviadero de una represa, o que deben encausarse para impedir las inundaciones, determinación de la luz de un puente, conservación de suelos, cálculo del diámetro de alcantarillas, entre otros.

### **Medir la intensidad de la precipitación**

La precipitación atmosférica de gotas de agua en estado líquido tienen, generalmente, un diámetro igual o mayor que medio milímetro, y caen, en el aire en calma, con una velocidad superior a 1 m/s pudiendo sufrir disgregación secundaria si los diámetros son muy grandes y las velocidad de caída supera los 2 m/s. (León Méndez, Hernández González, Garrido Monagas, & Andalia Gilbert, 2013) Como se mencionó anteriormente, la intensidad de la precipitación se expresa como lámina por unidad de tiempo y es práctica internacional emplear milímetros por hora (mm/h) para cuantificarla, dato utilizado en diversos estudios agronómicos, hidrológicos y de conservación y erosión de suelos.

La precipitación y sus intensidades se pueden clasificar en:

- **Débiles:** intensidad menor o igual que 2 mm/h.
- **Moderada:** intensidad mayor que 2 mm/h y menor o igual que 15 mm/h.
- **Fuerte:** intensidad mayor que 15 mm/h y menor que 30mm/h.
- **Muy fuertes:** intensidad mayor que 30 mm/h y menor o igual que 60 mm/h.
- **Torrenciales:** intensidad mayor que 60 mm/h

### Calcular el coeficiente de escurrimiento.

El pluviógrafo es utilizado para calcular el coeficiente de escurrimiento en el manejo de cuencas. Una cuenca es una zona de la superficie terrestre en donde las gotas de lluvia que caen sobre ella tienden a ser drenadas por el sistema de corrientes hacia un mismo punto de salida; el manejo de cuencas consiste en controlar la descarga del agua captada por las cuencas en cantidad, calidad y tiempo de ocurrencia.

### Obtener la erosividad de la lluvia sobre el suelo.

Una utilidad más del pluviógrafo es que a través de sus cálculos se puede obtener la erosividad de la lluvia sobre el suelo. Lo que ocasiona que las partículas del suelo se desprendan y/o se partan en pedazos más pequeños es el impacto de la gota de lluvia sobre las partículas del suelo. En términos científicos lo que ocasiona ese daño sobre el suelo es la energía cinética de la lluvia. Estudios sobre el efecto de la lluvia sobre el suelo indican que la erosividad de la lluvia para cualquier mes puede ser medida con la siguiente expresión:

$$R = \sum_{i=1}^n Eci^{30} \quad (1)$$

Dónde:

R= Erosividad de la lluvia para cualquier mes.

Ec = Energía cinética de la lluvia.

I<sub>30</sub> = Intensidad de la lluvia

n= Número de tormentas en el mes.

La erosividad de la lluvia es variable de acuerdo al mes y de acuerdo al lugar, lo cual por supuesto depende de la lámina precipitada, y aún más, depende de la intensidad con la que llueve.

Se aclara que algunos autores reportan que se consideran tormentas o eventos diferentes cuando están separadas por periodos mayores a 6 horas. Lo que normalmente se hace es que se calcula la energía cinética que ocurrió durante los minutos de la intensidad máxima de la tormenta (E<sub>I<sub>30</sub></sub>) para cada tormenta, y después se suman estos valores para obtener el valor total del mes. Una tormenta o evento puede durar desde unos minutos hasta varias horas, y durante el mismo las intensidades con las que llueve puede variar considerablemente, por lo que para cada tormenta *i*, se divide en varios tramos *j* y para cada tramo *j* tenemos:

- Una diferencial de tiempo, dada en minutos.
- Una diferencial de lámina precipitada, dada en mm
- Una intensidad, dada en mm/hr
- La Energía cinética *Ec<sub>j</sub>*.

Las ecuaciones básicas para obtener la energía cinética de cada uno de los tramos *j* en que se divide el evento son:

$$e_j = \begin{cases} 0.119 + 0.0873 \log_{10} I_j & \text{si } I_j < 76 \text{ mm/hr} \\ 0.283 & \text{si } I_j > 76 \text{ mm/hr} \end{cases} \quad (2)$$

Dónde:

*e<sub>j</sub>* = Energía cinética por cada milímetro de lluvia para el intervalo de tiempo *j*

$I_j$  = intensidad de la lluvia en el intervalo  $j$

Con esta información se calcula la energía cinética en los 30 minutos más intensos de cada tormenta (EI30) de la siguiente manera:

1. Se identifica la tormenta total. Es decir a qué hora inicia la tormenta y a qué hora termina, recordando que para que se consideren tormentas diferentes debe transcurrir un tiempo de 6 horas entre el punto de la gráfica en que dejó de llover y el punto en comenzó registrar precipitación de nuevo.
2. En cada tormenta se identifican los tramos de lluvia, es decir, en qué puntos la pendiente (intensidad) cambia. Cada tramo será identificado como el  $j$ -ésimo tramo de la tormenta.
3. Para cada  $j$ -ésimo tramo se identifica el intervalo de tiempo ( $\Delta t$ ) y el incremento de lámina ( $\Delta l_j$ ). Con estos dos datos se calcula la intensidad en mm/hr y la energía cinética ( $e_j$ ) con la ecuación **¡Error! No se encuentra el origen de la referencia..** Finalmente la energía cinética ocasionada por la lámina  $\Delta l_j$  para el tramo  $j$  será:

$$E_{cj} = e_j * \Delta l_j \quad (3)$$

4. Se suman las  $E_{cj}$  de todos los tramos de la tormenta y eso origina la energía cinética total de la tormenta.

$$E_c = \sum E_{cj} \quad (4)$$

5. Se calcula la intensidad máxima en 30 minutos para la tormenta, de la siguiente manera:
  - a. Se identifica en que tramo  $j$  de la tormenta la intensidad fue mayor.
  - b. Se toma el  $\Delta t$  que sucedió en el tramo  $j$  de mayor intensidad
  - c. Si el tiempo resultante es menor a 30, se identifica cual tramo es más intenso, si el que le antecedió o el que le sucedió.
  - d. Se toma el nuevo  $\Delta t$  y se suma al anterior.
  - e. Si el tiempo resultante es menor a 30 se repite el paso c.



- f. Si el tiempo resultante sobrepasa los 30 minutos, el último tramo elegido se divide para tomar únicamente los minutos necesarios para ajustar 30 minutos.
6. Se calcula el  $EI_{30}$  con la siguiente fórmula:

$$EI_{30} = E_c * I_{30} \quad (5)$$

7. Una vez calculado el  $EI_{30}$  para cada tormenta se calcula la erosividad del mes (R) sumando todos los  $EI_{30}$  de todas las tormentas registradas en el mes, como se muestra en la ecuación ((1)).

En la **¡Error! No se encuentra el origen de la referencia.** se muestra la información obtenida de procesar el segmento de pluviograma mostrado en la Figura 7.

Tabla 1. Datos obtenidos de un pluviograma

Tiempo(hh:mm)	Lamina(mm)	$\Delta t$	$\Delta l$	$I(\text{mm/hr})$	$e_j$	$E_{c_j}$
15:39	0	0	0	0		
15:45	6	6	6	60	0.27	1.62
15:47	6.6	2	0.6	18	0.23	0.138
15:55	7.1	8	0.5	3.75	0	0
16:37	8.5	42	1.4	2	0.15	0.203
17:45	8.65	68	0.15	0.13	0.04	0.006
18:09	8.9	24	0.25	0.63	0.10	0.025
<b>Suma</b>		150	8.9			2.016

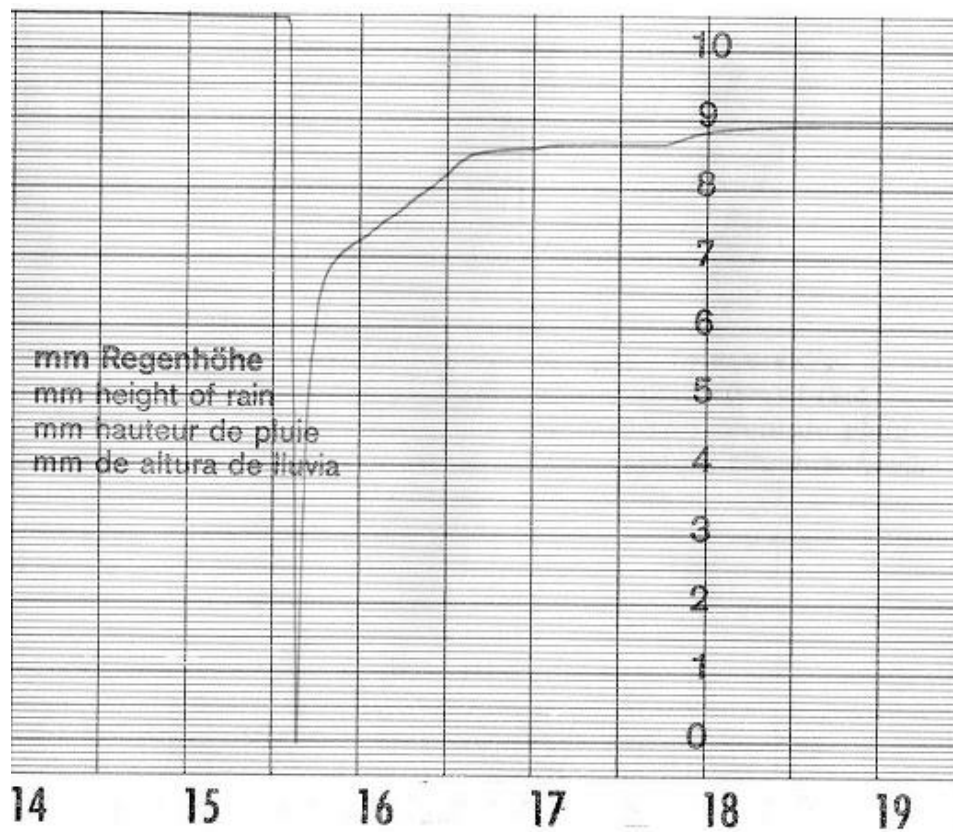


Figura 7. Segmento de un pluviograma

En la **¡Error! No se encuentra el origen de la referencia.** se puede observar que el evento constó de 6 tramos, comenzó a las 15:39 y el primer tramo ocurrió desde esa hora hasta las 15:45, en ese lapso llovieron 6 mm y ocurrió en un lapso de 6 minutos, para calcular la intensidad (I) es necesario convertir los minutos en horas.

Después calculamos la energía cinética por mm (ej) siguiendo la formula presentada en **¡Error! No se encuentra el origen de la referencia..**

$$e_j = 0.119 + 0.0873 \log_{10} 60 = 0.27$$

y la energía cinética total para ese tramo:

$$E_{c_j} = e_j * \Delta_j \Rightarrow 0.27 * 6 = 1.62$$

El segundo tramo ocurrió entre las 15:45 y 15:47, y en ese punto la gráfica marcaba 6.6 mm de lluvia, lo que indica que en este segundo tramo llovieron solo 0.6 mm. Se realizan los mismos cálculos que en el tramo anterior y así para todos los tramos identificados. Al final la Energía cinética (Ec) es la suma de todas las Ecj del evento, en este caso fue de 2.016. Después se calcula la intensidad máxima en 30 minutos (I30):

La intensidad máxima registrada fue 60, y ocurrió en 6 minutos, si se hubiera dado el caso que tuviera una intensidad anterior se compararía con la que le sucede para saber si fue más intenso el tramo predecesor o el sucesor, en este caso siempre es el sucesor. En la **¡Error! No se encuentra el origen de la referencia.** se muestran la lista de tramos de la tormenta donde ocurrió la intensidad máxima de la misma.

**Tabla 2. Identificación de los 30 minutos más intensos de la tormenta**

minutos	Lámina	I
6	6	60
2	0.6	18
8	0.5	3.75
42	1.4	2

El tramo de 42 minutos será dividido mediante una regla de tres para tomar únicamente la lámina llovida correspondiente a los 14 minutos necesarios para completar los 30 minutos más intensos de la tormenta. El ajuste se muestra en la **¡Error! No se encuentra el origen de la referencia.**

**Tabla 3. Ajuste a los 30 minutos más intensos de la tormenta**

minutos	Lámina
6	6
2	0.6
8	0.5
14	0.46

La lámina llovida en los 30 minutos más intensos de la tormenta fue de 7.56 mm. Esta cantidad se multiplica por 2, para así tener el equivalente a una hora, y con el valor de  $E_c$  obtenido, se procede a calcular el  $E_{I30}$  de la tormenta. Al sumar los valores de  $E_{I30}$  de todas las tormentas ocurridas en un mes, obtenemos el valor de la erosividad correspondiente.

### ***Errores en la medición***

Partiendo del hecho de que toda medición tiene una posibilidad de error, es necesario conocer cuales pueden ocurrir al trabajar con un equipo de medición dado para así determinar la forma de corregirlo. En la medición de las precipitaciones hay que tener en cuenta, en primer lugar, los errores inherentes a la lluvia y al lugar donde se realiza la medición, para luego analizar los errores propios del equipo colector empleado. Por último, es necesario considerar los errores producidos por el factor humano por ser los más perjudiciales y por lo tanto los más importantes de erradicar. El principal error es debido a las características de la forma de caída de la lluvia. La gran mayoría de las veces

la lluvia no cae en forma vertical, sino que es fácilmente desviada por el viento, dejando de ser perpendicular a la boca del equipo colector. La causa de los errores más frecuentes derivados del lugar donde se ubica el equipo de medición, radica en la cercanía de edificios, árboles u otros obstáculos próximos al aparato. Para minimizar los errores producidos por estos factores, se establecen normas o condiciones para la instalación de los aparatos.

Los errores fundamentales propios de los aparatos son: la pérdida por mojadura de las paredes del colector, errores por las salpicaduras hacia dentro o hacia fuera del aparato y errores debidos a la evaporación del depósito de agua del aparato (característica muy importante cuando se instala el aparato en regiones de clima cálido). Estos errores se pueden minimizar en el diseño del nuevo aparato, pero se deben seguir contemplando al momento de hacer las mediciones. También es frecuente la pérdida de información por atasco de papel en el sistema de inscripción, este error se va a erradicar al implementar el sistema electrónico para realizar las mediciones.

Por último, tenemos los errores propios de la medición hecha por el hombre. El sistema de registro de los pluviógrafos, que es el pluviograma, tiene una sensibilidad de 0,1 mm, y las precipitaciones se obtienen a partir de los cambios de pendiente de la gráfica y queda sujeto a la apreciación del observador o interpretador de pluviograma lo que introduce una fuente de incertidumbre. La mejor vía para eliminar estos errores humanos es registrar los datos de precipitación de forma digital en un archivo tipo texto.

# ***Capítulo 2: Análisis y desarrollo del hardware.***

En este capítulo se muestran los elementos con que se automatizará el pluviógrafo de tipo thies realizado a partir de los requisitos de diseño. Se exponen los aspectos teóricos más relevantes y el funcionamiento de los dispositivos utilizados en el desarrollo del prototipo.

## ***Arduino***

### ***¿Qué es Arduino?***

Arduino es una plataforma de hardware libre para construir proyectos de electrónica. Se basa en una sencilla placa electrónica con entradas (o pines) analógicos y digitales de E/S, un microcontrolador Atmel. (Wilcher D. , 2012) y un entorno de desarrollo Integrado (IDE). Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, aunque desde Octubre de 2012 las placas de Arduino trabajan también con microcontroladores CortexM3 que por su sencillez y bajo coste permiten el desarrollo de múltiples diseños.

### ***¿Para qué sirve?***

La placa de Arduino es capaz de tomar información del entorno a través de sus entradas analógicas y digitales, controlar luces, motores y otros actuadores. Con la placa Arduino se pueden desarrollar proyectos interactivos, autónomos donde el microcontrolador de la placa funcione como el cerebro del objeto y realiza el procesamiento de la información

obtenida del entorno, a través de sensores y otros dispositivos electrónicos, o pueden comunicarse con algún software que se ejecute en un ordenador (Flash, Processing, MaxMSP) o con algún lenguaje de programación de alto nivel (Java, Matlab).

### ***Modelos de placas Arduino.***

El proyecto Arduino ha adquirido una gran popularidad en los últimos años, tanto para proyectos escolares como para grandes proyectos de automatización. Esto ha impulsado el desarrollo de diferentes modelos de la placa de Arduino, con diferentes características físicas y lógicas que se adaptan a las necesidades de cada proyecto. De esta forma podemos tener, además de una placa básica, extensiones que permiten la comunicación en red vía Ethernet, WI-FI, bluetooth o GSM, o placas de Arduino más sofisticadas capaces de interactuar con el sistema operativo móvil Android (Evans, 2011).

A continuación se enlistan los diferentes modelos de placas Arduino existentes en el mercado (Arduino, Arduino, 2013):

- Arduino Uno.
- Arduino Leonardo.
- Arduino Due.
- Arduino Yún.
- Arduino Tre.
- Arduino Micro.
- Arduino Robot.
- Arduino Esplora
- Arduino Mega ADK.
- Arduino Ethernet.
- Arduino Mega 2560
- Arduino Mini
- LilyPad Arduino USB.
- LilyPad Arduino Simple.
- LilyPad Arduino SimpleSnap.
- Arduino nano.
- Arduino pro.
- Arduino Fio.

Existen placas que pueden ser conectadas encima de una placa Arduino para extender sus capacidades, y las más utilizadas son (Arduino, Arduino, 2013):

- Arduino GSM Shield.
- Arduino Ethernet Shield.
- Arduino WiFi shield.
- Arduino Wireless SD Shield
- Arduino USB Host Shield.
- Arduino Proto Shield.

En el desarrollo del prototipo de este trabajo de investigación se hará uso de la placa Arduino Uno (Figura 8), la cual tiene como microcontrolador base un ATmega328 de la marca Atmel (Atmel, 2013). Este microcontrolador tiene una resonancia de 16 MHz, 32 KB de almacenamiento y 2 KB de memoria de acceso aleatorio (Evans, 2011). Este modelo puede ser alimentado mediante una conexión USB o con una fuente de poder externa. Posee 14 pines digitales los cuales pueden ser configurados como entrada o como salida, y proveer o recibir un máximo de 40mA (Banzi, 2011).

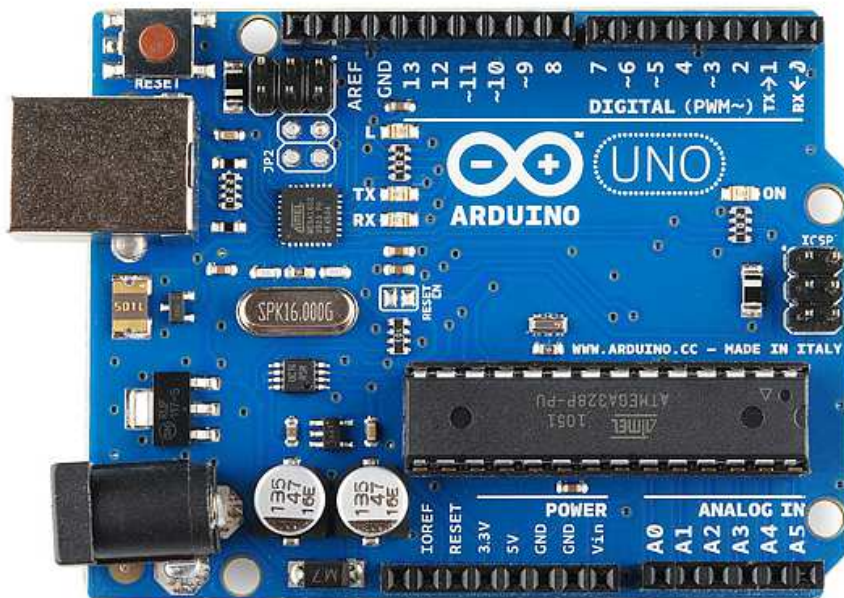


Figura 8. Placa Arduino Uno



## ***Ventajas sobre otros microcontroladores.***

Actualmente existen muchos microcontroladores y plataformas para la programación de hardware, la cual ha venido evolucionando a través de los años. Existen plataformas como Parallax Basic Stamp, entre otras, que ofrecen funcionalidades similares. Estas herramientas simplifican la complicada tarea de programar un microcontrolador, ya que no es necesario poseer una tarjeta grabadora que contenga las librerías necesarias para comunicarse con la memoria EEPROM del microcontrolador, porque su lenguaje de programación es capaz de comunicarse con la tarjeta haciendo uso únicamente de la interfaz USB. Arduino, además de proveer esta funcionalidad, tiene ciertas características que le dan ventaja sobre otras plataformas similares (Wilcher D. , 2012):

- Asequible. Comparadas con otras plataformas de microcontroladores, las placas de Arduino son más accesibles económicamente.
- Multiplataforma. El software de Arduino funciona en los sistemas operativos de Windows, Macintosh OSX y Linux.
- Software ampliable y de código abierto. El software Arduino está publicado bajo licencia libre por lo que puede ser ampliado por programadores experimentados.

## ***Plataforma Arduino.***

Massimo Banzi, cofundador de Arduino, (Banzi, 2011) señala que Arduino está compuesto por dos partes principales: La placa de Arduino, que corresponde al hardware donde se pueden construir prototipos y el IDE el cual es el software que se ejecuta en la computadora y con el que podemos crear programas (sketch) que pueden ser transferidos al microcontrolador de la placa Arduino.

## ***Lenguaje de Programación.***

La plataforma Arduino posee su propio lenguaje de programación, el núcleo de éste es el lenguaje de programación C desarrollado por primera vez en el instituto de investigación de los Laboratorios Bell en la década de 1970 para su uso con el sistema operativo UNIX (Evans, 2011). El lenguaje de programación de Arduino tiene la propiedad de ser multiplataforma y posee una biblioteca estándar que proporciona un conjunto de instrucciones y funciones que permiten al programador explotar al máximo el potencial de la placa Arduino. La mayor parte del código que se escribe para la placa de Arduino incluye la sintaxis, estructura, operadores, sentencias de control y funciones es igual al que funcionalmente encontramos en el lenguaje C; Sin embargo, existen funciones específicas propias de la biblioteca estándar de Arduino, entre ellas: `pinMode()`, `digitalWrite()` y `delay()`. (Banzi, 2011).

## ***Sketch de Arduino.***

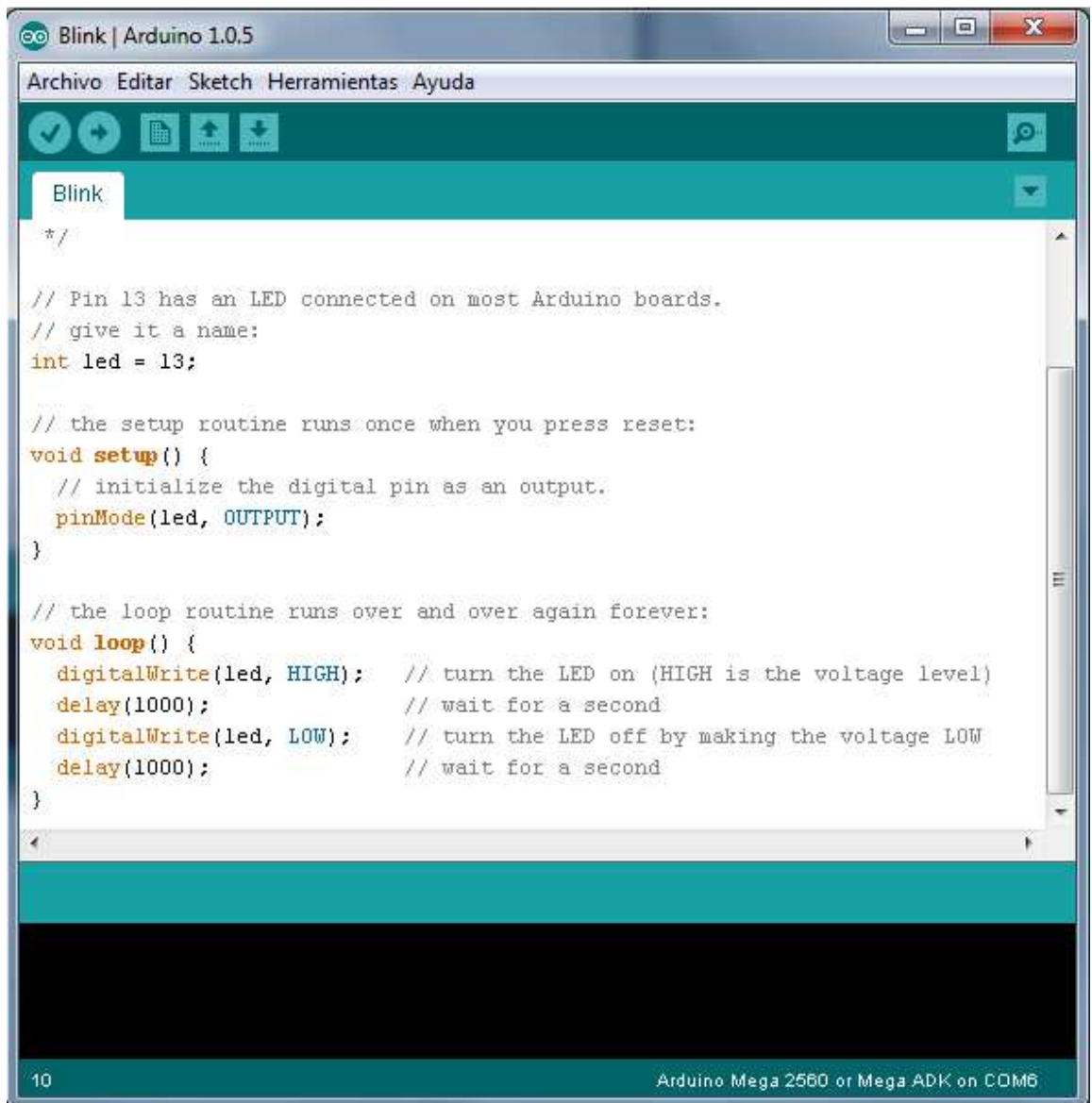
Cada archivo que contiene código de arduino es llamado sketch (Arduino, Arduino Sketch, 2013), cada archivo de sketch puede ser escrito en un Entorno de desarrollo de Arduino como el que se muestra en la Figura 9 y es almacenado con extensión `.ino`.

Los sketches empiezan con la inclusión de bibliotecas, posteriormente se declaran las variables y métodos.

Todos los programas incluyen un método "SETUP" que prepara como puerto de entrada o salida los pines de la placa arduino, y se establecen todos los valores con que debe iniciar cada proceso el microcontrolador.

Finalmente, el método "LOOP" se ejecuta una y otra vez mientras la tableta esté conectada a la corriente eléctrica. Este método revisa con cada ejecución el estado de los sensores y desencadena las acciones definidas por el programador. Este es el punto en el que se invoca a otros métodos cuando una condición se cumple.

En la Figura 9 se muestra el código necesario para encender un led de manera intermitente. En el método setup se establece que el pin 13 sea un puerto de salida de corriente. Un led conectado a la entrada 13 de la tableta Arduino se enciende, transcurre un segundo, se apaga y transcurre otro segundo. Este flujo se realizaría de manera indefinida.

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0.5". The menu bar includes "Archivo", "Editar", "Sketch", "Herramientas", and "Ayuda". Below the menu bar is a toolbar with icons for a checkmark, a right arrow, a grid, an upload arrow, and a download arrow. The main workspace shows a sketch named "Blink" with the following code:

```
*/  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```

The status bar at the bottom left shows the number "10" and the bottom right shows "Arduino Mega 2560 or Mega ADK on COM6".

Figura 9. IDE de Arduino

Los programas al ejecutarse generan una salida al monitor serial que es una ventana independiente mostrada en primer plano (ver Figura 10). En la parte inferior derecha del monitor serial se establecen los Baudios a los que operan los sensores que envían mensajes a la PC.



Figura 10. Monitor serial.

### ***Áreas de investigación donde se ha utilizado Arduino***

La plataforma Arduino ha sido utilizada en la educación, apoyando el proceso de enseñanza-aprendizaje de electrónica, tal ha sido el impacto en la educación tecnológica que ya se han desarrollado algunos proyectos en los que se desarrollan excelentes simuladores de la plataforma; con el fin de facilitar el trabajo en equipo y la interacción entre los usuarios inexpertos y ciertos elementos de hardware como leds, potenciómetros, servomotores, entre otros (Zavala, Morán, & Blanco, 2010).

La popularidad de la plataforma Arduino ha motivado el desarrollo de interfaces y frameworks que permiten apoyar en el proceso de la investigación científica tal es el caso de Instrumentino (Koenka, Saiz, & Hauser, 2014) un framework con Interfaz Gráfica de Usuario (GUI por sus siglas en inglés) desarrollado en lenguaje Python por investigadores de la Universidad de Basilea, Suiza y la Universidad de Alcalá en Madrid, España para

monitorear y controlar experimentos basados en Arduino, los cuales ya se desarrollan en diversos laboratorios al rededor del mundo.

En la robótica, además de apoyar la educación con el desarrollo de robots educativos (Balogh, 2011), presenta grandes desarrollos cuyo sistema principal es la placa de Arduino. Cuando hablamos de robots, la gente tiende a pensar que los robots son únicamente adecuados para su uso en la industria o en el campo científico. Sin embargo, la función principal de un robot es la de ayudar a los seres humanos; los robots cada vez se encuentran más integrados en reemplazar humanos en tareas que suelen ser repetitivas. La Federación Internacional de Robótica (IFR por sus siglas en inglés) define un robot de servicio como un robot que opera de forma automática o semiautomática para realizar servicios útiles para el bienestar de los seres humanos tal es el caso del brazo robótico presentado en (reza Ezuan & Babul Salam, 2012) capaz de ser controlado a través de internet y el cual fue desarrollado utilizando una placa de Arduino Uno y un Shield Ethernet para la comunicación en red. Paralelamente se desarrolló un brazo robótico capaz de ser controlado con tecnología wireless (Reza Ezuan & Babul Salam, 2012) el cual utiliza como cerebro una placa Arduino Mega 2560, un servomotor para mover el brazo robótico y un control de la consola de videojuegos Play Station 2 para controlar los movimientos del brazo robótico con la interfaz wireless del mismo. En la industria Arduino es ampliamente utilizado para tareas como controlar la velocidad de motores de inducción, los cuales son los más utilizados para electrodomésticos, control y automatización pero los cuales presentan problemas al modificar su velocidad según sus niveles de voltaje. Utilizando Arduino se pueden controlar la velocidad del motor de inducción generando la placa los impulsos de encendido necesarios en base a señales de entrada por parte del usuario hacia el programa de Arduino (Niranjan Kumar, Hima Bindu, Divya Sneha, & Sravani, 2013)

En el área de ingeniería, la plataforma Arduino ha servido como base para diferentes proyectos, se ha desarrollado un dispositivo capaz de analizar las redes Wi-Fi procesando datos tales como la intensidad de la señal y el tipo de encriptación entre otras, y mostrándolos en una pantalla LCD (Al-Kadi, 2013). El dispositivo utiliza una placa Arduino

Mega 2560 y un Arduino Wi-Fi shield para realizar el análisis de la red y un LCD/LED para mostrar la información.

En la investigación científica se han desarrollado proyectos que representan un gran avance en el área, y utilizan como motor principal en su diseño una placa Arduino. Por ejemplo, el desarrollo de un estimulador visual basado en light emitting diodes (LEDs) y una placa de Arduino Uno. La interfaz encargada de comunicar con la computadora se realizó en los lenguajes de programación Matlab y Python (Teikari, Najjar, & Malkki).

## ***Sensor de nivel de líquidos eTape.***

### ***Descripción y modo de operación.***

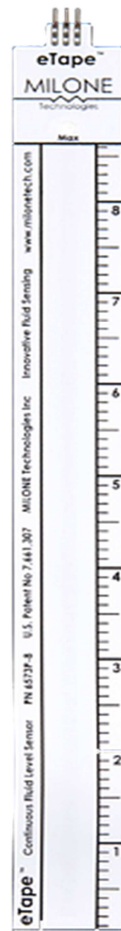
El sensor eTape (mostrado en la Figura 11) es un sensor de estado sólido cuya salida es una resistencia que varía con el nivel de líquido que este almacenado en el recipiente donde se encuentre instalado; esta salida es inversamente proporcional al nivel del líquido, mientras más bajo sea el nivel, la resistencia es mayor (Milone Technologies, 2012).

Es un sensor de nueva generación, se encuentra fabricado de PET, que es el mismo material del que están hechas las botellas de agua, haciéndolo menos susceptible a la corrosión y como es un sensor de estado sólido y no tiene partes móviles, su señal no se ve afectada por la afluencia de sal.

### ***Características.***

Existen diferentes tipos de sensores de nivel de líquidos, para poder elegir el adecuado para cada proyecto, hay que tomar en cuenta ciertas características como el tipo de

líquido que será medido, si el sensor puede ser insertado en el tanque que almacena el líquido, si el sensor puede entrar en contacto con el líquido procesado, entre otras.



**Figura 11. eTape sensor de nivel de líquidos.**

Por sus características físicas, el sensor eTape es el adecuado para la realización de este proyecto. Puede ser sumergido en el agua de lluvia e instalado en el recipiente colector de lluvia de la estructura.

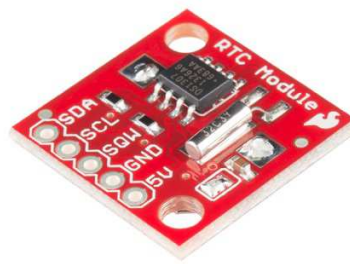
El sensor eTape presenta las siguientes especificaciones:

- Medida de largo: 257 mm.
- Medida de ancho: 25.4 mm.
- Medida de espesor: 0.381 mm.

- Área efectiva de medición: 213 mm.
- Salida cuando el tanque o recipiente está vacío: 1500  $\Omega$
- Salida cuando el tanque o recipiente está lleno: 300  $\Omega$

### ***Sensor DS1307 Real Time Clock.***

El DS1307 (ver Figura 12) es un módulo BCD de bajo consumo de energía con las funciones de reloj y calendario. Capaz de proveer horas, minutos, segundos, día de la semana, día del mes, mes y año. Cuando el contador llega al final del día, del mes o del año, la fecha se ajusta de modo automático incluso cuando se trata de un año bisiesto.



**Figura 12. Sensor DS1307 RealTime Clock**

### ***microSD Shield***

Las tarjetas de memoria con formato Secure Digital (SD) se encuentran en una amplia gama de dispositivos portátiles como cámaras fotográficas digitales, teléfonos inteligentes, laptops, e incluso consolas de videojuegos tanto de sobremesa como portátiles.

El estándar SD fue introducido en 1999 como una mejora evolutiva de las tarjetas multimedia. El estándar es mantenido por la SD Card Association y fue implementado en



más de 400 marcas de productos, cubriendo docenas de categorías y en más de 8.000 modelos.

En la plataforma Arduino una tarjeta microSD provee la posibilidad de almacenar grandes cantidades de información en un sistema de archivos así como recuperar información de la misma.

Para el desarrollo de este proyecto se utilizó el *microSD Shield* (Figura 13) el cual es un componente que se puede conectar a una placa de Arduino permitiendo que el microcontrolador de esta tenga acceso a una memoria microSD para el almacenamiento de datos.

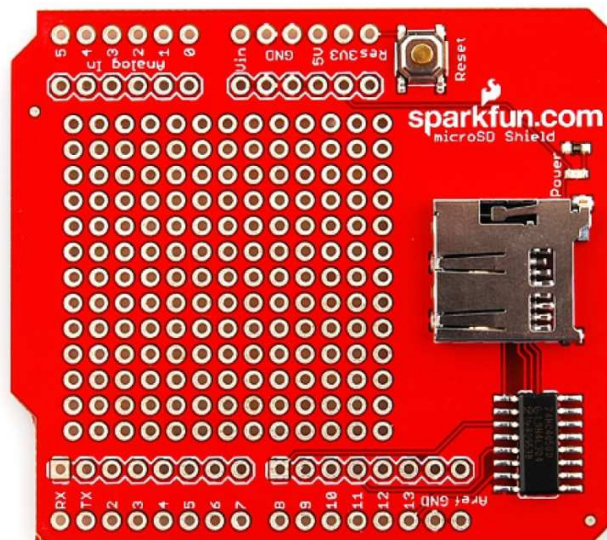


Figura 13. microSD shield.

## ***Diseño y desarrollo de estructura***

Se diseñó una estructura basada en un pluviógrafo tipo Hellman de la marca Lambrecht (véase Figura 14). La cual está constituida por las siguientes partes (véase Figura 15):

- **Brocal o anillo receptor.** Es la parte superior del instrumento, por donde se realiza la captación de la precipitación. Tiene un diámetro de 16 cm.
- **Sistema de relojería.** Es donde se coloca el papel para la gráfica, y lo va girando para que el brazo inscriptor dibuje la gráfica correspondiente a la cantidad de precipitación captada. Realiza un giro completo en 24 hrs.
- **Cámara de sifonaje.** Es donde se recolecta la precipitación captada por el brocal, posee un flotador interno que conforme va subiendo el nivel de agua va elevando el brazo inscriptor que a su vez va dibujando la gráfica.
- **Brazo inscriptor.** Está sujeto a la parte superior de la cámara de sifonaje, conforme asciende el nivel de agua dentro de ésta, el brazo inscriptor se va desplazando en la misma dirección, cuando la cámara de sifonaje se vacía el brazo inscriptor se desplaza hacia abajo, dibujando así las subidas y bajadas en el pluviograma.
- **Sifón.** Es un tubo que permite evacuar el agua contenida en la cámara de sifonaje, cuando ésta alcanza su nivel máximo.
- **Colector.** Es donde se almacena toda la precipitación recolectada durante el periodo de medición.

Para obtener una forma cilíndrica para el instrumento, la estructura se desarrolló a base de tubos de PVC, con dicho material se minimizan los costos de producción y se tiene una estructura sólida, resistente a la intemperie, que no se verá afectada por la temperatura, humedad, o las condiciones climáticas del sitio donde se instale el instrumento.

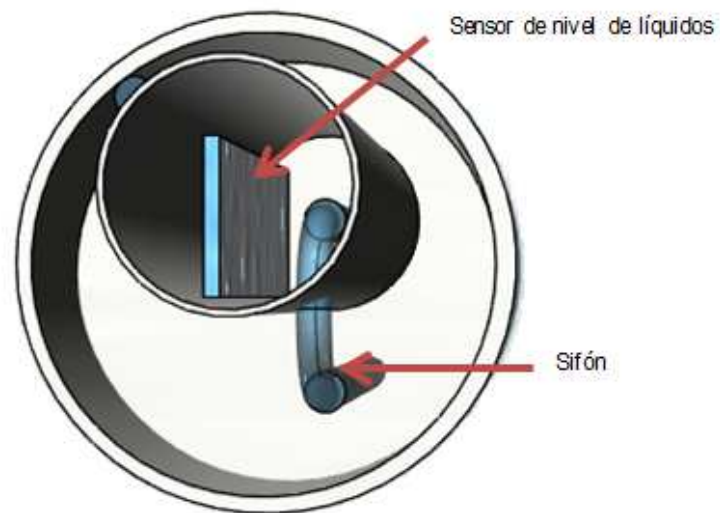


Figura 14. Pluviógrafo tipo Hellman marca Lambrecht

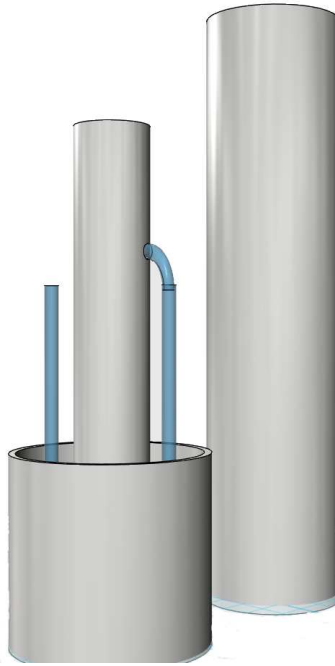


Figura 15. Partes que constituyen un pluviógrafo.

El sensor de nivel de líquidos se instaló dentro de la cámara de sifonaje, la cual se construyó con un tubo de PVC de 40 mm de diámetro y el sistema de sifón se construyó con un tubo de cristal curvo, instalado también dentro de la cámara de sifonaje (ver Figura 16), cuando el agua alcanza su nivel máximo permitido, que es la altura de un extremo del sifón, comienza a salir por el otro extremo hacia el colector, el cual es un pequeño recipiente fabricado también de PVC y una base, éste recipiente se integra a la estructura y es la parte inferior del instrumento . La cámara de sifonaje tendrá una cubierta para aislarla del exterior (véase Figura 17).



**Figura 16. Cámara de sifonaje**



**Figura 17. Cámara de sifonaje y cubierta vista lateral**

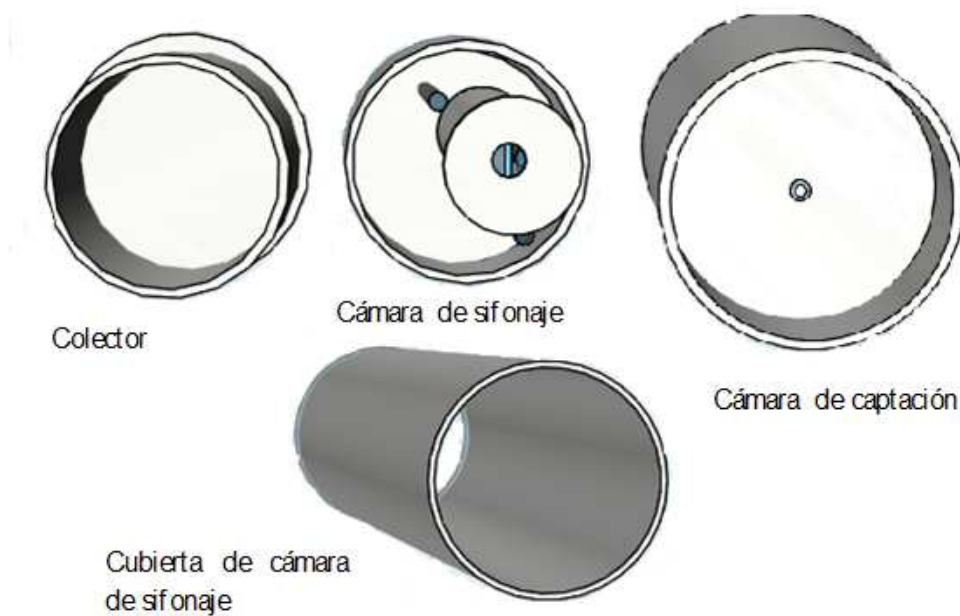
Para la parte superior del instrumento, encargada de coleccionar la precipitación pluvial, se diseñó una cámara de captación (Figura 18), con un tubo de la misma medida que la cubierta de la cámara de sifonaje, se le colocó una base, dejando únicamente un orificio para la captación de precipitación que conduce mediante un tubo de plástico el agua recolectada hacia la cámara de sifonaje.



**Figura 18. Cámara de captación.**

En la Figura 19 se muestra por separado cada uno de los elementos que constituyen la estructura final del aparato y en la Figura 20 como lucirá la estructura con todos los componentes integrados.

En la Figura 21 se muestra el diseño de como lucirá internamente la estructura final del aparato, en ella se describe donde quedará instalado el sensor. La lluvia entrará por la cámara de captación y se guiará hacia la cámara de sifonaje que es donde estará instalado el sensor. Cuando la precipitación pluvial alcance el nivel máximo en la cámara de sifonaje, esta se desalojará hacia el colector.



**Figura 19. Diseño del pluviógrafo**



**Figura 20. Estructura final del instrumento.**

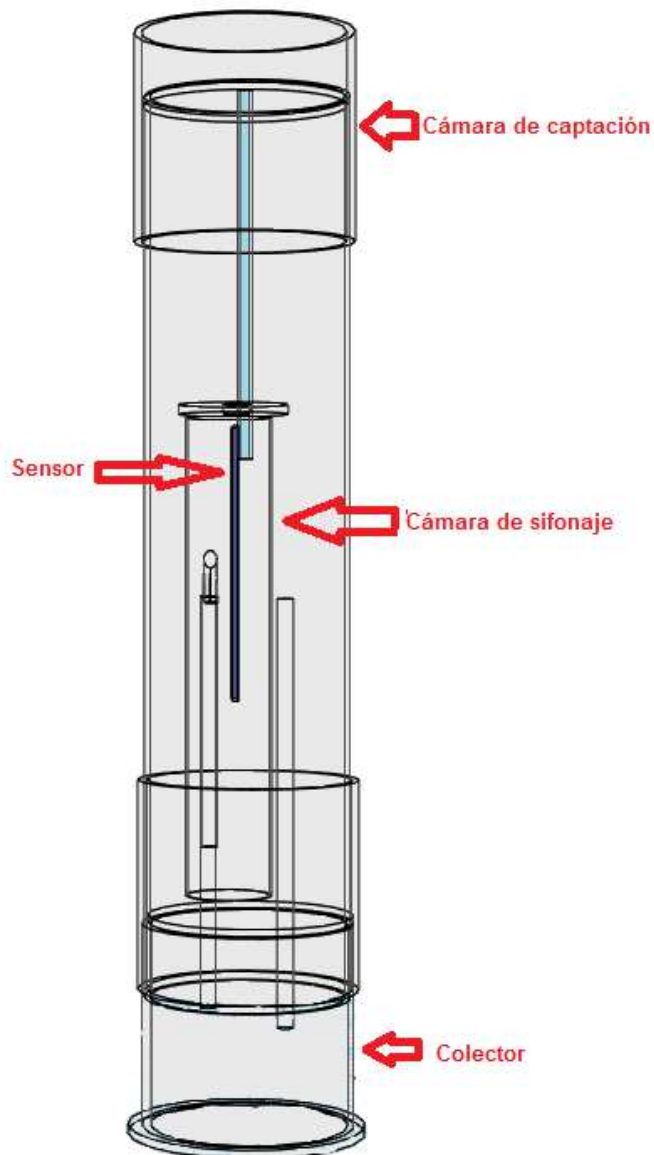


Figura 21. Vista interior del instrumento

El funcionamiento del circuito electrónico se describe en el diagrama de bloques de la Figura 22 y está basado en la interacción de los siguientes componentes:

- **Placa Arduino Uno.** Es una plataforma de hardware libre, se basa en una placa electrónica con entradas (o pines) analógicos y digitales de E/S, un microcontrolador Atmel y un entorno de desarrollo integrado. (Wilcher D. , 2012).



- **Sensor de nivel de líquidos eTape Millone.** Es un sensor de estado sólido cuya salida es una resistencia que varía con el nivel de líquido que este almacenado en el recipiente donde se encuentre instalado; esta salida es inversamente proporcional al nivel del líquido, mientras más bajo sea el nivel, la resistencia es mayor.
- **microSD shield.** Es una placa que brinda al Arduino comunicación con una microSD para almacenar información.
- **Real Time clock DS1307.** Este módulo se encarga de llevar un control del tiempo.
- **Pantalla Liquid Crystal Display (LCD).** Con una capacidad para mostrar 32 caracteres en dos filas. permitirá la interacción con el usuario.

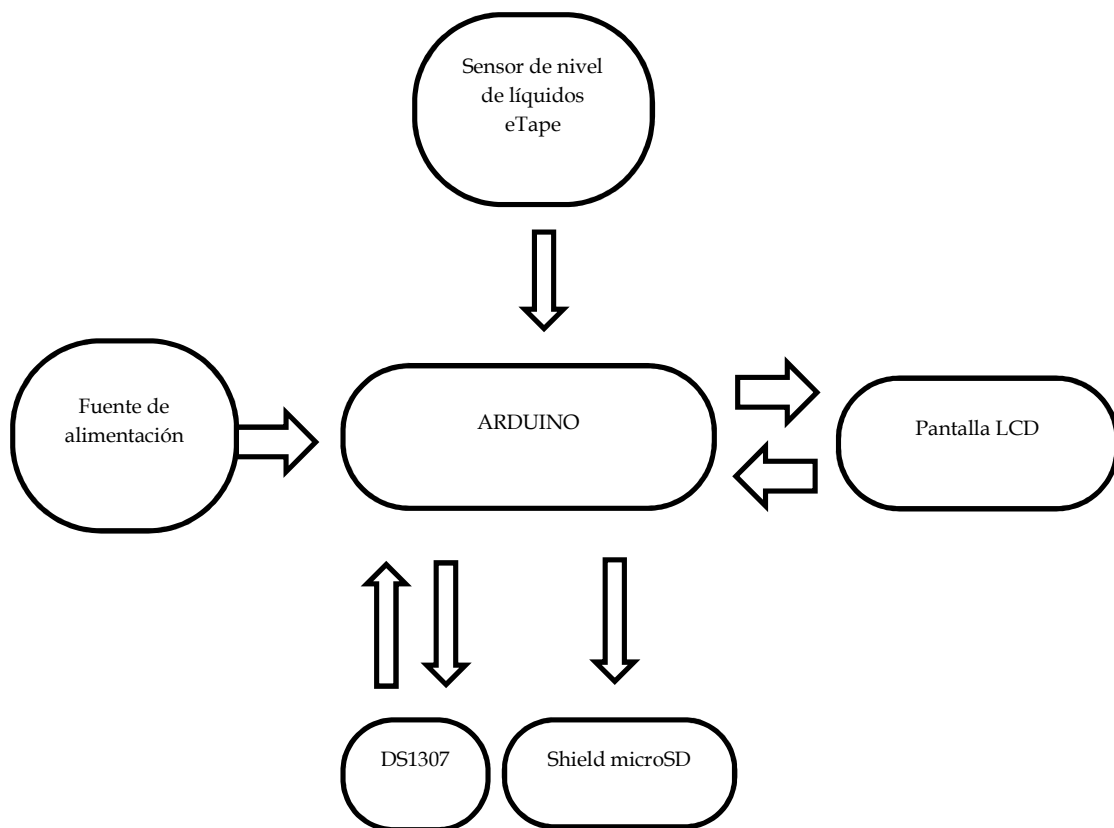


Figura 22. Diagrama de bloques de la plataforma

La plataforma Arduino se ha utilizado como base en diversos trabajos de investigación tanto en ingeniería, robótica, Inteligencia Artificial y en diversas áreas de la investigación científica. En este trabajo, Arduino realiza las funciones de control y procesamiento de la información captada por el sensor de nivel de líquidos. Conforme el nivel de agua va ascendiendo en la cámara de sifonaje, el sensor de nivel de líquidos emite una resistencia la cual se mide por una de las entradas analógicas de la placa de Arduino, donde se procesa para determinar el nivel de agua que presenta el instrumento, si existe una variación, se crea un archivo de texto con los datos correspondientes además de la fecha y hora de registro, las cuales pueden ser ajustadas por el usuario utilizando la interfaz de la pantalla LCD (véase Figura 23).



**Figura 23. Pantalla LCD.**

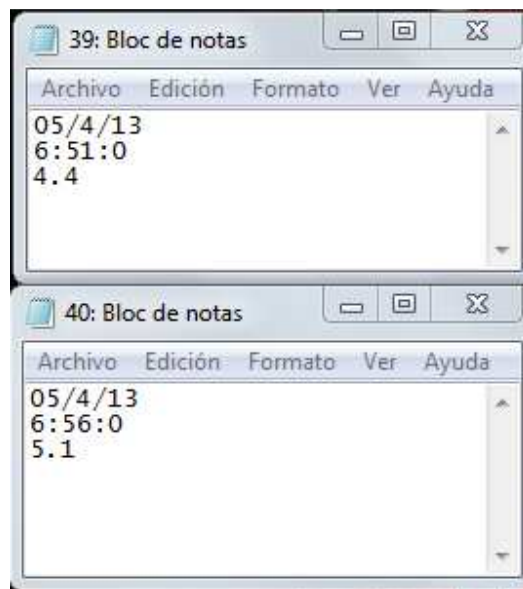
### ***Funcionamiento del circuito electrónico del instrumento.***

En el Anexo A se muestra el sketch programado y se describen los métodos que lo componen y su funcionamiento, y en la Figura 25 se muestra el diagrama de flujo donde

se describe el funcionamiento del circuito electrónico del instrumento, como se puede observar en el diagrama, el funcionamiento comienza cuando se alimenta el circuito por primera vez, se ajustan los valores necesarios, y no contiene un final ya que el instrumento se encontrará operando hasta que se deje alimentar de energía.

Hay que recordar que a mayor nivel de agua almacenado en el instrumento, menor es la lectura que brinda el sensor, si revisamos el diagrama de la Figura 25 cada que la lectura actual es menor que la lectura anterior significa que se tiene una variación en el nivel de agua captada por el instrumento, en ese momento el circuito debe almacenar información en la memoria microSD, en ella se guarda un archivo de texto con la fecha, hora, y cantidad de mm de lluvia que tenía el instrumento al momento realizar la lectura del sensor. Cada archivo de texto se numera a partir de 0 para llevar una secuencia.

En la Figura 24 se muestran dos archivos de datos generados en una de las pruebas del instrumento, allí podemos ver claramente que a las 6:51 hrs se tenía el equivalente a 4.4 mm de lluvia y a las 6:56 horas el nivel se ya había elevado a 5.1 mm de lluvia.



**Figura 24. Archivos de datos del instrumento**

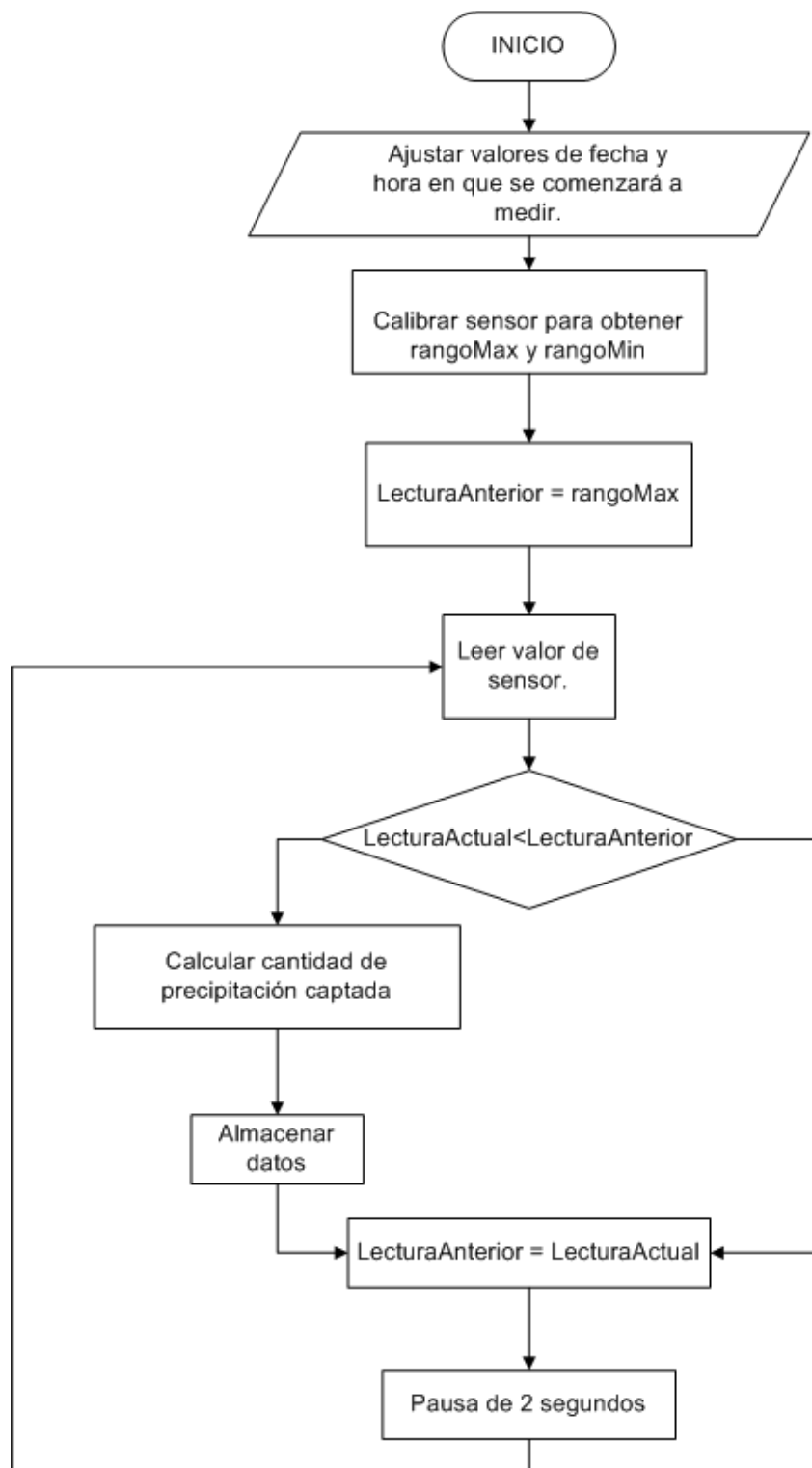


Figura 25. Diagrama de flujo de funcionamiento de circuito.

# ***Capítulo 3: Análisis y desarrollo del software***

Como parte fundamental de esta plataforma, se desarrolló un software para la interpretación de los datos procesados por el circuito electrónico. La primera fase de este proceso fue determinar una metodología a la cual someter el proceso de desarrollo del software.

*En la actualidad existen propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo de software. Por una parte existen propuestas tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado inconvenientes en muchos otros. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados, sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso frenar el progreso del proyecto en lugar de agilizarlo. Otra filosofía es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software, este es el objetivo de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones cortas. (Reingart, 2012)*

Para el desarrollo de este trabajo se eligió la metodología Personal Software Process (PSP) por centrarse en la gestión del trabajo individual de los programadores (von Kinsky, Ivins, & Robey), y es la base para la metodología Team Software Process encargada ya de coordinar un equipo de trabajo (Rodríguez Vazquez, 2013).

## ***Personal Software Process.***

El PSP es una metodología que se comenzó a desarrollar en 1989 por Watts Humphrey (Capra, Escala, & Morisio) en el Software Engineering Institute (SEI) de la Universidad Carnegie Mellon. Consta de un conjunto de prácticas disciplinadas para la gestión del tiempo (Salinas, Cerpa, & Rojas, 2011) y mejora de la productividad personal de los programadores o ingenieros de software, en tareas de desarrollo de software, incluyendo el desarrollo de programas pequeños, definición de requerimientos, tareas de documentación de programas, pruebas de sistemas, y mantenimiento de grandes sistemas. La metodología PSP se compone de siete versiones, cada una introduce nuevos elementos que son registrados en diversos formularios y cada una tiene ciertas prácticas claves que van reflejando la mejoría en la calidad del producto de software a medida que se avanza de versión. A continuación se presentan las siete versiones que componen la metodología PSP.

- **PSP 0.** En esta fase los desarrolladores aplican algunas técnicas PSP básicas: cómo registrar el tiempo de desarrollo y consignar los defectos introducidos y removidos, durante cada fase de desarrollo. Estas medidas se utilizan en el análisis y planificación del proceso y como base para mejorar estimaciones futuras.
- **PSP 0.1.** Esta versión agrega medidas para calcular el tamaño real del código y un registro donde se consignan propuestas para el mejoramiento del proceso.
- **PSP 1.** En esta versión se introduce el método PROBE (Proxy-Based Estimating) el cual utiliza datos históricos para estimar el tamaño del código empleando una regresión simple.
- **PSP 1.1.** Esta versión agrega estimación de recursos y calendarización de las tareas para el desarrollo del proyecto. La calendarización de las tareas permite identificar la importancia o peso de las tareas, para que el atraso en alguna de ellas no signifique una tardanza en la duración total del proyecto.

- **PSP 2.** Esta versión introduce al ciclo de desarrollo, las etapas de revisión de la fase de diseño y revisión de la fase de codificación, como también, cálculos de la eficiencia en la eliminación de defectos.
- **PSP 2.1.** Esta versión enseña técnicas de especificación de diseño y formas de prevenir defectos.
- **PSP 3.** En esta versión se cubren técnicas de verificación de diseño.

### ***Aplicación de la metodología.***

La estructura del proceso PSP se muestra conceptualmente en la Figura 26. Como ya se mencionó, los métodos utilizados por la metodología PSP se han introducido en una serie de 7 versiones, cada una con un conjunto similar de logs, formas, scripts y estándares. Los scripts definen los pasos a realizar para cada parte del proceso, y se pueden utilizar como listas de chequeo. En este trabajo se utilizó el script de la versión PSP0, la cual tiene 4 formularios (para ver los formularios a detalle consulte anexo B):

- **PSP0 Project Plan Summary.** Para resumir tiempos planeados y actuales y defectos por fase.
- **PSP0 Time Recording Log.** Bitácora de tiempo.
- **PSP0 Defect Recording Log.** Bitácora de defectos.
- **PSP0 Defect Type Standard.** Usado para definir tipos de defectos estándares.

Las actividades realizadas en cada fase fueron:

### ***Requerimientos.***

Se centró en las siguientes actividades:

1. Definir objetivo del sistema.
2. Definir el alcance del proyecto.
3. Enumeración de las características del sistema.

- Entrada de datos de sensor. Proceso que permite introducir los datos de lluvia capturados por el sensor.
- Mostrar gráfica de evento de lluvia. Proceso que permite visualizar gráficamente cómo se comporta una tormenta según los datos tomados por el sensor.
- Mostrar cálculo de los 5, 10, 15, 20, 25, 30 minutos más intensos del evento de lluvia.
- Obtener cálculos de la energía cinética del evento de lluvia.
- Crear archivo de Excel con los datos calculados.
- Mostrar hietograma del evento de lluvia.

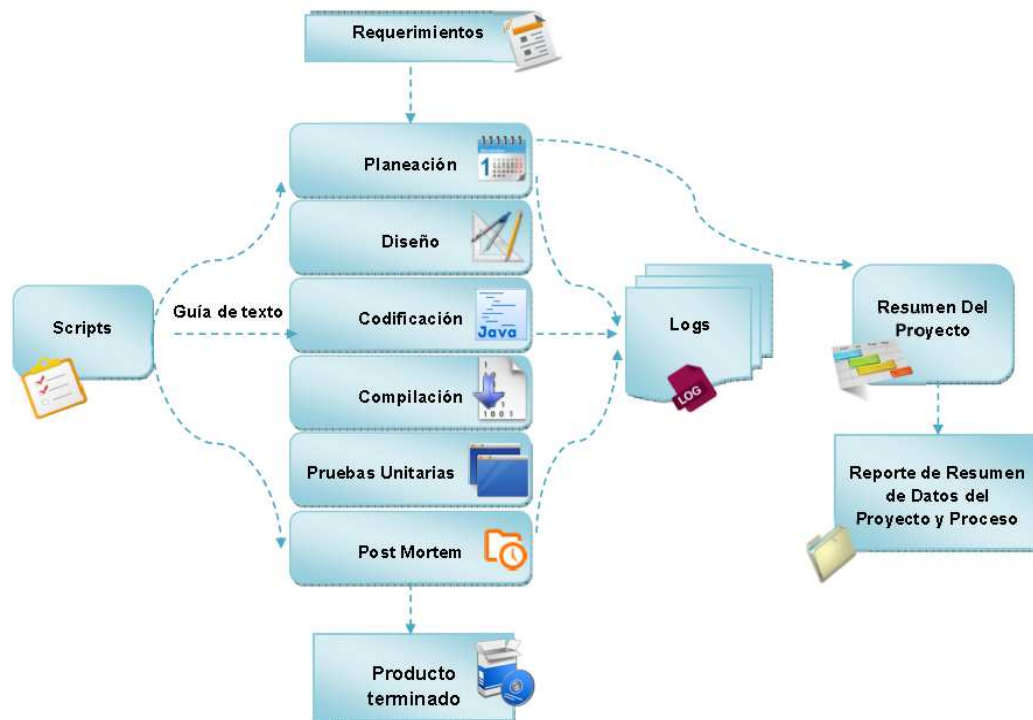


Figura 26. Estructura básica de PSP

### *Planeación.*



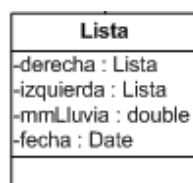
Para el presente trabajo se ha dividido al sistema en 5 módulos: Módulo principal, cálculo de eventos de lluvia, generación de reportes en Excel, creación de hietograma de evento de lluvia, creación de gráfica de pluviograma. La metodología PSP recomienda que cada módulo debe tener un tamaño de entre 100 y 250 Líneas de código (Lines of code LOC).

### ***Diseño.***

Se definió la forma del sistema para que fuera capaz de soportar todos los requisitos. Se realizaron las siguientes actividades.

1. Descripción de arquitectura.
2. Descripción de interfaces para que el usuario se comuniquen con el sistema.
3. Se desarrolló un diagrama de clases para representar el diseño del sistema (Anexo C).

Tomando en cuenta que el instrumento de medición sólo almacena datos en caso de presentarse lluvia, en un periodo de medición es variable la cantidad de información que el sistema debe procesar. Para tratar toda la información generada por el pluviógrafo, se utilizará una estructura de datos tipo lista doblemente enlazada, donde cada archivo de datos (ver Figura 24) será un nodo de la lista. De esta manera el sistema reduce los accesos a disco ya que todos los archivos se leerán de una sola vez, y para realizar los cálculos de eventos de lluvia e información para graficar, bastara con leer la información almacenada en cada objeto nodo de la estructura de lista. En la Figura 27 se muestra el diagrama UML de la clase Lista, con la que se crean los objetos nodos.



**Figura 27. Diagrama UML de clase Lista**

### ***Codificación.***

En esta etapa del desarrollo, se definió el código en términos de implementación del sistema y se implementaron las clases en término de los componentes del sistema.

El código fuente se desarrolló en lenguaje java, estableciendo los nombres de las clases, objetos y variables relacionados directamente a la función central que realizan, puesto que así lo establece PSP.

Se utilizó la biblioteca JFreeChart para apoyo en la elaboración de gráficas y biblioteca Apache POI para la elaboración de reportes en formato de Excel.

### ***Compilación.***

En esta etapa del flujo de trabajo se utilizó la plantilla de registro de defectos, donde se registran las fallas encontradas en el sistema y el tiempo que tomó repararlas.

### ***Prueba***

1. Se probó la eficacia del software y su fácil manejo para los usuarios. Se verificó el resultado de la implementación de cada módulo de manera individual hasta que ejecutara sin errores. Posteriormente todos los módulos integrados. Para el módulo de “cálculo de eventos de lluvia” se tomaron datos de prueba reales correspondientes a estaciones meteorológicas instaladas en el estado de Chiapas.
2. Se repararon todas las fallas referentes a la realización de cálculos.
3. Se registraron los defectos en el Log de registro de defectos.
4. Se registró el tiempo en el Log de registro del tiempo.

### ***Postmortem***

En esta etapa, y según lo señala el script de PSP0, se llenó el formulario de **PSP Project Plan Summary** con los datos reales de tiempo de desarrollo y fallas encontradas y corregidas en el sistema.

Los componentes que conforman el producto final del software se muestran en la **¡Error! No se encuentra el origen de la referencia..**

**Tabla 4. Componentes del Software**

Componente	Tecnología empleada	Propósito
Software Principal	Java	Su objetivo es ser un punto de comunicación entre la computadora y el usuario para que éste pueda llevar un sencillo control de la información referente a medición de precipitación pluvial de alguna(s) región(es) geográficas, que se tenga almacenada. Permite acceso a información de eventos de lluvia para su análisis
Reportes gráficos	JFreechart	Su objetivo es presentar información en forma de gráficas de calidad profesional, para que el usuario o analista pueda observar de una manera descriptiva la precipitación en función de tiempo.
Reportes en formato Excel	Apache POI (librería de código en JAVA)	Permite al usuario trasladar a hojas de cálculo la información referente a los eventos de lluvia calculados durante el periodo de medición. El objetivo de este módulo es que el usuario pueda seguir teniendo la información de la manera y en el formato que acostumbra, con la ventaja que no es

		el mismo usuario quien tiene que crear la hoja de cálculo sino que el sistema lo hace por él.
Reportes en formato PDF	iText	Su objetivo es permitir al usuario generar un documento portable, fácil de trasladar, con la información obtenida por el sistema, y así tenerla disponible en cualquier momento y lugar sin la restricción de tener que consultarla directamente en el sistema.
Software de microcontrolador	Arduino.	Es el encargado de procesar y almacenar todas las lecturas del instrumento, en un formato estándar fácil de entender, para que no sea estrictamente necesario contar con el software de interpretación, sino que una sencilla capacitación sea suficiente para comprender la información almacenada.

# Capítulo 4: Integración de la plataforma

## Hardware

El circuito electrónico permite la comunicación con el usuario a través de la pantalla LCD y los botones que le permiten ajustar la fecha y hora en que se comenzará a medir la cantidad de precipitación pluvial (ver Figura 28)

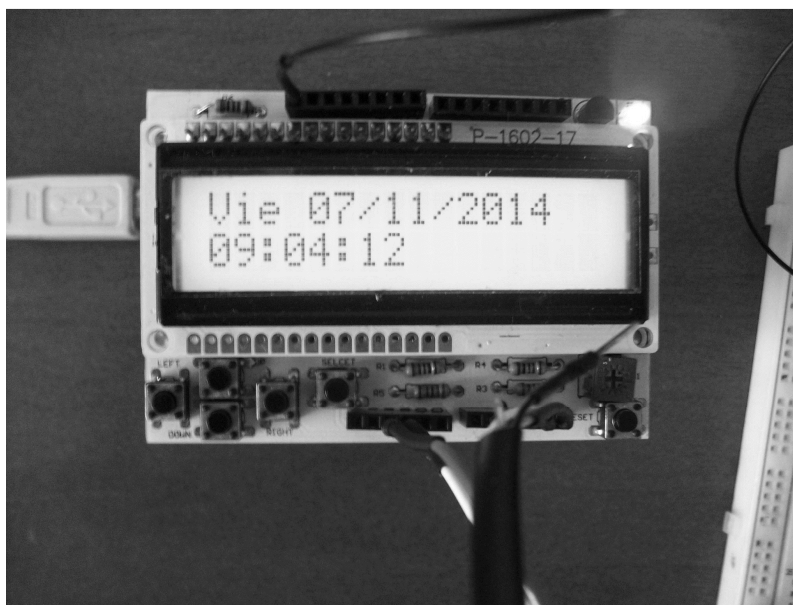
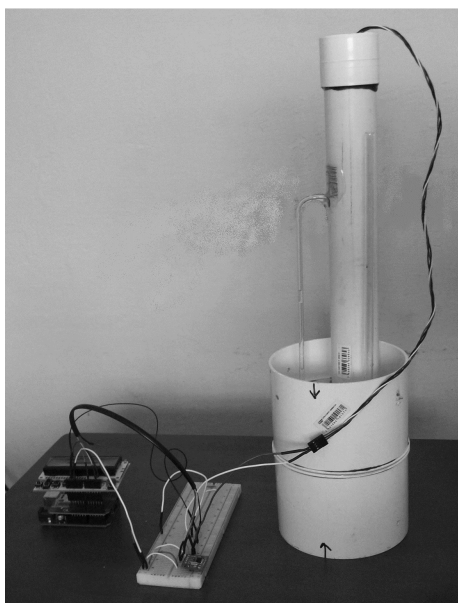


Figura 28. Interfaz del Hardware

El circuito se conecta al sensor instalado en la cámara de sifonaje como se muestra en la Figura 29.



**Figura 29. Conexión de estructura física y circuito electrónico.**

El material PVC con el que está fabricada la estructura física del aparato, permite que éste pueda ser instalado con facilidad en cualquier región geográfica sin importar las condiciones climatológicas ya que es resistente a humedad y altas temperaturas. Los dispositivos electrónicos que componen el circuito final no presentan aumento de temperatura en su funcionamiento, lo que contribuye a la seguridad de aparato final. Véase Figura 30.

### ***Costos***

En la **¡Error! No se encuentra el origen de la referencia.** se presenta el costo de cada elemento que compone la estructura construida.

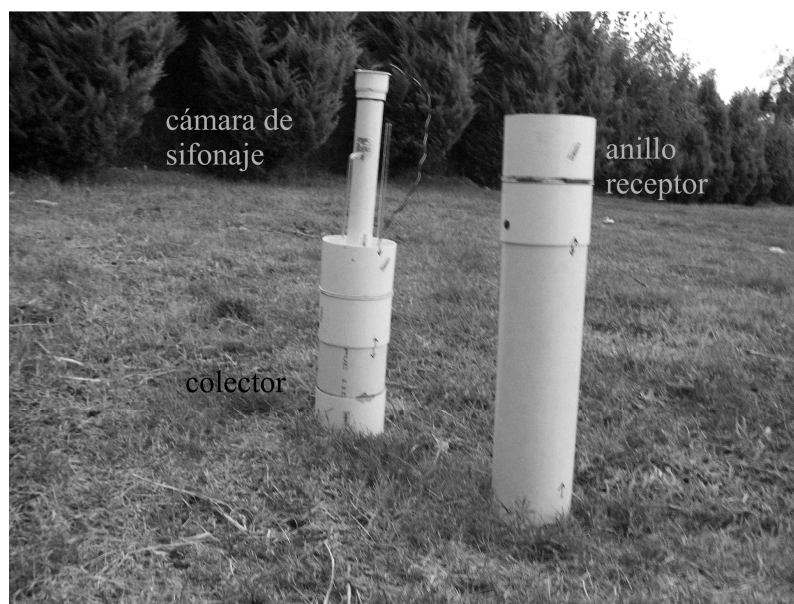


Figura 30. Instalación de aparato final.

Tabla 5. Costos de los componentes de la estructura

Elemento	Costo (en pesos)
Estructura PVC con sistema de sifonaje	\$1,000.00
Placa Arduino Uno	\$400.00
Sensor de nivel de líquidos eTape Millone	\$850.00
RealTime clock DS1307	\$220.00
microSD Shield para Arduino	\$240.00
Pantalla LCD	\$280.00
<b>Costo Total</b>	<b>\$2,990.00</b>

### ***Problemas en la construcción***

Cuando la cámara de sifonaje vierte la precipitación pluvial captada al colector mediante el sifón, la presión generada por el aire impide que la cámara de sifonaje pueda vaciarse

por completo, por lo que se tuvo que colocar un tubo de cristal al colector para evitar este problema.

El sensor de nivel de líquidos eTape tiene que ser instalado cuidadosamente, y debe quedar en posición vertical sin flexionarse, de modo contrario, puede provocar que los valores resistivos generados por el funcionamiento del sensor varíen y alteren la información de precipitación pluvial registrada por el aparato y por consiguiente provocar un mal funcionamiento del aparato.

## ***Software***

El software desarrollado para la plataforma se compone de varios módulos, cada uno destinado a una tarea en específico, a continuación se describe cada uno de ellos.

### ***Modulo Principal.***

Se encarga de presentar al usuario la información que se tiene almacenada en el equipo de cómputo, de mediciones realizadas previamente. En la Figura 31. Pantalla principal de software se muestra la pantalla de software que corresponde a este módulo, En esta parte del software donde el usuario añade la información de precipitación pluvial registrada para alguna región (sección *a* de Figura 31), puede acceder al módulo de presentación de Eventos (sección *b* de Figura 31) y eliminar información que ya no se requiera (sección *c* Figura 31).



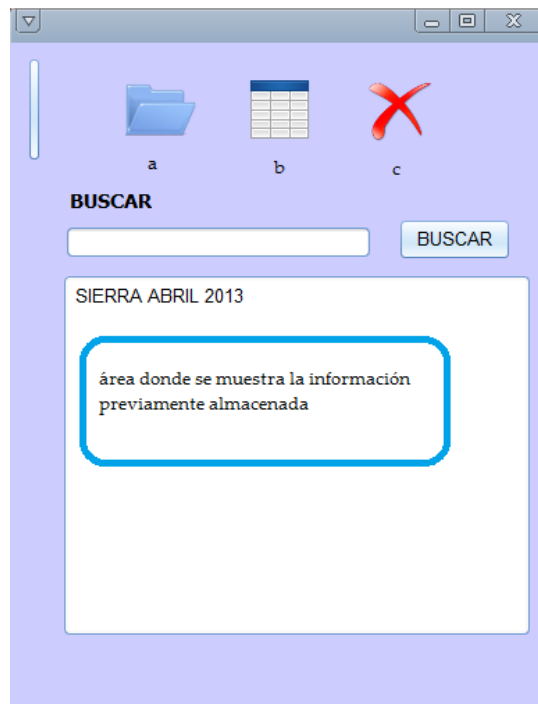


Figura 31. Pantalla principal de software

### ***Módulo de Eventos de Lluvia.***

Este módulo del software se encarga de calcular cuántos eventos de lluvia se registraron durante el periodo de tiempo en que el aparato estuvo midiendo. Para cada evento de lluvia presenta, fecha y hora en que comenzó a llover, duración en minutos del evento, cuantos mm de precipitación pluvial se registraron. En la Figura 32 se muestra la pantalla donde se exponen los eventos al usuario y podemos ver que se enlista cada evento de lluvia según la fecha en que ocurrió. Cada evento de lluvia se divide en secciones cada que se presenta una variación en la intensidad de la lluvia, para cada tramo se muestra, su duración en minutos, la cantidad de la intensidad de la lluvia en mm/hr y la energía cinética contenida en cada tramo de lluvia (ver Figura 33). Este módulo permite acceder los demás módulos que componen el software.



Figura 32. Pantalla de eventos de lluvia.

Dia y Tiempo (hr:min)	Lámina (mm)	Delta (t)	Delta (lam)	Intensidad (mm/hr)	Energía de la lluvia (por mm)	Energía d...
04 Abril 2013 17:49	0	0	0	0	0	0
04 Abril 2013 18:03	0.2	14	0.2	0.857	0.113	0.023
04 Abril 2013 18:09	0.3	6	0.1	1	0.119	0.012
04 Abril 2013 18:13	0.6	4	0.3	4.5	0.176	0.053
04 Abril 2013 18:39	0.7	26	0.1	0.231	0.063	0.006
04 Abril 2013 21:56	0.9	197	0.2	0.061	0.013	0.003
05 Abril 2013 3:50	1	354	0.1	0.017	0	0
05 Abril 2013 3:56	1.4	6	0.4	4	0.172	0.069
05 Abril 2013 4:01	1.7	5	0.3	3.6	0.168	0.05
05 Abril 2013 4:05	2.4	4	0.7	10.5	0.208	0.146
05 Abril 2013 4:11	2.6	6	0.2	2	0.145	0.029
05 Abril 2013 4:47	9.9	36	7.3	12.167	0.214	1.56
05 Abril 2013 4:53	11.1	6	1.2	12	0.213	0.256
05 Abril 2013 5:20	16.9	27	5.8	12.889	0.216	1.252
05 Abril 2013 5:22	17	2	0.1	3	0.161	0.016
05 Abril 2013 5:26	17.1	4	0.1	1.5	0.134	0.013
05 Abril 2013 5:34	17.4	8	0.3	2.25	0.15	0.045
05 Abril 2013 5:36	17.5	2	0.1	3	0.161	0.016
05 Abril 2013 5:38	17.9	2	0.4	12	0.213	0.085
05 Abril 2013 5:50	19	12	1.1	5.5	0.184	0.202

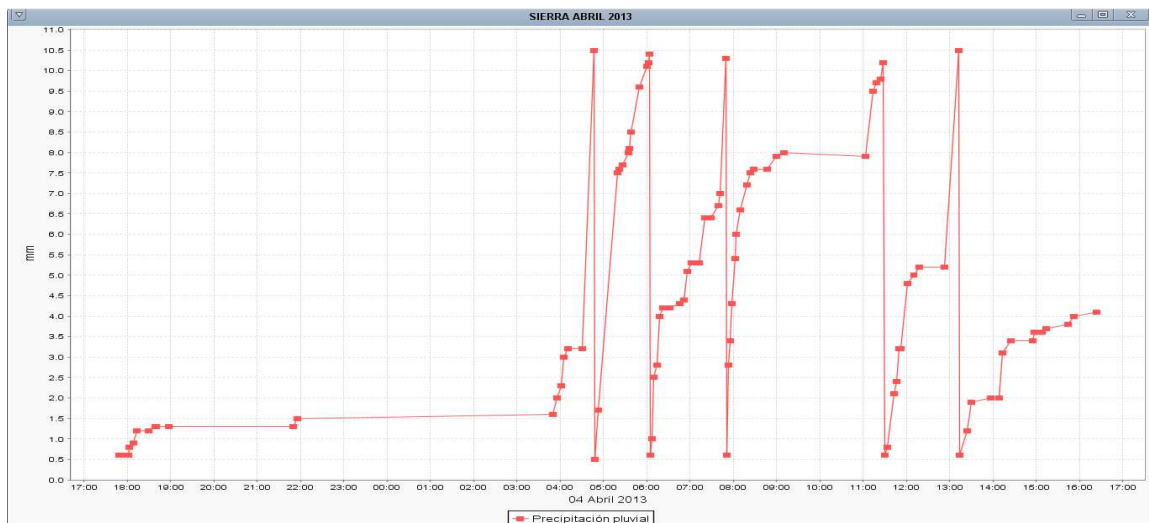
Figura 33. Descripción de un evento de lluvia.

## ***Módulo Excel.***

Se encarga de guardar en un archivo de Excel, la información de todos los eventos de lluvia de la región que el usuario se encuentre analizando en ese momento.

## ***Módulo Pluviogramas.***

Este módulo permite visualizar gráficamente como sucedió el evento de lluvia que se encuentre analizando el usuario. Esta gráfica es equivalente al pluviograma, generado por los pluviógrafos mecánicos actuales. En la Figura 34 se muestra el pluviograma correspondiente al evento de lluvia mostrado en la Figura 33.

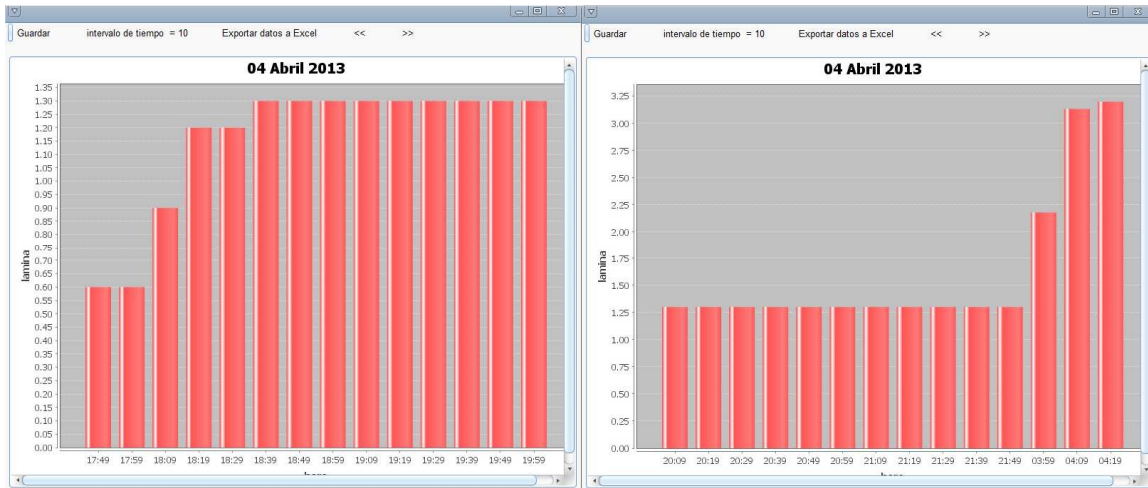


**Figura 34. Pantalla de sistema que muestra un pluviograma.**

## ***Módulo Hietogramas.***

Este módulo permite visualizar el hietograma correspondiente al evento de lluvia que se encuentre analizando el usuario, de inicio presenta el cálculo para un intervalo de 10 minutos, pero dicho intervalo es variable y se puede ajustar a consideración del usuario

que se encuentre analizando los datos de lluvia. En la Figura 35 podemos visualizar como el software presenta el hietograma del evento, como la dimensión de la gráfica puede variar según el intervalo de tiempo elegido por el usuario, el hietograma se presenta en varias secciones, en la Figura 35 se presentan las dos primeras secciones que componen el hietograma correspondiente al evento de lluvia mostrado en la Figura 33.



**Figura 35. Pantalla de sistema donde se muestran un hietograma.**

## ***Validación***

Se realizó un periodo de pruebas para validar los datos generados por el aparato utilizando el método descrito en la metodología, obteniendo un 99% de exactitud.

En el curso “instrumental meteorológico” impartido en el Colegio de postgraduados, en su práctica sobre instalación de pluviómetros y pluviógrafos se establece la metodología para la toma de lecturas de un pluviógrafo, la cual se menciona a continuación:

1. Verter el contenido volumétrico de agua recolectada en una probeta graduada mililitros, considerando que  $1 \text{ ml} = 1 \text{ cm}^3$ .
2. Se obtiene la cantidad de precipitación (en mm.) con la siguiente fórmula:

$$mm = \frac{p}{A}$$

2

Dónde:

mm: es la cantidad de precipitación registrada en mm de lluvia.

p: es la cantidad de precipitación en ml.

A: es el área del anillo receptor del pluviógrafo, en cm<sup>2</sup>.

Por ejemplo, para un pluviógrafo de la marca alemana Lambrecht (ver Figura 36) instalado en la UACH, el cual tiene una capacidad de 200 cm<sup>3</sup>, su anillo receptor mide 16 cm de diámetro y su rango de medición es de 0 a 10 mm de precipitación, si el agua recolectada en él alcanza los 100 ml. en la probeta sus mm de precipitación se calculan de la siguiente manera:

$$mm = \frac{100 \text{ ml}}{\pi*(16/2)^2} = 4.9$$

De esta manera el sensor de nivel de líquidos instalado se calibra en el primer funcionamiento para obtener el valor de resistencia máximo que corresponde al nivel de agua mínimo y del valor de resistencia mínimo que corresponde al nivel más alto de agua que se puede medir por el sensor y que equivaldría a 10 mm de precipitación. La cantidad de mm de precipitación se obtiene con la siguiente fórmula:

$$mm = \frac{((valor \text{ Maximo} - LecturaSensor) * 10)}{valorMaximo - valor \text{ M\u00ednimo}}$$

3



**Figura 36. Pluviógrafo thies**

## ***Conclusiones***

La validación del prototipo demuestra que se tiene exactitud en la toma de datos lo que permite disminuir errores en la lectura de los datos y una mayor productividad de las personas dedicadas al manejo de cuencas y otros estudios hidrológicos y de irrigación.

Además de la exactitud lograda en la toma de datos se redujeron los costos un 70% al utilizar materiales comunes, pero resistentes a las inclemencias del tiempo y al utilizar software libre para desarrollar la herramienta de interpretación de los datos.

Se desarrolló un software robusto y dedicado aplicado al análisis de la información de pluviogramas, siendo ésta la parte fundamental de la innovación de este trabajo de investigación, debido a la complejidad en la extracción de información de este tipo de gráficas, logrando un apoyo en diversas áreas de investigación.

El proceso de calibración del aparato depende de la posición vertical del sensor electrónico de nivel de líquidos ya que cualquier variación en un ángulo diferente a  $90^\circ$  entre el sensor y el suelo repercutirá en la lectura de los datos ya que el sensor generaría valores totalmente fuera del rango establecido por el fabricante.

El prototipo del pluviógrafo y el software respectivo para la interpretación de los datos puede ser utilizado como plataforma independiente o embebida, es decir, se puede utilizar el pluviógrafo con algún otro software comercial.

# Anexo A

En este anexo, se muestra el sketch de Arduino programado para el funcionamiento del circuito electrónico y se comenta cada sección.

```
#include <LiquidCrystal.h>
#include <LCDKeypad.h>
#include "Wire.h"
#include <SdFat.h>
SdFat sd;
SdFile myFile;
const int chipSelect = 8;

#define SERIESRESISTOR 1950//la resistencia que se conecto al sensor es
de 2k
#define SENSORPIN A1 //Entrada por la cual se lee el valor del sensor.
#define DS1307_ADDRESS 0x68
byte zero = 0x00; //para poder leer información del sensor de tiempo

//se prepara la pantalla LCD y sys botones
LiquidCrystal lcd(8, 13, 9, 4, 5, 6, 7);
int adc_key_val[5] ={50, 200, 400, 600, 800 };
int NUM_KEYS = 5;
int adc_key_in;
char miDia[4] = "Lun";

//valores para configurar la fecha
int day=1;
int month=1;
int year = 10;
int hour = 0;
int mins = 0;
int second = 0;
int opcionSeleccion = 0;
int dayWeek = 1;
float cantidadAgua;
float cantidadAguaAnterior;
float rangoMax, rangoMin; //se utilizan estas variables para calibrar los
valores máximo y mínimo que puede alcanzar el sensor

int numeroArchivo; //para llevar una secuencia en cada archivo de datos
float cantidadInicio; //estas variable se utiliza cuando comienza a subir
el aparato, baja y vuelve a comenzar a subir, sirve para saber en que
punto comenzo a subir
int dayInicio=1;
```



```

int monthInicio=1;
int yearInicio = 14;
int hourInicio = 0;
int minsInicio = 0;//
int secondInicio = 0;
boolean estaSubiendo;

```

**A continuación se describe el método *setup()* que es aquel que se ejecuta cuando se conecta o alimenta por primera ocasión el circuito.**

```

void setup(){
  Wire.begin();
  Serial.begin(9600);
  funcionReset(); // se invoca la función de reinicio del circuito, sirve
para establecer la fecha y hora al pluviógrafo
  cantidadAgua=0;
  cantidadAguaAnterior=0;
  cantidadInicio=0;
  numeroArchivo= 0;
  estaSubiendo=true;
  rangoMax=0; rangoMin=5000;
  calibracionSensor();// se invoca función que calibra el sensor
}

```

```

void loop(){

  printDate(); //muestra la hora en la pantalla LCD
  visualizarFechaYHoraEnPantalla();// muestra la hora en el monitor
Serial
  float valorSensor = funcionObtenerValorDeSensor();
  cantidadAgua = cantidadEnmmDeLluvia(valorSensor);

  float diferencia = cantidadAgua - cantidadAguaAnterior;

  if(abs(diferencia)>=0.1){//si la diferencia es considerable, hay que registrar, sino ignorar

    if(diferencia>0){
      // significa que debe guardar por que va subiendo

      if(!estaSubiendo){
        //entra aquí para guardar el punto mas bajo alcanzado cuando iba bajando el nivel del agua en la cámara de sifonaje.
      }

// a continuación se crea el nombre para el archivo de texto donde se guardaran los datos
      String str = String(numeroArchivo);
      str+=" .txt";
      int medidaNombre = str.length();
    }
  }
}

```

```

        char name[medidaNombre+1];

        str.toCharArray(name,(medidaNombre+1));
        //hay que invocar la función de guardado pero con los valores
que había cuando se alcanza el valor de cantidadInicio
        guardarArchivo(name, dayInicio, monthInicio, yearInicio,
hourInicio, minsInicio, secondInicio, cantidadInicio);
        numeroArchivo++;
        estaSubiendo=true;
    }
    String str = String(numeroArchivo);
    str+=" .txt";
    int medidaNombre = str.length();
    char name[medidaNombre+1];

    str.toCharArray(name,(medidaNombre+1));
    guardarArchivo(name, day, month, year, hour, mins, second,
cantidadAgua);
    numeroArchivo++;
}
}
else if(diferencia<0){
    estaSubiendo=false;
    cantidadInicio=cantidadAgua;
    dayInicio=day;
    monthInicio=month;
    yearInicio = year;
    hourInicio = hour;
    minsInicio = mins;
    secondInicio=second;
}

cantidadAguaAnterior = cantidadAgua;

    delay(2000);//Cada cuanto tiempo hacer el testeo
}

// A CONTINUACION SE PRESENTA LA FUNCION QUE SIRVE PARA OBTENER LOS
VALORES MAXIMO Y MINIMO QUE SE PUEDEN LEER DEL SENSOR
void calibracionSensor(){

    float maxAnterior;
    float minAnterior;
    delay(3000);

    lcd.clear();
    lcd.setCursor(0,0);

```

```

lcd.print("PARA CALIBRAR");
  lcd.setCursor(0,1);

//ESPERA LA CONFIRMACION DEL USUARIO PARA PODER COMENZAR A CALIBRAR.
lcd.print("PRESIONE OK");
int k =0;
while(k!=4){
  adc_key_in = analogRead(0);    // read the value from the sensor
  k = get_key(adc_key_in);
}

  lcd.clear();
  lcd.setCursor(0,0);

  lcd.print("CALIBRANDO...");
  rangoMax = funcionObtenerValorDeSensor();
  rangoMin = funcionObtenerValorDeSensor();
  do
  {
    delay(3000);
    maxAnterior = rangoMax;
    minAnterior = rangoMin;
    float lectura = funcionObtenerValorDeSensor();
    Serial.print("lectura: ");
    Serial.println(lectura);
    Serial.print("rangoMax ");
    Serial.println(rangoMax);
    Serial.print("rangoMin: ");
    Serial.println(rangoMin);
    Serial.print("rangoMaxAnterior: ");
    Serial.println(maxAnterior);
    Serial.print("rangoMinAnterior: ");
    Serial.println(minAnterior);

    if(rangoMax<lectura)
      rangoMax = lectura;

    if(rangoMin>lectura)
      rangoMin = lectura;

  }while(maxAnterior!=rangoMax || minAnterior!=rangoMin);

  lcd.clear();
  lcd.setCursor(0,0);

  lcd.print("listo");
  Serial.println("VALORES DE CALIBRACION ");
  Serial.print("rangoMax ");
  Serial.println(rangoMax);

```

```

    Serial.print("rangoMin ");
    Serial.println(rangoMin);
}

//ESTA ES LA FUNCION QUE SIRVE PARA LEER EL VALOR RESISTIVO DEL SENSOR
float funcionObtenerValorDeSensor(){
//Se encarga de leer el sensor, sobre eso determinar la altura en mm de
lluvia que tengamos.
float reading, promReading;

    reading = analogRead(SENSORPIN);

    Serial.print("Analog reading ");
    Serial.println(reading);

    return reading;
}

//esta función es la encargada de determinar que cantidad en mm de lluvia
ha medido el pluviógrafo.

float cantidadEnmmDeLluvia(float lecturaDelSensor){
float diferencia = ((rangoMax-lecturaDelSensor)*10)/(rangoMax-
rangoMin);
return diferencia;
}

//a continuación se presenta la función encargada de crear el archivo de
texto con los datos de medición.

void guardarArchivo(char name[], int dia, int mes, int anio, int hh, int
mm, int ss, float lluvia){
    Serial.println("Entro a funcion guardar");
    if (!sd.begin(chipSelect, SPI_HALF_SPEED)) sd.initErrorHalt();

    // open the file for write at end like the Native SD library
    if (!myFile.open(name, O_RDWR | O_CREAT | O_AT_END)) {
        sd.errorHalt("opening test.txt for write failed");
    }
    // if the file opened okay, write to it:
    Serial.print("Writing to test.txt...");
    myFile.print(dia); myFile.print("/"); myFile.print(mes);
myFile.print("/"); myFile.print(anio);
    myFile.println("");
    myFile.print(hh);myFile.print(":"); myFile.print(mm);myFile.print(":");
myFile.print(ss);
    myFile.println("");
    myFile.println(lluvia);
    // close the file:
    myFile.close();
}

```

```

    Serial.println("done.");
}

void funcionReset(){
    day=1;
    month=1;
    year = 10;
    hour = 0;
    mins = 0;
    opcionSeleccion = 0;
    dayWeek = 1;

    lcd.begin(16, 2);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("AJUSTE DE FECHA");

    lcd.setCursor(0,1);
    lcd.print("Y HORA");

    delay(3000);
    lcd.clear();
    ajustarValores();
    setDateTime();
}

void visualizarFechaYHoraEnPantalla(){

    lcd.clear();

    Wire.beginTransmission(DS1307_ADDRESS);
    Wire.write(zero);
    Wire.endTransmission();

    Wire.requestFrom(DS1307_ADDRESS, 7);

    second = bcdToDec(Wire.read());
    mins = bcdToDec(Wire.read());
    hour = bcdToDec(Wire.read() & 0b111111); //24 hour time
    dayWeek = bcdToDec(Wire.read()); //0-6 -> sunday - Saturday
    day = bcdToDec(Wire.read());
    month = bcdToDec(Wire.read());
    year = bcdToDec(Wire.read());

    /*Serial.print(dayWeek);
    Serial.print(" ");
    Serial.print(day);

```

```

Serial.print("/");
Serial.print(month);
Serial.print("/");
Serial.print(year);
Serial.print(" ");
Serial.print(hour);
Serial.print(":");
Serial.print(mins);
Serial.print(":");
Serial.println(second);*/

lcd.setCursor(0,0);
mostrarFecha(0);
  lcd.setCursor(0,1);
mostrarHora(1);

}

void ajustarValores(){
  lcd.setCursor(0,0);
  int k = 0;
  int oldK=-1;
  mostrarSeleccion(opcionSeleccion);
  mostrarFecha(1);
  while( k!=4){

    delay(50);
    adc_key_in = analogRead(0);
    k = get_key(adc_key_in);
    if (k != oldK)
      {
        evaluarK(k);
        mostrarSeleccion(opcionSeleccion);

        oldK = k;
      }
    delay(100);
  }
  Serial.println("valores ajustados");
  Serial.print(dayWeek);
  Serial.print(" ");
  Serial.print(day);
  Serial.print("/");
  Serial.print(month);
  Serial.print("/");
  Serial.print(year);
  Serial.print(" ");
  Serial.print(hour);
  Serial.print(":");
  Serial.println(mins);

```

```

    Serial.println("valores a pantalla");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("HECHO");
    delay(2000);
}

void evaluarK(int k){

    switch(k){
        case 0: if(opcionSeleccion<5) opcionSeleccion++; break;
        case 1: incrementarVariable(); break;
        case 2: decrementarVariable(); break;
        case 3: if(opcionSeleccion>0) opcionSeleccion--; break;
    }
}

void incrementarVariable(){

    switch(opcionSeleccion){
        case 0: if(day<31) day ++; else day = 1; mostrarFecha(1); break;
        case 1: if(month<12) month ++; else month = 1; mostrarFecha(1);
break;
        case 2: year ++; mostrarFecha(1); break;
        case 3: if(hour<23) hour ++; else hour = 0 ; mostrarHora(1); break;
        case 4: if(mins<59) mins ++; else mins = 0; mostrarHora(1); break;
        case 5: if(dayWeek<7) dayWeek++; else dayWeek = 1; mostrarFecha(1);
break;
    }
}

void decrementarVariable(){

    switch(opcionSeleccion){
        case 0: if(day>1) day --; else day = 31; mostrarFecha(1); break;
        case 1: if(month>1) month --; else month = 12; mostrarFecha(1);
break;
        case 2: year --; mostrarFecha(1); break;
        case 3: if(hour>0) hour --; else hour = 23 ; mostrarHora(1); break;
        case 4: if(mins>0) mins --; else mins = 59; mostrarHora(1); break;
        case 5: if(dayWeek>1) dayWeek--; else dayWeek = 7; mostrarFecha(1);
break;
    }
}

void mostrarSeleccion(int op){

```

```

lcd.clear();
  lcd.setCursor(0,0);

  switch(op){
  case 0:    lcd.print("Ajustando: dia");  mostrarFecha(1); break;
  case 1:    lcd.print("Ajustando: mes");  mostrarFecha(1); break;
  case 2:    lcd.print("Ajustando: anio");  mostrarFecha(1); break;
  case 3:    lcd.print("Ajustando: hora");  mostrarHora(1); break;
  case 4:    lcd.print("Ajustando: mins");  mostrarHora(1); break;
  case 5:    lcd.print("Ajustando: L-D");  mostrarFecha(1); break;
  }
}

int get_key(unsigned int input)
{
  int k;
  for (k = 0; k < NUM_KEYS; k++)
  {
    if (input < adc_key_val[k])
    {
      return k;
    }
  }
  if (k >= NUM_KEYS)k = -1; // No valid key pressed
  return k;
}

void mostrarFecha(int renglon){
lcd.setCursor(0,renglon);
//lcd.print(diaDeLaSemana(dayWeek));
diaDeLaSemana(dayWeek);
  lcd.print(miDia[0]);
lcd.print(miDia[1]);
lcd.print(miDia[2]);
lcd.print(" ");
  if(day<10)
    lcd.print("0");
    lcd.print(day);
  lcd.print("/");
  if(month<10)
    lcd.print("0");
    lcd.print(month);
  lcd.print("/20");
  lcd.print(year);
}

void mostrarHora(int renglon){
lcd.setCursor(0,renglon);

  if(hour<10)

```



```

    lcd.print("0");
    lcd.print(hour);
    lcd.print(":");
    if(mins<10)
        lcd.print("0");
        lcd.print(mins);
        lcd.print(":");
    if(second<10)
        lcd.print("0");
        lcd.print(second);
}
void diaDeLaSemana(int day){
switch(day){
    case 1: miDia[0] = 'L'; miDia[1] = 'u'; miDia[2] = 'n'; break;
    case 2: miDia[0] = 'M'; miDia[1] = 'a'; miDia[2] = 'r'; break;
    case 3: miDia[0] = 'M'; miDia[1] = 'i'; miDia[2] = 'e'; break;
    case 4: miDia[0] = 'J'; miDia[1] = 'u'; miDia[2] = 'e'; break;
    case 5: miDia[0] = 'V'; miDia[1] = 'i'; miDia[2] = 'e'; break;
    case 6: miDia[0] = 'S'; miDia[1] = 'a'; miDia[2] = 'b'; break;
    case 7: miDia[0] = 'D'; miDia[1] = 'o'; miDia[2] = 'm'; break;
}
}

//la siguiente funcion permite establecer valores al sensor de tiempo,
para asi programar el pluviógrafo con la fecha y hora que se comienza a
medir
void setDateTime(){
    byte second =    0; //0-59
    byte minute =   mins; //0-59
    byte hours =    hour; //0-23
    byte weekDay =  dayWeek; //1-7
    byte monthDay =  day; //1-31
    byte months =   month; //1-12
    byte years =    year; //0-99

    //print the date EG 3/1/11 23:59:59
    Serial.println("valores que se enviaron a setTime");
    Serial.print(weekDay);
    Serial.print(" ");
    Serial.print(monthDay);
    Serial.print("/");
    Serial.print(month);
    Serial.print("/");
    Serial.print(year);
    Serial.print(" ");

```

```

Serial.print(hour);
Serial.print(":");
Serial.print(minute);
Serial.print(":");
Serial.println(second);

Wire.beginTransmission(DS1307_ADDRESS);
Wire.write(zero); //stop Oscillator

Wire.write(decToBcd(second));
Wire.write(decToBcd(minute));
Wire.write(decToBcd(hours));
Wire.write(decToBcd(weekDay));
Wire.write(decToBcd(monthDay));
Wire.write(decToBcd(months));
Wire.write(decToBcd(years));

Wire.write(zero); //start

Wire.endTransmission();
}

byte decToBcd(byte val){
// Convert normal decimal numbers to binary coded decimal
return ( (val/10*16) + (val%10) );
}

byte bcdToDec(byte val) {
// Convert binary coded decimal to normal decimal numbers
return ( (val/16*10) + (val%16) );
}

void printDate(){
// Reset the register pointer
Wire.beginTransmission(DS1307_ADDRESS);
Wire.write(zero);
Wire.endTransmission();

Wire.requestFrom(DS1307_ADDRESS, 7);

int secondR = bcdToDec(Wire.read());
int minuteR = bcdToDec(Wire.read());
int hourR = bcdToDec(Wire.read() & 0b111111); //24 hour time
int weekDay = bcdToDec(Wire.read()); //0-6 -> sunday - Saturday
int monthDay = bcdToDec(Wire.read());
int monthR = bcdToDec(Wire.read());
int yearR = bcdToDec(Wire.read());

```

```
    day = monthDay;
    month = monthR;
    year = yearR;
    hour = hourR;
    mins = minuteR;
    second = secondR;
    dayWeek = weekDay;
    //print the date EG 3/1/11 23:59:59
    Serial.print(weekDay);
    Serial.print(" ");
    Serial.print(monthDay);
    Serial.print("/");
    Serial.print(monthR);
    Serial.print("/");
    Serial.print(yearR);
    Serial.print(" ");
    Serial.print(hourR);
    Serial.print(":");
    Serial.print(minuteR);
    Serial.print(":");
    Serial.println(secondR);
}
```

# Anexo B

## PSP Project Plan Summary

Student	Irving A. Gonzalez Manríquez	Date	
Program	Statistics	Program #	1
Instructor		Language	Java

Program Size (LOC):	Plan	Actual	To Date
Total New & Changed	120		
Maximum Size	200		
Minimum Size	75		

Time in Phase (min.)	Plan	Actual	To Date	To Date %
<i>Planning</i>	20			
<i>Design</i>	60			
<i>Code</i>	60			
<i>Code Review</i>	30			
<i>Compile</i>	15			
<i>Test</i>	15			
<i>Postmortem</i>	20			
Total	220			
Maximum Time	330			
Minimum Time	120			

Defects Injected	Plan	Actual	To Date	To Date %	Def./hour
------------------	------	--------	---------	-----------	-----------

**Planning**

**Design**

**Code**

**Code Review**

**Compile**

**Test**

**Total**

<b>Defects Removed</b>	<b>Plan</b>	<b>Actual</b>	<b>To Date</b>	<b>To Date %</b>	<b>Def./hour</b>
------------------------	-------------	---------------	----------------	------------------	------------------

**Planning**

**Design**

**Code**

**Code Review**

**Compile**

**Test**

**Total**

Comments: \_\_\_\_\_

\_\_\_\_\_

**Table 3.1 Time Recording Log**

Student \_\_\_\_\_ Date \_\_\_\_\_  
 Instructor \_\_\_\_\_ Class \_\_\_\_\_

Date	Start	Stop	Interruption Time	Delta Time	Activity	Comments	C	U

**Table 12.2 Defect Recording Log**

Student \_\_\_\_\_ Date \_\_\_\_\_

Instructor \_\_\_\_\_ Program # \_\_\_\_\_

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Description  
:

---

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Description  
:

---

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Description  
:

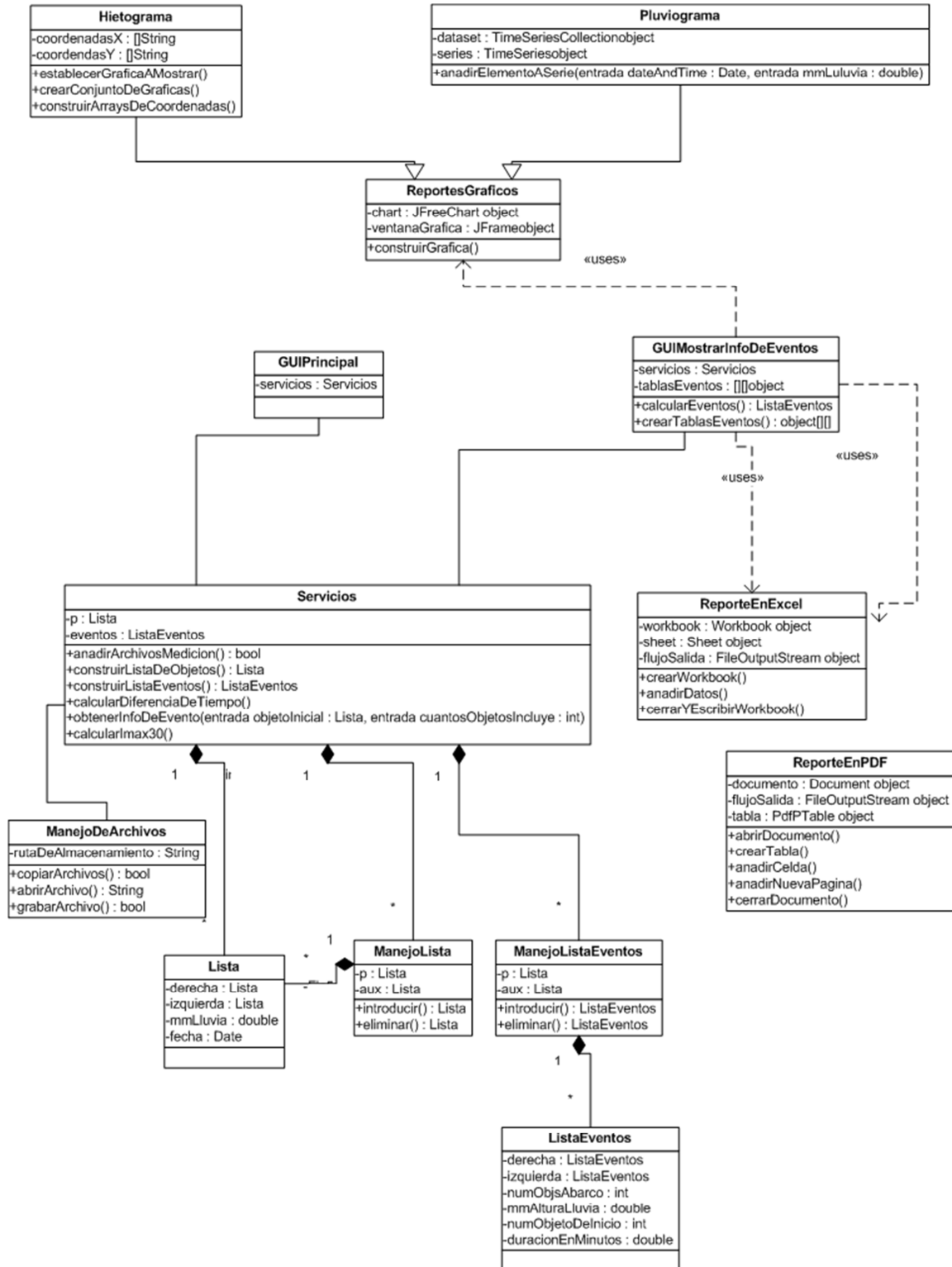
---

Date	Number	Type	Inject	Remove	Fix Time	Fix Defect
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Description  
:

---

# Anexo C





# ***Anexo D***

Oficio de acuse de recepción oficial del artículo “**Plataforma integral para el desarrollo y automatización de un pluviógrafo thies**” de la Revista Tecnología y Ciencias del Agua IMTA.



Jiutepec, Morelos a 14 de diciembre de 2014

ING. IRVING ALEJANDRO GONZÁLEZ MANRÍQUEZ.  
DR. OZIEL LUGO ESPINOSA  
DR. EN ED. JOEL AYALA DE LA VEGA  
UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO.

#### AUTORES

Con esta fecha recibimos en nuestra redacción su artículo "Plataforma integral para el desarrollo y automatización de un pluviógrafo thies" Apreciamos cumplidamente su colaboración y aprovechamos para comentarles que, conforme con la Política Editorial de la revista, el texto será sometido a un proceso de arbitraje, de cuyo resultado les informaremos oportunamente.

Sin más por el momento, les reitero nuestro agradecimiento por habernos distinguido con su confianza y esperamos seguir contando con su valiosa participación.

ATENTAMENTE  
LA COORDINADORA EDITORIAL

LIC. HELENA RIVAS LÓPEZ

# Bibliografía

Al-Kadi, T. (2013). Arduino Wi-Fi network analyzer.

Arduino. (20 de Febrero de 2013). *Arduino*. Obtenido de <http://arduino.cc/en/Main/Products>

Arduino. (20 de Febrero de 2013). *Arduino Sketch*. Obtenido de <http://arduino.cc/en/pmwiki.php?n=Tutorial/Sketch>

Atmel. (13 de Mayo de 2013). *8-bit Atmel microcontroller with 64k/128k/256k Bytes In-system Programmable Flash*. Obtenido de <http://www.atmel.com/Images/doc2549.pdf>

Balogh, R. (2011). Educational Robotic Platform based on Arduino.

Banzi, M. (2011). *Getting Started with Arduino*. O'REILLY.

Barker, A. (1966). *Patente nº 3243999*. U.S.A.

Capra, M., Escala, D., & Morisio, M. (s.f.). Adapting the PSP to an SME context.

Chappell, R. (1938). *Patente nº 2118459*.

Chow, V. (1964). *Handbook of Applied Hydrology*. Mc GrawHill.

Comisión Nacional del Agua. (2013). Compendio Estadístico de Administración del Agua.

Comisión Nacional del Agua. (2013). Estadísticas del agua en México. México D.F.

Delitala, A., Deidda, R., Mascaro, G., Piga, E., & Querzoli, G. (2010). Systematic recover of long high-resolution rainfall time series recorded by pluviographs during the 20th century. *Geophysical Research*.

Evans, B. (2011). *Beginning Arduino*. OREILLY.

- Fiser, O., & Kveton, V. (2006). Stability of rain-rate distribution: how many years to measure?
- Jennings, S. (2006). A High Resolution Point Rainfall Model Calibrated to Short Pluviograph or Daily Rainfall Data.
- Koenka, J., Saiz, J., & Hauser, P. (2014). Instrumentino: An open-source modular Python framework for controlling Arduino based experimental instruments.
- Kotowski, A., Dancewicz, A., & Kazmierczak, B. (2011). Accuracy of measurements of precipitation amount using standard and tipping bucket pluviographs in comparison to hellmann rain gauges. *Environment Protection Engineering*, 23-34.
- León Méndez, A., Hernández González, A., Garrido Monagas, M., & Andalia Gilbert, A. (2013). Captación de lluvia con pluviógrafos de cubeta y su postprocesamiento. *INGENIERÍA HIDRÁULICA Y AMBIENTAL*, 73-87.
- Milone Technologies, I. (3 de Diciembre de 2012). <http://www.adafruit.com/>. Obtenido de <http://www.adafruit.com/datasheets/eTapeApp.pdf>
- Mojardi Gilan, H., Ahmadi, H., Jaafari, M., Bohamta, M., & Salajegheh, A. (64-69). Study of the Temporal Distribution Pattern of Rainfall Effect on Runoff and Sediment Generation Using Rain Simulator (Case Study: Alvand Basin). *World Applied Sciences Journal*, 2010.
- Niranjan Kumar, Y., Hima Bindu, P., Divya Sneha, A., & Sravani, A. (2013). A novel implementation of phase Control Technique for Speed Control of Induction Motor Using Arduino.
- Pérez, G. (20 de Marzo de 2014). *Ciclo Hidrológico*. Obtenido de <http://www.ciclohidrologico.com/>

- Reingart, M. (2012). Plataforma de Desarrollo Rápido de Aplicaciones bajo el Proceso de Software Personal: en búsqueda de agilidad, solidez y disciplina para la Ingeniería de Software. *15 Concurso de Trabajos estudiantiles*.
- reza Ezuan, S., & Babul Salam, K. (2012). Internet Controlled Robotic Arm. *Procedia Engineering*, 1065-1071.
- Reza Ezuan, S., & Babul Salam, K. I. (2012). Wireless Mobile Robotic Arm. *Procedia Engineering*, 1072-1078.
- Rodriguez Vazquez, S. (2013). Arquitectura organizacional para soluciones empresariales de software. *Revista Cubana de Ciencias Informáticas*.
- Salinas, E., Cerpa, N., & Rojas, P. (2011). Arquitectura orientada a servicios para software de apoyo para el proceso personal de software. *Ingeniare. Revista Chilena de Ingeniería*, 40-52.
- Sandoval, G., Molano, J., Mosquera, V. H., & González, L. (2010). PLUVIÓGRAFO ELECTRÓNICO CON TRANSMISIÓN DE DATOS INALÁMBRICA. *Revista Colombiana de Tecnologías Avanzadas*, 67-73.
- Sansom, J. (1987). Digitising Pluviographs. *Journal of Hidrology*, 197-209.
- Sen, R., & Singh, M. (2007). Efect of Rain on Millimeter - Wave Propagation.
- Teikari, P., Najjar, R., & Malkki, H. (s.f.). An inexpensive Arduino-based LED stimulator system for vision research.
- Valdivieso, F. O. (2006). Apuntes de clase Hidrología.
- Vannitsem, S., & Naveau, P. (2007). Spatial dependences among precipitation maxima over Belgium. *Nonlinear Processes in Geophysics*, 621-630.

Vinardell Grandal, I. M., Rodríguez Rubio, J. E., & Fagundo Castillo, J. R. (1992). SAPHID: Sistema automatizado para el procesamiento de datos hidrológicos. *Taller Internacional sobre cuencas experimentales*, 183-188.

von Konsky, B., Ivins, J., & Robey, M. (s.f.). Using PSP to Evaluate Student Effort in Achieving Learning Outcomes in a Software Engineering Assignment.

Wikipedia. (14 de Enero de 2013). *Precipitación (meteorología)*. Obtenido de [http://es.wikipedia.org/wiki/Precipitación\\_\(meteorología\)](http://es.wikipedia.org/wiki/Precipitación_(meteorología))

Wilcher, D. (2012). *Learn electronics with Arduino*. Springer.

Wilcher, D. (2012). *Learn Electronics with Arduino*. Technology in action.

Zavala, G., Morán, R., & Blanco, S. (2010). A programming platform for Arduino on physical Etoys.